

Developing a Machine Learning Algorithm for Outdoor Scene Image Segmentation

By

Zangwa Yamkela

A dissertation submitted in fulfilment of the requirements of the Degree of Master
of Science in Computer Science

Department of Computer Science
Faculty of Science & Agriculture



Uni



University of Fort Hare
Together in Excellence

Supervised by

DR Z. Shibeshi

Co-supervised by

DR M. Ngxande

Abstracts

Image segmentation is one of the major problems in image processing, computer vision and machine learning fields. The main reason for image segmentation existence is to reduce the gap between computer vision and human vision by training computers with different data. Outdoor image segmentation and classification has become very important in the field of computer vision with its applications in woodland-surveillance, defence and security. The task of assigning an input image to one class from a fixed set of categories seem to be a major problem in image segmentation. The main question that has been addressed in this research is how outdoor image classification algorithms can be improved using Region-based Convolutional Neural Network (R-CNN) architecture. There has been no one segmentation method that works best on any given problem. To determine the best segmentation method for a certain dataset, various tests have to be done in order to achieve the best performance. However deep learning models have often achieved increasing success due to the availability of massive datasets and the expanding model depth and parameterisation. In this research Convolutional Neural Network architecture is used in trying to improve the implementation of outdoor scene image segmentation algorithms, empirical research method was used to answer questions about existing image segmentation algorithms and the techniques used to achieve the best performance. Outdoor scene images were trained on a pre-trained region-based convolutional neural network with Visual Geometric Group-16 (VGG-16) architecture. A pre-trained R-CNN model was retrained on five different sample data, the samples had different sizes. Sample size increased from sample one to five, to increase the size on the last two samples the data was duplicated. 21 test images were used to evaluate all the models. Researchers has shown that deep learning methods perform better in image segmentation because of the increase and availability of datasets. The duplication of images did not yield the best results; however, the model performed well on the first three samples.

Keywords

Image segmentation, Outdoor Scene Images, Deep Learning, Region-based Convolutional Neural networks, Computer Vision.



University of Fort Hare
Together in Excellence

Statement of Original Authorship

Declaration

I Yamkela Zangwa Hereby confirm that Development of an Outdoor Scene Image Segmentation using deep learning Technique is my own work and has not been submitted for any requirement at University of Fort Hare and any other university. The dissertation does not contain any published work without a reference or acknowledgment.

Signature

Date

.....



University of Fort Hare
Together in Excellence

Acknowledgements

To God almighty, the author and the finisher of our faith, to him be the glory for all the good things he has done for me. God gave me strength when I felt weak and discouraged, He gave me wisdom and patience to finish my work . I would like to thank my mother Sibongile Mkhize who always made sure that I do not sleep on an empty stomach, she also reminded me to pray every time when I was faced with challenges. A special thank you to my husband Lwandile Ncam for always supporting and believing in me, more that I believed in myself . Thank you to my lab mates Phumelela, Bulelani, Bongisa, Baphumelele and Oscar for their support and always making sure that we all eat lunch together when I did not have money. Lastly I would like to thank my supervisor Dr Z. Shibeshi for his guidance, constructive criticism and patience in this research. I acknowledge the financial assistance which came from the Council of Scientific & Industrial Research (CSIR) and Armaments Corporation of South Africa (ARMSCOR). I would also like to thank University of Fort hare for giving me an opportunity to further my studies and actually making me the person that I am today.

University of Fort Hare
Together in Excellence

Table of Contents

Contents

1. Chapter One: Introduction	12
1.1 Research Problem	12
1.2 Research Aim	13
1.3 Research Questions	13
1.4 Objectives	13
1.5 Justification	13
1.6 Results	14
1.7 Research Deliverables	14
1.8 Research Limitations	14
1.9 Dissertation Outline	14
1.10 Summary	15
2. Chapter Two: Background and Literature Review	16
2.1 Background	16
2.1.1 Formation of Digital Image	17
2.1.2 Image Segmentation	18
2.1.3 Image Processing and Computer Vision	19
2.2 Literature Review	20
2.2.1 Image Segmentation Methods	21
2.2.2 Neural Networks	23
2.2.3 Convolutional Neural Networks	24
2.2.4 The Common Deep Network Architectures	26
2.3 Related Work	30
2.4 Comparison and Analysis of Image Segmentation Methods	35
2.5 Summary	36
3. Chapter Three: Research Methodology	37
3.1 Methodology	37



University of Fort Hare
Together in Excellence

3.1	Transfer Learning.....	38
3.2	Conceptual Approach to R-CNN with VGG-16.....	39
3.1	Summary	39
4.	Chapter Four: Research Design & Implementation	40
4.1.1	Converting XML Files to CSV Files	45
4.1.2	Converting CSV Files to TFRecords	46
4.1.3	Training Proces.....	47
4.1.4	Testing Process.....	52
4.2	Summary	52
5	Chapter Five: Results	53
5.1	Sample Results.....	53
➤	Sample One.....	54
➤	Sample Two	55
➤	Sample Three	57
➤	Sample Four.....	58
➤	Sample Five.....	59
5.2	Performance Evaluation.....	61
5.3	Summary	64
6	Chapter Six: Discussion	65
6.1	Evaluation of Objectives.....	65
6.2	Results Evaluation.....	67
6.3	False discovery rate, Precision and Recall.....	68
6.4	Summary	69
7	Chapter Seven: Conclusion and Future Work	70
7.1	Empirical Findings vs Research Questions	70
7.2	Recommendations.....	71
7.3	Conclusion.....	72
8	References.....	73



University of Fort Hare
Together in Excellence

List of Figures

Figure 2.1: Image Formation [7].....	17
Figure 2.2: Segmentation Use Cases [8].	18
Figure 2.3: Neural Networks [28].....	24
Figure 2.4: An illustration of a Convolutional Layer (right) with Five Feature Maps [32].	25
Figure 2.5: Alex Net Architecture Training on Two GPUs [38].	27
Figure 2.6: SegNet Architecture [43].....	29
Figure 2.7: An Illustration of SegNet and FCN [43].	29
Figure 3.1: Research Process.	37
Figure 3.2: Learning Process of Transfer Learning [59].....	38
Figure 4.1: The Architecture of R-CNN [64].	40
Figure 4.2: VGG-16 Architecture [65].....	41
Figure 4.3: An Architecture for Experimental Design.	42
Figure 4.4: The LabImg Interface.	44
Figure 4.5: The XML File.	44
Figure 4.6: A Python Script for Conversion of XML File to CSV File.....	46
Figure 4.7: Training Process at Lower Steps.	48
Figure 4.8: Training Process at 200000 Global Step.	49
Figure 4.9: Losses for Sample 1: (a) Classification Loss, (b) Localization Loss1, (c) Localization Loss2 (d) Objectness Loss, (e) Total Loss and (f) Clone Loss.	49
Figure 4.10: Losses for Sample 2: (a) Classification Loss, (b) Localization Loss1, (c) Localization Loss2 (d) Objectness Loss, (e) Total Loss and (f) Clone Loss.	50
Figure 4.11: Losses for Sample 3: (a) Classification Loss, (b) Localization Loss1, (c) Localization Loss2 (d) Objectness Loss, (e) Total Loss and (f) Clone Loss.	50
Figure 4.12: Losses for Sample 4: (a) Classification Loss, (b) Localization Loss1, (c) Localization Loss2 (d) Objectness Loss, (e) Total Loss and (f) Clone Loss.	51
Figure 4.13: Losses for Sample 5: (a) Classification Loss, (b) Localization Loss1, (c) Localization Loss2 (d) Objectness Loss, (e) Total Loss and (f) Clone Loss.	52
Figure 5.1: Results of The Model Trained with 100 Train Images.	55
Figure 5.2: Results of The Model Trained with 300 Train Images.	56
Figure 5.3: Results of The Model Trained with 500 Train Images.	57
Figure 5.4: Results of The Model Trained with 700 Train Images.	59
Figure 5.5: Results of The Model Trained with 900 Images.....	60
Figure 5.6: True Positives Based on Train Sample Size.	62
Figure 5.7: False Positives Based on Train Sample Size.	62
Figure 5.8: False Negatives Based on Train Sample Size.....	62
Figure 5.9: Segmentation Performance Compared to Training Size.	64

Tables

Table 5.1: Segmentation Performance on The Model Trained with 100 Train Images.....	55
Table 5.2: Segmentation Performance on The Model Trained with 300 Train Images.....	56
Table 5.3: Segmentation Performance on The Model Trained with 500 Train Images.....	58
Table 5.4: Segmentation Performance on The Model Trained with 700 Train Images.....	59
Table 5.5: Segmentation Performance on The model Trained with 900 Images.....	60
Table 5.6: Confusion Matrix.....	Error! Bookmark not defined.
Table 5.7: Compiled Results for Different Train Sample Size.....	63



University of Fort Hare
Together in Excellence

Acronyms

CNN	Convolutional Neural Network
R-CNN	Region-based Convolutional Neural Network
HDR	High Dynamic Range
FDR	False Discovery Rate
BBS	Background Subtraction Algorithm
LOTS	Lehigh Omnidirectional Tracking System
MGM	Multiple Gaussian Model
SGM	Single Gaussian Model
FAR	False Alarm Rate
XML	Extensible Markup Language
VLC	VideoLan Client
CSV	Comma-Separated Values
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge.

HMM Hidden Markov Model

VGG Visual Geometry Group

ReLU Rectified-Linear Unit

Resnets Residual Networks

BN Batch Normalization

ELU Exponential Linear Unit

SVM Support Vector Machine

FDR False Discovery Rate



University of Fort Hare
Together in Excellence

1. Chapter One: Introduction

This research addresses supervised image segmentation algorithm in an outdoor environment. Image segmentation is the technique of partitioning a digital image into sets of pixels using various classifications that compares the image to various items, where pixels with similar spectral characteristics are grouped together. The images that are used in this research are the images captured from an outdoor environment, these images are called scene images or outdoor scene images to give a clear idea of where the images were captured. This chapter provides an introduction to the work contained in this dissertation. It covers the research problem, research aim, research questions, research objectives, justification, results, research deliverables, research limitations and dissertation outline.

1.1 Research Problem



There has been no segmentation method that works best on any given problem, unstructured objects (e.g., sky, road, tree, grass, etc.) usually comprise the backgrounds of the images. The background objects have nearly homogenous surfaces which makes it difficult to separate different objects in a scene and discriminate foreground objects from a cluttered background [1]. The background clutter causes background subtraction to be insufficient for image segmentation, this is influenced by false positives that persist due to background environment [2]. False positives are the results which wrongly indicate that a particular attribute is present and they can be produced due to critical conditions of image segmentation in an outdoor environment. Saturation, dynamic nature of the environment, camera motion and sensor noise are the factors which lead to these false positives. These factors have a negative impact on object recognition, which is part of image segmentation, detection and classification [3]. In trying to provide a solution to the problem, a machine learning algorithm is used in this dissertation. The algorithm solves the problem by tolerating the conditions which have a negative impact in outdoor scene image segmentation.

1.2 Research Aim

The aim of this research is to improve the algorithm of outdoor scene image segmentation.

1.3 Research Questions

1. Which methods are used for outdoor scene image segmentation?
2. Which techniques can be used to reduce false positives during segmentation and classification of images?
3. How can we improve the accuracy of image segmentation algorithms in an outdoor environment?

1.4 Objectives

1. To review and analyze methods for image segmentation and outdoor scene image segmentation algorithms.
2. To investigate ways of reducing false positives during image segmentation.
3. To improve outdoor scene image segmentation method by minimizing the occurrence of false positives.

1.5 Justification

Image segmentation is one of the major topics in image processing, it is also the key technology in intelligent monitoring systems and is involved in any military and civilian applications [4]. Monitoring an outdoor environment is important especially for the South African National Defence Force (SANDF) personnel to recognize every object that is in their working environment [5]. It is significant for them to see and locate their enemies for protection or attack.



University of Fort Hare
Together in Excellence

1.6 Results

The results of this research is a robust image segmentation algorithm which improves segmentation and classification accuracy, and reduces the occurrence of false positives occurring during the segmentation of outdoor scene images.

1.7 Research Deliverables

An efficient algorithm for segmentation of outdoor images and a dissertation.

1.8 Research Limitations

The work contained in this dissertation is image segmentation in an outdoor environment. Although there are many objects in an outdoor environment, this study only focuses on trees, grass, road and sky but the data used also contained other objects.



University of Fort Hare
Together in Excellence

1.9 Dissertation Outline

The dissertation consists of seven chapters inclusive of the introductory chapter, which is chapter one.

Chapter 2: presents a detailed background on image segmentation, literature review on algorithms that are used for outdoor scene image segmentation and image segmentation methods that were compared.

Chapter 3: gives an insight into the methodology used in completing the research. It also contains the experimental design and demonstrates the model used in this research.

Chapter 4: presents the design and implementation. The chapter further discusses how the model was trained based on the size increase of training data.

Chapter 5: discusses the results obtained in this study for each objective.

Chapter 6: presents the discussion of this study.

Chapter 7: presents conclusion and feature work.

1.10 Summary

The chapter introduced the research study by providing the relevant background information, listing the research objectives and outlined the research questions that will be answered by this study. The Chapter states the research scope and concludes with the dissertation structure. An extensive review of image segmentation algorithms and the concept of CNNs is discussed in Chapter 2.

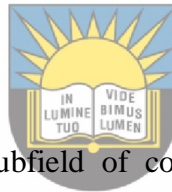


University of Fort Hare
Together in Excellence

2. Chapter Two: Background and Literature Review

This chapter presents a summary of the literature on image segmentation. It covers the research background, related work, and conclusion based on the related work. The background section consists of image formation, image segmentation, image processing, and computer vision. The literature review section consists of outdoor scene image segmentation work using the following methods: graph-based approach, region-based image segmentation, multiclass image segmentation, boundary detection approach, image Segmentation based on perceptual organization, and deep learning algorithms. The related work section discusses supervised and unsupervised work done on image segmentation. The conclusion that was reached in this work is that Region-based Convolutional Neural Network (R-CNN) is a suitable method for this research.

2.1 Background



Object segmentation which is a subfield of computer vision and image processing has become a focus area for the South African National Defence Force. The field of tracking and classifying objects is the centre of attraction under the Planning Tool for Resource Integration, Synchronization, and Management (PRISM) Program. The concepts of classification and identification are pertinent to detecting potential threats and reacting to them. They are also relevant for data mining in surveillance video. Current PRISM research work falls under the central category of target and weapon classification with the long term intention of developing systems for intent detection. [5]. It is obvious that outdoor scene image segmentation algorithms play an important role for military surveillance systems used by military personnel and as such can be embedded in the PRISM applications or programs [6]. This is the main motivation for this research project. One of the most efficient segmentation algorithms is the Convolutional Neural Network. In this research, a Pre-trained R-CNN with COCO dataset was retrained with the CVonline dataset. The next section presents how digital images are formed.

2.1.1 Formation of Digital Image

The goal of the research is to intelligently analyse and manipulate images or video frames to perform image segmentation from an outdoor environments images. Image manipulation and analysis follow a certain scene of geometry and image formation processes. The image formation is composed of geometric primitives which are points, lines, and plains to describe three- dimensional shapes, as described in Figure 2.1 [7].

Images are formed out of discrete colour and intensity values. The values relate to the environment's lighting, surface properties, camera optics, and properties of a sensor. The image is formed because of point or area light sources [7].

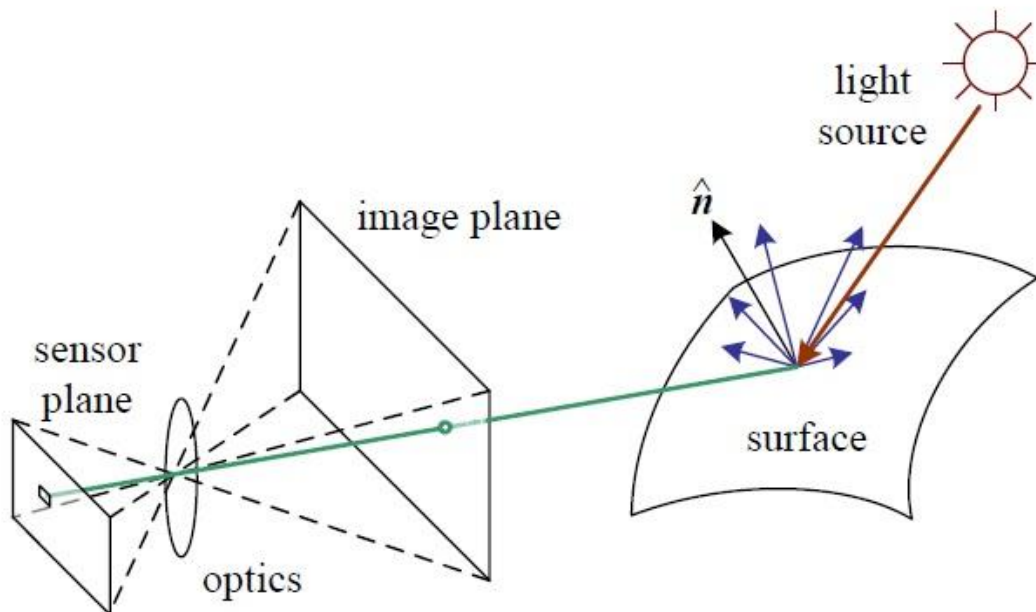


Figure 2.1: Image Formation [7].

When light hits the surface with an object that is captured by a camera, scattered, and reflected. This light reaches the camera, passes across the lens, and reaches the sensor. The photons arriving at the sensor are finally converted into digital Red, Green, Blue (RGB) values which are converted in digital images [7]. The digital images are then processed by applying image processing techniques to get more meaningful information from an image. The following section presents image segmentation which is one of the image processing techniques.

2.1.2 Image Segmentation

Image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. Image segmentation has played an important role in image processing and computer vision in the past few years, and it still is. There are two different types of image segmentation which are shown in Figure 2.2, namely semantic segmentation and instance segmentation [3].

