Department of Computing Sciences

Using Computer Vision to Categorize Tyres and Estimate the Number of Visible Tyres in Tyre Stockpile Images

Author:
Grant Eastwood
s210024747

Supervisors:
Dr Kevin Naudé
Prof. Charmain Cilliers

Submitted in fulfilment of the requirements for the degree of Magister Scientiae in the Faculty of Science at the Nelson Mandela Metropolitan University.

1 August 2016

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Nelson Mandela Metropolitan University will not infringe any third party rights and that I have not previously in its entirety or in any part submitted it for obtaining any qualification.

Date: December 2015

Abstract

Pressures from environmental agencies contribute to the challenges associated with the disposal of waste tyres, particularly in South Africa. Recycling of waste tyres in South Africa is in its infancy resulting in the historically undocumented and uncontrolled existence of waste tyre stockpiles across the country. The remote and distant locations of such stockpiles typically complicate the logistics associated with the collection, transport and storage of waste tyres prior to entering the recycling process.

In order to optimize the logistics associated with the collection of waste tyres from stockpiles, useful information about such stockpiles would include estimates of the types of tyres as well as the quantity of specific tyre types found in particular stockpiles. This research proposes the use of computer vision for categorizing individual tyres and estimating the number of visible tyres in tyre stockpile images to support the logistics in tyre recycling efforts.

The study begins with a broad review of image processing and computer vision algorithms for categorization and counting objects in images. The bag of visual words (BoVW) model for categorization is tested on two small data sets of tread tyre images using a random sub-sampling holdout method. The categorization results are evaluated using performance metrics for multiclass classifiers, namely the average accuracy, precision, and recall. The results indicated that corner-based local feature detectors combined with speeded up robust features (SURF) descriptors in a BoVW model provide moderately accurate categorization of tyres based on tread images.

Two feature extraction methods for extracting features for use in training neural networks (NNs) for tyre count estimations in tyre stockpiles are proposed. The two feature extraction methods are used to describe images in terms of feature vectors that can be used as input for NNs. The first feature extraction method uses the BoVW model with

histograms of oriented gradients (HOG) features collected from overlapping sub-images to create a visual vocabulary and describe the images in terms of their visual word occurrence histogram. The second feature extraction method uses the image gradient magnitude, gradient orientation, and edge orientations of edges detected using the Canny edge detector. A concatenated histogram is constructed from individual histograms of gradient orientations and gradient magnitude. The histograms are then used to train NNs using backpropogation to approximate functions from the feature vectors describing the images to scalar count estimations. The accuracy of visible object count predictions are evaluated using NN evaluation techniques to determine the accuracy of predictions and the generalization ability of the fit model. The count estimation experiments using the two feature extraction methods for input to NNs showed that fairly accurate count estimations can be obtained and that the fit model could generalize fairly well to unseen images.

Acknowledgments

This research would not have been possible if it had not been for the contributions of many individuals. These individuals provided support and guidance that was integral to the success of this project and I would like to extend my gratitude to them for the contributing roles they had in this project.

I would like to thank my supervisors Prof. Charmain Cilliers and Dr Kevin Naudé for advising, and reviewing my work at the various stages of the project. I would also like to thank them for supporting and encouraging me during times of doubt and for guiding me towards the completion of the project.

I would like to thank members of the Computer Science staff for the support they provided and the feedback that was provided.

I would like to thank Dr Percy Hlangothi for organizing the field trip to the REDISA tyre depot for data collection and REDISA for funding this research in full. Without the funding this research would not have been possible.

Contents

List of Figures

List of Tables

List of Acronyms

2D two-dimensional.

3D three-dimensional.

ANN artificial neural network.

BoVW Bag of Visual Words.

BW black and white.

CV computer vision.

DEA Department of Environmental Affairs.

DoG difference of Gaussians.

EBSR entropy based salient region.

FLANN fast approximate nearest neighbours.

GHT generalized Hough transform.

HOG histograms of oriented gradients.

HT Hough Transform.

IIWTMP Integrated Industry Waste Tyre Management Plan.

k-d k-dimensional.

KNN k-nearest neighbours.

LBP local binary patterns.

LoG laplacian of Gaussians.

MMH maximal margin hyperplane.

MSER maximally stable extremal region.

REDISA Recycling and Economic Development Initiative of South Africa.

RGB red, green, and blue.

RSA Republic of South Africa.

SIFT scale invariant feature transform.

SURF speeded up robust features.

SVM support vector machine.

Mathematical Notation

I An image.

$I(x, y)$ Value of pixel intensity at pixel grid location (x, y).

$*$ Convolution operator.

$\Pr(X)$ Probability of occurrence of X.

$\setminus$ Set difference.

$\bigtriangledown^2 f$ Laplace of function f.

$\det(M)$ Determinant of a matrix M.

$\equiv$ Equivalent.

$\in$ Element of.

λ Eigen value.

μ Mean value.

σ Standard deviation.

σ^2 Variance.

$\subset$ Subset.

$\sum$ Summation.

$\text{trace}(M)$ Trace of a matrix M.

$|$ Such that.

$|x|$ Absolute value of x.

$||w||$ Euclidean norm of vector w.

Chapter 1

Introduction

There are currently an estimated 60-100 million waste tyres stockpiled in historical waste tyre stockpiles across South Africa (RSA) with an estimated annual increase of 10 million waste tyres (IOLscitech, 2013; South African Department of Environmental Affairs, 2012). Large tyre stockpiles pose several risks to the environment. These risks include fire hazards which lead to air and soil pollution, as well as posing serious health risks, especially in areas with warmer climates where mosquito-borne diseases such as encephalitis and dengue fever occur (Lula & Bohnert, 2000).

The Waste Act, in Section 28(1) states that if waste affects more than one province or where such an activity is conducted in more than one province then an industry waste management plan must be created (South African Department of Environmental Affairs, 2012). The problem of waste tyres in RSA led to the creation of REDISAs Integrated Industry Waste Tyre Management Plan (IIWTMP). In 2012 the Department of Environmental affairs (DEA) gave notice of approval for REDISAs IIWTMP. REDISA is a non-profit organisation that, through the IIWTMP, aims to set up and manage a sustainable national network for the collection and temporary storage of waste tyres, delivery of waste tyres to recyclers, as well as supporting the development of a waste tyre recycling industry (IOLscitech, 2013; REDISA, 2014).

The IIWTMP provides a hierarchy for the recycling of waste tyres. Figure 1.1 shows a summary of waste tyre hierarchy as it should be after the implementation of the IIWTMP. Tyre producers who are registered with REDISA must pay a waste tyre management fee in order to fund the IIWTMP activities. The fees collected by REDISA will be used to cover the costs of waste tyre management processes. Through marketing and education,

REDISA plans to raise awareness of the importance of extending tyre life and waste tyre management. The marketing and education initiative will include educating consumers on the benefits of proper waste tyre management. These benefits include minimising health risks and other environmental issues as well as the retreading of tyres to extend tyre life.

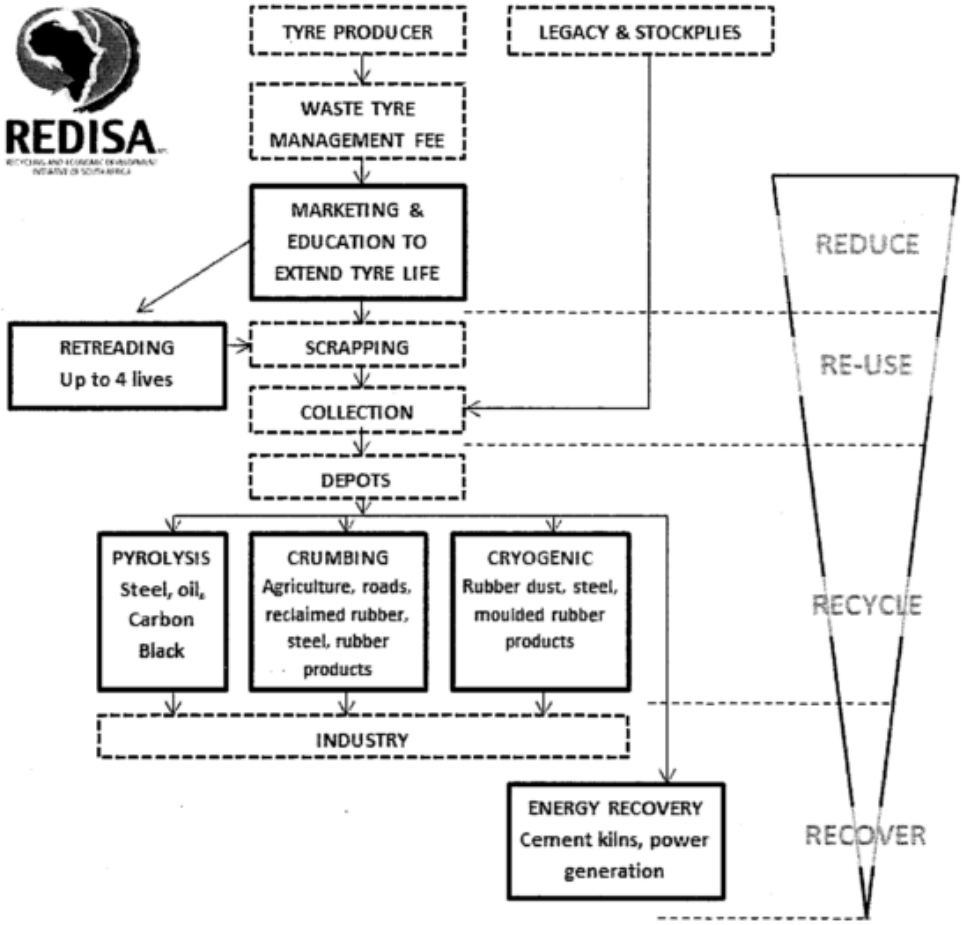


Figure 1.1: Waste tyre recycling hierarchy. (South African Department of Environmental Affairs, 2012).

Once tyres have been collected from either stockpiles or from entities subscribed to the IIWTMP, they are then transported to tyre depots to be stored until they can be sent to a recycling plant to undergo one of the recycling or recovery methods that can be seen in the Recycle and Recover section of Figure 1.1.

Since tyre stockpiles contain large numbers of various types of tyres, which need to be recycled in different ways at different places, different trucks need to collect certain types of tyres in order to ensure the correct tyres are delivered to the correct places at the correct time. If information on a stockpile such as types of tyres, number of tyres, and tyre type ratios is unknown, then decision making during resource allocation and planning can be difficult. This research focuses on using computer vision techniques that can be used to extract information from waste tyre stockpile images to support planning of the Collection phase seen in Figure 1.1.

Computer vision is described as the enterprise of automating and integrating a wide range of processes and representations used for vision perception (Sebe & Lew, 2003a). The topics within the field of computer vision that are included in this research include image pre-processing, object recognition for categorization, and count estimation techniques. The objective of image pre-processing is to correct defects in digital images and to prepare and possibly simplify images to be used as input for another operation. Some of the most commonly used image pre-processing techniques include noise suppression, thresholding, line and edge detection, and segmentation (Zhou *et al.*, 2010). Object recognition methods are investigated in order to address the categorization of individual tyres and is most commonly achieved using classification. Count estimation methods are investigated to produce estimations of the number of visible tyres in tyre stockpiles. Count estimation of objects in images has been approached from the perspectives of counting by detection, segmentation, or regression.

1.1 Problem Decomposition

The following problem statement describes the research problem. The problem statement is used to keep research efforts focused. The problem is to:

Identify and evaluate computer vision concepts and algorithms for categorizing individual tyre images and for estimating the number of visible tyres in tyre stockpile images.

In order for tyre collection to be an effective and efficient process, specific information about the stockpile of collected tyres needs to be known. The information that is required by REDISA for effective and efficient waste tyre collection includes (South African Department of Environmental Affairs, 2012):

1. Categories of waste tyres,
2. Details of the stockpile owner (if ownership information can be obtained),
3. Registration with the Department of environmental affairs,
4. Physical address of the stockpile / GPS coordinates, and
5. Estimation of the number of tyres per stockpile.

This dissertation focuses on determining the categories of tyres (1) and estimating the number of tyres visible in images of a waste tyre stockpiles (2) using computer vision. To provide an understanding of the problem of extracting the aforementioned information about waste tyre stockpiles from images, the problem is decomposed as follows:

1. The categorization of tyre images containing single isolated tyres;
2. Estimating the number of visible tyres in randomly arranged tyre stockpiles.

Tyre stockpiles that do not follow an orderly arrangement will typically contain tyres that are partially occluded in a number of different ways in the acquired images. In addition to the partial occlusion problem, the appearance of individual tyres can vary with regard to their individual rotation and sizes which may also be impacted by the viewpoint of the camera as well as the positioning of individual tyres. In determining suitable methods to recognize tyres, consideration must be given to the level of invariance of the recognition methods with regards to rotation, scale, viewpoint, and partial occlusion.

1.2 Thesis statement

The following thesis statement will be used to guide the research towards answering the research questions and meeting the research objectives.

Computer vision techniques can be used to categorize individual tyres and obtain estimations of the number of visible tyres in tyre stockpile images.

The thesis statement defines the hypothesis that this dissertation aims to prove. The hypothesis defined by the thesis statement is closely linked to the problem statement in that only through identifying and assessing appropriate computer vision concepts and algorithms in the context of tyre categorization and count estimation, can the hypothesis be proven.

1.3 Research Questions

The research questions to be answered are aimed at finding an efficient and effective way to use computer vision and machine learning algorithms to extract information about tyres and tyre stockpiles from images. The following main research question will be used to guide this research:

How can computer vision be used to categorize individual tyres and estimate the number of visible tyres in tyre stockpile images?

The main research question is used to guide the research towards proving or disproving the thesis statement. In answering the main research question, the problem statement is also addressed. Through identifying, applying, and assessing computer vision concepts and algorithms, the main research question can be answered.

1.3.1 Research Objectives

The main research question and the problem statement together are used to formulate the research objectives. In order for the proposed project to be successful, the following two research objectives will need to be met. The two research objectives are:

RO_1 To identify methods to categorize individual tyres and estimate the number of visible tyres in tyre stockpile images.

RO_2 To apply and evaluate selected methods for the categorization and counting of tyres in images.

The main research question should be answered through meeting both of the research objectives. The research objectives cover the process that is used to drive the development of this dissertation by separating the work into two main parts.

Research objective RO_1 requires a broad review of available computer vision concepts and algorithms so that suitable concepts and algorithms can be identified. Once suitable computer vision concepts and algorithms are identified, Research objective RO_2 can be addressed. Research objective RO_2 aims to address the suitability of the identified computer vision concepts and algorithms through applying the methods and evaluating their performance in the context of tyre categorization and visible tyre count estimation in tyre stockpile images.

1.3.2 Sub Research Questions

The main research question is broken down into six sub questions. Each of the six sub questions contributes to answering the main research question by focusing attention to a particular aspect of one of the research objectives. There is thus a 1:N relationship between the research objectives and the research questions. The six sub research questions are:

RQ_1 What pre-processing methods are available for preparing images for categorization and count estimation?

RQ_2 What approaches are available for object categorization and object counting from images?

RQ_3 What are suitable image representations for categorization and count estimation?

RQ_4 How can machine learning methods be used for the categorization of individual tyres and estimating the number of visible tyres in tyre stockpiles?

RQ_5 How can experiments be designed to determine the appropriateness of the identified categorization and count estimation methods?

RQ_6 How well do the identified methods work for tyre categorization and visible tyre count estimation?

The sub research questions provide an ordered set of research questions. The six research questions allow a broad review of categorization and count estimation in the computer vision field from various application domains. The broad review then allows the concepts to be identified and assessed for the specific application domain of tyres and tyre stockpiles images.

Research Objective	**Research Question**
RO_1	RQ_1,RQ_2,RQ_3,RQ_4
RO_2	RQ_5,RQ_6

Table 1.1: Relationship between research objectives and questions.

Table 1.1 shows the relationship between the research objectives and research questions. RQ_1 to RQ_4 are related to RO_1. To meet RO_1 the related research questions are answered through a broad literature review and a discussion of the concepts and algorithms

regarding their applicability in the domain of tyre categorization and count estimation. To meet RO_2 the related research questions are answered through applying the identified techniques and evaluating their performance for the tasks of tyre categorization and visible tyre count estimation.

1.3.3 Research [Question] Techniques

The methods used to answer each of the sub research questions is outlined in this section. Each research question outline provides context for each of the research questions and gives an explanation as to why the sub research question was formulated and how it will be answered.

RQ_1 *What pre-processing methods are available for preparing images for categorization and count estimation from images?*

Image pre-processing methods are reviewed in a literature review. The domain of digital image processing is reviewed in Chapter 2 to determine how digital image processing is used for preparing images for subsequent algorithms for object categorization and object count estimation.

RQ_2 *What approaches are available for object categorization and object counting from images?*

The approaches that are available for object categorization and object counting from images are identified and reviewed in a literature review. Out of the identified methods for categorization, the method that is found to be the most suitable out of the candidate categorization approaches is chosen for experimentation. The approaches that are reviewed for object counting in images are narrowed down by reviewing their limitations in the domains in which they have previously been used. The identified limitations are then related to the domain of tyre stockpiles and an appropriate approach is chosen for estimating visible tyre counts in tyre stockpiles.

RQ_3 *What are suitable image representations for categorization and count estimation from images?*

In order to use machine learning methods for categorization and count estimation, images must be represented in a such a way that they can be used by machine learning algorithms. The representations need to be focused on providing information about the image that is relevant to the task of categorization or count estimation. Image representations in terms of image features are reviewed as part of a literature

review and related to the specific tasks of individual tyre categorization and tyre count estimation.

RQ_4 *How can machine learning methods be used for the categorization of individual tyres and estimating the number of visible tyres in tyre stockpiles?*

The machine learning concepts that are identified in the review of categorization and count estimation approaches are reviewed. The machine learning methods are discussed in terms of the inputs that are required, how the algorithms work, and how they are evaluated.

RQ_5 *How can experiments be designed to determine the appropriateness of the identified categorization and count estimation methods?*

Two separate experiments are designed to evaluate the suitability of identified image representations in terms of feature descriptors and their use in machine learning algorithms for the tasks of tyre categorization and visible tyre count estimation. The reason for designing two separate experiments is that the categorization task and count estimation tasks are, although not mutually exclusive, considered separately and thus an experiment is designed for each.

RQ_6 *How well do the identified methods work for tyre categorization and visible tyre count estimation?*

To determine how well the identified methods work for tyre categorization and count estimation, models that are fit to training data for each task are evaluated. The categorization of individual tyres is evaluated using measures for classifier evaluation and the estimation of the number of visible tyres is evaluated in terms of prediction error and the generalization ability of the fit models.

1.4 Scope and Constraints

The scope of this project is limited to the categorization of tyres based on cropped images containing only tyre tread and count estimation of the number of visible tyres in tyre stockpile images in which the stockpile of focus has been manually segmented from the background. A broad literature review is conducted to identify appropriate methods for categorization and count estimation. The identified methods are then applied in the context of tyre categorization and count estimation in images.

The constraints of such a project include constraints that are intrinsic to many computer vision algorithms. The constraints identified with computer vision in the context of this project include dealing with highly cluttered scenes with multiple similarly shaped objects, dealing with high degrees of occlusion, environmental variables in outdoor scenes, and identification of highly worn out tyres.

1.5 Envisioned Contribution

This research aims to find ways to categorize tyres in images and estimate the number of visible tyres in tyre stockpiles images. The envisioned contributions are to provide an automated approach to acquiring information about tyre stockpiles. The required information for effective and efficient logistics planning for tyre recycling (Section 1.1) includes the categories of tyres in tyre stockpiles and an estimation of the number of tyres in tyre stockpiles. This research should form a foundation for future research concerning the use of computer vision to acquire the required information.

The overall envisioned contribution is the identification of computer vision concepts and algorithms that are suitable for achieving the tasks of tyre categorization and visible tyre count estimation from images separately so that future research can be conducted for combining the techniques to automatically acquire the required information from images of tyre stockpiles.

1.6 Chapter Outline

To provide an indication of how each chapter of this dissertation contributes to the research objectives, the diagram in Figure 1.2 has been constructed to guide the reader.

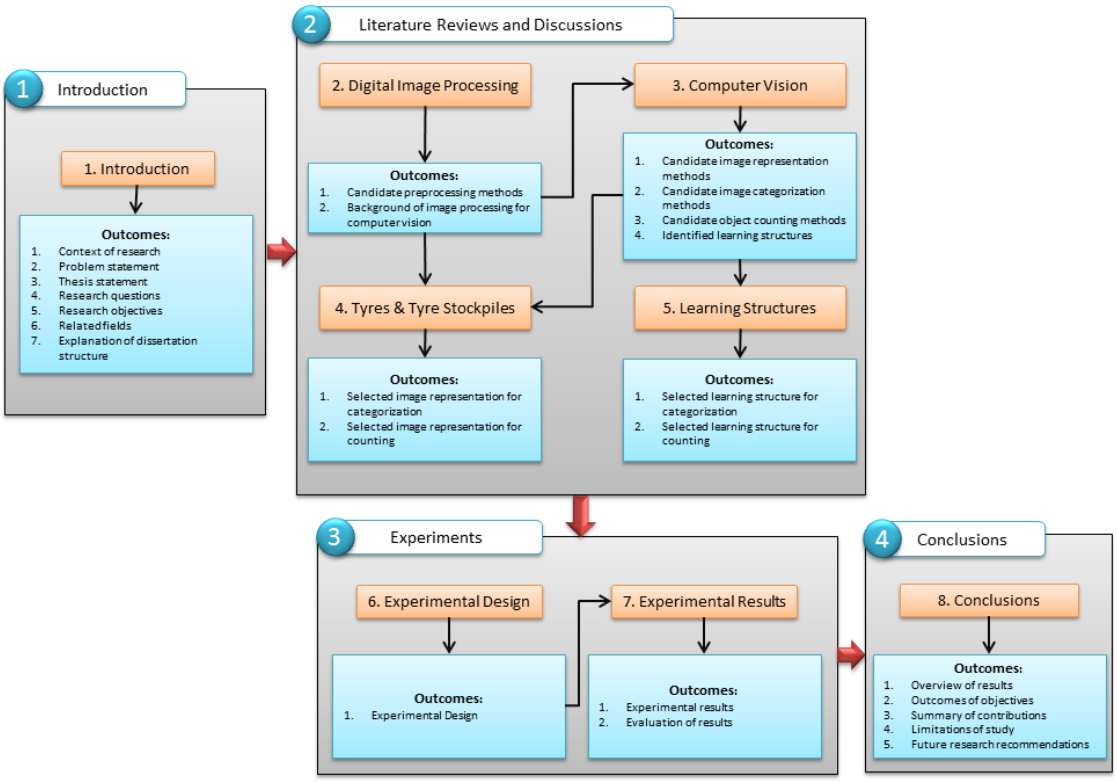


Figure 1.2: Dissertation outline

Figure 1.2 shows the outline of this dissertation. The chapters fall into four main parts, namely the introduction, the literature review and discussions, the experiments, and the conclusions. The introduction provides the context of this research and leads into the literature review and discussions part. The literature reviews and discussions review methods and approaches for digital image processing, computer vision, and machine learning structures and relates concepts in these chapters to the domain of tyres in the tyres and tyre stockpiles chapter. The outcomes from the literature reviews and discussions leads to the experiments part that contains a discussion of the experimental design and the results of the experiments. The experiments part leads to conclusions about the validity of using the selected methods for tyre categorization and count estimation.

The following is an outline of each chapter of this dissertation:

Chapter 1 Introduction - The introduction provides the context for this research. The problem statement and thesis statement are stated and explained. Research questions are

formulated to guide the research and research objectives are set to ensure the research questions are answered. The fields that are related to the research are identified as image processing, computer vision, and machine learning. Finally this chapter gives an outline and explains the remainder of this dissertation.

Chapter 2 Digital Image Processing - Digital image processing is reviewed to provide an overview of methods that are used to pre-process images. This chapter addresses research question RQ_1 by reviewing digital image processing methods for the purpose of preparing images for subsequent algorithms. The main outcomes of the chapter are candidate image pre-processing methods and a background of image processing for computer vision.

Chapter 3 Computer Vision - The Computer Vision chapter identifies approaches for categorization and count estimation from images by addressing research question RQ_2. Image representations in terms of image features that are appropriate for the tasks are also identified and discussed. Candidate categorization and object counting approaches are identified along with the learning structures that are used to achieve the tasks of categorization and count estimation from images.

Chapter 4 Tyres and Tyre Stockpiles - The Tyres and Tyre Stockpiles chapter uses the outcomes of Chapters 2 and 3 to provide a discussion of the candidate categorization and count estimation approaches in the particular domain of tyres and waste tyre stockpiles and addresses RQ_3. The outcomes of the chapter are selected image representations for categorization and count estimation for tyre categorization and count estimation of visible tyres in tyre stockpile images.

Chapter 5 Learning Structures - The Learning Structures chapter describes the machine learning algorithms identified in Chapter 3 for categorization and count estimaiton in more detail. The discussion provides insight to the machine learning algorithms used in the experiments. Research question RQ_4 is addressed through the discussions of machine learning algorithms for categorization and count estimation. The outcome of the chapter is an explanation of the machine learning algorithms used for categorization and count estimation in the domain of tyres and tyre stockpiles.

Chapter 6 Experimental Design - The Experimental Design chapter describes the way in which the selected pre-processing, computer vision, and machine learning algorithms are used to evaluate their use in the task of tyre categorization and tyre stockpile count estimation. The experimental design chapter addresses the fifth research

question RQ_5. Through the design of experiments to evaluate the performance of the identified methods for categorization and count estimation.

Chapter 7 Experimental Results - The experimental results chapter describes the results of the experiments that are conducted. The experimental results are discussed in terms of the performance of the categorization approach and the count estimation approach. The sixth research question RQ_6 is addressed through the presentation of the results for the experiments for categorization and count estimation.

Chapter 8 Conclusions - The Conclusions chapter provides an overview of the experimental results from Chapter 7. The research objectives are reviewed and related to specific sections to ensure that they have been met and the research questions have been answered. The contributions of this work are summarized and the limitations of this study are discussed. Finally, recommendations for future research are given.

Chapter 2

Digital Image Processing

Digital image processing encompasses processes for altering image data or extracting information from images. This chapter aims to investigate digital image processing techniques for pre-processing images in order to answer the first research question:

RQ_1 *What pre-processing methods are available for preparing images for categorization and count estimation from images?*

This chapter describes identified digital image processing methods for the purpose of pre-processing image data. The selected image processing methods have been identified as commonly used pre-processing methods that are used to prepare image data for use in categorization and counting in images. There are four main reasons for pre-processing image data for the purposes of object categorization and count estimation. The reasons for pre-processing image data are (Brahmbhatt & Samarth, 2012; Canny, 1986; Huang *et al.*, 2010; Zhou *et al.*, 2010):

1. Noise suppression,
2. Image Enhancement,
3. Feature Enhancement, and
4. Data reduction.

Colour conversion is discussed in terms of its use for data reduction (Section 2.1). Histogram equalization is often used to enhance image contrast (Section 2.2). Image filtering is discussed for its use in noise reduction and image enhancement (Section 2.3). Image segmentation can reduce the amount of data to be processed by computer vision

algorithms by separating groups of homogeneous pixels (Section 2.4). Edge detection that is used for feature enhancement and data reduction through the identification of large gradient changes is discussed (Section 2.5). Morphological image processing that can be used to enhance features and alter structures in images are discussed (Section 2.6).

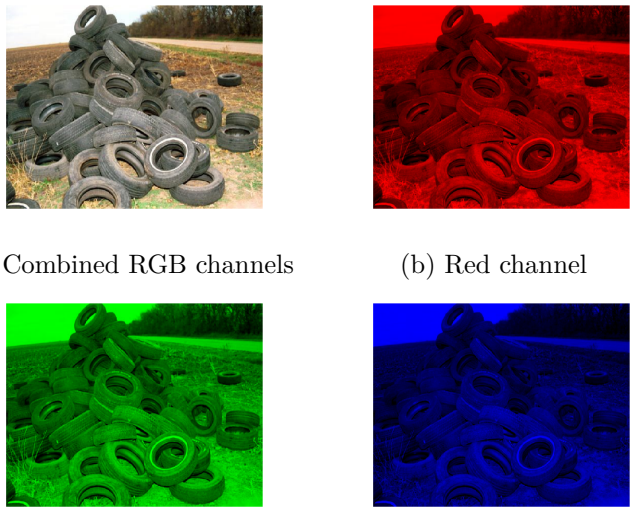
2.1 Colour Conversions

Colour conversions are used to reduce the amount of data to be processed in digital images. In digital imaging three broad categories describing image colour representation are colour, grayscale, and binary images. The *colour depth* measures the amount of colour information that is used to display each pixel of a digital image with colour depths typically in the range of 1 to 64 bits (Frery & Perciano, 2013). Colour-grayscale and grayscale-binary conversions typically result in a loss of information (Cadik, 2008). The reason for the loss of colour information is due to the reduction of bit depth. Section 2.1.1 describes colour images, Section 2.1.2 discusses grayscale images, and Section 2.1.3 discusses binary images. The three image colour categories are discussed in terms of their standard representations and common conversion methods.

In order to formalize the operations that can be performed, the foundational framework by Marr (1982) for vision is briefly described to provide an overview of the notation used in the remainder of this dissertation. In the framework proposed by Marr (1982), an image I is considered to be a function $I(x, y)$ where x is the column index of the image and y is the row index of the image. The value of the function at (x, y) represents the intensity at pixel location (x, y) for the particular image I. In digital images I is often given as a 2D matrix of values and x denotes the column index of I and y denotes the row index of I.

2.1.1 Colour images

Colour images contain more information than grayscale or binary images (Cadik, 2008). The RGB colour model is the standard in computer graphics (Kelda, 2014). The RGB model has a red, green, and blue colour component to describe the colour of each pixel. Colour information can be used to segment objects of a particular colour in images by determining colour correspondences between the object being sought and pixels in an image.



(a) Combined RGB channels (b) Red channel

(c) Green channel (d) Blue channel

Figure 2.1: RGB channels comprising a full RGB colour image of a tyre stockpile

Figure 2.1 shows a full RGB colour image (Figure 2.1a) and the red (Figure 2.1b), green (Figure 2.1c), and blue (Figure 2.1d) components that make up the RGB image. Each of the RGB channels is a 2D matrix with rows and columns corresponding to the image height and width respectively.

2.1.2 Grayscale images

Grayscale images are represented by a single intensity value per pixel. Grayscale images do not contain colour information although they often preserve structural information which is evident from the use of edge detectors. Various algorithms for the conversion between colour images and grayscale images have been proposed (Cadik, 2008). A popular approach for converting from colour to grayscale is to take a computed luminance channel from the colour image and treat the luminance channel as a grayscale representation of the image.

(a) (b)

Figure 2.2: Colour and grayscale versions of the same image

The grayscale image in Figure 2.2 was created by means of a weighted summation of the RBG values at each $I(x,y)_{gray}$ pixel location. The weighted summation is given by (MathWorks, 2016c):

$$I(x,y)_{gray} = (0.2989 \times I(x,y)_{red}) + (0.5870 \times I(x,y)_{green}) + (0.1140 \times I(x,y)_{blue}) \quad (2.1)$$

The resulting $I(x,y)_{gray}$ is equivalent to the Y component which represents the luminance of the YIQ[1] colour model. The constants in Equation 2.1 are the weights for each of the red $I(x,y)_{red}$, green $I(x,y)_{green}$, and blue $I(x,y)_{blue}$ channels. The luminance is often taken as the grayscale representation of the image (Kelda, 2014). The data reduction when converting between RGB and grayscale is a result of the conversion from three dimensions per pixel to a single dimension per pixel.

2.1.3 Binary images

A binary image representation is an image representation in which each pixel can only take on one of two values. Binary images are also known as black and white (BW) images in which pixels are either black or white. One way to produce a binary image is to start off with a grayscale image and transform it according to a threshold α. The value of the threshold α is a proportion of the maximum possible grayscale pixel value. The threshold

[1]The YIQ colour model is a alternative to the RGB colour model for representing colours. Y is the luminance channel and I and Q refer to chrominance coordinates (Pizer & McAllister, 1994).

operation can be stated as (MathWorks, 2016a).

$$I(x,y)_{bw} = \begin{cases} 0, & \text{if } I(x,y)_{gray} < \alpha, \text{where } \alpha \text{ is a threshold value.} \\ 1, & \text{otherwise} \end{cases} \tag{2.2}$$

The input image is a grayscale image. The conversion algorithm evaluates each pixel at location (x, y). If the pixel value is above the threshold then it is set to 1 (white). If the pixel value is below the threshold then it is set to 0 (black).

(a) (b)

Figure 2.3: Grayscale image and resulting binary image after conversion

Figure 2.3 shows a binary image with the threshold set to 50% of the maximum grayscale value. As the threshold value decreases, more of the resulting binary image becomes white while increasing the threshold value results in more of the image becoming black.

Other methods of creating binary images include using edge detection (Section 2.5) where edge pixels are represented by 1 and non-edge pixels are represented by 0. The binary nature of bw images allows them to be stored and evaluated efficiently using a single bit of information per pixel (Kumar & Verma, 2010). Although binary images can be efficiently stored and evaluated, it can be seen in Figure 2.3 that structural information is lost through thresholding when pixels containing valuable structural information all fall below or above the selected threshold and thus all become either black or white.

2.2 Histogram Equalization

Histogram equalization belongs to a class of techniques for image enhancement called *intensity transformations*. The goal of histogram equalization in image processing is to transform a grayscale image's intensity histogram to a more uniformly distributed

histogram (Raju *et al.*, 2013). The transformation to a more uniformly distributed histogram compensates for differences in camera input gains and improves contrast in some cases (Rowley *et al.*, 1998). Through the use of histogram equalization, sections of an image that can hardly be seen are made more visible (Zhou *et al.*, 2010).

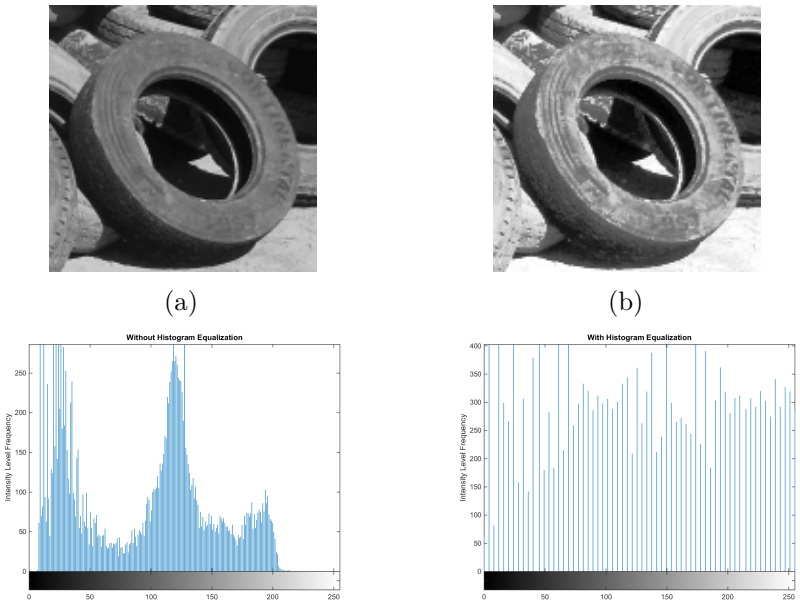


Figure 2.4: Grayscale image before and after histogram equalization

Figure 2.4c shows the pixel intensity histogram for the image in Figure 2.4a. It can be seen that the most commonly occurring gray levels are at the lower and middle of the gray level range. The contrast difference can be seen where the majority of pixels represent the section of the image that shows the shadow cast by the tyre as well as the dark tyre holes in Figure 2.4a. To make the section covered by the shadow area more visible, the pixel intensity distribution is transformed to a more uniform distribution as can be seen in Figure 2.4d with the resulting image in Figure 2.4b. Histogram equalization is performed by the following transformation for an image with grayscale values in the range $[0, L-1]$ (Gonzalez & Woods, 2002):

$$s(k) = \sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0, 1, 2, ..., L-1. \tag{2.3}$$

where n is the total number of pixels in the image, k refers to the k^{th} gray level, n_j is

the number of pixels at the j^{th} gray level, L is the number of possible intensity values that pixels in the image can assume. Three disadvantages are associated with traditional histogram equalization techniques, namely the lack of a mechanism to control the degree of enhancement, over-enhancement resulting in visual artefacts, and changes to the average luminance of the image (Wang, 2007).

2.3 Image Filtering

Image filtering is used for image enhancement, noise reduction, edge detection, and sharpening. Image filters are broadly separated into two groups, namely linear filters and non-linear filters (Zhou *et al.*, 2010). Linear filters include mean and Gaussian filters (Section 2.3.1) and non-linear filters include median filters (Section 2.3.2). Linear filters are used to replace a pixel with a value resulting from a linear combination of input pixels (Rao *et al.*, 2006), the input pixels are typically in a neighborhood around the pixel defined by a filter kernel (or mask) centred at the pixel being evaluated. Linear filters can be applied by using convolution. The convolution operator is given by (Zhou *et al.*, 2010):

$$s(x,y) = \sum_{m=-M/2}^{M/2} \sum_{n=-N/2}^{N/2} h(m,n)I(x-m,y-n) \tag{2.4}$$

where I is the input image, h is the filter, $s(x,y)$ is the resultant pixel at location (x,y), M and N are the filter width (columns) and height (rows) respectively. A filter kernel of finite size is scanned across the image and the pixel corresponding to the centre of the filter kernel is replaced by the result of the convolution (Rao *et al.*, 2006). Non-linear filters replace the pixel corresponding to the centre pixel of the kernel with a value that is not determined by a linear combination of the pixels in the neighbourhood, for example the minimum, maximum, or median value in the neighbourhood.

2.3.1 Gaussian Filtering

Gaussian filtering is used to remove detail and noise (Rao *et al.*, 2006). Gaussian filtering causes a blurring effect as a result of each pixel becoming a weighted summation of the pixels in its neighbourhood. Detail and noise are typically sharp gradient changes in the image and a Gaussian filtering results in the gradient changes becoming smoother. The weights in the filter kernel are representative of a discrete 2D Gaussian function. The amount of blurring is dependent on the size of the filter kernel and the value of the

standard deviation σ of the corresponding Gaussian distribution.

$$\frac{1}{115}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Figure 2.5: Example of a 5×5 Gaussian filter kernel with $\sigma = 1.4$ (Rao *et al.*, 2006)

Figure 2.5 shows a 5×5 Gaussian filter kernel that is a discretized form of a Gaussian function with standard deviation $\sigma = 1.4$. Although the Gaussian filters and other linear filters can be used to suppress noise in images (Gupta, 2011), linear filters often do not preserve edges (Hamza *et al.*, 1999).

(a) (b)

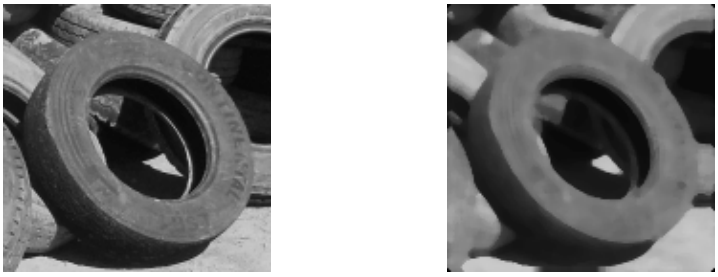
Figure 2.6: Effect of Gaussian filtering with kernel size 5×5 and $\sigma = 1.4$

Figure 2.6b shows the blurring effect achieved by applying a Gaussian filter to the image in Figure 2.6a. Although small discontinuities in the image have been removed, the edges around the tyre have been blurred resulting in a loss of finer structural detail in the image.

2.3.2 Median Filtering

The median filter replaces the current pixel being evaluated with the median pixel value in the surrounding neighborhood. The min and max smoothing filters work in a similar way by replacing each pixel with the min or max pixel value from the neighbourhood.

Median filters have some advantages over linear filters when used for smoothing images. The advantages are edge preservation and efficient noise attenuation with robustness against impulsive-type noise (Hamza *et al.*, 1999).



(a) (b)

Figure 2.7: Effect of median filtering with a 5×5 kernel

Figure 2.7b shows the result of applying a median filter to the image shown in Figure 2.7a. In comparison to the Gaussian filtered image in Figure 2.6b, the small discontinuities in the image have been removed but the edges forming the tyre boundary have been preserved.

2.3.3 Image Sharpening

Image sharpening is used to enhance edges and other discontinuities in images. It can be seen as the opposite of blurring. One common way to sharpen an image is to use spatial differentiation (Gonzalez & Woods, 2002). This is done by using the result of an operator that finds edges and other discontinuities. The 2D Laplace operator can be used to achieve this. The 2D Laplacian operator is given as:

$$\bigtriangledown^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{2.5}$$

where $\bigtriangledown^2 f$ is the Laplacian of the function f, f is a function of x and y. For an image, the equivalent of summing the two components in the context of an image I is given as:

$$\bigtriangledown^2 I = [I(x+1,y) + I(x-1,y) + I(x,y+1) + I(x,y-1)] - 4I(x,y) \tag{2.6}$$

Figure 2.8 shows filter kernels based on Equation 2.6. Filtering the image with the filter kernels results in an image where edges and other discontinuities are shown with zero values where there are no edges or image discontinuities.

0	1	0
1	-4	1
0	1	0

(a)

0	-1	0
-1	4	-1
0	-1	0

(b)

Figure 2.8: Example of a 3×3 Laplacian filter kernel

The result of applying the Laplacian filter is then combined with the image to enhance edges and other discontinuities.

$$F(x,y) = \begin{cases} I(x,y) - \bigtriangledown^2 I(x,y) & \text{if Laplace kernel centre negative} \\ I(x,y) + \bigtriangledown^2 I(x,y) & \text{if Laplace kernel centre positive} \end{cases} \tag{2.7}$$

where $F(x,y)$ is the sharpened image function. The choice of adding or subtracting $\bigtriangledown^2 f(x,y)$ to or from $I(x,y)$ depends on the implementation of the Laplacian kernel used in the convolution step of the filtering process (Gonzalez & Woods, 2002).

(a) (b) (c)

Figure 2.9: Image sharpened by subtracting the result of the Laplacian filtering

Figure 2.9b shows the result of applying a Laplacian filter to the image in Figure 2.9a. Each pixel in the image in Figure 2.9b is subtracted from the original image in Figure 2.9a. The result of the subtraction can be seen in Figure 2.9c. It can be seen in Figure 2.9c that edges and other discontinuities have become more apparent, specifically on the side wall of the tyre and on the tyre treads.

2.4 Image Segmentation

Image segmentation is defined as partitioning an image into homogenous regions by assigning labels to pixels belonging to homogenous regions (Roerdink & Meijster, 2000; Tulsani, 2013). Many detection, recognition, and counting applications rely on an initial segmentation of the objects so that further object detection or recognition processes can be carried out on the segmented regions or the regions can be counted, often separating objects from the background or from each other (Arteta & Lempitsky, 2012, 2013; Chan *et al.*, 2008; Kong *et al.*, 2005, 2006; Ryan *et al.*, 2009). The following section describes segmentation techniques for general image segmentation. Otsu thresholding is used to find the best separation for two classes of pixels based on image intensities (Section 2.4.1). The watershed transform treats images as topographic surfaces and floods the image from local minima, resulting in image regions being segmented into regions representing catchment basins in the topographic surface (Section 2.4.2). K-means clustering is reviewed for its use in grouping pixels according to their colour values (Section 2.4.3). Finally, the Chan-Vese segmentation method which optimizes a fitting energy function to optimize the fit of a curve around objects in images is reviewed (Section 2.4.4).

2.4.1 Otsu Threshold

The Otsu thresholding method makes use of the gray-level histogram of an image to exhaustively evaluate threshold levels to find an optimal threshold to separate two classes of pixels in an image (Otsu, 1979). Otsu thresholding aims to find the best separation of two classes of pixels, $C_0 = [1, ..., k]$ and $C_1 = [k + 1, ..., L]$, by maximizing the between-class variance. Given a gray-level image, L represents the number of possible values that image pixels could assume. To find the optimum threshold value the gray-level histogram is normalized to a probability distribution according to,

$$p_i = \frac{n_i}{n} \tag{2.8}$$

where p_i is the probability of a pixel in the image having gray-level i. n_i denotes the number of pixels that take on the value associated with gray-level i. The sum of all n_i components up to n_k, being equal to the total number of pixels in the image or considered image region, is given by n. The probabilities of the class occurrence can then be calculated from the probability distribution using:

$$\omega_0 = \Pr(C_0) = \sum_{i=1}^{k} p_i = \omega(k) \tag{2.9}$$

$$\omega_1 = \Pr(C_1) = \sum_{i=k+1}^{L} p_i = 1 - \omega(k) \tag{2.10}$$

and the class means μ_0 and μ_1 can be calculated by:

$$\mu_0 = \sum_{i=1}^{k} i \Pr(i|C_0) = \sum_{i=1}^{k} \frac{ip_i}{\omega_0} = \frac{\mu(k)}{\omega(k)} \tag{2.11}$$

$$\mu_1 = \sum_{i=k+1}^{L} i \Pr(i|C_1) = \sum_{i=k+1}^{L} \frac{ip_i}{\omega_1} = \frac{\mu_T - \mu(k)}{1 - \omega(k)} \tag{2.12}$$

The between class variance can then be calculated by:

$$\sigma_B^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2 \tag{2.13}$$

The optimum value for the threshold value, k^* is:

$$\sigma_B^2(k^*) = \max_{1 \leq k < L} \sigma_B^2(k) \tag{2.14}$$

The between-class variance is maximized by separating the classes at various thresholds and taking the threshold which gives the greatest between-class variance.

(a) (b) (c)

Figure 2.10: Tyre image segmented using the Otsu threshold value

Figure 2.10b shows the result of applying the threshold value found by the Otsu method to the image in Figure 2.10a. The threshold is applied according to Equation 2.2 to produce a binary image (Section 2.1.3). The image in Figure 2.10c is a representation of the normalized Euclidean distance from each white pixel in the Otsu threshold image (Figure 2.10b) to the nearest black pixel, giving a pixel intensity in [0,1]. The image representing the normalized Euclidean distance from each white pixel to the nearest black pixel can be used as input to the watershed algorithm (Section 2.4.2).

2.4.2 Watershed Transform

The watershed transform is regarded as a region-based image segmentation technique (Roerdink & Meijster, 2000). When segmenting an image using the watershed transform, a gray-level image is treated as a topographic surface where the intensity values $I(x, y)$ indicate the height of the surface at each pixel location. The watershed transform then floods the topographic surface from local minima, resulting in two sets of components, namely catchment basins and watershed lines.

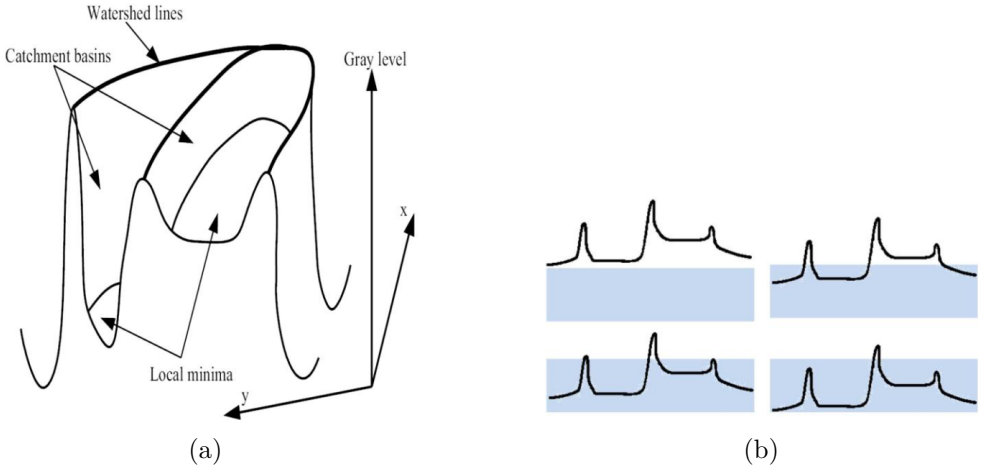


Figure 2.11: Visualization of watershed model and flooding simulation (Tulsani, 2013)

Figure 2.11a shows how a gray-level image is viewed as a topographical surface. The local minima are located at the bottom of the catchment basins and the watershed lines occur along the line where the water from two catchment basins meet. The watershed transform for segmentation results in enclosed regions with boundaries that are a single pixel wide. Enclosed single pixel wide boundaries also allow the watershed transform to be used as an edge detection technique that results in closed contours unlike many other edge detection strategies. The watershed algorithm has been used as a means to separate overlapping objects (Verma, 2013). Disadvantages of the watershed transform is that it tends to over-segment images (Parvati *et al.*, 2008) due to multiple local minima (Verma, 2013), noise and other image irregularities as well as performing poorly for segmenting regions that are thin structures or have low contrast (Tulsani, 2013). To reduce over-segmentation, the watershed transform algorithm can be initialized with markers that serve as starting locations for the flooding algorithm.

(a) (b)

Figure 2.12: Application of watershed algorithm to image of a tyre.

Figure 2.12a shows the source image used as input for the segmentation. Figure 2.12b shows the result of applying a watershed transform to the image shown in Figure 2.10c. The distance transformed image is flooded from the dark regions and watershed lines occur where the flooding of the catchment basins meet.

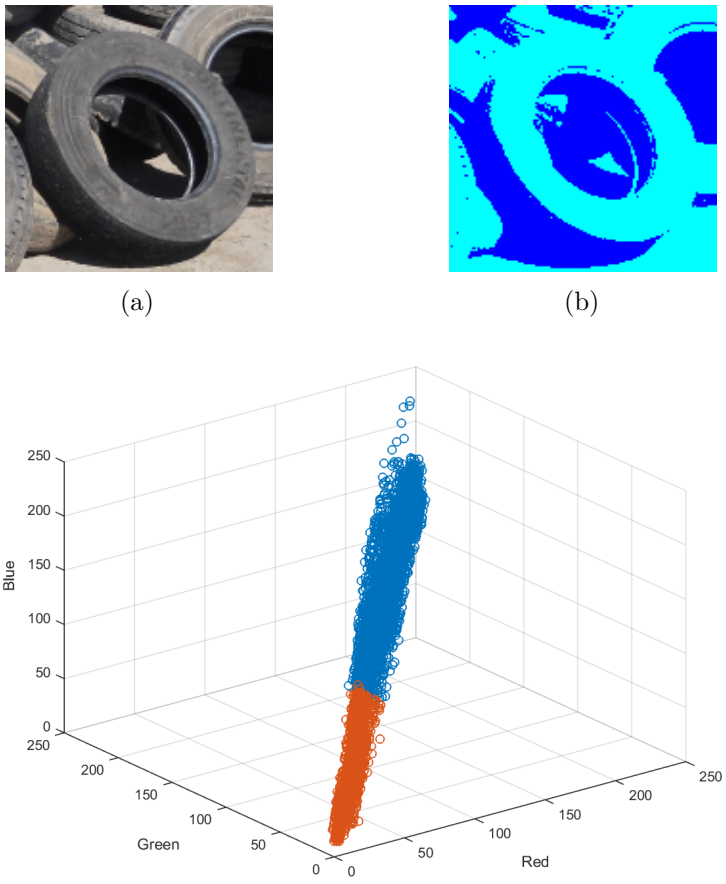
2.4.3 K-Means Clustering

K-Means clustering is a data clustering technique that can be used to cluster pixels in an image. The k-means algorithm is used for its simplicity and ease of use in image segmentation (Kamencay *et al.*, 2012). The K-means algorithm attempts to find the cluster centres of data that naturally form clusters using an iterative re-assignment of cluster centres (Bradski & Kaehler, 2008). For an RGB image, each data point can be specified as a three-dimensional vector containing the RGB values associated with a particular pixel. Each pixel thus forms part of one of the k clusters in a three-dimensional space. The algorithm proceeds as follows (Bradski & Kaehler, 2008):

1. Take input containing the data set to cluster and the desired number of clusters K.
2. Assign k initial cluster centres by random selection of k distinct RGB colour values.
3. Associate each data point with the nearest cluster centre.
4. Recalculate cluster centres to the centroid of their clusters.
5. Return to step 3 until cluster centres do not change.

