# Computer vision based detection and identification of potato blemishes

Michael Barnes

School of Computer Science

University of Lincoln

A thesis submitted for the degree of

*Doctor of Philosophy (PhD)*

May 2012

# Abstract

This thesis addresses the problem of automatic detection and identification of blemishes in digital images of potatoes. Potatoes are an important food crop, with clear unblemished skin being the main factor affecting consumer preference. Potatoes with defects, diseases and blemishes caused by otherwise benign (to human) skin infections, are strongly avoided by consumers. Most potatoes are sorted into different grades by hand, with inevitable mistakes and losses.

The currently deployed computer vision systems for sorting potatoes require manual training and have limited accuracy and high unit costs. A further limitation of typical machine vision systems is that the set of image features for pattern recognition has to be designed by the system engineer to work with a specific configuration of produce, imaging system and operating conditions. Such systems typically do not generalise well to other configurations, where the required image features may well differ from those used to design the original system.

The objective of the research presented in this thesis is to introduce an automatic method for detecting and identifying blemishes in digital images of potatoes, where the presented solution involves classifying individual pixels. A human expert is required to mark up areas of blemishes and non-blemishes in a set of training images. For blemish detection, each pixel is classified as either blemish or non-blemish. For blemish identification, each pixel is classified according to a number of pre-determined blemish categories. After training, the system should be able to classify individual pixels in new images of previously unseen potatoes with high accuracy.

After segmenting the potato from the image background, a very large set of candidate features, based on statistical information relating to the colour

and texture of the region surrounding a given pixel, is first extracted. The features include statistical summaries of the whole potato and local regions centred on each pixel as well as the data of the pixel itself. Then an adaptive boosting algorithm (AdaBoost) is used to automatically select the best features for discriminating between blemishes and non-blemishes. The AdaBoost algorithm (Freund and Schapire, 1999) is used to build a classifier, which combines results from so-called “weak” classifiers, each constructed using one of the candidate features, into one “strong” classifier that performs better than any of the weak classifiers alone. With this approach, different features can be selected for different potato varieties, while also handling the natural variation in fresh produce due to different seasons, lighting conditions, etc.

For blemish detection, the classifier was trained using a subset of pixels which had been marked as blemish or non-blemish. Tests were done with the full set of features, “lesion experiments” were carried out to explore the impact of removing specific feature types, and experiments were also carried out on methods of speeding up classification both by restricting the number of weak classifiers and restricting the numbers of unique candidate features which can be used to produce weak classifiers. The results were highly accurate with visible examples of disagreement between classifier output and markup being caused by human inaccuracies in the markup rather than classifier inaccuracy.

For blemish identification, a set of classifiers were trained on subsets of pixels marked as each blemish class against a subset of pixels drawn from all other classes. For classification, each pixel was tested with all classifiers and assigned to the classifier which returned the highest confidence of a positive result. Experiments were again performed with methods of speeding up classification as well as lesion experiments. Finally, to demonstrate how the system would work in an industrial context, the classification results were summarised for each potato, providing a high overall accuracy in detecting the presence or absence of significant blemish coverage for each blemish type.

To the memory of Fred Cave

# Acknowledgements

Thanks to the staff of the Potato Council Ltd.

Thanks to all staff at the University of Lincoln, especially my supervisors Tom Duckett and Grzegorz Cielniak. Thanks to Jose Gonzalez-Rodriguez for his invaluable advice regarding SCOPUS and Hongying Meng for translating Chinese papers for me. Thanks also to the staff at the Potato Council Ltd, particularly to Graeme Stroud for his assistance with marking up potatoes, Adrian Cunnington, Jeff Peters and Glyn Harper.

Thanks to everybody involved with the conferences Image and Vision Computing, New Zealand and the International Conference on Computer Vision and Graphics in Poland.

Countless others know who they are.

*"A straight line may be the shortest distance between two points, but it is by no means the most interesting."* – Robert Holmes

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

This thesis investigates the problem of detecting and identifying blemishes on the surface of potatoes using computer vision. A novel solution is developed, in which machine learning classifiers are trained to detect and classify common blemish types affecting potatoes. The key innovation of this approach is that the system developed is "trainable" – allowing it to be applied to different potato varieties and different customer requirements, while also handling the natural variation in fresh produce due to different seasons, lighting conditions, etc. In each case, the system will automatically select the best image features (from a large set of candidate features) to distinguish between the different blemish types, by learning from a set of sample images that have been marked up by a human expert. This innovation should help to mitigate the high costs of currently deployed industrial systems, requiring only low-cost "off-the-shelf" components for digital imaging and minimal set-up costs for training and retraining the system as required. As well as describing the technical developments, this thesis also presents extensive experiments using real world datasets (collected in collaboration with researchers at Sutton Bridge Crop Storage Research and the Potato Council, a division of The Agriculture and Horticulture Development Board) to evaluate the performance of the system developed.

## 1.1 Motivation

Potatoes (Solanum tuberosum), with an estimated worldwide production of over 374 million tonnes in 2011 (Food and Agriculture Organisation, 2011), account for 70-

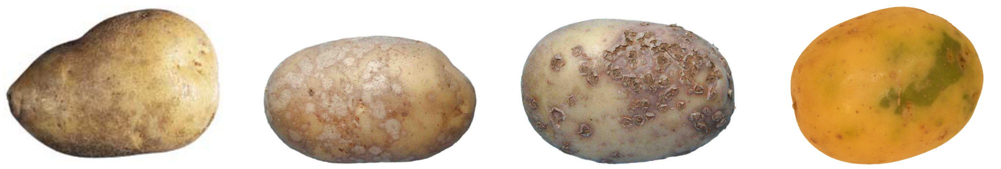
80% of the carbohydrate consumed in the UK. For the fresh market the main factor affecting consumer preference is physical appearance and, to maximise return, great effort is expended ensuring that the appearance best matches a particular market. Most potatoes are still sorted by hand. Problems with manual sorting include the subjectivity, fatigue and high cost of human inspectors (Narendra and Hareesh, 2010), while currently deployed artificial vision systems require manual calibration and may have limited accuracy, especially with varieties that the systems have not been pre-calibrated to accommodate. So there is a clear motivation for providing a system which can be quickly and easily trained to work with different lighting conditions, different customer requirements, new varieties of potatoes, and new blemishes as they become relevant. Potato blemish diseases present a variety of different coloured, sized and textured symptoms on the skin surface. Such diverse visual information provides us with a rich source of indicators (or features) that can be used for training an automatic blemish detector.

Properly sorted potatoes can command a much higher price overall. A low quality potato can be separated out and sent to a customer who requires only raw potato material, e.g. for the production of chips or reformed potato goods, which can be made by cutting out the bad parts of the potato and cutting up or mashing the remainder. A good potato, meanwhile, will sell for a much higher price to a customer who would not be willing to pay for the low quality product (Bowbrick, 1982). Individual retailers often place specific requirements on which blemishes are allowed and the permitted coverage per potato, typically expressed in terms of the percentage of surface area affected.