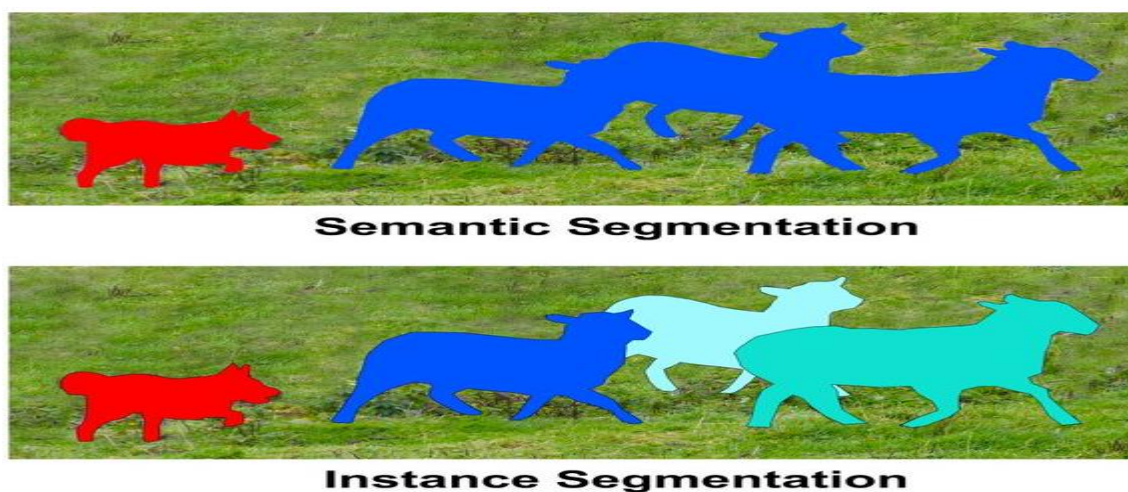


Figure 2.2: Segmentation Use Cases [8].

Semantic segmentation is a process of associating each pixel of an image with a class label whereas instance segmentation is a process of labeling each foreground pixel with object and instance [9]. Image segmentation methods are aimed at concurrent multi-class object recognition and attempt to classify all pixels in an image. Multi-class image segmentation uses several classes (e.g., road, sky, water, etc) for pixel-labeling of an image. It first over-segment the image into super-pixels (or small coherent regions) and then classify each region since classifying every pixel can be computationally expensive [10].

On the other hand, objects in outdoor scenes can be divided into two categories, namely, unstructured objects (e.g., sky, roads, trees, grass, etc.) and structured objects (e.g., cars, buildings, people, etc.). Unstructured objects usually comprise the backgrounds of images.

The background objects usually have nearly homogenous surfaces and are distinct from the structured objects in images [11]. Many recent object segmentation methods, namely, scale-invariant feature transform (SIFT), histogram of oriented gradients (HOG) and Features from Accelerated Segment Test (FAST) have achieved high accuracy in recognizing these background object classes [12], [13],[14].

The challenge for outdoor segmentation comes from the structured objects that are often composed of multiple parts, with each part having distinct surface characteristics (e.g., colours, textures, etc.). Without certain knowledge about an object, it is difficult to group these parts. Many researchers have tackled this difficulty by using object-specific models. However, these models do not perform well when the images contain objects that have not been seen before. As mentioned in Chapter One image segmentation is one of the most vital precursors for image processing based applications and has an essential impact on the whole performance of algorithms developed by many researchers in the field of computer vision [15].



2.1.3 Image Processing and Computer Vision

University of Fort Hare
Together in Excellence

Image processing operations are the main aspects included at the beginning of computer vision methods to process the images for further image analysis. These include exposure correction, colour balancing, image noise reduction, increasing sharpness, and image rotation. Image transforms can manipulate each pixel independently of neighbouring pixels (point operators) and they can also manipulate the pixels depending on its neighbour [16].

Computer vision began in the 1970s to simulate human behaviour and upgrade robots with exceptional intelligence. In the 1980s people focused more on the mathematical side of image analysis. In the 1990s more computer vision projects including object recognition became very popular. In the 2000s much focus was on vision graphics fields which consist stitching of images, light field capture and rendering, high dynamic range (HDR) capturing of images by bracketing the exposure [17].

Although computer vision methods have become the solution to many image and video processing problems, the outdoor environment still presents some challenges for computer

vision [18]. As mentioned above, the outdoor environment is divided into different categories which are the cause of difficulties in computer vision methods, namely, foreground and background objects [19]. Other factors resulting in difficulties in computer vision include loss of information from three dimensions to two dimensions which occur during the capturing of an image with a camera or an eye. When human beings attempt to learn from images, they use prior knowledge to interpret the current images. With humans, the knowledge gathered in the past allows them to reason and solve new problems. In the past decade artificial intelligence attempted to teach computers to learn and understand observations, which progressed tremendously but computer learning ability is still limited. When interpretation is introduced to computer vision, the use of mathematical logic, linguistic as syntax, and semantics are followed [20]. The resulting images in computer vision can be understood as an instance of semantics.

Noise is one of the common effects of images that make image processing difficult. The presence of noise in images requires mathematical tools such as probability theory, to deal with uncertainty. When more complex methods are used, image analysis becomes very complicated as compared with standard tools. On the other hand, the size of images and videos can lead to difficulties in achieving real-time performance if the size is too big [21]. The measured brightness in images is represented by complex image formation physics. Radiance (Brightness, image intensity) depends on the irradiance (intensity, the type of light source and position), the location of the observer, the surface local geometry, and the reflectance properties of the surface [20]. Algorithms for image analysis use a particular storage bin in operational memory and its local neighbourhood, during this process the computer perceives the image through a keyhole. It becomes more difficult for a computer to understand a more global context when perceiving the world through a keyhole. One of the image analysis methods is image segmentation, the following section discusses the literature review.

2.2 Literature Review

This section introduces outdoor image segmentation techniques and related underlying concepts. Furthermore, it introduces deep learning algorithms used for image segmentation. Presented also are the issues which impact outdoor image segmentation.

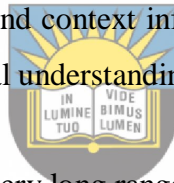
2.2.1 Image Segmentation Methods

There are different types of image segmentation methods that are generally used for outdoor scenes. The aim of using image segmentation methods is to improve image data, to suppress the unwanted distortions and enhance some features of the input image [22]. Outdoor image segmentation, namely, graph-based segmentation, region-based image segmentation, multiclass image segmentation, boundary detection segmentation, and image segmentation based on perceptual organization are discussed below.

- **Graph-based Segmentation:** The graph-based image segmentation approach defines the boundaries between regions by measuring the dissimilarity between the neighbouring pixels. Each pixel is equivalent to a node in the graph. Weights on each edge determine the dissimilarity between pixels. Different methods follow this approach and one common technique is Normalized cut (Ncut). Ncut method organizes nodes into groups so that within the group the similarity is high and between the groups the similarity is low. This method is relatively robust and can be recursively applied to get more than two clusters. Each time the subgraph is partitioned to have the maximum number of nodes. When the normalized cut (Ncut) criterion is implemented, the method removes the significant solutions of cutting small sets of isolated nodes in the graph [23].
- **Region-based image segmentation:** Region-based techniques make use of common patterns in intensity values within a cluster of neighbouring pixels. The cluster is referred to as the region, and the goal of the region based image segmentation algorithm is to group regions according to their anatomical or functional roles [1]. The authors in [11] did a work on segmenting semantic street scenes into coherent regions, simultaneously categorizing each region as one of the predefined categories representing an object or background class. A small blob based super-pixels was used for segmentation and it exploited a visual vocabulary tree as an image representation. The goal of the semantic labeling of street scenes was to automatically annotate different regions by labels of commonly encountered object and object categories. In

this technique, instead of modelling co-occurrences of class labels, the spatial co-occurrences between visual words of neighbouring super-pixels were evaluated.

- **Multiclass image segmentation:** Multi-class image segmentation uses one of several classes (e.g., road, sky, water, etc.) for labeling every pixel in an image. Many state-of-the-art methods that use multiclass image segmentation first over-segment an image into super-pixels and classify each region since classifying every pixel can be computationally expensive. Authors in [24] described the need to label each pixel in the image with one of a set of predefined object class labels. In this approach a class label is assigned to a pixel based on a joint appearance, shape and context model. The aim of the approach is that the system should be capable of automatically partitioning an image into semantically meaningful regions each labeled with a specific object class. For this a discriminative model for object class needs to be learned to incorporate texture, layout, and context information efficiently. The learned model is then used for automatic visual understanding and semantic segmentation of images.



This technique can model a very long range of contextual relationships extending over half the size of the image. The primary limitation is the performance of the texture-layout potentials learned by boosting systems. The classification cost grows sub-linearly with the number of classes due to the use of Joint Boosting although training time increases quadratically. When moving to more classes, the simple ontological model is used where each pixel is assigned only one class label. This can lead to semantic confusion.

- **Boundary detection Segmentation:** A boundary is a contour in the image plane that represents dissimilar pixels between the neighbouring objects. In [25], a boundary detection algorithm was proposed, where a boundary detection algorithm based on a large number of generic features calculated over a large image patch. In this algorithm, the context information was provided by a large aperture. The algorithm selected and combined a set of features out of a pool in the learning stage with tens of thousands of generic, efficient Haar wavelets to learn a discriminative model. True probabilities are output in this method whereas other edge detection methods either

output a soft value based on edge strength or a binary value which is not a true probability. When making a decision, this method combines low-level, mid-level and context information across different scales. Learning edge probability can be done by the classification framework used which is an extended Probabilistic Boosting Tree that combines the bootstrapping procedure directly into the tree formation while properly maintaining priors. This approach uses tens of thousands of very simple features considered over a much larger region. This approach is highly adaptive and scalable.

- **Image Segmentation based on perceptual organization:** Perceptual organization refers to a basic capability of the human visual system to obtain relevant groupings and structures from an image without having prior knowledge of the contents of the image. The Gestalt psychologists summarized some underlying principles (e.g., proximity, similarity, continuity, symmetry, etc.) that lead to human perceptual grouping. Authors in [11] proposed a state-of-the-art solution for the problems related to finding contours (segmentation curves), and finding junction (points joined by multiple contours). The contours were found by combining the local and global features. The local cues were combined in a multi-scale oriented signal including brightness, colour and texture gradients. The global information was considered to be in the first 9 generalized eigenvectors, from which a signal was extracted with Gaussian directional derivatives at multiple orientations. The local and global information were then linearly combined, resulting in a globalized probability of boundary, which claims the top spot in the standard Berkeley segmentation benchmark.

In general the different methods used to achieve outdoor scene image segmentation which is discussed above, achieved good results in image segmentation, however, there are limitations presented by these methods. In trying to solve the limitations presented by outdoor scenes image segmentation techniques, a deep learning approach was introduced. Neural networks in which the fundamental principles behind deep learning are discussed in the next section.

2.2.2 Neural Networks

The attempts to find the mathematical representations of information processing in biological systems led to the discovery of neural networks which are also known as artificial neural

networks. A neural network is an interconnected gathering of simple components, units or nodes, whose usefulness is loosely based on the biological neuron. The preparing capacity of the network is put away in between inter-unit connection strengths, or weights, acquired by a procedure of adaptation to gain from a set of training patterns [26].

An illustration of an organic neuron, contrasted with its numerical abstraction, is shown in Figure 2.3. A neuron's input, x_i (an axon from another neuron), is multiplied by a related association weight, w_i (synapse strength), and combined with all other neural inputs to shape the neuron's pre-activation: $Z = b + \sum_i w_i x_i$

Where b represents the bias of the neuron. This bias term can likewise be explained as the spiking limit of the neuron from a biological neuron's perspective. The neuron would then be able to play out some predefined activity f (alluded to as the activation function) on the pre-activation z , resulting to the yield (or activation) $o = f(z)$ [27].

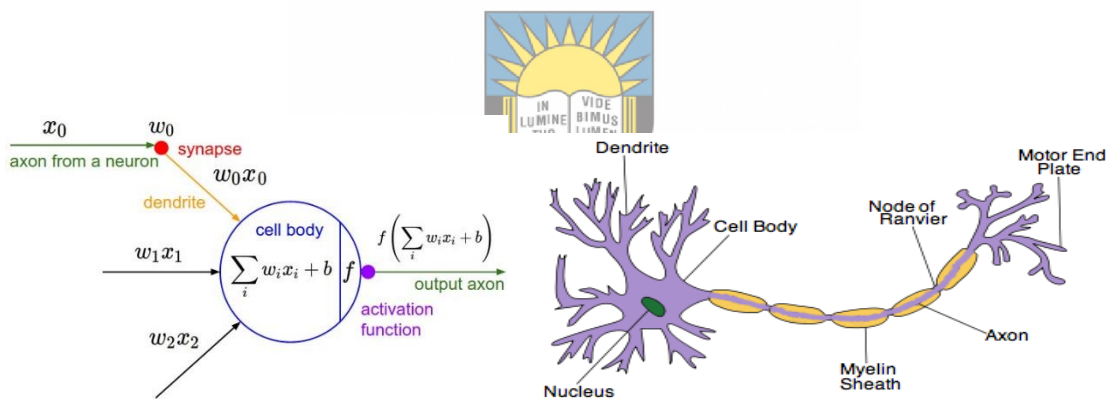


Figure 2.3: Neural Networks [28].

2.2.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are deep artificial neural networks that are used essentially for image segmentation and classification, they are grouped by similarities and perform object recognition in scenes, they are comprised of neurons that have learnable weights and biases [29]. Every neuron gets an input, performs a dot product, and alternatively

tails it with a non-linearity. The entire network despite everything communicates a single differentiable score function from the raw image pixels toward one side to class scores at the other [29]. R-CNN is a family of CNN model designed for object detection and segmentation, it uses pre-trained CNN architectures as its feature extraction networks [30].

How Convolutional Neural Network Works

CNN's are usually composed of a set of layers that can be grouped by their functionalities. Three main types of layers are used to build Convolutional Neural Network architectures: **Convolutional Layer**, **Pooling Layer**, and **Fully-Connected Layer** [31].

Convolutional Layer: Convolutional layers are similar to regular fully connected layers, in that they also have a filter containing a trainable weight matrix, bias and perform the dot product to obtain the pre-activations followed by an activation function. The difference is that the connectivity of a convolutional layer to its input is restricted in such a way that it is especially useful for working with images. The thing that matters is that the network of a convolutional layer to its information is restricted so that it is particularly useful for working with images [32]. To better understand convolution, the following two finite are considered, discrete, 1D functions f and g , where $supp(g) = \{-M, -M + 1, \dots, M- 1, M\}$ then the convolution of these two functions is defined as follows: $(f * g)(n) = \sum_{m=-M}^M f(n - m) g(m)$ where n is the index into f where the convolution is performed [33].

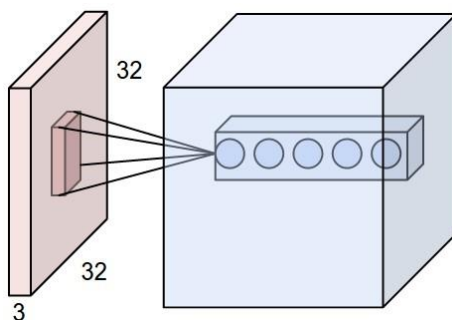
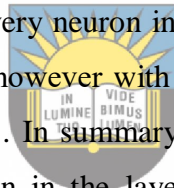


Figure 2.4: An illustration of a Convolutional Layer (right) with Five Feature Maps [32].

- **Pooling Layer:** It is also called down sampling and is used to reduce the size of its input to an extent depending on the hyper parameters that were chosen for the layer. Similar to a convolutional layer, the pooling layer also has a filter, but not a bias one, and differs from regular convolutional layers in that the filter does not contain trainable weights [33]. Instead, a function is applied over the input values within the scope of the filter, as it is slid over the layer input with a given filter stride [34].

A popular decision of function for pooling layers is the maximum activity; however different functions can likewise be used, like an averaging function.

- **Fully Connected Layer:** A fully connected layer can have an arbitrary number of neurons, each associated to all of the M input to the layer (either tests from a data set or yield of neurons in the first layer), The weights of the neurons in the layer can be gathered into a single $N \times M$ weight network W and an N -dimensional bias vector b . Accordingly, every neuron in the layer receives a similar input which is $x = (x_0, \dots, x_m, \dots, x_M)$, however with different weights $w_n = (w_{0,n}, \dots, w_{m,n}, \dots, w_{M,n})$ and bias b_n . [35]. In summary a fully connected layer is essentially any layer where every neuron in the layer is connected to all of the layer inputs. Deeper network architectures are discussed below.



University of Fort Hare
Together in Excellence

2.2.4 The Common Deep Network Architectures

- **AlexNet Architecture:** AlexNet is a convolutional neural network that was structured by the Supervision Group, comprising of Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever. The group made a large, deep convolutional neural system that was used to win the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [36]. The group named the network architecture as AlexNet. They used a relatively simple layout, the network is made of 5 convolutional layers, max-pooling layers, dropout layers, and 3 fully connected layers. The network designed was used for classification with 1000 possible categories. The first time a model performed well on a historically difficult ImageNet dataset was in 2012, with AlexNet. The model Used methods that are as yet used today, for

example, data augmentation and dropout. AlexNet shown in figure 2.5 largely affects the field of AI, especially in the use of deep learning to machine vision [37].

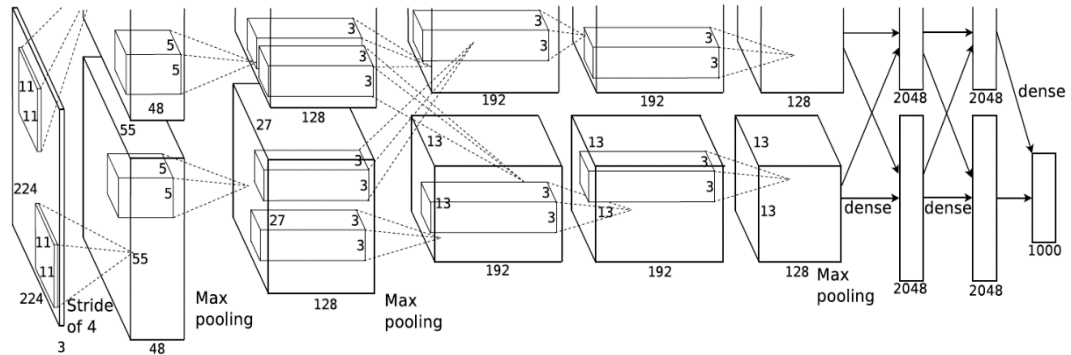
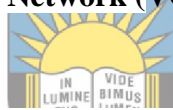


Figure 2.5: Alex Net Architecture Training on Two GPUs [38].