Although the k-means clustering algorithm is a popular choice for clustering and segmentation, there are some disadvantages associated with its use. The first disadvantage is that the number of clusters must be specified beforehand. The result of the algorithm

may differ depending on the initialization of the cluster centres. The algorithm does not take into account the between or within-cluster variance.



(a) (b)

(c)

Figure 2.13: K-means clustering of RGB pixel values with $k = 2$

Figure 2.13b shows the result of clustering the RGB pixel values in three-dimensional space using the euclidean distance as a measure of distance between the pixel RGB values. The clustering of pixel values was done with $k = 2$. The results are comparable in this image to the results in the Otsu threshold technique in Figure 2.10b. Figure 2.13c shows the result of the k-means clustering applied to the pixel intensities in the RGB space. It should be noted that the RGB values in 2.13c approximate a line in a 3D space from (0, 0, 0) to (255, 255, 255). The line approximation is due to the source image having a very low variance in the chroma dimensions. In other words, the source image is already almost grey and exhibits little colour diversity.

2.4.4 Chan-Vese Segmentation

Chan & Vese (1999) proposed an active contour model without edges. In many active contour models the aim is to evolve a curve around an object with the evolution being subject to a set of constraints from the image being evaluated. The basic idea is to start with a curve around an object. The curve moves towards its interior normal under some constraints until the object boundary is reached. Many active contour models rely heavily on gradient information in the image to determine where the object boundaries are although this can cause issues due to a lack of continuous object boundaries or noise. The issues include the curve moving past the object boundary or stopping before the object boundary is reached. Chan & Vese (1999) use a fitting energy to determine how well a curve fits around a region of the image. The fitting energy is given by:

$$F_1(C) + F_2(C) = \int_{inside(C)} |u_0 - c_1|^2 dxdy + \int_{outside(C)} |u_0 - c_2|^2 dxdy \tag{2.15}$$

where C is the curve surrounding a region, c_1 and c_2 are the constants representing the averages of the image u_0 inside and outside the curve. The boundary of the object of C is considered the minimizer of the fitting energy.

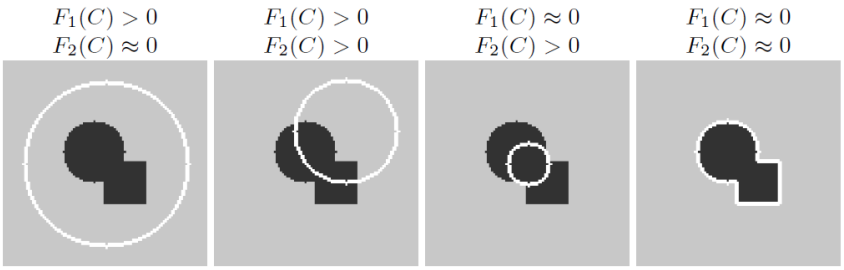


Figure 2.14: Possible cases of the curve position. The fitting energy is minimized when the curve is on the object boundary (Chan & Vese, 1999)

Figure 2.14 shows approximations for the functions $F_1(C)$ and $F_2(C)$ for different curve positions relative to the object being segmented. The energy is minimized when the curve is on the object boundary resulting in $F_1(C) \approx 0$ and $F_2(C) \approx 0$. The Chan-Vese model for segmentation is a particular case of the minimal partition problem from Mumford & Shah (1989) and is formulated and solved using the level set method from Osher & Sethian (1987).

(a) (b) (c)

Figure 2.15: Result of Chan-Vese segmentation with 300 iterations

The Chan-Vese image segmentation model does not require smoothing of the image being segmented. By not requiring a smoothing step, the resulting segmentation contains well defined object boundaries. Object boundaries are not required to be defined by image gradient information or smooth continuous boundaries. Interior contours are also well detected even when only one initial curve is used for evolution and the initial curve does not need to initially surround the object to be segmented. Figure 2.15c shows the segmentation result when the Chan-Vese segmentation algorithm is applied to the original image of the tyre with initial starting points formed by the mask in Figure 2.15a. The initial location of the mask is shown overlaying the image in Figure 2.15b. The segmentation in Figure 2.15c shows the segmentation of the image into three foreground objects surrounded by white pixels and the background.

2.5 Edge Detection

Edge detection refers to the process of detecting pixels in the image for which the intensity with respect to the neighbouring pixels indicates a high in the gradient change (Brahmbhatt & Samarth, 2012). The image gradient is typically calculated in the x and y directions separately and then combined. The gradient orientations of edge pixels can be calculated as a function of the gradient magnitude values used to determine the location of edge pixels. Edge detection serves as a means to simplify the analysis of images by reducing the amount of data that requires processing and preserves only structural information contained in the image (Canny, 1986). Sobel edge detection makes use of special kernels that can be used to determine edge pixels by providing a measure of intensity changes in images (Section 2.5.1). The Canny edge detector is an extension of using the Laplacian kernels for edge detection and uses a set of constraints that result in more accurate and well localized edges (Section 2.5.2).

2.5.1 Sobel

Sobel edge detection is one of the earliest edge detection techniques (Sobel & Feldman, 1968). The idea behind sobel edge detection is that a 3×3 kernel convolved with an image can be used to calculate approximations of image gradient magnitudes and directions. The original image is convolved with two kernels, one to find the image derivative approximations in the x direction and another in the y direction. Two possible implementations of Sobel filter kernels are shown in Figure 2.16.

Figure 2.16: Example of a 3×3 Sobel filter kernel

The gradient image is obtained through convolution with the original image I with the two kernels separately. The resulting gradient image is given as:

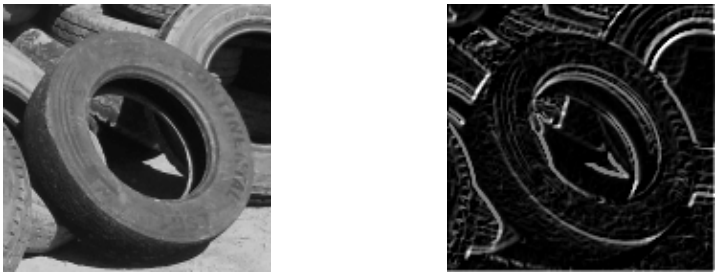
$$I_x = I * k_{S,x} \quad \text{and} \quad I_y = I * k_{S,y} \tag{2.16}$$

The image gradient magnitude approximations can then be obtained by summing the absolute values of I_x and I_y. In addition to calculating the gradient magnitude approximations, the gradient orientation can also be calculated from I_x and I_y. The gradient magnitude approximations and orientation can be calculated as follows:

$$I_G(x,y) = \sqrt{(I_x(x,y))^2 + (I_y(x,y))^2} \approx |I_x(x,y)| + |I_y(x,y)| \tag{2.17}$$

$$I_\theta(x,y) = \arctan\left(\frac{I_y(x,y)}{I_x(x,y)}\right) \tag{2.18}$$

The Sobel edge detector works well for finding edges by determining the gradient magnitudes of an image. Real images often contain noise and other discontinuities that are not necessarily edges. The Sobel edge detector often results in many edges that are not necessarily well localized in relation to the actual edges in the image.



(a) (b)

Figure 2.17: Result of applying the Sobel filter to a tyre image

The resulting edge image in Figure 2.17 shows the result of applying the Sobel filter for edge detection to the image in Figure 2.17a. It can be seen that in addition to finding edge structures, other discontinuities in the image are also found. Each pixel value in the resulting image measures the gradient magnitude of the image at that pixel. A threshold can be applied (Section 2.1.3) to remove pixels where the gradient magnitude is low, resulting in only keeping pixels corresponding to sharp intensity changes in the image.

2.5.2 Canny

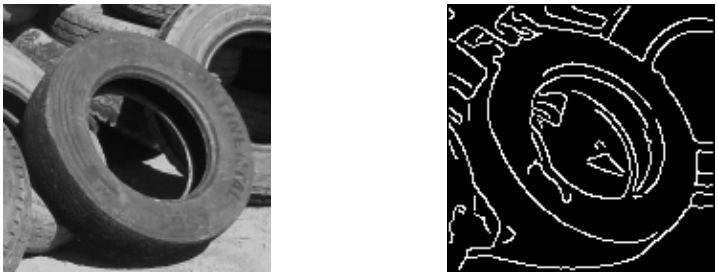
The Canny edge detector is an extension on using the Laplace operator for edge detection. Three performance criteria for edge detection algorithms are described by Canny (1986) and the Canny edge detector attempts to meet the performance criteria. The three performance criteria are:

1. good detection,
2. good Localization, and
3. only one response to a single edge.

Good detection refers to the ability of the edge detection scheme to mark real edge pixels as such without falsely marking non-edge pixels as edge pixels. *Good localization* refers to how close the detected edges are to the true edge in the image. *Only one response to a single edge* refers to the ability of the edge detection scheme to only accept one set of edge pixels that is associated with one edge since edge finding operators may find multiple edges corresponding to only a single true edge.

The Canny edge detection scheme proceeds by computing the first derivatives in x and y which are then combined into four directional derivatives. Pixels where the directional

derivatives are maximum are marked as candidate edge points. Candidate edge pixels are then selected to form contours using a technique called *hysteresis thresholding*. The hysteresis thresholding is performed by applying an upper threshold and a lower threshold to pixel gradients. Pixels are accepted as edge pixels if they have a gradient that is greater than the upper threshold and pixels that have a gradient below the lower threshold are rejected. In the case of a pixel gradient being between the upper and lower threshold values, the pixel is accepted if it is connected to a pixel that is above the upper threshold and must lie in the edge direction associated with the connected pixel otherwise the pixel is rejected (Bradski & Kaehler, 2008).



(a) (b)

Figure 2.18: Canny edge detection applied to a tyre image

Figure 2.18b shows the result of applying the Canny edge detection algorithm to the image in Figure 2.18a. It can be seen in Figure 2.18b that edges are well detected, localized, and there is a single response for each edge.

2.6 Morphological Image Processing

The geometric structures in images often need to be transformed in some way. Morphological image processing contains a subset of image processing algorithms that can be used to transform the geometric structures in images (Rao *et al.*, 2006). Morphological image processing applies concepts from set theory to the domain of image processing to alter or transform structures in images (Parvati *et al.*, 2008). Morphological image processing algorithms require a structuring element which is typically a small set of object points centred around an origin (Dawson-howe, 2014) that describes a simple shape (Parvati *et al.*, 2008). In morphological image processing, the image being processed is typically a binary image although operations can be extended to grayscale images as well. The basic morphological operators are opening and closing and they are both a combination of two other morphological operators called dilation and erosion. The dilation of X by B

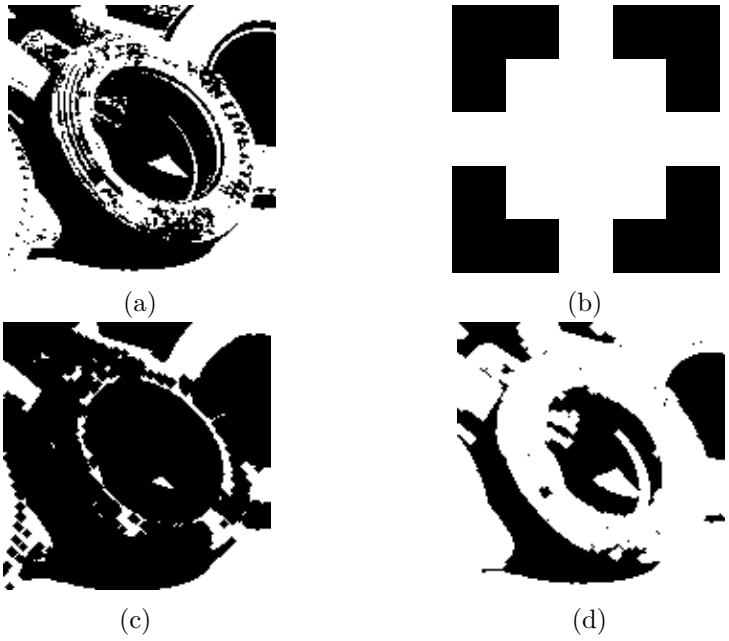
is given as (Stenning *et al.*, 2012):

$$D_B(X) \equiv \{x \in \mathbb{Z}^d \mid B_x \cap X \neq \emptyset\} \tag{2.19}$$

where B_x is the structuring element placed over the image with its origin at x. Dilation can also be stated as: if any element in the structuring element B, when it is centred at x, has a corresponding element in the image then the point x forms part of the dilated image $\mathbb{Z}^d$. The dilation operation results in small holes being closed, narrow gaps being filled in, and objects being enlarged (Dawson-howe, 2014). The erosions of X by B is given as (Stenning *et al.*, 2012):

$$E_B(X) \equiv \{x \in \mathbb{Z}^d \mid B_x \subset X\} \tag{2.20}$$

If the structuring element B centred at x does not fit wholly inside X then the element x does not form part of the eroded image. The erosion of an image results in small points and features being removed as well as objects becoming smaller (Dawson-howe, 2014).



(a) (b) (c) (d)

Figure 2.19: Application of disk shaped structuring element to binary threshold image

Figure 2.19a shows a binary image obtained by applying a threshold to the original image. Figure 2.19b shows an enlarged version of the 5×5 disk shaped structuring element used in the erosion and dilation of the image in Figure 2.19a. The image in Figure 2.19c

shows the result of applying the erosion operator. It can be seen that the erosion removes small and thin white structures as well as reducing the overall size of larger structures representing objects in the image. Figure 2.19d shows the result of applying dilation to the threshold image. It can be seen that dilation results in small black holes being filled in as well as an enlargement of the white structures that represent the objects in the image.

2.7 Conclusions

The image processing methods in this chapter serve as methods to suppress image noise, correct colour distributions, enhance image features, or reduce the amount of data in an image by removing unnecessary data.

Three broad image types were identified and discussed in Section 2.1. Colour conversions from colour representations to grayscale representations result in a loss of colour information but tend to preserve structural information as well as reduce the amount of image data that requires processing in subsequent algorithms. The conversion from grayscale to binary images can result in a loss of structural information if there is no significant contrast between object or structural boundaries although binary images allow image data storage and processing requirements to be reduced.

Image contrast can be improved through the use of histogram equalization (Section 2.2) techniques although traditional histogram equalization techniques can over-enhance images and lead to visual artefacts. Image filtering (Section 2.3) is used for a variety of reasons in image processing with a common application being image smoothing. Linear smoothing filters tend to reduce structural information, for example blurring edges, while non-linear smoothing filters such as the median filter remove noise while preserving sharp intensity changes. Sharpening filters (Section 2.3.3) can be used to enhance the edges and other discontinuities in images.

Four image segmentation methods were identified (Section 2.4). Segmentation using the Otsu threshold method only allows for a binary segmentation of the image with pixels groups either belonging to the background or foreground. The watershed transform is prone to over-segmentation and poor performance where image structures are thin or have low contrast between regions. The Chan-Vese method for segmentation does not depend on edges and is therefore more suitable for cases when object boundaries are not

well defined. Two well known edge detection algorithms were discussed (Section 2.5). The Sobel edge detector results in responses for object edges as well as other discontinuities in images while the Canny edge detection algorithm employs strict criteria for good edge detection that leads to better edge detection.

Morphological image processing (Section 2.6) makes use of set theory concepts to achieve various image processing goals. Dilation and erosion operators can be used to remove from or add to image structures.

In conclusion, the use of image processing algorithms for preprocessing images to prepare them for subsequent computer vision algorithms depends on goals of the computer vision system, the nature of the domain from which the image is sourced, and the features that are required to be extracted from the image under evaluation. It is not evident from the discussion in this chapter alone which digital image processing methods will be suitable for pre-processing tyre and tyre stockpile images for the tasks of categorization and count estimation. In Section 6.1.2 the methods that are selected and applied for preparing tyre images for the task of categorization are stated. In Section 6.2.2 the methods that are selected and applied for preparing stockpile images for the task of count estimation are stated.

Chapter 3

Computer Vision: Concepts and Algorithms

Vision is described as the process of producing a description of the real world, from an image, that is not cluttered with irrelevant information (Marr & Nishihara, 1978; Marr & Poggio, 1976). The field of computer vision encompasses many concepts that are used to extract meaningful information from images. This chapter addresses the second research question:

RQ_2 What approaches are available for object categorization and object counting from images?

In addressing the second research question general strategies for categorization and counting from images are described and candidate image representations in terms of image features are discussed. Each application domain has its own set of specific requirements and constraints (Sebe & Lew, 2003b). This chapter describes candidate computer vision concepts for object recognition, detection, and count estimation.

The outcomes of this chapter are:

1. A background of computer vision concepts,
2. Candidate image representation methods,
3. Candidate object categorization methods,
4. Candidate object counting methods, and

5. Identified machine learning methods for categorization and count estimation.

Section 3.1 gives a background of the two approaches to general vision to provide context for computer vision algorithms and methods. Recognition and detection strategies (Section 3.3) and counting strategies (Section 3.4) are briefly described as an introduction to describing images in terms of their features (Sections 3.2, 3.5, 3.6, and 3.7) as well as providing context for the discussion on related recognition, detection, and counting systems (Sections 3.8, 3.9, and 3.10).

3.1 Top-down and Bottom-up Approaches

The bottom-up approach to vision follows the philosophy that only through restricting certain areas of an image can further high-level analysis of those areas take place (Amit, 2002). In the bottom-up approach an image is scanned in order to create a set of saliency maps. Each saliency map in the set indicates the relative importance of regions defined by local geometrical features and depth (Cavanagh, 1999; Itti & Koch, 2001). The saliency maps in the set are combined into a single saliency map (Itti & Koch, 2001). The combined saliency map is used to determine salient regions in the image where visual attention should be focused.

In a top-down approach stored information about a particular object is used to verify or reject some hypothesis about the image (Cavanagh, 1999). For example, the presence of a finger in an image would suggest a hand may be visible. It is only through relating stored information about a hand to regions of the image that the presence or absence of a hand can be confirmed.

The bottom-up and top-down approaches to vision are complementary (Cavanagh, 1999). In computer vision, often a combination of the two approaches is used. An image is segmented into regions (bottom-up). Regions selected based on the relative saliency of their features are then evaluated to determine their correspondence to a model (top-down).

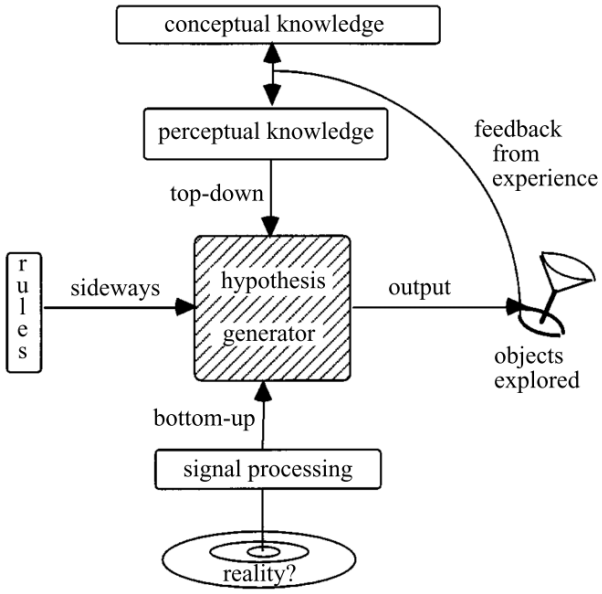


Figure 3.1: Combination of top-down and bottom-up for hypothesis generation in vision. (Gregory, 1997).

Figure 3.1 illustrates how top-down and bottom-up approaches to vision are combined to generate a hypothesis about objects in a scene. The signal forming the visual scene is processed to give low-level image cues. The conceptual and perceptual knowledge form the models from experience that provide high-level information from experience. Gregory (1997) describes the sideways approach which is a combination of top-down and bottom-up approaches. Both the top-down and bottom-up approaches to vision are used to generate hypotheses about objects in a visual scene. The generated hypotheses are then used to explore objects and update knowledge.

3.2 Image Features

Image features are extracted and described to create models for object recognition, detection, and counting. The features extracted from images are in a form that can be used for creating classifiers (for recognition or detection) or regression models (for counting). An image feature refers to a salient image property that can be used to quantitatively characterize the image as a whole or parts of the image. Local features are image properties that are located on single points or within small regions. A local feature quantitatively describes some image property within a local feature's neighbourhood or

small region around it. Since local features typically represent only a part of an image of an object, they allow robustness in occluded and cluttered scenes (Mikolajczyk & Schmid, 2002). According to Roth & Winter (2008), local features are required to be robust to illumination changes, noise, scale changes, affine transformations, rotation, and changes in viewing direction in order for robust object recognition. A global feature describes the contents of an image or image region as a whole. In contrast to local features, global features either take all pixels over the image or image region into account or describe a global characteristic, for example the shape, size, and eccentricity of an image region containing an object.

3.3 Recognition and Detection Strategies

In the context of detection the hypothesis is that an image region either contains or does not contain a particular object. Recognition aims to determine which category an image or image region belongs to while detection aims to determine if an object is present in an image and where it is located. Both recognition and detection require the extraction of visual feature descriptions. The feature descriptions can include descriptions such as edge descriptions, colour descriptions, or gradient orientation descriptions that can be used to create a model for an object or scene. The level of categorization in recognition is determined by two factors, namely what level of categorization is required for the categorization to be meaningful and what level of categorization can be achieved based on the visual appearance of objects between the categories. Solutions have been proposed to the detection and recognition problems with each solution following one or a combination of the following approaches (Gallo & Nodari, 2011):

- Bag of Visual Words Models,
- Parts and Structures Models,
- Discriminative Methods, and
- Combined Segmentation and Recognition.

The four approaches to recognition and detection are briefly described in the remainder of this section to provide context to the discussion on describing images in Section 3.5 to Section 3.7.

In the *bag of visual words model* (Csurka *et al.*, 2004) the image is treated as a document that is made up of a number of visual "words" (image patches). Various works can be found in computer vision literature that use a variation of the bag of visual words model (Ferrari *et al.*, 2008; Lazebnik *et al.*, 2006; Li Fei-Fei & Pietro Perona, 2005; Tongphu *et al.*, 2009). Any mechanism that consistently finds local image structures (local features) can be used to identify appropriate image patches although the two most popular choices are the Scale-Invariant Feature Transform (SIFT) (Lowe, 1999) and Speeded Up Robust Features (SURF) (Bay *et al.*, 2006). A query image can then be matched to other images containing the target object through classification or by computing their similarities in the same way documents are matched using histogram similarity measures.

Parts and structure models refer to a broad category of object detection algorithms in which objects are decomposed into their parts. Once the object parts are defined by some model, various regions of the image are searched separately for each object part. A model of the geometric structure of the object parts can either be learned or explicitly defined (Grauman & Leibe, 2010). Object recognition literature on work that falls into the category of parts and structure models can be found in Crandall *et al.* (2005); Felzenszwalb *et al.* (2008, 2010); Fergus *et al.* (2003); Fischler & Elschlager (1973).

Discriminative methods make use of various training or supervised learning techniques such as support vector machines (SVMs) (Cherkassky, 1997) or boosting (Torralba *et al.*, 2004) with various descriptions of the image representation of the object as a whole (global features) such as histograms of oriented gradients (HOG) (Dalal & Triggs, 2005), Haar-like features (Viola & Jones, 2001), or local binary patterns (LBP).

Combined segmentation and recognition models aim to recognize and segment objects, thus providing an accurate localization of the object by defining a boundary around the pixels in the image that are associated with the target object. Multiple approaches to combined segmentation and recognition have been proposed and can be found in Flohr & Gavrila (2013); Leibe & Schiele (2003); Leibe *et al.* (2004); Nevatia (2008); Tu *et al.* (2005).

3.4 Counting Strategies

The counting problem in computer vision can be defined as estimating the overall number of objects of a particular object instance that are visible in a still image or video frame

(Lempitsky & Zisserman, 2010). There are three main approaches to counting objects in images, namely (Lempitsky & Zisserman, 2010):

1. Counting by detection,
2. Counting by regression, and
3. Counting by segmentation.

A variety of solutions to the counting problem have been proposed. The proposed solutions are in the fields of surveillance for people counting (Chan & Vasconcelos, 2009; Chan *et al.*, 2008; Idrees *et al.*, 2013; Junior *et al.*, 2010; Marana *et al.*, 1998; Ryan *et al.*, 2009), car counting (Arteta & Lempitsky, 2014; Tongphu *et al.*, 2009), and in medical imaging for cell counting (Arteta & Lempitsky, 2012, 2013; Lempitsky & Zisserman, 2010; Tulsani, 2013).

Counting by detection refers to the process of using a visual object detector to detect individual instances of an object in an image and then counting the number of found instances. The major disadvantage with this approach is that the object detection problem is far from being solved, especially for overlapping and/or occluded object instances (Lempitsky & Zisserman, 2010). Tongphu *et al.* (2009) make use of a sliding window approach with a bag of visual words model for detecting individual instances of cars in aerial image views of parking lots. Arteta & Lempitsky (2012) count cells by detection using maximally stable extremal region (MSER) trees to localize candidate cell regions and then classify them. They show that binary classification of MSER regions can be used and that structured learning can be used to improve counting performance.

Counting by regression is the process of learning a mapping from a set of global image features to an overall object count for the image (Lempitsky & Zisserman, 2010). Counting by regression or density estimation avoids the difficult task of object detection and thus individual objects are not localized in the query image (Lempitsky & Zisserman, 2010) although count estimates can be used to improve the results of counting by detection (Rodriguez & Laptev, 2011). A detection system can be provided with an estimation of the number of object instances to search for in a query image.

Counting by segmentation refers to the partitioning of an image into a number of heterogeneous segments where a segment is considered to be a set of spatially connected group of pixels or groupings of pixels according to given pixel labels (Tulsani, 2013).

Counting by segmentation is considered to be a combination of counting by detection and counting by regression. Counting by segmentation typically involves segmenting objects into various clusters and then using the global properties of individual clusters for regression (Lempitsky & Zisserman, 2010). Chan *et al.* (2008) use counting by segmentation and regression by segmenting out individual crowds of people and learning a mapping from global features within each segmented region to the number of people within that region. Ryan *et al.* (2009) use a similar approach in which foreground blob segments are extracted and a mapping from local features to the number of people present in each region is learned. Tulsani (2013) uses a marker-based watershed transform to segment and count individual cells.

3.5 Local Feature Detectors

A local feature detector is used for the detection of local features in images. Local feature detectors are also referred to as region of interest detectors or interest point detectors. A region of interest detector detects a collection of pixels based on some criteria unique to the detector that is used. Interest point detectors refer to local feature detectors that detect specific pixels or points (Grauman & Leibe, 2010; Roth & Winter, 2008). For robust object recognition, local feature detectors are required to find local regions in a repeatable manner. Repeatable refers to whether the same feature will be found in two different images of the same scene under varying viewing conditions. Ideal local features also return scale and orientation information of the region of interest or the interest point and its neighbourhood. The most popular algorithms for feature detection are classified into three groups (Roth & Winter, 2008):

- Corner based detectors,
- Region based detectors, and
- Other approaches.

Corner based detectors locate points of interest or regions of interest that contain corner-like structures. Corner based detectors are not well suited for the detection of uniform regions or regions with smooth transitions. Region based detectors are focused on detecting local blobs or regions of uniform brightness. Region based detectors are better suited than corner based detectors for uniform regions and regions with smooth transitions (Roth & Winter, 2008; Sebe & Lew, 2003b).

3.5.1 Corner based detectors

Corner based detectors detect points at which the neighbourhood around the point exhibits strong gradient changes in two orthogonal directions (Grauman & Leibe, 2010). Two early popular algorithms for corner detection are the Harris detector (Harris & Stephens, 1988) and the Harris-Laplace detector (Mikolajczyk & Schmid, 2004).

The *Harris detector* (Harris & Stephens, 1988) was explicitly designed for geometric stability (Grauman & Leibe, 2010). It proceeds by searching for points x where the second moment matrix[1] C around x has two large eigenvalues. The matrix C is computed from the first derivatives in a window around x weighted by a Gaussian (Chen *et al.*, 2009; Grauman & Leibe, 2010; Harris & Stephens, 1988):

$$C(x, \sigma, \tilde{\sigma}) = G(x, \tilde{\sigma}) * \begin{bmatrix} I_x^2(x, \sigma) & I_x I_y(x, \sigma) \\ I_x I_y(x, \sigma) & I_y^2(x, \sigma) \end{bmatrix} \tag{3.1}$$

where convolution with $G(x, \tilde{\sigma})$ performs a weighted summation of the pixels in the neighbourhood to be considered. The parameter $\tilde{\sigma}$ is typically set to 2σ so that the size of the considered neighbourhood around a point is larger than the size of the kernel used for computing the derivative images. The convolution with a Gaussian kernel is done in the same way as Gaussian smoothing (Section 2.3.1). The eigenvalues of the matrix C are not explicitly computed, instead, the following equivalences are used:

$$\det(C) = \lambda_1 \lambda_2 \tag{3.2}$$

$$\text{trace}(C) = \lambda_1 + \lambda_2 \tag{3.3}$$

The ratio is checked to determine whether the ratio $r = \frac{\lambda_1}{\lambda_2}$ of the eigenvalues is below or above a threshold, the equivalences can be used as follows,

$$\frac{\text{trace}^2(C)}{\det(C)} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} = \frac{(r\lambda_2 + \lambda_2)^2}{r\lambda_2^2} = \frac{(r+1)^2}{r} \tag{3.4}$$

and the condition for determining whether the point should be accepted or rejected can be stated as,

$$R(C) = \det(C) - \alpha \text{trace}^2(C) > t \tag{3.5}$$

[1]The second moment matrix summarizes the predominant directions of the gradient in a specified point neighbourhood. In images it is always a 2 x 2 matrix.

where

$$\alpha = \frac{r}{(r+1)^2} \tag{3.6}$$

α is typically in the range (0.04 – 0.06). The equivalences in Equation 3.2 and Equation 3.3 allow exact explicit computation of the eigenvalues of C to be avoided.

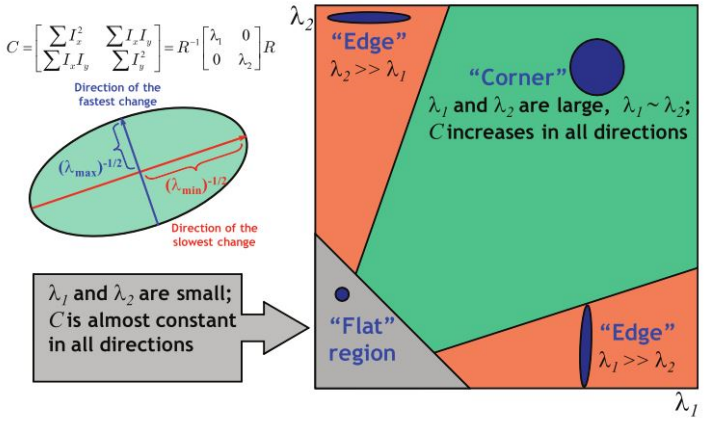


Figure 3.2: Eigenvalue decision boundaries. (Pellegrini, 2007).

Figure 3.2 shows the decision boundaries based on the eigenvalues. If λ_1 is large and λ_2 is small or vice-versa then the neighbourhood being evaluated most likely lies on an edge. If λ_1 and λ_2 are small then the region is considered to be flat. If both λ_1 and λ_2 are large the region being evaluated is considered to be a corner. The Harris corner detector is useful when searching for corners and when precise localization is required although it can only cater for small scale changes due to the fixed kernel size and σ value used during convolution (Grauman & Leibe, 2010).

The *Harris-Laplace* detector localizes points in each level of a Laplacian-of-Gaussian (Section 3.5.2) scale-space representation of the image using the same measure as the Harris corner detector and then selects the points that are maxima in the scale-space.

3.5.2 Region based detectors

Region based detectors aim to detect local blobs of uniform brightness within images (Roth & Winter, 2008). In the contex of region based detectors, a region is regarded as a set of connected pixels with the set of pixels being a subset of the image. The definition

of a region for region based detection differs from the definition of regions in segmentation in that region boundaries are not specified by changes in texture or colour but rather refer only to a set of connected pixels (Mikolajczyk *et al.*, 2005). The Hessian detector uses the Hessian matrix to detect blob-like structures. The LoG detector searches over a LoG scale-space to find maximal points corresponding to blob-like regions. The DoG detector uses successive resized, Gaussian smoothed images to approximate the LoG scale-space that will be searched to find blob-like regions.

The *Hessian Detector* makes use of the Hessian matrix[2]. Since the derivatives of an image are highly susceptible to noise, the derivative operator is combined with a Gaussian smoothing step with the smoothing parameter σ (Grauman & Leibe, 2010; Roth & Winter, 2008).

$$H(x,\sigma) = \begin{bmatrix} I_{xx}(x,\sigma) & I_{xy}(x,\sigma) \\ I_{xy}(x,\sigma) & I_{yy}(x,\sigma) \end{bmatrix} \tag{3.7}$$

The Hessian detector proceeds by determining the second order derivitives I_{xx}, I_{yy}, and I_{xy} and then computing the determinant of the Hessian matrix at each point:

$$\det(H) = I_{xx}I_{yy} - I_{xy}^2 \tag{3.8}$$

The Hessian detector has strong responses on blob-like structures due to the use of the second order derivatives (Roth & Winter, 2008). Typically, a Hessian determinant result image is created and certain values are suppressed using non-maximal suppression using a 3 x 3 window where only points whose pixel values are greater than the pixel values of its 8 immediate neighbours and are above a certain threshold are kept (Grauman & Leibe, 2010; Roth & Winter, 2008). The Hessian detector, like the Harris detector, only has rotational invariance properties (Roth & Winter, 2008).

The *Laplacian-of-Gaussian detector* (LoG) (Lindeberg, 1998) detects roughly circular blob like regions (Grauman & Leibe, 2010). The LoG detector proceeds by evaluating each pixel of each level in a LoG scale space [3]. Each level in the image must be normalized by a factor of σ^2. Each point in the 3D scale space is evaluated and must satisfy two conditions. The first condition is that the point must be a local maxima in the scale space. The second condition is that it must be above a set threshold. To determine

[2]Hessian matrix consists of second order partial derivatives.

[3]The LoG scale space is a 3D space where each point is referenced by (x, y, i). (x, y) denote the pixel location at scale space level i. For level i the σ parameter of the LoG kernel used for filtering is σ^i.

local maxima a point is compared to its 8 neighbours in its corresponding scale level as well as its 9 neighbours in scale levels $i+1$ and $i-1$. If the value at (x, y, i) is a local maxima then it is considered a keypoint. The scale of the region around the keypoint is determined by its corresponding level in the scale space.

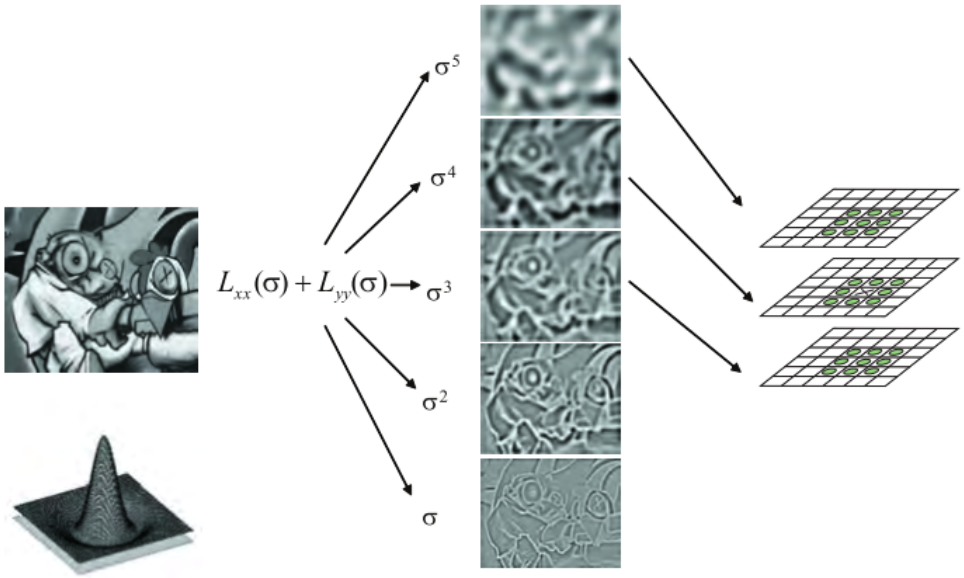


Figure 3.3: LoG scale space construction and local maxima search (Grauman & Leibe, 2010).

Figure 3.3 shows the LoG filter mask at the bottom left. The LoG filter mask has a circular structure with positive weights inside the circlular structure and negative weights outside the circular structure. The circular structure of the filter mask results in a maximum response being obtained from neighbourhoods with roughly circular structure (Grauman & Leibe, 2010; Mikolajczyk & Schmid, 2004). The LoG detector locates regions and selects a characteristic scale of the detected region. The selection of a characteristic scale gives the LoG detector the property of scale invariance (Grauman & Leibe, 2010). The *Difference-of-Gaussians detector* (DoG) (Lowe, 2004) is based on the LoG detector in that it approximates the LoG responses from the DoG. The difference of Gaussians is given by:

$$D(x, \sigma) = (G(x, k\sigma) - G(x, \sigma)) * I(x) \tag{3.9}$$

Lowe (2004) shows that when the scaling factor k is constant then the required scale

normalization is already included unlike the LoG scale space creation procedure which requires an additional normalization step for each level. The DoG scale space is made up of octaves with each octave image resolution scaled by a factor of two in relation to the previous octave. Each octave contains a number of scale space images that have been convolved with a Gaussian kernel. Each scale octave is divided into an equal number of K intervals such that $k = 2^{\frac{1}{K}}$ and $\sigma_n = k^n \sigma_0$, or more directly, $\sigma_n = 2^{\frac{n}{K}} \sigma_0$, with the image being scaled by a factor of two after each octave scale. Once the DoG scale space has been constructed, the candidate regions of interest are found in the same way as they are found for the LoG detector. The candidate regions of interest are found by searching for local maxima in the DoG scale space (Lowe, 2004).

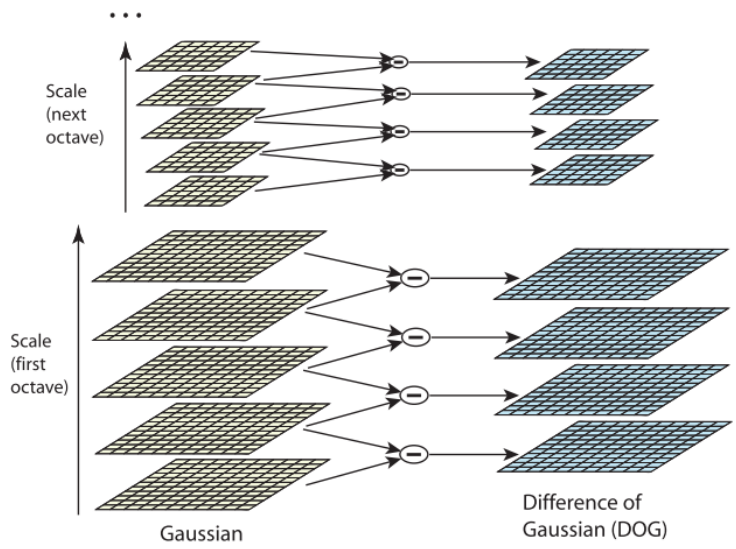


Figure 3.4: DoG scale space construction (Lowe, 2004).

Figure 3.4 shows the process of building a scale space using the LoG operator. Once the levels of the scale space have been created then the neighbouring pixels at appropriate levels can be compared to the current pixel being evaluated. The motivating factor behind using the DoG detector rather than the LoG detector is its computational efficiency improvement over the LoG detector while still yielding similar results. The DoG detector also has scale invariance properties as a result of the search through image plane as well as the scale space.

The *maximally stable extremal regions* (MSER) detector, proposed by Matas *et al.* (2002), is a watershed-like algorithm based on intensity values. The MSER detector detects

arbitrary shapes that are defined by border pixels enclosing a region where all values inside the region are consistently higher or lower than the intensities of the regions bordering pixel intensities Grauman & Leibe (2010). The MSER algorithm first proceeds by finding maximal regions. Finding maximal regions is often described intuitively by imagining all possible thresholdings of a gray-level image. When a pixel value is below the threshold it is considered black and when it is above the given threshold, the pixel value is considered white. A movie of thresholded images I_t with frame t corresponding to threshold t will initially be white. As the frames progress and the threshold increases, black spots corresponding to local minima will begin to emerge and increase in size. At certain thresholds, some local intensity minima will begin to merge until the final image appears all black. The set of all connected components from all frames are maximal regions and are MSER candidates. Candidates that are stable over a wide range of thresholds are considered maximally stable. Since many extremal regions are nested, an appropriate formalization to enumerate regions is required. A connected components tree is created and each component is scored according to the following equation:

$$q_i = |Q_{i+\triangle} \backslash Q_{i-\triangle}|/|Q_i| \tag{3.10}$$

Local minima scores in the tree are then taken to be MSER regions. An MSER is formally defined as follows: Let $Q_1, ..., Q_{i-1}, Q_i, ...$ be a sequence of nested extremal regions, i.e. $Q_i \subset Q_{i+1}$. Extremal region Q_{i*} is maximally stable iff $q_i = |Q_{i+\triangle} \backslash Q_{i-\triangle}|/|Q_i|$ has a local minimum at $i*$(|.| denotes cardinality). $\triangle \in S$ is a parameter of the method. The MSER detector detects regions that are of interest since they are invariant to affine and scale changes, as well as being efficient to compute (Grauman & Leibe, 2010; Matas *et al.*, 2002).

3.5.3 Other Approaches

Local feature detectors that are categorized as other detectors do not attempt to find blob-like regions or corners specifically. *The Entropy Based Salient Region (EBSR)* detector is an example of a detector that does not fall into the corner or region -based categories. The EBSR detector was developed by Kadir & Brady (2001). The EBSR detector is based on the saliency definition and entropy measures for image description and matching proposed by Gilles (1998) but introduces a scale invariance through a scale space search and a normalization process for each scale. Gilles (1998) defined saliency in terms of local signal complexity or unpredictability. The complexity of a signal can therefore be used as a measure of saliency.

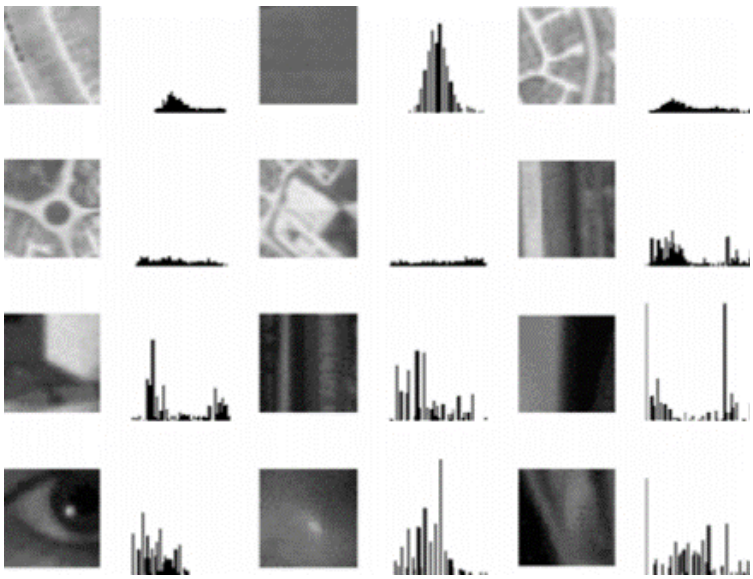


Figure 3.5: Gray level histograms for various image structures (Gilles, 1998)

Figure 3.5 shows how the gray level histogram corresponding to a uniform image patch has a significant peak while image patches containing more complex structures have flatter gray level intensity histograms. The EBSR has scale invariance properties (Roth & Winter, 2008).

3.6 Local Feature Descriptors

Once local feature points or regions have been detected, a description of each local feature can be computed. A local feature description is a description over a limited area within an image. Local feature descriptors can then be used for image matching and object recognition (Csurka *et al.*, 2004) as well as count estimation (Ryan *et al.*, 2009). Local feature descriptors should provide a certain level of invariance to illumination, 3D projective transformations, and common object variations but still be distinctive enough to identify particular objects amongst other objects (Lowe, 1999). A feature vector describing the local feature is computed which can then be used to create a visual model or compare images by computing a similarity between feature points detected in images (Azad *et al.*, 2009; Mikolajczyk & Schmid, 2004)

3.6.1 Scale-Invariant Feature Transform (SIFT)

One of the most popular feature descriptors is the Scale-Invariant Feature Transform (SIFT) developed by Lowe (2004). SIFT is the combination of a DoG feature detector and SIFT-key descriptor. The resulting features are highly distinctive which allows features to be matched with high probability. The four major stages of computation for the SIFT keypoints are (Lowe, 2004):

1. Scale-space extrema detection,
2. Keypoint localization,
3. Orientation Assignment, and
4. Keypoint descriptor.

The Scale-space extrema detection and keypoint localization are discussed in Section 3.5.2. Once keypoint locations and scale have been found using the DoG detector, the SIFT-keys that describe the region can undergo orientation assignment and keypoint description.

Orientation Assignment

The Gaussian smoothed image, L, that most closely represents the keypoint scale is used to perform further computations to maintain scale-invariance. For each pixel, $L(x, y)$, at the selected scale, the gradient magnitude and orientation are calculated using pixel differences as follows:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{3.11}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))) \tag{3.12}$$

where m is a matrix that contains the resulting gradient magnitude values for each pixel (x, y) in the image. θ is a matrix that contains the resulting gradient direction for each pixel (x, y). An orientation histogram is created from the region's gradient magnitude and orientation. The orientation histogram has 36 bins covering a 360 degree range of rotation. Each gradient orientation that is added to the histogram is weighted by its gradient magnitude and by a Gaussian weighted circular window with a σ that is 1.5 times than that of the scale of the keypoint. Histogram peaks are considered to be

the dominant orientation. For histograms where there are bins that are at least 80% of the greatest peak, a separate keypoint is created and assigned the orientation of its corresponding peak in the orientation histogram.

Keypoint Descriptor

To create the keypoint descriptor, the region around the keypoint is divided into 4x4 non-overlapping cells. An orientation histogram with 8 bins is created for each cell within the 4x4 grid. Each of the histograms are then normalized to unit length and concatenated, yielding a feature descriptor with dimensionality 4x4x8=128. A pair of SIFT descriptors can be compared by determining the Euclidean distance between them.

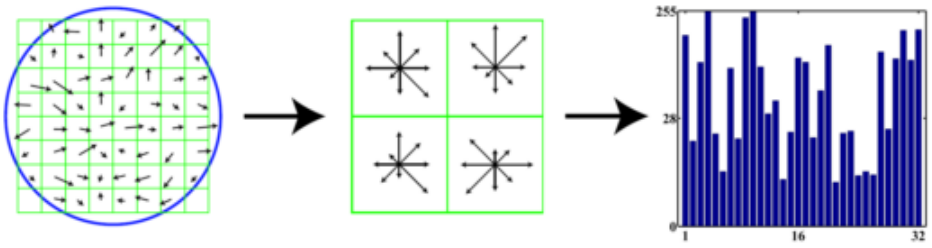


Figure 3.6: SIFT keypoint descriptor creation. (Roth & Winter, 2008)

Figure 3.6 shows a visualization of the creation of a SIFT descriptor. Although a 4x4 division is usually used, Figure 3.6 shows a 2x2 division. Each cell contains 16 gradient orientations. An orientation histogram with 8 bins is created for each of the cells. Each of the the resulting histograms is then normalized and concatenated. The example in Figure 3.6 would result in a descriptor with 2x2x8=32 dimensions.

3.6.2 Speeded Up Robust Features (SURF)

Speeded Up Robust Features (SURF) (Bay *et al.*, 2006) are constructed using the same steps as in SIFT although their construction differs in the details within the steps. SURF is a detector-descriptor combination that uses an approximation of the determinant of the Hessian matrix for keypoint detection and the distribution of Haar-wavelets in the keypoint neighbourhood for the feature description. Detection and description can both be computed efficiently by using the integral image. Bay *et al.* (2006) indicate that their detector-descriptor scheme is as robust and accurate as SIFT features but can be

computed more efficiently as a result of using the integral image in the computations.

To detect SURF features, an approximation of the second-order Gaussian derivatives in x, y, and xy directions are used. The Gaussian derivative approximations are formed by box-filters as shown in Figure 3.7. The Hessian matrix consisting of second-order partial derivatives with a Gaussian smoothing with σ is given by:

$$H(x, \sigma) = \begin{bmatrix} I_{xx}(x, \sigma) & I_{xy}(x, \sigma) \\ I_{xy}(x, \sigma) & I_{yy}(x, \sigma) \end{bmatrix} \tag{3.13}$$

and the determinant of the Hessian matrix is given by:

$$\det(H) = I_{xx} I_{yy} - I_{xy}^2 \tag{3.14}$$

As with the original Hessian detector (Section 3.5.2), the determinant of the Hessian matrix responds at points where vertical, horizontal, or diagonal lines intersect. By using the box filter approximations, the convolution of the image with the second-order Gaussian derivatives can be efficiently approximated through convolution with the integral image (Bay *et al.*, 2006). The integral image H at a pixel (x, y) is the cumulative sum of all pixels up to that pixel and is defined by:

$$H(x, y) = \sum_{i \leq x, j \leq y} I(i, j) \tag{3.15}$$

Given the definition of the integral image, the sum of pixels within a rectangular region in the image can efficiently be computed using:

$$\text{Sum of shaded region} = H(i, j) - H(i - w, j) - H(i, j - h) + H(i - w, j - h) \tag{3.16}$$

where i and j refer to the right lower corner location of the rectangular region and w and h refer to the rectangular regions width and height respectively. The sum of shaded values is computed effectively using the second order Guassian derivative box approximations shown in Figure 3.7. Since the kernel element in the box filter are constants, the convolution can be performed by summing the pixels under the kernel and multiplying them by the constant kernel value.

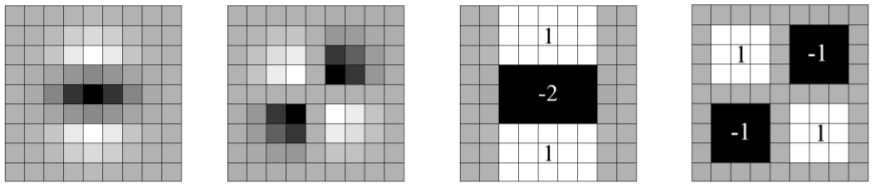


Figure 3.7: Second order Gaussian derivatives in y and xy directions and their respective box-filter approximations (Bay *et al.*, 2006).

To determine the scale of the keypoint, the determinant of the Hessian matrix is evaluated for varying values of θ in the same way as in scale selection for the DoG detector by varying θ by a constant value, comparing the point's value with that of its 26 neighbours in the scale space and selecting extrema points above a threshold.