Few systems exist for the purpose of automatically grading potatoes in this way. Examples of industrially-fielded systems include the Odenberg Titan Sorter, which takes two photographs of each potato, front and back, in free-fall past a camera. The Herbert Upgrader takes multiple photographs of potatoes rolling along a conveyor belt designed to encourage this motion. The information from these images can then be combined to produce summary statistics for a single potato, if no single image displays sufficient blemish to automatically classify that potato as a reject. The existing industrially-field systems use pre-programmed rules to reject the most obvious cases of blemish, such as anomalously coloured pixels, particularly dark pixels or long edges, since the only long edges in a good potato should be the outline. However, these systems do not attempt to identify the particular diseases or processes underlying the observed blemishes (Barnes et al., 2010). Identification of blemish causes is an original contribution of this thesis, as documented in Chapter 6. The existing industrially-fielded systems are pre-programmed and require calibration for different species, e.g. white and red potatoes (Maf Roda Group, 2008), making it difficult to adapt them for new potato varieties or variations resulting from, for example, a particularly wet or dry growing season (R.J. Herbert Engineering Ltd, 2008). This thesis introduces a trainable vision system, allowing the system to be adapted to new varieties or new circumstances much more easily.

There are a number of diseases affecting potato tubers that, although superficial and of no health consequence to humans, strongly and negatively influence consumer choice. These include black dot, silver scurf, powdery scab, common scab, and skin spot. The fungal species of Rhizoctonia solani also causes significant skin blemish manifesting as black scurf and elephant hide. The causes of blemish diseases are known (Fiers et al., 2010). However, customer preference may be for susceptible potato varieties, and different environmental and field conditions during cultivation favour different diseases. These inevitably lead to some crops being infected in a generally unpredictable fashion. Other forms of blemish include physical damage, e.g. growth cracks, mechanical and slug damage as well as physiological changes, e.g. greening and sprouting. Greening and sprouting are examples of a healthy potato tuber attempting to grow. However, there are serious concerns (Alexander et al., 1948) about the edibility of greened potatoes, which contain a harmful chemical named solanine, while in the case of sprouting the texture and flavour of the potato flesh is negatively affected. These features are therefore treated as any other blemish. Potatoes and their tubers are also susceptible to more significant diseases, in particular blight, and other fungal and bacterial rots.

In thesis, a subset of the most commonly occurring blemishes was investigated, and grouped into four distinct categories or classes, namely (i) black dot, (ii) silver scurf, (iii) scabs, and (iv) greening and sprouting. Different types of scabs were merged into a single category, and similarly, greening and sprouting were also merged into a single category, due to their similarity in appearance and the limited frequency of these conditions in the data sets used in the experiments presented. Figure 1.1 shows examples of these classes, representative of the blemish types investigated in this thesis.



(a) A potato blemished by black dot.

(b) A potato blemished by silver scurf.

(c) Potato blemished by powdery scab.

(d) Physiological greening of the skin of a potato.

**Figure 1.1:** Examples of the potato blemish types investigated in this thesis.

## 1.2 Overview of this thesis

This thesis describes the research underpinning the development of a system to detect and identify blemishes in potatoes using computer vision and machine learning techniques.

**Chapter 2** begins with an introduction to computer vision systems for food and agriculture. The topics covered include pattern classification, which is the problem of identifying the class to which new observations belong, on the basis of a training set of data containing observations whose class is known (Duda et al., 2000). In this thesis, pattern classification techniques from the field of Machine Learning (a subfield of Artificial Intelligence) are applied, referred to subsequently as “machine learning classifiers” or simply “classifiers”. In this work, the input to a classifier consists of image “features” comprising information extracted from an area surrounding a particular pixel in an image, relating to the visual appearance of the potato. The classes assigned to each pixel by the classifiers can be either “blemish” versus “non-blemish” in the blemish detection stage, or one of the four blemish classes described above when identifying blemishes.

The subsequent chapters describe the development of the computer vision system presented in this thesis, following the general stages shown in Figure 1.2.

**Chapter 3** describes the imaging set-up used for this research, including details of the camera, lighting, and the image pre-processing steps, which include removal of the image background to isolate the foreground objects/potatoes (the process of image segmentation) and scaling of the image to a chosen resolution. The approach for providing the correct category (class) data required for training the classifiers is also described, whereby an image is marked manually by a human expert so that the object pixels are labelled as either "good potato" or the particular blemish type. The term "ground truth" data is sometimes used to refer to the correct category data for individual pixels, following the conventions used in the field of aerial imaging to relate image pixels to actual features or materials (quite literally) on the ground. Examples of manually marked-up ("ground truth") images can be seen in the second column of Figure 1.3.

**Chapter 4** describes the subsequent processes of feature extraction, feature selection and pattern classification. The novelty of the approach presented in this chapter involves the use of an adaptive boosting algorithm called AdaBoost (Freund and Schapire, 1999) to automatically select good features for a particular pattern classification task. A minimal set of features is selected from a very large set of *candidate features*, which measure statistical properties of the colour and texture distribution of the image region surrounding a given pixel. After the system training phase has been completed, only these *selected features* then need to be extracted for the new images presented to the system, and are then used to classify object pixels corresponding to potatoes.

**Chapter 5** describes the application of this system to the problem of blemish detection, i.e. classification of individual pixels as "blemish" or "non-blemish". Examples of the classifier output can be seen in the fourth column of Figure 1.3, as well as so-called error images in the third column showing where the output of the trained classifier differs from the "ground truth" data provided by a human expert. Experiments are presented to evaluate the performance on the approach using two different data sets corresponding to white and red potato varieties, respectively.

**Chapter 6** describes the subsequent application of the same system to the problem of blemish identification, i.e. further classification of the individual pixels already identified as "blemish" according to the different blemish types (using the four classes described previously in Section 1.1). Again, experiments are presented to evaluate the performance of the approach using the same data sets as in the previous chapter.

Finally, **Chapter 7** concludes the thesis, presenting also open questions, limitations of the system, and possible topics for future work.