- **Visual Geometry Group Network (VGGNet) Architecture**



Visual Geometry Group Network is one of the neural network architectures that performed very well in the (ILSVRC) in 2014. It took first place on the image localization task and second place on the image classification task [32]. The authors in [39] found that training VGG-16 and VGG-19 is challenging regarding convergence on the deeper networks, so they trained smaller versions of VGG with less weight layers first to make training easier [32]. As a result, the smaller networks converged were then used as initializations for the larger, deeper networks[32].

A comparison of network architectures was done in [40] to investigate the best convolutional neural network architecture, for the fine-grained disease severity classification problem with few training data. The comparison of the networks revealed that fine-tuning on pre-trained deep models can fundamentally improve the performance of little data. Because of this, the fine-tuned VGG-16 model performed best, accomplishing an accuracy of 90.4% on the test set and demonstrating that deep learning is the new encouraging technology for fully automatic disease severity classification.

- **Google Net Architecture**

Google Net is a 22 layers deep network that accomplished the state of the art for classification and detection of objects in the (ILSVRC2014). Researchers have developed deepened network structures that do not increase computational complexity. Google Net uses inception modules that use different convolutions in equal, to extract different feature points [32]. In [41] a technique to classify leaves using the CNN model was proposed and the Google Net model achieved the state of the art for leaf detection, with softmax functions. As per the results obtained, the recognition rate of the system was above 94% when using CNN, even when 30% of the leaf was damaged. The system, therefore, improved past investigations, which accomplished a recognition rate of approximately 90%.

- **Residual Network Architecture**

Residual Network Architecture (ResNet) is presently one of the most stable methods for training CNN models. ResNets comprising of more than 1200 layers and have been trained successfully on most of object recognition benchmarks. ResNet introduced a novel architecture with skip connections and features heavy batch normalization. The skip connections are also known as gated recurrent units and have a strong similarity to recent successful elements applied in recurrent neural networks [42]. ResNet achieved a top 5 error rate of 3.57% which beats human-level performance on the dataset used. In [32] a residual learning framework was presented to ease the training of networks that are substantially deeper than those used in the previous studies. An ensemble of these residual nets achieved a 3.57% error on the ImageNet test set. The result for ResNets won first place on the 2015 ILSVRC classification task. Another supervised method is discussed below.

- **Segmentation Network.**

Understanding of visual scenes is one of the primary goals in a supervised model. Scene understanding involves numerous tasks including recognizing what objects are present, localizing the objects in 2D and 3D, determining the objects' and scene's attributes, characterizing relationships between objects, and providing a semantic description of the scene [43]. Figure 2.6 shows the Segmentation Network architecture

which comprises of the encoder-decoder network, designed for image segmentation and classification [43]. The dataset that is commonly used for segmentation and classification of objects is the COCO dataset [8].

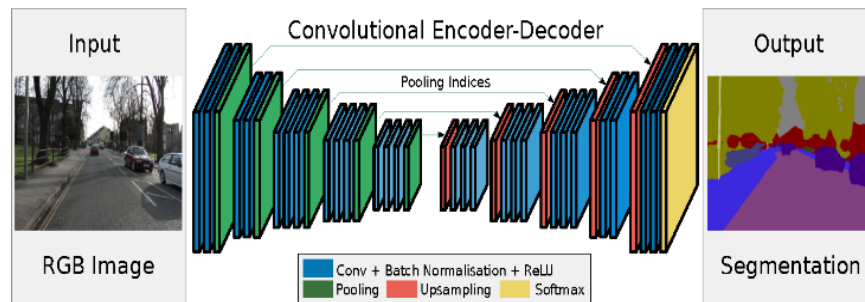


Figure 2.6: SegNet Architecture [43].

- **The Encoder-Decoder Network**

The encoder comprises of 13 convolutional layers that relate to the initial 13 convolutional layers in the VGG-16 network design for object classification. Each encoder layer has a corresponding decoder layer and subsequently the decoder network has 13 layers, shown in the Fully Connected Network and Segmentation Network (SegNet) in figure 2.7. The last decoder yield is fed to a multi-class softmax classifier to produce class probabilities for every pixel autonomously [43].

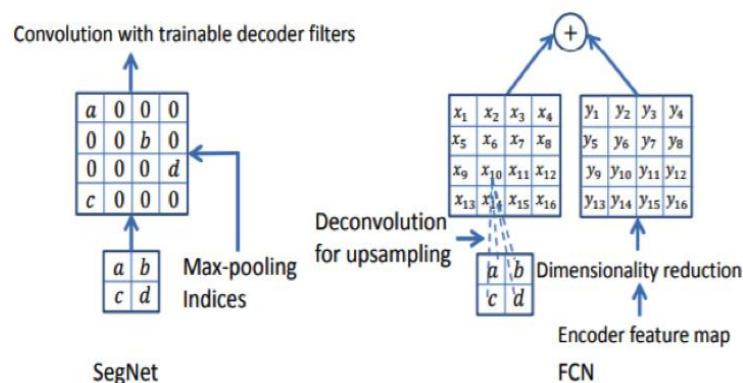
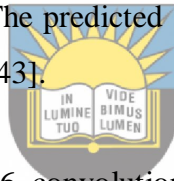


Figure 2.7: An Illustration of SegNet and FCN [43].

Every encoder in the encoder network performs convolution with a filter bank to create a set of feature maps. The sets are then batch normalized, then an element-wise rectified-linear non-linearity (ReLU) $\max(0, x)$ is applied. After that, maximum pooling with a 2x2 window and stride 2 (non-overlapping window) is performed and the resulting yield is sub-tested by a factor of 2. Max-pooling, in this case, is used to achieve interpretation invariance over spatial shifts in the input image [30].

The appropriate encoder in the encoder network takes its input feature map(s) using the memorized max-pooling records from the related encoder feature map(s). This step produces a sparse feature map(s). These feature maps are then convolved with a trainable decoder channel bank to produce dense feature maps. A batch normalization step is then connected to every one of these maps. The high dimensional element portrayal at the yield of the last decoder is fed to a trainable soft-max classifier. This softmax characterizes every pixel autonomously [44]. The output of the softmax classifier is a K channel image of probabilities where K is the number of classes. The predicted segmentation corresponds to the class with maximum probability at each pixel [43].



The encoder has the underlying 16 convolutional layers of the VGG-19 network. This network depends on two decoupled CNNs. The first one is planned to perform a convolution procedure, while the second performs deconvolution. The network performs semantic pixel-wise segmentation. The image maps are handled through a set of conv-layers, Batch Normalization (BN), and Exponential Linear Unit (ELU) activation function units [44]. The following section, 2.2 discusses related work done on supervised and unsupervised segmentation approaches.

2.3 Related Work

Object segmentation approaches can be divided into supervised and unsupervised approaches. Supervised segmentation algorithms use prior knowledge including the ground truth of a training set of images, whereas unsupervised algorithms have input data only and there are no corresponding output variables [45]. Background subtraction, classification, detection, and point detectors are included as categories for object segmentation [46]. Many

researchers did segmentation, classification, and detection focusing on small targets with small signal-to-noise ratio, large targets, moving targets, or stationary targets using scenes captured with ordinary cameras or sensor-based platforms. There has been a need to lessen costs and that has led to a rapid increase of sensors, computational complexity of image processing is reduced by performing low-level computations on the sensor focal plane [47]. It is substantial to do robust image segmentation and detection on small objects for self-defence during attacks and in an infrared search. The cluttered environment where military personnel are deployed, results in unresolved issues with a false alarm rate when using most of the present algorithms [3]. Researchers have used different approaches which address the challenges faced by military personnel in a clustered environment.

A research was done by comparing adaptive background models. This research gave an insight into performance detecting methods, time of computation, and how these methods are used. Due to the greater demand for video surveillance domain for real-time image processing, systems algorithms that are reliable and efficient were recommended for target detection. Authors in [48] compared five adaptive background differencing techniques and the same public benchmark datasets were used by the detectors for evaluation. A large dataset was used and the results were compared concerning autonomously hand-labeled ground truth. Other background subtraction methods were compared, namely, basic background subtraction algorithm (BBS), the W4, Multiple Gaussian Mixture (MGM), Single Gaussian Model (SGM), and Lehigh Omnidirectional System (LOTS) [49].

Basic background subtraction (BBS) was the easiest algorithm that detected objects by computing the difference between image background for each colour channel and the current frame. Furthermore, classifying one by one pixel as a foreground, a threshold operation was used. Objects were segmented from the background using connected component analysis. The second algorithm denoted as W4 was performed in grayscale images, to form the background scene three values were used to represent each pixel. The values used were maximum intensity (Max), minimum intensity (Min), and maximum intensity difference (D) between sequenced frames as a period of training was continuing. In computing foreground objects four steps were followed: thresholding, region-based noise removal, filtering and object detection [50].

Single Gaussian Model (SGM) algorithm was operated in pfinder which is a real-time for tracking objects and it assumed that each pixel was a comprehension of a haphazard variable

with a Gaussian distribution. Independently estimation for each pixel was done for first and second-order statistics of the distribution. In a Mixture of Multiple Gaussians (MGM) all of the background pixels were modelled using the mixture of Gaussians and to adapt the weights and parameters of the Gaussians frames. This method has been used mostly for modelling complex and time-varying backgrounds. LOTS algorithm was applied to greyscale images and uses two image backgrounds with two per-pixel thresholds. Each pixel was treated differently by the per-pixel threshold image to allow the robustness of the detector as to localize noise in image regions with low size. The steps of this algorithm include background and threshold initialization, detection and labeling and backgrounds, and threshold adaptation [51]. Image classification algorithms also play an important role in resolving challenges faced by military personnel.

In [9] classification of scene objects was performed, the unstructured objects which are the sky, trees, grass, etc. and the structured items which are individuals, buildings, vehicles, etc. In most of the images, objects that are structured are the object's background made out of numerous parts and the unstructured objects are the background objects having homogeneous surfaces. It is hard to classify outside scene images as it is made out of both the structured and unstructured objects. Structured objects are not easy to classify as it is poised with various parts with each part having diverse surface attributes. However image classification and segmentation can be performed using top-down methodology or bottom up approach.

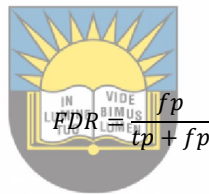
Top-down methodology uses prior knowledge about an object, for example, its shape, shading to manage the segmentation, it follows the hypothesis that the image contains a specific object and can be arranged as a specific type of scene. In the bottom-up approach, the image is first segmented into region and the image regions that relate to a single object are distinguished. The pixels are grouped according to the grey level or texture consistency of an image regions, just as smoothness and coherence of bounding contours [52]. Both-top down and bottom-up methodologies can be combined in one approach.

The hybrid technique combines techniques for both the top-down and bottom-up segmentation approaches. In the top-down methodology, the necessity is to make K as close as conceivable to the initial top-down. The bottom-up limitation expects K to coordinate the image structure, so pixels within the homogeneous image regions, as characterized by the bottom up process are likely to be segmented together into either the figure or background part of the image [53].

The goal of combining top-down and bottom-up segmentation approaches is to construct a classification map $C(x,y)$ that makes the best possible compromise between top-down requirements and bottom-up constraints so that pixels within homogenous image regions, as defined by the bottom-up process, are segmented together into background part of the image. As mentioned above, classification is used to assign the segmented homogeneous regions to particular classes [53]. Assigning segmented homogenous regions to classes can be time-consuming but neural network architectures, are efficient and provide better segmentation model performance.

The performance of segmentation models is influenced by the span of the training data, the quality of an image, and the kind of deep network architecture used [49]. The following are the evaluation measurements used to evaluate the performance of the model.

False Discovery Rate is the proportion of false positive compared to total detected objects.



University of Fort Hare

Accuracy is the extent of the models which were accurately classified and every examples. It is calculated as:

$$Accuracy = \frac{tp+tn}{tp+tn+fp+fn}$$

Precision is the extent of the precedents which truly have class x among each one of those which were named class x. It is calculated as:

$$Precision = \frac{tp}{tp+fp}$$

Recall / Sensitivity is the extent of precedents which were named class x, among all examples which truly have class x. It is calculated as:

$$Recall = \frac{tp}{tp+fn}$$

F1 measure: is the harmonic mean of precision and recall. It is calculated as:

$$F1 = 2 * \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Where

tp = true positives: number of examples anticipated positive that are really positive.

fp = false positives: number of examples anticipated positive that are really negative.

tn = true negatives: number of examples anticipated negative that are really negative.

fn = false negatives: number of examples anticipated negative that are really positive.



The confusion matrix which is a technique used for summarizing the performance of classification algorithms, is then used to summarise the performance of the algorithms after the above evaluation measures have been used.

Commonly used evaluation measures including Recall, Precision, F-Measure, and Rand Accuracy are biased and should not be used without a clear understanding of the biases, and corresponding identification of chance or base case levels of the statistic. Using these measures a system that performs worse in the objective sense of informedness (a measure of how system M is informed about negatives and positives), can appear to perform better under any of these commonly used measures. Informedness is = $\frac{TP}{TP+FP}$ - $\frac{RN}{RN+FN}$, TP is true positives, RP is real positives, FP is false positives, and RN is real negatives.

The authors in [54] discussed that several concepts and measures that reflect the probability which prediction is informed versus chance. Informedness and Markedness (a measure of trustworthiness of positives and negatives by system M) as a dual measure for the probability that prediction is marked versus chance. Lastly there were demonstration of elegant

connections between the concepts of informedness, Markedness, correlation, and significance as well as their intuitive relationships with Recall and Precision, and outline the extension from the dichotomous case to the general multi-class case. A comparison of how outdoor image segmentation performed is presented in 2.3.

2.4 Comparison and Analysis of Image Segmentation Methods

The outdoor environment is a dynamic environment with a lot of clutter. Most segmentation algorithms produce false positives because of the background clutter. Background subtraction methods posed a challenge when the camera is moving while taking videos. Supervised machine learning methods produce better results when segmenting objects in images. Support Vector Machines (SMV), Hidden Markov Models (HMM) and Convolutional Neural Network were compared. SVM classification is essentially a binary (two-class) classification technique, which has to be modified to handle the multiclass tasks in real-world situations. In [55] efficient machine learning approach for classification purposes was proposed. The approach involved a typical image processing steps such as transforming to greyscale and boundary enhancement. Similarly SMV with Hu moments and local binary patterns was proposed and, different leaf features such as colour, shape, and texture were used as well as different classifiers. The algorithm was tested on leaf images from standard benchmark database and compared with other approaches from literature where it proved to be more successful [56].

The HMM is a sequence model. A sequence model or classifier is a model whose job is to assign a label to each unit in a sequence, thus mapping a sequence of observations to a sequence of labels. A new method for plant stress classification that uses supervised learning, via Hidden Markov Models (HMMs) in [57] showed the value and potential to enable more accurate and specific classification of plant stressor types and stressor levels. The evaluation of the potential of Hidden Markov Models for crop classification gave good results. HMM is mostly used in remote sensing and it has been proven to have many advantages, although

there are still some problems that need be solved [57]. Research has shown that CNN is one of the methods that perform better in image segmentation.

The CNN model used Google Net which uses inception modules that use multiple convolutions in parallel, to extract various feature points. The results showed that the recognition rate of this system was above 94% when using CNN, even when 30% of the leaf was damaged. Results observed in the comparative study with other traditional methods suggest that CNN gives better accuracy and boosts the performance of the system due to unique features like shared weights and local connectivity [58]. The goal of this research was to produce segmented images with reduced false positives and track the performance as training data increases.

2.5 Summary

Eventually, this chapter discussed the background work on the field of this research. The background explained image formation, image segmentation methods, machine learning, neural networks, image processing, and computer vision. The related work included most of the work in object detection, including classic algorithms and modern algorithms. The last part of this chapter represented the common aspects, differences, and challenges in the related work. The chapter finally concludes by explaining the significance of using CNN's in this project, the literature showed that CNN's outperforms other image segmentation methods. CNNs are found to give the most accurate results in solving real-world problems. Chapter 3 presents the research methodology, design, and implementation of R-CNN.

3. Chapter Three: Research Methodology

This chapter discusses how the research was conducted and also elaborate more on the approach used to answer research questions. It shows the research process, which includes the formulation of the research problem, research design, R-CNN model, and VGG-16 architectures.

3.1 Methodology

There are different approaches that can be used in a research methodology depending on its nature. Research methodology is the gathering of knowledge through scientific approaches in order to find an outstanding solution to a problem. The different types of research methodology are descriptive, analytical, applied, fundamental, quantitative, conceptual, and empirical. In this study, empirical research methodology is used, which contains the use of experiments to uncover and clarify facts and revise theories. Empirical methodology can be validated or invalidated based on observations and experiments. The empirical research methodology used in this study was conducted to answer questions on the existing methods for outdoor scene image segmentation, to evaluate the performance and accuracy of these methods using False Discovery Rate, Precision, and recall. Related work and literature on outdoor scene image segmentation was revised; furthermore, a pre-trained R-CNN with VGG-16 was retrained with CVonline data and the results were analyzed quantitatively.

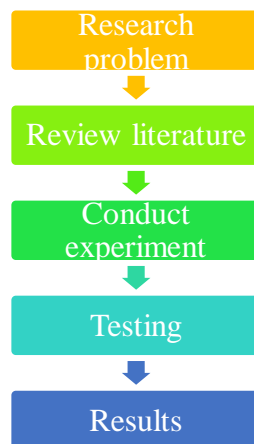


Figure 0.1: Research Process.