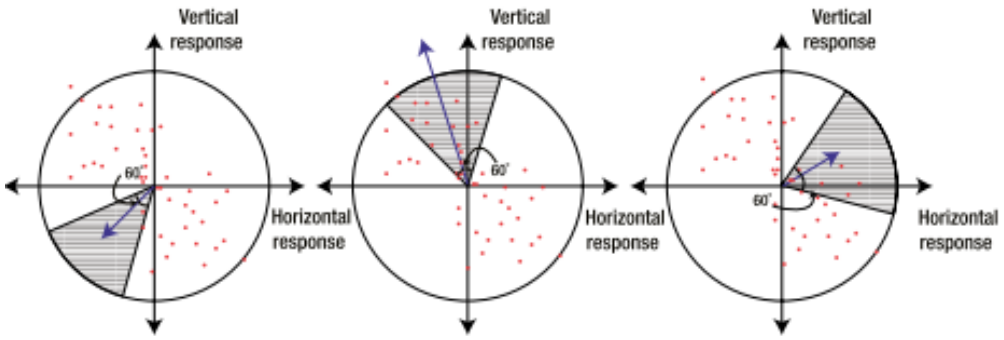


Figure 3.8: SURF orientation assignment. (Roth & Winter, 2008)

Figure 3.8 shows the process for orientation assignment in the SURF description process. Keypoint orientation is found by evaluating pixels in a neighbourhood that is 6 times larger than the selected scale for the keypoint. At every point within the radius, responses to vertical and horizontal box filters called Haar-wavelets are recorded. The responses are then smoothed with a Gaussian filterca with $\theta = 2.5$ times the keypoint scale. The horizontal and vertical response strengths are then represented in a 2D space with the horizontal response strength on the x-axis and the vertical response strength on the y-axis. A sliding arc with an angle of 60 degrees then sweeps through the space and for every location of the sliding arc, the summation of responses is recorded. The largest of the responses is selected as the orientation of the keypoint.

The SURF descriptor is computed for each keypoint by defining a square region around a keypoint that is 20 times the selected scale and oriented according to the selected orientation. The region is split into 4x4 subregions. Within each subregion, vertical and horizontal wavelet responses are computed at 5x5 regularly spaced intervals. The responses for the wavelets as well as their absolute values are summed. This results in a 4-element vector,

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \tag{3.17}$$

A 4-element vector is created for each subregion, resulting in a 64-dimensional feature vector for each keypoint.

3.7 Global Features

Global features refer to characteristics of a region within an image, for example, region size, perimeter, colour, texture, or shape as well as more sophisticated methods that use dimensionality reduction. Global features are usually computed over all pixels within a region or are computed using the boundary pixels of a region (Jain *et al.*, 1995; Shashikanth & Kulkarni, 2013). Global features are not robust to occlusions and do not have invariance properties that can be provided by local features (Shashikanth & Kulkarni, 2013). The HOG feature descriptor provides a comprehensive description of images by computing histograms of gradient orientations for overlapping blocks in images (Section 3.7.1). Haar-like features are used to create image descriptions by computing responses of convolution of the Haar-like features with the image (Section 3.7.2).

3.7.1 Histograms of Oriented Gradients (HOG)

The HOG descriptor, created by Dalal & Triggs (2005), is a successor of many other orientation histogram feature descriptors. It is considered a global feature descriptor as it describes the image as a whole in terms of its gradient orientations and magnitudes. The HOG computation requires several steps to compute a feature vector that describes the image. The steps include:

1. Gradient Computation,
2. Spatial/Orientation Binning,
3. Normalization and descriptor blocks creation, and

4. Histogram of gradient orientation collection over detection window.

Computing the image gradients is performed using a simple 1D mask [-1,0,1] as it was found to provide good results (Dalal & Triggs, 2005). For RGB images, the gradients are calculated separately for each channel and the channel with the largest for a pixel is taken as the gradient vector for that pixel. For the spatial/orientation binning step, each pixel contributes a weighted vote for an edge oriented histogram channel. The image is split up into cells over which the orientation bins are accumulated. The orientation bins are evenly spaced over 0° - 180° or 0° - 360°. To reduce aliasing, the votes are bilinearly interpolated between the two neighbouring bins that the vote contributes towards. To avoid issues with changing illumination and foreground/background contrast, local contrast normalization is performed. Rectangular regions covering a number of cells are used so each cell contributes several components to the final feature vector with the components being normalized with respect to different blocks. Pixels are downweighted near block edges by applying a Gaussian spatial window before accumulating orientation votes into cells.

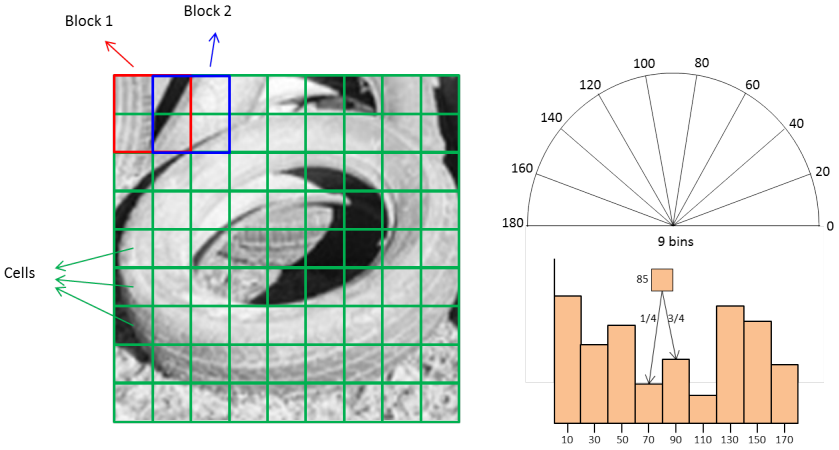


Figure 3.9: Visualization of HOG descriptor extraction process.

Figure 3.9 shows the extraction process for the HOG feature descriptor where there are 8x8 blocks, each containing 2x2 cells and each cell results in a histogram with 9 bins. The normalization takes place in each block to increase the performance of the descriptor by minimizing the effects of illumination and background/foreground contrast. Once the histograms have been created for each cell within each block, the resulting histograms are concatenated. In the example in Figure 3.9, the resulting feature vector will be a (8x8)x(2x2)x9=2304 dimensional feature vector.

3.7.2 Haar-like Features

The Haar-like features were introduced by Viola & Jones (2001) as an alternative to the Haar basis functions used by Papageorgiou *et al.* (1998). The Haar-like features are fast to compute as they use the difference between pixel summations within specified rectangles and can be computed quickly using the integral image. There are three kinds of features in the Haar-like features scheme. The three kinds are the *two-rectangle feature*, the *three-rectangle feature*, and the *four-rectangle feature*.

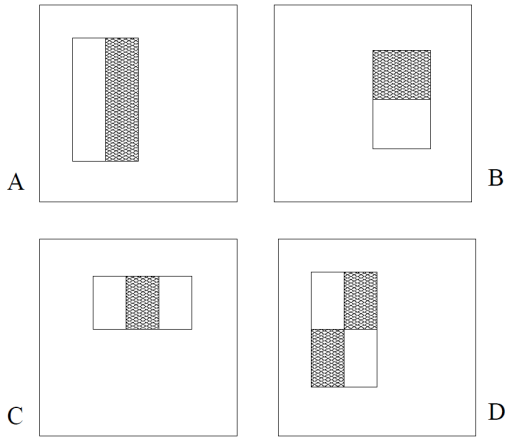


Figure 3.10: Example rectangle features shown relative to the enclosing detection window (Viola & Jones, 2001).

Figure 3.10 shows the three kinds of Haar-like features. A two-rectangle feature is calculated by subtracting the summation of the pixel values in the dark rectangle region from the summation of the pixel values in the light rectangle. The rectangles in a two-rectangle feature are always horizontally or vertically adjacent. A three-rectangle feature is used to calculate the value of the two outside rectangles subtracted from the centre rectangle. The four-rectangle feature is used to calculate the difference between diagonal pairs of rectangles. Depending on the resolution of the detector window, the complete set of Haar-like features can become quite large. Viola & Jones (2001) use a weak learning algorithm to select which feature best separates positive and negative examples over a series of stages.

3.8 Related Recognition System: The Bag of Visual Words

The bag of visual words strategy for object category recognition is the standard paradigm for image classification (Grauman & Leibe, 2010). The BoVW model describes a systematic approach for the categorization of objects in images. It was briefly described in Section 3.3 to provide context to the discussion of image features. This section discusses the original BoVW experiments and implementation by Csurka *et al.* (2004) in more detail to provide context of the model used for the categorization experiments in this study.

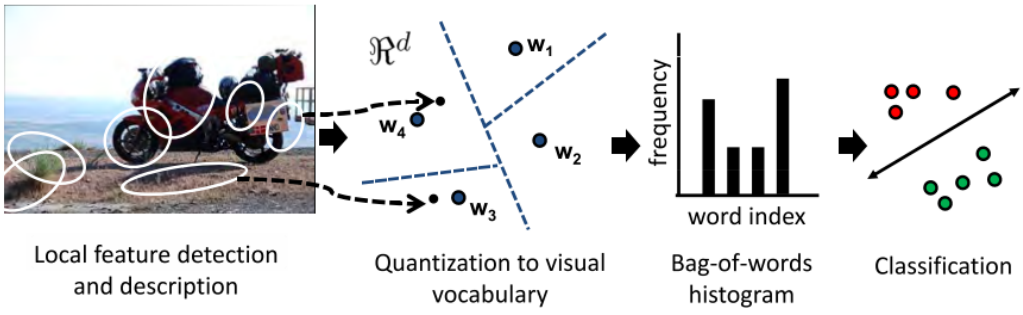


Figure 3.11: Bag-of-words image classification pipeline (Grauman & Leibe, 2010).

Figure 3.11 shows the pipeline for the bag of visual words approach. Local feature detection and description aims to select and describe salient regions in the image that can be clustered to form the visual vocabulary. A histogram is then created for the image by associating each of the local feature descriptions from the image with a visual word in the vocabulary. The histograms are then used to create a classifier that can be used to classify novel images based on their histograms formed using the visual vocabulary. The bag of visual words approach is split into a training phase and a recognition phase (Csurka *et al.*, 2004; Grauman & Leibe, 2010).

The training phase is conducted as follows:

1. Collect training image examples from each category of interest.
2. Sample local features from all training images using feature detection at selected regions.
3. Extract a descriptor for each sampled point.

4. Form a visual word vocabulary using a clustering algorithm such as k-means. The clusters form the visual words in the vocabulary.

5. Relate each training image's feature descriptors to their respective visual words to create a k-dimensional histogram for each training image.

6. Use the histograms and corresponding image class labels to train a classifier.

The categorization phase is conducted as follows:

1. Given a novel test image, repeat the feature detection and description steps in the same way as used during training.

2. Use the vocabulary created during training to associate feature descriptors to their visual word clusters.

3. Create histogram of visual word occurrences for the novel image.

4. Assign a class label to the image by classifying its visual word occurrence histogram using the classifier created during the training process.

Csurka *et al.* (2004) evaluate the classification error rate for varying numbers of clusters k using a Naïve Bayes classifier. They show that the number of clusters that provides the optimal trade-off between accuracy and speed is $k = 1000$. Using a cluster size of $k = 1000$, they also compare the classification performance of a Naïve Bayes classifier against a multiclass SVM. They found that the classification accuracy of the SVM is superior to the Naïve Bayes classifier for the purpose of categorization in the BoVW model. Three major choices that must be made by the practitioner when using the bag of visual words strategy are the feature extraction method, the clustering method, and the classification method (Grauman & Leibe, 2010).

3.9 Related Detection Systems

This section discusses two discriminative detection systems that have been shown to achieve good results. The Viola-Jones face detector and the HOG person detector have both received significant attention in the computer vision field and within their respective domains of face detection and person detection. The two detectors are discussed to provide an explanation of the disadvantages associated with using a counting by detection strategy in the context of this study. Preliminary experiments using the two detectors highlighted the issues associated with using object detectors for counting objects in cluttered scenes where objects are highly occluded.

3.9.1 Viola-Jones Face Detector

Viola & Jones (2001) introduce a discriminative learning procedure to discover regularities in 2-dimensional texture patterns of different face instances. There are three key contributions made by Viola & Jones (2001), namely, the integral image which allows Haar-like feature responses to be computed efficiently, a learning algorithm based on Adaboost that selects a set of relevant features from a large set of possible features, and a method for combining increasingly complex classifiers in a cascade of classifiers.

A discussion of calculating responses using the integral image and Haar-like features is given in Section 3.7.2. Viola & Jones (2001) use a variant of Adaboost that selects relevant Haar-like features for a series of weak classifiers which ultimately results in a single strong classifier.

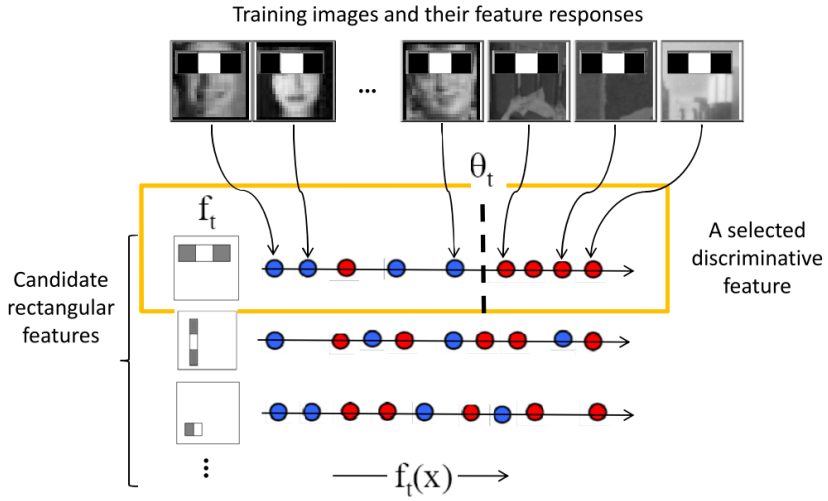


Figure 3.12: The detector training procedure uses AdaBoost to both identify discriminative rectangular features and determine the weak classifier (Grauman & Leibe, 2010).

Figure 3.12 shows the part of the selection process for creating weak learners. f_t denotes the Haar-like feature. $f_t(x)$ is the response of the Haar-like feature f_t for training image x. The threshold is given as θ_t. The t subscript references the current Haar-like feature being evaluated. The weak classifiers are learned by determining the optimal threshold separating the positive and negative training examples for each possible Haar-like feature.

Each weak classifier is given as (Grauman & Leibe, 2010):

$$h_t(x) = \begin{cases} 1, & f_t(x) > \theta_t \\ -1, & \text{otherwise} \end{cases} \tag{3.18}$$

The variant of the Adaboost algorithm proceeds to determine the weak classifiers that make up the cascade of classifiers by performing the following training process (Viola & Jones, 2001):

- Given example images $(x_1, y_1), ..., (x_n, y_n)$ where $y = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = 1/(2m), 1/(2l)$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, ..., T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}; \tag{3.19}$$
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$
  3. Update the weights:
  $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i} \tag{3.20}$$
  where $e_i = 0$ if x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \dfrac{e_t}{1 - e_t}$
- The final strong classifier is:
$$h_t(x) = \begin{cases} 1, & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{t} \alpha_t \\ 0, & \end{cases} \tag{3.21}$$
where $\alpha_t = \log \dfrac{1}{\beta_t}$.

The resulting classifier is an ensemble of weak classifiers each with a corresponding Haar-like feature and threshold value. The value T indicates the number of weak classifiers that will form part of the final strong classifier. Each round of boosting selects one feature out of 180 000 potential features. The Adaboost variant both selects features and trains a classifier. By increasing the number of weak classifiers in the ensemble, the classification

accuracy is improved however it is at the cost of efficiency since more feature responses are computed. In order to provide a detector that has high discriminative power and is still efficient in searching an image for an object, Viola & Jones (2001) use a cascade classification approach in which the image window being evaluated must pass multiple stages of classification. Each stage is more complex than the previous and requires more computations to be performed. By separating the classifier into stages, image regions that do not contain faces can be classified as negative and disregarded before more time consuming computations, by subsequent classifiers, are performed.

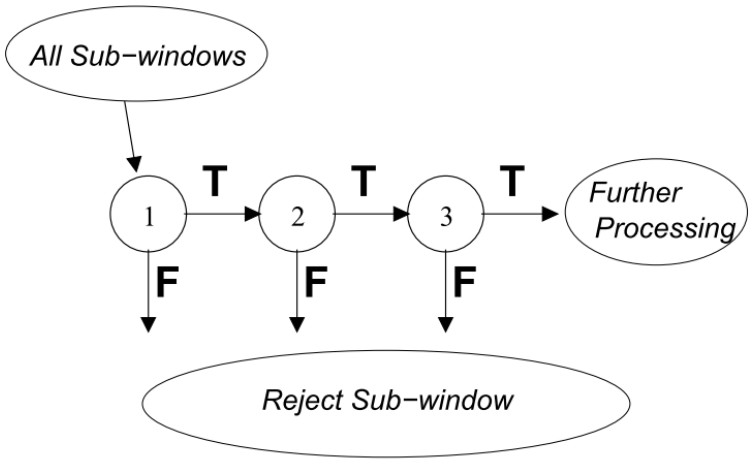


Figure 3.13: Depiction of the detection cascade for Viola-Jones face detection algorithm (Viola & Jones, 2001).

Figure 3.13 shows the cascade classifier that is produced from the weak classifiers that were produced using the variant of the Adaboost algorithm. All sub-windows of a query image are presented to the classifier individually. If an image is classified as positive by classifier 1 then it is presented to classifier 2 and so on until the final, most discriminative classifier is reached. If any classifier in the cascade classifies the image sub-window as negative then the image sub-window is rejected. If the final classifier assigns a positive classification then a detection is made. Each stage of the classifier reduces the false positive rate and decreases the detection rate. To produce the cascade, a target is selected for the minimum reduction in false positives and the maximum decrease in detection rate for each stage. Each stage is trained by adding features until the target detection and false positive rates are met which are determined by classification of images in a validation set. Stages are added to the cascade until the overall target for the false positive and detection rate is met. Grauman & Leibe (2010) note that this object detector only caters

for the viewpoint of the object that is used in the positive training images.

3.9.2 HOG Person Detector

Like the Viola-Jones facial detection algorithm, HOG person detector proposed by Dalal & Triggs (2005) makes use of a feature extraction and description technique based on gradient orientations in sample images. The training set consists of positive images containing people and negative images that do not contain people. Once the HOG features are extracted, a linear SVM is trained to separate positive and negative instances. A sliding window over all possible locations and selected scales is applied to a query image. For each sub-window of the query image, its HOG feature representation is extracted and classified using the SVM from the training phase. HOG features and SVMs are discussed in Sections 3.7.1 and 5.1.2 respectively.

3.10 Related Counting Systems

The problem of automatic object counting in images has been approached from the perspectives of counting by detection, segmentation, regression, and a combination as discussed in Section 3.4. This section reviews three proposed counting methods. Each method that is described falls into the category of either counting by detection, segmentation, or regression. The purpose of the discussion about these counting systems is to provide an indication of the advantages and disadvantages of each and provide an indication of which counting approach is most appropriate in the context of this study.

3.10.1 Car Counting by Detection

Tongphu *et al.* (2009) propose a method for the rapid detection of multiple object instances. The domain of their study is vehicle counting. The aim was to produce a classifier that could detect individual stationary vehicle instances from aerial images of parking lots.

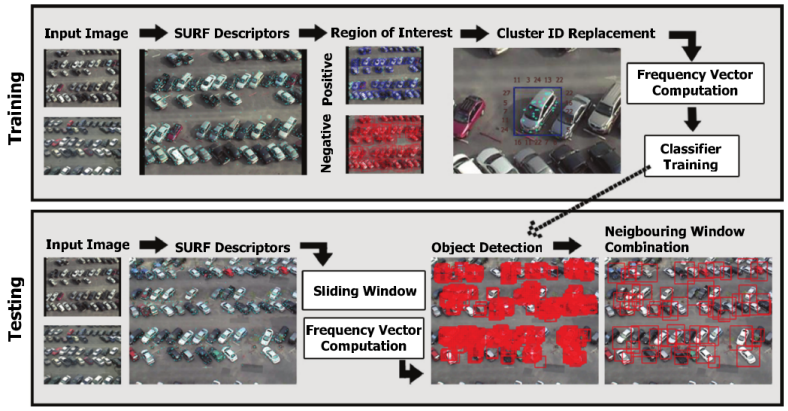


Figure 3.14: System overview of a vehicle counting method proposed by Tongphu *et al.* (2009).

Figure 3.14 shows an overview of the method for vehicle counting proposed by Tongphu *et al.* (2009). The feature extraction step during training consisted of annotating vehicle instances with bounding boxes. SURF or SIFT features were then extracted from each bounding box in each training image and quantized into k visual word clusters as in the BoVW model. A histogram of visual word occurrences was then computed for each bounding box in each image. An Adaboost classifier was then trained using positive and negative histogram feature vectors. The detection process used a sliding window approach. SURF or SIFT features were extracted for each sub-window and quantized into SURF or SIFT feature occurrence histograms using the clusters created during training. Each sub-window was then classified as either containing a vehicle or not.

Figure 3.15: Visualization of detections for a vehicle counting method Tongphu *et al.* (2009).

Figure 3.15 shows the detection output of the system proposed by Tongphu *et al.* (2009). It can be seen that in some instances there are two detections for a single car or there are detections when two cars are both partially inside the detection window. A possible reason for this could be that the histograms created for those sub-windows are similar to the positive examples used during training although this is not stated by the original author. The similarity could occur as a result of features found on the back and front of the car being assigned to the same cluster which would result in a feature from the back of the car contributing to the same histogram bin.

To evaluate the proposed method, Tongphu *et al.* (2009) used the Viola-Jones method for detection that was trained using the sub-windows represented by their bounding box annotations as a baseline for comparison. They compared the number of true positives (hits), false negatives (misses), false positives, and the time taken for feature extraction

and classification. The results showed that the BoVW with SIFT or SURF features with a sliding window detection scheme was superior to the Viola-Jones detector in their particular problem domain of vehicle counting from aerial images.

3.10.2 Cell Counting by Segmentation

Tulsani (2013) makes use of marker-based watershed segmentation to segment and count occurrences of white blood cells (WBCs), platelets, and red blood cells (RBCs) in magnified images of blood smears. The process for creating markers for each of the cell types to be counted consists of:

1. Image preprocessing,
2. WBC and platelet extraction,
3. Computing foreground markers for each cell type, and
4. Applying watershed transform and counting the regions.

The image preprocessing step consists of applying a mean filter to remove noise. After noise has been removed, the image is converted to a Ycbcr format. The Cb component of the Ycbcr colour representation is chosen for further computations WBCs and platelets appear lighter than RBCs and the background. The image is then converted to a binary image that contains only the WBCs and platelets. Morphological image processing is then used on the binary image to remove the small platelets and to reconstruct the WBCs. This results in a mask for the WBCs. The WBC mask is then subtracted from the original binary image containing the WBCs and platelets to obtain a mask containing only the platelets. The original image is converted to grayscale and morpholocial image processing is applied. The grayscale image is converted to a binary image that contains all the cells and platelets. The mask containing the WBCs and platelets is then subtracted from the binary image containing all the cells and platelets. The binary image resulting from the subtraction contains only RBC markers.

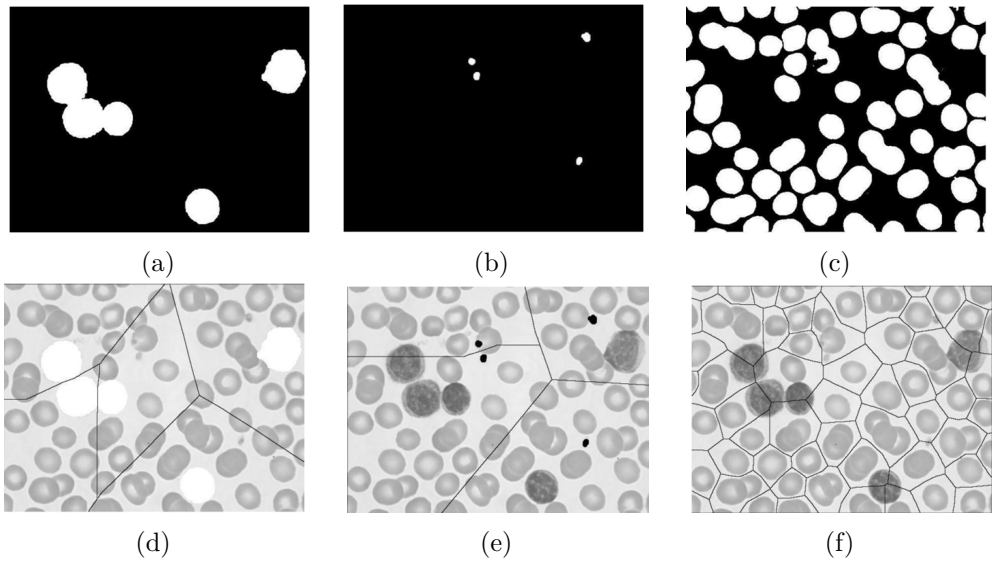


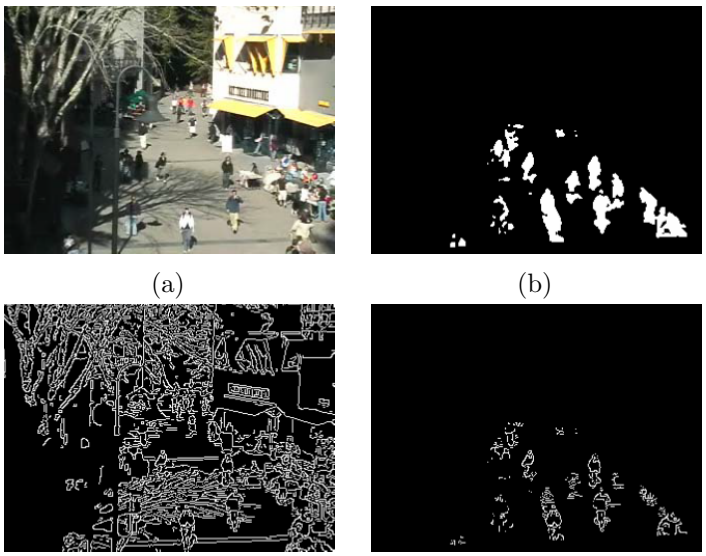
Figure 3.16: Markers for the three different cell types and visualizations of the watershed lines separating cell regions (Tulsani, 2013).

Figures 3.16a, 3.16b, and 3.16c show the masks for the WBCs, platelets, and RBCs respectively. The results of applying the watershed transform to each of the images are shown in Figures 3.16d, 3.16e, and 3.16f. In Figures 3.16d and 3.16e it can be seen that there is a one to one correspondence of WBCs and platelets to regions respectively. In Figure 3.16f it can be seen that there are multiple cells in few regions, particularly where RBCs are overlapping. Methods have been proposed that combine segmentation with classification or regression on individual regions to produce more accurate count estimations (Arteta & Lempitsky, 2012, 2013; Chan *et al.*, 2008; Kong *et al.*, 2006; Ryan *et al.*, 2009).

3.10.3 People Counting by Regression

Kong *et al.* (2005) propose a viewpoint invariant approach for crowd counting. Their approach counts the number of pedestrians in each frame of a video. The approach consists of creating a foreground mask for pedestrians in the image using the mixture-of-Gaussian based adaptive background modeling method (Stauffer & Grimson, 1999). The foreground mask contains blobs that represent potential pedestrians. Canny edge detection is applied to the original image to detect edges. An **AND** operator is applied to the edge image and the foreground mask resulting in an edge image that contains only the edges corresponding to blobs in the foreground mask. Edge and blob pixels in the two

images are normalized with respect to the relative density and scale that are estimated during a calibration step. After normalizing the edge and blob pixels, histograms are created for the blob sizes and gradient orientations. The feature vector for an image is the concatenated blob size and edge orientation histograms.



(a) (b)

(c) (d)

Figure 3.17: Blob and edge feature extraction (Kong *et al.*, 2005).

Figure 3.17a shows a single frame of video containing pedestrians. The foreground mask is shown in Figure 3.17b. The result of applying the **AND** operator to the edge image (Figure 3.17c) and the foreground mask (Figure 3.17b) results in an edge image containing the edges of pedestrians (Figure 3.17d).

The training procedure aims to find a relationship between the features and the number of pedestrians in the image. During training a reference ROI in the first video frame is chosen to determine the variables to be used in the blob and edge pixel normalization step. Edge orientation and blob size histograms are extracted from the normalized features at every twenty frames of the video. Each edge orientation histogram has eight bins quantized from 0 to 180 degrees and the blob size histogram contains six bins with size 500 and uniform spacing. They compare the results of fitting a linear model with the results obtained from training a feed-forward neural network. The neural network contained a single hidden layer and was trained using standard back propagation with batch presentation.

A linear model was fit and a neural network trained at two different sites. The site where pedestrians were sparse and there was little occlusion, the results of the neural network and the linear model were comparable. At the site where pedestrians were occluded by each other, the results of the neural network had better estimation performance than the linear model. Kong *et al.* (2006) state that accurate estimation becomes challenging when there is significant occlusion and show that a non-linear mapping of feature histograms to count estimates, using a neural network, has superior performance compared to fitting a linear model.

3.11 Conclusions

Computer vision is inspired by biological vision. It is viewed as being comprised of top-down and bottom up approaches (Section 3.1). Many applications of computer vision use a combination of top-down and bottom-up approaches to vision. Four categories for recognition and detection in computer vision were identified (Section 3.3). Three strategies for counting objects in images were also identified (Section 3.4). Candidate image representation methods were reviewed in terms of feature detection and description (Sections 3.2, 3.5, 3.6, and 3.7).

Candidate image recognition and detection systems were identified for the purposes of categorizing and counting by detection. For categorization, the BoVW model is the standard paradigm and provides a basis for categorizing images based on local feature patches (Section 3.8). The BoVW model makes use of local feature detectors and descriptors for identifying and describing salient features. In the context of tyre categorization based on tread, corner based detectors (Section 3.5.1) could prove to be more suitable than region based detectors (Section 3.5.2) due to the sharp intensity changes resulting from tread grooves in tyre tread images.

The Viola-Jones face detection method was reviewed for counting by detection but only caters for a single view of the target object (Section 3.9.1). The HOG people detector was also reviewed for the purpose of counting by detection (Section 3.9.2). Models created using HOG features allow variations in the appearance of the target object. The global features used in both detection systems do not perform well when objects are occluded in the images. Since the majority of visible tyres in tyre stockpiles are partially occluded, a detection strategy for isolating regions containing individual tyres will not be suitable.

Three counting strategies were identified in Section 3.10. The three categories for counting objects in images using computer vision are counting by detection (Section 3.10.1), segmentation (Section 3.10.2), and regression (Section 3.10.3). In the viewpoint invariant approach to counting that uses a combination of segmentation and regression, it was found that a non-linear mapping of feature descriptions to count estimations showed good counting estimations ability where there were occlusions.

The research question RQ_2 stated at the beginning of this chapter was answered through a thorough review of Computer Vision literature. Chapter 4 reviews the domain of tyres and tyre stockpiles with the goal of selecting appropriate feature representations for the purpose of categorizing individual tyres and count estimation of visible tyres in tyre stockpiles.

Chapter 4

Tyres and Waste Tyre Stockpiles

Creating visual models for the purposes of categorization and count estimation of objects in images requires the selection of salient visual properties in the form of image features. Once salient visual properties have been detected and described, a model can be created using algorithms for categorization or count estimation using the appropriate machine learning techniques discussed in Chapter 5. This chapter reviews the concepts discussed in Chapter 2 and Chapter 3 in the domain of tyre categorization and count estimation in order to answer the third research question:

RQ_3 What are suitable image representations for categorization and count estimation from images?

To determine which features are suitable for tyre categorization and count estimation, tyre tread images and tyre stockpile image characteristics are discussed in terms of their visual appearances and what advantages and disadvantages are associated with particular features for representing images of tyres and tyre stockpiles. The outcomes for this chapter are:

- Selected features for categorizing individual tyres, and
- Selected features for estimating visible tyre counts in tyre stockpiles.

The selection of features for categorizing and detecting individual tyres requires an analysis of the tyre from a visual perspective. The selection of features for estimating the number of visible tyres in a waste tyre stockpile image requires an analysis of both the visual properties of individual tyres and tyre stockpiles. This section discusses the issues associated with the use of detection and segmentation for counting tyres in stockpile

images and a conclusion is reached for which approaches to apply and evaluate for categorization and count estimation in waste tyre stockpile images (Section 4.1). The chosen categorization approach requires local image features to be discussed in terms of their suitability (Section 4.2). The chosen counting strategy requires global image characteristics in the form of global image feature descriptions. Two global feature descriptions are described and discussed for visible tyre count estimation in tyre stockpile images (Section 4.3).

4.1 Detection and Segmentation in Stockpiles

Detection and segmentation of individual tyres in waste tyre stockpiles are difficult tasks. The challenges with detection are a result of varying object appearances due to multiple object angles and the amount of occlusions in tyre stockpile images. The challenges with segmentation are a result of low contrast boundaries and the prominent intensity changes around tyre tread regions. In this section the challenges with these approaches are described in more detail and the reasons for using a counting by regression strategy are explained.

The counting by detection and counting by segmentation strategies from the three strategies for counting objects in images (Section 3.4) are discussed in terms of tyre stockpiles to motivate the use of a counting by regression strategy. The challenges associated with using a counting by detection strategy for visible tyre count estimation become evident through an analysis of a single randomly arranged tyre stockpile image (Section 4.1.1). A discussion on using traditional image segmentation techniques for isolating tyre stockpiles and separating individual tyre instances is given and disadvantages of using a counting by segmentation approach become evident (Section 4.1.2). Finally, the scope for individual tyre categorization and visible tyre count estimation in tyre stockpile images is given (Section 4.1.3).

4.1.1 Detection in Tyre Stockpiles

A major issue with using a counting by detection strategy is that individual tyres can have a number of appearances due to occlusion and clutter in the stockpile. The occlusions are a result of tyres partially or completely covering one another. The clutter is a result of the random placement of the tyres in the stockpiles.

Figure 4.1: Features from single image matched to subregion of a tyre stockpile image.

Figure 4.1 shows some possible appearances of occluded tyres in a waste tyre stockpile. Each bounding box around the selected occluded tyres in Figure 4.1 indicates that detection methods that use a sliding window approach, such as those discussed in Section 3.9.1, Section 3.9.2, and Section 3.10.1, would require a detector that is able to distinguish individual tyres with a large number of visual appearances. Even with a detector that could cater for the large number of visual appearances, the individual tyres would require bounding boxes with varying aspect ratios to tightly surround individual detections. Detection systems such as the Viola-Jones detector (Section 3.9.1) and HOG people detector (Section 3.9.2) are therefore not suitable for localizing individual tyre instances as they make use of global feature descriptors, which cater for single views with small variations in visual appearance, with a sliding window approach for localization.

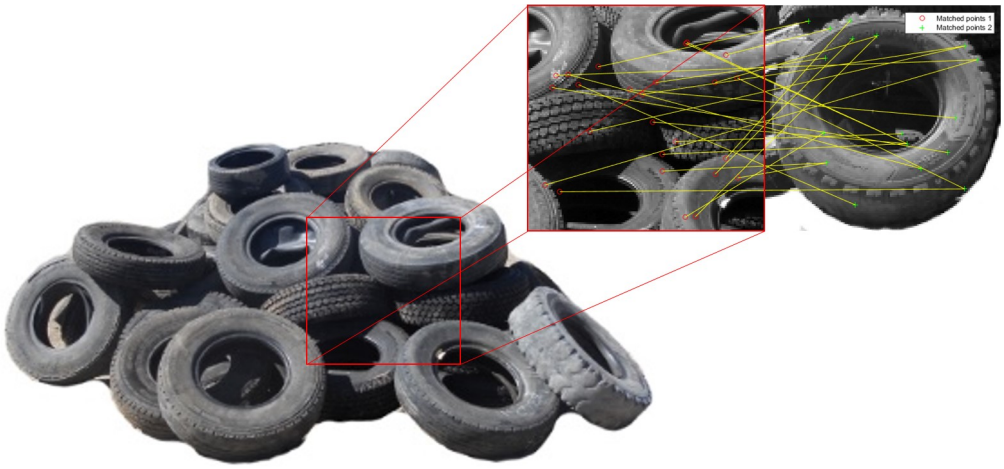
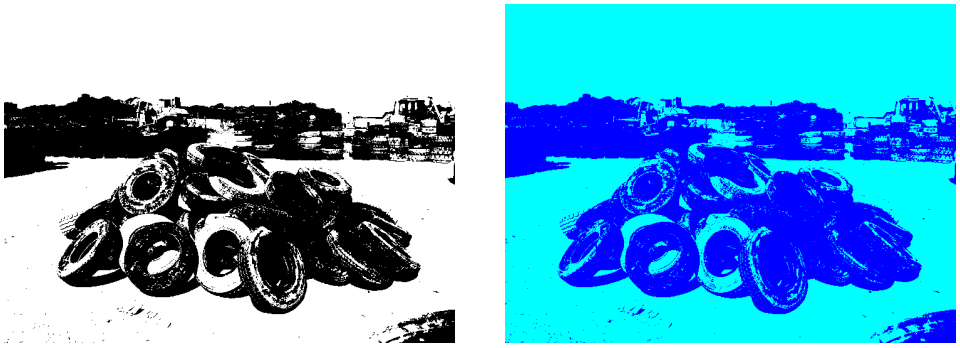


Figure 4.2: Examples of occluded tyres in a tyre stockpile

Using local features with a sliding window approach (Section 3.10.1) could provide an alternative detector that allows for more variation in the appearance of the objects. One disadvantage associated with using local features is that in a pile of tyres, multiple local feature descriptors that are similar are found across the entire stockpile. For example, a bounding box that partially covers two or more tyres may contain very similar local features to a bounding box containing one tyre. Figure 4.2 shows a tyre stockpile image with local features detected in a subregion containing multiple tyres matched to the local features detected in an image containing a single tyre. The similarity of feature descriptions for the two bounding boxes will result in multiple detections that do not provide a one-to-one correspondence with individual tyres. Local features detected over multiple instances of the same object instance cause object detections that do not have a one-to-one correspondence with the objects in the image which is evident in the work of Tongphu *et al.* (2009) in the context of car counting. Since the tyres in tyre stockpiles have less separation between the tyre instances than cars in a parking lot, a detector that uses local features is also considered not suitable for counting tyres by detection.

4.1.2 Segmentation of Tyres and Tyre Stockpiles

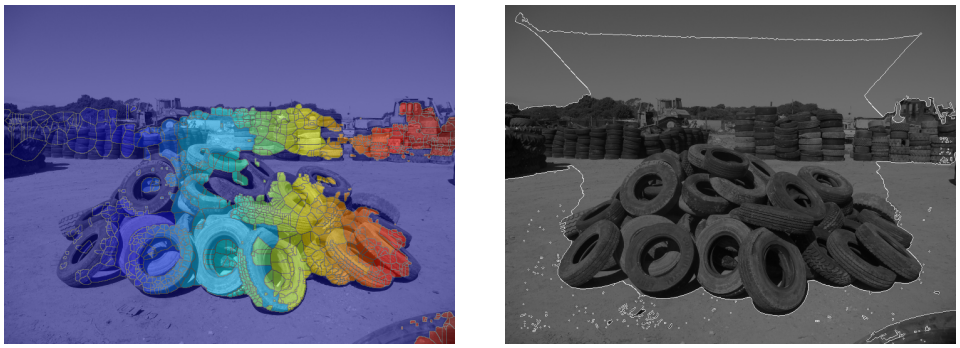
Another possibility could be to segment individual tyres using a segmentation algorithm such as the segmentation algorithms discussed in Section 2.4. The segmentation of image regions into individual tyres could allow the individual regions to be further analyzed for categorization purposes.



(a) (b)

Figure 4.3: Segmentation algorithms applied to tyre stockpiles

Figure 4.3 shows the results of applying the segmentation algorithms discussed in Section 2.4, to a tyre stockpile. Implementations of the Otsu threshold and k-means from Matlab's Image Processing Toolbox and the Statistics and Machine Learning Toolbox (MathWorks, 2016b,d) were used to generate the images in Figure 4.3. It can be seen that there is no clear separation of regions containing tyres. Figure 4.3a and Figure 4.3b shows the results of applying the Otsu threshold value and k-means ($k = 2$) to an image of a tyre stockpile. Both the Otsu segmentation and k-means with $k = 2$ are binary segmentations which were investigated for separating the pixels belonging to the tyre pile from background pixels.



(a) (b)

Figure 4.4: Segmentation algorithms applied to tyre stockpiles

Figure 4.4a and Figure 4.4b show the results of applying watershed segmentation and Chan-Vese segmentation respectively. The images in Figure 4.4 were generated using the respective implementations in Matlab's Image Processing Toolbox (MathWorks, 2016b). The watershed algorithm results in over-segmentation in areas with complex structure such

as tread regions and under-segmentation in areas where there is a low contrast between tyre boundaries. If the image is smoothed prior to applying the watershed algorithm in an attempt to remove the complexities of tread regions, the resulting image contains lower contrast boundaries between tyres and ultimately the tyres are under-segmented. The opposite result occurs in using a sharpening filter to improve tyre boundary contrast. By sharpening the image to improve tyre boundary contrast, the tread structures also become more prominent, resulting in over-segmentation. Although the Chan-Vese segmentation algorithm does not require well defined object boundaries, the contrast between the piles of tyres and the background is far greater than the contrast between individual tyres resulting in the fitting energy for the curve to be minimized when the curve is around the entire pile. It can be seen in Figure 4.4b that the largest image region surrounded by a curve formed by Chan-Vese algorithm surrounds both tyre pixels as well as non-tyre pixels.

Counting by segmentation is also not suitable for estimating the number of tyres in a tyre stockpile. It is evident from the review of cell counting by segmentation (Section 3.10.2) and the above discussion that tyres in tyre stockpiles are located too close to each other and the colour characteristics lead to large over-segmentation and under-segmentation. The over and under -segmentation problem means that it is difficult to obtain a one-to-one correspondence of segmented regions to tyre instances.

4.1.3 Scoping Tyre Categorization and Stockpile Count Estimation

Detection schemes such as those discussed in Section 3.9 require singular views of the target object with only slight variations in their appearances. The discussion on the localization of tyres by detection in Section 4.1.1 indicates that the number of similar visual appearances of individual visible tyres in stockpile images is small in terms of structural appearance while similarities in colour characteristics exist. These observations lead to the conclusion that isolating image regions that correspond to individual tyres by means of a bounding box is a difficult task. The segmentation algorithms discussed in Section 2.4 were investigated as alternative candidates for separating tyres within tyre stockpiles. From the discussion on segmentation in Section 4.1.1 it is evident how the low contrast between individual tyre boundaries and the high contrast in image regions containing tread either cause over-segmentation or under-segmentation.

Due to the difficulty of isolating image regions that contain individual tyres, the categorization of individual tyres is considered in the context of manually segmented images

containing only tyre treads. The manual segmentation of tyre treads allows the investigation of tyre categorization to be considered separately to the individual tyre localization and stockpile segmentation problems.

The problems associated with detecting and segmenting individual tyre regions indicate that out of the three strategies for counting objects in images (Section 3.4), counting by regression is the most appropriate as it avoids the hard problem of localizing individual object instances in the images (Lempitsky & Zisserman, 2010). To consider count estimation approaches, individual piles are manually segmented from the background and only pixels within the image region containing the stockpile are considered.

(a) (b)

Figure 4.5: Example images used for investigations of tyre categorization and count estimation

Figure 4.5a and Figure 4.5b show example images from the manually segmented images used for categorization and count estimation respectively. The remainder of this chapter investigates feature extraction and pre-processing methods for the purpose of tread categorization and count estimation based on the manually segmented images of tyre treads and tyre stockpiles.

4.2 Categorization of Tyres

Recognition is generally considered as having two types, namely, the specific case and the generic case. In the specific case, the aim is to recognize specific instances of an object. The generic case aims to recognize objects from a generic category (Grauman & Leibe, 2010). In the case of tyre recognition, a categorization hierarchy from the generic category of tyres to specific makes and models must be evaluated. The levels of the categorization hierarchy will support the task of determining which levels between the specific and generic case are suitable for categorization based on visual characteristics.

At REDISA's waste tyre depots, tyres are categorized as being either a 4x4, truck, or passenger tyre.

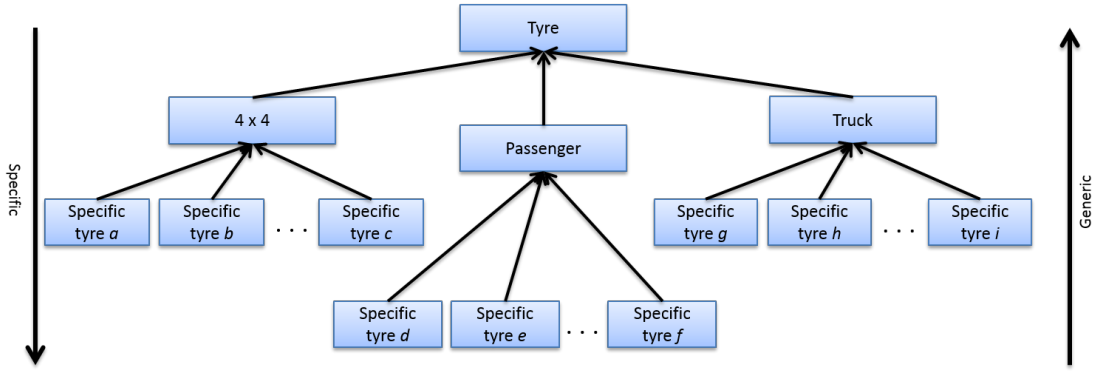


Figure 4.6: Hierarchy of tyre categorization to guide recognition and detection feature selection

Figure 4.6 shows a possible tyre category hierarchy. At tyre depots, REDISA separates tyres according to the second level of the hierarchy in Figure 4.6 (4x4, truck, or passenger tyre). Going from the top level in the hierarchy to the bottom level in the hierarchy is considered going from a generic object category of tyres to specific instances of tyres.

A metric based on the feature-matching distances over a data set is proposed to determine the suitability of various local feature detector-descriptors (Section 4.2.1). The metric is applied to a data set containing multiple images of tyre treads separated according to their specific level of categorization to make a hypothesis about which local feature detector-descriptor combination would provide the best separation of categories (Section 4.2.2). Finally, the feature detector-descriptor combinations for further investigation are discussed (Section 4.2.3) with a hypothesis for the specific case of categorization.

4.2.1 A Metric for Feature Detector-Descriptor Suitability

To decide which feature detector-descriptor combinations to use for categorization in the BoVW model (Section 3.8), a metric is required to determine the suitability of feature detector-descriptor combinations. A commonly used measure of distance between data points for k-means clustering in the BoVW model is the Euclidean distance. In feature matching, a commonly used measure to determine the similarity between features is the Sum of Squared Differences (SSD) which is equivalent to the Sum of Squared Error (SSE). For a perfectly matched feature, the SSD between the two features is equal to

zero. The Euclidean distance is related to the SSD in that the SSD can be viewed as the squared Euclidean distance (Derpanis, 2005).

To determine suitable feature detector-descriptor combinations out of the potential candidates for use in the BoVW model with k-means clustering, the ratio of the MSE for matched feature pairs on tyres of the same category to the MSE for matched feature pairs of tyres in different categories is proposed. Images that fall into the specific case are used to illustrate this concept.

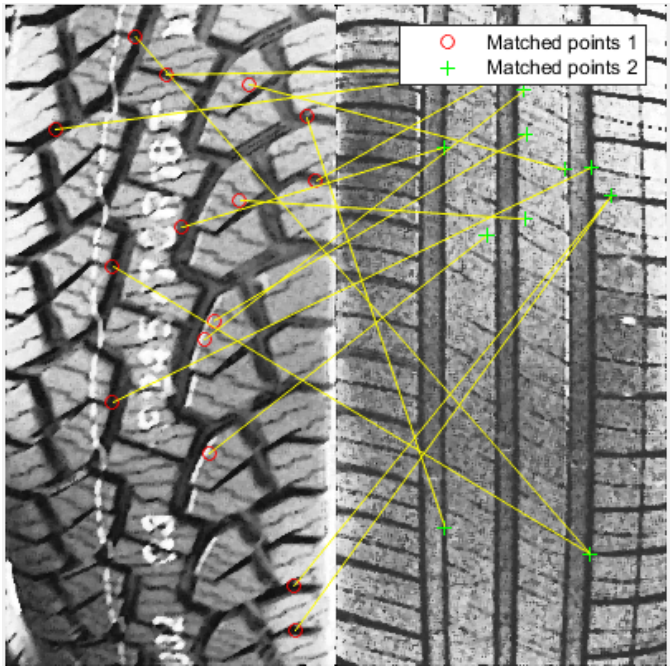


Figure 4.7: Matching of features for tyres belonging to different categories

Figure 4.7 shows the best feature matches between two tyres of different categories. Best matches refers to the feature pairs with the lowest SSD. The image features were detected and described by using the Fast-Hessian detector with the SURF descriptor and their similarity calculated using the SSD of matched feature pairs. It can be seen that the matched features have low similarity in terms of their visual appearance. The low similarity in visual appearance means that the SSD between matched feature pairs will be high. The MSE for matched feature pairs should provide an indication of the feature pair similarity between the two images.

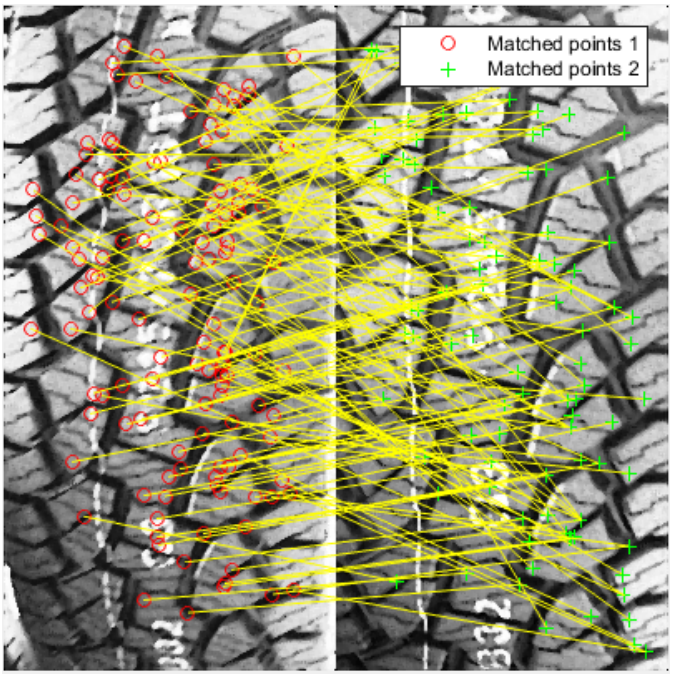


Figure 4.8: Matching of features for tyres belonging to the same category

Figure 4.8 shows the best feature matches between two tyres of the same category. The image features were also detected and described by using the Fast-Hessian detector with the SURF descriptor and their similarity calculated using the SSD. It can be seen that the matched features are visually similar. The high similarity in visual appearance means that the SSD between matched feature pairs will be low. The MSE between all features should therefore be lower in comparison to the MSE of matched features from two different tyre categories.

For feature detector-descriptor combinations to be ranked in terms of their suitability for separating tyre categories, the MSE for matched features for tread images of the same category should be low and the MSE for matched features for tread images of different categories should be high. The ratio of the MSE for same category matches to the MSE for different category matches can be stated as:

$$R(a,b) = \frac{MSE_w}{MSE_d} \tag{4.1}$$

The ratio R is considered a function of the feature descriptor(a)-detector(b) combination. MSE_w is the MSE for a matched feature pair within the same category while MSE_d is

the MSE for a matched feature pair where each feature in the pair is from a different category tread image. Feature detector-descriptor suitability can then be ranked in ascending order of their resulting R values.