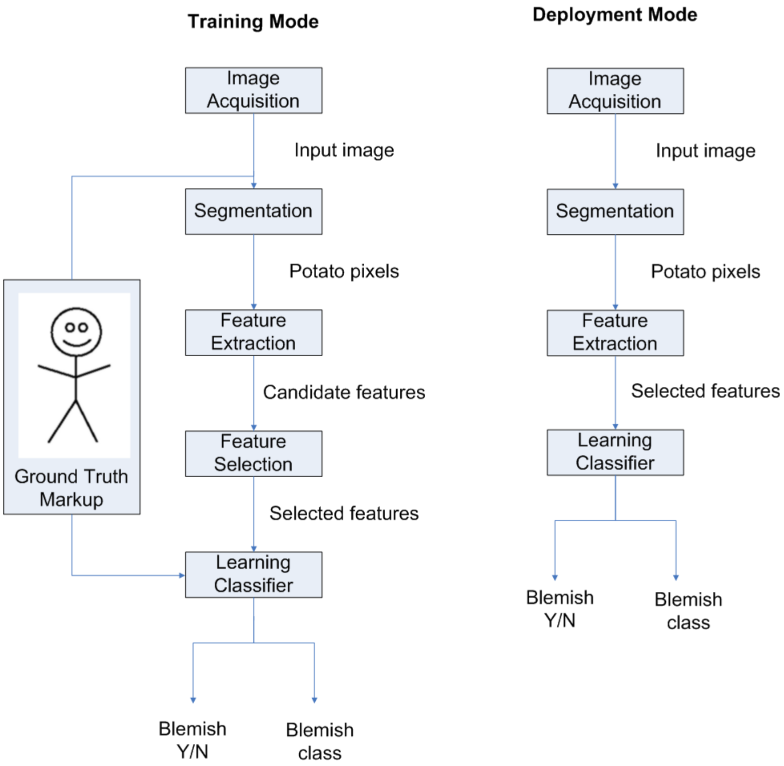


**Figure 1.2:** Overview of the computer vision system presented in this thesis. Left: during the training phase, a human expert is required to mark-up the sample image data, in order to give the machine learning classifier the correct answers (blemish categories), and the system also learns to select the best image features for image pixel classification on a particular data set. Right: the trained system is then deployed, and will automatically carry out the extraction of the previously selected features and blemish classification, including both detection and identification of the common blemish types.

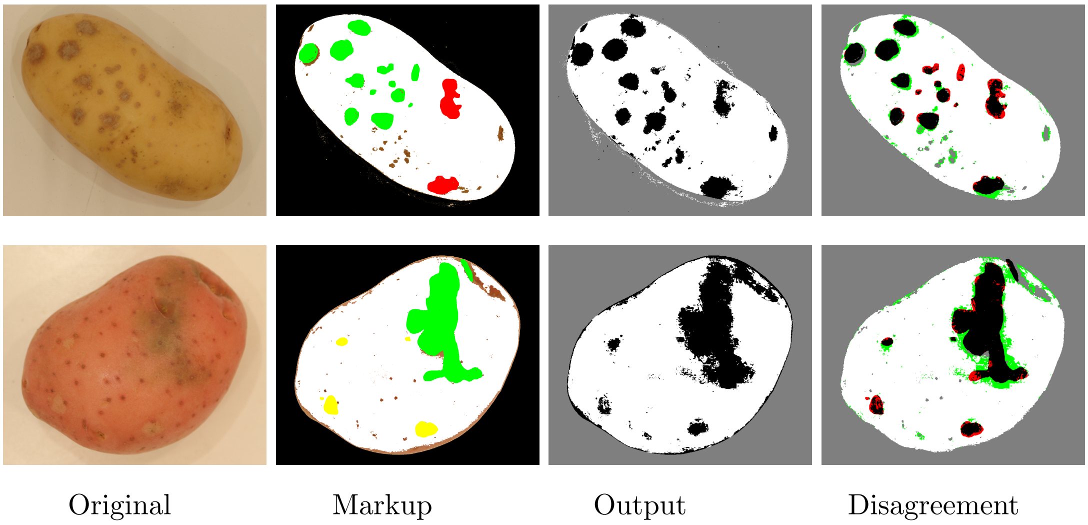


**Figure 1.3:** Example images, showing two examples of potatoes analysed by the blemish detection system, one per row. From left to right: first an original photograph, then a "ground truth" image labelled manually by a human expert, then the outputs of the trained system, without any human editing, with blemishes in black and good potato in white. The final image shows an error image, showing false positive results in red and false negative results in green, for the detection of blemish.

## 1.3 Contributions

This thesis presents a novel system for detection and identification of potato blemishes in digital images, allowing coverage to be summarised by blemish type and potatoes to be graded for quality according to customer requirements. The specific scientific contributions of this thesis include:

- Automatic detection and identification of common blemishes in images of potatoes, by application of machine learning classifiers trained from sample images marked by human experts.
- Automatic selection of image features for blemish classification from a very large set of candidate features, based on statistical information relating to the colour and texture of the region surrounding a given pixel.
- A comprehensive, quantitative evaluation of the whole system using real world data sets.

## 1.4 Publications

Some of the work described in this thesis has been presented in journals and conferences. Below is a complete list of publications arising during the course of this PhD study.

- Barnes, Michael and Cielniak, Grzegorz and Duckett, Tom (2010). Minimalist AdaBoost for blemish identification in potatoes. In: Proceedings of the International Conference on Computer Vision and Graphics 2010, 20-22 September 2010, PJIIT - Polish-Japanese Institute of Information Technology. pp. 209-216
- Barnes, Michael and Duckett, Tom and Cielniak, Grzegorz (2009). Boosting minimalist classifiers for blemish detection in potatoes. In: Proceedings of the conference of Image and Vision Computing New Zealand (IVCNZ), 23-25 November 2009, Wellington, New Zealand. pp. 397 - 402
- Barnes, Michael and Duckett, Tom and Cielniak, Grzegorz and Stroud, Graeme and Harper, Glyn (2010). Visual detection of blemishes in potatoes using minimalist boosted classifiers. Journal of Food Engineering, Vol. 98 No. 3. pp. 339-346. ISSN 0260-8774

# 2

# Background

This chapter contains an overview of significant concepts relevant to the use of computer vision within the field of food produce production, grading and monitoring. The current state of the art is explored along with key issues in computer vision relating to the processes of image acquisition, preprocessing, segmentation, feature extraction and classification.

## 2.1 Destructive inspection methods

Other existing systems for analysis of potato blemishes include destructive methods. These include ELISA (Enzyme-linked immunosorbent assay) testing, whereby a dye or similar substance is activated if a key antibody reacts with a specific pathogen. ELISA is used by De Haan and van den Bovenkamp (2005), along with Polymerase Chain Reaction (PCR), which enhances the DNA of the pathogen present. The DNA can then be analysed for the presence of genes specific to a pathogen of interest. These methods were compared to the effectiveness of inspecting visually by microscope. Tests were done on 103 samples of 5 tubers from the harvest of 2003 along with another 15 samples from the harvest of 2002.

The Potato Operation (Pun et al., 1991) introduced an application of computer vision into an ELISA-based potato analysis process in order to to automate the selection and removal of potato pulp. This was done by locating the “germ” within the eye of the potato, which will eventually form a sprout, since they were interested in detecting diseases which are far more likely to infect the eye than other parts of the potato.