Figure 3.1 shows the steps carried out in this study. The problem led to this study was formulated and the research questions were outlined. The problem of this study is that false positives are produced due to critical conditions of image segmentation in an outdoor environment and can wrongly indicate that a particular attribute is present in an image. Transfer learning is used to solve the problem of the study. A background and review study was completed on machine learning, artificial neural networks, computer vision, convolutional neural networks, and existing image segmentation approaches. The method used for the experimental design of this research is discussed. Testing phase followed, raw data was collected and analysed. Through the analysis of data, results were formulated.

3.1 Transfer Learning

Transfer learning is an important tool in machine learning to solve the basic problem of insufficient training data. It tries to transfer the knowledge from the source domain to the target domain by relaxing the assumption that the training data and the test data must be independent and identical [59]. This leads to a great positive effect on many domains that are difficult to improve because of insufficient training data. The learning process of transfer learning is illustrated in the Figure 3.2.

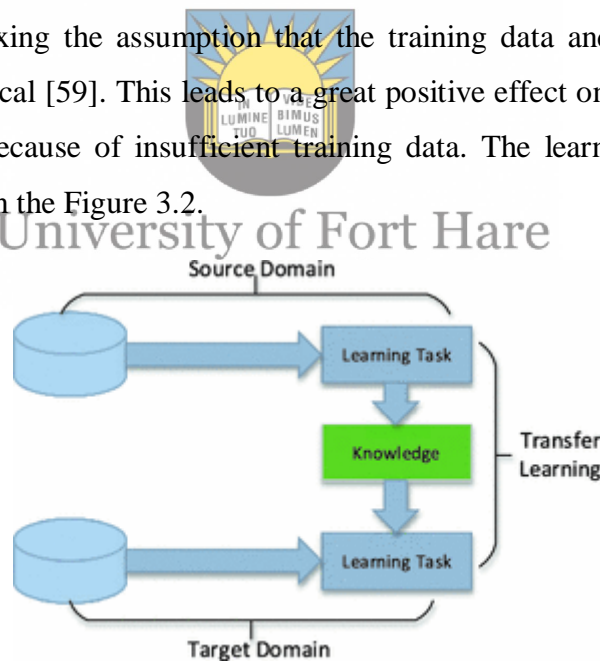


Figure 0.2: Learning Process of Transfer Learning [59].

Given a learning task T_t based on D_t and get the help from D_s for the learning task T_s . Transfer learning aims to improve the performance of predictive function $f_T(\cdot)$ for learning task T_t by discovering and transfer latent knowledge from D_s and T_s where $D_s \neq D_t$ and $T_s \neq T_t$ [59]. The following section discusses the conceptual approach to R-CNN and VGG-16.

3.2 Conceptual Approach to R-CNN with VGG-16

The empirical method which is used in this research is defined as a method that involves the use of objective, quantitative observation in a systematically controlled, replicable situation, in order to test or refine a theory. The empirical method is based on experiment or experience [60]. Chapter two discusses different types of outdoor scene image segmentation techniques and it reveals that traditional segmentation methods do not perform well in an outdoor environment. The purpose of choosing an empirical method is to test, through experiments how machine learning algorithms perform in an outdoor environment. The literature review shows that machine learning algorithms used in this research can tolerate the facts that impact image segmentation in an outdoor environment [61]. R-CNN with VGG-16 is then retrained to give a better scene image segmentation algorithm.

Before looking at different machine learning algorithms, there should be a clear picture of the data that is going to be used, research problem, and constraints. The data used to retrain the R-CNN model are images that were captured from an outside environment which why a deep learning algorithm R-CNN is chosen, to recognise, classify, and segment scene without having to miss its details. Deep learning techniques have achieved state-of-the-art results for object detection, classification, and segmentation such as on standard benchmark datasets and in computer vision competitions. Most notably is the R-CNN with VGG-16, VGG-16 architecture is utilized due to its high accuracy among CNN architectures.

3.1 Summary

This chapter has outlined the methods used and data techniques implemented in our study. The conceptual framework followed for the experiments was presented. This chapter also stated measures that were considered when choosing the model for this research. Chapter four presents the results of the experiments done and their respective discussions and findings.

4. Chapter Four: Research Design & Implementation

This sub-section presents the design and implementation of the research. The problem undertaken in this study is the outdoor scene image segmentation. The problem was tackled in a supervised manner, so a pre-trained R-CNN with VGG-16 [62] was trained using labeled data from CVonline containing images with objects labeled as a tree, grass, road and sky. An open-source framework, Tensorflow image segmentation API [63] was used. Tensorflow image segmentation API is built on Tensorflow to train and evaluate computer vision projects like image segmentation. This framework contains pre-built architectures and weights for R-CNN as shown in Figure 4.1 with VGG-16 model shown in Figure 4.1.

4.1 Model Workflow

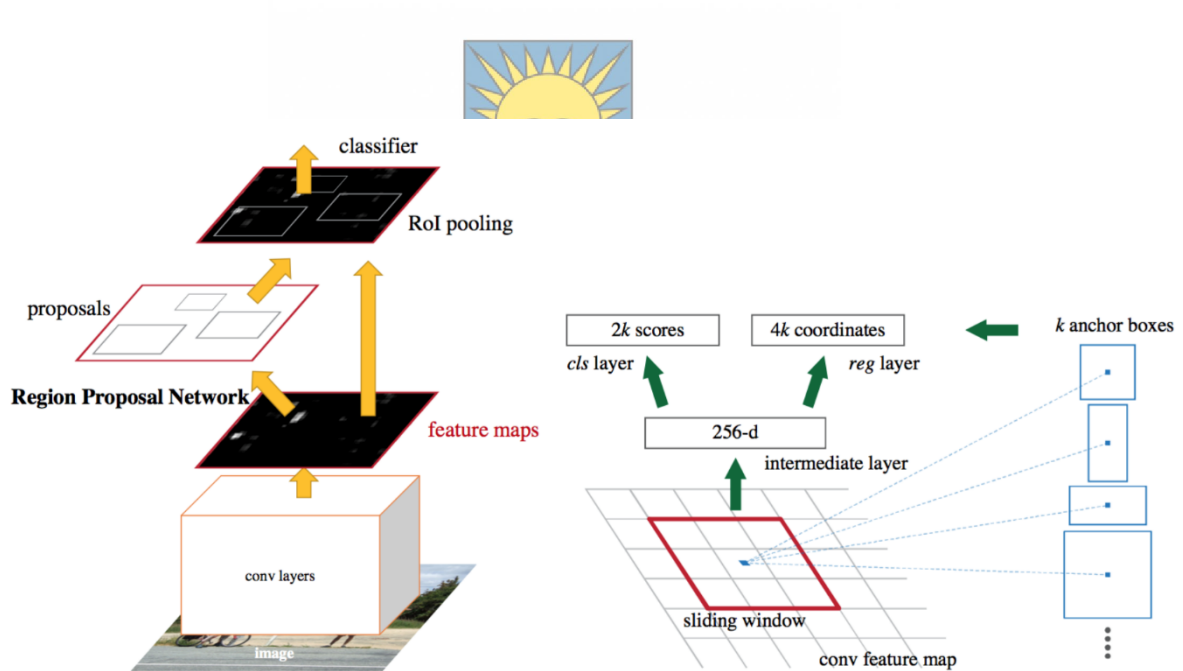


Figure 0.3: The Architecture of R-CNN [64].

R-CNN pre-trains a CNN network called VGG-16 on image segmentation task, it then proposes category-independent regions of interest by selective search (~2k candidates per image). Those regions may contain target objects and they are of different sizes. Region

candidates are warped to have a fixed size as required by CNN. It continues to fine-tune the VGG-16 network on warped proposal regions for $K + 1$ classes. The additional one class refers to the background (no object of interest). In the fine-tuning stage, a much smaller learning rate is used and the mini-batch oversamples the positive cases because most proposed regions are just background [61]. Given every image region, one forward propagation through the VGG-16 generates a feature vector. This feature vector is then consumed by a binary SVM trained for each class independently. The positive samples are proposed regions with IoU (intersection over union) overlap threshold ≥ 0.3 , and negative samples are irrelevant to others. To reduce the localization errors, a regression model is trained to correct the predicted detection and segmentation window on bounding box correction offset using VGG-16 features.

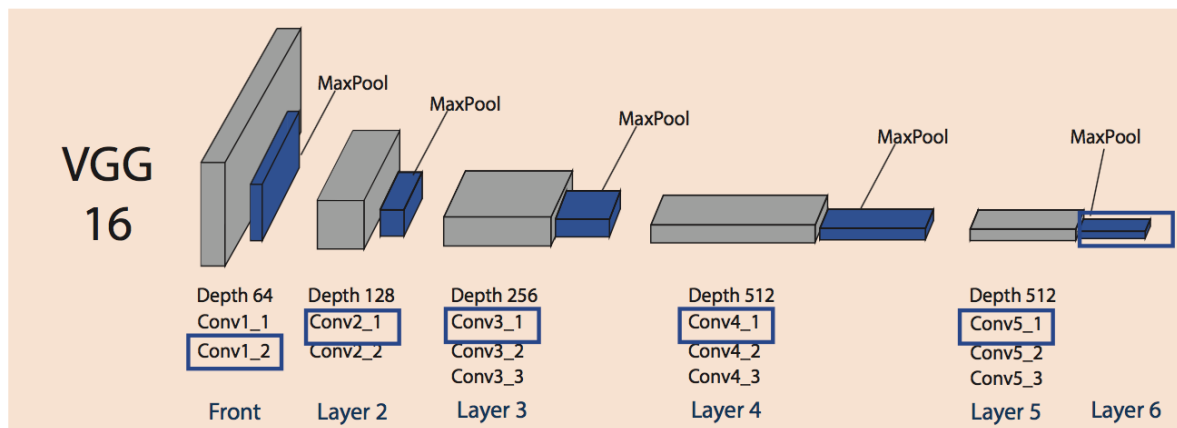


Figure 0.4: VGG-16 Architecture [65].

The input to the conv1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional layers, where the filters are used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). It also utilizes 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel, the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution. Spatial pooling is carried out by five max-pooling layers, which follow some of the convolutional layers but not all the convolutional layers are followed by max-pooling. Max-pooling is performed over a 2×2 pixel window, with stride 2 [65].

The experimental design consists of five stages.

- Image labeling
- Converting XML files to CSV
- Converting CSV to TFRecords
- Training and testing

Figure 4.3 below shows an architecture for experimental design.

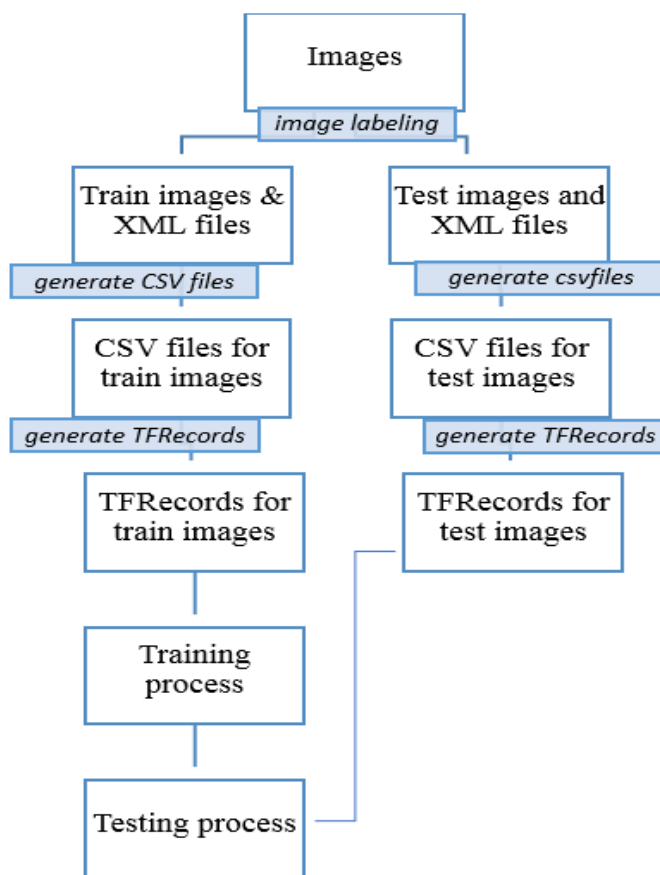


Figure 0.5: An Architecture for Experimental Design.

4.2 Dataset and Labeling

The dataset used in this research was downloaded from the CVonline image databases. The web page has computer vision videos and images which researchers use in evaluating algorithms. The dataset also contains videos from Berkeley dataset, the videos are also used in evaluating algorithms [47], RGB color model, and grayscale images from the outdoor scene environment [66]. The Berkeley videos were captured using 70D cameras in an outdoor environment. They were captured in the high definition (1080 x 1920). The dataset also includes videos which were captured by a fixed and a moving camera. The videos were taken during different times of the day, like in the morning, midday, and afternoon [47]. The images from CVonline dataset were used mostly for training in this study. ISVI IC-C25 RGB camera was used to capture (5056 x 5056) pixel images. During the capturing of these images the camera focus and exposer were adjusted to be suitable for all times of the day [66].

Berkeley dataset videos were converted into frames using VideoLan Client (VLC). Video frames and images were used as initial data for the training and testing phase. The images were labeled using the LabelMe tool. During the labeling, each image was opened, the grasses, road trees, and sky present in the image were rounded with a rectangular box and each image was saved. After the labeling process was thorough, the directory where these labeled images were stored contained XML files in addition to each labeled image.

Two folders were created, one for the training set and one for the testing set. Most of the data was used for training and a smaller portion of the data was used for testing, the ration was 80/20 respectively, the training and testing data used were similar to minimize the effects of data discrepancies and better understanding the characteristics of the model. Understanding the characteristics of the models allows the testing of the models by predicting against the testing set because the data in the testing set already contains known values for the attribute that are needed to do prediction, hence the testing set was duplicated on some of the samples.

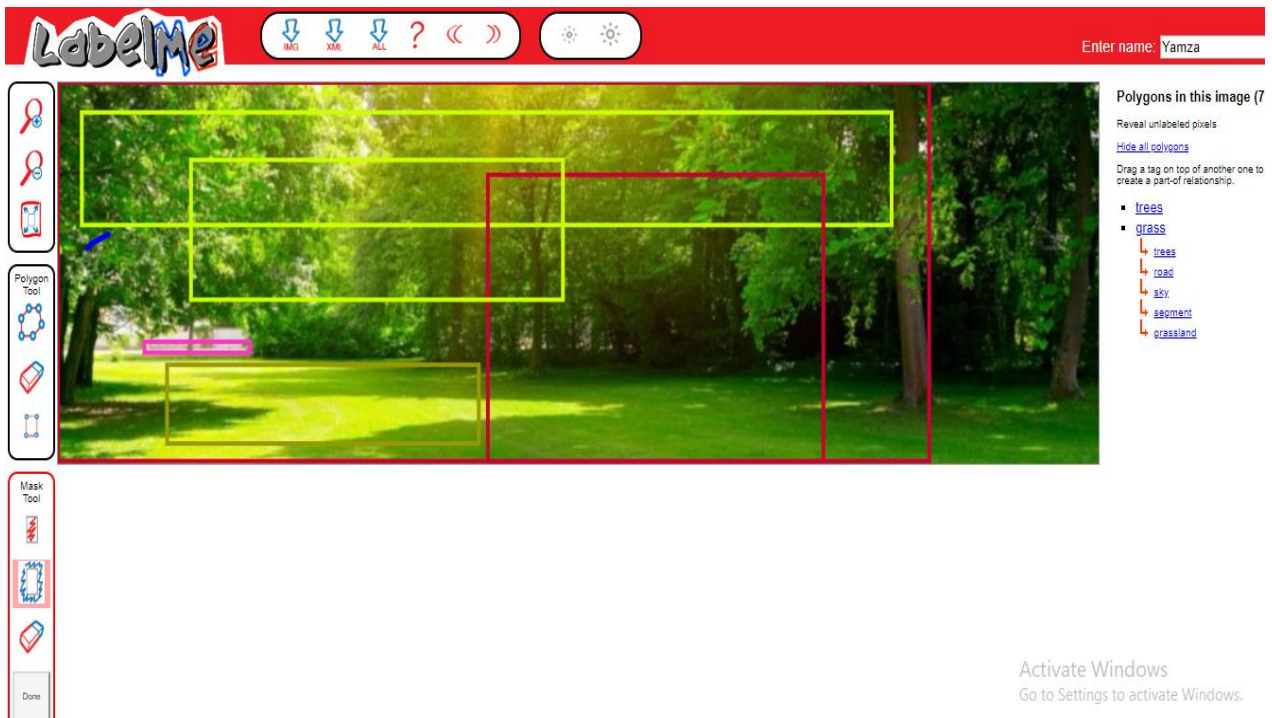


Figure 0.6: The LablImg Interface.

Figure 4.4 represents the LabelMe interface. The left column items show the labeling tools which are used to open the directory containing the data, browse through the directory using the next and previous icon, creating a textbox, zoom in and out and save. The list of labels for each image is represented at the right corner of the interface. The list of files at the bottom right corner represents the images contained in the working directory. When clicking in each image on the file list, it appears in the center of the LabelMe interface and allows for labeling. The labelled images are represented by the XML file.

```

▼<annotation>
  <filename>outdoor-scene.jpg</filename>
  <folder>users/likuy3333/images</folder>
  ▼<source>
    <submittedBy>YamkeLa Zangwa</submittedBy>
  </source>
  ▼<imagesize>
    <nrows>281</nrows>
    <ncols>915</ncols>
  </imagesize>
  ▼<object>
    <name>trees</name>
    <deleted>0</deleted>
    <verified>0</verified>
    <occluded>no</occluded>
    <attributes/>
  ▼<parts>
    <hasparts/>
    <ispartof/>

```

Figure 0.7: The XML File.

Figure 4.5 shows the XML file, each XML file contains information about the image that it represents. It contains the name of the image, size of the image, image source, and the name of all labeled objects together with their sizes.