4.2.2 Selecting Features for Tyre Categorization

The discussion on feature detectors in Section 3.5 resulted in the selection of candidate local feature detectors. The local feature detectors that form part of the candidate local feature detectors are the Harris, Harris-Laplace, DoG, Fast-Hessian, and MSER. The candidate feature descriptors are the SIFT descriptor and the SURF descriptor. To evaluate local feature descriptor-detector combinations for use in the BoVW model for categorization, the level of categorization according to the categorization hierarchy must be considered.

Specific Tyre Instance Categorization

In the case of specific recognition where the aim is to recognize exact tyre matches, tyre images were organized into classes, with one class per tyre instance at the specific level in the category hierarchy. A single image was chosen from each class as a representative of that tyre instance. Each representative image's local features were matched with the rest of the local features of the other images in its category and the MSE for matched local feature pairs from same category images was obtained. The process was repeated by matching each representative image to the images in other classes and the MSE for matched local feature pairs from different categories was obtained. The results of the ratio R for the detector-descriptor combinations is summarized in Table 4.1:

Descriptor \ Detector	Harris	Harris-Laplace	DoG	Fast-Hessian	MSER
SIFT	0.91	0.87	0.87	0.92	0.96
SURF	0.81	0.85	0.82	0.82	0.90

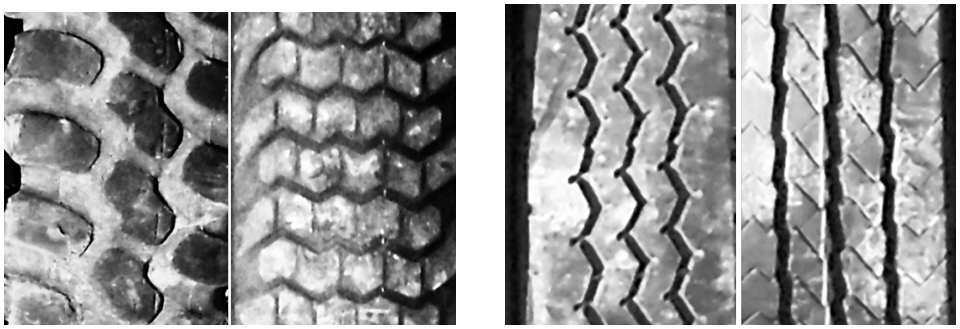
Table 4.1: Ratios of MSE for the matching feature pairs in the category to matched feature pairs in different categories for detector-descriptor combinations.

The ratios of MSE_w and MSE_d for the various detector-descriptor combinations show that the Harris detector combined with the SURF descriptor minimizes the ratio of MSE_w to MSE_d. The results indicate that for recognition of specific tyre treads, the Harris

corner detector combined with the SURF descriptor should provide a good separation of image patches around Harris point features within the BoVW model.

Categorization Into 4x4, Truck, and Passenger Categories

In the case of categorizing tyres as either 4x4, passenger, or truck tyres the technique based on the MSE ratios for feature detector-descriptor combinations may not be a good indicator of detector-descriptor suitability since tyres within the same category have different tread patterns. The regions around detected points will be different across treads that fall into the same category. For categorization at a higher level than specific instances, the creation of visual word frequency histograms capture more general characteristics than specific tread patterns.



(a) (b)

Figure 4.9: Tread examples for 4x4 and truck categories.

Figure 4.9a shows two examples of 4x4 treads. It can be seen that the structural characteristics differ more than for tyres of the same instance. Figure 4.9b shows two examples of truck treads which also differ more than two tyres belonging to the same category at the specific level. For this reason there is uncertainty as to how the clusters could be formed in the visual vocabulary. The histograms of visual word occurrences may be able to distinguish between treads based on characteristics such as the smooth or sharp gradient changes within regions determined by the local feature detectors. It is not clear which detector-descriptor combinations would be the most suitable for the categorization of tyres into 4x4, truck, or passenger tyre categories.

4.2.3 Discussion

A metric to determine the suitability of particular detector-descriptor combinations for the purposes of instance level recognition was presented (Section 4.2.1). It is expected

that image points detected and described by each of the local feature detectors and descriptors will be similar for images that contain the same instance of tyre tread. For higher level categorization such as the categorization into 4x4, passenger, or truck tyres it is uncertain which combination of local feature detector and descriptor will yield the best results when used in a BoVW model. The discussions of tyre categorization are used to hypothesize that in the specific case, the Harris detector with SURF descriptors will provide a suitable detector-descriptor combination for instance level categorization. It is uncertain which detector-descriptor combination will yield the best results for higher level categorization in the BoVW model and will be determined from experimentation.

4.3 Count Estimation of Visible Tyres in Stockpiles

The task of estimating the number of visible tyres in tyre stockpile images requires features that are relevant to the task to be extracted. In Section 4.1 issues associated with estimating the number of visible tyres through detection and segmentation were discussed. The counting by regression strategy avoids the difficult problems associated with detecting and segmenting individual tyres in images of tyre stockpiles for the purpose of count estimation. A counting by regression strategy is therefore chosen for the count estimation problem.

In order to create a non-linear mapping from image features to count estimates, appropriate features need to be extracted. Typically counting by regression approaches attempt to find a mapping from a set of global image characteristics to a real-valued output where the features are described by histograms (Lempitsky & Zisserman, 2010). This section reviews tyre stockpiles in terms of features that could be appropriate for use with a neural network to create a non-linear mapping from image features to an overall count estimate. Appropriate features are features that could be used for non-linear function approximation where the function takes the feature descriptor as input and can produce an accurate prediction about the number of visible tyres in the stockpile image described by the feature descriptor.

4.3.1 Feature Selection

The extraction of suitable features from images of stockpiles requires descriptions of the structural characteristics of the stockpiles. Colour information cannot be relied on due to the similarity of colour across the stockpile as a whole. The similarity in

colour also creates issues for edge detectors in that object boundaries are difficult to determine. A summary description of structural characteristics that does not rely only on object edges is required. The HOG feature descriptor (Section 3.7.1) is global feature descriptor that could potentially capture structural information across tyre stockpiles without relying on edges or distinct colour changes that represent the object boundaries. Local feature detectors and descriptors do not seem as suitable in the context of stockpile count estimation as different tyres have different tread complexities and two tyres may result in greatly different numbers of features being detected on their tread.

4.3.2 Summarizing Global Stockpile Structure with HOG features

In order to summarize the structure of stockpiles resulting from a series of individual tyres being piled on top of each other, gradient orientation information could be used as a global descriptor. The HOG descriptor could provide the required structural information for stockpiles. An issue with describing an image of a stockpile as a whole is that the HOG feature vectors become prohibitively large. In order to reduce the size of feature vectors describing a stockpile, it could be possible to extract HOG features for small regions over the image and use those features in a BoVW model for feature vector creation.

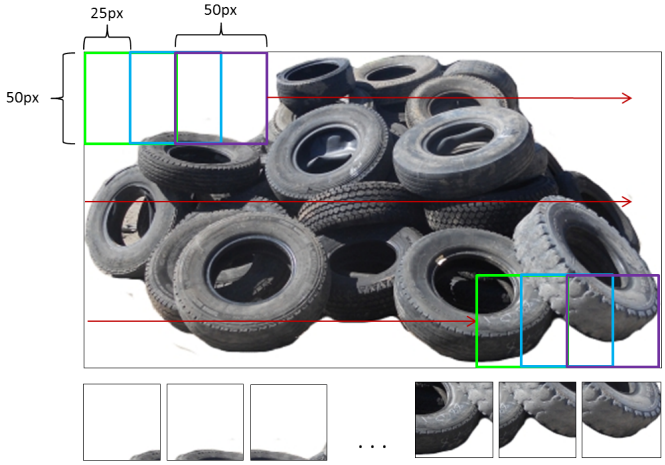
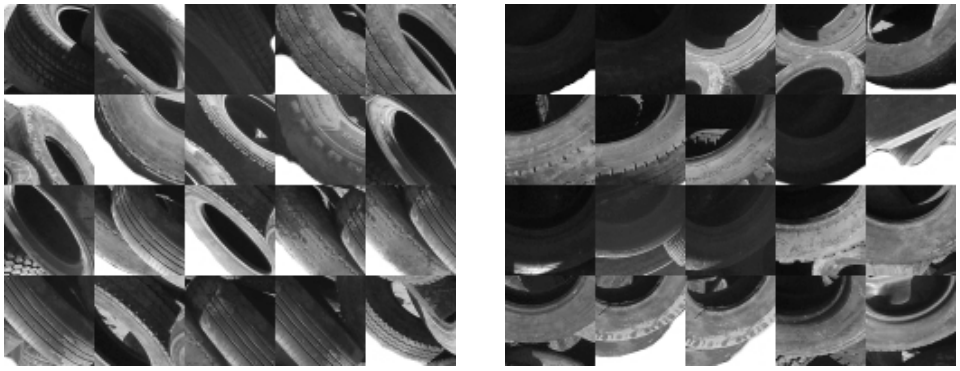


Figure 4.10: Extraction of overlapping image patches.

Figure 4.10 shows how the image patches from the tyre stockpile are extracted. Each image patch is collected from the image using a sliding window (or dense grid) with cell sizes 50 pixels wide and 50 pixels high with each cell shifted by 25 pixels in the x direction and 25 pixels in the y direction. Once the image patches have been extracted, each patch is described using the HOG feature descriptor. The HOG feature vectors are

then clustered and the cluster centres form the visual vocabulary as in the BoVW model.



(a) (b)

Figure 4.11: Two clusters of image patches formed by k-means clustering on the HOG feature representation of the image patches

Figure 4.11 shows image patches belonging to two clusters formed by clustering the individual patches HOG feature representation using k-means. Each cluster centre in the HOG feature space is then considered a visual word. The cells collected for each image are associated with a cluster centre as in the BoVW model to create a visual word occurrence histogram for each image. Feature vectors used as input data for creating a non-linear regression model are then histograms containing the visual word occurrence frequencies for each training image with a label specifying the ground truth in terms of the visible number of tyres.

4.3.3 Summarizing Global Stockpile Structure with Other Histograms

An alternative approach to describing stockpiles using HOG feature cluster centre occurrences is to use histograms constructed from other structural information such as the approach used by Kong *et al.* (2006) described in Section 3.10.3. They use concatenated blob size histograms and edge orientation histograms of pedestrians after segmentation by means of a foreground mask. Based on the selection of features by Kong *et al.* (2006), concatenated histograms of the gradient magnitudes, gradient orientations, and edge orientations are proposed as a possible feature description technique.

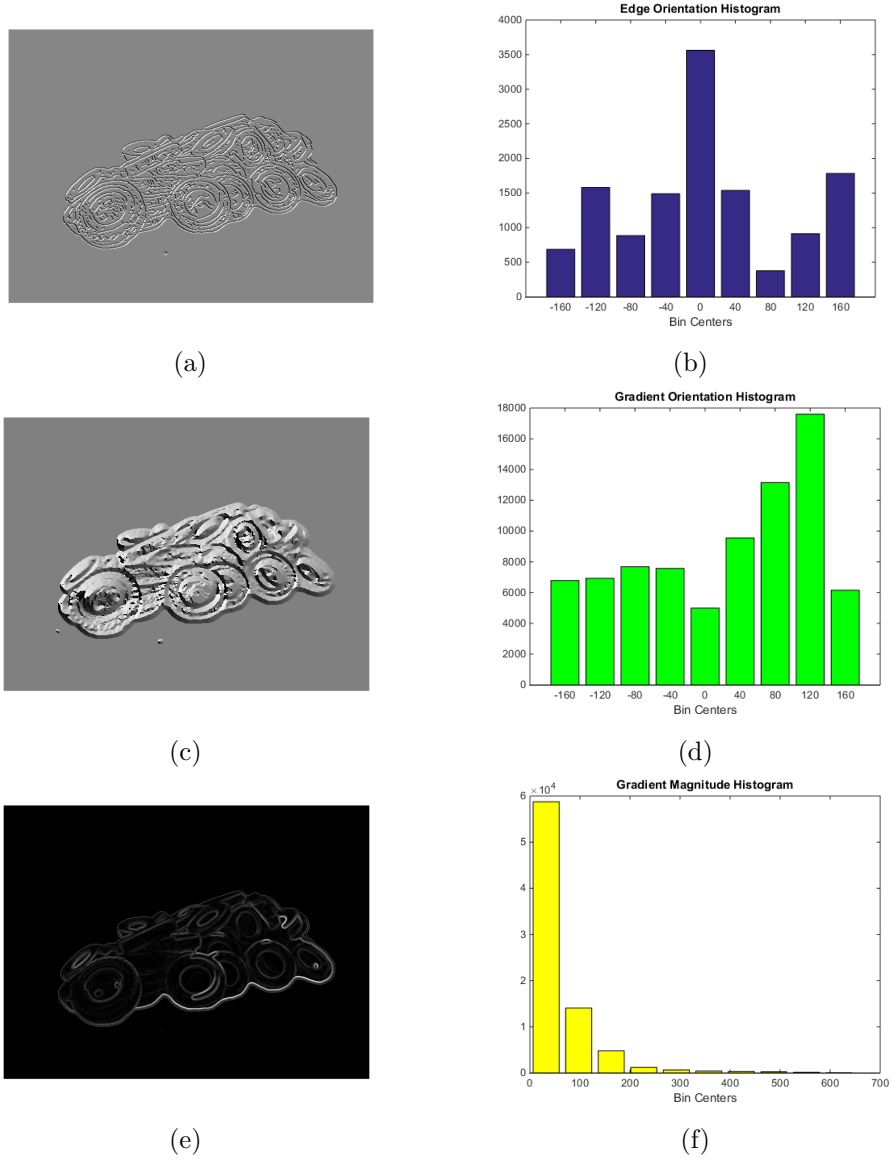


Figure 4.12: Image representations of the (a) edge orientations and (b) edge orientation histogram, (c) gradient orientations and (d) gradient orientation histogram, and (e) gradient magnitudes and (f) gradient magnitude histogram.

Figure 4.12 shows three images with their corresponding histograms. Figure 4.12a shows the image representation of the edge orientations and its corresponding histogram is shown in Figure 4.12b. Edge orientations of zero are represented by gray pixels. As the edge orientations go towards -180, pixel intensities goes towards black and as the edge orientations go towards 180, the pixels become whiter. The histogram constructed from

the edge orientation image captures the distribution of edge orientations, focusing on where the image gradient is maximal and the conditions for good edge detection are satisfied.When creating the histogram of edge orientations, only pixels that correspond to edge pixels found using Canny edge detection are binned.

Figure 4.12c shows the image representation of the gradient magnitude. The pixel values of the gradient magnitude image can be interpreted in the same was as for the edge orientation image. The gradient magnitude histogram (Figure 4.12d) captures the distribution of the gradient magnitudes across the entire stockpile. When creating the gradient orientations histogram, only the gradient orientation pixels corresponding to non-zero gradient magnitude pixels are binned.

Figure 4.12e shows a representation of the magnitude of the image gradient. Black pixels in the gradient magnitude image correspond to image gradients of zero. Gray pixels represent gradients that are greater than zero but less than the maximum gradient. White pixels indicate sharp intensity changes. When creating the gradient magnitude histogram, only the gradient magnitude pixels corresponding to non-zero gradient magnitude pixels are binned.

The three image are used to produce the three histograms summarizing the edge orientation, gradient orientation, and gradient magnitude data for a single stockpile image. The three histograms are concatenated to form a final global image feature vector that can be used as input for a machine learning algorithm.

4.3.4 Discussion

Feature selection for image features should not rely on colour information due to the similarity of the colours of tyres in stockpiles (Section 4.3.1). Due to the colour similarity across tyres, feature descriptors that summarize structural characteristics of tyre stockpiles are considered.

Two candidate feature representations have been discussed for summarizing the structural characteristics of tyre stockpiles in images. It is not evident which image representation will result in better results although they both aim to summarize the structural characteristics of the stockpiles through gradient information.

The HOG feature descriptors result in high dimensional feature descriptors. To produce a lower dimensional feature descriptor for summarizing global stockpile structure, HOG representations of image patches can be used in a BoVW model. Using the HOG descriptor in a BoVW (Section 4.3.2) allows the creation of a histogram where each bins represents the occurrences of structurally similar image patches thus providing a histogram representing the distribution of different structures in the tyre stockpile image.

The second feature descriptor that was discussed provides a global descriptor that makes use of image gradient information. Unlike the grid of HOG features in a BoVW model, the custom histograms (Section 4.3.3) do not split the image into sub images but instead the gradient data at the pixel level are considered. The custom histograms only take into account gradient data where the image intensity is non-uniform and therefore summarizes data, from a pixel level, only where there are intensity changes that correspond to tread regions and tyre boundaries.

4.4 Conclusions

This chapter discussed the use of the computer vision concepts from Chapter 3 in the context of tyres and waste tyre stockpiles. Based on the discussion of the computer vision concepts in Chapter 3 and the analysis of tyre and tyre stockpile images in this chapter, suitable image features for the individual tyre categorization and tyre stockpile count estimation problems have been identified and proposed.

Segmentation and detection (Section 4.1) were discussed as a means to limit regions of images for further analysis. Specifically if individual tyres could be detected with a high level of confidence then detection could have been used as a means of counting. Segmentation could possibly have provided a means of automatically selecting pixels forming regions containing tyres, either separating individual tyres from each other or separating a tyre pile from the background. An analysis of the visual characteristics of tyre stockpiles indicated that both detection of individual tyres in a pile and segmentation of individual tyres, or segmentation of a tyre pile from the background, are challenging tasks. An analysis of tyre stockpiles considering the limitations of state of the art detection systems was given in Section 4.1.1. Characteristics of tyre stockpiles such as the number of different appearances of the tyres due to occlusion and clutter or the similar colour of the tyres result in a challenging detection task. In a sliding window approach to detection, various aspect ratios would need to be searched, for

example different sized bounding rectangles would be required for fully visible tyre and a tyre that is fifty percent occluded. An exhaustive search through aspect ratios, and detectors trained for a number of possible appearances would be too computationally expensive and makes the assumption that all possible appearances could be accurately enumerated and the bounding rectangles could be accurately merged so that there would be a one-to-one correspondence from bounding rectangle to tyre. Attempting to segment individual tyres from each other or from the background results in over-segmentation or under-segmentation. The main contributing factors to the over- and under- segmentation are the varying conditions of the tyre tread and varying illumination of the scenes.

The analysis of the visual characteristics of tyre stockpiles while considering the limitations of computer vision algorithms discussed in Chapter 3 leads to a much more refined scope for categorization of individual tyres and estimating the number of visible tyres in an image of a waste tyre stockpile. Since automatic segmentation and detection of regions of interest could have provided an avenue for estimating the number of visible tyres in an image, the complexities associated with varying environmental variables (such as lighting conditions) and domain variables (such as varying tread complexity) prompts the use of manual segmentation for separating tyre stockpiles from the background before visible tyre count estimation. The other conclusion drawn from the discussion of segmentation and detection in the context of tyre stockpiles is that out of the three count estimation strategies, counting by regression is the most appropriate.

In Section 4.2 the categorization of individual tyres was discussed. The level of categorization that can be performed is determined by the visual differences between the levels of categorization. Two cases for tyre categorization will be considered, namely the specific case where a particular tyre model is categorized and the general level where a tyre is labelled as being a 4x4, passenger, or truck tyre. The categorization depends on whether there are distinguishing enough features that can be discovered and described to separate tyre treads into their respective categories at the two levels of categorization. A metric for feature detector-descriptor suitability based on MSE ratios of and detected feature similarities was proposed in Section 4.2.1. This metric is used to determine which feature detector-descriptor combination is the most suitable for use in the BoVW model. The results on the tyre tread datasets were similar and thus all of the considered detector-descriptor combinations will be tested in the BoVW model.

Count estimation of visible tyres in Stockpiles was discussed in Section 4.3. Due to the

issues associated with counting visible tyres in tyre stockpiles by using detection and segmentation, a counting by regression approach will be used. Counting by regression approaches typically attempt to find a mapping from global image characteristics to real-valued output where the image characteristics are described by histograms. Two global feature descriptors are proposed for the purpose of counting by regression. Feature selection (Section 4.3.1) is be based on structural characteristics of the stockpile. The feature descriptions do not rely soley on colour due to the colour similarity across tyres. Summary descriptions that do not rely soley on edges or colour characterisitcs but rather a combination of gradient orientations and magnitudes are used. The BoVW model using global descriptions of grid cells is proposed for the first descriptor. The HOG descriptor provides a descriptor for image patches that can, with a fair degree of accuracy, separate image patches into clusters of structurally similar image patches (Section 4.3.2). A second descriptor is proposed and consists of concatenated histograms of gradient magnitudes, orientations and edge orientations (Section 4.3.3). The second descriptor captures structural information about the stockpile with a faster training training time over that of grid of HOG descriptors since many training steps in the BoVW for creating feature vectors, such as the clustering process, are not required. It is unclear which descriptor will result in more accurate count estimations, therefore experiments will be conducted for each of the two proposed descriptors.

Chapter 5

Machine Learning for Tyre Classification and Count Estimation

Through the discussions of computer vision concepts and algorithms in Chapter 3, machine learning algorithms for object categorization and count estimation in images were identified. A critical analysis of the identified machine learning algorithms is required to form a foundation for the creation of models for the purposes of tyre categorization and visible tyre count estimation in images. This chapter addresses the fourth research question:

RQ_4 How can machine learning methods be used for the categorization of individual tyres and estimating the number of visible tyres in tyre stockpiles?

In the context of this work, focus is given to machine learning using feature vectors for categorizing tyres and count estimation of visible tyres in images. Producing feature vector representations of the images allows data that is not useful for the particular count estimation or categorization task to be omitted from the final feature vector description of the image. In Chapter 3 computer vision concepts that are considered applicable for tyre categorization and stockpile count estimation were discussed. The major concepts discussed in Chapter 3 were:

1. Top-down and bottom-up approaches to vision,
2. Image feature extraction, and

3. Object recognition, detection, and counting strategies.

The BoVW model that is a standard model for categorization makes use of three algorithms for image classification. The machine learning algorithms used in the BoVW model for image classification are k-means clustering (Section 2.4.3) for producing a finite number of visual words, a nearest-neighbour search (Section 5.1.1) for for creating visual word occurrence histograms, and SVMs (Section 5.1.2) for classifying the quantized feature vectors. This chapter discusses the algorithms involved in the BoVW model for tyre classification in Section 5.1.

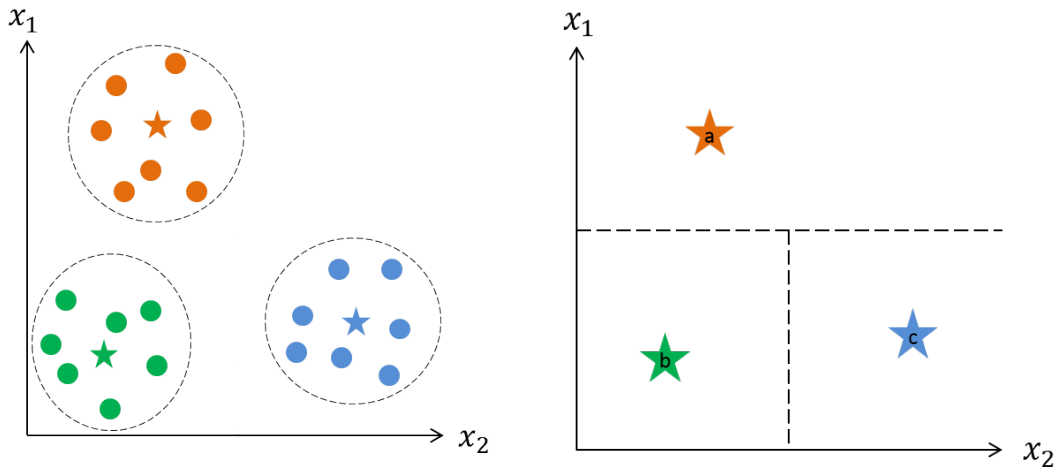
The approach for the counting of pedestrians by Kong *et al.* (2006), that is discussed in Section 3.10.3, indicates that feature descriptions of global characteristics of the objects being counted can be used by a neural network to approximate a non-linear function from global image features to count estimations. In the context of this research the application of a neural network can be classed as a function approximation problem where the aim is to learn a functional relationship from input feature vectors produced from tyre stockpile images to real-valued count estimations. This chapter discusses the use of neural networks for count estimation in Section 5.2.

5.1 Categorization Through Classification

In the BoVW model, the problem of visual categorization is reduced to a multi-class classification problem in which there are as many classes as there are visual categories. Csurka *et al.* (2004) compared two classifiers, the naïve Bayesian classifier and the support vector machine (SVM). Csurka *et al.* (2004) found that the SVM showed superior performance for categorization compared to the naïve Bayesian classifier. In general, the process for the BoVW begins by extracting and describing features in a set of training images. The features that are found are then clustered using the k-means clustering algorithm. K-means is discussed in Section 2.4.3 for the three-dimensional case although for clustering image features, the clustering is done in p-dimensional space where p is the length of the feature vector. This section describes the use of k-nearest neighbours (Section 5.1.1) for assigning feature descriptors extracted from images to one of the cluster centres formed by the k-means algorithm as well as describing the SVM algorithm (Section 5.1.2) for binary classification and its extension for multi-class classification.

5.1.1 Nearest-Neighbour Searches For Building Visual Word Occurrence Histograms

The cluster centres formed by k-means clustering on the features found in the training images are regarded as visual words. To create histograms of visual word occurrences to be used as feature vectors describing an image, each of the features found in the image must be related to a visual word by determining which cluster centre is closest. The most commonly used measure of closeness is the Euclidean distance between the query feature vector and the training feature vectors. The candidate feature vector is considered to be the visual word represented by the cluster centre to which it is the closest. The nearest neighbour search can be performed by comparing a query feature vector to each of the cluster centres formed by the k-means algorithm. Although comparing each query image feature to each cluster centre ensures that the true nearest neighbour will be found, the time complexity is linear and can be improved by using a tree structure. Techniques using a K-d trees have been used to provide a fast approximate nearest neighbour (FLANN) search to improve on the time taken to return the results of a nearest neighbour search (Han *et al.*, 2011).



(a) (b)

Figure 5.1: Visualization of clusters in two-dimensions and data space partitioning based on cluster centres (adapted from (Han *et al.*, 2011; Vedaldi & Fulkerson, 2010))

An example of k-means clustering is presented in Figure 5.1. The figure shows a visualization of three clusters formed by k-means in a 2D data space. In order to bound the search time, an approximation using k-d trees that recursively splits the data space

by decision boundaries can be used. Figure 5.1b shows a possible partitioning of the data space to separate the cluster centres by decision boundaries. A typical procedure for creating a k-d tree for the purpose of nearest neighbour searches proceeds by starting with a complete set of data points. A balanced binary tree is created by iteratively splitting the data set in two about the mean or median values of the dimensions that exhibit the greatest variance (Beis & Lowe, 1997). Each internal node stores the dimension and median value that was used to split the data at that node.

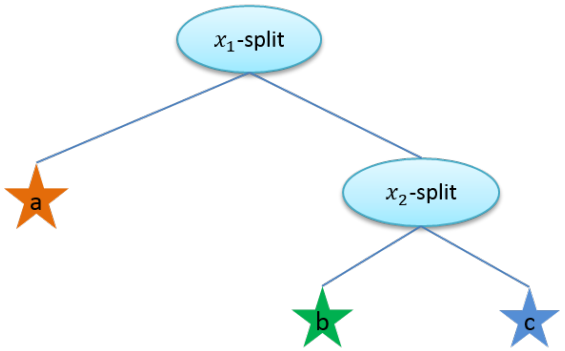


Figure 5.2: K-d tree used for approximate nearest neighbour matching of two-dimensional vectors to the cluster centres shown in Figure 5.1b

Figure 5.2 shows how the partitioning of the two-dimensional feature space in Figure 5.1b can be represented by a k-d tree. Partitioning using k-d trees is an approximate partitioning which results in approximate nearest neighbour search results. In the BoVW model, the query feature vector is considered to be an instance of the visual word represented by its nearest neighbour. In the example k-d tree in Figure 5.2, given a two-dimensional feature vector, if its x_1 value is greater than the value of the x_1 split then it is counted as an occurrence of the visual word (cluster centre) shown by star a. Otherwise if it is less than the x_1 split then determine if its x_2 value is less than or greater than the value representing the x_2 split. If it is less than the x_2 split value then it is an occurrence of the visual word represented by star b, otherwise it is an occurrence of the visual word represented by star c. The resulting nearest neighbour is an approximate nearest neighbour and the probability of finding the true nearest neighbour can be improved using a priority search.

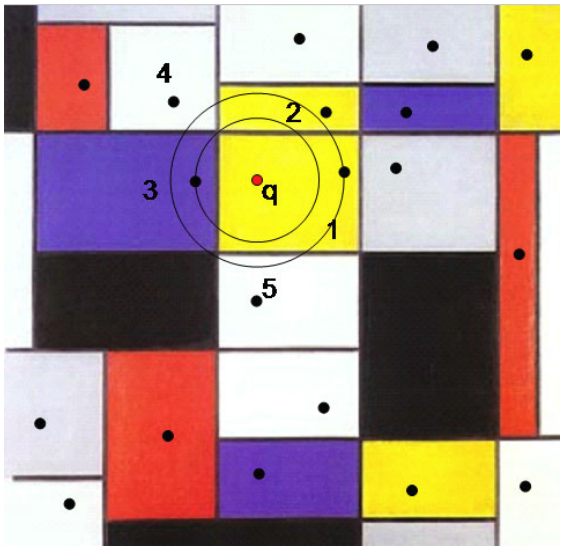


Figure 5.3: Search space partitioned using k-d tree (Silpa-Anan & Hartley, 2008)

Given a query point the tree is descended using the dimension i and median or mode value m to determine the cell in which the query point falls. The query point may not fall into the cell that contains the true nearest neighbour of the query point but contains the first candidate nearest neighbour. The priority search then proceeds to search other cells in order of their distance from the query point and within the hypersphere of radius r where r is initially defined as the distance from the query point to the first candidate nearest neighbour. The radius r is then adjusted when a candidate nearest neighbour is found within the search radius with a lower distance from the query vector. The search terminates when all the cells within the radius r have been searched. Figure 5.3 shows how the query vector q falls into cell 1 while the correct nearest neighbour is in cell 3. The outer circle corresponds to the initial radius while the inner circle corresponds to the final radius where the vector in cell 3 is found to be the nearest neighbour in the cells considered in the search. Silpa-Anan & Hartley (2008) extend this technique to improve the accuracy and speed of a nearest neighbour search by producing several k-d trees each using a randomly rotated data sets. Nearest neighbour searches are then performed simultaneously on each of the trees to determine the nearest neighbour.

5.1.2 Support Vector Machines (SVM)

The Support Vector Machine (SVM) is a classifier that was introduced by Boser *et al.* (1992). The SVM is a binary classifier that can be used for the classification of both linearly and non-linearly separable data. For non-linearly separable data, the input data

is transformed to a higher dimensional space. The SVM searches for a maximal margin hyperplane (MMH) between two classes of data. The SVM takes a training set D that consists of feature vectors and their corresponding labels as $(X_1, y_1), (X_2, y_2), ..., (X_D, y_D)$ so each X_i has a corresponding y_i. The values of y_i are given as +1 for positive samples and -1 for negative samples.

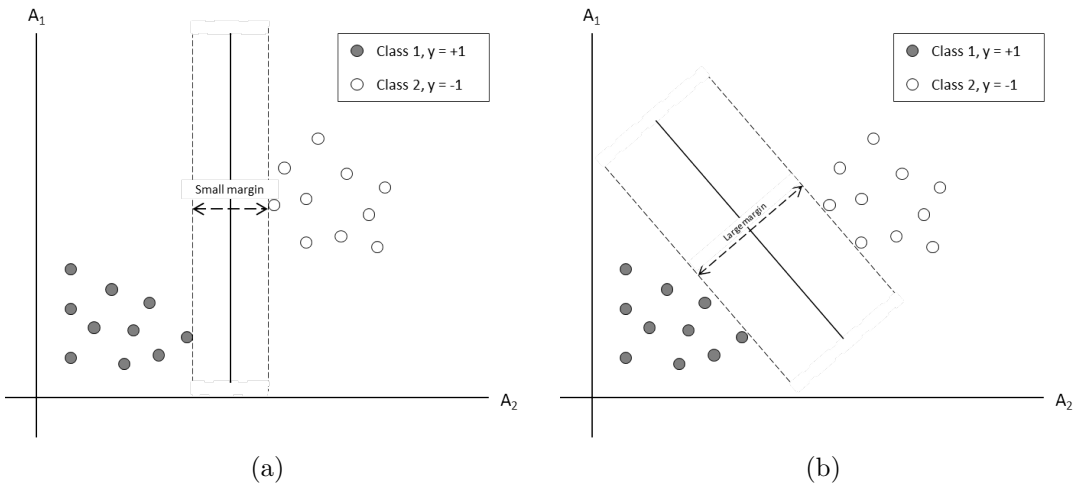


Figure 5.4: Visualization of decision boundary margins for two dimensional data points (Han *et al.*, 2011)

Figure 5.4 shows two examples of possible separating hyperplanes. Figure 5.4b shows the better decision boundary of the two since the margin is larger which leads to greater generalization accuracy. A separating hyperplane is written as:

$$b + W \cdot X = 0 \tag{5.1}$$

where W is a weight vector, X is the feature vector, and b is the bias. The weights defining the hyperplane can be adjusted so that the margin sides are written as:

$$H_1 : b + W \cdot X \geq 1 \text{ for } y_i = +1 \tag{5.2}$$

$$H_2 : b + W \cdot X \leq -1 \text{ for } y_i = -1 \tag{5.3}$$

The two regions of the data separated by the MMH are given by:

$$y_i(W \cdot X + b) \geq 1, \forall i \tag{5.4}$$

The MMH is based on the support vectors which are data points that lie on either side, H_1 or H_2 of the margin. The distance from any point on H_1 to the separating hyperplane is $1/||W||$. The same distance from a support vector to the margin side H_2 indicates that the width of the maximal margin can then be given as $2/||W||$. The support vectors and MMH are then found by solving a Legrangian formulation using Karush-Kuhn-Tucker (KKT) conditions. After training, a new sample can be classified by determining which side of the hyperplane the sample lies. This can be determined using:

$$b + W \cdot X \tag{5.5}$$

If the result is greater than zero, the sample is classified as positive while a result of less than zero indicates a negative classification. The SVM can be extended to multi-class classification by employing a one-vs-all (OVA), all-vs-all (AVA), or error correcting output codes (ECOC) strategy with multiple SVMs. SVMs typically train slowly but are very accurate and less prone to over-fitting than other classifiers. Another advantage is the compact description of the learned model in the form of the weight vector (Han *et al.*, 2011).

Extension to a multi-class classifier

Three main approaches exist for extending binary classifiers to multi-class classifiers (Han *et al.*, 2011):

1. One-vs-all,
2. All-vs-all, and
3. Error correcting output codes.

In the *one-vs-all (OVA)* extension of binary classifiers, a classifier is trained for each class. Each classifier j is trained using feature vectors from class j as positive examples while feature vectors from all other classes are considered negative training examples. Each classifier learns to separate its corresponding class from examples in other classes. Given a candidate query vector the query vector can be classified in one of two ways. The first way is to assign the query vector the class of the classifier with the highest output is taken as the class label for the query vector (Csurka *et al.*, 2004). An alternative way to assign a class label is to vote for each class according to classifier outputs. If the classifier classifies vector X as negative for class j then all other classes get a vote. If the classi-

fier classifies a vector X as positive for class j then class j gets one vote (Han *et al.*, 2011).

The *all-vs-all (AVA)* method of extending binary classifiers to multiclass classifiers is used to construct a classifier for each pair of classes with one class of the pair being considered positive and the other class considered negative. To assign a class label to a candidate query vector, each classifier votes and the class with the most votes is taken as the class label for the query vector (Han *et al.*, 2011).

One issue with both the OVA and AVA approaches to extending binary classification problems is that binary classifiers are sensitive to errors (Han *et al.*, 2011). An alternative approach proposed by Dietterich & Bakiri (1994) makes use of *error-correcting output codes (ECOC)* to improve classification accuracy. In the error-correcting output codes scheme for extending a binary classifier to a multiclass classifier, each class is assigned a bit vector of length B. B separate classifiers are trained to predict each bit of the bit vector. The class label corresponding to the bit vector with the lowest Hamming distance from the bit vector produced by the B classifiers is assigned to the query vector (Murphy, 1991). ECOC provides a multiclass extension to binary classifiers that is more resistant to individual misclassifications by the individual binary classifiers than the OVA and AVA extensions (Han *et al.*, 2011; Murphy, 1991).

Evaluating Multiclass Classifiers

There are a number of evaluation metrics that can be used to measure the performance of multiclass classifiers. To provide context for the remainder of the discussion on performance measures used for multiclass classifiers, four terms that are used in calculating performance measures for binary classifiers are discussed. The four terms are (Han *et al.*, 2011):

- True positives (TP): True positives refer to positive vectors that are correctly classified. TP refers to the number of correctly classified positives.
- True negatives (TN): True negatives refer to negative vectors that are correctly classified as negative. TN refers to the number of correctly classified negatives.
- False positives (FP): False positives refer to negative examples that are classified as positive. FP refers to the number of negative vectors classified as positive.
- False negatives (FN): False negatives refer to positive vectors that are classified as negatives. FN refers to the number of positive vectors classified as negatives.

The results of a binary classifier can be summarized using these terms in a confusion matrix as shown in Table 5.1

Prediction / Truth	predicted member	predicted non-member
class member	TP	FN
not a class member	FP	TN

Table 5.1: Confusion matrix for binary classification.

The confusion matrix can be extended for multiclass classifier evaluation. Given m classes, an $m \times m$ confusion matrix is constructed. The entries along the diagonal represent the number of or the percentage of correctly classified instances. The four terms discussed above for evaluating the performance of binary classifiers can be extended to multiclass classifier performance evaluation as follows (Sokolova & Lapalme, 2009) for each class C_i:

- tp_i: The number of true positives for C_i.
- fp_i: The number of false positives for C_i.
- tn_i: The number of true negatives for C_i.
- fn_i: The number of false negatives for C_i.

Given the values of the aforementioned terms, several measures can be calculated and used to evaluate the performance of a classifier. Each measure has a different evaluation focus. Four measures for the evaluation of multiclass classifier performance have been identified, namely the:

- Average accuracy (recognition rate),
- Error rate,
- Precision, and
- Recall.

The *average accuracy* of a classifier is a measure of its effectiveness in accurately assigning class labels. The average accuracy is also known as the recognition rate (Han *et al.*, 2011).

The average accuracy can be calculated using (Sokolova & Lapalme, 2009):

$$\frac{\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{l} \tag{5.6}$$

The average accuracy is a measure of the per-class effectiveness of a classifier. The *error rate* for a multiclass classifier is used to measure the degree to which the classifier makes errors in terms of misclassifications. The error rate is also referred to as the misclassification rate (Han *et al.*, 2011). The average error can be calculated as (Sokolova & Lapalme, 2009):

$$\frac{\sum_{i=1}^{l} \frac{fp_i + fn_i}{tp_i + fn_i + fp_i + tn_i}}{l} \tag{5.7}$$

The error rate is the average per-class classification error. The *precision* of a classifier is a measure indicating the ability of the classifier to assign the correct class labels to the query vectors. For each class C_i the precision measures how many examples the classifier labeled as belonging to class C_i are actually of class C_i (Han *et al.*, 2011). The precision of a multiclass classifier is calculated as (Sokolova & Lapalme, 2009):

$$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l} (tp_i + fp_i)} \tag{5.8}$$

The precision only takes into account the true positives and the false positives, meaning that it is a measure of the agreement of actual class labels and the class labels assigned by the classifier (Sokolova & Lapalme, 2009). The *recall* is measure of the effectiveness of a classifier to correctly assign class labels by taking into account the number of false negatives for each class. The recall can be calculated as (Sokolova & Lapalme, 2009):

$$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l} (tp_i + fn_i)} \tag{5.9}$$

The recall is used to determine the average ratio, over all classes, of the correctly classified positive instances for a class to the incorrectly classified negative instances for a class.

5.2 Artificial Neural Networks (ANN)

A regression model is a model that can be fit in order to make numerical predictions about data. Neural networks can be used to produce a non-linear mapping from image feature vectors to real-valued count estimations. Artificial neural networks for numerical prediction have been used in a variety of fields with a number of applications. Some example applications include: crowd density estimation and counting (Yang *et al.*, 2014), stock predictions (Chang *et al.*, 2012; De Oliveira *et al.*, 2013; Guresen *et al.*, 2011), and neural network aided cell counting (Sjostrom *et al.*, 1999). The following section reviews artificial neural networks for the purpose of producing a non-linear mapping from input feature vectors to real-valued count estimations.

Artificial neural networks are based on the algorithmic modeling of biological neural systems. The brain is a complex system that is able to perform non-linear and parallel computations. The brain is capable of solving large complex problems and artificial neural networks attempt to mimic neural systems to solve problems. Biological neural systems consist of nerve cells referred to as neurons. A neuron consists of a cell body, an axon, and dendrites. Neurons are interconnected with the axon of one neuron being connected to the dendrite of another neuron. Signals propagate from the dendrites through the cell body to the axon and the signal from the axon is propagated to all connected dendrites. A signal is only transmitted to the axon of a cell if the neuron is activated (Engelbrecht, 2007).

Artificial neural networks have been applied to several classes of problems including classification, pattern matching, pattern completion, optimization, function approximation/time series modeling, and data mining (Engelbrecht, 2007). This research investigates the use of neural networks for the purpose of non-linear function approximation to estimate the number of visible tyres in an image based on feature vector representations of the images.

5.2.1 The Artificial Neuron

Artificial neurons are the building blocks for artificial neural networks. An artificial neuron implements a non-linear mapping from an input vector $\mathbb{R}^I$ to [0,1] or [-1,1] depending on the activation function used in the neuron (Engelbrecht, 2007). I is the number of input signals which is equivalent to the length of the input vector. The

mapping from the input vector to the output can be written as:

$$f_{AN} : \mathbb{R}^I \rightarrow [0, 1] \qquad OR \qquad f_{AN} : \mathbb{R}^I \rightarrow [-1, 1] \tag{5.10}$$

The input vector for the artificial neuron is a vector z of length I with each element of the vector representing an input signal, that is:

$$z = (z_1, z_2, ..., z_I) \tag{5.11}$$

Each entry in the input vector is associated with a weight v_i that serves the purpose of increasing or decreasing the signal of its corresponding input signal z_i. The artificial neuron calculates a net input signal and uses an activation function to compute the output signal o for the artificial neuron. The strength of the output signal is also influenced by a bias value θ.

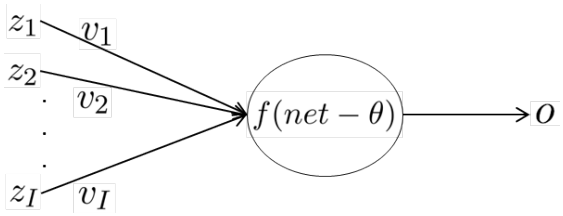


Figure 5.5: An artificial neuron (Engelbrecht, 2007)

The net input signal for an artificial neuron is typically calculated using a weighted summation of all the input signals,

$$net = \sum_{i=1}^{I} z_i v_i \tag{5.12}$$

Artificial neurons using a weighted summation of the input signals are called summation units. Alternative ways to compute the net signal for a neuron exist, such as using product units to compute the net input signal, although the most common net input function is the weighted summation of input signals (Beale *et al.*, 2014).

Once the net input signal has been calculated, the output o of the neuron is determined by an activation function. The activation function is typically a monotonically increasing mapping with the input domain $[-\infty, \infty]$ and the output range $[0, 1]$ or $[-1, 1]$.

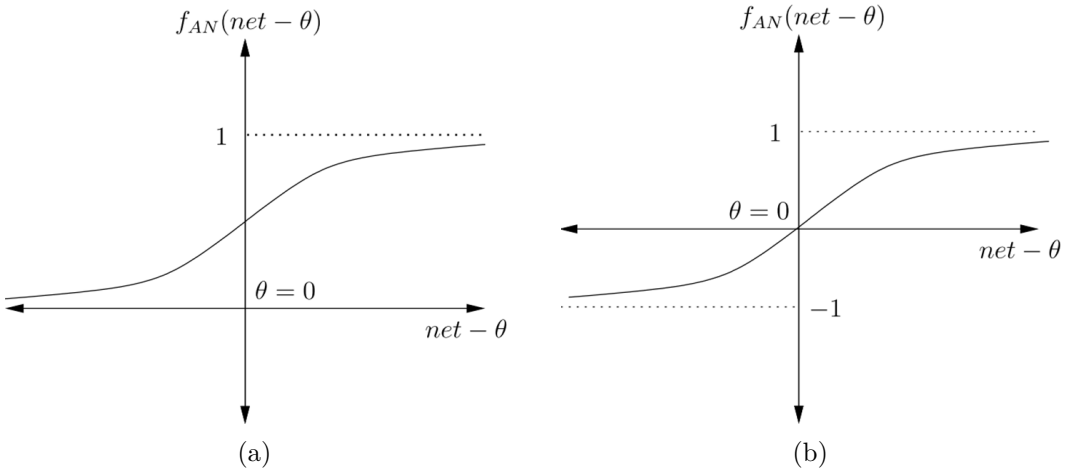


Figure 5.6: Two commonly used activation functions. The sigmoid function (a) and the hyperbolic tangent function (b)

Figure 5.6 shows examples of two commonly used activation functions. Another activation function to be considered is the linear activation function. In the case of numerical predictions, the linear activation function is not bound to the range [0,1] or [-1,1]. The results of informal experiments showed that the sigmoid activation function performs marginally better than the hyperbolic tangent function in the context of this study. The sigmoid function is defined as:

$$f(net - \theta) = \frac{1}{1 + e^{-\lambda(net-\theta)}} \tag{5.13}$$

where λ controls the steepness of the slope. It is also noteworthy that the output of the function is in the range [0,1]. Single artificial neurons can be used to realize linearly separable problems (Engelbrecht, 2007). For the purposes of producing a non-linear mapping from image feature vectors to real-valued output for count estimation, a more complex neural structure consisting of layers of neurons is required.

5.2.2 Feedforward Neural Networks

One commonly used neural network architecture is the feedforward neural network (FFNN). The FFNN is an artificial neural network consisting of layers containing artificial neurons. Typically a FFNN contains one input layer with one or more hidden layers and an output layer.

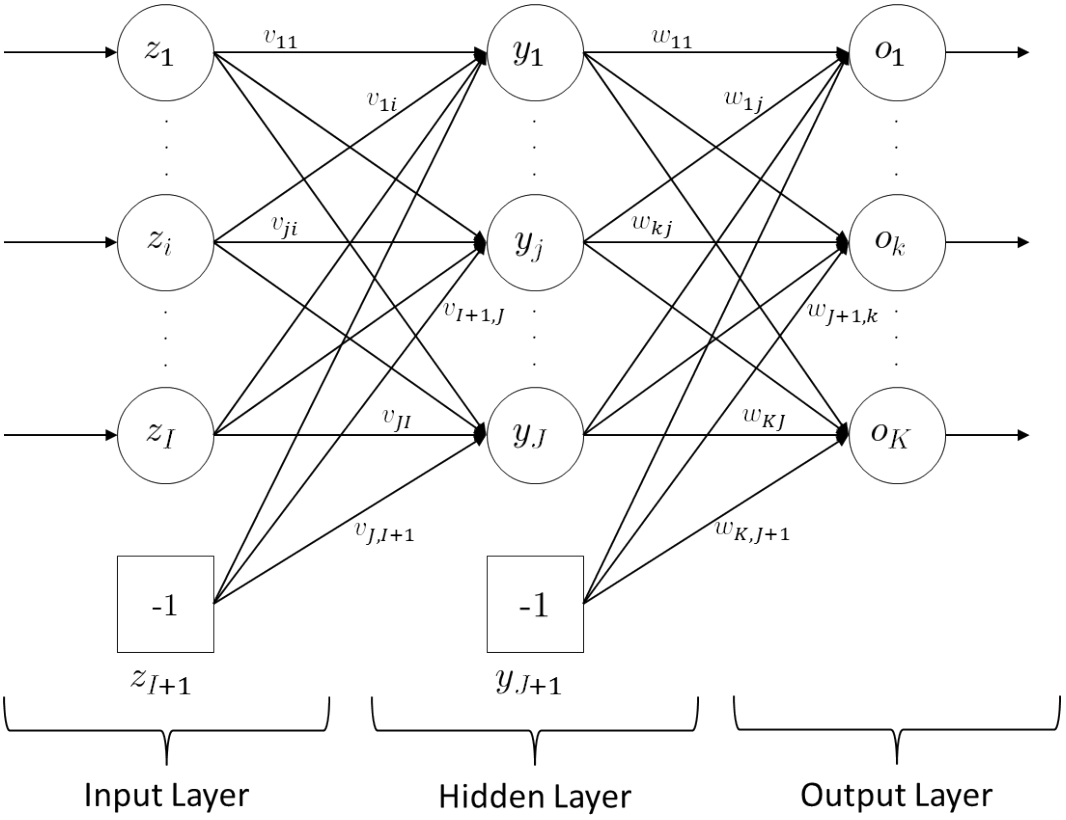


Figure 5.7: A feedforward neural network (Adapted from Engelbrecht (2007))

Figure 5.7 shows a standard feedforward neural network with an input layer, a hidden layer, and an output layer. Each of the I input signals for a single input pattern is z_i where $1 < i < I$ and z_{i+1} represents the bias input for the hidden layer. Each of the neurons in the hidden layer is represented by y_j where $1 < j < J$, also with y_{j+1} for the bias input for the output layer. The output layer contains output neurons each of which is represented as o_k where $1 < k < K$. The connections between each z_i input signal and each y_j neuron in the hidden layer is represented by a weight v_{ji}. Similarly, the connections between each hidden layer neuron y_j and each output layer neuron o_k is represented by a weight w_{kj}. For a network using summation units, each output unit o_k when evaluating input pattern p can be calculated using:

$$o_{k,p} = f_{o_k}\left(\sum_{j=1}^{J+1} w_{kj} f_{y_j}\left(\sum_{i=1}^{I+1} v_{ji} z_{i,p}\right)\right) \tag{5.14}$$

where f_{o_k} and f_{y_j} are the activation functions for the output unit o_k and hidden unit y_j

respectively. Each unit's activation function can be different and it is typically assumed that the input units have linear activation functions. The values output by the output layer units are dependent on the input pattern z_p, the sets of weights, and the bias values. To produce a non-linear mapping for input patterns to a numerical prediction, weights and bias values must be found that minimize the error of a network when presented with input patterns. Methods used to find suitable weight values and bias values for a function approximation problem fall into the category of supervised learning. Supervised learning involves updating the weight and bias values in a neural network based on input patterns and their target outputs through a training process.

5.2.3 Weight Updates Through Training

During training by means of supervised learning, a neural network is presented with training patterns and target outputs for those patterns. The aim of the training procedure is to minimize an error function by adjusting the weights associated with connections between artificial neurons in the network. A popular method for learning optimal weights is backpropagation (Engelbrecht, 2007). There are a number of variations of backpropagation in terms of the optimization techniques. Two examples are the Conjugate Gradient and Levenberg-Marquardt optimization techniques. One of the first backpropagation algorithms is the Generalized Delta backpropagation learning algorithm. The rest of this section discusses backpropagation in terms of Generalized Delta backpropagation. Generalized Delta backpropagation is based on gradient descent (GD) in that error derivatives are calculated for each weight and adjusting them in order to to move the error towards a minimum. Gradient descent optimization is a local optimizer and can potentially get stuck at local minima. The backpropagation algorithm consists of two phases in each iteration, also known as an epoch. The two phases are:

1. The **feedforward pass**, which calculates the output values of the neural network, and
2. **Backward propagation**, which propagates an error signal back from the output so that the weights can be adjusted to minimize the error.