Imported seed potatoes were assessed by Tsror et al. (2012), who again used ELISA and PCR testing to detect the presence of Dickeya bacteria, the pathogen responsible for the slow wilt and blackleg diseases in potato plants. Sub-samples of 50 tubers were collected from each of 277 batches from 2006-2010. Tests using PCR provided an accuracy of 74% in predicting the presence of the diseases in field crops grown from the seed crop, with a false negative rate of only 1.3%. ELISA testing produced a higher 83.8% accuracy but a false negative rate of 6.5%, meaning that while PCR testing falsely rejected more batches, it nonetheless resulted in improved removal of diseased potatoes compared to ELISA.

In general, destructive methods can be used to provide an estimate of the quality of a crop based on a small subset of the crop. This introduces many limitations, including the number of times that a set of potatoes can be tested, since more potatoes are destroyed every time. This increases waste, especially where supermarkets want to test crops separately from farmers who may have different testing standards. A non-destructive test is therefore cheaper in terms of waste, as well as being preferable since potentially every potato in the crop can be inspected on-line.

## 2.2 Overview of computer vision systems for food product analysis

This research looks into the application of computer vision methods to classify potato blemishes by type and by coverage. The design of a computer vision solution requires that a number of factors be considered for different stages of the solution. As indicated in Figure 2.1, the typical stages of a computer vision system for food analysis include image acquisition, preprocessing, segmentation, feature extraction and classification. Some of the major issues relating to each stage are discussed below.

### 2.2.1 Image acquisition

The first stage in any computer vision based system is a method of obtaining the input data, consisting of digital images obtained from a camera or other scanning device. The main factors involved in designing such a system include the choice of image capture sensor, the lighting and the portion of the electromagnetic spectrum being investigated.

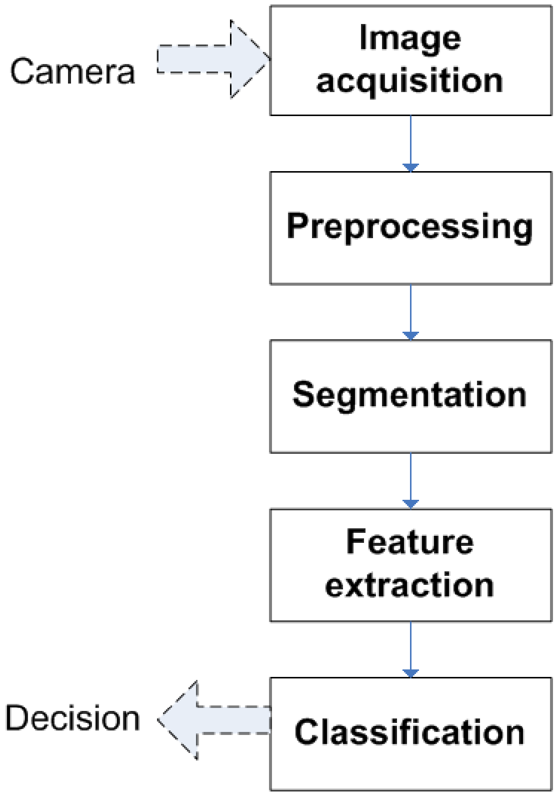


**Figure 2.1:** The general structure of a computer vision classification system and its typical stages.

A large range of specialised imaging sensors exist, from X-ray and radar systems to visible light or multispectral cameras. For this thesis the decision was made to work with an off-the-shelf visible light camera to provide a low-cost solution, which simulates the grading processes currently in use by trained quality control staff in the industry.

The means of illuminating the object under investigation can often be the one of most important choices. An example of a judicious choice of lighting helping to solve the major part of a computer vision problem is described by Uthaisombut (1995), who used custom lighting to highlight anomalies in cherries, having observed that undamaged cherry skin did not reflect green light. Damaged cherries and stems did reflect green light, allowing a simple solution based on colour thresholding of the image into areas with low levels of green reflectance that should appear in cherries and areas with high levels of green reflectance that should not appear in cherries. Damaged cherries and non-cherries could then be identified by the level of green reflectance in the corresponding image region.

A system intended to operate in changing lighting conditions, such as a system that monitors produce in outdoor environments, needs additional software considerations to allow a classifier to generalise over different lighting conditions. In contrast, a system operating indoors, e.g. over a conveyor belt, makes it easier to control how the subjects are illuminated rather than trying to compensate for changes in illumination in the system software.

As well as lighting conditions, the portion of the electromagnetic spectrum detected by a camera can be an important consideration. The majority of cameras detect light of wavelengths between 400 and 700 nanometers, separated into three different bands, red green and blue. These bands correspond to the wavelengths and colour bands detected by the cones and rods of the human eye. However, there can be sound reasons to work outside the visible light spectrum, or to work with customised wavelengths within the visible spectrum. Hyperspectral cameras, or spectrographs, are devices which can record light at a high number of different frequencies simultaneously. These devices can be used to help choose which frequencies of light to incorporate into a system. The hyperspectral camera itself is likely to have a frame rate too slow to integrate into a real-time grading system, since there are limits to the number of frequencies which can be captured at once, although current innovations are increasing the number of frequencies per snapshot (Gorman et al., 2010).

Methods of capturing light at different spectral lengths vary, with some sensors capturing the light directly on a custom sensor, while others use a wideband light sensor behind a filter which blocks all light not of the wavelength of interest. The result is effectively the same, but the filter option is usually easier to customise. Other forms of filters can include polarisation filters, which will eliminate all vibration of the electromagnetic waveform except in the direction in which the filter is oriented. This technique is often used to reduce glare since reflected light tends to vibrate most in a direction close to Brewster's angle (Lakhtakia, 1989). Lighting and light wavelengths in particular are significant considerations for any computer vision system, but in this project there was a desire to remain as hardware-independent as possible, so this project included an image acquisition set-up, described in Section 3.1, to provide an even white illumination and an off-the-shelf RGB camera.

### 2.2.2 Preprocessing

Once an image has been acquired, pre-processing algorithms can be applied to improve the image quality, for example to highlight important features or to remove noise (Gonzalez et al., 2004). Examples of noise reduction can include blurring of an image to reduce noise produced by randomness in the world or the sensor, or scaling the image to reduce the impact of individual pixels which differ greatly from their immediate neighbours. Image sharpening to increase the prominence of edges and other texture details can be used on its own or in combination with blurring for noise reduction. The choice of colour space can also affect how clearly individual features can be detected.

The choice of colour space is a major part of preprocessing. Colour is an interpretation of the frequencies and amplitude of a light wave. The human eye interprets these frequencies in the red, green and blue frequency ranges. Depending on the task, it can be preferable to interpret light using a different colour model. Several colour models and their applications are described below.

The RGB colour space is based on the principle that every visible colour can be produced from a mix of the three primary colours of red, green and blue, inspired by the colour-sensitive cells in the human eye. Used for producing colour images in computer monitors by means of red, green and blue light sources, this has become a popular colour model in computer imaging as it does not need converting to display on screen.