There were 560 labeled images in the first labeling stage, 500 images were reserved for training and 60 images were reserved for testing. The images with XML files were grouped into five parts for the training phase. The first sample contained 100 labeled images, the second sample contained 300 labeled images, the third sample contained 500 labeled images, the fourth sample contained 700 images and the fifth sample contained 900 images. Since there were five hundred labeled training images, the second set of 430 images was labeled to make the maximum of 900, labeled train images and 90 labeled test images. Each training sample had a ten percent labeled test sample. The XML file representing images was converted into CSV for further image processing.

4.1.1 Converting XML Files to CSV Files



Tensorflow requires data in TFRecord format for training, Therefore XML data was converted into CSV files which was further converted into TFRecords. The conversions were accomplished for each sample. A new directory was created to store the created CSV files for training labels and testing labels. The conversion of XML files to CSV file was done using python3 on google colab. Figure 4.6 below shows how an XML file is converted into a CSV file.

```

import os
import numpy as np
import xml.etree.ElementTree as ET
from collections import OrderedDict
import matplotlib.pyplot as plt
import pandas as pd

def extract_single_xml_file(tree):
    Nobj = 0
    row = OrderedDict()
    for elems in tree.iter():

        if elems.tag == "size":
            for elem in elems:
                row[elem.tag] = int(elem.text)
        if elems.tag == "object":
            for elem in elems:
                if elem.tag == "name":
                    row["bbx_{}_{}".format(Nobj,elem.tag)] = str(elem.text)
                if elem.tag == "bndbox":
                    for k in elem:
                        row["bbx_{}_{}".format(Nobj,k.tag)] = float(k.text)
                    Nobj += 1
    row["Nobj"] = Nobj
    return(row)

df_anno = []

```

Figure 0.8: A Python Script for Conversion of XML File to CSV File.

4.1.2 Converting CSV Files to TFRecords

Tensorflow records were generated using python3 and tensorflow. During the conversion from CSV files to TFRecords, training labels were taken as input to generate train TFRecords and testing labels were taken as input to generate testing TFRecords. The parameters image width, image height, filename, image source id, image format, image bounding box x-minimum, image bounding box x-maximum, image bounding box y-minimum, image bounding box y-maximum, and image class label were considered during conversion. The training TFRecords and testing TFRecords were generated as output and were put in the same folder as training and testing labels.

4.1.3 Training Proces

In this study supervised learning is used as a machine learning method for image segmentation. R-CNN with VGG-16, configured in the MSCOCO dataset was used to segment and classify outdoor scene images, trees, road, grass, and sky. The image resizer was set to keep the aspect ratio resizer at a minimum dimension of 600 and a maximum dimension of 900 to accommodate all images with varying sizes. The first stage features stride was set to 16. The first stage maximum was set to 300, localization loss weight was set to 2.0 for the first stage, objectness loss weight was set to 1.0 for stage 1, initial crop size was set to 14, max-pool kernel size was set to 2 and the max-pool stride was set to 2. Maximum segmentation per class was set to 100 and total classifications were set to 300.

SOFTMAX was used as a score converter. Localization loss weight for the first stage was set to 2.0 and classification loss weight for the second stage was set to 1.0. The batch size was set to one because the CPU used for this research was incapable of taking a large batch size. The initial learning rate was set to 0.0002 and momentum optimizer value to 0.9. The model was set to train until the training process reaches global step 200000. It was believed that at this stage the model has fully learned. The TFRecords and label maps were used as input. Each sample data was trained using the above configuration settings.

A virtual environment called segmentation was created on Ubuntu 16.0 machine. Tensorflow was installed in this environment and all the training process were run in the same environment using python3. Figure 4.7 represents an overview of the training process.

```

INFO:tensorflow:global step 3: loss = 2.1150 (2.302 sec/step)
I1111 07:17:12.810045 140616085546752 tf_logging.py:115] global step 3: loss = 2.1150 (2.302 sec/step)
INFO:tensorflow:global step 4: loss = 1.6971 (1.971 sec/step)
I1111 07:17:14.781958 140616085546752 tf_logging.py:115] global step 4: loss = 1.6971 (1.971 sec/step)
INFO:tensorflow:global step 5: loss = 1.5183 (2.399 sec/step)
I1111 07:17:17.182029 140616085546752 tf_logging.py:115] global step 5: loss = 1.5183 (2.399 sec/step)
INFO:tensorflow:global step 6: loss = 1.2645 (1.829 sec/step)
I1111 07:17:19.012061 140616085546752 tf_logging.py:115] global step 6: loss = 1.2645 (1.829 sec/step)
INFO:tensorflow:global step 7: loss = 1.5945 (2.678 sec/step)
I1111 07:17:21.690442 140616085546752 tf_logging.py:115] global step 7: loss = 1.5945 (2.678 sec/step)
INFO:tensorflow:global step 8: loss = 1.1004 (2.047 sec/step)
I1111 07:17:23.738678 140616085546752 tf_logging.py:115] global step 8: loss = 1.1004 (2.047 sec/step)
INFO:tensorflow:global step 9: loss = 1.2616 (2.052 sec/step)
I1111 07:17:25.791588 140616085546752 tf_logging.py:115] global step 9: loss = 1.2616 (2.052 sec/step)
INFO:tensorflow:global step 10: loss = 1.0291 (1.747 sec/step)
I1111 07:17:27.539938 140616085546752 tf_logging.py:115] global step 10: loss = 1.0291 (1.747 sec/step)
INFO:tensorflow:global step 11: loss = 1.0310 (1.695 sec/step)
I1111 07:17:29.235872 140616085546752 tf_logging.py:115] global step 11: loss = 1.0310 (1.695 sec/step)
INFO:tensorflow:global step 12: loss = 1.0732 (1.590 sec/step)
I1111 07:17:30.826790 140616085546752 tf_logging.py:115] global step 12: loss = 1.0732 (1.590 sec/step)
INFO:tensorflow:global step 13: loss = 0.8241 (2.457 sec/step)
I1111 07:17:33.285054 140616085546752 tf_logging.py:115] global step 13: loss = 0.8241 (2.457 sec/step)
INFO:tensorflow:global step 14: loss = 0.9226 (1.638 sec/step)
I1111 07:17:34.923600 140616085546752 tf_logging.py:115] global step 14: loss = 0.9226 (1.638 sec/step)
INFO:tensorflow:global step 15: loss = 1.4193 (2.164 sec/step)
I1111 07:17:37.088469 140616085546752 tf_logging.py:115] global step 15: loss = 1.4193 (2.164 sec/step)
INFO:tensorflow:global step 16: loss = 1.4158 (2.428 sec/step)
I1111 07:17:39.516920 140616085546752 tf_logging.py:115] global step 16: loss = 1.4158 (2.428 sec/step)
INFO:tensorflow:global step 17: loss = 0.8162 (1.283 sec/step)
I1111 07:17:40.800829 140616085546752 tf_logging.py:115] global step 17: loss = 0.8162 (1.283 sec/step)
INFO:tensorflow:global step 18: loss = 0.9996 (1.902 sec/step)
I1111 07:17:42.703782 140616085546752 tf_logging.py:115] global step 18: loss = 0.9996 (1.902 sec/step)
INFO:tensorflow:global step 19: loss = 1.0699 (2.007 sec/step)
I1111 07:17:44.711306 140616085546752 tf_logging.py:115] global step 19: loss = 1.0699 (2.007 sec/step)
INFO:tensorflow:global step 20: loss = 1.4657 (2.239 sec/step)
I1111 07:17:46.950629 140616085546752 tf_logging.py:115] global step 20: loss = 1.4657 (2.239 sec/step)
INFO:tensorflow:global step 21: loss = 0.8330 (1.839 sec/step)
I1111 07:17:48.790961 140616085546752 tf_logging.py:115] global step 21: loss = 0.8330 (1.839 sec/step)
INFO:tensorflow:global step 22: loss = 0.8940 (2.214 sec/step)
I1111 07:17:51.006606 140616085546752 tf_logging.py:115] global step 22: loss = 0.8940 (2.214 sec/step)
INFO:tensorflow:global step 23: loss = 0.9187 (1.919 sec/step)
I1111 07:17:52.926630 140616085546752 tf_logging.py:115] global step 23: loss = 0.9187 (1.919 sec/step)

```

Figure 0.9: Training Process at Lower Steps.

The training process started at global step one to global step 200000. While the training process was running, the time was recorded, global step number, the loss and the number of seconds it took for each step to run. Figure 4.8 displays the training process at 200000 global steps.



University of Fort Hare
Together in Excellence

```

T0216 13:24:32.965208 140639331838976 tf_logging.py:115] global step 199976: loss = 0.0133 (2.495 sec/step)
INFO:tensorflow:global step 199976: loss = 0.0133 (2.495 sec/step)
T0216 13:24:36.035468 140639331838976 tf_logging.py:115] global step 199977: loss = 0.0078 (3.074 sec/step)
INFO:tensorflow:global step 199977: loss = 0.0078 (3.074 sec/step)
T0216 13:24:39.029887 140639331838976 tf_logging.py:115] global step 199978: loss = 0.0269 (2.793 sec/step)
INFO:tensorflow:global step 199978: loss = 0.0269 (2.793 sec/step)
T0216 13:24:42.001098 140639331838976 tf_logging.py:115] global step 199979: loss = 0.0576 (3.771 sec/step)
INFO:tensorflow:global step 199979: loss = 0.0576 (3.771 sec/step)
T0216 13:24:45.067092 140639331838976 tf_logging.py:115] global step 199980: loss = 0.0122 (2.465 sec/step)
INFO:tensorflow:global step 199980: loss = 0.0122 (2.465 sec/step)
INFO:tensorflow:Saving checkpoint to path training/model.ckpt
T0216 13:24:48.380838 14063789726404 tf_logging.py:115] Saving checkpoint to path training/model.ckpt
INFO:tensorflow:global step 199981: loss = 0.0355 (3.391 sec/step)
T0216 13:24:48.466346 140639331838976 tf_logging.py:115] global step 199981: loss = 0.0355 (3.391 sec/step)
INFO:tensorflow:Recording summary at step 199981.
T0216 13:24:51.031198 140637544240896 tf_logging.py:115] Recording summary at step 199981.
INFO:tensorflow:global step 199982: loss = 0.0116 (3.994 sec/step)
T0216 13:24:52.061622 140639331838976 tf_logging.py:115] global step 199982: loss = 0.0116 (3.994 sec/step)
INFO:tensorflow:global step 199983: loss = 0.0098 (2.873 sec/step)
T0216 13:24:55.335487 140639331838976 tf_logging.py:115] global step 199983: loss = 0.0098 (2.873 sec/step)
INFO:tensorflow:global step 199984: loss = 0.0279 (2.707 sec/step)
T0216 13:24:58.043463 140639331838976 tf_logging.py:115] global step 199984: loss = 0.0279 (2.707 sec/step)
INFO:tensorflow:global step 199985: loss = 0.0119 (3.749 sec/step)
T0216 13:24:58.335487 140639331838976 tf_logging.py:115] global step 199985: loss = 0.0119 (3.749 sec/step)
INFO:tensorflow:global step 199986: loss = 0.0171 (2.491 sec/step)
T0216 13:24:58.643463 140639331838976 tf_logging.py:115] global step 199986: loss = 0.0171 (2.491 sec/step)
INFO:tensorflow:global step 199987: loss = 0.0169 (1.983 sec/step)
T0216 13:25:04.285108 140639331838976 tf_logging.py:115] global step 199987: loss = 0.0169 (1.983 sec/step)
INFO:tensorflow:global step 199988: loss = 0.0145 (2.544 sec/step)
T0216 13:25:08.813593 140639331838976 tf_logging.py:115] global step 199988: loss = 0.0145 (2.544 sec/step)
INFO:tensorflow:global step 199989: loss = 0.0249 (3.644 sec/step)
T0216 13:25:09.974077 140639331838976 tf_logging.py:115] global step 199989: loss = 0.0249 (3.644 sec/step)
INFO:tensorflow:global step 199990: loss = 0.0079 (2.047 sec/step)
T0216 13:25:14.506234 140639331838976 tf_logging.py:115] global step 199990: loss = 0.0079 (2.047 sec/step)
INFO:tensorflow:global step 199991: loss = 0.0445 (3.528 sec/step)
T0216 13:25:18.035273 140639331838976 tf_logging.py:115] global step 199991: loss = 0.0445 (3.528 sec/step)
INFO:tensorflow:global step 199992: loss = 0.0344 (3.886 sec/step)
T0216 13:25:21.922129 140639331838976 tf_logging.py:115] global step 199992: loss = 0.0344 (3.886 sec/step)
INFO:tensorflow:global step 199993: loss = 0.0262 (3.787 sec/step)
T0216 13:25:25.709989 140639331838976 tf_logging.py:115] global step 199993: loss = 0.0262 (3.787 sec/step)
INFO:tensorflow:global step 199994: loss = 0.0289 (3.384 sec/step)
T0216 13:25:29.095835 140639331838976 tf_logging.py:115] global step 199994: loss = 0.0289 (3.384 sec/step)
INFO:tensorflow:global step 199995: loss = 0.0072 (2.210 sec/step)
T0216 13:25:31.305502 140639331838976 tf_logging.py:115] global step 199995: loss = 0.0072 (2.210 sec/step)
INFO:tensorflow:global step 199996: loss = 0.0479 (2.768 sec/step)
T0216 13:25:34.074066 140639331838976 tf_logging.py:115] global step 199996: loss = 0.0479 (2.768 sec/step)
INFO:tensorflow:global step 199997: loss = 0.0135 (3.247 sec/step)
T0216 13:25:37.321780 140639331838976 tf_logging.py:115] global step 199997: loss = 0.0135 (3.247 sec/step)
INFO:tensorflow:global step 199998: loss = 0.0199 (3.145 sec/step)
T0216 13:25:40.467098 140639331838976 tf_logging.py:115] global step 199998: loss = 0.0199 (3.145 sec/step)
INFO:tensorflow:global step 199999: loss = 0.0126 (2.319 sec/step)
T0216 13:25:42.787824 140639331838976 tf_logging.py:115] global step 199999: loss = 0.0126 (2.319 sec/step)
INFO:tensorflow:global step 200000: loss = 0.0228 (2.674 sec/step)
T0216 13:25:45.462098 140639331838976 tf_logging.py:115] global step 200000: loss = 0.0228 (2.674 sec/step)
INFO:tensorflow:Stopping Training.
T0216 13:25:45.462098 140639331838976 tf_logging.py:115] Stopping Training.
INFO:tensorflow:Finished training. Saving model to disk.

```


Figure 0.10: Training Process at 200000 Global Step.

The training process took seven days until it reached a global step 200000. the training process was finished after 200000 global steps and the model was saved to disk. During and after the training phase, the performance of the training was recorded in the tensorboard. Figure 4.9 to Figure 4.13 illustrates the losses during the training of data for the model. The Figure 4.9 shows losses for a model trained with 100 labeled images, Figure 4.10 presents losses for a model trained with 300 labeled images, Figure 4.11 shows losses for a model trained with 500 labeled images, Figure 4.12 presents losses for a model trained with 700 labelled images and Figure 4.13 shows losses for a model trained with 900 trained images.

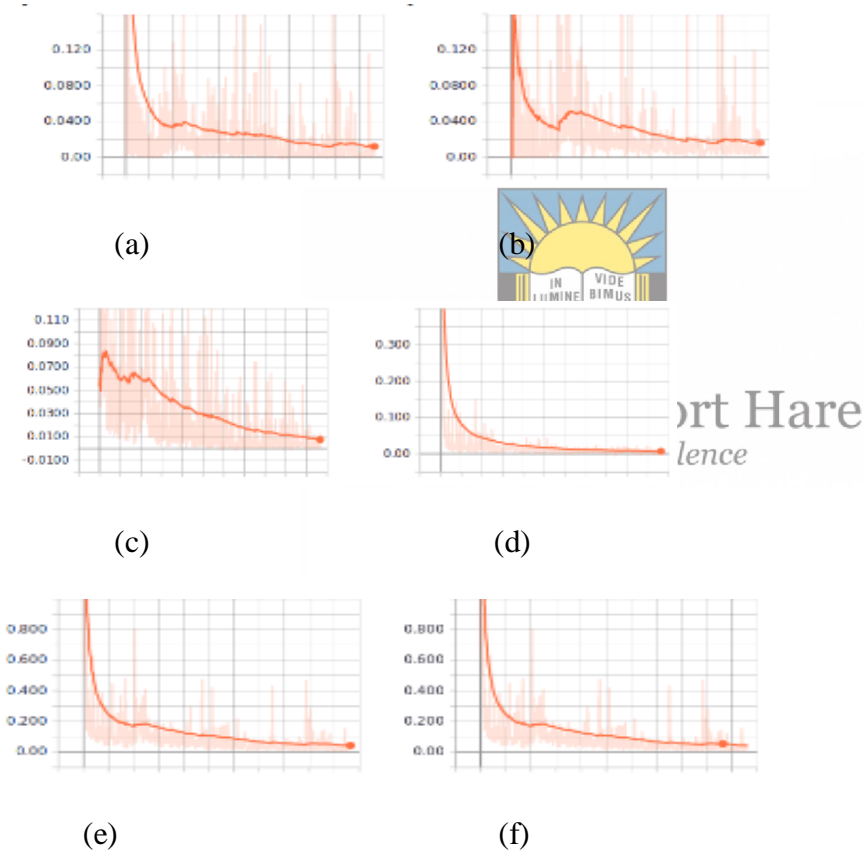
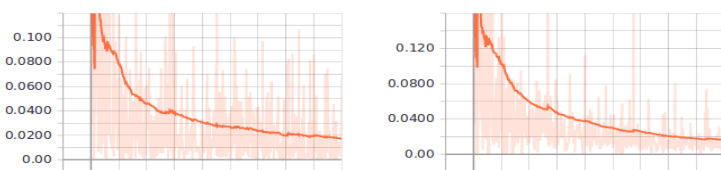


Figure 0.11: Losses for Sample 1: (a) Classification Loss, (b) Localization Loss1, (c) Localization Loss2 (d) Objectness Loss, (e) Total Loss and (f) Clone Loss.