In the feedforward pass, the network outputs are calculated according to Equation 5.14. The outputs can then be used to determine the error. A typical error function to minimize

is the mean-squared error (MSE). The error for a pattern is then:

$$\mathcal{E}_p = \sum_{k=1}^{k} (t_{k,p} - o_{k,p})^2 \tag{5.15}$$

and the MSE for an epoch is:

$$MSE = \frac{1}{P} \sum_{p=1}^{P} E_p \tag{5.16}$$

where P is the total number of training patterns and K is the number of output neurons in the network. $t_{k,p}$ represents the k-th neurons target output value and $o_{k,p}$ represents the output predicted by the k-th output neuron.

During backward propagation the error is propagated back through the network and the weights are updated accordingly to minimize the error. Assuming the use of a sigmoid function, the weights are updated using the partial derivative of the error with respect to each weight between the hidden layer and the output layer. The partial derivative is calculated using the chain rule,

$$\frac{\partial \mathcal{E}_p}{\partial w_{k,j}} = \frac{\partial \mathcal{E}_p}{\partial o_{k,p}} \frac{\partial o_{k,p}}{\partial net_{o_{k,p}}} \frac{\partial net_{o_{k,p}}}{\partial w_{k,j}} \tag{5.17}$$

$$= -(t_{k,p} - o_{k,p}) o_{k,p} (1 - o_{k,p}) y_{j,p} \tag{5.18}$$

where $(1 - o_{k,p})o_{k,p}$ is the derivative of the sigmoid activation function. The weight updates for the weights between the input layer and the hidden layer for each pattern can be calculated similarly using:

$$\frac{\partial \mathcal{E}_p}{\partial v_{j,i}} = \sum_{k=1}^{K} \frac{\partial \mathcal{E}_p}{\partial o_{k,p}} \frac{\partial o_{k,p}}{\partial net_{o_{k,p}}} \frac{\partial net_{o_{k,p}}}{\partial f_{y_{j,p}}} \frac{\partial f_{y_{j,p}}}{\partial net_{y_{j,p}}} \frac{\partial net_{y_{j,p}}}{\partial v_{j,i}} \tag{5.19}$$

$$= \sum_{k=1}^{K} -(t_{k,p} - o_{k,p}) o_{k,p} (1 - o_{k,p}) w_{k,j} y_{j,p} (1 - y_{j,p}) z_{i,p} \tag{5.20}$$

For an epoch t the weights are then adjusted after the presentation of each pattern (stochastic learning) or the weight updates are accumulated and applied after all the patterns have been presented (batch learning) (Engelbrecht, 2007). Weights are updated

according to:

$$w_{k,j}(t) = w_{k,j}(t-1) + \Delta w_{k,j}(t), \text{ where } \Delta w_{k,j} = \eta\Big(- \frac{\partial \mathcal{E}_p}{\partial w_{k,j}}\Big) \tag{5.21}$$

$$v_{j,i}(t) = v_{j,i}(t-1) + \Delta v_{j,i}(t), \text{ where } \Delta v_{j,i} = \eta\Big(- \frac{\partial \mathcal{E}_p}{\partial v_{j,i}}\Big) \tag{5.22}$$

where η is the learning rate parameter that determines the size of the step taken in the direction of descent of error. Stochastic learning spends a lot of time unlearning what what has happened during previous steps due to the fluctuation of the sign of the error derivatives. In addition, stochastic learning is affected by the order of presentation of the training patterns. To alleviate the effects of the pattern ordering, patterns need to be re-ordered for every epoch which is also a time consuming process. Two solutions to the problems associated with stochastic learning are batch learning and momentum. The momentum term is used to effectively average the weight changes to ensure that the search for optimal weights occurs in a descending direction. Batch learning provides a solution by accumulating weight changes and applying them after all the training patterns have been presented in an epoch t. The accumulation of weight updates is given as (Engelbrecht, 2007):

$$\Delta w_{kj}(t) = \sum_{p=1}^{P_T} \Delta w_{kj,p}(t) \tag{5.23}$$

$$\Delta v_{ji}(t) = \sum_{p=1}^{P_T} \Delta v_{ji,p}(t) \tag{5.24}$$

Batch learning is selected in this study as the input data is inherently noisy due to a number of factors concerning the variability of environmental conditions, camera quality, and the result of feature extraction from images.

5.2.4 Network Architecture and Parameter Selection

Using NNs to fit a model to data requires the selection of a network architecture as well as a selection of various parameters. Some of the parameters that require consideration when designing a NN and selecting parameters are the *learning rate*, the *size and number of hidden layers*, *the training set*, and the *termination criteria* (Beale *et al.*, 2014; Engelbrecht, 2007; Hagan *et al.*, 1995).

Learning rate (η)

The learning rate determines the size of the steps taken towards the minimum of the error function. Since backpropagation is a local optimization technique, a small learning rate means small weight adjustments which runs the risk of falling into a local minimum when a better solution may exist elsewhere in the weight space. If the learning rate is too large then the weight adjustments may overshoot potentially good local minima. Various methods have been proposed for automatically adapting learning rates during training (Engelbrecht, 2007). One such method for an adaptive learning rate is to increase the learning rate by some factor as the error increases and decrease the learning rate by some factor as the error decreases (Hagan *et al.*, 1995).

Size and Number of Hidden Layers

Beale *et al.* (2014); Hagan *et al.* (1995) state that almost any function can be approximated by multilayer networks provided there are a sufficient number of neurons in the hidden layers. However, there is no general way to determine how many neurons or layers are necessary. For problems such as the count estimation problem, the number of hidden layers and the number of neurons in the layers that can best approximate a function are determined experimentally. A common setup for numerical prediction is to use one or more hidden layers with sigmoid activation functions and an output layer using a linear input activation function (Beale *et al.*, 2014). Increasing the number of neurons in the hidden layer(s) increases the dimensionality of the weight vector and impacts the time taken for training.

The Training Set

The training set should contain samples that are representative of the population. Large training sets will allow network training to provide better approximations of the underlying function by negating noise in the data. Suitably sized training sets are required to accurately approximate a function. A tradeoff is often required between the amount of available data and the training time and performance of the NN.

The Termination Criteria

The process of optimizing the set of weights to minimize the error requires conditions stipulating when training should end (Engelbrecht, 2007; Hagan *et al.*, 1995; Han *et al.*, 2011). Typically used terminating criteria used to determine when training should stop are the

epoch limit, the *minimum training error threshold*, and when the error on the *validation* set does not change over a number of iterations.

- *Epoch limit*: The epoch limit is used to ensure that training will eventually terminate even if a satisfactory error threshold is never met or the algorithm becomes stuck in a local minimum.
- *Training error threshold*: The training error threshold is used to terminate training when the error of the network is below an acceptable threshold.
- *Validation checks*: Overfitting refers to the network essentially memorizing the training patterns. A subset of the training patterns are kept for the purposes of testing the network on unseen data (patterns that are not presented for the purpose of tuning the network weights). The unseen patterns are used to test the generalization ability of the network. The point at which the network performance continues to improve for the training set but stays constant or decreases for the test set is the point at which overfitting has started to occur. Early stopping can be used to minimize the effects of overfitting by checking the validation set error after each epoch. If the validation set error increases over a number of epochs then training is terminated.

5.2.5 Data Division for NN Training

Preparing the data set for training a neural network generally requires the training data to be split into three sets, namely the training set, validation set, and test set. The training set is used for computing the error gradient for each epoch and updating the weights accordingly. The validation set error is monitored during training to detect overfitting. The error on the training and validation set typically decrease at the beginning of training although when the network starts overfitting to the training data, the validation error typically begins to increase or stay constant. The error of the test set is not used during training but is often plotted during training and can give an indication of a poor division of the training data into the three sets. The test set error is also used as an indication of the fit model's ability to generalize (Engelbrecht, 2007) and to compare different models (Beale *et al.*, 2014).

5.2.6 Evaluating NN Performance

Evaluation measures are used to determine whether a NN can fit a model to a training data set that can be generalized to examples that are not used in the training set.

Two commonly used measures for NN performance are measuring the error on training, validation, and test sets as well as evaluating regression plots (Beale *et al.*, 2014). Over the course of training, the network performance in terms of the error on the training, validation, and test sets can be plotted for the epochs.

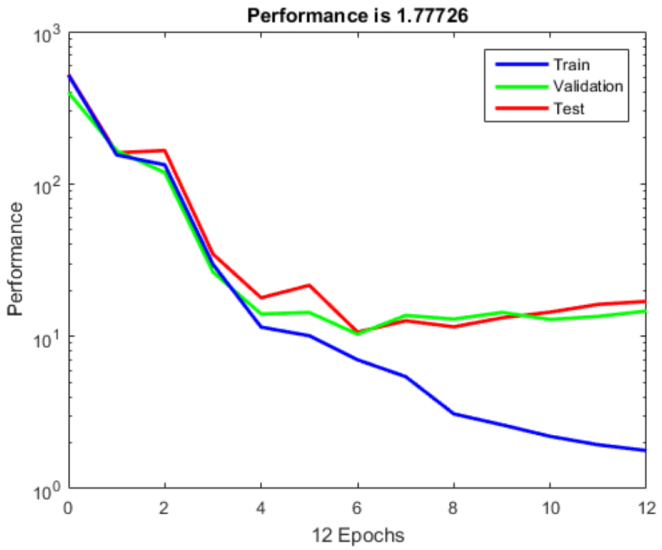


Figure 5.8: Performance plot showing the MSE plotted for twelve epochs. (Beale *et al.*, 2014).

Figure 5.8 shows an example performance plot for NN training for which the performance is given by the MSE for the train, validation, and test sets. The similarity of the validation and test performance curves indicate that there are no major problems with the NN training.

Another evaluation measure that is used in conjunction with the performance plot is the degree of linearity between the predicted output values and their respective targets (Beale *et al.*, 2014). The predicted outputs can be plotted against the target outputs in a regression plot. The regression plot allows outliers to easily be detected. The correlation coefficient can be computed from the targets and predicted output to determine how well the predictions match the targets.

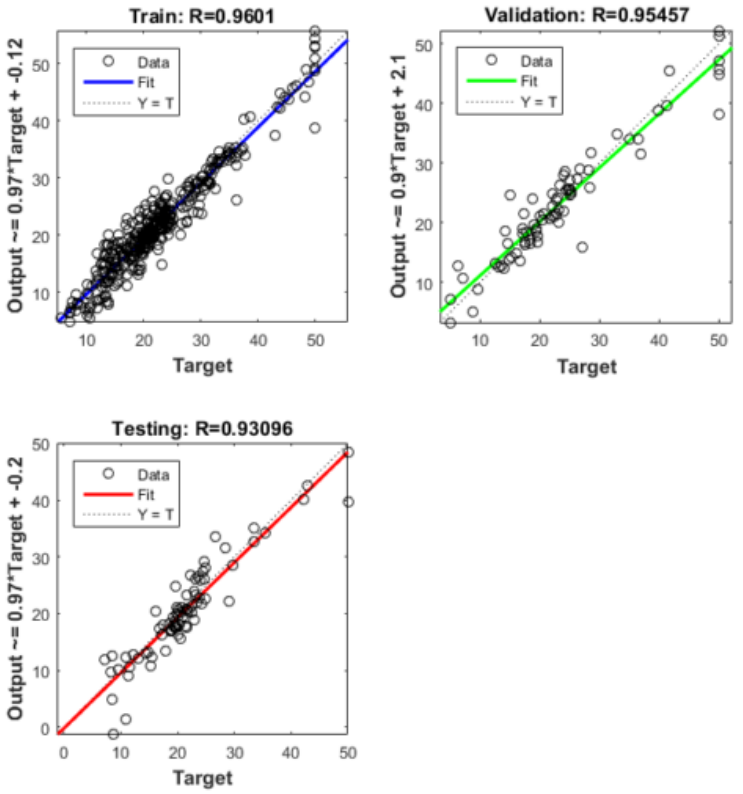


Figure 5.9: Example regression plots for training, validation, and test targets vs predictions. (Beale *et al.*, 2014).

Figure 5.9 shows example regression plots for targets vs predicted outputs. The correlation coefficient (R value shown above the plots) is a measure of linearity between the target outputs and the predicted outputs. The correlation coefficient measures how well the model fits the data. An R value of one indicates a perfect linear relationship, zero indicates that there is no linear relationship, and negative one indicates an inverse linear relationship.

5.3 Conclusions

This chapter discussed machine learning for tyre categorization and visible tyre count estimation based on the machine learning techniques identified in the computer vision chapter (Chapter 3).

The categorization problem is approached as a multiclass classification problem (Section

5.1). To create feature descriptors that are vectors of equal dimension, a clustering step is performed followed by the quantization step to form histograms of cluster centre occurrences. The k-means algorithm is selected as the clustering algorithm due to the conceptual simplicity of the algorithm. K-d trees can be used for approximating the closest cluster centres for assigning feature vectors to their respective bins during histogram creation. Since k-d tree approximation is not as accurate as comparing the Euclidean distances of each feature vector with the cluster centres, a priority search is used to improve the approximation of the assignment of feature vectors. Through using a k-d tree based search, the time taken for assigning descriptors to cluster centres is reduced. To improve the accuracy of the assignments, a priority search is used. The classifier chosen to train and perform the classifications is the SVM. It was found, in the literature, that for multiclass classification using the BoVW model, SVMs result in superior classifiers over their counterparts. The ECOC binary to multiclass extension of binary SVMs provides a more resistant multiclass classifier extension than the OVA and AVA extensions, therefore the ECOC extension for SVMs is used in this study. K-d trees are used for finding the approximate closest cluster centres formed by the k-means clustering. Although k-d trees only provide approximations, they require fewer comparisons than a linear search. To measure the performance of the trained classifier, values are calculated from a confusion matrix and the values are used to determine the overall performance of the trained classifiers.

The artificial neural network, discussed in Section 5.2, is selected for the visible tyre count estimation experiments due to its effectiveness in creating non-linear mappings from vector input to numerical output. ANNs have been used for counting by regression in a number of domains, particularly for numerical predictions, and object counting tasks in computer vision. The sigmoid function is used as the activation function as preliminary experiments indicated that ANNs trained using the Sigmoid activation function showed better results than ANNs trained using the hyperbolic tangent function as an activation function. The Generalized Delta backpropogation algorithm is selected as the training algorithm due to its popularity and its ability to achieve good results. Batch learning is selected over stochastic learning due to its robustness against inherently noisy data. To avoid bias in the resulting trained neural network, the training data is divided a number of times. Section 5.2.5 describes splitting a data set randomly three times into a training and test set. Three ANNs can then be trained and evaluated to ensure similar results are obtained on the test set for each of the trained ANNs. Splitting the training data more than once, ensures that bias that may be introduced as a result of a

single random partitioning of the training data is minimized. Metrics for performance (Section 5.2.6) include descriptive stats on error as well as MSE and correlation coefficient. The descriptive stats and MSE are used as measures of accuracy and the difference of the correlation coefficients for predictions on the training set and test set are used to determine the trained ANNs generalization ability.

Chapter 6

Experimental Design

Chapters 1 to 5 aimed to meet the first research objective RO_1 by identifying suitable categorization and count estimation techniques for the domain of tyres and tyre stockpile images. The remaining chapters address research objective RO_2, which is comprised of research questions RQ_5 and RQ_6. The specific focus of this chapter is:

RQ_5 *How can experiments be designed to determine the appropriateness of the identified categorization and count estimation methods?*

This chapter describes the experimental design and the approach taken for both the individual tyre categorization (Section 6.1) and the visible tyre count estimation from tyre stockpile images (Section 6.2). The experimental design is discussed in terms of its three main elements, namely:

- Experimental procedure,
- Evaluation methods, and
- Implementation tools.

Through answering research question RQ_5, experiments are designed that apply the identified computer vision and machine learning algorithms for the tasks of tyre categorization and visible tyre count estimation. In Chapter 7 the results obtained from applying the identified techniques, in the designed experiments, are presented and discussed.

6.1 Individual Tyre Categorization

In this study, the BoVW framework is used for the categorization of individual tyres. The experiments consist of two separate data sets. The first data set consists of tyre

tread images that are used for categorization at the specific instance level. The second data set is used for the categorization of tyres into more general categories, namely 4x4, passenger, and truck tyres. All categorizations are done based on images of the tyre tread.

In the remainder of this section the experimental procedure for the categorization experiments are discussed in terms of data collection (Section 6.1.1), data pre-processing (Section 6.1.2), and parameter selection and model fitting (Section 6.1.3). After fitting models to the training data using the BoVW model for categorization, the trained classifiers must be evaluated. The evaluation process used to determine the performance of tyre categorization using the BoVW model is explained (Section 6.1.4). Finally, the implementation tools used for the tyre categorization experiments are given (Section 6.1.5).

The experimental procedure was conducted in two stages. One for the categorization of tyres at the specific category level and another for the generic category level. Both stages were investigated using the same experimental setup using the BoVW model. The experimental procedure consisted of three distinct stages:

1. Data collection,
2. Data pre-processing, and
3. Parameter Selection and Model fitting.

The data collection (Section 6.1.1) and data pre-processing (Section 6.1.2) stage describes how tyre tread image data was acquired and which pre-processing algorithms were applied to prepare the images for feature extraction. The parameter selection and model fitting stage for the BoVW model in the context of tyre tread are discussed in terms of the visual vocabulary formation and the chosen multiclass classifier extension (Section 6.1.3). The performance measures used to evaluate the multiclass classifier for categorization are discussed (Section 6.1.4). Finally, an overview of the implementation tools used to implement the experiments is given (Section 6.1.5).

6.1.1 Training Data Collection

For tyre tread categorization at the specific level, images were collected by taking photos of individual tyres at a local tyre retailer. The tread portion of each image was then cropped to provide an image consisting only of the tread. Specific tyres were photographed from slightly different angles. The images of each tyre tread were then manually sorted into their respective categories by placing them into a class per tyre.

Tyre Instance	Number of Images
Goodyear	6
Maxtrek	8
Continental	18
Hankook	15
Michelin	19
Michelin2	20
Pirelli	17
Dunlop	8
Vokohoma	14

Table 6.1: Number of specific tyre instance images in the specific tyre dataset.

Table 6.1 shows the labels assigned to each category of tyre at the specific level. The number of tyres in each category is also shown. A total of 125 tyre images were used to form the data set used for the specific level categorization experiments. A total of 125 tyre images representing 9 different tyre instances were used to form the data set used for the specific level categorization experiments.

The collection of tyre tread images for the purpose of categorization at the more general level was performed by cropping tyre tread portions from images of stacked tyres at REDISAs Port Elizabeth tyre depot (REDISA, 2016). The cropped tyre tread images were then manually grouped according to 4x4, passenger, and truck tyres. The image data set for tyre categorization at the general level consisted of 20 images per category. A total of 60 images were used to form the data set used for general level categorization experiments.

6.1.2 Data Pre-processing

The images in the data sets for both specific instance and more general categorization are preprocessed in the same way. A combination of colour transformation, histogram equalization, and median filter smoothing is used to prepare the images for extracting features.

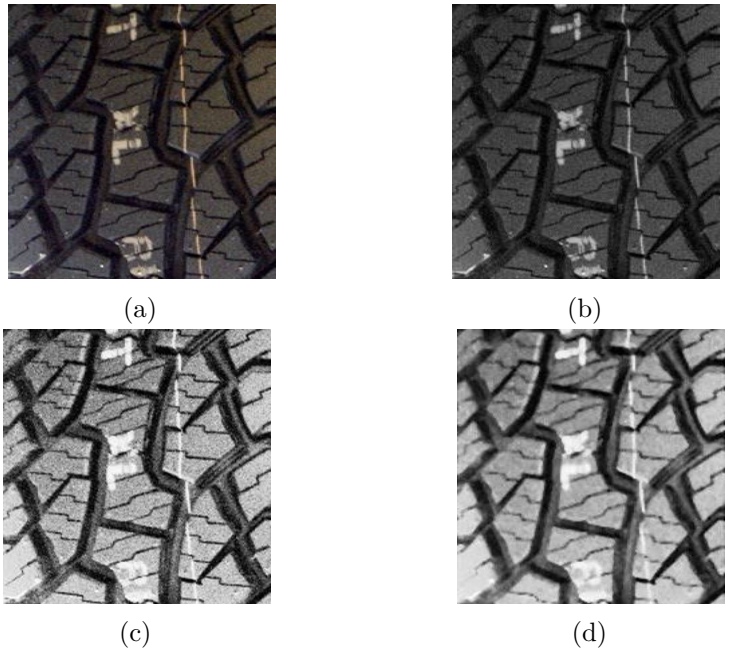


Figure 6.1: Preprocessing example applied to a section of tyre tread.

Figure 6.1 shows the result of the preprocessing of images to prepare them for extracting image features. The original image in Figure 6.1a is first transformed to a grayscale image that can be seen in Figure 6.1b. Histogram equalization is then applied to the grayscale image, which results in the image shown in Figure 6.1c, to improve the contrast in the tread and to normalize illumination levels. A small median filter is applied with a 3x3 neighbourhood to remove noise while maintaining sharp gradient changes in the tread grooves. The result of applying all of the pre-processing steps can be seen in Figure 6.1d. The preprocessing steps applied to the images were experimentally chosen as an increase in classification performance was observed when applying these preprocessing techniques in informal experiments.

6.1.3 Parameter Selection and Model Fitting

The BoVW is an established method for the classification of images. In the context of categorizing waste tyres, the three parameters of the BoVW that require investigation are the number of clusters used to create the bag of visual words, the feature detector, and the feature descriptor. In Section 3.8 it was noted that Csurka *et al.* (2004) found that an appropriate number of clusters, in their study, was $k = 1000$ in terms of the trade-off between time taken to cluster and overall accuracy. The feature detection and description

methods used to identify and describe image patches that were discussed in Section 3.5 and Section 3.6 were informally investigated in the context of tyre categorization based on tread patterns in Section 4.2.2. The results led to the hypothesis that for tyre categorization at the specific level. Points detected using the Harris corner detector and described using the SURF descriptor would result in good category separation based on the MSE ratio for within and between class feature matches. Although a hypothesis was made for the case of specific level tyre instance categorization, no hypothesis was made about which the feature detector-descriptor combination would be the most suitable for higher level categorization. To this end, experiments are conducted with each of the detector-descriptor combinations.

Once the features are described for each of the detector-descriptor combinations, each image is encoded to form its visual word frequency histogram using a nearest neighbour approximation using the nearest neighbour search strategy based on k-d trees developed by (Silpa-Anan & Hartley, 2008). The ECOC scheme for extending the binary SVM to a multiclass SVM was chosen to create a multiclass classifier. Each class in the generic level of categorization experiments was assigned a bit vector containing three bits. Each class in the specific level of categorization experiments was assigned a bit vector containing nine bits. The general level and specific level categorization experiments were considered separately. The number of bits in the bit vectors corresponds to the number of categories that the multiclass classifier must distinguish.

6.1.4 Evaluation methods

Each detector-descriptor combination is used to create a BoVW three times for cluster size $k = 1000$ (following Csurka *et al.* (2004)) resulting in three sets of results for each of the detector-descriptor combinations. Each time a BoVW is created, 70% of the available training data is selected using random subsampling in which the hold out method is repeated k times and the results are averaged as described in Kohavi (1995). The data used to create the BoVW is encoded using the created BoVW and forms the training set. The remaining 30% of the available training data is encoded using the BoVW and forms the data for the test set.

The random subsampling method is a variation of the holdout method. In the holdout method, data are randomly partitioned into two distinct sets, namely the training set and the test set. The training set is then used to derive the model and the test set is used to

obtain an accuracy for the derived model. In random subsampling, the holdout method is repeated k times. The accuracy estimate is then taken as the average of accuracies obtained in each of the k iterations (Han *et al.*, 2011).

The process is repeated three times using the random subsampling method. The average accuracy, precision, and recall over the three sets of results for each detector-descriptor combination are evaluated to determine how well the detector-descriptor combinations perform for tyre categorization in the BoVW model.

6.1.5 Implementation Tools for Categorization

The environment used for the categorization of tyre instances is Matlab R2014b. Three toolboxes from the suite of Matlab toolboxes were used, namely the Computer Vision System Toolbox, the Image Processing toolbox, and the Statistics and Machine Learning toolbox. The VLFeat library was used for feature extraction (Vedaldi & Fulkerson, 2010). For pre-processing images the Image Processing toolbox was used. Matlab's Computer Vision Toolbox was used for the detection of Harris, Fast-Hessian, and MSER features and VLFeat was used for the DoG and Harris-Laplace detectors. SURF descriptors were computed using the Computer Vision Toolbox implementation for SURF feature description and SIFT descriptors were computed using the VLFeat SIFT descriptor implementation. The VLFeat implementation of the k-means clustering algorithm was used to find cluster centres. The VLFeat implementation for constructing k-d trees was used to create the BoVW from cluster centres. Matlab's Statistics and Machine Learning Toolbox was used for the multiclass classifier extension of the SVM using ECOC.

6.2 Visible Tyre Count Estimation

The experimental design for tyre count estimation includes similar high level stages to the tyre categorization. The three distinct stages are again:

1. Training data collection and preprocessing,
2. Parameter Selection and Model fitting, and
3. Model evaluation.

The focus of the experiments regarding visible tyre count estimations is to determine how well the feature descriptors, proposed in Section 4.3.1, work when used as input to NNs to predict the number of visible tyres in tyre stockpiles.

6.2.1 Data collection

Data collection for the experiments was conducted through a field trip to a local REDISA waste tyre depot. Three piles of tyres were created, each consisting of different numbers of tyres and different types of tyres. The three piles were photographed from various angles to provide image of waste tyre stockpiles that differ from each other in appearance. Each photograph was taken from approximately three meters away from the nearest tyre in the pile although the space limitation in the area of the stockpiles required some photos to be taken closer or further away from the stockpiles. The ground truth in terms of the number of visible waste tyres in each image was obtained by manually counting the number of visible tyres in the tyre stockpile images. The real world stockpile image set consisted of 185 images with dimensions 720x640. The number of visible tyres in the images were in the range [17,63].

(a) (b) (c)

Figure 6.2: Examples of the three stockpiles

The images in Figure 6.2 show examples of the three individual stockpiles at a REDISA tyre depot. The three images are representative of stockpiles contains different tyre types. The stockpile in Figure 6.2a contains only passenger tyres. The stockpile shown in Figure 6.2b contains a mixture of passenger and truck tyres. The stockpile shown in Figure 6.2c contains truck tyres. It can be seen in Figure 6.2a and Figure 6.2b that the images contain tyres in the background other than the tyres in the pile of focus. The preliminary experiments with segmentation indicated that is was difficult to automatically segment tyre piles from the background. Another limitation was that many of the images contained stockpiles or stacked tyres other than the tyre pile that was of focus. The stockpiles and tyre stacks in the background were a result of the limited space available for creating tyre piles in the designated depot area. As an initial pre-processing step the tyre stockpiles of focus were manually segmented.

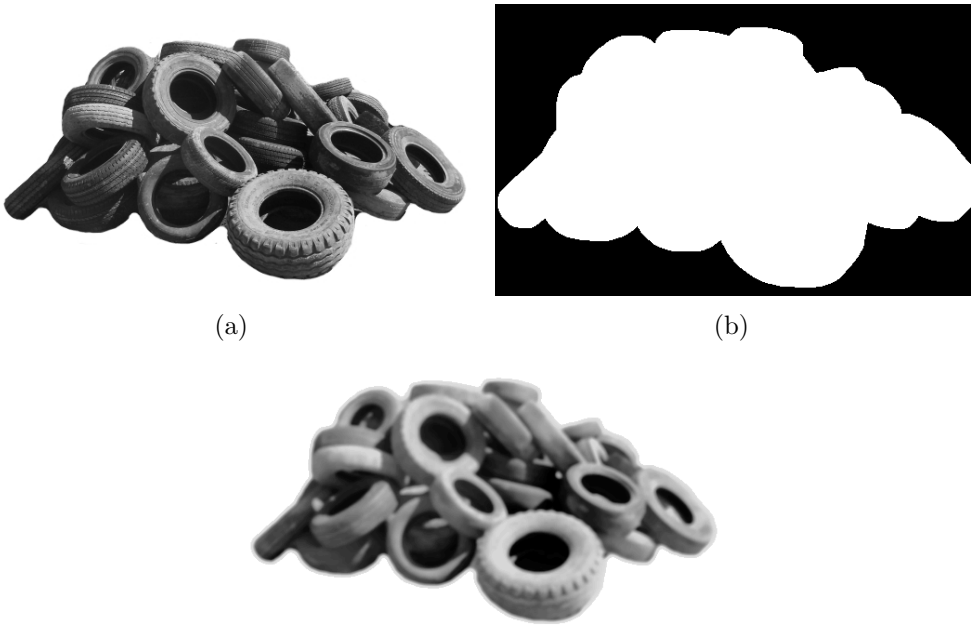
In addition to evaluating the proposed feature descriptions with NNs on the real world stockpile images, the methods were also tested using a data set of images containing various numbers of circles and a data set containing images of 3D generated stockpiles. The circle images data set is used to determine how well the methods work in the context of simple objects with minimal variation in appearance. 1000 images containing circles were generated with dimensions 766x766 with the number of circles for the images in the range [0,40]. The 3D stockpile images were generated to provide a larger data set with more stockpile appearance variation in comparison to the real world stockpile images data set. A total of 650 images were generated with dimensions 960x540 with the number of visible tyres in images in the range [2,48].

The area of the images that actually contain tyre stockpiles depends on the number of tyres in the stockpile and the distance of the stockpile from the camera. When photographing real-world stockpiles, the distance from the nearest tyre was chosen to be roughly three meters as it provided sufficient distance for the image to contain the entire stockpile while still capturing sufficient detail for visible tyres to be manually counted in the image. In generating stockpile images, the distance was also kept fairly constant while the number of tyres in the computer generated piles was varied. The size and volume of the real-world stockpiles were constrained due to the amount of space that was available for creating randomly arranged piles of tyres. The images containing circles and the generated images were produced so that the range of visible objects in the images would be similar to the range of visible tyres in the real-world stockpile images. The remainder of this discussion refers to stockpile images although the same process is applied for all data sets used for evaluation.

6.2.2 Data pre-processing

All stockpile photographs for evaluation were manually segmented to create images that isolated the foreground pile. For the experiments using HOG features to summarize global stockpile structure and using other histograms to summarize stockpile structure (Section 4.3.1), the same pre-processing procedures are applied. After manually segmenting stockpiles in the image, a mask is created in order to apply pre-processing only to images within the region covered by the mask. The reason for using a mask in the context of the pre-processing is for histogram equalization. If all pixels were included in the histogram equalization, including the pixels from the background that had been segmented, the background pixels that do not contain any information would be included in the histogram

equalization of the entire image as opposed to only the pixels representing the stockpile in the image.



(a) (b)

(c)

Figure 6.3: Results of pre-processing steps. (a) conversion to grayscale image. (b) Mask created from pre-segmented image. (c) Result of applying all pre-processing steps to image.

Figure 6.3 shows the results of pre-processing steps for a single example. The same pre-processing steps are applied to the images for all experiments. Figure 6.3a shows the grayscale version of the original image after manual segmentation. The mask shown in Figure 6.3b is created by applying a threshold to the image, inverting the result, and performing a disk shape dilation with 10x10 neighbourhood on the threshold result. The dilation on the mask is performed to increase its size so that boundary pixels are included in further computations where only pixels within the mask are considered. Figure 6.3c shows the result of first applying a median filter with a neighbourhood size of 5x5. Then a Gaussian filter is applied with a neighbourhood size 5x5 and $\sigma = 1.8$. After the smoothing, the pixels within the mask region are histogram equalized. The median filter is applied to remove noise resulting from the environment and equipment used while preserving sharp intensity changes. The Gaussian filter is applied in an attempt to suppress tread structures. The histogram equalization is applied to enhance the contrast in the image.

Once the images have been pre-processed, the feature vectors are created. Once feature vectors for the images have been created, they are required to be scaled. The activation functions, for example the sigmoid activation function, used by artificial neurons typically output values in a specified range. The functions become essentially saturated when the values are large (Beale *et al.*, 2014). It is thus standard practice to normalize the inputs and the targets to the NN. All of the training data feature vector entries and the target outputs are normalized to lie in the range [-1,1] using vector normalization. To acquire scalar count estimates, the predicted output values are mapped back to the original range using the reverse of the process used to normalize the target values. Once the reverse of the normalization process has been applied to the predictions, each prediction is then rounded to the nearest integer.

6.2.3 Parameter Selection and Model Fitting

The parameter selection process for fitting a model for visible tyre count estimation requires selection of parameters for feature extraction processes as well as a selection of parameters for the NN. The parameter selection for using HOG features in the BoVW model to describe the global stockpile structure include cell size, grid step, and the number of clusters. The parameters that need to be selected for describing stockpile images using other histograms include the selection of appropriate features from which to form the histograms as well as the number of bins used for each feature histogram.

For training the neural network, the free parameters that need to be selected are the network architecture in terms of the number of hidden layers, the number of neurons within each of the hidden layers, and the learning rate η. To determine the best selection of the learning rate η, NNs using various learning rates were informally tested. A learning rate of $\eta = 0.0001$ was found to provide a good trade-off between training time and convergence.

The hidden layer neurons are summation units with a sigmoid activation function. The output layer consists of a single summation unit with a linear activation function since numerical output is required. Experiments are conducted with NNs consisting of two hidden layers. The results of training NNs with three different structures are collected. The NN structures are 5 neurons in each hidden layer, 10 neurons in each hidden layer, and 20 neurons in each hidden layer.

The termination criteria that is set for the NN so that training does not continue indefinitely are:

1. *Epoch limit*: The maximum number of epochs is set to 100 000.
2. *Training error threshold*: The training error threshold is set to zero so the NN training will terminate if there is no error on the training set.

Training will terminate if either one of the termination criteria are met. The epoch limit is set so that training does not continue indefinitely. The training error threshold is set to zero so that the
For each considered NN structure, k-fold cross validation is used to evaluate the performance of the NN structure with $k = 3$. 3-fold cross validation splits the available training data is partitioned into 3 data sets D_1, D_2, D_3. Three NNs are trained for each NN structure. For each of the three NNs, one of the data partitions is used for testing while the remaining two partitions are used for testing.

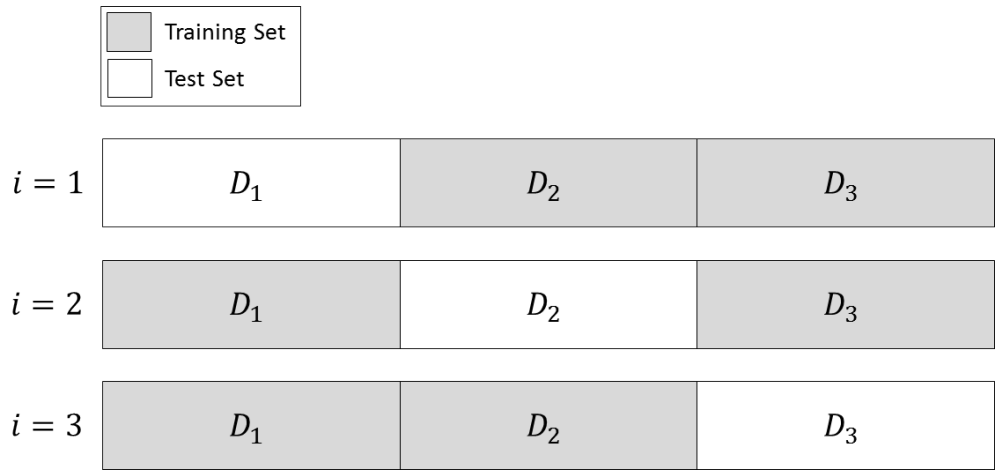


Figure 6.4: Illustration of data partitioning for 3-fold cross validation.

The methods are evaluated using 3-fold cross validation to reduces bias that may arise as a result of random selection of training data. By performing training and testing on the data three times with the different partitions of the data, each datum occurs in the test set during one of the k iterations.

6.2.4 Experimental procedure

The experimental procedure consists of first extracting the feature vectors that form global descriptions of the region of the image containing the stockpile. The result is a list of feature vectors. The list of feature vectors is then randomly partitioned into three subsets of equal size. The rest of the experimental procedure can be described as,

- For each considered NN structure
  - For $i = 1$ to 3
    - Randomly initialize network weights.
    - Test set $= D_i$
    - Training set $= D - D_i$
    - Train NN with current structure using training set.
    - Evaluate model on test data.
    - Collect results.
  - Average the results obtained over the three iterations for current NN structure.
  - Save results

The results obtained from training the model include the MSE, the error variance, and descriptive statistics on the error. The results are compared for each NN structure for both the feature extraction using HOG features for global stockpile description and other histograms for stockpile description.

6.2.5 Evaluation methods

To evaluate the feature descriptors for count estimation and NN structure combinations, a table is created of the summary statistics, MSE, error variance, and correlation coefficient for the testing and training set for each NN structure. A table is created for each of the k NNs trained on the $k - 1$ data partitions. The results for each of k trained classifers are averaged to give an average indication of the results obtained by each feature description method with an NN structure. The prediction error for NNs trained using the different feature descriptions are then compared.

6.2.6 Implementation Tools for Count Estimation

The experiments for tyre count estimations were also conducted using Matlab R2014b (MATLAB, 2014). The feature extraction for summarizing HOG features using BoVW

makes use of the Matlab's Computer Vision System toolbox for extracting the HOG features. The BoVW using the HOG features is created using VLFeat's k-means and k-d tree implementations as in the categorization of tyres experiments. For image preprocessing and constructing other feature histograms, Matlab's image processing toolbox is used. Matlab's Neural Network toolbox is used to create and train neural networks.

6.3 Conclusions

The experimental design is split into one set of experiments for the categorization of individual tyres (Section 6.1) and a second set of experiments for estimating the count of visible tyres in tyre stockpile images (Section 6.2).

Tyre categorization experiments are conducted using the BoVW model. Two separate data sets are created to evaluate the use of the BoVW model for categorizing tyres at a general level and at a specific level (Section 6.1.1). Since there is no standard data set for evaluating tyre categorization, tyre images were collected from a local retailer and local tyre depot. Each tyre image was cropped to contain only the tread portion of the tyre so that categorization based on only the tread portion of the images can be evaluated. A combination of preprocessing methods have been selected to enhance image features and suppress noise in the images (Section 6.1.1). The number of clusters, and therefore the dimensionality of the resulting feature descriptors, used for the tyre categorization in the BoVW model is $k = 1000$. The number of clusters used is based on the original BoVW experiments. The categorization of tyres at the two levels of categorization is evaluated with all detector-descriptor combinations reviewed in Section 4.2.1. The data sets are split three times using random subsampling. The split is done to reduce bias that may result from a single partitioning of the data into a training and test set. An SVM is trained on each of the training data sets and tested using the corresponding test set (Section 6.1.3). The average accuracy, precision, and recall for the three SVMs are evaluated over the three sets and are used to determine the performance of the classifiers created using the various detector-descriptor combinations (Section 6.1.4).

The visible tyre count estimation experiments are conducted using feature extraction and description methods that were proposed specifically for the purpose of tyre count estimation. To build a model for mapping the feature descriptors to numerical estimations, neural networks are trained and evaluated. The two proposed feature extraction methods are tested separately to determine their suitability as input to a neural network for the

purpose of tyre count estimation. Three data sets are used to determine the performance of the resulting neural network (Section 6.2.1). The data sets range from the simplest case in which the images consist of only circles, to 3d generated stockpile images that do not contain noise and lighting variations, to the most difficult case of real world stockpiles that contain noise resulting from a range of environmental variables. The three data sets are used to investigate how well the count estimation approach with images of scenes of different complexity. To prepare the real world stockpile images, the tyre stockpiles are manually segmented out from the background since the automatic segmentation is considered to be outside the scope of this study. The images are then preprocessed to eliminate small structures and enhance tyre boundary structures (Section 6.2.2). A learning rate for training the neural network of $n = 0.0001$ was found to provide a good trade off between time to convergence and accuracy. Two hidden layer neural networks are considered with different layer sizes since a single layer ANN was found to perform poorly during preliminary experiments. Three different size hidden layers are considered during the experiments. An epoch limit of 100 000 and training error threshold of 0 are used for the termination criteria (Section 6.2.3). K-fold cross validation is used to minimize the bias that may be introduced as a result of a single partitioning of the training data (Section 6.2.4). The performance of the NN is evaluated using summary statistics on the error as well as the MSE, error variance, and correlation coefficient. The implementation of the visible tyre count estimation experiments is done using MATLAB, MATLAB toolboxes, and the VLFeat library (Section 6.2.6).

Chapter 7 presents the results of the experiments for tyre categorization and visible tyre count estimation. The various detector-descriptor combinations are compared for the general and specific levels of categorization that are considered. The count estimation results are followed by a discussion comparing the accuracy of ANNs trained using the two proposed feature description methods.

Chapter 7

Experimental Results

The previous chapter outlined the experimental designs for tyre categorization and visible tyre count estimation. The aim of the tyre categorization experiments is to determine the level of classification accuracy that can be achieved through using selected local features (Section 4.2.2) in a BoVW model (Section 3.8). The aim of the count estimation experiments is to provide an indication as to the level of accuracy of visible tyre count estimation that can be achieved by using the global feature descriptions (Section 4.3.1) with NNs (Section 5.2).

This chapter presents and discusses the results of the experiments conducted to determine how well each of the approaches for categorization and count estimation performed, thus answering research question RQ_6.

RQ_6 How well do the identified methods work for tyre categorization and visible tyre count estimation?

The discussion is separated into a discussion for the categorization results (Section 7.1) and a section for the count estimation results (Section 7.2). The results for categorization at both the specific level (Section 7.1.1) of categorization and the general level (Section 7.1.2) of categorization are presented and discussed. The results for the count estimation experiments using the grid of HOG features in a BoVW model (Section 7.2.1) and the custom histograms (Section 7.2.2) with NNs for function approximation are presented and followed by a discussion of the results.

7.1 Categorization Experimental Results

The experimental design for the categorization experiments (Section 6.1) discussed the experimental design in terms of the parameter selection, model fitting, and evaluation methods for using the BoVW approach for categorization.

The remainder of this section discusses the results of the experiments conducted using each of the identified detector-descriptor combinations considered for the categorization task with a visual vocabulary of $k = 1000$. The results for categorization of tyres at the specific level are reviewed in Section 7.1.1. The results for categorization for the general case are reviewed in Section 7.1.2.

7.1.1 Identification of Specific Tread Instances

For the specific case of tyre categorization it was hypothesized in Section 4.2.2 that the Harris feature detector with the SURF descriptor would create the best separation of classes based on the similarity of matched feature pairs within and between images within the category. Each of the detector-descriptor combinations were evaluated three times using a random sub-sampling method where each time the data set was split randomly into 70% of the data for creating the visual vocabulary and training the classifier and the remaining 30% of the data was used for testing the classifier. The following tables show the average results of the three classifiers trained using each of the detector-descriptor combinations in terms of their average accuracies, precisions, and recalls:

Feature Detector	**Average Accuracy**	**Average Precision**	**Average Recall**
Harris	0.74	0.65	0.74
DoG	0.72	0.64	0.72
Fast-Hessian	0.76	0.68	0.76
MSER	0.72	0.63	0.72
Harris-Laplace	0.77	0.68	0.77

Table 7.1: Average accuracy over 3-folds for detectors with SIFT descriptors (Specific categorization)

Table 7.1 shows the average accuracy over the three training iterations for each detector using the SIFT descriptor to describe the image regions around the detected local features. It can be seen in Table 7.1 that the Harris-Laplace detector has the greatest accuracy (0.77) if the regions around the detected local features are described using SIFT

features while the MSER and DoG detectors with the SIFT descriptor shows the lowest average accuracy (0.72). The Harris-Laplace and Fast-Hessian detectors results in the greatest average precision (0.68) and the Harris-Laplace detector with the SIFT descriptor showing the highest recall (0.77). The DoG and MSER detectors combined with the SIFT descriptor had the lowest average recall (0.72). All of the detectors combined with the SIFT descriptor displayed similar levels of accuracy for the given data set. There were no statistically significant differences between the accuracy (p-value = 0.764), the precision (p-value = 0.705), and recall (p-value = 0.766) over the five detectors when using SIFT descriptors for the categorization of tyres into specific level tyre categories.

Feature Detector	**Average Accuracy**	**Average Precision**	**Average Recall**
Harris	0.80	0.73	0.80
DoG	0.78	0.81	0.78
Fast-Hessian	0.76	0.74	0.76
MSER	0.80	0.75	0.80
Harris-Laplace	0.73	0.75	0.73

Table 7.2: Average accuracy over 3-folds for detectors with SURF descriptors (Specific categorization)

Table 7.2 shows the average accuracy over the three training iterations for each detector combined with SURF features to describe the image regions around the detected local features. It was hypothesized that the Harris detector combined with the SURF detector would yield the best categorization performance due to the similarity of matched features for the specific case. The results in Table 7.2 show that the Harris detector and the MSER detector achieved the same average accuracy (0.8) when the regions are described using SURF features while the Harris-Laplace detector with the SURF descriptor showed the lowest average accuracy (0.73). The DoG detector combined with the SURF descriptor results in the greatest precision (0.81) while the Harris detector combined with the SURF descriptor shows the lowest precision (0.73). The Harris detector and MSER detector have the highest recall (0.8) when combined with the SURF descriptor while the Harris-Laplace shows the lowest recall (0.73). There were no statistically significant differences between the accuracy (p-value = 0.747), the precision (p-value = 0.806), and the recall (p-value = 0.745) over the five detectors when using SURF descriptors for the categorization of tyres into specific level tyre categories.

The results of categorization for the specific case of tyre categorization indicate that

overall the Harris detector and the MSER detector combined with the SURF descriptor performed the best out of the considered detector-descriptor combinations in terms of average accuracy and average recall. The DoG detector combined with the SURF descriptor shows the highest average precision of all the detector-descriptor combinations.

7.1.2 Identification of 4x4, Passenger, and Truck Tread Instances

The differing tread structures for tyres within the same categories requires that a sufficient visual vocabulary be created that can be used to construct histograms of visual word occurrences that capture sufficient characteristics of each tyre category at the general level. The following two tables show the results of using the feature detector-descriptor combinations at the general level of categorization:

Feature Detector	**Average Accuracy**	**Average Precision**	**Average Recall**
Harris	0.81	0.77	0.77
DoG	0.80	0.81	0.80
Fast-Hessian	0.76	0.83	0.75
MSER	0.59	0.54	0.59
Harris-Laplace	0.81	0.84	0.81

Table 7.3: Average accuracy over 3-folds for detectors with SIFT descriptors (General categorization)

Table 7.3 shows the results for the considered detectors combined with the SIFT descriptor to describe regions of the detected local features. The results show that the SIFT descriptor yields the greatest average accuracy when used in combination with either the Harris or Harris-Laplace local feature detectors (0.81). The lowest average accuracy when using the SIFT descriptors occurs when local features are detected using the MSER detector (0.59). The highest average precision (0.84) and recall (0.81) for detectors combined with SIFT descriptors is for the Harris-Laplace detector while the MSER detected features resulted in the lowest average precision (0.54) and recall (0.59). Statistically significant differences were found between the accuracy (p-value = 0.02), the precision (p-value = 0.019), and the recall (p-value = 0.031) over the five detectors when using SIFT descriptors for the categorization of tyres into general level tyre categories. The Harris, DoG, and Harris-Laplace had greater accuracies than MSER (p-value = 0.022, 0.035, and 0.022 respectively).The DoG, Fast-Hessian, and Harris-Laplace had greater precisions than MSER (p-value = 0.034, 0.0235, and 0.018 respectively). The

DoG and Harris-Laplace had greater recalls than MSER (p-value = 0.041 and 0.025 respectively).

Feature Detector	Average Accuracy	Average Precision	Average Recall
Harris	0.87	0.89	0.87
DoG	0.76	0.83	0.76
Fast-Hessian	0.78	0.74	0.78
MSER	0.78	0.80	0.78
Harris-Laplace	0.78	0.85	0.78

Table 7.4: Average accuracy over 3-folds for detectors with SURF descriptors (General categorization)

Table 7.4 shows the results of using the various local feature detectors with the SURF descriptor. The Harris local feature detector combined with the SURF descriptor shows greater performance in comparison to the other local feature detectors in terms of average accuracy. The DoG detector shows the lowest average accuracy (0.76) when combined with the SURF descriptor while the Fast-Hessian, MSER, and Harris-Laplace detectors yielded similar results in terms of average accuracy (0.78) and recall (0.78). The Harris detector combined with the SURF descriptor shows the highest average accuracy (0.87), precision (0.89), and recall (0.87) of all the detectors combined with the SURF descriptor. There were no statistically significant differences between the accuracy (p-value = 0.603), the precision (p-value = 0.670), and the recall (p-value = 0.580) over the five detectors when using SURF descriptors for the categorization of tyres into specific level tyre categories

7.2 Count Estimation Experimental Results

In Section 6.2 the experimental design for the visible tyre count estimations was discussed in terms of the parameter selection, model fitting, and evaluation methods for two proposed global feature descriptions to use as inputs to NNs. The two descriptors used for describing the tyre piles are the grid of HOG descriptors in a BoVW approach for extracting feature descriptors and the creation of histograms based on the gradient magnitudes, gradient orientations, and the Canny edge pixel gradient orientations.

The remainder of this section presents the results obtained for visible object count estimation for three data sets. The first data set contains images with white circles

and is used to show the performance of the visible object count estimation methods for simple overlapping objects with circular shape. The second data set contains computer generated tyre piles to show the results of the performance of the visible object count estimation methods on a set of images of sufficient size that contains minimal noise. The third data set contains images of three distinct tyre stockpiles from different angles that were acquired from one of REDISA's tyre depots.

The results obtained are presented in terms of descriptive statistics and variance on the absolute error values as well as the MSE and correlation coefficient on the training sets and the testing sets. The mean is the average of the average absolute error for each of the 3 folds. The minimum column is calculated by averaging the minimum absolute error of each of the three folds. The maximum is calculated by averaging the maximum absolute error of the three folds. The MSE represents the average of the MSE values obtained for each of the three folds.

7.2.1 Grid of HOG Descriptor

This section presents the results of training neural networks with the HOG based visual word occurrence histograms (Section 4.3.2). The results for using the grid of HOG features in a BoVW model for NN training are presented for the three data sets discussed in Section 6.2.1. The first set of results was obtained using the data set of images containing circles. The second set of results was obtained using images of generated tyre stockpiles. The third set of results were obtained using images of real world stockpiles.

Results for Images Containing Circles

[5,5]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	1.14	0.00	6.33	1.00	1.00	1.22	2.43	0.99	0.99	0.99
Test	1.23	0.00	5.67	1.00	1.00	1.40	2.84	0.99	0.99	0.99
[10,10]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	1.11	0.00	5.33	1.00	1.00	1.18	2.35	0.99	0.99	0.99
Test	1.22	0.00	6.00	1.00	1.00	1.45	2.87	0.99	0.99	0.99
[20,20]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	1.06	0.00	6.33	1.00	1.00	1.13	2.16	0.99	0.99	0.99
Test	1.20	0.00	5.67	1.00	1.00	1.45	2.84	0.99	0.99	0.99

Table 7.5: Results of using neural network with HOG based visual word occurrences for three different neural network structures on data set of circle structure images.