The colour model can also be normalised in order to remove the brightness element. This is done by turning the absolute values into fractions totalling 1. $r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B}$

The HSI colour space uses the three channels hue, saturation and intensity. These channels represent concepts that are easy for a human to understand. The hue, which represents the dominant frequency of the light, is referred to as the pure colour, while saturation represents the amount of white light mixed in with the pure colour and the intensity represents the overall brightness of the colour (Gonzalez et al., 2004).

The CIELAB colour space, also named L*a*b* or simply Lab (though Lab is ambiguous) is a colour space intended to be close to the human perception of colours. CIELAB uses three channels. L* represents the luminosity, or brightness of a colour, a* represents its location on a scale from red to green and b* on a scale of blue to yellow.

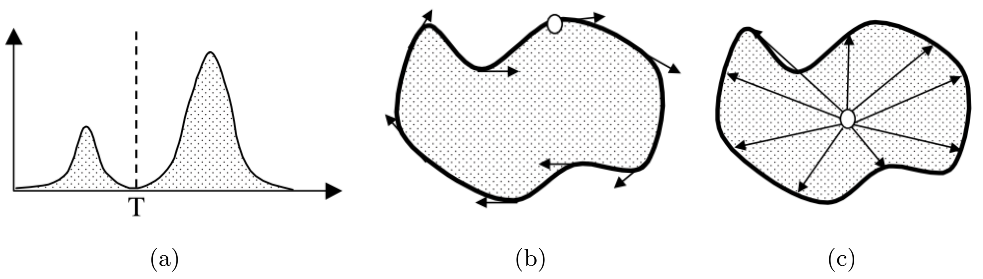


**Figure 2.2:** Typical segmentation techniques: (a) thresholding, (b) edge- based segmentation and (c) region-based segmentation (Brosnan and Sun, 2004)

The use of multiple colour spaces is common, often mixing RGB with another chromaticity-based colour space, or mixing normalised and non-normalised versions of the same colour space. This allows features to be observed that might only be obvious in one such type of space.

A major benefit of different colour spaces is the access to raw sensor data, brightness and chromaticity, which is the colour independent of brightness. Chromaticity in particular is advantageous for handling variations in lighting. We chose to ensure these three qualities were present in the image information used in this thesis, using the colour spaces RGB, normalised RGB and the intensity channel I, defined as $I = \frac{R+G+B}{3}$, giving a total of seven channels for subsequent processing.

### 2.2.3 Segmentation

Image segmentation is a method of partitioning an image into regions that have a strong correlation with objects or areas of interest (Brosnan and Sun, 2004). In this thesis, these areas are the potato and background areas. Unay and Gosselin (2006) provide another example, where the stem and calyx, or flower, of apples were segmented in order to remove them from consideration by a classifier which distinguished between good and bad apple skin.

Three of the more common types of segmentation are, thresholding, edge-based and region-based segmentation, as shown in Figure 2.2.

Thresholding is a simple form of segmentation. From a greyscale image, thresholding can be used to create binary images (Gonzalez et al., 2004). This can be one of the fastest segmentation algorithms, since it does not require any additional preprocessing. Edge-based segmentation involves the detection of discontinuities in the grey level, colour or texture of pixels. Region-based segmentation involves the grouping together of similar pixels to form regions representing single objects within the image. The criteria for like-pixels can be based on grey level, colour and/or texture.

Background subtraction is a common method of region-based segmentation, where the system can identify foreground objects versus background, for example, because the background has a known appearance or the background is stationary while the foreground objects are moving. In a system such as a potato grader the physical material and colour of the background can be chosen specifically to provide a strong contrast with potatoes. Industrial systems such as the Herbert Upgrader often use blue conveyor belts for this purpose, because natural food products are rarely blue. Pixels with a high blue value can then be removed by thresholding. Blasco et al. (2008) compared a variety of different coloured backgrounds for segmentation of pomegranate seeds, finding that blue or light green colours were most effective.

It is also possible for background removal to be aided by the use of sensor fusion, such as an RGB-D image from the popular Kinect sensor, which includes a depth channel (D), allowing items to be thresholded by distance or by locating a planar surface on which an object of interest is placed (Heimann and Meinzer, 2009). A similar option is to combine images under changing lighting conditions when the object is much closer to the camera than the background, which causes a more pronounced effect on the illumination of the object than the illumination of the background, as described by Bolle et al. (1996).

### 2.2.4 Feature extraction

In computer vision, a feature, also referred to as a descriptor, refers to an abstraction of image information which represents individual measurable heuristic properties for recognising some phenomena present in that image or a region within that image (Gutierrez-Osuna and Hierlemann (2010), IBM (2012)). Features extracted from digital images can include information relating to the colour, texture and shape of an object of interest. These features may be summarised statistically, such as by a histogram or the statistical moments (mean, standard deviation, skewness, etc.) of a feature value measured for individual pixels over a region of interest. Such “region features” can be constructed at different levels of granularity, with the lowest possible granularity representing features of the whole image, then of an object within the image, then of some other region such as a pixel neighbourhood or a so-called superpixel. Usually a superpixel would be produced by oversegmenting an image and using each smaller segment as a region. The smallest region possible is generally an individual pixel.

Munkevik et al. (2007) developed a pixelwise classifier as part of a system to describe meals based on the layout of different ingredients, specifically meatballs, gravy, vegetables (potatoes, peas and carrots) and jam. The pixelwise classifier used features relating to the colour of each pixel, as well as statistics describing the pixels in a $9 \times 9$ square region surrounding each pixel of interest.

The beans classified by Kilic et al. (2007) give an example of a region representing a whole object. The features used were the first four statistical moments of the red, green and blue colour channels of all pixels comprising the bean, as well as the length and width of the bean shape when segmented from the rest of the image. This system was able to identify damaged beans with a 91% accuracy.

A similar set of features to those used in this thesis were used in Savakar and Anamiy (2009) to classify pictures of grains, fruits and flowers by their contents. In this instance, the colour features used were the mean, variance and range of the whole image for the raw red, green and blue colour channels, as well as the hue, saturation and intensity. In addition texture features were used, with a coocurrence matrix being produced for each of the red, green and blue colour channels. As the coocurrence matrices resemble grayscale images, they were treated similarly with statistics extracted from them including the mean, variance, range, energy, maximum probability, contrast, inverse difference moment and correlation. This resulted in a total of 18 colour features and 24 texture features. These features were used to train a neural network, which was then used to classify images of grains into ten categories, grapes into ten categories, mangoes into five categories and flowers into ten categories, with accuracies ranging from 84.0% to 94.1%.

Colour features are very common in computer vision systems since they are straightforward to obtain after selection of an appropriate colour space, as discussed in Section 2.2.2. In systems such as Guannan et al. (2009), which aimed to detect greening in potatoes the colour features alone can be sufficient to achieve the required level of system performance.