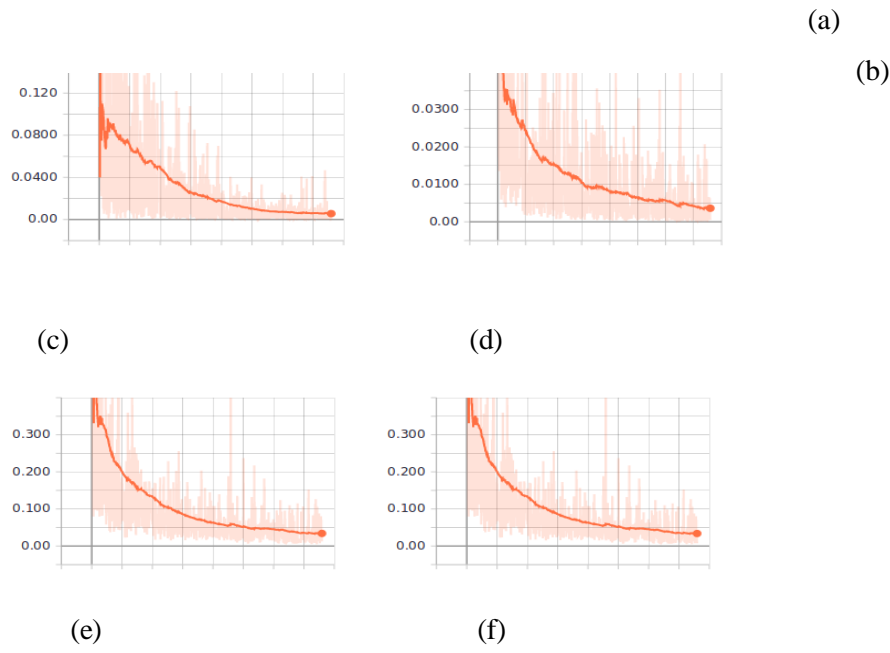


Figure 4.12: Losses for Sample 2: (a) Classification Loss, (b) Localization Loss 1, (c) Localization Loss 2 (d) Objectness Loss, (e) Total Loss and (f) Clone Loss.

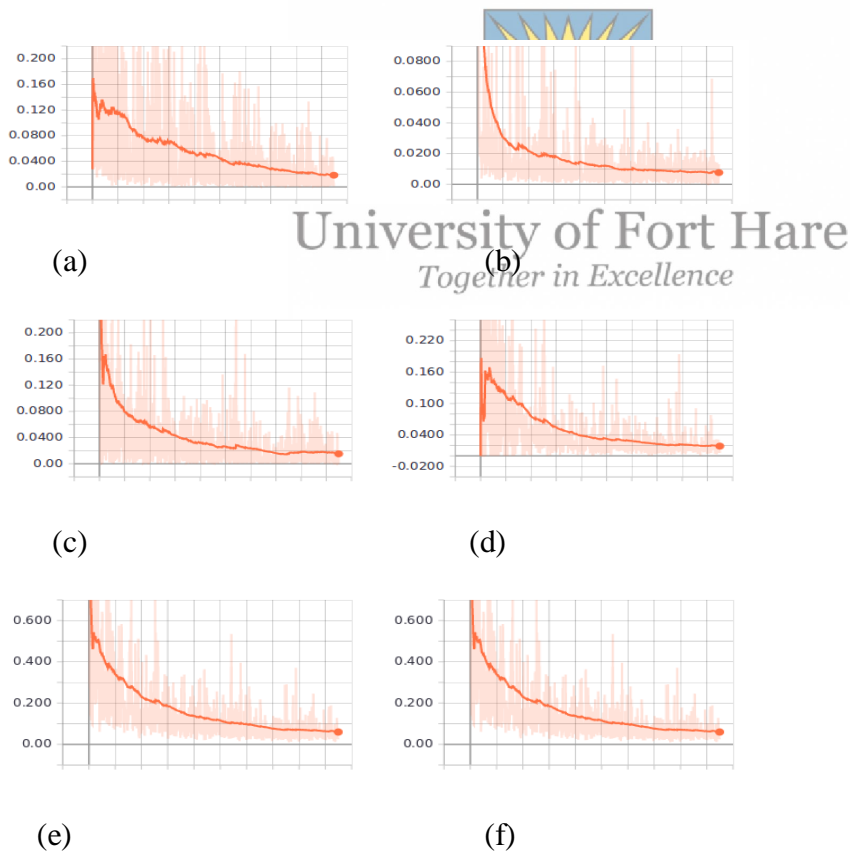
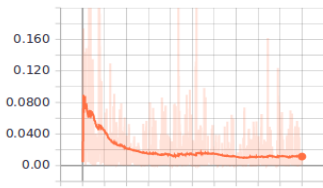
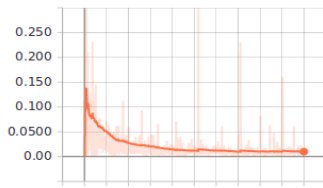


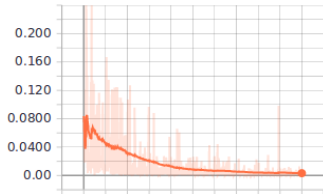
Figure 0.131: Losses for Sample 3: (a) Classification Loss, (b) Localization Loss 1, (c) Localization Loss 2 (d) Objectness Loss, (e) Total Loss and (f) Clone Loss.



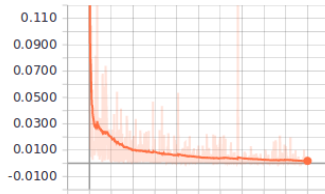
(a)



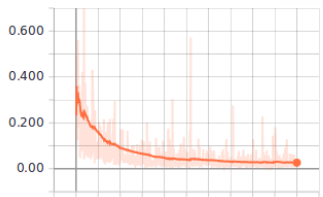
(b)



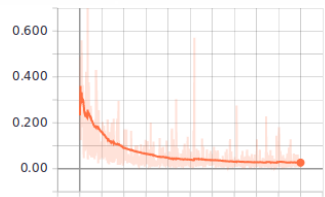
(c)



(d)



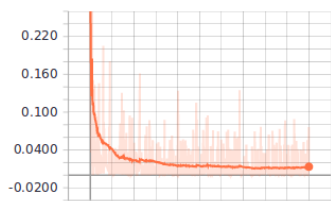
(e)



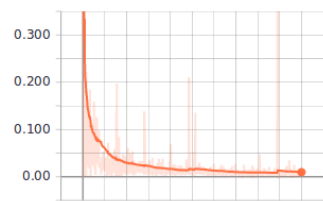
(f)

University of Fort Hare
Together in Excellence

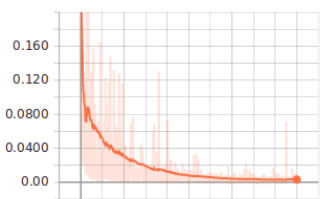
Figure 0.142:: Losses for Sample 4: (a) Classification Loss, (b) Localization Loss1, (c) Localization Loss2 (d) Objectness Loss, (e) Total Loss and (f) Clone Loss.



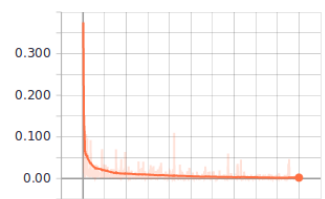
(a)



(b)



(a)



(b)

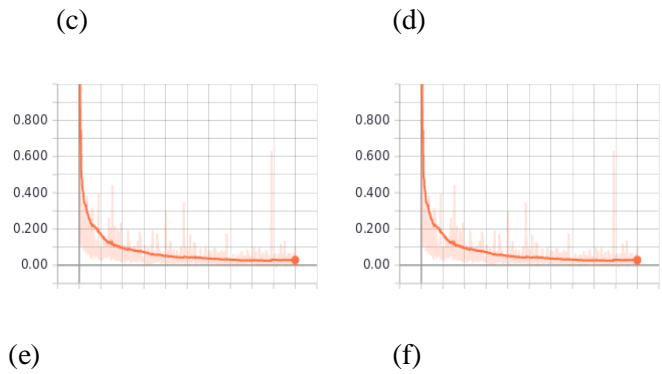


Figure 0.153: Losses for Sample 5: (a) Classification Loss, (b) Localization Loss1, (c) Localization Loss2 (d) Objectness Loss, (e) Total Loss and (f) Clone Loss.

4.1.4 Testing Process

The trained model was saved on the laptop then tested using python3 on a jupyter notebook. The model was tested using different datasets to track the performance of the models based on Sample size. The test data contained 21 images and the output was recorded as results. The results were analyzed and the information was recorded in the form of tables and graphs.

4.2 Summary


University of Fort Hare
Together in Excellence

Chapter 4 has explained the methodology of this study. It outlined how data was collected, converted and implementation of R-CNN with VGG-16 model. The chapter also explains more on the approach followed in computing the experiments. How R-CNN was trained with CVonline dataset and the performance of the training process in terms of losses were presented graphically from sample 1 to sample 5, shown in Figure 4.9 to Figure 4.13 The study results are represented in the next chapter.

5 Chapter Five: Results

In this research transfer learning was conducted on the R-CNN model with VGG-16 which was pre-trained on the COCO dataset. Five different samples of data with different sizes were used to train the model. This chapter presents the results of the experiments, and also how classification performance was evaluated. Each training sample size results are described using tables and mapped in graphs. False Discovery Rate (FDR), precision, and recall matrix are calculated and presented in graphs.

5.1 Sample Results

The training process was done in five sample data to see which sample produced better results. Sample one contained 100 training images, Sample two contained 300 training images, sample three contained 500 training images, sample four contained 700 training images and sample five contained 900 training images. During the testing phase, testing images were taken from each sample to see how the model performances. The segmented images of trees, road, grass, and sky are shown in Figure 5.1 to Figure 5.5, 21 images labelled as image1 up to image 21 for each sample.

The following subsections present the results based on the five sample data. On the testing part, the results that were obtained after testing the R-CNN model with VGG-16 using different samples, sample one to sample five showed that data needed to be increased in order to track the performance of the model. As mentioned in chapter four, the data split of 80/20 had an impact on the performance of the model.

An image is input and a decision of category for each individual pixel is output. Images are classified into one of several possible categories. This means, all pixels bearing trees would be classified into a single category, so are pixels with grass, road, and sky. The subsections below further present the interpretation of results in the form of tables.

Each image was observed to identify, false positives, false negatives, true positives, segmented images, and classified pixels. On the Tables 5.1 to 5.5, image number represents

the order of images from image one to image twenty-one, total objects represents the number of objects present in each image, total segmentations represents the number of segments in each image, false positives denotes false detection instances, true positives denotes correct detection instances and false negative depicts the number of objects not detected.

➤ Sample One



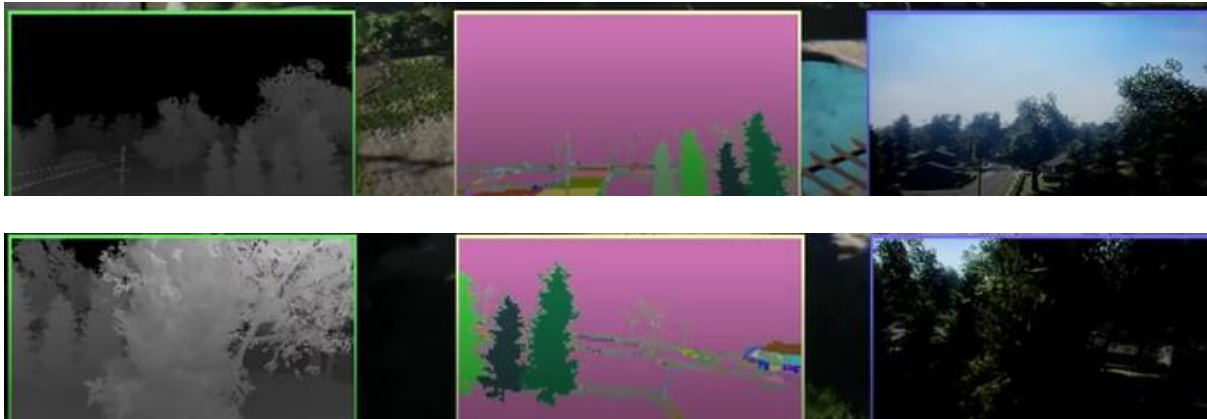


Figure 5.1: Results of The Model Trained with 100 Train Images.

Table 5.1: Segmentation Performance on The Model Trained with 100 Train Images

Image no	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Objects	8	7	8	4	5	5	8	4	6	4	4	5	4	5	8	7	8	5	11	5
Segmentations	5	6	5	3	5	4	5	2	6	2	4	4	3	4	3	6	4	4	8	1
FP	1	1	3	0	0	1	1	0	1	1	1	0	1	0	0	1	1	2	1	0
TP	4	5	2	3	5	3	4	2	5	1	3	4	2	4	3	5	3	2	7	1
FN	3	2	3	1	0	1	4	2	0	2	1	1	1	1	5	2	4	1	3	4

University of Port Harcourt
Together in Excellence

➤ Sample Two

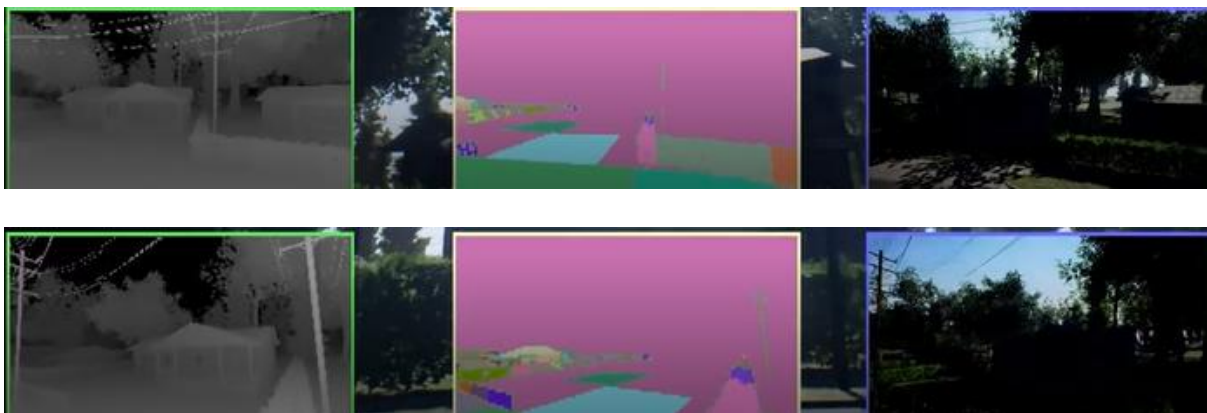




Figure 5.2: Results of The Model Trained with 300 Train Images.

Table 5.2: Segmentation Performance on The Model Trained with 300 Train Images.

<i>Image no</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>	<i>13</i>	<i>14</i>	<i>15</i>	<i>16</i>	<i>17</i>	<i>18</i>	<i>19</i>	<i>20</i>
<i>Objects</i>	8	7	8	4	5	5	8	4	6	4	4	5	4	5	8	7	8	5	11	5
<i>Segmentations</i>	6	6	3	5	4	5	3	1	6	4	7	4	2	4	5	4	6	5	9	1
<i>FP</i>	2	0	0	2	0	1	2	0	0	1	3	0	0	0	1	1	1	0	1	0
<i>TP</i>	4	6	3	3	4	4	1	1	6	3	4	4	2	4	4	3	5	5	8	1
<i>FN</i>	2	1	5	1	1	1	5	3	1	1	0	1	2	0	2	3	3	1	2	3

➤ Sample Three

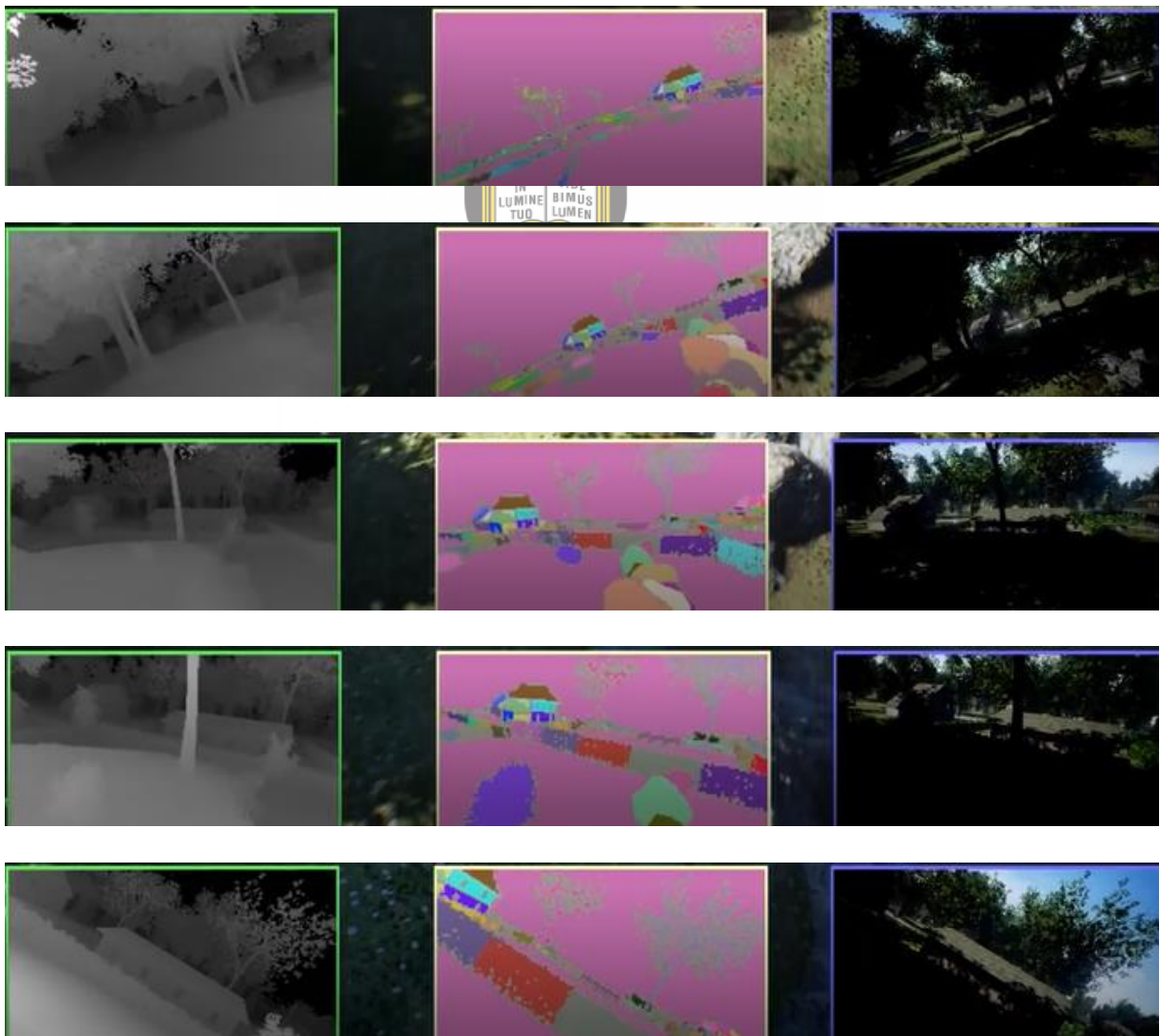


Figure 5.3: Results of The Model Trained with 500 Train Images.