Table 7.5 show that the mean error is low amongst all the considered NN structures. The lowest mean error is obtained for both the training (1.06) and testing (1.20) data sets when 20 neurons are used in each of the two hidden layers. The minimum, maximum, median, and mode of the absolute error amongst the three considered NN structures are similar. On average both the variance (1.45) and MSE (2.84) for two hidden layers of 20 neurons in each is the lowest amongst the considered NN structures. All considered structures achieved similar correlation coefficient (0.99) for ground truth and predicted values for each of the k iterations of the cross validation. The correlation coefficients indicate the the NN structures are all able to produce a model with a strong linear relationship between target and actual values that the trained NN is able to generalize well to previously unseen data in the context of simple images containing an unknown number of circles when using the HOG based image patch word occurrence histograms as input feature vectors.

Results for Images Containing Generated Stockpiles

[5,5]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	1.76	0.00	13.67	1.00	1.00	2.90	5.88	0.98	0.97	0.98
Test	2.28	0.00	22.33	1.33	1.33	8.83	14.00	0.95	0.97	0.94
[10,10]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	1.48	0.00	10.00	1.00	1.00	1.97	4.07	0.98	0.99	0.99
Test	2.48	0.00	23.33	2.00	0.67	8.76	14.81	0.95	0.96	0.94
[20,20]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	1.29	0.00	6.67	1.00	1.00	1.38	2.99	0.99	0.99	0.99
Test	2.41	0.00	25.33	1.33	1.00	9.09	14.76	0.95	0.95	0.95

Table 7.6: Results of using neural network with HOG based visual word occurrences for three different neural network structures on data set of generated tyre pile images.

Table 7.6 shows the results obtained for training and testing on 3D generated stockpile images. The results show that a NN with five neurons in each of the hidden layers has the lowest average error on the test set (2.28). The three considered structures all have a minimum error of zero while the maximum error is greatest (25.33) for the NN structure with 20 neurons in each hidden layer and lowest (22.33) for the NN structure with five neurons in each of the hidden layers. The median values are the same (1.33) for two layers of five and twenty neuron hidden layer structures and the median value is greatest (2) for the NN structure with ten neurons in each hidden layer. The variance of the error and MSE for the test sets is greater, for all considered NN structures, than for the circle images. The greatest variance (9.09) on the test set error for 20 neurons in the two hidden layers and the lowest (8.76) for 10 neurons in the two hidden layers. The correlation coefficients for each of the considered NN structures on the test sets indicate that there is a moderate to strong linear relationship between the ground truth and the predicted count values for each of the k iterations of the cross validation.

Results for Images Containing Real World Stockpiles

[5,5]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	1.33	0.00	6.67	1.00	0.67	1.82	3.47	0.99	0.99	0.99
Test	4.78	0.00	28.33	3.67	1.67	25.05	47.78	0.80	0.85	0.84
[10,10]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	0.54	0.00	3.00	0.17	0.00	0.54	0.77	1.00	1.00	1.00
Test	5.73	0.00	34.67	4.33	2.00	38.56	70.73	0.73	0.78	0.76
[20,20]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	0.20	0.00	2.33	0.00	0.00	0.22	0.23	1.00	1.00	1.00
Test	6.08	0.00	32.67	4.00	0.67	47.18	83.92	0.71	0.76	0.79

Table 7.7: Results of using neural network with HOG based visual word occurrences for three different neural network structures on data set of real world stockpile images.

Table 7.7 shows the results obtained for using real world images of tyre stockpiles. The lowest mean error (4.78) on the test set was achieved by the NN structure with five neurons in each of the two hidden layers while the greatest mean error (6.08) on the test set was obtained by the NN structure with 20 neurons in each of the two hidden layers. All of the NN structures obtained a minimum error of zero. The NN structure with five neurons in each layer obtained the lowest maximum (28.33) and median error (3.67) on the test set while the NN structure with ten neurons in each of the two hidden layers had the highest maximum (34.67) and median (4.33) error on the test set. The NN structure with 20 neurons in each of the two hidden layers had the lowest mode (0.67) error value on the testing data while the NN structure with ten hidden neurons in each of the hidden layers had the greatest mode (2.00) error on the test set. The errors for the neural network with five hidden neurons in each layer had the lowest variance (25.05) and MSE (47.78) while the neural network with twenty hidden neurons in each layer had the greatest variance (47.18) and MSE (83.92) on the test set. The correlation coefficients for all NN structures indicate that the targets and predictions have a moderate linear relationship. The results obtained from the NN structure with five neurons in each of the two hidden layers showed the greatest linear relationship between ground truth and predicted values while the NN structures with 10 and 20 hidden neurons in each layer showed similar linear relationships between the targets and predicted values over the k

iterations of the cross validation.

7.2.2 Custom Histogram Descriptor

This section presents the results of training neural networks with the custom histograms (Section 4.3.3) based on the gradient orientation, magnitude, and edge orientations. The results of using NNs for function approximation with custom histograms as input were obtained for each of the three data sets discussed in Section 6.2.1. The first set of results was obtained using the data set of images containing circles. The second set of results was obtained using images of generated tyre stockpiles. The third set of results were obtained using images of real world stockpiles.

Results for Images Containing Circles

[5,5]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	0.99	0.00	6.00	1.00	1.00	0.91	1.85	0.99	0.99	0.99
Test	1.07	0.00	6.67	1.00	1.00	1.10	2.17	0.99	0.99	0.99
[10,10]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	0.95	0.00	5.67	1.00	0.67	0.92	1.75	0.99	0.99	0.99
Test	1.05	0.00	6.00	1.00	1.00	1.12	2.13	0.99	0.99	0.99
[20,20]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	0.91	0.00	5.67	1.00	0.00	0.84	1.58	0.99	0.99	0.99
Test	1.03	0.00	5.33	1.00	1.00	1.08	2.03	0.99	0.99	0.99

Table 7.8: Results of using neural network with custom histograms for three different neural network structures on data set of circle structure images.

Table 7.8 shows the results from using a NN with custom histograms for count estimation on images of circles. The lowest mean error (1.03) on the test set was found for the NN structure with 20 hidden neurons in each hidden layer and the greatest mean error (1.07) on the test set was produced by a NN with five hidden neurons in each layer. All of the considered NN structures achieved a minimum error of zero on both the training and testing data. All of the considered NN structures showed a similar maximum error on the test sets although the NN structure containing 20 neurons in each hidden layer had the lowest maximum error (5.33). The lowest variance (1.08) on testing sets was

found to be be the NN structure with 20 hidden neurons in each hidden layer and a NN structure with 10 neurons in each hidden layer had the highest error variance (1.12) on the testing data. The lowest MSE (2.03) on the testing data was for a NN structure with twenty neurons in each hidden layer and a NN structure with five neurons in each hidden layer had the highest MSE (2.17). The k iterations for each considered NN structure all resulted in similar correlation coefficient values and indicate that for all considered structures there is a strong positive linear relationship between the target values and the predicted values.

Results for Images Containing Generated Stockpiles

[5,5]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	1.61	0.00	8.67	1.00	1.00	2.04	4.56	0.98	0.99	0.98
Test	2.10	0.00	13.33	1.67	1.00	4.58	8.96	0.97	0.96	0.98
[10,10]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	1.44	0.00	7.33	1.00	1.00	1.71	3.70	0.99	0.99	0.99
Test	1.98	0.00	12.67	1.33	1.00	3.95	7.73	0.98	0.97	0.97
[20,20]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	1.37	0.00	6.67	1.00	1.00	1.61	3.43	0.99	0.99	0.99
Test	2.06	0.00	11.33	1.67	1.00	4.30	8.56	0.97	0.96	0.98

Table 7.9: Results of using neural network with custom histograms for three different neural network structures on data set of generated stockpile images.

Table 7.9 shows the results from the experiments using computer generated images of tyre stockpiles described using custom histograms. The results show that the lowest mean error (1.98) on the test set was obtained for the NN structure with 10 neurons in each of the two hidden layers and the highest mean error (2.10) was obtained for a NN structure with five hidden neurons in each layer. All of the considered NN structures achieved a minimum error value of zero for both the test set and training set data. The lowest maximum error (11.33) resulted from a NN structure with 20 hidden neurons in each layer and greatest maximum error (13.33) was obtained for a NN structure with 5 neurons in each hidden layer. A NN structure with 10 hidden neurons in each hidden layer had the lowest median error (1.33) and the other two NN structures resulted in

the same median error (1.67) on the test set. All considered NN structures resulted in the same mode (1.00) for the prediction error on both the testing and training set data. The lowest error variance (3.95) for the test set was obtained for the NN structure with 10 neurons in each hidden layer and the greatest variance (4.58) was for NN structure with 5 neurons in each hidden layer. The MSE (7.73) for the NN structure containing ten neurons in each hidden layer was the lowest and the NN structure with 5 neurons in each of the hidden layers resulted in the greatest MSE (8.96). All of the correlation coefficients for each iteration k of the cross validation for all considered NN structures are similar and indicate a strong linear relationship between the target and predicted values.

Results for Images Containing Real World Stockpiles

[5,5]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	1.26	0.00	5.67	1.00	1.00	1.47	3.03	0.99	0.99	0.99
Test	3.67	0.00	21.00	2.67	1.33	14.19	27.40	0.91	0.92	0.89
[10,10]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	0.94	0.00	3.67	1.00	0.67	0.81	1.60	1.00	1.00	0.99
Test	3.78	0.00	25.67	2.67	1.67	18.29	32.51	0.87	0.90	0.92
[20,20]; Averages for 3 folds										
	Mean	**Min**	**Max**	**Median**	**Mode**	**Variance**	**MSE**	$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$
Train	0.46	0.00	3.33	0.00	0.00	0.49	0.62	1.00	1.00	1.00
Test	4.04	0.00	20.33	2.83	0.67	17.92	33.89	0.85	0.89	0.93

Table 7.10: Results of using neural network with custom histograms for three different neural network structures on data set of real world stockpile images.

Table 7.10 shows the results obtained using real world stockpile images. The lowest mean error (3.67) on the test set data was obtained when using a NN structure with 5 neurons in each of the two hidden layers and the greatest mean error (4.04) was obtained when using 20 neurons in each hidden layer. All of the NN structures achieved a minimum error of zero on both the testing and training set. The lowest maximum error (20.33) on the test set was found for a NN structure of 20 hidden neurons in each hidden layer and the greatest maximum error (25.67) for a NN structure containing 10 neurons in each hidden layer. NN structures with five neurons in both layers and ten neurons in both layers had similar median values (2.67) while a NN structure with 20 neurons in

each hidden layer had the greatest median value (2.83). The lowest mode (0.67) for the error was for a NN structure with 20 neurons in each hidden layer and the greatest mode (1.67) for the error was for a NN structure of 10 neurons in each hidden layer. A NN structure of 5 neurons in each hidden layer had the lowest error variance (14.19) while a NN structure with 10 neurons in each hidden layer resulted in the greatest error variance (18.29). A NN structure with 5 neurons in each hidden layer resulted in the greatest correlation coefficient (0.91) and the lowest correlation coefficient (0.89) for a NN structure of 20 neurons in each hidden layer. The correlation coefficients for the test set for each of the k iterations of the cross validation for all considered NN structures indicate that there is a moderate linear relationship between the target values and the predicted values for all considered NN structures.

7.2.3 Comparison of Results For the Two Feature Extraction Methods

Based on the descriptive statistics with the error variance, MSE, and correlation coefficients it can be seen that using histograms consisting of gradient orientation, gradient magnitude, and edge orientation data provide superior count estimations in comparison to the HOG features visual word occurrence histograms regardless of the NN structure. The results presented indicate that using the histograms of gradient orientations, gradient magnitudes, and edge orientations result in a better generalization ability to unseen images because of lower variances (p-value = 0.011), lower MSE values (p-value = 0.015), and larger correlation coefficients (p-value = 0.0135).

7.3 Conclusions

This chapter presented the results of the experiments conducted for tyre categorization (Section 7.1) and visible tyre count estimation (Section 7.2). The categorization results were evaluated for various local feature detector-descriptor combinations in a BoVW model for categorization and their performance measured using average accuracy, precision, and recall. The Harris and Harris-Laplace corner based local feature detectors combined with the SURF descriptor showed better performance than other local detector-descriptor combinations for categorization of tyres at both the specific instance level (Section 7.1.1) of categorization and the general level (Section 7.1.2) of categorization.

The results for tyre categorization at the specific level and general level show that for both cases using SURF descriptors have a superior performance to that of SIFT descriptors

regardless of the detector that is used. The Harris detector combined with the SURF descriptor seems to outperform the other detector-descriptor combinations in the context of tyre categorization. One possible reason for the average accuracy of the Harris detector resulting in superior results could be attributed to it being a corner-based detector. The regions surrounding corner-like structures, found by the Harris detector, exhibit the greatest and most consistent image intensity data across tyre treads within the same category.

It can be seen from the results that the average accuracy when using SURF descriptors versus the average accuracy when using SIFT descriptors indicates that the SURF descriptors provide superior descriptions, in the context of tyre categorization from images of tyre treads, when used in the BoVW model. The better results achieved by the SURF descriptor in comparison to the SIFT descriptor indicates that the second derivative Gaussian box-filter approximation responses, in SURF descriptors, for the local feature regions provide a better summary of visual appearance to that of the histogram of gradient orientations used by the SIFT descriptor for describing tyre tread images for categorization.

The count estimation experiments (Section 7.2) indicate that using histograms of the gradient orientation, gradient magnitude, and edge orientations results in trained NNs (Section 7.2.1) obtaining lower prediction error on all of the considered data sets in comparison to trained NNs using grid of HOG descriptors in a BoVW model (Section 7.2.2) for global stockpile feature description. In addition to the lower prediction errors, the MSE, error variance, and correlation coefficients differences between the training and testing data sets indicate that using histograms of the gradient orientation, gradient magnitude, and edge orientations results in superior generalization ability over describing the images using grid of HOG descriptors in a BoVW model for global stockpile feature description.

The grid of HOG descriptors in the BoVW model for use in function approximation work well given that the mode of prediction error is small across all considered data sets. Although the mode of the predictions errors is small, the variance and MSE indicate that the feature extraction process combined with NNs for function approximation achieves good results for images containing objects that have a similar appearance in the image (circle images) while the generated tyre piles result in a higher variance on the errors and the real world images show the worst performance in terms of variance and MSE of the

prediction errors. The increase in error for generated stockpiles could be caused by the large number of views and visual appearances of the tyres in comparison to simple circles. The greater prediction error for real world stockpile images could be due to factors such as the differing angles and appearances of individual tyres as well as dirt on the tyres which would affect the gradient orientation information.

The custom histograms based on gradient orientation, gradient magnitude, and edge orientation work well as feature descriptors for function approximation with NNs considering the median and mode error are between 1 and 3. The simple circles data set achieved the best results with a maximum error being fairly low and the correlation coefficients for the testing and training data indicating that the trained NNs are able to generalize well to unseen data. The generated stockpile images indicate that the feature extraction technique works fairly well given that the median and mode error is around one on average and the variance on average is less than five. The correlation coefficients for testing and training data sets for the generated stockpile images also indicate that the trained NNs are able to generalize fairly well although not as well as in the case of simple circle count estimation. The results obtained when using real world stockpile images also indicate that using NNs for function approximation based on the proposed image feature descriptions work fairly well given that the median and mode error is between one and three although the error variance is greater than for circles and generated stockpile image data sets. The correlation coefficients obtained when using real world stockpile images indicate that the function approximated by the NNs has the ability to generalize to unseen data although not as well as for circle images and generated tyre stockpile images.

Chapter 8

Conclusions

This study investigated the possibility of using computer vision and machine learning algorithms for categorizing tyres based on tread images and estimating visible tyre counts in tyre stockpile images. For categorizing tyres at two different levels, namely specific and general levels, the BoVW model was evaluated using various local feature detector and descriptor combinations. Estimating the count of visible tyres in tyre stockpile images was conducted through extracting global image characteristics and training NNs based on the extracted feature descriptors. Digital image processing methods were discussed for use in noise suppression, image enhancement, feature enhancement, and data reduction for pre-processing images (Chapter 2). A broad literature review was conducted that investigated topics in computer vision for categorization approaches and counting approaches to determine what methods were available for the categorization and visible object count estimation in images (Chapter 3). Approaches for categorization and visible object count estimation approaches were discussed by relating the disadvantages of approaches to the specific domain of tyres and tyre stockpiles (Chapter 4). To build models using the identified computer vision concepts, learning structures for image categorization and count estimation were identified and described (Chapter 5). Experiments were designed to determine the suitability of selected algorithms in terms of feature extraction, machine learning algorithms, and evaluation measures for categorizing tyres based on tread images and visible tyre count estimation in tyre stockpiles (Chapter 6). The experimental results for tyre categorization using various local feature detector-descriptor combinations and visible tyre count estimation using two global feature description methods were presented and discussed (Chapter 7).

The research questions and research objectives that formed the basis of this research

are considered (Section 8.1), followed by a summary of the theoretical and practical contributions of this research to the existing knowledge base (Section 8.2). Although some of the results appear promising, there were a number of limitations influencing the generalization ability of the investigated solutions (Section 8.3). Given the findings and limitations of this research, recommendations for future research are presented (Section 8.4). The final section presents a final conclusion with closing remarks for this research study (Section 8.5).

8.1 Overview of Results and Outcomes of Research Objectives

At the beginning of this study the main research question (Section 1.3) was broken down into a set of sub research questions (Section 1.3.2) to guide this research. The research questions were addressed in subsequent chapters to ensure that the research objectives (Section 1.3.1) were met. The research questions and objectives together aimed to address the thesis statement (Section 1.2):

Computer vision techniques can be used to categorize individual tyres and obtain estimations of the number of visible tyres in tyre stockpile images.

To prove the hypothesis specified by the thesis statement, a main research question was formulated (Section 1.3) that needed to be answered. Two research objectives were specified that needed to be met through answering a number of sub research questions. In answering the first four sub research questions, the first research objective RO_1 was met (Section 8.1.1). The last two sub research questions were answered to meet the second research objective RO_2 (Section 8.1.2).

8.1.1 Research Objective One: Identification of Categorization and Count Estimation Methods

In order to meet the first research objective RO_1, research questions RQ_1 to RQ_4 required answering. This section discusses the outcomes of answering the sub research questions that are related to the first research objective.

Research Question One: What pre-processing methods are available for preparing images for categorization and count estimation?

To answer the first research question, a background of digital image processing had to be given to determine what image processing techniques were available for pre-processing images to prepare them for subsequent use in computer vision algorithms. Research question one was answered through a review of pre-processing methods that are frequently found in computer vision literature for preparing images.

Colour conversions were reviewed for conversions from colour images to gray and binary images. Colour conversions in this context are used to transform images that can efficiently represent images in which irrelevant information is suppressed (Section 2.1). A histogram equalization method was discussed that forces pixels in the image to have a more uniform distribution, thus enhancing image features (Section 2.2). Image filtering was reviewed for the purpose of smoothing images to remove noise using linear filters and non-linear filtering as well as enhancing sharp intensity changes through sharpening (Section 2.3). Image segmentation methods were reviewed for isolating image regions by grouping homogenous pixels based on some criteria (Section 2.4). Edge detection algorithms that are used for extracting edge pixels representing sharp intensity changes that often correspond to object boundaries were described (Section 2.5). Morphological image processing was discussed for its use in altering geometric structures in images in order to enhance particular image features (Section 2.6).

Chapter 2 provided a summary of digital image processing methods that are found in computer vision literature for pre-processing images prior to, or during, a feature extraction phase. The image processing methods discussed apply to both categorization and count estimation and thus the first research question was answered through the discussion.

Research Question Two: What approaches are available for object categorization and object counting from images?

The second research question was answered through a broad review of computer vision concepts and algorithms in Chapter 3. The broad review supported the identification of potential feature extraction methods, commonly used machine learning techniques, and strategies for using the methods and techniques to create models for the task of categorization and counting from images.

General vision was discussed in terms of top-down approaches, bottom-up approaches, and combinations of top-down and bottom-up approaches (Section 3.1). The discussion of top-down and bottom-up approaches introduced the general process for vision and provides a fundamental understanding of vision for the remainder of the discussion on computer vision concepts and algorithms. A brief introduction to image features as a means of describing relevant data from images from model creation were discussed (Section 3.2) to provide an understanding of what image features are and how they are used in the strategies for categorization (Section 3.3) and strategies for count estimation (Section 3.4) from images. A more in depth discussion was provided for local feature detectors (Section 3.5) for detecting localized image regions that can be described using local feature descriptors (Section 3.6). Global image features were discussed for their use in describing images or objects as a whole by some global characteristic or characteristics (Section 3.7). The discussion of local and global features revealed that local features are better suited for describing objects in images when there is occlusion and when viewpoint or object angles are expected to vary across images. In contrast to local features, global features are not well suited to cases where viewpoint changes and object angles are expected to change in the images and they do not provide sufficiently consistent descriptions when objects in images are highly occluded. The BoVW model was described for categorization as it is considered the standard model for visual object categorization (Section 3.8). Two state-of-the-art detection systems were reviewed with a view towards identifying their potential for use in counting by detection. The discussion revealed that there are inherent issues with global feature descriptors for describing objects where high degrees of occlusion and viewing angle are present (Section 3.9) and there are disadvantages associated with using non-maximal suppression of detection regions. Related counting systems that each follow a counting strategy of either counting by detection, regression, or segmentation were discussed to provide a background in terms of how they work and general advantages and disadvantages (Section 3.10) so that the concepts could be reviewed in the domain of tyre categorization and count estimation. Neural networks were identified as a machine learning method for non-linear function approximation for estimating the number of objects in images where object occlusions exist.

The broad review of computer vision concepts and algorithms in Chapter 3 identified the four strategies for recognition and detection for the purpose of categorization and localization of objects in images. The BoVW model was identified as a categorization

method since it is a standard method for categorizing objects in images and has been shown to yield good results. Three strategies for counting objects in images were identified and general disadvantages associated with the strategies were identified. The second research objective was met by answering the second research question through the broad review of computer vision concepts and algorithms.

Research Question Three: What are suitable image representations for categorization and count estimation?

The third research question was answered through an analysis of tyre images and tyre stockpile images in Chapter 4 taking into account the findings from the discussion of computer vision concepts and algorithms.

Counting by segmentation was found to be inappropriate in the context of visible tyre count estimation due to low contrast between individual tyre boundaries and counting by detection was also found to be inappropriate when using global features or local features for detection (Section 4.1). Global feature based detection methods are unsuitable due to high levels of occlusion in tyre stockpiles and the large number of visual appearances resulting from occlusions and tyre angles in the stockpiles. Alternatively local feature based detection methods, that are used for detection when there is object occlusion, were also found to be unsuitable due to the similarity in features across all tyres in the tyre stockpiles which greatly reduces the ability of finding one-to-one correspondences of detections and actual tyre instances in the images. The problems of tyre categorization and count estimation were then scoped to an experimental investigation into using different local feature detector-descriptor combinations in the BoVW model for categorization and a counting by regression strategy for estimating the number of visible tyres in tyre stockpile images. For tyre categorization (Section 4.2) based on tread patterns, a metric based on the MSE of matched features within and between tyre categories was proposed as an indicator of local feature detector-descriptor suitability and a hypothesis was made as to which local feature detector-descriptor combination would be most suitable within a BoVW context for tyre categorization based on tread patterns. For visible tyre count estimation in tyre stockpiles (Section 4.3), two global feature detection methods were proposed to describe global stockpile image characteristics for use in a regression method. The first global feature description discussed was based on using a dense grid to extract HOG features and then describe the stockpile using a BoVW for feature quantization resulting in a visual word occurrence histogram de-

scriptor. The second proposed global feature description method was based on combined histograms of gradient orientations, gradient magnitudes, and edge orientations. Both descriptors for summarizing the global stockpile characteristics based on their visual appearance aim to summarize the structural characteristics through image intensity changes.

The strategies for categorization and counting in images were reviewed for the domain of tyres. Disadvantages associated with counting by segmentation and detection strategies were highlighted which led to the conclusion that counting by regression is the most appropriate strategy for estimating the number of visible tyres in tyre stockpile images. Local feature detector-descriptor combinations were reviewed and it was hypothesized that the Harris detector combined with the SURF descriptor would perform well for categorization at the specific instance level of categorization while for more general categorization, the local feature detector-descriptor combination suitability would need to be determined experimentally in the BoVW model for representing images. Finally, two image representations for count estimation were proposed.

Research Question Four: How can machine learning methods be used for the categorization of individual tyres and estimating the number of visible tyres in tyre stockpiles?

Chapter 5 described machine learning algorithms to answer research question four. The machine learning algorithms that are used in the original implementation of the BoVW model were discussed and Neural Networks were discussed for non-linear function approximation from global stockpile image features to scalar count estimations.

The machine learning algorithms used in the BoVW model for feature vector quantization using k-means and k-nn were discussed (Section 5.1). Categorization by means of classification using multiclass extensions of the binary SVM were described and the ECOC extension was chosen due to increased classification robustness. Various methods for evaluating the performance of multiclass classifiers were discussed. The performance measures discusseed included average accuracy, average error rate, average precision, and average recall.

Regression has been used in a number of computer vision counting applications in various fields. For applications that must deal with high levels of occlusion, the discussion on the counting by regression section in Chapter 3, it was found in the literature that

non-linear mappings provide increased accuracy for count estimation where object occlusions are expected and that NNs provide increased count estimation accuracy over linear regression models. The feedforward neural network was described with the standard backpropagation algorithm for updating weights to optimize the network error given the available training data (Section 5.2). The evaluation methods discussed for measuring the performance of a trained NN included summary statistics and variance of the prediction error as well as the MSE and correlation coefficients for both the testing and training data. The difference between the results for the training and testing set provide an indication of the generalization ability of the trained NN when presented with new data.

Through descriptions and discussion of classification methods used in the BoVW model and descriptions and discussion of using NNs for non-linear regression, the fourth research question was answered.

8.1.2 Research Objective Two: Application and Evaluation of Identified Methods

The second research objective RO_2 focuses on the application and evaluation of identified methods for tyre categorization and visible tyre count estimation in tyre stockpile images. To meet the second research objective the fifth sub research question RQ_5 and the sixth sub research question RQ_6 had to be answered. This section describes how RQ_5 and RQ_6 were answered to meet the second research objective.

Research Question Five: How can experiments be designed to determine the appropriateness of the identified categorization and count estimation methods?

The fifth sub research question was answered in Chapter 6. Separate experiments were designed for categorization of tyres based on tyre tread images (Section 6.1) and visible tyre count estimation (Section 6.2).

Two data sets were used for evaluating the BoVW at the specific level and general level of categorization. Specific level images were collected by photographing tyre treads at a tyre retailer and general level tyre images were sourced by cropping individual tyres from images sourced at the REDISA tyre depot. All images were pre-processed for subsequent algorithms by converting them to grayscale images to remove irrelevant colour information while maintaining structural information. A histogram equalization step

was used to enhance image contrast by transforming the image pixel distributions to a more uniform distribution. Finally, a median filter was used to remove noise while preserving edges formed by tread groves. A cluster size of $k = 1000$ was used in the experiments as it was found to provide a good trade off between classification accuracy and training time. The selected local feature detector-descriptor combinations were evaluated using the averaged results obtained after using three randomly subsampled splits on the training images were each split was a random split of 70% training to 30% testing data.

Three data sets were used to evaluate the use of the proposed feature extraction methods for count estimation. The data sets consisted of a data set of images containing multiple circles, a data set of 3D rendered stockpile images, and a data set of real world stockpile images. The circle images were generated using Matlab, the 3D generated stockpile images were produced using blender, and the real-world stockpile images were sourced by photographing stockpiles at a REDISA depot. The tyre stockpiles images were manually segmented. All images in all data sets were preprocessed by applying histogram equalization to the object, applying a small Gaussian filter, followed by a median filter. The images containing circles were used to evaluate the methods on a data set that was simpler in terms of the occlusions and object appearances in the images. The images containing generated 3D stockpile images were used to evaluate the counting methods on a data set that was more similar to real world stockpiles while containing less noise. Finally, the methods were evaluated on the real world stockpile images. For each data set, three NNs were trained for each of the three considered NN structures per data set. The three NNs per NN structure for each data set were trained using different subsets of the data set. The NNs were evaluated using the averaged results from 3-fold cross validation for each NN structure for each data set.

To answer the fifth research question RQ_5, experimental designs were discussed for both categorization and count estimation from data collection to model evaluation.

Research Question Six: How well do the identified methods work for tyre categorization and visible tyre count estimation?

In Chapter 7 the experimental results for categorizing individual tyres (Section 7.1) based on tread images and visible tyre count estimation in tyre stockpiles (Section 7.2) were presented.

The categorization results showed that corner based local feature detectors combined with the SURF descriptor resulted in the best categorization accuracy while performing only slightly better than the other detector-descriptor combinations in terms of average precision and average recall. Overall the results of categorization experiments indicate that local feature detectors and descriptors in a BoVW model, for categorization at both the specific instance and general levels of categorization, provide a potential avenue for the categorization of tyres based on their tread.

The results of the count estimation experiments were split into three sections for each feature extraction technique considered. The results of applying the two feature extraction techniques for NN input resulted in similar good results in terms of accuracy and generalization ability on images containing simple circles. The results from using both feature extraction techniques for images of generated tyre stockpile images indicated that it is more difficult to predict object count when there is more variation in object appearance, specifically due to random occlusions although reasonable visible object count estimations can be obtained. The results obtained when using each of the feature extraction methods on real world stockpile images indicate that it is more difficult to obtain count estimations from real world stockpile images. The discussion noted that the increased difficulty could be due to a number of factors such as image noise and prominent dirt on the tyres both affecting the extracted gradient magnitude and orientation data as well as the extracted edges and their corresponding edge pixel orientations. The results also showed that as the number of visible objects in the images increased, the prediction error increased. The use of custom histograms based on gradient orientation, magnitude, and edge orientations resulted in better prediction and generalization ability for both the generated tyre stockpile images and the real world stockpile images.

The sixth research question was answered through the presentation and discussion of count estimation results. The results showed that while the proposed feature extraction techniques work well when combined with NNs for count estimation on simple images, the application to the domain of tyre stockpiles requires further research if improved accuracy and generalization ability are required for visible tyre count estimation in tyre stockpiles.

8.2 Summary of Contributions

The contributions of this research to the existing body of knowledge are largely a result of limited research in the particular domain of tyre and tyre stockpile image analysis. Concepts from the field of computer vision that have been applied in various domains were reviewed to provide specific focus to techniques applicable to the domain of tyres and tyre stockpiles. This study made contributions to the existing body of knowledge in terms of theoretical contributions (Section 8.2.1) as well as practical contributions (Section 8.2.2).

8.2.1 Theoretical Contributions

The theoretical contributions of this study include a broad review of digital image processing, computer vision, and a selection of algorithms that can be used for the purposes of tyre categorization and count estimation. The difficulties associated with tyre counting in tyre stockpile images by segmentation and detection were highlighted. Two feature extraction methods were proposed for use with NNs for the purpose of estimating the number of similarly shaped objects in images. The feature extraction techniques proposed for object count estimation are based on existing work for feature extraction from images but attempt to provide a global summary of structural information that does not rely only on edge structures but takes into account aspects such as low contrast boundaries between individual tyres in tyre stockpiles by using image gradient information. The domain of tyre image processing has limited sources, and thus this work offers a theoretical contribution to the domain.

The experimental design also offers a practical contribution in that the experimental design for both tyre categorization and visible tyre count estimation can be used by other researchers for determining the performance of classification models and regression models for the purposes of both image categorization and visible tyre count estimation from tyre stockpile images. The experimental design can be applied to similar categorization and count estimation experiments in the domain of tyres and tyre stockpiles respectively or it can be applied to other domains where categorization and count estimations of objects in images are required.

8.2.2 Practical Contributions

The practical contributions of this study include the use of the BoVW model in the domain of tyre categorization and an evaluation of results obtained from using various local feature detector-descriptor combinations in the BoVW model for tyre categorization.

The practical contributions of this study for visual object counting are the application and evaluation of NNs for visible tyre count estimation from tyre stockpile images. The BoVW model has previously not been used to describe images for the purposes of regression and is also considered a contribution to the existing body of knowledge. The second feature descriptor constructed from concatenated histograms of image gradient orientations and magnitudes, and edge orientations is considered a contribution since it was found to provide promising results when used as input for NNs for the purpose of object count estimation in images.

The results from the categorization and visible object count estimation offer a practical contribution. The results of the experiments provide an indication of the accuracy and potential usefulness of such algorithms for categorization and count estimation of tyres and tyre stockpiles in real life applications. In addition, the results combined with the critical analysis of computer vision and machine learning algorithms for categorization and count estimation provide indications of potential future avenues that can be explored to improve these results in future research.

8.3 Limitations of Study

The domain of tyre tread categorization has previously only been considered for tyre skid marks Wang (2007) and binary tread pattern images (Huang *et al.*, 2010). There has not been any research on count estimation in the domain of tyre stockpile images. Count estimation has been researched in the domains of surveillance, car counting, and cell counting. In designing a feature extraction technique for count estimation, the high degree of occlusion was considered when designing feature description techniques although the high degree of occlusion in the domain of study was considered to be a significant limitation in terms of potential accuracy of a count estimation solution.

The impact limitations of this study concerned with the usability of this research are that much more research is required for the solutions to be usable by REDISA. Although

the results seem promising for 3D generated stockpile images, the range of visible tyres in the images is relatively small in comparison to the size of real-world stockpile images. The solutions were evaluated on images containing less than sixty visible and partially visible objects and so the results of the proposed solutions for object counting can not be extrapolated to images containing greater numbers of object instances unless images containing more object instances are included in the training data during model fitting.

The data collection in terms of image acquisition for tyre categorization and visible tyre count estimation presented the greatest limitations in this study. Although multiple photos were taken during the field trip to the REDISA depot, only three stockpiles of random arrangements were available, two of which had similar visible tyre counts when photographed from all angles which was one reason for creating randomly arranged 3D generated stockpiles. Another limitation in terms of data is that manual segmentation was required due to the nature of images sourced from the real world. The photographs of stockpiles obtained at the REDISA depot contained background clutter in terms of tyre piles in the background in all directions, making it difficult to acquire photos where there was only one pile of focus in the obtained image which led to manual segmentation of the tyre stockpiles for the purpose of this study.

8.4 Recommendations for Future Research

The research conducted in this study provides a starting point for computer vision applications that can be used in the domain of tyre recycling. A number of future research prospects have been identified through this research. Since there is little research in the field of computer vision that is directly related to the domain of tyres and tyre stockpiles, there are many prospects for future research.

Further research is required that focuses specifically on the feature extraction techniques for tyre categorization and tyre stockpile count estimation. Localization of tyres in tyre stockpiles could also form part of future research if the study is highly focused on developing a segmentation technique that is specific to tyre stockpile segmentation from the background or the segmentation of individual tyres from each other. Due to the limitations around the data collection for learning models for count estimation, the techniques proposed in this study require testing with larger data sets of real tyre stockpile images that are representative of a much larger number of physical tyre stockpiles. A system for classifying tyre stockpiles prior to estimating the count could also form part

of a future research study. Tyre stockpiles vary in the number of tyres they contain so there is potential for classifying the tyres into classes representing a possible range. For example tyre stockpiles could be classified as containing between zero and 100, 100 to 200, or 300 to 400 so that the there size range classification could be used to determine a more accurate count using a model trained specifically for that size range. Finally the techniques for feature extraction for count estimation could be studied in the context of different tyre stockpile arrangements specifically flat or stacked stockpile arrangements.

8.5 Final Conclusion

The experimental results presented in this study for categorization indicate that the BoVW model can be used with various feature detector-descriptor combinations for categorizing tyres based on images of the tyre tread. The results indicated that corner based local feature detectors combined with the SURF descriptor result in more accurate categorizations than when using the SIFT descriptor and region based local feature detectors. Although the categorization results indicate that a good level of categorization accuracy can be achieved using the BoVW model, a much larger set of training data is required to determine whether a classifier can be created that is accurate enough for practical applications.

The experimental results for count estimation using the two feature extraction methods to create feature vectors for input to ANNs showed that fairly accurate count estimations can be obtained on images of simple objects. As the number of possible views of the objects increases as in the case of tyres, the accuracy decreases in comparison to the case of simple object counting. Using concatenated histograms of gradient orientation, magnitude, and edge orientations resulted in superior accuracy and generalization ability compared to that of the HOG visual word occurrence feature extraction method particularly when there is more variation in object appearances in the images.

Due to the lack of research concerning computer vision in the particular domain of tyre categorization and count estimation, this research is unique in its experimental work in the domain. This research forms a foundation for future research on using computer vision for tyre categorization and count estimation which will enable real life application for the identification and collection of tyres for recycling.

Appendices

A Data Set Image Examples

To perform the experiments discussed in Chapter 6 data sets of images were required. This appendix shows sample images used for the various experiments concerning tyre categorization and visible tyre count estimation.

A.1 Tyre Categorization Images

Two data sets of tyre tread images were used in the categorization experiments. The image set consisted of tyres that were organized into specific tyre instance categories and the second image set consisted of tyres that were separated into the categories of passenger, 4x4, or truck tyres.

Figure 1: Subset of images used in specific level categorization of tyres.

Figure 2: Subset of images (truck, passenger, 4x4) used in general level categorization of tyres.

A.2 Images For Count Estimation

Three data sets were used for the experiments on visible object count estimation. The first data set consisted of images containing various numbers of circles. This data set was used to determine the performance of the proposed methods for visible tyre count estimation on a data set consisting of simple objects.

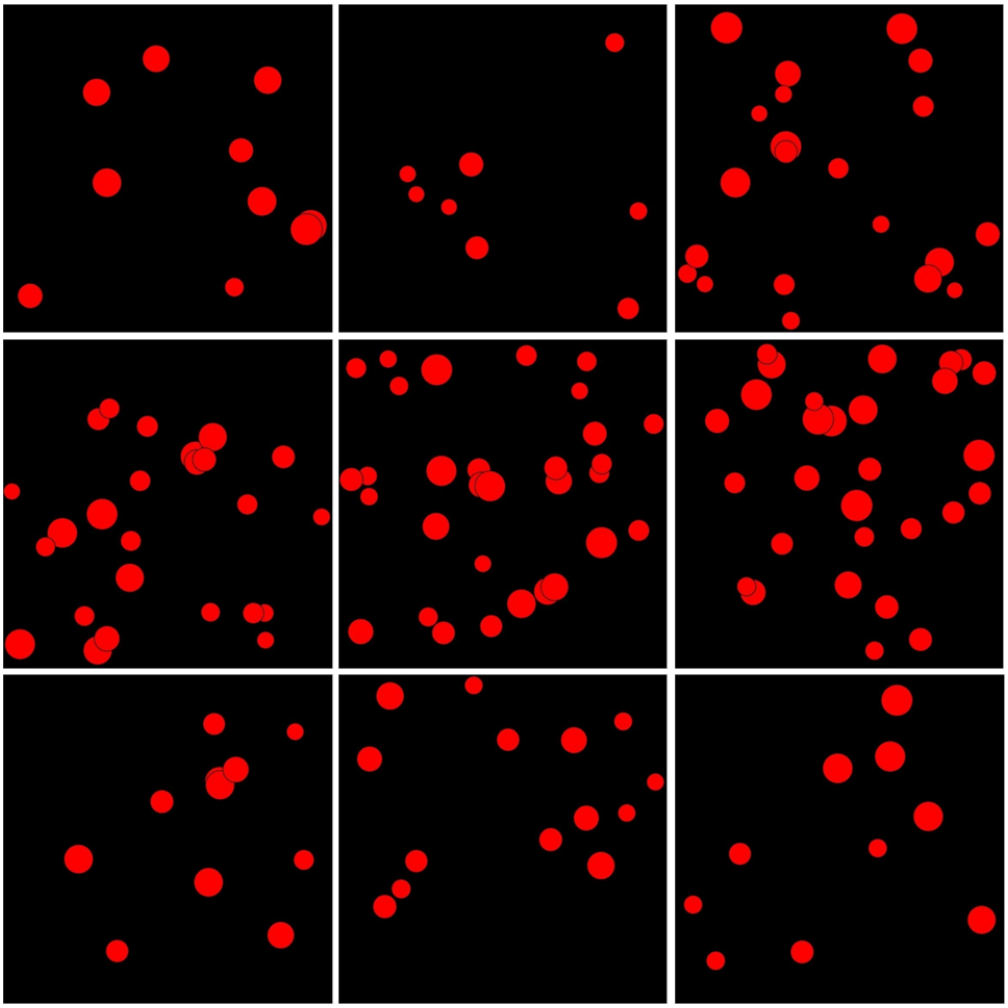


Figure 3: Subset of circle images used for testing count estimation methods on images containing simple objects with minimal variation in appearance.

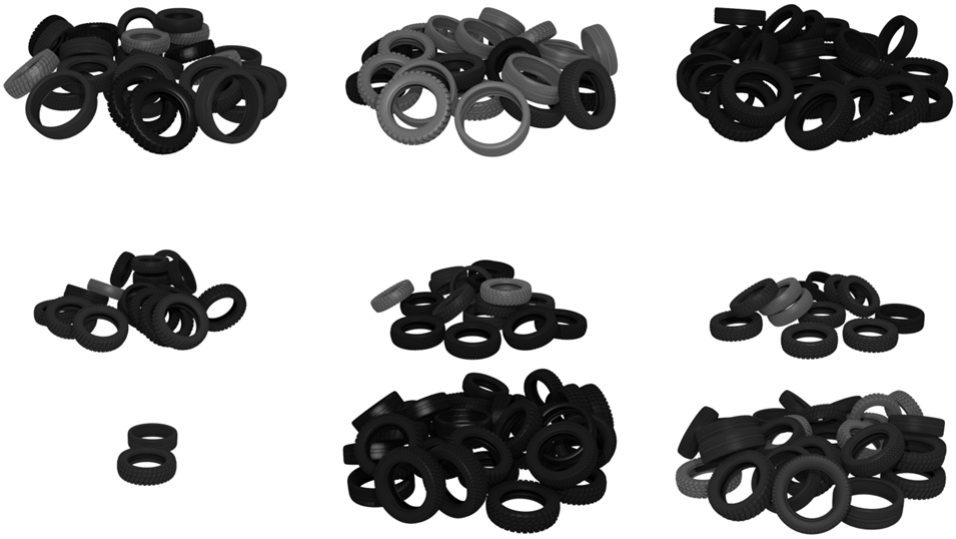


Figure 4: Subset of 3D generated tyre stockpiles used for testing count estimation methods on larger data set than real world stockpile images data set and containing more object appearance variation than simple circles.

Figure 5: Subset of real world stockpile images collected from REDISA depot field trip used for testing count estimation methods on real world stockpile images.

B Stockpile Visible Tyre Count Estimation Results

The averaged results from 3-fold cross validation were given in Chapter 7. The results from each iteration of the 3-fold cross validation for both feature extraction methods used for describing images for counting are given here for completeness. The results are organized into tables, each with a heading showing $[m, n]$ where m and n are the number of hidden layers in the NN and k is the iteration of the k-fold cross validation.

B.1 Custom Histogram Feature Extraction

The first set of experiments were conducted using the second feature extraction method discussed in Section 4.3.1. The results for training a NN using feature vectors extracted using the second method are given for training and testing in 3-fold cross validation for the circle images data set, the generated tyre stockpile images data set, and the real world stockpile images data set.

[5,5]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.932534	0	7	1	1	0.954900928	1.803530721	0.99286
Test	1.109445	0	7	1	1	1.232748491	2.361457204	0.99203
[5,5]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.010511	0	6	1	1	0.88259613	1.816765702	0.99277
Test	1.092814	0	7	1	1	1.141509773	2.220119621	0.99094
[5,5]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.04048	0	5	1	1	0.891752322	1.921600111	0.99254
Test	1	0	6	1	1	0.927710843	1.923612373	0.99246
[5,5]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.994508	0	6	1	1	0.909749793	1.847298844	0.992723333
Test	1.06742	0	6.666666667	1	1	1.100656369	2.168396399	0.99181
[10,10]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.956522	0	6	1	1	0.933542238	1.813119106	0.99277
Test	1.096096	0	5	1	1	1.207605196	2.273725014	0.99135
[10,10]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.90991	0	6	1	0	0.924202398	1.672830114	0.99334
Test	1.068862	0	8	1	1	1.235483987	2.32940485	0.99082
[10,10]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.983508	0	5	1	1	0.917145031	1.771619681	0.99289
Test	0.975976	0	5	1	1	0.915083758	1.783643053	0.99269
[10,10]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.94998	0	5.666666667	1	0.666666667	0.924963222	1.752522967	0.993
Test	1.046978	0	6	1	1	1.11939098	2.128924306	0.99162
[20,20]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.898051	0	5	1	1	0.785386586	1.516425305	0.99378
Test	1.051051	0	4	1	1	1.144976302	2.127491013	0.99197
[20,20]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.882883	0	6	1	1	0.819345661	1.508384404	0.99393
Test	1.047904	0	7	1	1	1.096797396	2.099653635	0.9915
[20,20]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.937031	0	6	1	1	0.920953937	1.712464636	0.99321
Test	1.006006	0	5	1	1	0.993939723	1.864373369	0.99215
[20,20]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.905988	0	5.666666667	1	1	0.841895395	1.579091449	0.99364
Test	1.034987	0	5.333333333	1	1	1.07857114	2.030506006	0.991873333

Table 1: NN results for different NN structures concatenated histograms of image gradient direction histogram, gradient magnitude histogram, and edge image gradient direction histogram on images containing circles.

[5,5]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.605990783	0	8	1	1	1.915991741	4.405123831	0.98407
Test	1.990740741	0	14	1	1	4.381309216	8.26758299	0.9716
[5,5]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.508083141	0	8	1	1	1.94495766	4.089809656	0.98533
Test	2.276497696	0	16	2	1	6.460232122	11.61694301	0.96019
[5,5]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.70669746	0	10	1	1	2.272570781	5.171671166	0.9819
Test	2.02764977	0	10	2	1	2.897380099	7.009415735	0.97572
[5,5]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.606923795	0	8.666666667	1	1	2.044506727	4.555534885	0.983766667
Test	2.098296069	0	13.33333333	1.666666667	1	4.579640479	8.964647246	0.96917
[10,10]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.313364055	0	7	1	1	1.490490736	3.183805641	0.98863
Test	1.833333333	0	14	1	1	3.76744186	7.109721022	0.97584
[10,10]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.473441109	0	8	1	1	1.786908733	3.840753069	0.98625
Test	2.165898618	0	14	2	1	4.694572453	9.272979832	0.96748
[10,10]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.538106236	0	7	1	1	1.846345479	4.084032014	0.98531
Test	1.926267281	0	10	1	1	3.401945725	6.797206461	0.97494
[10,10]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.441637133	0	7.333333333	1	1	1.707914983	3.702863575	0.98673
Test	1.975166411	0	12.66666667	1.333333333	1	3.954653346	7.726635772	0.972753333
[20,20]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.368663594	0	7	1	1	1.508115069	3.281703701	0.98804
Test	1.912037037	0	13	1	1	4.238738157	7.992487391	0.9731
[20,20]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.230946882	0	6	1	1	1.321540074	2.806678078	0.99016
Test	2.331797235	0	13	2	1	5.796808329	11.12471827	0.96176
[20,20]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.512702079	0	7	1	1	1.986528099	4.191440973	0.9851
Test	1.949308756	0	8	2	1	2.872418501	6.555962924	0.97691
[20,20]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.370770852	0	6.666666667	1	1	1.605394414	3.426607584	0.987766667
Test	2.064381009	0	11.33333333	1.666666667	1	4.302654996	8.557722862	0.97059

Table 2: NN results for different NN structures concatenated histograms of image gradient direction histogram, gradient magnitude histogram, and edge image gradient direction histogram on images containing 3D generated stockpiles.

[5,5]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.447154472	0	8	1	1	2.101692656	4.176486391	0.98639
Test	3.564516129	0	14	3	1	10.38101534	23.35606062	0.91265
[5,5]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.290322581	0	5	1	1	1.378442172	3.077392599	0.9879
Test	3.786885246	0	21	2	2	14.83715847	28.36472331	0.91748
[5,5]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.048780488	0	4	1	1	0.915633747	1.84225181	0.99334
Test	3.661290323	0	28	3	1	17.34241142	30.48963914	0.88761
[5,5]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.262085847	0	5.666666667	1	1	1.465256191	3.0320436	0.98921
Test	3.670897233	0	21	2.666666667	1.33333	14.18686174	27.40347435	0.905913333
[10,10]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.780487805	0	3	1	0	0.664534186	1.18106141	0.99589
Test	4.338709677	0	30	3	1	24.78503437	42.13063574	0.86787
[10,10]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.032258065	0	4	1	1	0.958300551	1.917812485	0.99197
Test	3.967213115	0	21	3	3	16.8989071	32.61644384	0.90393
[10,10]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1	0	4	1	1	0.803278689	1.70386778	0.99404
Test	3.032258065	0	26	2	1	13.17927023	22.77903563	0.91616
[10,10]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.937581956	0	3.666666667	1	0.66667	0.808704475	1.600913892	0.993966667
Test	3.779393619	0	25.66666667	2.666666667	1.66667	18.28773723	32.50870507	0.895986667
[20,20]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.520325203	0	3	0	0	0.546714647	0.711529795	0.99737
Test	4.338709677	0	28	2.5	1	25.93257536	44.36298404	0.85144
[20,20]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.403225806	0	3	0	0	0.405192762	0.481281882	0.99776
Test	4.31147541	0	19	3	0	18.61803279	36.25459334	0.89111
[20,20]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.455284553	0	4	0	0	0.512328402	0.675493333	0.99763
Test	3.467741935	0	14	3	1	9.203860391	21.04136902	0.92969
[20,20]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.459611854	0	3.333333333	0	0	0.488078604	0.622768337	0.997586667
Test	4.039309008	0	20.33333333	2.833333333	0.66667	17.91815618	33.88631547	0.890746667

Table 3: NN results for different NN structures concatenated histograms of image gradient direction histogram, gradient magnitude histogram, and edge image gradient direction histogram on images containing real world stockpiles.

B.2 HOG BoVW from Overlapping Sub-Images Feature Extraction

The second set of experiments were conducted using the first feature extraction method discussed in Section 4.3.1. The results for training a NN using feature vectors extracted using the first method are given for training and testing in 3-fold cross validation for the circle images data set, the generated tyre stockpile images data set, and the real world stockpile images data set.

[5,5]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.109445277	0	7	1	1	1.232748491	2.361457204	0.99037
Test	1.204204204	0	6	1	1	1.494319621	2.899862843	0.98918
[5,5]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.15015015	0	6	1	1	1.222533812	2.451297441	0.99036
Test	1.203592814	0	6	1	1	1.357824891	2.720641948	0.989
[5,5]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.164917541	0	6	1	1	1.197986592	2.47821398	0.99028
Test	1.291291291	0	5	1	1	1.345616701	2.890501207	0.9884
[5,5]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.141504323	0	6.333333333	1	1	1.217756298	2.430322875	0.990336667
Test	1.233029437	0	5.666666667	1	1	1.399253738	2.837001999	0.98886
[10,10]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.128935532	0	6	1	1	1.235602919	2.411954804	0.99019
Test	1.228228228	0	6	1	1	1.459803177	2.9271116	0.98916
[10,10]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.100600601	0	5	1	1	1.080089864	2.21766259	0.99133
Test	1.24251497	0	7	1	1	1.529598461	2.974802182	0.98797
[10,10]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.113943028	0	5	1	1	1.221231726	2.406977949	0.99063
Test	1.204204204	0	5	1	1	1.373837693	2.720241719	0.98917
[10,10]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.114493054	0	5.333333333	1	1	1.178974837	2.345531781	0.990716667
Test	1.224982467	0	6	1	1	1.45441311	2.874051834	0.988766667
[20,20]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.026986507	0	6	1	1	1.056327692	2.002863899	0.99175
Test	1.264264264	0	6	1	1	1.598628749	3.138345267	0.9884
[20,20]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.085585586	0	6	1	1	1.146047551	2.257657537	0.9912
Test	1.095808383	0	6	1	1	1.270072468	2.462290973	0.99033
[20,20]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.062968516	0	7	1	1	1.200233217	2.229141998	0.99124
Test	1.225225225	0	5	1	1	1.482253338	2.905624594	0.98819
[20,20]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.058513536	0	6.333333333	1	1	1.13420282	2.163221145	0.991396667
Test	1.195099291	0	5.666666667	1	1	1.450318185	2.835420278	0.988973333

Table 4: NN results for different NN structures using HOG visual word occurrence histograms on images containing circles.