Texture features include simple features, such as the range and gradient of a pixel neighbourhood. These are both calculated from one colour channel at a time, with the range being the difference between the highest and lowest pixels in the pixel neighbourhood, while the gradient is typically calculated by convolving the image with a matrix in order to obtain a value indicating how quickly the pixel values are changing in a particular direction, such as left to right. Other more processor-intensive features can include features extracted from the image's frequency domain (Henrici, 1993), which is a means of representing a linear sequence according to how much of the sequence changes at what frequency. The same approach in two dimensions is used to analyse the textures of images, with the result that generally smooth images tend to give more "compact" results than generally rough images. Two examples are given in Figure 2.3 for comparison. Textons are statistical representations of pixel neighbourhoods which are typically sorted into clusters for classification purposes (Johnson and Shotton, 2010). Textons were removed from our early experiments because they took much longer to extract than the entire remaining feature set.

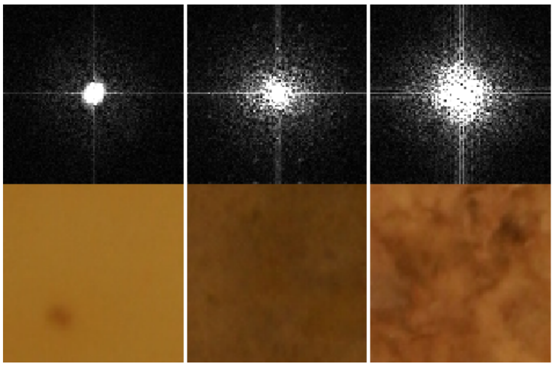


**Figure 2.3:** Frequency domain outputs, top, for three close-up images of potato skin, bottom. From left to right these represent an area of good skin, an area of silver scurf and an area of scab, respectively. The values further from the centre tend to be higher the rougher the respective input images. Features describing this might compare the mean values of different areas within the frequency domain output.

Shape features may also be used. The shape of an object might be analysed during the segmentation phase. Some of the features that might be used include the difference between the longest and shortest diameter, the length of the perimeter, the area and the roundness, which can be defined as the ratio of perimeter to area. For instance, a typical potato should be a mostly rounded shape with no sharp corners. A sharp corner might be detected by comparing the rate of change of the radius and would be likely to indicate a damaged potato.

Sometimes a problem may initially appear to be colour based but a better solution will be texture based, as in Quevedo et al. (2009). In this case, the best features for detecting enzymatic browning in pears proved to be fractal dimension features, describing the complexity of the L*colour channel in a specific area of the image. In other tasks there may already be well known features in use by humans for visual inspection. Mismi et al. (2007) developed a system to grade salmon fillets, a task that had been previously carried out by human graders. The humans had been trained using the colour chart in Figure 2.4, which also proved to be suitable for training a machine learning classifier. Regions of the salmon were compared to the chart in both the normalised RGB and CIELAB colour spaces, with the results corresponding to those of human graders 95% of the time.

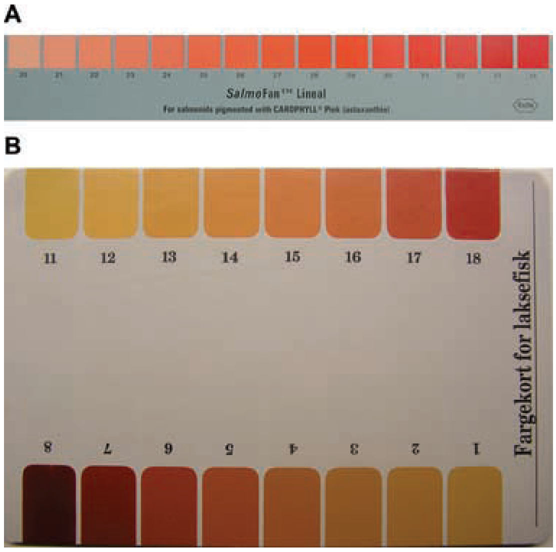


**Figure 2.4:** A guide for human salmon graders, used as training data by Mismi et al. (2007).

It is common to include both colour and texture features in the input to a pattern recognition system, as discussed in Munkevik et al. (2007). The VeggieVision system (Bolle et al., 1996) used HSI-colour and texture histograms to classify different types of fruit and vegetables, with application to a supermarket check-out for automatic produce recognition. Intending to supplement human classification rather than to replace it, the system provides a list of four likely candidates for the produce class, with the correct class appearing in the list 95% of the time. The operator can thus select a product with no more than one keypress 95% of the time, while using the normal slower process the remaining one time in twenty. The prototype system includes a camera below the glass on which the product sits. In order to separate items on the glass with items above the glass, the system first captures two images, one with no lighting and one lit from below. Pixels which were significantly brighter when lit from below are considered to be foreground pixels. These pixels are then segmented from the background pixels and used to produce colour histograms as well as two histograms providing texture indicators, one measured using two crossed bar masks and the other measuring the deviation of pixel intensities from the intensities of pixels in a surrounding region. When classifying all items on the shelf with ten test images of each product, the inclusion of the texture indicators increased the probability of the correct item being in the top four list from 93% to 96%. Adding size and shape features further increased this success rate but did so at the cost of slowing down the system below acceptable speeds for real-world application, a problem we also found with some more complex features.

### 2.2.5 Feature Selection

When classifying data with a high number of available features, it can become necessary to find ways to reduce the number of features which the classifier has to process. This is because every additional feature increases the complexity of the classification. Most methods of reducing the number of features either work by selecting a subset of features from a larger candidate feature set, while others such as principal component analysis (PCA) produce a smaller feature set composed of mathematical functions of the original features (Forsyth and Ponce, 2003).

PCA works by performing an orthogonal transformation on a data set with multiple possibly-correlated features, with the intention of producing a set of non-correlated features, known as principal components. The algorithm used presents the data such that the largest variance is along the first component, then the second largest is along the second component, and so on. For example, Figure 2.5 shows a distribution with two features, $x_1$ and $x_2$ best described by two principal components, $pc_1$ and $pc_2$.

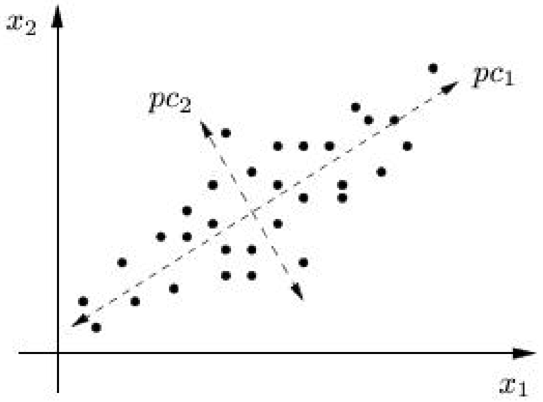


**Figure 2.5:** An example of principal component analysis, with two features represented by two principal components.

Most other feature selection algorithms compare individual features and either choose the next best for the classification problem at hand (forward selection) or remove the next worst (backward selection). For example, Sequential Forward Selection (SFS) starts with an empty set of features, then does a greedy search for the next best feature to add to the classifier, until either a preset number of features is reached or adding new features stops improving the classification performance.