Table 5.3: Segmentation Performance on The Model Trained with 500 Train Images

Image no	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Objects	8	7	8	4	5	5	8	4	6	4	4	5	4	5	8	7	8	5	11	5
Segmentations	3	6	4	3	4	4	6	1	5	3	6	4	2	5	6	6	5	4	9	2
FP	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1	1	0
TP	3	6	4	3	4	4	5	1	5	2	5	4	1	5	6	5	5	3	8	2
FN	5	1	4	1	1	1	2	2	1	1	1	0	2	0	2	1	3	2	1	3

➤ Sample Four



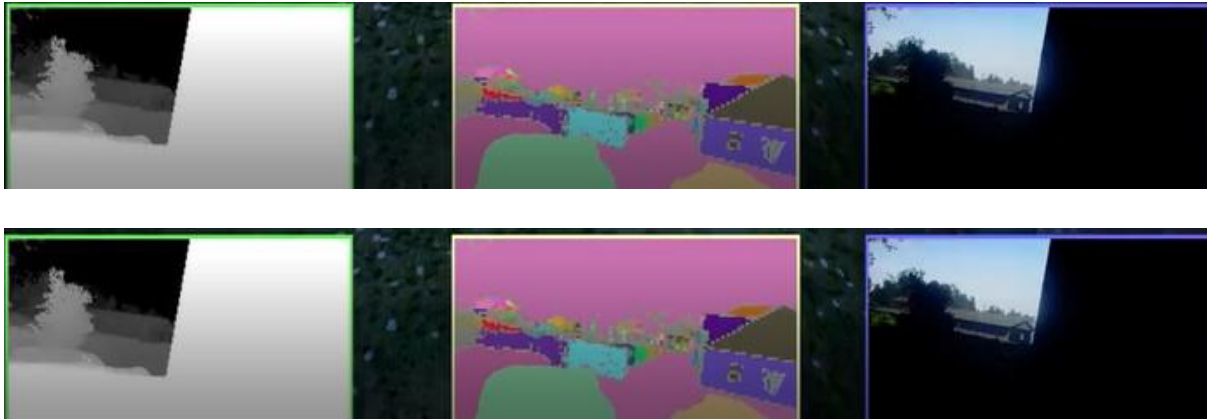


Figure 5.4: Results of The Model Trained with 700 Train Images

Table 5.4: Segmentation Performance on The Model Trained with 700 Train Images.

Image no	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Objects	8	7	8	4	5	5	8	4	6	4	4	5	4	5	8	7	8	5	11	10	5
Segmentations	6	3	2	5	5	4	4	2	5	3	5	4	1	5	4	6	6	3	8	7	2
FP	0	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	1	1	1	1	0
TP	6	3	1	4	5	3	3	2	4	2	4	4	1	5	4	5	5	2	7	6	2
FN	2	4	6	1	0	2	4	2	2	1	2	1	3	0	4	0	2	4	1	2	3

➤ Sample Five

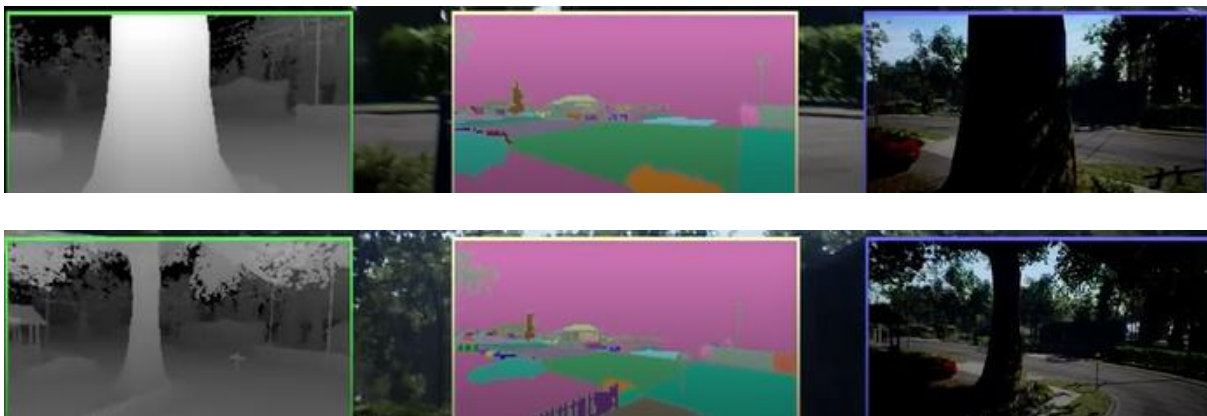




Figure 5.5: Results of The Model Trained with 900 Images.

Table 5.5: Segmentation Performance on The model Trained with 900 Images.

Image no	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Objects	8	7	8	4	5	5	8	4	6	4	4	5	4	5	8	7	8	5	11	7	5
Segmentations	2	2	1	4	2	3	3	2	3	4	4	4	4	4	3	5	4	4	6	5	1
FP	0	0	1	2	0	0	1	0	0	1	0	0	2	0	0	0	1	1	0	1	0
TP	2	2	1	2	2	3	2	2	3	3	4	4	2	4	3	5	3	3	0	0	1
FN	6	4	6	1	3	3	5	3	3	1	1	1	2	1	5	2	4	1	5	4	4

5.2 Performance Evaluation

The segmentation performance of R-CNN with VGG-16 was evaluated in the testing phase. The test data contained 21 images with 121 objects in total. This study made use of a multiclass classification method. The Confusion matrix is used to present the multiclass classification of results. The confusion matrix contains the possible outcomes which describe the segmentation performance. It shows the actual instances and predicted instances. Data split of 80/20 and fine-tuning parameters are used to show the model's performance.

Data Split: The data split into two subsets: training data and testing data and fit the model on the train data, in order to make predictions on the test data. The split is done for, one of two things might happen, overfitting the model or under-fitting the model. The 80/20 data split is done to avoid overfitting and under-fitting because they affect the predictability of the model.

Fine-tuning the model: To reduce the number of parameters in a very deep network like R-CNN, VGG-16 uses a very small 3×3 filters in all convolutional layers (the convolution stride is set to 1). Reducing the number of parameters shows that a significant improvement in the prior-art configurations can be achieved by pushing the depth to 16-19 weight layers. The following table shows the confusion matrix.

Table 5.6: Confusion Matrix

Actual class	Predicted class	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

- True Positive (TP): a tree, road and grass instances were correctly segmented and classified as a tree, road and grass respectively.
- True Negative (TN): a non-tree, non-road and non-grass instances were correctly classified as non-tree, non-road and non-grass respectively.
- False Positive (FP): a non-tree, non-road and non-grass instances were incorrectly segmented and classified as a tree, road and grass respectively.
- False Negative (FN): a tree, road and grass instances were incorrectly classified as non-tree, road and grass respectively.

The entities true positives, false positives, and false negatives are represented in graphs. Figure 5.6 represents the true positives based on increasing train sample size, Figure 5.7 presents false positives based on increasing train sample size and Figure 5.8 shows false negatives based on the increasing sample size.

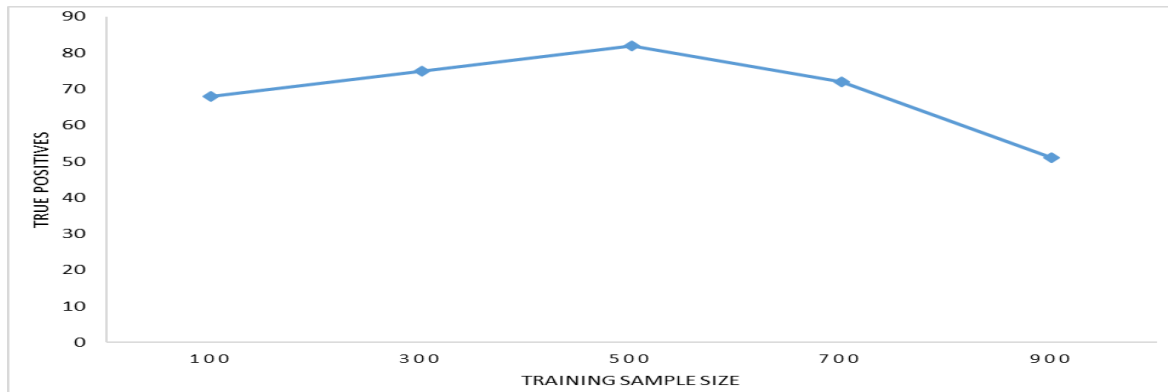


Figure 5.6: True Positives Based on Train Sample Size.

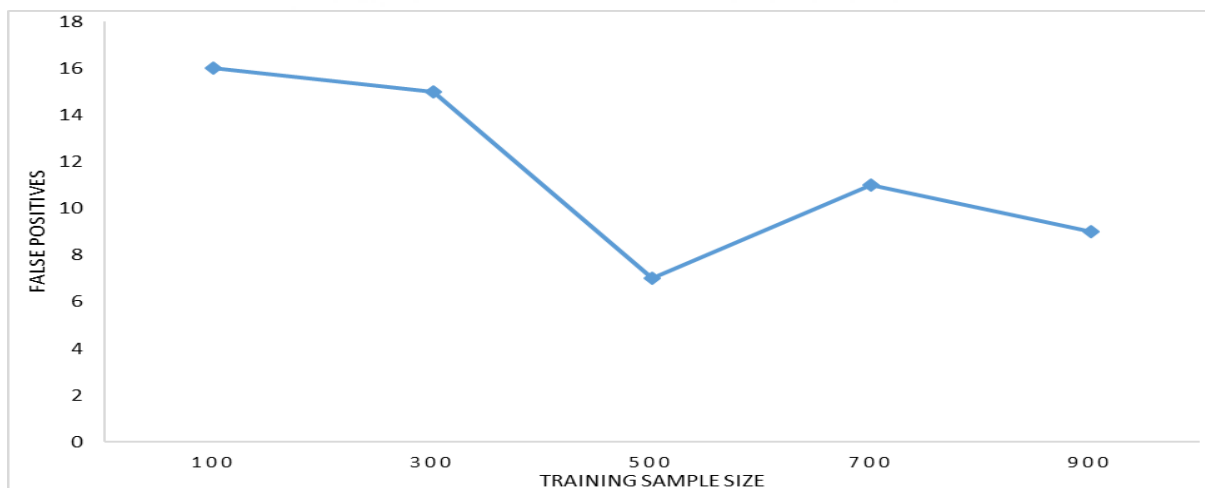


Figure 5.7: False Positives Based on Train Sample Size.



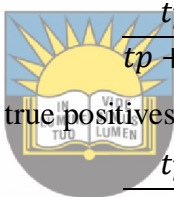
Figure 5.8: False Negatives Based on Train Sample Size.

The results obtained from R-CNN models trained with different sample sizes were added together for each entity. The 21 test images contained 121 objects and segmented objects for each training sample were added together. All false positives and true positives were added together for a model trained with 100 training images, a model 300 trained images, a model trained with 500 images, a model trained with 700 images and a model trained with 900 images. False Discovery Rate (FDR), precision and recall are calculated and presented in graphs, the results shown in Table 5.7 are the ones obtained from each model. The following formulas are used to determine False Discovery rate, precision and recall.

➤ False discovery rate is the proportion of false positive compared to total segmented or detected objects.
$$FDR = \frac{fp}{tp+fp}$$

➤ Precision is the proportion of all true positives compared to all segmented objects.

• Recall is the proportion of all true positives compared to all possible segmentations.



$$\frac{tp}{tp + fp}$$

$$\frac{tp}{tp + fn}$$

Table 5.1: Compiled Results for Different Train Sample Size.

<i>Trained images</i>	<i>Total objects</i>	<i>segmented objects</i>	<i>False Positives</i>	<i>True Positives</i>	<i>False negatives</i>	<i>False discovery rate</i>	<i>Precisio n</i>	<i>Recall</i>
100	121	84	16	68	41	0,180476	0,909524	0,623853
300	121	90	15	75	38	0,177776	0,733333	0,663717
500	121	89	7	82	34	0,088653	0,821348	0,706897
700	121	83	11	72	44	0,14252	0,96747	0,62069
900	121	66	9	51	61	0,14	0,77	0,455357

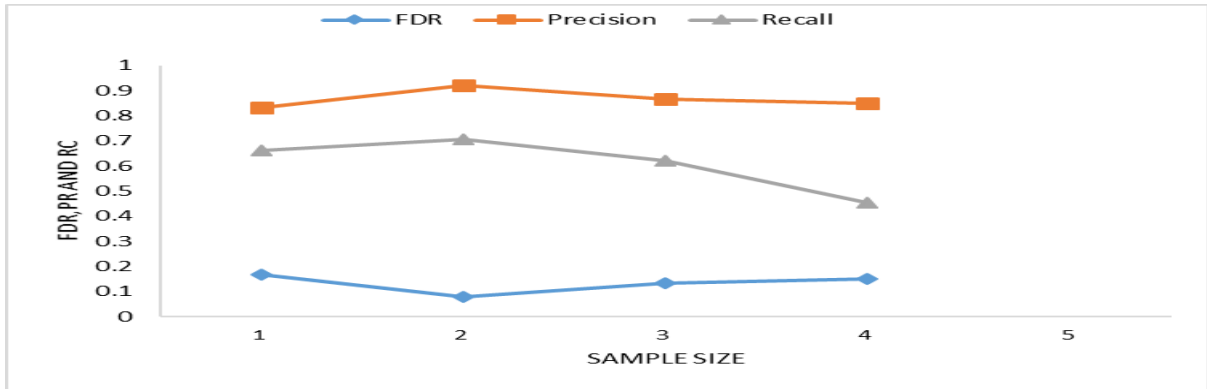


Figure 5.1: Segmentation Performance Compared to Training Size.

The Figure 5.9 presents the false discovery rate (FDR), precision and recall compared to the train sample size.

5.3 Summary




The results obtained from this study are shown and explained in this chapter. It shows the results obtained from training R-CNN VGG-16 using 100 training images, 300 training images, 500 training images, 700 training images, and 900 training images. The results obtained were represented in tables, where the number of false positives, the number of false negatives, and the number of true positives for each image were mapped. Furthermore, graphs of false positives, false negatives and true positives were represented for each sample size. The segmentation performance was evaluated using a confusion matrix. FDR, precision, and recall were computed and mapped in a graph. The results are discussed in chapter 6.

6 Chapter Six: Discussion

This chapter presents and discusses the results from the empirical experiments in chapter four, the chapter is sectioned as follows, evaluation of objectives, results evaluation, FRD, precision and recall, then summary. By interpreting and analysing graphs from the experiments, the chapter explores how the segmentation performance of R-CNN is empirically calculated using Accuracy on the test instances, and how is it affected by model complexity, the sample size of the training set, and type of score functions.

6.1 Evaluation of Objectives

The results of this research depended upon answers to research questions concerning the research objectives. The first two objectives were met through literature and their solution led to the completion of objective three through experiments. In answering the research question as to which methods are used for outdoor scene image segmentation, the focus was not only put on methods used in scene environment but the research included other image segmentation methods. The literature also reviewed methods used in videos since videos are also made up of frames.



University of Fort Hare
Together in Excellence

Objective one: To review and analyse methods for scene image segmentation.

The observations that were found on literature review is that, background subtraction is a commonly used method for object classification, segmentation and detection. The problem is that foreground contains many objects than background which leads to difficulties in discriminating foreground objects from a background [19]. The background clutter caused background subtraction to be insufficient for segmentation. Adaptive background models aided as an improvement to background subtraction, but false positives persisted due to background environment. Visible images and deep learning or real time algorithm were used in this research. Deep learning algorithm showed great performance with less computation time. It was then concluded that convolutional neural networks are the best for outdoor scene image segmentation. Training a model with real-life examples enhances its background knowledge for better image segmentation, but it can be challenging to segment specific

objects using background subtraction methods only so deep learning techniques are recommended.

Objective two: To review and analyze methods for image segmentation and outdoor scene image segmentation algorithms.

The main challenge in object segmentation and classification is the occurrence of false positives. To address this problem many researchers have used these algorithms. The graph based image segmentation, region-based technique, multi-class image segmentation, boundary detection and image segmentation based on perceptual organization. These algorithms had a common drawback, the image regions that correspond to a single object cannot be easily identified [67]. In this research, a real-time deep learning method was used to segment outdoor images while reducing the number of false positives. Deep learning methods improve segmentation because CNN has many layers with different weights [68]. A pre trained R-CNN model was trained with different data sample sizes to track the performance when data increases.



Objective three: To improve outdoor scene image segmentation method by minimizing the occurrence of false positives.

As stated above R-CNN with VGG-16 pre-trained in COCO dataset was retrained in this research. Transfer learning was used, the model was re-trained with the available data which contained four classes, trees, grass, road and sky. Transfer learning is a method whereby upper layers are fine-tuned with existing data to solve a new problem at hand. COCO dataset is an 80 class large data for object detection, segmentation and captioning.