[5,5]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.661290323	0	14	1	1	2.603255606	5.205838102	0.98048
Test	2.277777778	0	26	1	1	11.45736434	16.54237821	0.94522
[5,5]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.877598152	0	16	1	1	3.890075699	7.332770629	0.97494
Test	1.98156682	0	17	1	1	4.212621608	7.929254283	0.96929
[5,5]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.755196305	0	11	1	1	2.208450945	5.109073601	0.98139
Test	2.566820276	0	24	2	2	10.8114866	17.5211411	0.93888
[5,5]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.764694927	0	13.66666667	1	1	2.900594083	5.882560777	0.978936667
Test	2.275388292	0	22.33333333	1.333333333	1.333333333	8.827157517	13.9975912	0.95113
[10,10]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.442396313	0	13	1	1	2.12715914	4.219380307	0.98472
Test	2.615740741	0	16	2	1	9.5958441	16.35870945	0.94559
[10,10]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.521939954	0	9	1	1	1.97231845	4.194148739	0.98559
Test	2.447004608	0	20	2	1	5.674261819	11.56884814	0.9567
[10,10]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.473441109	0	8	1	1	1.82394577	3.790878396	0.98597
Test	2.377880184	0	34	2	0	11.01395289	16.50748996	0.94274
[10,10]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.479259125	0	10	1	1	1.974474454	4.068135814	0.985426667
Test	2.480208511	0	23.33333333	2	0.666666667	8.761352937	14.81168252	0.948343333
[20,20]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.320276498	0	6	1	1	1.312890455	3.043416335	0.98893
Test	2.412037037	0	23	1	1	10.28990095	16.13691261	0.94693
[20,20]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.304849885	0	8	1	1	1.531851424	3.138115304	0.98915
Test	2.456221198	0	28	2	1	7.684417136	13.48718524	0.94931
[20,20]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.256351039	0	6	1	1	1.297557951	2.798275808	0.98992
Test	2.350230415	0	25	1	1	9.284178187	14.65841616	0.94895
[20,20]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.293825807	0	6.666666667	1	1	1.38076661	2.993269149	0.989333333
Test	2.406162883	0	25.33333333	1.333333333	1	9.086165424	14.760838	0.948396667

Table 5: NN results for different NN structures using HOG visual word occurrence histograms on images containing 3D generated stockpiles

[5,5]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.284552846	0	5	1	1	1.369185659	2.915560382	0.98927
Test	5.596774194	0	33	5	2	31.52326811	62.25569056	0.80099
[5,5]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.290322581	0	7	1	1	1.768686074	3.340461868	0.98805
Test	4.737704918	0	26	3	2	26.93005464	48.73587768	0.84519
[5,5]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.406504065	0	8	1	0	2.325203252	4.148716531	0.98549
Test	4	0	26	3	1	16.68852459	32.36327275	0.88347
[5,5]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	1.327126497	0	6.666666667	1	0.666666667	1.821024995	3.46824626	0.987603333
Test	4.778159704	0	28.33333333	3.666666667	1.666666667	25.04728245	47.784947	0.843216667
[10,10]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.398373984	0	3	0	0	0.37278422	0.532539004	0.99812
Test	5.806451613	0	39	5	1	44.28979376	77.16207721	0.72784
[10,10]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.677419355	0	3	0.5	0	0.67558353	1.037204263	0.99607
Test	5.62295082	0	36	4	3	39.67213115	69.77503225	0.7754
[10,10]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.536585366	0	3	0	0	0.56217513	0.754308054	0.99715
Test	5.774193548	0	29	4	2	31.71866737	65.2596822	0.76327
[10,10]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.537459568	0	3	0.166666667	0	0.536847627	0.774683773	0.997113333
Test	5.734531994	0	34.66666667	4.333333333	2	38.56019743	70.73226388	0.755503333
[20,20]; k=1								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.18699187	0	3	0	0	0.235239238	0.224772348	0.99905
Test	6.14516129	0	30	4	0	48.8802221	86.88761022	0.71427
[20,20]; k=2								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.306451613	0	3	0	0	0.344348282	0.363831701	0.99849
Test	6.737704918	0	36	5	1	46.19672131	90.63803318	0.75638
[20,20]; k=3								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.105691057	0	1	0	0	0.095295215	0.114476642	0.99964
Test	5.370967742	0	32	3	1	46.46668429	74.22953785	0.79421
[20,20]; Averages for 3 folds								
	Mean	Minimum	Maximum	Median	Mode	Variance	MSE	R
Train	0.199711513	0	2.333333333	0	0	0.224960912	0.23436023	0.99906
Test	6.084611317	0	32.66666667	4	0.666666667	47.18120924	83.91839375	0.754953333

Table 6: NN results for different NN structures using HOG visual word occurrence histograms on images containing real world stockpiles.

C Individual Tyre Categorization Results

The average results for individual tyre categorization were presented in Section 7.1. The SURF and SIFT local feature descriptors were combined with various local feature detectors to determine which feature detector-descriptor combination provides the most accurate categorization results for categorizing tyres at a specific level of categorization and at a general level of categorization in the BoVW model. The results for each of the three iterations of training and testing using the random sub-sampling for each experiment are presented here for completeness.

C.1 Specific Instance Categorization Results

The specific level of categorization was evaluated for the various local feature detector-descriptor combinations. The following tables contain the confusion matrices, per class precision and recall, as well as the average accuracy, average precision, and average recall for each considered local feature detector-descriptor combinations for the specific level categorization experiments..

Feature Detectors with SIFT Descriptor

The following table shows the results of classifying the tyre tread images based on various local feature detectors and features described using the SIFT descriptor in the test set for each iteration of the random sub-sampling at the specific level of categorization.

Harris Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	**1**
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	1	0	0	0	4	1	0	0	0	0.6666667
	Michelin2	0	1	1	0	0	4	0	0	0	0.6666667
	Pirelli	0	0	1	0	0	0	4	0	0	0.8
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	4	0	0	0	0	0	0	0
	Precision	0.666667	0.666667	0.4	1	1	0.8	1	1	0	

# Features	1674
# Clusters	1000
Accuracy	0.79
Precision	0.7259259
Recall	0.7925926

Harris Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	1	0	0	0	1	0	0	0	0	0.5
	Caravan	0	1	0	0	0	1	0	0	0	0.5
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	1	0	0	0	5	0	0	0	0	0.8333333
	Michelin2	0	1	0	0	0	5	0	0	0	0.8333333
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	1	0	1	0	1	0	1	0.25
	Precision	0.5	0.5	0.8	1	0.71429	0.833333	0.833333	1	0	

# Features	1926
# Clusters	1000
Accuracy	0.77
Precision	0.6867725
Recall	0.7685185

Harris Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	2	0	0	0	0	0	0	0	0	0
	Continental	0	0	3	0	0	0	0	0	1	0.75
	Hankook	0	0	0	3	0	0	0	1	0	0.75
	Michelin	2	0	0	0	4	0	0	0	0	0.6666667
	Michelin2	0	0	0	1	1	4	0	0	0	0.6666667
	Pirelli	0	0	0	0	0	0	4	1	0	0.8
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	1	0	2	0	0	0	1	0.25
	Precision	0.333333	0	0.75	0.75	0.57143	1	1	0.5	0	

# Features	2061
# Clusters	1000
Accuracy	0.65
Precision	0.5449735
Recall	0.6537037
Average Accuracy	0.7366667
Average Precision	0.6525573
Average Recall	0.7382716

DoG Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	3	0	0	0	0	0	1	0.75
	Hankook	0	0	0	2	0	0	1	1	0	0.5
	Michelin	0	1	0	0	5	0	0	0	0	0.8333333
	Michelin2	0	0	0	0	0	2	0	0	4	0.3333333
	Pirelli	0	0	0	0	0	0	3	2	0	0.6
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	1	1	1	0	0	0	1	0.25
	Precision	1	0.666667	0.75	0.66667	0.83333	1	0.75	0.4	0	

# Features	1782
# Clusters	1000
Accuracy	0.7
Precision	0.6740741
Recall	0.6962963

DoG Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	0	1	1	0	0	0	0	0	0	0
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	0	2	0	0	3	0	0	1	0	0.5
	Michelin2	0	0	0	1	0	5	0	0	0	0.8333333
	Pirelli	0	0	0	0	0	0	4	1	0	0.8
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	4	0	0	0	0	0	0	0
	Precision	0	0.4	0.44444444	0.8	1	1	1	0.5	0	

# Features	1782
# Clusters	1000
Accuracy	0.68
Precision	0.5716049
Recall	0.6814815

DoG Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	3	0	0	0	1	0	0	0.75
	Hankook	0	0	0	2	0	0	0	1	1	0.5
	Michelin	0	2	0	0	3	0	0	1	0	0.5
	Michelin2	0	0	0	0	1	5	0	0	0	0.8333333
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	2	0	0	0	0	0	2	0.5
	Precision	1	0.5	0.6	1	0.75	1	0.833333	0.5	0	

# Features	1459
# Clusters	1000
Accuracy	0.79
Precision	0.687037
Recall	0.787037
Average Accuracy	0.7233333
Average Precision	0.6442387
Average Recall	0.7216049

Fast-Hessian Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	1	1	0	0	0	0	0	0	0	0.5
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	0	2	0	0	4	0	0	0	0	0.6666667
	Michelin2	0	0	0	0	0	5	0	1	0	0.8333333
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	2	0	0	0	1	1	0	0
	Precision	0.666667	0.333333	0.66666667	1	1	1	0.833333	0.5	0	

# Features	1674
# Clusters	1000
Accuracy	0.78
Precision	0.6666667
Recall	0.7777778

Fast-Hessian Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	1	1	0	0	0	0	0	0	0	0.5
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	0	2	0	0	3	1	0	0	0	0.5
	Michelin2	0	1	0	0	1	4	0	0	0	0.6666667
	Pirelli	0	0	0	0	0	0	4	1	0	0.8
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	2	1	0	0	0	0	1	0.25
	Precision	1	0.333333	0.66666667	0.8	0.75	0.8	1	0.666667	0	

# Features	2070
# Clusters	1000
Accuracy	0.75
Precision	0.6685185
Recall	0.7462963

Fast-Hessian Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	3	0	0	0	1	0	0	0.75
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	0	0	0	0	5	0	0	1	0	0.8333333
	Michelin2	0	0	0	0	1	3	0	1	1	0.5
	Pirelli	0	0	0	1	0	0	4	0	0	0.8
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	0	0	1	1	2	0	0	0
	Precision	1	1	1	0.8	0.71429	0.75	0.571429	0.5	0	

# Features	1782
# Clusters	1000
Accuracy	0.76
Precision	0.7039683
Recall	0.7648148
Average Accuracy	0.7633333
Average Precision	0.6797178
Average Recall	0.762963

MSER Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	1	1	0	0	0	0	0	0	0	0.5
	Continental	0	0	3	0	0	0	1	0	0	0.75
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	2	2	0	0	2	0	0	0	0	0.3333333
	Michelin2	0	1	0	1	0	4	0	0	0	0.6666667
	Pirelli	0	0	0	0	0	0	4	0	1	0.8
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	1	2	0	0	0	0	0	1	0.25
	Precision	0.4	0.2	0.6	0.8	1	1	0.8	1	0	

# Features	2070
# Clusters	1000
Accuracy	0.7
Precision	0.6444444
Recall	0.7

MSER Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	3	0	0	0	0	0	1	0.75
	Hankook	0	0	0	3	0	0	0	1	0	0.75
	Michelin	0	2	0	0	4	0	0	0	0	0.6666667
	Michelin2	1	0	0	0	2	3	0	0	0	0.5
	Pirelli	0	0	1	1	0	0	3	0	0	0.6
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	1	0	0	0	0	0	3	0.75
	Precision	0.666667	0.5	0.6	0.75	0.66667	1	1	0.666667	0	

# Features	1683
# Clusters	1000
Accuracy	0.78
Precision	0.65
Recall	0.7796296

MSER Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	0	1	0	0	1	0	0	0	0	0
	Caravan	0	1	0	0	0	1	0	0	0	0.5
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	1	3	0	0	0	0	0	0.75
	Michelin	0	0	0	0	6	0	0	0	0	1
	Michelin2	0	0	0	0	1	5	0	0	0	0.8333333
	Pirelli	0	0	0	0	0	0	4	0	1	0.8
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	0	0	1	1	1	0	1	0.25
	Precision	0	0.5	0.8	1	0.66667	0.714286	0.8	1	0	

# Features	1818
# Clusters	1000
Accuracy	0.68
Precision	0.6089947
Recall	0.6814815
Average Accuracy	0.72
Average Precision	0.6344797
Average Recall	0.7203704

Harris-Laplace Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	1	1	0	0	4	0	0	0	0	0.6666667
	Michelin2	0	0	0	0	2	3	0	1	0	0.5
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	3	1	0	0	0	0	0	0
	Precision	0.666667	0.666667	0.57142857	0.8	0.66667	1	1	0.666667	0	

# Features	1926
# Clusters	1000
Accuracy	0.8
Precision	0.6708995
Recall	0.7962963

Harris-Laplace Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	1	0	0	1	0	0	0	0	0.5
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	3	0	0	0	1	0	0.75
	Michelin	0	0	0	0	5	1	0	0	0	0.8333333
	Michelin2	0	0	0	0	3	3	0	0	0	0.5
	Pirelli	0	0	1	0	0	0	3	1	0	0.6
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	2	0	0	0	1	1	0	0
	Precision	1	1	0.57142857	1	0.55556	0.75	0.75	0.4	0	

# Features	1827
# Clusters	1000
Accuracy	0.69
Precision	0.6696649
Recall	0.687037

Harris-Laplace Detector - SIFT Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	1	0	0	1	0	0	0	0	0.5
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	1	0	0	0	5	0	0	0	0	0.8333333
	Michelin2	0	1	0	0	0	5	0	0	0	0.8333333
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	2	0	1	0	0	0	1	0.25
	Precision	0.666667	0.5	0.66666667	1	0.71429	1	1	1	0	

# Features	2070
# Clusters	1000
Accuracy	0.82
Precision	0.7275132
Recall	0.8240741
Average Accuracy	0.77
Average Precision	0.6893592
Average Recall	0.7691358

Table 7: Confusion matrices and averaged accuracy, precision, and recall for each random sub-sample testing iteration using the SIFT feature descriptor for the specific level categorization.

Feature Detectors with SURF Descriptor

The following table shows the results of classifying the tyre tread images based on various local feature detectors and features described using the SURF descriptor in the test set for each iteration of the random sub-sampling at the specific level of categorization.

Harris Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	2	0	0	0	2	0	0.5
	Michelin	1	2	0	0	3	0	0	0	0	0.5
	Michelin2	0	1	0	0	0	5	0	0	0	0.833333
	Pirelli	0	0	0	0	0	0	4	1	0	0.8
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	3	0	0	0	0	1	0	0
	Precision	0.666667	0.4	0.5714286	1	1	1	1	0.333333	0	

# Features	5022
# Clusters	1000
Accuracy	0.74
Precision	0.663492
Recall	0.737037

Harris Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	0	1	0	0	5	0	0	0	0	0.833333
	Michelin2	0	0	0	0	1	5	0	0	0	0.833333
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	1	2	0	0	0	1	0	0	0
	Precision	1	0.5	0.6666667	1	0.83333	1	0.833333	1	0	

# Features	4878
# Clusters	1000
Accuracy	0.85
Precision	0.759259
Recall	0.851852

Harris Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	3	0	0	0	0	0	1	0.75
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	0	1	0	0	5	0	0	0	0	0.833333
	Michelin2	0	1	0	0	2	3	0	0	0	0.5
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	1	0	0	0	1	1	1	0.25
	Precision	1	0.5	0.75	1	0.71429	1	0.833333	0.666667	0.5	

# Features	5085
# Clusters	1000
Accuracy	0.81
Precision	0.77381
Recall	0.814815
Average Accuracy:	0.8
Average Precision	0.732187
Average Recall	0.801235

DoG Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	1	0	0	1	0	0	0	0	0.5
	Continental	0	0	2	0	0	0	0	0	2	0.5
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	1	0	0	0	5	0	0	0	0	0.833333
	Michelin2	0	0	0	0	3	3	0	0	0	0.5
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	1	0	0	0	1	0	0.5
	Vokohama	0	0	2	0	0	1	0	0	1	0.25
	Precision	0.666667	1	0.5	0.8	0.55556	0.75	1	1	0.333333	

# Features	7965
# Clusters	1000
Accuracy	0.68
Precision	0.733951
Recall	0.675926

DoG Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	3	0	0	0	0	0	1	0.75
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	0	3	0	0	2	1	0	0	0	0.333333
	Michelin2	0	2	0	1	0	3	0	0	0	0.5
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	0	0	0	1	0	0	3	0.75
	Precision	1	0.285714	1	0.8	1	0.6	1	1	0.75	

# Features	8316
# Clusters	1000
Accuracy	0.81
Precision	0.82619
Recall	0.814815

DoG Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	0	2	0	0	4	0	0	0	0	0.666667
	Michelin2	0	1	0	0	2	3	0	0	0	0.5
	Pirelli	0	0	0	0	1	0	4	0	0	0.8
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	1	0	0	0	0	0	3	0.75
	Precision	1	0.4	0.8	1	0.57143	1	1	1	1	

# Features	8748
# Clusters	1000
Accuracy	0.86
Precision	0.863492
Recall	0.857407
Average Accuracy	0.783333
Average Precision	0.807878
Average Recall	0.782716

Fat-Hessian Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	0	0	0	0	0	2	0	0	0	0
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	3	0	0	0	1	0	0.75
	Michelin	0	2	0	0	3	1	0	0	0	0.5
	Michelin2	0	0	0	0	2	4	0	0	0	0.666667
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	2	0	0	0	0	0	2	0.5
	Precision	0	0.5	0.6666667	1	0.6	0.571429	1	0.666667	1	

# Features	4238
# Clusters	1000
Accuracy	0.71
Precision	0.667196
Recall	0.712963

Fat-Hessian Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	1	0	0	0	0	1	0	0	0	0.5
	Caravan	0	1	0	0	1	0	0	0	0	0.5
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	1	0	0	0	2	3	0	0	0	0.333333
	Michelin2	0	0	0	0	0	6	0	0	0	1
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	1	0	0	0	0	0	3	0.75
	Precision	0.5	1	0.8	1	0.66667	0.6	1	1	1	

# Features	4986
# Clusters	1000
Accuracy	0.79
Precision	0.840741
Recall	0.787037

Fat-Hessian Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	2	0	0	0	0	0	2	0.5
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	0	2	0	0	4	0	0	0	0	0.666667
	Michelin2	1	0	0	0	0	5	0	0	0	0.833333
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	4	0	0	0	0	0	0	0
	Precision	0.666667	0.5	0.3333333	1	1	1	1	1	0	

# Features	3951
# Clusters	1000
Accuracy	0.78
Precision	0.722222
Recall	0.777778
Average Accuracy	0.76
Average Precision	0.743386
Average Recall	0.759259

MSER Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	1	0	0	0	1	0	0	0	0.5
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	3	0	0	0	1	0	0.75
	Michelin	0	2	0	0	4	0	0	0	0	0.666667
	Michelin2	1	0	0	0	1	4	0	0	0	0.666667
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	3	0	0	1	0	0	0	0
	Precision	0.666667	0.333333	0.5714286	1	0.8	0.666667	1	0.666667	0	

# Features	1683
# Clusters	1000
Accuracy	0.73
Precision	0.633862
Recall	0.731481

MSER Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	0	2	0	0	4	0	0	0	0	0.666667
	Michelin2	0	0	0	1	1	3	0	0	1	0.5
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	0	0	0	0	3	0	1	0.25
	Precision	1	0.5	1	0.8	0.8	1	0.625	1	0.5	

# Features	1782
# Clusters	1000
Accuracy	0.82
Precision	0.802778
Recall	0.824074

MSER Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	1	1	0	0	4	0	0	0	0	0.666667
	Michelin2	0	0	0	0	1	4	0	1	0	0.666667
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	2	0	0	0	1	0	1	0.25
	Precision	0.666667	0.666667	0.6666667	1	0.8	1	0.833333	0.666667	1	

# Features	1926
# Clusters	1000
Accuracy	0.84
Precision	0.811111
Recall	0.842593
Average Accuracy	0.796667
Average Precision	0.74925
Average Recall	0.799383

Harris-Laplace Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	2	0	0	0	0	0	0	0	0	1
	Caravan	0	2	0	0	0	0	0	0	0	1
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	0	1	0	0	5	0	0	0	0	0.833333
	Michelin2	1	0	0	0	2	3	0	0	0	0.5
	Pirelli	0	0	0	0	0	0	5	0	0	1
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	2	0	1	0	0	0	1	0.25
	Precision	0.666667	0.666667	0.6666667	1	0.625	1	1	1	1	

# Features	2070
# Clusters	1000
Accuracy	0.84
Precision	0.847222
Recall	0.842593

Harris-Laplace Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	1	0	0	0	1	0	0	0	0	0.5
	Caravan	0	1	0	0	0	1	0	0	0	0.5
	Continental	0	0	4	0	0	0	0	0	0	1
	Hankook	0	0	0	4	0	0	0	0	0	1
	Michelin	1	1	0	0	4	0	0	0	0	0.666667
	Michelin2	0	0	0	1	2	3	0	0	0	0.5
	Pirelli	0	0	0	0	0	0	4	1	0	0.8
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	0	0	1	1	0	0	1	0	1	0.25
	Precision	0.5	0.5	0.8	0.66667	0.57143	0.75	0.8	0.666667	1	

# Features	1926
# Clusters	1000
Accuracy	0.69
Precision	0.694974
Recall	0.690741

Harris-Laplace Detector - SURF Descriptor

		Predicted									
		Car	Caravan	Continental	Hankook	Michelin	Michelin2	Pirelli	Trailer	Vokohama	**Recall**
Known	Car	1	0	0	0	1	0	0	0	0	0.5
	Caravan	1	1	0	0	0	0	0	0	0	0.5
	Continental	0	0	3	0	0	0	1	0	0	0.75
	Hankook	0	0	0	3	0	0	0	1	0	0.75
	Michelin	0	0	0	0	5	0	0	1	0	0.833333
	Michelin2	0	1	0	0	1	3	0	1	0	0.5
	Pirelli	0	0	0	0	0	0	3	2	0	0.6
	Trailer	0	0	0	0	0	0	0	2	0	1
	Vokohama	1	0	0	0	0	0	1	0	2	0.5
	Precision	0.333333	0.5	1	1	0.71429	1	0.6	0.285714	1	

# Features	1683
# Clusters	1000
Accuracy	0.66
Precision	0.714815
Recall	0.659259
Average Accuracy	0.73
Average Precision	0.752337
Average Recall	0.730864

Table 8: Confusion matrices and averaged accuracy, precision, and recall for each random sub-sample testing iteration using the SURF feature descriptor for the specific level categorization.

C.2 General Level Categorization Results

The general level of categorization for categorizing tyres into 4x4, passenger, and truck tyres was evaluated using the various local feature detector-descriptor combinations. The following tables contain the confusion matrices, per class precision and recall, as well as the average accuracy, average precision, and average recall for each considered local feature detector-descriptor combinations for the general level categorization experiments.

Feature Detectors with SIFT Descriptor

The following table shows the results of classifying the tyre tread images based on various local feature detectors and features described using the SIFT descriptor in the test set for each iteration of the random sub-sampling at the general level of categorization.

Harris Detector - SIFT Descriptor

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	5	0	1	0.8333
	Passenger	0	6	0	1
	Truck	1	0	5	0.8333
	Precision	0.83333333	1	0.8333	

# Features	19602
# Clusters	1000
Accuracy	0.89
Precision	0.88888889
Recall	0.88888889

	Predicted			
	4x4	Passenger	Truck	Recall
4x4	6	0	0	1
Passenger	0	5	1	0.8333
Truck	1	3	2	0.3333
Precision	0.85714286	0.625	0.6667	

# Features	20172
# Clusters	1000
Accuracy	0.72
Precision	0.71626984
Recall	0.72222222

	Predicted			
	4x4	Passenger	Truck	Recall
4x4	6	0	0	1
Passenger	0	5	1	0.8333
Truck	1	3	2	0.3333
Precision	0.85714286	0.625	0.6667	

# Features	20172
# Clusters	1000
Accuracy	0.83
Precision	0.71626984
Recall	0.72222222

Average Accuracy	0.81333333
Average Precision	0.77380952
Average Recall	0.77777778

DoG Detector - SIFT Descriptor

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	5	1	0.8333
	Truck	2	1	3	0.5
	Precision	0.75	0.83333333	0.75	

# Features	43443
# Clusters	1000
Accuracy	0.78
Precision	0.77777778
Recall	0.77777778

	Predicted			
	4x4	Passenger	Truck	Recall
4x4	5	0	1	0.8333
Passenger	0	6	0	1
Truck	2	1	3	0.5
Precision	0.71428571	0.85714286	0.75	

# Features	41259
# Clusters	1000
Accuracy	0.78
Precision	0.77380952
Recall	0.77777778

	Predicted			
	4x4	Passenger	Truck	Recall
4x4	6	0	0	1
Passenger	0	6	0	1
Truck	3	0	3	0.5
Precision	0.66666667	1	1	

# Features	38217
# Clusters	1000
Accuracy	0.83
Precision	0.88888889
Recall	0.83333333

Average Accuracy	0.79666667
Average Precision	0.81349206
Average Recall	0.7962963

Fast-Hessian Detector - SIFT Descriptor

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	2	1	3	0.5
	Precision	0.75	0.85714286	1	

# Features	25251
# Clusters	1000
Accuracy	0.83
Precision	0.86904762
Recall	0.83333333

	Predicted			
	4x4	Passenger	Truck	Recall
4x4	6	0	0	1
Passenger	0	6	0	1
Truck	3	2	1	0.1667
Precision	0.66666667	0.75	1	

# Features	26631
# Clusters	1000
Accuracy	0.72
Precision	0.80555556
Recall	0.72222222

	Predicted			
	4x4	Passenger	Truck	Recall
4x4	6	0	0	1
Passenger	0	6	0	1
Truck	4	1	1	0.1667
Precision	0.6	0.85714286	1	

# Features	27471
# Clusters	1000
Accuracy	0.72
Precision	0.81904762
Recall	0.72222222

Average Accuracy	0.75666667
Average Precision	0.83121693
Average Recall	0.75925926

MSER Detector - SIFT Descriptor

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	5	0	1	0.8333
	Passenger	1	3	2	0.5
	Truck	1	2	3	0.5
	Precision	0.71428571	0.6	0.5	

# Features	534
# Clusters	534
Accuracy	0.61
Precision	0.6047619
Recall	0.61111111

	Predicted			
	4x4	Passenger	Truck	Recall
4x4	5	1	0	0.8333
Passenger	0	5	1	0.8333
Truck	2	2	2	0.3333
Precision	0.71428571	0.625	0.6667	

# Features	531
# Clusters	531
Accuracy	0.67
Precision	0.66865079
Recall	0.66666667

	Predicted			
	4x4	Passenger	Truck	Recall
4x4	5	1	0	0.8333
Passenger	1	4	1	0.6667
Truck	5	1	0	0
Precision	0.45454545	0.66666667	0	

# Features	507
# Clusters	507
Accuracy	0.5
Precision	0.37373737
Recall	0.5

Average Accuracy	0.59333333
Average Precision	0.54905002
Average Recall	0.59259259

Harris-Laplace Detector - SIFT Descriptor

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	1	1	4	0.6667
	Precision	0.85714286	0.85714286	1	

# Features	32070
# Clusters	1000
Accuracy	0.89
Precision	0.9047619
Recall	0.88888889

	Predicted			
	4x4	Passenger	Truck	Recall
4x4	6	0	0	1
Passenger	0	6	0	1
Truck	1	2	3	0.5
Precision	0.85714286	0.75	1	

# Features	30702
# Clusters	1000
Accuracy	0.83
Precision	0.86904762
Recall	0.83333333

	Predicted			
	4x4	Passenger	Truck	Recall
4x4	6	0	0	1
Passenger	0	5	1	0.8333
Truck	4	0	2	0.3333
Precision	0.6	1	0.6667	

# Features	27849
# Clusters	1000
Accuracy	0.72
Precision	0.75555556
Recall	0.72222222

Average Accuracy	0.81333333
Average Precision	0.84312169
Average Recall	0.81481481

Table 9: Confusion matrices and averaged accuracy, precision, and recall for each random sub-sample testing iteration using the SIFT feature descriptor for the general level categorization.

Feature Detectors with SURF Descriptor

The following table shows the results of classifying the tyre tread images based on various local feature detectors and features described using the SURF descriptor in the test set for each iteration of the random sub-sampling at the general level of categorization.

Harris Detector - SURF Descriptor

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	0	1	5	0.8333
	Precision	1	0.8571429	1	

# Features	16671
# Clusters	1000
Accuracy	0.94
Precision	0.95238
Recall	0.94444

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	2	1	3	0.5
	Precision	0.75	0.8571429	1	

# Features	16641
# Clusters	1000
Accuracy	0.83
Precision	0.86905
Recall	0.83333

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	5	1	0.8333
	Truck	0	2	4	0.6667
	Precision	1	0.7142857	0.8	

# Features	17043
# Clusters	1000
Accuracy	0.83
Precision	0.8381
Recall	0.83333

Average Accuracy	0.86667
Average Precision	0.88651
Average Recall	0.87037

DoG Detector - SURF Descriptor

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	3	2	1	0.1667
	Precision	0.66667	0.75	1	

# Features	34659
# Clusters	1000
Accuracy	0.72
Precision	0.80556
Recall	0.72222

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	1	3	2	0.3333
	Precision	0.85714	0.6666667	1	

# Features	36015
# Clusters	1000
Accuracy	0.78
Precision	0.84127
Recall	0.77778

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	1	3	2	0.3333
	Precision	0.85714	0.6666667	1	

# Features	36045
# Clusters	1000
Accuracy	0.78
Precision	0.84127
Recall	0.77778

Average Accuracy	0.76
Average Precision	0.82937
Average Recall	0.75926

Fast-Hessian Detector - SURF Descriptor

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	2	4	0	0
	Precision	0.75	0.6	0	

# Features	23970
# Clusters	1000
Accuracy	0.67
Precision	0.45
Recall	0.66667

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	0	1	5	0.8333
	Precision	1	0.8571429	1	

# Features	24696
# Clusters	1000
Accuracy	0.94
Precision	0.95238
Recall	0.94444

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	4	1	1	0.1667
	Precision	0.6	0.8571429	1	

# Features	24741
# Clusters	1000
Accuracy	0.72
Precision	0.81905
Recall	0.72222

Average Accuracy	0.77667
Average Precision	0.74048
Average Recall	0.77778

MSER Detector - SURF Descriptor

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	5	1	0.8333
	Truck	3	1	2	0.3333
	Precision	0.66667	0.8333333	0.6667	

# Features	624
# Clusters	624
Accuracy	0.72
Precision	0.72222
Recall	0.72222

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	2	2	2	0.3333
	Precision	0.75	0.75	1	

# Features	603
# Clusters	603
Accuracy	0.78
Precision	0.83333
Recall	0.77778

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	1	4	1	0.6667
	Truck	0	1	5	0.8333
	Precision	0.85714	0.8	0.8333	

# Features	528
# Clusters	528
Accuracy	0.83
Precision	0.83016
Recall	0.83333

Average Accuracy	0.77667
Average Precision	0.79524
Average Recall	0.77778

Harris-Laplace Detector - SURF Descriptor

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	0	5	1	0.1667
	Precision	1	0.5454545	1	

# Features	25329
# Clusters	1000
Accuracy	0.72
Precision	0.84848
Recall	0.72222

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	1	1	4	0.6667
	Precision	0.85714	0.8571429	1	

# Features	28851
# Clusters	1000
Accuracy	0.89
Precision	0.90476
Recall	0.88889

		Predicted			
		4x4	Passenger	Truck	Recall
Known	4x4	6	0	0	1
	Passenger	0	6	0	1
	Truck	3	2	1	0.1667
	Precision	0.66667	0.75	1	

# Features	27522
# Clusters	1000
Accuracy	0.72
Precision	0.80556
Recall	0.72222

Average Accuracy	0.77667
Average Precision	0.85293
Average Recall	0.77778

Table 10: Confusion matrices and averaged accuracy, precision, and recall for each random sub-sample testing iteration using the SURF feature descriptor for the general level categorization.

D Statistical Significance Tests

The following appendix shows the results of statistical significance tests performed on the results obtained for the categorization experiments and the count estimation experiments. The tests on the categorization results were used to determine whether the five feature detectors have significantly different categorization abilities. The first test was to determine whether there is a statistically significant difference between the categorization results over the five detectors when using SURF. The second test was used to determine whether there is a statistically significant difference in categorization over the five detectors when using SIFT. The test were performed on the results obtained for both the specific level and general level of categorizations.

A test was also performed on the results of count estimation to determine whether a statistically significant difference exists between the two feature extraction techniques proposed for count estimation.

D.1 Categorization

```
ONEWAY PrecisionGSIFT RecallGSIFT AccuracyGSURF PrecisionGSURF RecallGSURF
AccuracySSIFT
    PrecisionSSIFT RecallSSIFT AccuracySSURF PrecisionSSURF RecallSSURF AccuracyGSIFT
BY Detector
  /PLOT MEANS
  /MISSING ANALYSIS
  /POSTHOC=BONFERRONI ALPHA(0.05).
```

Oneway

Notes

Output Created		02-MAR-2017 14:51:23
Comments		
Input	Data	C:\Users\Kirstie\Desktop\Spreadsheet17.sav
	Active Dataset	DataSet2
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	16
Missing Value Handling	Definition of Missing	User-defined missing values are treated as missing.
	Cases Used	Statistics for each analysis are based on cases with no missing data for any variable in the analysis.
Syntax		ONEWAY PrecisionGSIFT RecallGSIFT AccuracyGSURF PrecisionGSURF RecallGSURF AccuracySSIFT PrecisionSSIFT RecallSSIFT AccuracySSURF PrecisionSSURF RecallSSURF AccuracyGSIFT BY Detector /PLOT MEANS /MISSING ANALYSIS /POSTHOC=BONFERRONI ALPHA(0.05).
Resources	Processor Time	00:00:05,30
	Elapsed Time	00:00:03,32

ANOVA

		Sum of Squares	df	Mean Square	F	Sig.
PrecisionGSIFT	Between Groups	.179	4	.045	4.909	.019
	Within Groups	.091	10	.009		
	Total	.269	14			
RecallGSIFT	Between Groups	.096	4	.024	4.161	.031
	Within Groups	.058	10	.006		
	Total	.153	14			
AccuracyGSURF	Between Groups	.022	4	.005	.711	.603
	Within Groups	.077	10	.008		
	Total	.099	14			
PrecisionGSURF	Between Groups	.038	4	.009	.601	.670
	Within Groups	.156	10	.016		
	Total	.194	14			
RecallGSURF	Between Groups	.023	4	.006	.750	.580
	Within Groups	.078	10	.008		
	Total	.102	14			
AccuracySSIFT	Between Groups	.006	4	.002	.460	.764
	Within Groups	.034	10	.003		
	Total	.040	14			
PrecisionSSIFT	Between Groups	.007	4	.002	.548	.705
	Within Groups	.030	10	.003		
	Total	.037	14			
RecallSSIFT	Between Groups	.006	4	.002	.457	.766
	Within Groups	.034	10	.003		
	Total	.040	14			
AccuracySSURF	Between Groups	.010	4	.003	.485	.747
	Within Groups	.053	10	.005		
	Total	.063	14			
PrecisionSSURF	Between Groups	.010	4	.003	.397	.806
	Within Groups	.066	10	.007		
	Total	.076	14			
RecallSSURF	Between Groups	.011	4	.003	.487	.745
	Within Groups	.054	10	.005		
	Total	.065	14			
AccuracyGSIFT	Between Groups	.104	4	.026	4.787	.020
	Within Groups	.054	10	.005		
	Total	.158	14			

Post Hoc Tests

Multiple Comparisons

Bonferroni

Dependent Variable	(I) Detector	(J) Detector	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval Lower Bound	95% Confidence Interval Upper Bound
PrecisionGSIFT	1.00000000	2.00000000	-.03968253970	.07785444800	1.000	-.3185109420	.2391458630
		3.00000000	-.05740740740	.07785444800	1.000	-.3362358100	.2214209950
		4.00000000	.22475950000	.07785444800	.162	-.0540689030	.5035879030
		5.00000000	-.06931216930	.07785444800	1.000	-.3481405720	.2095162330
	2.00000000	1.00000000	.03968253970	.07785444800	1.000	-.2391458630	.3185109420
		3.00000000	-.01772486770	.07785444800	1.000	-.2965532710	.2611035350
		4.00000000	.26444203900	.07785444800	.068	-.0143863634	.5432704420
		5.00000000	-.02962962960	.07785444800	1.000	-.3084580320	.2491987730
	3.00000000	1.00000000	.05740740740	.07785444800	1.000	-.2214209950	.3362358100
		2.00000000	.01772486770	.07785444800	1.000	-.2611035350	.2965532710
		4.00000000	.28216690700*	.07785444800	.047	.0033385044	.5609953100
		5.00000000	-.01190476190	.07785444800	1.000	-.2907331650	.2669236410
	4.00000000	1.00000000	-.22475950000	.07785444800	.162	-.5035879030	.0540689030
		2.00000000	-.26444203900	.07785444800	.068	-.5432704420	.0143863634
		3.00000000	-.28216690700*	.07785444800	.047	-.5609953100	-.0033385044
		5.00000000	-.29407166900*	.07785444800	.036	-.5729000720	-.0152432663
	5.00000000	1.00000000	.06931216930	.07785444800	1.000	-.2095162330	.3481405720
		2.00000000	.02962962960	.07785444800	1.000	-.2491987730	.3084580320
		3.00000000	.01190476190	.07785444800	1.000	-.2669236410	.2907331650
		4.00000000	.29407166900*	.07785444800	.036	.0152432663	.5729000720
RecallGSIFT	1.00000000	2.00000000	-.01851851850	.06197481680	1.000	-.2404755120	.2034384750
		3.00000000	.01851851850	.06197481680	1.000	-.2034384750	.2404755120
		4.00000000	.18518518500	.06197481680	.136	-.0367718080	.4071421780
		5.00000000	-.03703703700	.06197481680	1.000	-.2589940300	.1849199560
	2.00000000	1.00000000	.01851851850	.06197481680	1.000	-.2034384750	.2404755120
		3.00000000	.03703703700	.06197481680	1.000	-.1849199560	.2589940300
		4.00000000	.20370370400	.06197481680	.082	-.0182532895	.4256606970
		5.00000000	-.01851851850	.06197481680	1.000	-.2404755120	.2034384750
	3.00000000	1.00000000	-.01851851850	.06197481680	1.000	-.2404755120	.2034384750
		2.00000000	-.03703703700	.06197481680	1.000	-.2589940300	.1849199560
		4.00000000	.16666666700	.06197481680	.227	-.0552903265	.3886236600
		5.00000000	-.05555555560	.06197481680	1.000	-.2775125490	.1664014380
	4.00000000	1.00000000	-.18518518500	.06197481680	.136	-.4071421780	.0367718080
		2.00000000	-.20370370400	.06197481680	.082	-.4256606970	.0182532895

		3.00000000	-.16666666700	.06197481680	.227	-.3886236600	.0552903265
		5.00000000	-.22222222200*	.06197481680	.050	-.4441792150	-.0002652290
	5.00000000	1.00000000	.03703703700	.06197481680	1.000	-.1849199560	.2589940300
		2.00000000	.01851851850	.06197481680	1.000	-.2034384750	.2404755120
		3.00000000	.05555555560	.06197481680	1.000	-.1664014380	.2775125490
		4.00000000	.22222222200*	.06197481680	.050	.0002652290	.4441792150
AccuracyGSURF	1.00000000	2.00000000	.10666666700	.07167829360	1.000	-.1500424190	.3633757520
		3.00000000	.09000000000	.07167829360	1.000	-.1667090850	.3467090850
		4.00000000	.09000000000	.07167829360	1.000	-.1667090850	.3467090850
		5.00000000	.09000000000	.07167829360	1.000	-.1667090850	.3467090850
	2.00000000	1.00000000	-.10666666700	.07167829360	1.000	-.3633757520	.1500424190
		3.00000000	-.01666666670	.07167829360	1.000	-.2733757520	.2400424190
		4.00000000	-.01666666670	.07167829360	1.000	-.2733757520	.2400424190
		5.00000000	-.01666666670	.07167829360	1.000	-.2733757520	.2400424190
	3.00000000	1.00000000	-.09000000000	.07167829360	1.000	-.3467090850	.1667090850
		2.00000000	.01666666670	.07167829360	1.000	-.2400424190	.2733757520
		4.00000000	.00000000000	.07167829360	1.000	-.2567090850	.2567090850
		5.00000000	.00000000000	.07167829360	1.000	-.2567090850	.2567090850
	4.00000000	1.00000000	-.09000000000	.07167829360	1.000	-.3467090850	.1667090850
		2.00000000	.01666666670	.07167829360	1.000	-.2400424190	.2733757520
		3.00000000	.00000000000	.07167829360	1.000	-.2567090850	.2567090850
		5.00000000	.00000000000	.07167829360	1.000	-.2567090850	.2567090850
	5.00000000	1.00000000	-.09000000000	.07167829360	1.000	-.3467090850	.1667090850
		2.00000000	.01666666670	.07167829360	1.000	-.2400424190	.2733757520
		3.00000000	.00000000000	.07167829360	1.000	-.2567090850	.2567090850
		4.00000000	.00000000000	.07167829360	1.000	-.2567090850	.2567090850
PrecisionGSURF	1.00000000	2.00000000	.05714285710	.10206025000	1.000	-.3083763560	.4226620710
		3.00000000	.14603174600	.10206025000	1.000	-.2194874670	.5115509600
		4.00000000	.09126984130	.10206025000	1.000	-.2742493720	.4567890550
		5.00000000	.03357383360	.10206025000	1.000	-.3319453800	.3990930470
	2.00000000	1.00000000	-.05714285710	.10206025000	1.000	-.4226620710	.3083763560
		3.00000000	.08888888890	.10206025000	1.000	-.2766303250	.4544081020
		4.00000000	.03412698410	.10206025000	1.000	-.3313922290	.3996461980
		5.00000000	-.02356902360	.10206025000	1.000	-.3890882370	.3419501900
	3.00000000	1.00000000	-.14603174600	.10206025000	1.000	-.5115509600	.2194874670
		2.00000000	-.08888888890	.10206025000	1.000	-.4544081020	.2766303250
		4.00000000	-.05476190480	.10206025000	1.000	-.4202811180	.3107573090
		5.00000000	-.11245791200	.10206025000	1.000	-.4779771260	.2530613010
	4.00000000	1.00000000	-.09126984130	.10206025000	1.000	-.4567890550	.2742493720
		2.00000000	-.03412698410	.10206025000	1.000	-.3996461980	.3313922290

		3.00000000	.05476190480	.10206025000	1.000	-.3107573090	.4202811180
		5.00000000	-.05769600770	.10206025000	1.000	-.4232152210	.3078232060
	5.00000000	1.00000000	-.03357383360	.10206025000	1.000	-.3990930470	.3319453800
		2.00000000	.02356902360	.10206025000	1.000	-.3419501900	.3890882370
		3.00000000	.11245791200	.10206025000	1.000	-.2530613010	.4779771260
		4.00000000	.05769600770	.10206025000	1.000	-.3078232060	.4232152210
RecallGSURF	1.00000000	2.00000000	.11111111100	.07219847660	1.000	-.1474609610	.3696831830
		3.00000000	.09259259260	.07219847660	1.000	-.1659794790	.3511646650
		4.00000000	.09259259260	.07219847660	1.000	-.1659794790	.3511646650
		5.00000000	.09259259260	.07219847660	1.000	-.1659794790	.3511646650
	2.00000000	1.00000000	-.11111111100	.07219847660	1.000	-.3696831830	.1474609610
		3.00000000	-.01851851850	.07219847660	1.000	-.2770905900	.2400535530
		4.00000000	-.01851851850	.07219847660	1.000	-.2770905900	.2400535530
		5.00000000	-.01851851850	.07219847660	1.000	-.2770905900	.2400535530
	3.00000000	1.00000000	-.09259259260	.07219847660	1.000	-.3511646650	.1659794790
		2.00000000	.01851851850	.07219847660	1.000	-.2400535530	.2770905900
		4.00000000	.00000000000	.07219847660	1.000	-.2585720720	.2585720720
		5.00000000	.00000000000	.07219847660	1.000	-.2585720720	.2585720720
	4.00000000	1.00000000	-.09259259260	.07219847660	1.000	-.3511646650	.1659794790
		2.00000000	.01851851850	.07219847660	1.000	-.2400535530	.2770905900
		3.00000000	.00000000000	.07219847660	1.000	-.2585720720	.2585720720
		5.00000000	.00000000000	.07219847660	1.000	-.2585720720	.2585720720
	5.00000000	1.00000000	-.09259259260	.07219847660	1.000	-.3511646650	.1659794790
		2.00000000	.01851851850	.07219847660	1.000	-.2400535530	.2770905900
		3.00000000	.00000000000	.07219847660	1.000	-.2585720720	.2585720720
		4.00000000	.00000000000	.07219847660	1.000	-.2585720720	.2585720720
AccuracySSIFT	1.00000000	2.00000000	.01333333330	.04774934550	1.000	-.1576764690	.1843431360
		3.00000000	-.02666666670	.04774934550	1.000	-.1976764690	.1443431360
		4.00000000	.01666666670	.04774934550	1.000	-.1543431360	.1876764690
		5.00000000	-.03333333330	.04774934550	1.000	-.2043431360	.1376764690
	2.00000000	1.00000000	-.01333333330	.04774934550	1.000	-.1843431360	.1576764690
		3.00000000	-.04000000000	.04774934550	1.000	-.2110098020	.1310098020
		4.00000000	.00333333333	.04774934550	1.000	-.1676764690	.1743431360
		5.00000000	-.04666666670	.04774934550	1.000	-.2176764690	.1243431360
	3.00000000	1.00000000	.02666666670	.04774934550	1.000	-.1443431360	.1976764690
		2.00000000	.04000000000	.04774934550	1.000	-.1310098020	.2110098020
		4.00000000	.04333333330	.04774934550	1.000	-.1276764690	.2143431360
		5.00000000	-.00666666667	.04774934550	1.000	-.1776764690	.1643431360
	4.00000000	1.00000000	-.01666666670	.04774934550	1.000	-.1876764690	.1543431360
		2.00000000	-.00333333333	.04774934550	1.000	-.1743431360	.1676764690

		3.00000000	-.04333333330	.04774934550	1.000	-.2143431360	.1276764690
		5.00000000	-.05000000000	.04774934550	1.000	-.2210098020	.1210098020
	5.00000000	1.00000000	.03333333330	.04774934550	1.000	-.1376764690	.2043431360
		2.00000000	.04666666670	.04774934550	1.000	-.1243431360	.2176764690
		3.00000000	.00666666667	.04774934550	1.000	-.1643431360	.1776764690
		4.00000000	.05000000000	.04774934550	1.000	-.1210098020	.2210098020
PrecisionSSIFT	1.00000000	2.00000000	.00831863610	.04485775070	1.000	-.1523351910	.1689724630
		3.00000000	-.02716049380	.04485775070	1.000	-.1878143210	.1334933330
		4.00000000	.01807760140	.04485775070	1.000	-.1425762250	.1787314280
		5.00000000	-.03680188120	.04485775070	1.000	-.1974557080	.1238519450
	2.00000000	1.00000000	-.00831863610	.04485775070	1.000	-.1689724630	.1523351910
		3.00000000	-.03547912990	.04485775070	1.000	-.1961329570	.1251746970
		4.00000000	.00975896531	.04485775070	1.000	-.1508948610	.1704127920
		5.00000000	-.04512051730	.04485775070	1.000	-.2057743440	.1155333090
	3.00000000	1.00000000	.02716049380	.04485775070	1.000	-.1334933330	.1878143210
		2.00000000	.03547912990	.04485775070	1.000	-.1251746970	.1961329570
		4.00000000	.04523809520	.04485775070	1.000	-.1154157310	.2058919220
		5.00000000	-.00964138742	.04485775070	1.000	-.1702952140	.1510124390
	4.00000000	1.00000000	-.01807760140	.04485775070	1.000	-.1787314280	.1425762250
		2.00000000	-.00975896531	.04485775070	1.000	-.1704127920	.1508948610
		3.00000000	-.04523809520	.04485775070	1.000	-.2058919220	.1154157310
		5.00000000	-.05487948270	.04485775070	1.000	-.2155333090	.1057743440
	5.00000000	1.00000000	.03680188120	.04485775070	1.000	-.1238519450	.1974557080
		2.00000000	.04512051730	.04485775070	1.000	-.1155333090	.2057743440
		3.00000000	.00964138742	.04485775070	1.000	-.1510124390	.1702952140
		4.00000000	.05487948270	.04485775070	1.000	-.1057743440	.2155333090
RecallSSIFT	1.00000000	2.00000000	.01666666670	.04759895720	1.000	-.1538045340	.1871378670
		3.00000000	-.02469135800	.04759895720	1.000	-.1951625580	.1457798420
		4.00000000	.01790123460	.04759895720	1.000	-.1525699660	.1883724350
		5.00000000	-.03086419750	.04759895720	1.000	-.2013353980	.1396070030
	2.00000000	1.00000000	-.01666666670	.04759895720	1.000	-.1871378670	.1538045340
		3.00000000	-.04135802470	.04759895720	1.000	-.2118292250	.1291131760
		4.00000000	.00123456790	.04759895720	1.000	-.1692366320	.1717057680
		5.00000000	-.04753086420	.04759895720	1.000	-.2180020650	.1229403360
	3.00000000	1.00000000	.02469135800	.04759895720	1.000	-.1457798420	.1951625580
		2.00000000	.04135802470	.04759895720	1.000	-.1291131760	.2118292250
		4.00000000	.04259259260	.04759895720	1.000	-.1278786080	.2130637930
		5.00000000	-.00617283951	.04759895720	1.000	-.1766440400	.1642983610
	4.00000000	1.00000000	-.01790123460	.04759895720	1.000	-.1883724350	.1525699660
		2.00000000	-.00123456790	.04759895720	1.000	-.1717057680	.1692366320

		3.00000000	-.04259259260	.04759895720	1.000	-.2130637930	.1278786080
		5.00000000	-.04876543210	.04759895720	1.000	-.2192366320	.1217057680
	5.00000000	1.00000000	.03086419750	.04759895720	1.000	-.1396070030	.2013353980
		2.00000000	.04753086420	.04759895720	1.000	-.1229403360	.2180020650
		3.00000000	.00617283951	.04759895720	1.000	-.1642983610	.1766440400
		4.00000000	.04876543210	.04759895720	1.000	-.1217057680	.2192366320
AccuracySSURF	1.00000000	2.00000000	.01666666670	.05929212050	1.000	-.1956825010	.2290158350
		3.00000000	.04000000000	.05929212050	1.000	-.1723491680	.2523491680
		4.00000000	.00333333333	.05929212050	1.000	-.2090158350	.2156825010
		5.00000000	.07000000000	.05929212050	1.000	-.1423491680	.2823491680
	2.00000000	1.00000000	-.01666666670	.05929212050	1.000	-.2290158350	.1956825010
		3.00000000	.02333333330	.05929212050	1.000	-.1890158350	.2356825010
		4.00000000	-.01333333330	.05929212050	1.000	-.2256825010	.1990158350
		5.00000000	.05333333330	.05929212050	1.000	-.1590158350	.2656825010
	3.00000000	1.00000000	-.04000000000	.05929212050	1.000	-.2523491680	.1723491680
		2.00000000	-.02333333330	.05929212050	1.000	-.2356825010	.1890158350
		4.00000000	-.03666666670	.05929212050	1.000	-.2490158350	.1756825010
		5.00000000	.03000000000	.05929212050	1.000	-.1823491680	.2423491680
	4.00000000	1.00000000	-.00333333333	.05929212050	1.000	-.2156825010	.2090158350
		2.00000000	.01333333330	.05929212050	1.000	-.1990158350	.2256825010
		3.00000000	.03666666670	.05929212050	1.000	-.1756825010	.2490158350
		5.00000000	.06666666670	.05929212050	1.000	-.1456825010	.2790158350
	5.00000000	1.00000000	-.07000000000	.05929212050	1.000	-.2823491680	.1423491680
		2.00000000	-.05333333330	.05929212050	1.000	-.2656825010	.1590158350
		3.00000000	-.03000000000	.05929212050	1.000	-.2423491680	.1823491680
		4.00000000	-.06666666670	.05929212050	1.000	-.2790158350	.1456825010
PrecisionSSURF	1.00000000	2.00000000	-.07569077010	.06608916380	1.000	-.3123829110	.1610013710
		3.00000000	-.01119929450	.06608916380	1.000	-.2478914360	.2254928470
		4.00000000	-.01706349210	.06608916380	1.000	-.2537556330	.2196286490
		5.00000000	-.02014991180	.06608916380	1.000	-.2568420530	.2165422290
	2.00000000	1.00000000	.07569077010	.06608916380	1.000	-.1610013710	.3123829110
		3.00000000	.06449147560	.06608916380	1.000	-.1722006660	.3011836170
		4.00000000	.05862727810	.06608916380	1.000	-.1780648630	.2953194190
		5.00000000	.05554085830	.06608916380	1.000	-.1811512830	.2922329990
	3.00000000	1.00000000	.01119929450	.06608916380	1.000	-.2254928470	.2478914360
		2.00000000	-.06449147560	.06608916380	1.000	-.3011836170	.1722006660
		4.00000000	-.00586419753	.06608916380	1.000	-.2425563390	.2308279440
		5.00000000	-.00895061728	.06608916380	1.000	-.2456427580	.2277415240
	4.00000000	1.00000000	.01706349210	.06608916380	1.000	-.2196286490	.2537556330
		2.00000000	-.05862727810	.06608916380	1.000	-.2953194190	.1780648630

		3.00000000	.00586419753	.06608916380	1.000	-.2308279440	.2425563390
		5.00000000	-.00308641975	.06608916380	1.000	-.2397785610	.2336057210
	5.00000000	1.00000000	.02014991180	.06608916380	1.000	-.2165422290	.2568420530
		2.00000000	-.05554085830	.06608916380	1.000	-.2922329990	.1811512830
		3.00000000	.00895061728	.06608916380	1.000	-.2277415240	.2456427580
		4.00000000	.00308641975	.06608916380	1.000	-.2336057210	.2397785610
RecallSSURF	1.00000000	2.00000000	.01851851850	.06024767250	1.000	-.1972528700	.2342899070
		3.00000000	.04197530860	.06024767250	1.000	-.1737960790	.2577466970
		4.00000000	.00185185185	.06024767250	1.000	-.2139195360	.2176232400
		5.00000000	.07037037040	.06024767250	1.000	-.1454010180	.2861417580
	2.00000000	1.00000000	-.01851851850	.06024767250	1.000	-.2342899070	.1972528700
		3.00000000	.02345679010	.06024767250	1.000	-.1923145980	.2392281780
		4.00000000	-.01666666670	.06024767250	1.000	-.2324380550	.1991047210
		5.00000000	.05185185190	.06024767250	1.000	-.1639195360	.2676232400
	3.00000000	1.00000000	-.04197530860	.06024767250	1.000	-.2577466970	.1737960790
		2.00000000	-.02345679010	.06024767250	1.000	-.2392281780	.1923145980
		4.00000000	-.04012345680	.06024767250	1.000	-.2558948450	.1756479310
		5.00000000	.02839506170	.06024767250	1.000	-.1873763260	.2441664500
	4.00000000	1.00000000	-.00185185185	.06024767250	1.000	-.2176232400	.2139195360
		2.00000000	.01666666670	.06024767250	1.000	-.1991047210	.2324380550
		3.00000000	.04012345680	.06024767250	1.000	-.1756479310	.2558948450
		5.00000000	.06851851850	.06024767250	1.000	-.1472528700	.2842899070
	5.00000000	1.00000000	-.07037037040	.06024767250	1.000	-.2861417580	.1454010180
		2.00000000	-.05185185190	.06024767250	1.000	-.2676232400	.1639195360
		3.00000000	-.02839506170	.06024767250	1.000	-.2441664500	.1873763260
		4.00000000	-.06851851850	.06024767250	1.000	-.2842899070	.1472528700
AccuracyGSIFT	1.00000000	2.00000000	.01667	.06018	1.000	-.1989	.2322
		3.00000000	.05667	.06018	1.000	-.1589	.2722
		4.00000000	.22000*	.06018	.044	.0045	.4355
		5.00000000	.00000	.06018	1.000	-.2155	.2155
	2.00000000	1.00000000	-.01667	.06018	1.000	-.2322	.1989
		3.00000000	.04000	.06018	1.000	-.1755	.2555
		4.00000000	.20333	.06018	.070	-.0122	.4189
		5.00000000	-.01667	.06018	1.000	-.2322	.1989
	3.00000000	1.00000000	-.05667	.06018	1.000	-.2722	.1589
		2.00000000	-.04000	.06018	1.000	-.2555	.1755
		4.00000000	.16333	.06018	.218	-.0522	.3789
		5.00000000	-.05667	.06018	1.000	-.2722	.1589
	4.00000000	1.00000000	-.22000*	.06018	.044	-.4355	-.0045
		2.00000000	-.20333	.06018	.070	-.4189	.0122

	3.00000000	-.16333	.06018	.218	-.3789	.0522
	5.00000000	-.22000*	.06018	.044	-.4355	-.0045
5.00000000	1.00000000	.00000	.06018	1.000	-.2155	.2155
	2.00000000	.01667	.06018	1.000	-.1989	.2322
	3.00000000	.05667	.06018	1.000	-.1589	.2722
	4.00000000	.22000*	.06018	.044	.0045	.4355

*. The mean difference is significant at the 0.05 level.