Sequential Backward Selection (SBS), is essentially the reverse of SFS. SBS begins with a feature set consisting of every available feature, then does a greedy search for the next best feature to remove, until either the classification accuracy drops below a preset threshold or a target number of features is reached. Backward selection methods such as SBS are extremely computationally expensive since they use more features per iteration than forward selection.

Floating search (Pudil et al., 1994) is a method that can be applied to both SFS and SBS to allow them to take back earlier choices if these are later shown to be suboptimal. SFS begins with a random set of features, then performs a greedy search for the best feature to add and the best feature to remove. If the least significant feature is the one which was just added then it is kept in the selected feature set, otherwise the least significant feature is removed from the selected feature set and returned to the unselected candidate feature set. Sequential Backward Floating Selection employs the same technique in reverse, reselecting the best discarded feature if it is not the one most recently discarded. Pudil et al. compared the method to the older “Branch and

Bound" feature selection algorithm and found Sequential Floating Selection to perform similarly with less computational overhead.

Evolutionary algorithms (EA) can also be applied to feature selection (Duda et al., 2000). An EA uses mechanisms inspired by biological evolution such as reproduction, mutation, recombination and selection. The concept involves representing every possible classifier design as a binary number, with every single bit representing one aspect, such as whether or not a specific feature is selected for use. The algorithm begins with a selection of these binary numbers, known as "chromosomes", each of which is rated by a "fitness function" based on the classification accuracy. The fittest chromosomes are then "bred", which involves two random events. The first event, the "crossover", typically selects a random point in the chromosome to split. A new chromosome is then produced by joining together the left side of the first parent and the right side of the second. The second event, "mutation", involves swapping bits at random. Which can sometimes add beneficial modifications. The whole process is then repeated for other high ranking pairs of chromosomes in order to produce a new generation of chromosomes, which are again ranked by fitness and bred until the system stops producing improvements or a certain fitness level is met as a stopping criterion.

Sometimes a feature selection method is linked directly to the classifier used. For example AdaBoost (Freund and Schapire, 1999), a method for combining weak classifiers, can also be used to select features when trained using weak classifiers that each use a single feature. This technique is further described in the following Section 2.2.6.

AdaBoost was chosen for feature selection in this thesis because it offers a single efficient approach for both feature selection and classification. PCA was not used in this thesis, because we were primarily interested in selecting a small subset of useful features from a large set of candidate features of unknown importance, rather than finding a set of linearly uncorrelated variables that summarise all of the original feature values. The features selected for one variety, e.g. red potatoes, will usually be different from those selected for another variety, e.g. white potatoes. The sequential methods described above were not used due to their higher computational requirements. While evolutionary algorithms can be effective, they are prone to local optima, whereby a selection is returned that is only better than similar options, rather than an optimal solution.

### 2.2.6 Classification

In machine learning, the process of classification (Duda et al., 2000) refers to an algorithmic process to assign a data point to one of a set of classes, for instance "clean" vs. "dirty" for the area under a robotic vacuum cleaner, or "spam" vs. "non-spam" for an email filtering application. The data point, also termed an *instance* or an *example*, consists of a vector of features which together present all the information the classifier has about the example in question. In this thesis a data point represents a pixel in an image. Classification in machine learning is usually a supervised process as the classifier will first be trained on a selection of data points for which a human has already provided the correct results (or ground truth). The classifier can then deduce relationships between the provided results and the features of the data points, which it will then be able to make use of to provide a classification decision for a new data point.

This thesis investigates the use of machine learning classifiers for analysis of potato blemishes. Not all classifiers use machine learning, since a classifier to determine whether someone is tall enough to join a profession with a height requirement need only perform a simple comparison between their height and the requirement. The benefit of a machine learning approach is that it can potentially handle more complicated and less clearly defined problems.

In the following, we describe some of the most commonly used approaches to classification in the food engineering literature.

#### 2.2.6.1 Minimum distance classifier

A minimum distance classifier, as described in Gonzalez et al. (2004), represents each class of training examples by a mean vector. When classifying a new data point, the class of that data point will be set to the class of the closest mean vector. The distance may be calculated using a variety of metrics. Euclidean distance represents a generalisation of Pythagoras' theorem and is the most common metric used, alternatively Mahalanobis distance, which applies weighting on different dimensions according to the covariance of the sample group. Other metrics include Manhattan distance, which simulates travelling in each dimension sequentially, and Chebyshev distance, which is like Manhattan distance but also measures orthogonal distances as equal to the distance moved in one dimension, meaning that the distance between two vectors is the greatest of their differences along any coordinate dimension. Weightings may also be applied to the training examples in order to reduce the impact of outliers (data points which appear to deviate markedly from other members of the sample in which they occur) on the mean vector for each class. A minimum distance classifier works best for classes which form clusters and can benefit from a good choice of features.

#### 2.2.6.2 *k*-Nearest neighbour

The *k*-nearest neighbour (*k*NN) classifier is a lazy learning algorithm, meaning that most of the computation takes place when classifying data rather than when training. When classifying, a datapoint is classified according to the closest stored training examples. As with minimum distance classifiers, the distance to nearby training examples can be Euclidean or Mahalanobis, however no clustering is involved. Rather than finding the nearest cluster center, the classifier finds the closest $k$ training examples and then chooses the class of the majority. $k$ can be any number, usually odd in the case of binary classification.

The only preprocessing required for a *k*NN classifier is normalisation of individual features, usually by applying a linear rescaling to set their mean value to zero and their standard deviation to 1. This is achieved by taking the mean $\bar{x}$ and variance $\sigma$ of the training set $i$ and then subtracting the mean from each example and dividing the result by the variance as $\tilde{x}_i^n = \frac{x_i^n - \bar{x}_i}{\sigma_i}$. Without this rescaling, some features may become more significant purely because of the units used. Additional preprocessing methods exist to speed up the search for nearby neighbours, for example, by subdividing into regions such that the classifier can be sure that near neighbours must be in regions adjacent to the region in which the data point falls, by using kd-trees or by accepting some approximation in the results.

Due to the need to store all of the training data for each class, *k*NN has a high memory requirement.