The results presented in chapter four were tabulated to identify total segments, false positives, true positives, and false negatives for each image. The image number denoted the position of the image for image one to ten of the test data. The total number of segments included true and false positives for each image. For each of the five samples of training data, the outputs were tabulated in the same format but on different tables. The tables were used to make the results readable and compared correctly classified against incorrectly classified segments.

6.2 Results Evaluation

The results of a model trained with 100 images were presented in Figure 5.1 and Table 5.1. The total number of false positives in each image was lower than the total number of true positives. The compiled results of all the models were represented in table 5.7. As observed on the table the total number of objects for 20 test images was 121 and total segments including incorrectly classified segments were 84. The number of true positives was 68, the total number of false positives was 16 and 41 false negatives. These results mean that as much as the number of true positives increased, false positives were still high and the number of unallocated pixels in each image was high. To improve the results, the train set was increased with 200 images to 300 train images.

The model was re-trained with 300 images. The results produced after evaluation were presented in Figure 5.2 and Table 5.2. These results showed that true positives were higher than false positives. The total number of segments was 90 including incorrectly classified segments. False positives were reduced from 16 to 15 and the number of true positives increased from 68 to 75. The results were promising because false positives were reduced, false negatives reduced and true positives increased. When false positives were reduced by one, it means there were still incorrectly classified objects, it also shows that the model was not behaving well. A further step was taken to increase the train size and see what happens.

The evaluation of the model trained with 500 images produced much better results as compared to the previous models. As shown in Figure 5.3 and Table 5.3 a total of 89 segments was produced. With this model True positives increased from 75 to 82, false positives decreased from 15 to 7 and false negatives were reduced from 38 to 34. This means that the more data was increased the better the results were produced.

Finally the dataset with 500 images was increased to 700 images. During the training phase, an error was encountered, the model could not complete the training with the new data which was added, and it trained for a few global steps and stopped. The problem persisted for several weeks without a working solution. After a long time of waiting for the model to respond to the persisting problem, it was decided that the data should be duplicated with 200 images to make up 700 images in order to see if there were any changes. The model trained well and it was tested. The results yield from this model were completely different from what was expected. There were 83 segments, 72 true positives, 11 false positives, and 44 false

negatives. As compared to the previous model the number of segments and true positives dropped and the number of false positives and false negatives increased. This means that duplicating the images was not a good idea because, during the training phase each image was processed two times. To prove the assumption of images being processed two times, the following stage was carried out.

Lastly, 400 images were duplicated from sample 3 data which contained 500 images to make 900 images. The produced results were extremely bad than the previous model with 200 duplicates. The number of segments was 66, 51 false positives, 9 false positives. The number of segments and the number of true positives were lower as compared to the previous models. The presence of duplication led to unexpected results because there were 10 percent labeled images in each sample, however, the first three results proved that as data increased the results were improved until the fourth results. The evaluation measures that were used to evaluate the performance of the model are discussed in the following section.



6.3 False discovery rate, Precision and Recall

University of Fort Hare

Together in Excellence

The graph on figure 5.9 further mapped false discovery rate (FDR), precision, and recall against training sample size. False discovery rate (FDR) was used to predict the occurrence of false positives and compare the results when the model was trained with increasing sample size. The false discovery rate for 100 training images was 19%, 17% for 300 train images, 8% for 500 train images, 13% for 700 train images, and 15% for 900 train images. The rate of false positives decreased from 100 to 500 train images, and it increased at 700 train images and decreased at 900 train images. Precision was 81% at 100 train images, 83% at 300 train images, 92% at 500 train images, 87% at 700 train images and 85% at 900 train images. These results mean that precision increased with increasing sample size. The recall was 62% for 100 train images, 66% for 300 train images, 71% for 500 train images, 62% for 700 train images, and 71% for 900 images. It means that the recall also increased when data was increased, and otherwise when data was duplicated. False discovery rate, precision, and recall were affected by the duplication of data.

6.4 Summary

The main challenge was false negative during classification of segments because of the environment that images were captured from. The outdoor environment makes it difficult to segment images without a challenge. Data from CVonline could not all be used due to low computational resources. The model was only trained with less than 1000 images and it performed well under this small data and with low computational costs.



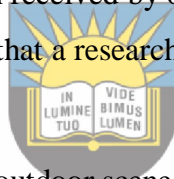
University of Fort Hare
Together in Excellence

7 Chapter Seven: Conclusion and Future Work

This research has been an empirical investigation to answer the critical question: how do the model complexity, size of training set, false negatives collectively affect the segmentation and classification performance. We made use of CVonline dataset for image segmentation and pre-trained R-CNN with VGG-16 to explore this question in-depth by performing several experiments. This closing chapter aims to conclude the results that were obtained. The chapter also provides a comprehensive summary of the highlights of the research. Further, the chapter is closed by describing areas of possible further exploration of this study.

7.1 Empirical Findings vs Research Questions

Empirical findings is the information received by observation and documentation whereas research questions are the questions that a research project sets out to answer.



- 1) Which methods are used for outdoor scene image segmentation?

University of Fort Hare

Together in Excellence

The research introduced the concept of image segmentation in Chapter 2. Image segmentation which includes, image classification and detection is presently very important in the computer vision field. It comes with a wide variety of benefits that are appealing in image processing. However, there exist challenges that hinder the algorithms especially in an outdoor environment. False negatives that are produced during image segmentation make it difficult to classify image segments according to their classes.

- 2) Which techniques can be used to reduce false negatives during the classification and segmentation of images?

The research firstly discussed the concepts of image segmentation in chapter 3. A few definitions were present to explain the concept. The use of CNNs was justified for segmentation and classification tasks because they reduce false negatives during the classification of image segments and they have the following advantages:

- Probability theory.

- Graphical structure.
- Ability to add more layers in a model.
- Its learning ability.

3) How can we improve the accuracy of image segmentation in an outdoor environment

The purpose of this research was to re-train R-CNN models with VGG-16 which was pre-trained using the COCO dataset. R-CNN was re-trained with data from CVonline dataset and it performed well. However there is a need for an increase in labeled image datasets that are available online and providing more virtualized computing resources. This model yields best results when observing the outcome of each sample size as they increase, it would perform better if all the images were labeled. When looking at all of the results based on increasing sample sizes the number of false positives decreased as the train sample increased and it was always lower than the number of true positives. Although the training sample size was very low, transfer learning improved the learning capability. Trees, grass, road, and sky were segmented and the number of false positives reduced as training data was increased. It can be concluded that the object segmentation problem can be tackled better with supervised CNN algorithms. There are still challenges regarding outdoor scene image segmentation.

7.2 Recommendations

During the course of the study of any field of interest, various questions will remain unanswered. With respect to this study, this research's findings have considerable areas that need further exploration. The study can be extended by investigating convolutional neural networks as an image segmentation algorithms. To empirically investigate whether it will provide better classification and segmentation performance. This study can be modified to study why VGG-16 works better than VGG-19 and other network architectures. Furthermore, investigations can be done on the low computational costs during the training phase instead of determining computational cost after training. The focus of this research was on trees,

grass, road, and sky. Therefore to improve the quality of this research, the algorithm should focus on every object that are present in an image.

The greatest challenge in image segmentation, classification, and detection is training data. It is therefore tedious as well as expensive to label large training samples required for optimum classification performance. With this in mind, further exploration can be done to determine how small training samples can be made to augment the classification performance of convolutional neural networks, and also collaboration should be made with other researchers to develop a large corpus dataset for training and testing image segmentation algorithms.

7.3 Conclusion

Convolutional neural networks have been applied in a wide range of applications. The focal point of this research has been to apply them in segmenting images in outdoor environments. Literature provides numerous advantages that make convolutional neural networks to be the better segmentation algorithm. They have proven to be excellent in classification and segmentation tasks. This research embarked on the analysis of the impact of structural complexity, training sample size and score functions on the classification performance in outdoor scene segmentation. The hope of the researcher is that this study will provide choices for researchers intending to use a convolutional neural network in this field of study. Poor choices of the sample size, discretization technique will have adverse effects on segmentation and classification accuracy of the chosen algorithm. By virtue of this research, CNNs should be recommended since the parameters involved have been analysed and presented.

8 References

- [1] T. Zuva and S. Ngwira, "Image Segmentation , Available Techniques , Developments and Open Issues," *Can. J.*, vol. vol.2, No., no. January 2011, 2015.
- [2] L. Maddalena and A. Petrosino, "The SOBS algorithm: What are the limits?," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 21–26, 2012.
- [3] V. Singh, D. Girish, and A. Ralescu, "Image understanding - A brief review of scene classification and recognition," *28th Mod. Artif. Intell. Cogn. Sci. Conf. MAICS 2017*, pp. 85–91, 2017.
- [4] *Motion Detection in Static Backgrounds.* .
- [5] A. Bachoo, B. Duvenhage, and J. De Villiers, "PRISM Project List," 2015.
- [6] Y. Xu, J. Dong, B. Zhang, and D. Xu, "Background modeling methods in video analysis: A review and comparative evaluation," *CAAI Trans. Intell. Technol.*, vol. 1, no. 1, pp. 43–60, 2016.
- [7] R. Szeliski, *Computer Vision : Algorithms and Applications*. Keith Price, 2010.
- [8] T. Lin, C. L. Zitnick, and P. Doll, "Microsoft COCO : Common Objects in Context," pp. 1–15.
- [9] T. Hackel, J. D. Wegner, and K. Schindler, "Fast Semantic Segmentation of 3D Point Clouds With Strongly Varying Density," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. III–3, pp. 177–184, 2016.
- [10] N. Dabhi and P. Hirenmewada, "A Review on Outdoor Scene Image Segmentation," *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 6, pp. 5419–5423, 2012.
- [11] C. Cheng, A. Koschan, C. Chen, D. L. Page, and M. A. Abidi, "Outdoor Scene Image Segmentation Based on Background Recognition and Perceptual Organization," no. September, 2011.
- [12] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextronBoost for Image Understanding : Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture , Layout , and Context," 2007.
- [13] D. Link, "Object categorization by learned universal visual dictionary pdf," 2008.
- [14] R. Vieux *et al.*, "Segmentation-based multi-class semantic object detection To cite this version : HAL Id : hal-00572863," 2011.
- [15] T. Maintz, "Chapter 10. Segmentation," *Digit. Med. Image Process.*, 2005.
- [16] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," 2011.
- [17] J. Verne, "Image Pre-Processing," *Semant. Sch.*, pp. 35–74, 2016.
- [18] A. Taneja, L. Ballan, and M. Pollefeys, "Modeling dynamic scenes recorded with freely moving cameras," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6494 LNCS, no. PART 3, pp. 613–626, 2011.
- [19] S. Jeeva and M. Sivabalakrishnan, "SURVEY ON BACKGROUND MODELING AND FOREGROUND DETECTION FOR," *Procedia - Procedia Comput. Sci.*, vol.

- 50, pp. 566–571, 2015.
- [20] M. İlsever and C. Ünsalan, “Two-Dimensional Change Detection Methods,” pp. 7–22, 2012.
- [21] D. Redundancy, “Chapter 2 Digital Image Compression,” pp. 4–15.
- [22] M. Tejasvini, S. M. Riyajoddin, and M. J. Reedy, “APPENDIX-II Outdoor Scene Image Segmentation- State of the Art,” pp. 546–553.
- [23] J. Shi, D. Martin, C. Fowlkes, and E. Sharon, “Tutorial Graph Based Image Segmentation,” *Image (Rochester, N.Y.)*, 2010.
- [24] A. D. Costea and S. Nedevschi, “Multi-class segmentation for traffic scenarios at over 50 FPS Multi-Class Segmentation for Traffic Scenarios at Over 50 FPS,” no. April 2016, 2014.
- [25] P. Doll and S. Belongie, “Supervised Learning of Edges and Object Boundaries.”
- [26] “Introduction to Convolutional Neural Networks,” 2018.
- [27] G. Roig, “Team and Sponsors.”
- [28] K. J. Cios and M. E. Shields, “The handbook of brain theory and neural networks,” *Neurocomputing*, vol. 16, no. 3, pp. 259–261, 1997.
- [29] A. Bhandare, M. Bhide, P. Gokhale, and R. Chandavarkar, “Applications of Convolutional Neural Networks,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 7, no. 5, pp. 2206–2215, 2016.
- [30] H. A. Atabay, “a Convolutional Neural Network With a New Architecture Applied on Leaf Classification,” vol. 7, no. 5, pp. 326–331, 2016.
- [31] F.-F. Li, J. Johnson, and S. Yeung, “CS231 Lecture 09 : CNN Architectures,” *Cs 231*, pp. 1 – 101, 2017.
- [32] C. Szegedy *et al.*, “Going deeper with convolutions,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07–12-June, pp. 1–9, 2015.
- [33] S. Wiehman, “Investigating Fully Convolutional Networks for Bio-image Segmentation,” no. March, 2018.
- [34] T. Sainath, A. R. Mohamed, B. Kingsbury, and B. Ramabhadran, “Deep convolutional neural networks for LVCSR,” *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 8614–8618, 2013.
- [35] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst.* 25, pp. 1–9, 2012.
- [36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.
- [37] I. Sutskever and G. E. Hinton, “AlexNet,” 2012.
- [38] S. Regina Lourdhu Suganthi, M. Hanumanthappa, and S. Kavitha, “Event Image Classification using Deep Learning,” *ICSNS 2018 - Proc. IEEE Int. Conf. Soft-Computing Netw. Secur.*, no. August, 2018.

- [39] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," pp. 1–14, 2014.
- [40] G. Wang, Y. Sun, and J. Wang, "Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning," *Comput. Intell. Neurosci.*, vol. 2017, 2017.
- [41] W.-S. Jeon and S.-Y. Rhee, "Plant Leaf Recognition Using a Convolution Neural Network," *Int. J. Fuzzy Log. Intell. Syst.*, vol. 17, no. 1, pp. 26–34, 2017.
- [42] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimed. Tools Appl.*, pp. 1–17, 2017.
- [43] V. Badrinarayanan, A. Kendall, R. Cipolla, and S. Member, "SegNet : A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," vol. 39, no. 12, pp. 2481–2495, 2017.
- [44] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and C. V. Van, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation."
- [45] P. Kamavisdar, S. Saluja, and S. Agrawal, "A survey on image classification approaches and techniques," *Int. J. Adv. ...*, vol. 2, no. 1, pp. 1005–1009, 2013.
- [46] Y. Benezeth, P. Jodoin, B. Emile, and C. Rosenberger, "Comparative study of background subtraction algorithms To cite this version :," 2012.
- [47] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video Processing From Electro-Optical Sensors for Object Detection and Tracking in a Maritime Environment: A Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 1–23, 2017.
- [48] J. Lee and M. Park, "An Adaptive Background Subtraction Method Based on Kernel Density Estimation," pp. 12279–12300, 2012.
- [49] D. Hall *et al.*, "Comparison of target detection algorithms using adaptive background models," *IEEE Int. Work. Vis. Surveill. Perform. Eval. Track. Surveill.*, no. 1, pp. 113–120, 2005.
- [50] C. Gibbes, S. Adhikari, L. Rostant, J. Southworth, and Y. Qiu, "Application of object based classification and high resolution satellite imagery for Savanna ecosystem analysis," *Remote Sens.*, vol. 2, no. 12, pp. 2748–2772, 2010.
- [51] I. No and S. Yadav, "Available Online at www.ijarcs.info An Advanced Motion Detection Algorithm with Video Quality Analysis for Video Surveillance Systems," vol. 5, no. 8, pp. 186–190, 2014.
- [52] I. Huerta, *Foreground Object Segmentation and Shadow Detection for Video Sequences in*. 2010.
- [53] A. Nguyen and B. Le, "3D point cloud segmentation: A survey," *IEEE Conf. Robot. Autom. Mechatronics, RAM - Proc.*, no. November, pp. 225–230, 2013.
- [54] I. Classification, "Chapter 4 : Classification & Prediction."
- [55] J. Kim, B.-S. Kim, and S. Savarese, "Comparing Image Classification Methods: K-Nearest-Neighbor and Support-Vector-Machines," *Appl. Math. Electr. Comput. Eng.*, pp. 133–138, 2012.

- [56] Z. Q. Zhao, B. J. Xie, Y. M. Cheung, and X. Wu, "Plant leaf identification via a growing convolution neural network with progressive sample learning," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9004, pp. 348–361, 2015.
- [57] D. Jurafsky and J. Martin, "Hidden Markov Models," *Speech Lang. Process.*, no. Chapter 20, p. 21, 2017.
- [58] P. Pawara, E. Okafor, O. Surinta, L. Schomaker, and M. Wiering, "Comparing Local Descriptors and Bags of Visual Words to Deep Convolutional Neural Networks for Plant Recognition," *Proc. 6th Int. Conf. Pattern Recognit. Appl. Methods*, pp. 479–486, 2017.
- [59] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A Survey on Deep Transfer Learning," pp. 1–10.
- [60] N. Vamsi and K. Jasti, "A literature review of empirical research methodology in lean manufacturing," no. July 2014, 2015.
- [61] R. Girshick, "Fast R-CNN."
- [62] "Tensorflow detection model zoo," 2018. .
- [63] "tensorflow/models," 2018. .
- [64] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou, "Multi-scale object detection in remote sensing imagery with convolutional neural networks ISPRS Journal of Photogrammetry and Remote Sensing Multi-scale object detection in remote sensing imagery with convolutional neural networks," *ISPRS J. Photogramm. Remote Sens.*, no. May, 2018.
- [65] S. Tammina, "Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images," no. October, 2019.
- [66] M. M. Zhang, J. Choi, K. Daniilidis, M. T. Wolf, and C. Kanan, "VAIS : A Dataset for Recognizing Maritime Imagery in the Visible and Infrared Spectrums," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 10–16, 1972.
- [67] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Underst.*, vol. 122, pp. 4–21, 2014.
- [68] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, "Compression of Deep Convolutional Neural Networks for Fast and Low Power Mobile Applications," pp. 1–16, 2015.