D.2 Count Estimation

```
T-TEST GROUPS=FeatureDetector(1 2)
  /MISSING=ANALYSIS
  /VARIABLES=Variance MSE R
  /CRITERIA=CI(.95).
```

T-Test

Notes

Output Created		02-MAR-2017 15:07:31
Comments		
Input	Data	….
	Active Dataset	DataSet3
	Filter	<none>
	Weight	<none>
	Split File	<none>
	N of Rows in Working Data File	54
Missing Value Handling	Definition of Missing	User defined missing values are treated as missing.
	Cases Used	Statistics for each analysis are based on the cases with no missing or out-of-range data for any variable in the analysis.
Syntax		T-TEST GROUPS=FeatureDetector(1 2) /MISSING=ANALYSIS /VARIABLES=Variance MSE R /CRITERIA=CI(.95).
Resources	Processor Time	00:00:00,00
	Elapsed Time	00:00:00,02

Group Statistics

	FeatureDetector	N	Mean	Std. Deviation	Std. Error Mean
Variance	1.00000000	27	7.3920358300	7.64272078600	1.47084230100
	2.00000000	27	15.7519277800	16.760137420000	3.22548994900
MSE	1.00000000	27	13.9305919400	13.628491230000	2.62280436000
	2.00000000	27	28.2835767200	30.447602150000	5.85964376500
R	1.00000000	27	.9533848150	.04366429900	.00840319827

	2.00000000	27	.9075714810	.09513008260	.01830779290

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means	
		F	Sig.	t	df
Variance	Equal variances assumed	18.177	.000	-2.358	52
	Equal variances not assumed			-2.358	36.365
MSE	Equal variances assumed	21.633	.000	-2.236	52
	Equal variances not assumed			-2.236	36.016
R	Equal variances assumed	21.378	.000	2.274	52
	Equal variances not assumed			2.274	36.490

Independent Samples Test

		t-test for Equality of Means		
		Sig. (2-tailed)	Mean Difference	Std. Error Difference
Variance	Equal variances assumed	.022	-8.35989195000	3.54501939200
	Equal variances not assumed	.024	-8.35989195000	3.54501939200
MSE	Equal variances assumed	.030	-14.35298478000	6.41985418500
	Equal variances not assumed	.032	-14.35298478000	6.41985418500
R	Equal variances assumed	.027	.04581333330	.02014420570
	Equal variances not assumed	.029	.04581333330	.02014420570

Independent Samples Test

		t-test for Equality of Means	
		95% Confidence Interval of the Difference	
		Lower	Upper
Variance	Equal variances assumed	-15.47349379000	-1.24629011400
	Equal variances not assumed	-15.54702116000	-1.17276274200
MSE	Equal variances assumed	-27.23536467000	-1.47060488500
	Equal variances not assumed	-27.37284981000	-1.33311974400
R	Equal variances assumed	.00539102733	.08623563930
	Equal variances not assumed	.00497801669	.08664865000

Image Analysis of Waste Tyre Stockpiles (Work-In-Progress Paper)

Grant Eastwood
Computing Sciences
Nelson Mandela Metropolitan University
Port Elizabeth, South Africa
Grant.Eastwood@nmmu.ac.za

Charmain Cilliers
Computing Sciences
Nelson Mandela Metropolitan University
Port Elizabeth, South Africa
Charmain.Cilliers@nmmu.ac.za

Kevin Naudé
Computing Sciences
Nelson Mandela Metropolitan University
Port Elizabeth, South Africa
Kevin.Naude@nmmu.ac.za

ABSTRACT

Pressures from environmental agencies contribute to the challenges associated with the disposal of waste tyres, particularly in South Africa. Recycling of waste tyres in South Africa is in its infancy resulting in the historically undocumented and uncontrolled existence of waste tyre stockpiles across the country. The remote and distant locations of such stockpiles typically complicate the logistics associated with the collection, transport and storage of waste tyres prior to entering the recycling process.

In order to optimize the logistics associated with the collection of waste tyres from stockpiles, useful information about such stockpiles would include estimates of the types of tyres as well as the quantity of specific tyre types found in particular stockpiles. This research proposes the use of image analysis for quantifying the tyre distributions of such stockpiles to support the logistics in tyre recycling efforts.

This paper presents a discussion of the problem context and a discussion of image analysis techniques and algorithms applied to sample images of waste tyre stockpiles. Initial findings identify a number of challenges emerging as a result of the application of some known image analysis techniques being applied to such images.

Categories and Subject Descriptors

I.4.9: Image Processing and Computer Vision Applications

General Terms

Algorithms, Experiment

Keywords

Image analysis, tyres, waste tyres, stockpiles, algorithms, computer vision

1. Introduction

Currently an estimated 60-100 million waste tyres are stockpiled in historical waste tyre stockpiles across South Africa (RSA) with an estimated increase of 10 million waste tyres per year [20, 21]. Large tyre stockpiles pose several risks to the environment. For example, stockpiles represent a fire hazard which can lead to air and soil pollution. Tyre stockpiles also pose serious health risks in warmer climates where mosquito-borne diseases such as encephalitis and dengue fever occur [16].

Knowledge of the composition of particular waste tyre stockpiles has the potential to assist in the strategic and optimal collection of waste tyres for entry into the recycling process. The composition and volume of waste tyres in stockpiles can potentially be estimated by means of the analysis of images of the stockpiles.

This research seeks to investigate mining images of waste tyre stockpiles for information that may enable effective recycling and disposal activities. The paper focuses upon the overall goals, and describes possible methods for applying image analysis techniques to the stated problem.

The aim of this paper is to discuss a subset of image analysis techniques and algorithms that have been applied to waste tyre stockpile images in order to determine their suitability in supporting the task of analyzing waste tyre stockpile images to extract information to support tyre collection planning activities. The extracted information includes the number of tyres, types of tyres, and the ratios of various types of tyres in a waste tyre stockpile. The waste tyre stockpile image analysis problem is presented and decomposed into four sub-problems (Section 2). Relevant features of image analysis are the pre-processing of images, object recognition, and peripheral issues novel to the analysis of images within the context of waste tyre stockpile image analysis (Section 3). The experimental procedure emphasizes the relevance of a structured approach to the investigation (Section 4). Preliminary findings on the application of selected image analysis techniques and algorithms on waste tyre stockpiles (Section 5) indicate the existence of numerous challenges as well as lay the foundation for further experimental work (Section 6).

2. Problem Decomposition

In order to provide an understanding of the problem of extracting information about waste tyre stockpiles from images, the problem is decomposed as follows:

1. Recognizing individual tyres that are partially occluded,
2. Counting and classifying individual tyres that appear in an image containing multiple tyres,
3. Estimating the 3D shape and volume of the stockpile, and
4. Estimating the overall tyre distribution of the stockpile.

2.1 Recognizing individual tyres that are partially occluded

Tyre stockpiles that do not follow an orderly arrangement will typically contain tyres that are partially occluded in a number of different ways in the acquired image. In addition to the partial occlusion problem, the appearance of individual tyres can vary with regard to their individual rotation and sizes which may also be impacted by the viewpoint of the image acquisition device. In

determining suitable methods to recognize tyres, consideration must be given to the level of invariance of the recognition methods with regards to rotation, scale, viewpoint, and partial occlusions.

2.2 Counting and classifying individual tyres that appear in an image containing multiple tyres

Through the recognition of individual tyres, an estimate of the overall number of tyres visible in the image can be obtained by simply counting the number of recognized tyres. In order to count the number of tyres in each class of tyre to be recognized, a classification method that minimizes intra-class variation and maximizes inter-class variation enough to distinguish between different tyre classes is required. The different tyre classes can be characterized by their size and tread complexity. The characteristics defining tyre classes present several issues. The issues include having multiple tyres from different classes appearing in the image with a high degree of similarity due to intrinsic characteristics of tyres such as their circular shape as well as colour similarities. Size can be difficult to determine without additional user input due to varying distances of the image acquisition device and the actual shape of the stockpile. The issue concerning the use of tread as a defining characteristic for tyre class is that the appearance of tyre tread can vary due to distance from the viewpoint and the quality of the acquired image.

2.3 Estimating the 3D shape and volume of the stockpile

Estimating the 3D shape and volume of a waste tyre stockpile could be accomplished by allowing user interaction with the system. For example a user could annotate an image with specific dimensions and a shape around the stockpile and/or a set of typical arrangements could be provided for the user to select the one most closely representing the stockpile in an image.

2.4 Estimating the overall tyre distribution of the stockpile

To estimate the overall tyre distribution a count could be obtained of the tyres in each class to acquire a frequency distribution. If for example it is assumed that the tyres are uniformly distributed throughout the entire stockpile, the overall tyre distribution could be acquired through multiplying the estimated surface area by some depth to acquire a volume estimate. The volume estimate to total volume ratio could then be multiplied by the count of tyres in each class to estimate the overall distribution of the stockpile.

3. Relevant Literature

Image analysis is the process of applying a multiple techniques and algorithms to an image in order to extract useful information. A large number of techniques and algorithms for image analysis exist [22]. The following literature review section discusses techniques and algorithms, in their respective topics, that have been investigated thus far in the work. These topic categories include image pre-processing and object recognition.

3.1 Pre-processing

The objective of image pre-processing is to correct defects in digital images and to prepare and possibly simplify images to be used as input for another operation. Some of the most commonly used image pre-processing techniques include noise reduction, thresholding, line and edge detection, and segmentation [29]. For example, Images for banknote recognition [24] were pre-processed by using Guassian blurring for noise reduction and a Canny Edge detector [6] to extract edges from an image. To further correct defects in the image, a closing operator was applied (dilation followed by erosion) in order to create a set of closed contours in the edge image. Numerous papers about image analysis use edge detectors in image pre-processing [4, 9].

Image segmentation is another technique used in image pre-processing. Image segmentation is defined as the process of partitioning an input image into homogenous regions based on some property [29]. Image segmentation forms a large part of image analysis, particularly object detection. Many image segmentation algorithms can be found in the literature [10, 13, 23, 26].

The circular shape characteristic of all tyres indicates that shape-based detection should be appropriate for individual tyre recognition while tyre classification relies on the size and tread complexity characteristics which may require more detailed input images for feature-based detection and classification methods. The different methods will therefore require images to be pre-processed in different ways. For example an input image for shape-based detection could be pre-processed to remove unnecessary detail such as the tread detail while maintaining the overall tyre shape. Such a simplification could for example be accomplished by using segmentation and/or smoothing operations before applying an edge detection algorithm.

3.2 Object Detection and Recognition

Object detection techniques can be categorised as belonging to one of three categories. These categories are shape (Section 3.2.1), feature (Section 3.2.2), and graph (3.2.3) –based detection methods. The algorithms in each category may have overlapping processes although these processes are used in different ways to achieve their required outcomes of correctly recognizing and classifying individual tyres. Object detection and recognition techniques are investigated in order to address the recognition and classification of individual tyres in a waste tyre stockpile as discussed in Sections 2.1 and 2.2.

3.2.1 Shape-based Detection

Shape-based object detection considers the shape of the object to be recognized as a whole. A shape in this context can be described as a set of contours describing an object boundary [23]. The shape of an object is therefore considered a global feature or property of an object. Shape-based detection is typically performed using point sets or mathematical models describing the shape to be detected along with a matching strategy.

One of the most well-known algorithms for shape detection is the Hough Transform which has many variations and uses that can be found in the literature [1, 3, 4, 7–9, 17, 18, 23, 25, 28]. This algorithm uses a voting system, with an accumulator designed on the mathematical representation of a particular shape, to decide whether a given shape is present in an image. The Hough Transform is most commonly used for the detection of lines, circles, and ellipses. A generalized Hough Transform exists which can be used to identify arbitrary shapes. The generalized Hough Transform could be useful in the tyre recognition context in that the shapes produced by 3D tyres projected onto a 2D image could be identified using a generalized Hough Transform approach. Another example of shape detection is creating common class boundary models from training images and then matching the

common class boundaries to the contours of an object in an edge image [9, 23].

Since tyres are circular objects in a 3D space, when they are rotated around any axis, the shape when projected onto a 2D plane could be seen as a set of ellipses, which indicates that ellipse and circle shape detection algorithms could potentially be useful. A few examples of ellipse detection techniques and algorithms are found in [11, 17, 28]. A downfall of using circle or ellipse detection to detect tyres in a disorderly arrangement is that few of the tyre edges extracted from an image appear elliptical. The tread of the tyre that occludes the part of the tyre facing away from the image acquisition device causes the extracted contours to represent part of the ellipse formed by the circle of a rotated tyre although at some point the curvature changes from what is expected from an ellipse due to the occlusion resulting from the tyre itself. An example of this change can be seen in Figure 1a annotated in red.

3.2.2 Feature-based Detection

A feature is any information contained in an image that can be used to support the object recognition task. Features can be classified as global or local features. Global features describe the object as a whole and refer to global object properties such as area, perimeter, and/or circularity. Local features are features that describe only part of the area covered by the object [25].

Three identified algorithms for feature-based detection are the Scale Invariant Feature Transform (SIFT) [15], Speeded Up Robust Features (SURF) [5], and the Haar Classifier [27].

The SIFT algorithm generates feature descriptors that are invariant to image scale and rotation as well as being partially invariant to changes in illumination and 3D camera viewpoint. A database of feature descriptors is generated by the algorithm that can be used for object detection and image matching [15].

The SURF algorithm is similar to the SIFT algorithm with a few changes. This algorithm improves performance time and robustness over existing feature descriptors and detectors such as the SIFT algorithm [5].

The SURF algorithm uses a box filter to approximate the Laplacian of Guassian (LOG). By using a box filter, convolution can be computed efficiently with the help of image integrals. The SURF descriptor makes use of the Hessian matrix for scale and location invariance and uses wavelet responses for the orientation [5, 12].

The Haar Classifier is a machine learning approach, which uses a cascaded neural network to learn features of an object in an image. This algorithm uses integral images when training the classifier. The learning algorithm is based on Adaboost, which allows critical features to be selected from a larger set of features. The Haar classifier is described as "an object specific focus-of-attention algorithm" [27].

Feature-based detection methods could potentially be used in solving the classification problem. Once regions of an image containing tyres have been found, by a method more suitable for the detection of simple objects, feature-based methods could prove useful in classifying tyres within the isolated regions based on features obtained from model tyre tread and the recognized tyre's tread.

3.2.3 Graph-based Detection

Graph-based object detection is the process of recognizing an object in an image by making use of graphs. Each node in the graph represents a feature or set of features that describe an image region. Each edge in the graph connects two nodes and contains some description of the geometric relationship between the connected nodes. A typical matching step in graph-based detection involves matching an unknown object graph from the scene image to each of the graphs constructed from the known objects. The unknown object can then be classified according to the best match providing the best match is above some threshold [14, 25].

Graph-based image matching techniques can be constructed in such a way as to allow scale invariance. Scale invariance is achieved for example by defining lengths or regions relative to one another [2, 14, 19]. One way in which rotation invariance can be achieved is to set all images to a standard orientation [19]. Graph-based matching is also invariant to partial occlusion if the object is visible enough for enough of its graph representation to be matched [25].

Graph-based object detection appears to provide a number of suitable candidate methods for the tyre recognition and classification problem. The potential for graph-based methods to be designed in such a way as to allow some degree of scale, rotation, and translation invariance, as well as being invariant to partial occlusion, indicates that graph-based detection methods could play an essential role in the recognition and/or classification of waste tyres in a waste tyre stockpile image.

4. Experimental Procedure

This research is concerned with experimentally investigating image analysis techniques and algorithms in order to extract information about a waste tyre stockpile. The process of extracting the information can be broken down into three phases: The first phase to identify individual tyres in a waste tyre stockpile based on a general tyre model. In the second phase, an attempt is made to recognize each tyre based on a more specific tyre model. The third phase is the classification phase, where the output from the first and second phases can be used to classify individual tyres.

The research objectives are as follows:

- O1. To investigate image analysis techniques and algorithms to recognize, classify, and count tyres in waste tyre stockpile images.
- O2. To determine the level of user interaction that is required to estimate a waste tyre stockpile's 3D shape, volume, and overall tyre distribution.
- O3. To implement a system to extract the required information about a waste tyre stockpile.
- O4. To evaluate the accuracy and efficiency of the implemented system.

The experimental nature of this research allows research objectives one and two to be considered simultaneously. The algorithms and techniques that can potentially be implemented and tested will provide feedback about the suitability of the method in the context of this research domain. Furthermore, the level of user input for the system should also become evident. The implemented system will be based on the results of the first two research objectives.

The measures to be used to evaluate the system include accuracy and efficiency as these are two metrics common to many object recognition systems [19]. The system should be evaluated for how accurately and efficiently it can perform the required operations. The accuracy of the system will be determined by the counting, classification, and estimations errors made during the analysis of waste tyre stockpile images of stockpiles with known tyre distributions. The efficiency will be measured by the amount of time taken for the system to perform the required operations.

5. Discussion

At present, pre-processing of waste tyres has been conducted. The images have been pre-processed by first applying a Guassian blur, followed by a Canny Edge Detector. A binary image is then created using binary thresholding. Circle and ellipse detection is then applied to the binary edge image. One could intuitively picture a tyre at varying angles and observe one or more elliptical shapes unless viewed from the front, in which case it may be represented by a rectangle with rounded corners. This observation has led to shape-based features being investigated, more specifically; circle and ellipse detection, and they are still currently being investigated.

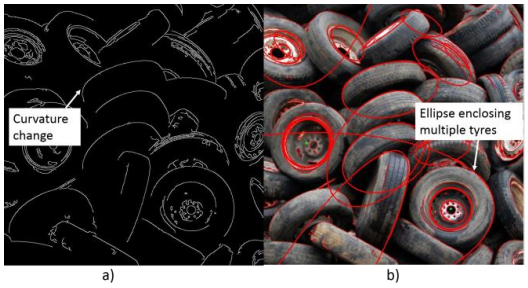


Figure 1 - Pre-processed edge image (a). Detected ellipses and circles drawn on original image (b).

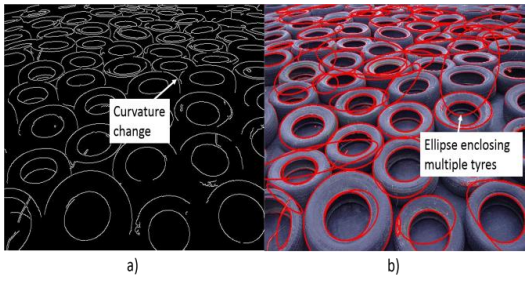


Figure 2 - Arranged tyres edge image (a). Detected circles and ellipses drawn on original image (b).

Figure 1 and Figure 2 show the results of shape matching algorithms that match ellipses and circles when they are applied to the respective edge images. Figure 1a and Figure 2a are annotated showing a change in curvature differing from what is expected from the ellipses as discussed in Section 3.2.1. Figure 1b and Figure 2b show that some ellipses are enclosing multiple tyres. The enclosing of multiple tyres could be due to multiple tyre edges contributing points to a larger ellipse. Another reason for incorrect ellipses and circles being identified is noise, indicating that further pre-processing may be required in order to eliminate edges that do not contribute to the identification of circles or ellipses. A possible solution could be user interaction in which the user segments the image into regions in the image and specifies the minimum and maximum distance covered by an individual tyre in each region.

Figure 1 shows a disorderly arrangement of tyres. In this case, far fewer desired detections were made, because of the high level of occlusion, which creates a problem. When the number of points required to match an ellipse is increased, the chance of partially occluded tyres being missed also increases. When the number of points required to match an ellipse decreases, many more incorrect ellipses are found.

Although it is not directly evident in Figure 1 and Figure 2, another contributing factor to the detection of unwanted ellipses/circles is uneven illumination. Since waste tyre stockpiles are typically located outdoors, the illumination conditions vary significantly between locations and/or over time. Illumination artifacts can result in unwanted edges being extracted and as a result unwanted ellipses/circles are found.

From these short experiments it can be seen that circle ellipse detection may provide initial information that could potentially be used as input for more complex approaches. For example, the presence of a circle or ellipse at a particular location could provide an indication of locations where further matching techniques, such as the techniques mentioned in Section 3.2, should focus there processes although shape matching techniques that are better suited to the detection of arbitrary shaped objects may be required in order to cater for the multiple rotation angles of tyres in waste tyre stockpiles..

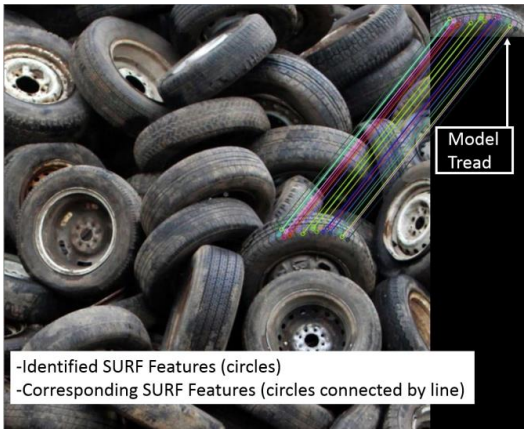


Figure 3 - SURF Feature Matching between image of stockpile containing multiple tyres and single tyre example manually extracted from the image.

Figure 3 and 4 show the location of located SURF features in both images and a line is drawn between the best matching features to indicate correspondences. The model images are shown on the right side of both Figure 3 and Figure 4 shown in the yellow rectangles. The lines between the identified features on the model image and the scene image indicate that the two features connected by the line were found to be a good match.

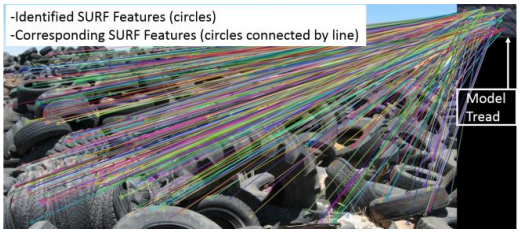


Figure 4 – SURF Feature Matching between image of stockpile containing multiple tyres and tyre tread extracted from a different image.

From the above Figures showing the SURF features it can be seen that in order to use local region descriptors such as the SIFT or SURF features for matching in an image containing multiple instances of similar objects, regions of the image would first need to be isolated using some other technique before SURF or SIFT methods can be applied. Shape-based techniques could prove to be important in isolating regions of the image from which SIFT or SURF features can be extracted for matching to some model.

6. Conclusions and Future Work

The main issues identified include detection of similar highly occluded objects, changes in illumination of an outdoor scene, detecting objects that have similar shapes and colour, and the lack of orderly arrangements in waste tyre stockpiles.

The future work for this research includes a further investigation into the creation of a general tyre model for high-level tyre identification which will involve shape-based detection approaches. The next step will include an investigation into using more specific, feature-based, techniques to identify different types of tyres in the stockpile and classify them accordingly. For the shape, feature, and graph -based detection, algorithms that require training or learning will be investigated.

Statistical models should also be investigated so that estimations that are made can be made within a confidence interval. These statistics should be drawn from annotated training data and previously analyzed images. Through the investigations, areas for user interaction with the system may be identified and incorporated into the design of the prototype.

7. REFERENCES

[1] Amit, Y. and Felzenszwalb, P. 2013. Object Detection. (2013).

[2] Baeza-Yates, R. and Valiente, G. 2000. An image similarity measure based on graph matching. *String Processing and Information …*. (2000).

[3] Ballard, D. 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*. (1981).

[4] Barinova, O. et al. 2012. On detection of multiple object instances using Hough transforms. *IEEE transactions on pattern analysis and machine intelligence*. 34, 9 (Sep. 2012), 1773–84.

[5] Bay, H. et al. 2006. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*. (2006).

[6] Canny, J. 1986. A computational approach to edge detection. *IEEE transactions on pattern analysis and machine intelligence*. 8, 6 (Jun. 1986), 679–98.

[7] Collet, A. et al. 2009. Object recognition and full pose registration from a single image for robotic manipulation. *2009 IEEE International Conference on Robotics and Automation*. (May 2009), 48–55.

[8] Dunlop, M. et al. 2005. Design and development of Taeneb City Guide - From paper maps and guidebooks to electronic guides. (2005).

[9] Ferrari, V. et al. 2009. From Images to Shape Models for Object Detection. *International Journal of Computer Vision*. 87, 3 (Jul. 2009), 284–303.

[10] Flohr, F. and Gavrila, D. 2013. PedCut: an iterative framework for pedestrian segmentation combining shape models and multiple data cues. *Procedings of the British Machine Vision Conference 2013*. (2013), 66.1–66.11.

[11] Ho, C. and Chen, L. 1995. A fast ellipse/circle detector using geometric symmetry. *Pattern Recognition*. (1995).

[12] Introduction to SURF (Speeded-Up Robust Features): 2011. *http://docs.opencv.org/trunk/doc/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html*. Accessed: 2014-04-16.

[13] Johnson, A.E. 1998. Using Spin-Images for Efficient Object Recognition in Cluttered 3-D Scenes. July (1998).

[14] Lee, W. et al. 2006. An Efficient Graph-Based Symbol Recognizer. (2006).

[15] Lowe, D.G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. 60, 2 (Nov. 2004), 91–110.

[16] Lula, J.W. and Bohnert, G.W. 1997. Scrap tire recycling. (Mar. 1997).

[17] McLaughlin, R. 1997. Technical report-randomized hough transform: Improved ellipse detection with comparison. *University of Western Australia*. (1997).

[18] Murphy, K. et al. 2006. Object detection and localization using local and global features. *... Category-Level Object ...*. (2006), 1–20.

[19] Petrakis, E.G.M. 2002. Design and evaluation of spatial similarity approaches for image retrieval. *Image and Vision Computing*. 20, 1 (Jan. 2002), 59–76.

[20] REDISA 2012. Government Gazette Staatskoerant. November (2012), 1–56.

[21] SA eyes recycling plan for tyres: 2013. *http://www.iol.co.za/scitech/science/environment/sa-eyes-recycling-plan-for-tyres-1.1522860#.UyKlkT-SxIE*. Accessed: 2014-03-14.

[22] Sebe, N. and Lew, M.S. 2003. *Robust Computer Vision: Theory and Applications*. Kluwer Academic Dordrecht.

[23] Toshev, A. et al. 2012. Shape-based object detection via boundary structure segmentation. *International journal of computer vision*. (2012).

[24] Toytman, I. and Thambidurai, J. 2011. Banknote recognition on Android platform. *unpublished. Available at: http://www. stanford. ...*. (2011).

[25] Treiber, M. 2010. *An Introduction to Object Recognition: Selected algorithms for a wide variety of applications*. Springer.

[26] Tu, Z. et al. 2005. Image Parsing: Unifying Segmentation, Detection, and Recognition. *International Journal of Computer Vision*. 63, 2 (Feb. 2005), 113–140.

[27] Viola, P. and Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. 1, (2001), I–511–I–518.

[28] Xie, Y. 2002. A new efficient ellipse detection method. 00, c (2002), 0–3.

[29] Zhou, H. et al. 2010. *Digital Image Processing: Part II*.

List of References

Amit, Yali. 2002. *2D Object Detection and Recognition.* The MIT Press.

Arteta, Carlos, & Lempitsky, Victor. 2012. Learning to detect cells using non-overlapping extremal regions. *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, 1–8.

Arteta, Carlos, & Lempitsky, Victor. 2013. Learning to Detect Partially Overlapping Instances. *Conference: Computer Vision and Pattern Recognition (CVPR).*

Arteta, Carlos, & Lempitsky, Victor. 2014. Interactive Object Counting. *Computer Vision - ECCV 2014*, 1–15.

Azad, Pedram, Asfour, Tamim, & Dillmann, Rudiger. 2009. Combining Harris interest points and the SIFT descriptor for fast scale-invariant object recognition. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, oct, 4275–4280.

Bay, Herbert, Tuytelaars, Tinne, & Gool, Luc Van. 2006. Surf: Speeded up robust features. *Pages 404—-417 of: Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I.* Springer Berlin Heidelberg.

Beale, Mark H., Hagan, Martin T., & Demuth, Howard B. 2014. *Neural Network Toolbox ^{TM} User ' s Guide.* The Mathworks , Inc.

Beis, Jeffrey S., & Lowe, David G. 1997. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. *Pages 1000–1006 of: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*

Boser, Bernhard E., Guyon, Isabelle M., & Vapnik, Vladimir N. 1992. A Training Algorithm for Optimal Margin Classifiers. *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, 144–152.

Bradski, Gary, & Kaehler, Adrian. 2008. *Learning OpenCV*. O'Reilly Media, Inc.

Brahmbhatt, & Samarth. 2012. *Practical OpenCV*. Apress.

Cadik, Martin. 2008. Perceptual Evaluation of Color-to-Grayscale Image Conversions. *Computer Graphics Forum*, **27**(7), 1745–1754.

Canny, John. 1986. A computational approach to edge detection. *IEEE transactions on pattern analysis and machine intelligence*, **8**(6), 679–698.

Cavanagh, Patrick. 1999. Top-Down Processing in Vision. *MIT Encyclopedia of Cognitive Science*, 844–845.

Chan, Antoni B, & Vasconcelos, Nuno. 2009. Bayesian Poisson regression for crowd counting. *2009 IEEE 12th International Conference on Computer Vision*, sep, 545–551.

Chan, Antoni B, Liang, Zhang-Sheng J., & Vasconcelos, Nuno. 2008. Privacy preserving crowd monitoring: Counting people without people models or tracking. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, jun, 1–7.

Chan, Tony, & Vese, Luminita. 1999. An Active Contour Model without Edges. *Scale-Space'99*, **LNCS 1682**, 141–151.

Chang, Pei Chann, Wang, Di Di, & Zhou, Chang Le. 2012. A novel model by evolving partially connected neural network for stock price trend forecasting. *Expert Systems with Applications*, **39**(1), 611–620.

Chen, Jie, Zou, Li-hui, Zhang, Juan, & Dou, Li-hua. 2009. The Comparison and Application of Corner Detection Algorithms. *Journal of Multimedia*, **4**(6), 435–441.

Cherkassky, Vladimir. 1997. The Nature Of Statistical Learning Theory. *IEEE Transactions on Neural Networks*, **8**(6).

Crandall, David, Felzenszwalb, Pedro, & Huttenlocher, Daniel. 2005. Spatial priors for part-based recognition using statistical models. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, **1**, 10–17.

Csurka, Gabriella, Dance, Christopher R., Fan, Lixin, Willamowski, Jutta, & Bray, Cedric. 2004. Visual categorization with bags of keypoints. *Pages 1–22 of: Workshop on statistical learning in computer vision, ECCV*.

Dalal, Navneet, & Triggs, Bill. 2005. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005.*

Dawson-howe, Kenneth. 2014. *A Practical Introduction to Computer Vision with OpenCV*. Wiley Publishing, Inc.

De Oliveira, Fagner A., Nobre, Cristiane N., & Zárate, Luis E. 2013. Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index - Case study of PETR4, Petrobras, Brazil. *Expert Systems with Applications*, **40**(18), 7596–7606.

Derpanis, Konstantinos G. 2005. Relationship Between the Sum of Squared Difference (SSD) and Cross Correlation for Template Matching. *York University*, **2**(4), 4.

Dietterich, Thomas G., & Bakiri, Ghulum. 1994. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Jouranal of artifical intelligence research*, **2**, 263–286.

Engelbrecht, A P. 2007. *Computational intelligence: an introduction*. 2nd edn. Wiley Publishing, Inc.

Felzenszwalb, Pedro, McAllester, David, & Ramanan, Deva. 2008. A discriminatively trained, multiscale, deformable part model. *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 1–8.

Felzenszwalb, Pedro F., Girshick, Ross B, McAllester, David, & Ramanan, Deva. 2010. Object Detection with Discriminative Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(9), 1627–1645.

Fergus, Rob, Perona, Pietro, & Zisserman, Andrew. 2003. Object class recognition by unsupervised scale-invariant learning. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**, 264–271.

Ferrari, Vittorio, Fevrier, Loic, Jurie, Frederic, & Schmid, Cordelia. 2008. Groups of adjacent contour segments for object detection. *IEEE transactions on pattern analysis and machine intelligence*, **30**(1), 36–51.

Fischler, Martin A., & Elschlager, R.a. 1973. The Representation and Matching of Pictorial Structures. *IEEE Transactions on Computers*, **C-22**(1), 67–92.

Flohr, Fabian, & Gavrila, Dariu. 2013. PedCut: an iterative framework for pedestrian segmentation combining shape models and multiple data cues. *Procedings of the British Machine Vision Conference 2013*, 66.1—-66.11.

Frery, Alejandro C., & Perciano, Talita. 2013. *Introduction to Image Processing Using R*. Springer London.

Gallo, Ignazio, & Nodari, Angelo. 2011. Learning Object Detection using Multiple Neural Networks. *In: VISAPP 2011. INSTICC*.

Gilles, S. 1998. *Robust Description and Matching of Images*. Ph.D. thesis, University of Oxford.

Gonzalez, Rafael C., & Woods, Richard C. 2002. *Digital image processing*. Pearson Education, Inc.

Grauman, Kristen, & Leibe, Bastian. 2010. Visual Object Recognition. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 1–159.

Gregory, Richard L. 1997. Knowledge in perception and illusion. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, **352**(1358), 1121–1127.

Gupta, Ankur. 2011. *Using line and ellipse features for rectification of broadcast hockey video*. Ph.D. thesis, The University Of British Columbia.

Guresen, Erkam, Kayakutlu, Gulgun, & Daim, Tugrul U. 2011. Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, **38**(8), 10389–10397.

Hagan, Martin T., Demuth, Howard B., Beale, Mark H., & Jesus, De Orlando. 1995. *Neural Network Design*. Boston: PWS Publishing Company.

Hamza, Abdessamad Ben, Luque-Escamilla, Pedro L., Martínez-Aroza, José, & Román-Roldán, Ramón. 1999. Removing noise and preserving details with relaxed median filters. *Journal of Mathematical Imaging and Vision*, **11**(2), 161–177.

Han, Jiawei, Kamber, Micheline, & Pei, Jian. 2011. *Data Mining: Concepts and Techniques*. 3rd edn. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Harris, Chris, & Stephens, Mike. 1988. A Combined Corner and Edge Detector. *Procedings of the Alvey Vision Conference 1988*, 147–151.

Huang, Deng-Yuang, Hu, Wu-Chi h, Wang, Ying-Wei, Chen, Ching-I, & Cheng, Chih-Hsiang. 2010. Recognition of tire tread patterns based on Gabor wavelets and support vector machine. *Pages 92–101 of: Computational Collective Intelligence. Technologies and Applications: Second International Conference, ICCCI.*

Idrees, Haroon, Saleemi, Imran, Seibert, Cody, & Shah, Mubarak. 2013. Multi-source Multi-scale Counting in Extremely Dense Crowd Images. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, jun, 2547–2554.

IOLscitech. 2013. *SA eyes recycling plan for tyres.*

Itti, Laurent, & Koch, Christoff. 2001. Computational modelling of visual attention. *Nature reviews. Neuroscience*, **2**(March), 194–203.

Jain, Ramesh, Kasturi, Rangachar, & Schunck, Brian G. 1995. Object Recognition. *Machine Vision*, 459–491.

Junior, Julio C.S.J., Musse, Soraia R., & Jung, Claudio R. 2010. Crowd analysis using computer vision techniques. *IEEE Signal Processing Magazine*, 66–77.

Kadir, Timor, & Brady, Michael. 2001. Saliency, Scale and Image Description. *International Journal of Computer Vision*, **45**(2), 83–105.

Kamencay, Patrik, Zachariasova, Martina, & Breznan, Martin. 2012. A New Approach for Disparity Map Estimation from Stereo Image Sequences using Hybrid Segmentation Algorithm. *ijmer.com*, **2**, 3201–3206.

Kelda, Harmeet Kaur. 2014. A Review : Color Models in Image Processing. *International Journal of Computer Technology & Applications*, **5**(April), 319–322.

Kohavi, Ron. 1995. A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. *Pages 1137–1143 of: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2.* IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Kong, Dan, Gray, Doug, & Tao, Hai. 2005. Counting Pedestrians in Crowds Using Viewpoint Invariant Training. *Procedings of the British Machine Vision Conference 2005*, 63.1—-63.10.

Kong, Dan, Gray, Doug, & Tao, Hai. 2006. A Viewpoint Invariant Approach for Crowd Counting. *18th International Conference on Pattern Recognition (ICPR'06)*, 1187–1190.

Kumar, Tarun, & Verma, Karun. 2010. A Theory Based on Conversion of RGB image to Gray image. *International Journal of Computer Applications*, **7**(2), 5–12.

Lazebnik, Svetlana, Schmid, Cordelia, & Ponce, Jean. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**, 2169–2178.

Leibe, Bastian, & Schiele, Bernt. 2003. Interleaved Object Categorization and Segmentation. *Bmvc*, **2**(15752), 759–768.

Leibe, Bastian, Leonardis, Ales, & Schiele, Bernt. 2004. Combined Object Categorization and Segmentation with an Implicit Shape Model. *ECCV'04 Workshop on Statistical Learning in Computer Vision*, 1–16.

Lempitsky, Victor, & Zisserman, Andrew. 2010. Learning to count objects in images. *Pages 1–9 of: Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010.*

Li Fei-Fei, & Pietro Perona. 2005. A Bayesian Hierarchical Model for Learning Natural Scene Categories. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, **2**, 524–531.

Lindeberg, Tony. 1998. Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*, **30**, 79–116.

Lowe, David G. 1999. Object recognition from local scale-invariant features. *Page 1150 of: ICCV '99 Proceedings of the International Conference on Computer Vision*, vol. 2.

Lowe, David G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, **60**(2), 91–110.

Lula, James, & Bohnert, George. 2000. Scrap Tire Recycling. *Scrap Tire Recycling Europe*, 2.

Marana, A.N., Velastin, S.A., Costa, L.F., & Lotufo, R.A. 1998. Automatic estimation of crowd density using texture. *Safety Science*, **28**(3), 165–175.

Marr, D, & Nishihara, H K. 1978. *Representation and recognition of the spatial organization of three-dimensional shapes.*

Marr, D, & Poggio, T. 1976. Cooperative computation of stereo disparity. *Science (New York, N.Y.)*, **194**(4262), 283–287.

Marr, David. 1982. *A computational investigation into the human representation and processing of visual information.*

Matas, J., Chum, O., Urban, M., & Pajdla, T. 2002. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. *Pages 384–393 of: In British Machine Vision Conference.*

MathWorks. 2016a. *Matlab Documentation (R2016b) im2bw.* https://www.mathworks.com/help/images/ref/im2bw.html. Accessed: 2016-06-15.

MathWorks. 2016b. *Matlab Documentation (R2016b) Image Processing Toolbox.* https://www.mathworks.com/help/images. Accessed: 2016-04-11.

MathWorks. 2016c. *Matlab Documentation (R2016b) rgb2gray.* https://www.mathworks.com/help/images/ref/rgb2gray.html. Accessed: 2016-06-15.

MathWorks. 2016d. *Matlab Documentation (R2016b) Statistics and Machine Learning Toolbox.* https://www.mathworks.com/help/stats. Accessed: 2016-04-10.

MATLAB. 2014. *version 8.4.0 (R2014b).* Natick, Massachusetts: The MathWorks Inc.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., & Gool, V.L. 2005. A Comparison of Affine Region Detectors. *International Journal of Computer Vision*, **65**(1-2), 43–72.

Mikolajczyk, Krystian, & Schmid, Cordelia. 2002. An affine invariant interest point detector. *Pages 128–142 of: Computer Vision - ECCV 2002*, vol. 2350.

Mikolajczyk, Krystian, & Schmid, Cordelia. 2004. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, **60**(1), 63–86.

Mumford, David, & Shah, Jayant. 1989. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, **42**(5), 577–685.

Murphy, Kevin P. 1991. *Machine Learning: A Probabilistic Perspective.* The MIT Press.

Nevatia, Ram. 2008. Segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, jun, 1–8.

Osher, Stanley, & Sethian, James a. 1987. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 12–49.

Otsu, Nobuyuki. 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, **20**(1), 62–66.

Papageorgiou, Constantine P., Oren, Michael, & Poggio, Tomaso. 1998. A general framework for object detection. *In: Computer Vision - ECCV 1998*.

Parvati, K, Prakasa Rao, B S, & Mariya Das, M. 2008. Image Segmentation Using Gray-Scale Morphology and Marker-Controlled Watershed Transformation. *Discrete Dynamics in Nature and Society*, **2008**, 1–8.

Pellegrini, Stefano. 2007. Articulated Object Recognition. *Society for Optics and Photonics*, 14–24.

Pizer, Stephen M., & McAllister, David F. 1994. *The Image Processing Handbook*. Vol. 29. CRC Press.

Raju, A, Dwarakish, G.S., & Reddy, D. Venkat. 2013. A Comparative Analysis of Histogram Equalization based Techniques for Contrast Enhancement and Brightness Preserving. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, **6**(5), 353–366.

Rao, Daggu Venkateshwar, Patil, Shruti, Babu, Naveen Anne, & Muthukumar, V. 2006. Implementation and Evaluation of Image Processing Algorithms on Reconfigurable Architecture using C-based Hardware Descriptive Languages. *International Journal of Theoretical and Applied Computer Sciences*, **1**(1), 9–34.

REDISA. 2014. *What is REDISA?* http://www.redisa.org.za. Accessed: 2014-04-22.

REDISA. 2016. *REDISA Depots*.

Rodriguez, Mikel, & Laptev, I. 2011. Density-aware person detection and tracking in crowds. *International Computer Vision (ICCV)*.

Roerdink, Joe B.T.M., & Meijster, Arnold. 2000. The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. *Fundamenta Informaticae*, **41**(1-2), 187–228.

Roth, Peter M, & Winter, Martin. 2008. Survey of Appearance-Based Methods for Object Recognition. *Transform*.

Rowley, Henry a, Baluja, Shumeet, & Kanade, Takeo. 1998. Neural Network-Based Face Detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*.

Ryan, David, Denman, Simon, Fookes, Clinton, & Sridharan, Sridha. 2009. Crowd counting using multiple local features. *Pages 81–88 of: DICTA '09 Proceedings of the 2009 Digital Image Computing: Techniques and Applications*.

Sebe, Nicu, & Lew, Michael S. 2003a. *Robust Computer Vision: Theory and Applications*. book; computer vision: Kluwer Academic Dordrecht.

Sebe, Nicu, & Lew, Michael S. 2003b. *Robust Computer Vision: Theory and Applications*. Springer.

Shashikanth, CC, & Kulkarni, Parag. 2013. Region based Image Similarity using Fuzzy based SIFT Matching. *Citeseer*, **67**(3), 47–50.

Silpa-Anan, Chanop, & Hartley, Richard. 2008. Optimised KD-trees for fast image descriptor matching. *IEEE Conference on Computer Vision and Pattern Recognition*, **0**, 1–8.

Sjostrom, Per Jesper, Frydel, Beata Ras, & Wahlberg, Lars Ulrik. 1999. Artificial neural network-aided image analysis system for cell counting. *Cytometry*, **36**(October), 18–26.

Sobel, Irwin, & Feldman, Gary. 1968. *A 3x3 Isotropic Gradient Operator for Image Processing*.

Sokolova, Marina, & Lapalme, Guy. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, **45**(4), 427–437.

South African Department of Environmental Affairs. 2012. *Government Gazette Straatskoerant*.

Stauffer, Chris, & Grimson, W.E.L. 1999. Adaptive background mixture models for real-time tracking. *Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Cat No PR00149*, **2**(c), 246–252.

Stenning, D, Kashyap, V, & Lee, T.C.M. 2012. Morphological image analysis and its application to sunspot classification. *Pages 329–342 of: Imperial.Ac.Uk*, vol. 902. Springer New York.

Tongphu, Suwan, Thongsak, Naddao, & Dailey, M.N. 2009. Rapid detection of many object instances. *Advanced Concepts for Intelligent Vision Systems*, 434–444.

Torralba, A.ntonio, Murphy, Kevin .P., & Freeman, William T. 2004. Sharing features: efficient boosting procedures for multiclass object detection. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, **2**.

Tu, Zhuowen, Chen, Xiangrong, Yuille, Alan L., & Zhu, Song-Chun. 2005. Image Parsing: Unifying Segmentation, Detection, and Recognition. *International Journal of Computer Vision*, **63**(2), 113–140.

Tulsani, Hemant. 2013. Segmentation using Morphological Watershed Transformation for Counting Blood Cells. *IJCAIT*, **2**(Iii), 28–36.

Vedaldi, A, & Fulkerson, B. 2010. VLFeat - An open and portable library of computer vision algorithms. *In: ACM International Conference on Multimedia*.

Verma, Ashish. 2013. The Marker-Based Watershed Segmentation- A Review. *International Journal of Engineering and Innovative Technology (IJEIT)*, **3**(3), 171–174.

Viola, Paul, & Jones, Michael. 2001. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, **1**, I—-511—-I—-518.

Wang, Yw. 2007. A tire mark localization method for forensic image analysis. *Journal of the Eastern Asia Society for Transportation Studies*, **7**, 2881–2890.

Yang, Jianjie, Li, Jin, & He, Ye. 2014. Crowd Density and Counting Estimation Based on Image Textural Feature. *Journal of Multimedia*, **9**(10), 1152–1159.

Zhou, Huiyu, Wu, Jiahua, & Zhang, Jianguo. 2010. *Digital Image Processing: Part I.* Bookbook.