#### 2.2.6.3 Bayes optimal classifier

The Bayes classifier for a $0-1$ loss function (Gonzalez et al., 2004) has decision functions of the form $d_j(x) = p(x|w_j)P(w_j)j = 1, 2, ..., W$ where $p(x|w_j)$ is the probability density function (PDF) of the pattern vectors of class $w_j$ and $P(w_j)$ is the probability (a scalar) that class $w_j$ occurs. Given an unknown pattern vector, the process is to compute a total of $W$ decision functions and assign the pattern to the class whose decision function yields the largest numerical value. Ties are resolved arbitrarily. When the probability density functions are, or are assumed to be, Gaussian distributed, this $n$-dimensional PDF can be presented as $p(x|w_j) = \frac{1}{(2\pi)^{n/2}|C_j|^{1/2^e}} - \frac{1}{2}[(x-m_j)^T C_j^{-t}(x-m_j)]$ where $C_j$ and $m_j$ are the covariance matrix and mean vector of the pattern population of class $w_j$ and $|C_j|$ is the determinant of $C_j$.

Because the logarithm is a monotonically increasing function, choosing the largest $d_j(x)$ to classify patterns is equivalent to choosing the largest $\ln[d_j(x)]$, so the decision function can be rewritten in the form $d_j(x) = \ln[p(x|w_j)P(w_j)] = \ln p(x|w_j) + \ln P(w_j)$ where the logarithm is guaranteed to be real because $p(x|w_j)$ and $P(w_j)$ are non-negative.

#### 2.2.6.4 Neural networks

An artificial neural network (ANN), also referred to as a neural network (NN), is a mathematical model or computational model built from connected nodes, each comprising a very simple processing unit, that work in parallel. The design is inspired by the structure and/or functional aspects of biological neural networks such as the brain. These nodes, or artificial neurons, are employed to process information using a so-called connectionist approach to computation. A neural network obtains knowledge stores in the weighting of the connections between the nodes during the training process. They have been shown to be able to model complex relationships between inputs and outputs and to solve arbitrarily complex pattern recognition problems.

Neural networks come in a variety of forms and configurations, but the fundamental building block is the neuron, which represents many features of biological neurons. In Figure 2.6, the configuration of neurons produces the AND operator, since if either of the two input neurons are firing, their weights will only provide an input of 1, which is below the output neuron's threshold of 1.5. For the output of this binary neural network to be positive, both input neurons should be firing, giving an input to the output neuron of 2, sufficient to trigger it. If the threshold were reduced this could produce an OR gate instead, since just one input would be sufficient to produce a positive output. Thus the main features defining the operation of a neural network are the weights, which are all set to 1 in this example, the thresholds of neurons and their connectivity. Figure 2.7

shows a far more complex multi-layer neural network, with four inputs to four input-layer neurons. Only one of these input neurons directly affects an output neuron with the others instead affecting mid-level “hidden” neurons, as well as one case where an input neuron affects another input neuron. This model also necessitates the use of the well-known back-propagation algorithm (Bishop, 1995) for training, where the output neuron $U_3$ affects the hidden neuron $U_6$ as well as connectivity between neurons on the same level and in the case of neuron $U_9$, the output it gave the last time it was calculated will affect its own output the next time it is calculated.

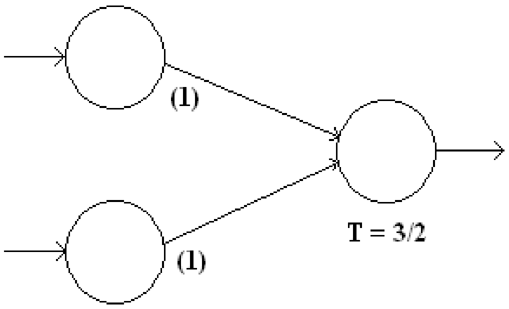


**Figure 2.6:** A simple neural network producing an AND gate.

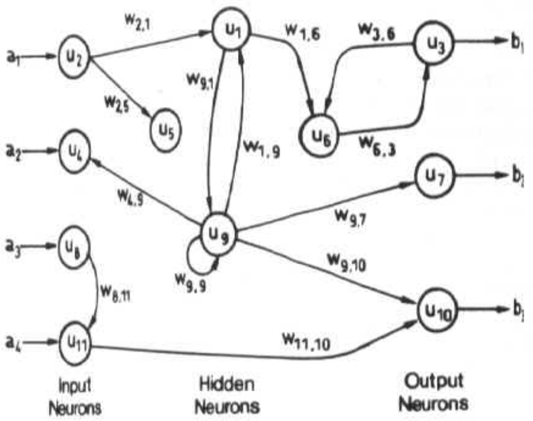


**Figure 2.7:** An example of a complex neural network, from Stergiou and Siganos (2012).

#### 2.2.6.5 Support vector machines

A support vector machine (SVM) (Duda et al., 2000) is a supervised learning classifier that constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, typically formed from the features provided and from "support vectors", which are combinations of two or more features. This use of these support vectors addresses classification problems whereby different classes may not be linearly separable in the finite dimensions provided as input data. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, in order to find a space which can make separation of the classes much easier.

#### 2.2.6.6 AdaBoost

The AdaBoost algorithm (Freund and Schapire, 1999) is a boosted classifier. Boosting is a method of combining weak classifiers, with a certainty greater than 50%, to produce a single strong classifier with high classification accuracy. Given sufficient iterations and sufficient training data, AdaBoost can always produce a strong classifier which perfectly classifies the training data (Russell and Norvig, 2003).

AdaBoost and its variants take a group of weak classifiers, sort them by usefulness and combine the best weak classifiers with specific weights to produce a single strong classifier with a higher success rate than the available weak classifiers. Any classifiers can be combined using this method. In this thesis, the weak classifier used was a decision stump, which is a single level decision tree. Each decision stump provides a single threshold classifier using a single feature. Using decision stumps like this, the system will iterate through all available features, then investigate all the different ways the training data can be subdivided using a single threshold on that feature. Since a threshold can be any value, the candidate values are usually chosen to lie halfway between consecutive values of the feature, so if a feature is represented by an 8-bit integer (0-255), there will be no more than 255 possible thresholds. The system will then select the feature and threshold which best classify the training data. Once a new classifier has been chosen it will then weight all training examples such that misclassified examples are given more importance than correctly classified examples and select another weak classifier until a stopping condition has been met, such as a specific number of weak classifiers having been chosen. The re-weighting of training data provides a practical alternative to using an entirely new training dataset chosen to include more of the misclassified data.

AdaBoost was chosen for classification in this thesis because it offers a single efficient approach for both feature selection and classification. KNN classifiers were not used due to the high computational requirements, requiring the storage of and comparison with a large number of individual data points. Bayes and Minimum Distance classifiers both make strong assumptions about the distribution of the data, so were not used in this thesis. Other classifiers, such as the Neural Networks and Support Vector Machines described above, could have been applied for classification in this thesis after first selecting the relevant features using AdaBoost. However such an approach would increase the complexity of the overall solution by using two algorithms instead of one. Additionally, AdaBoost offers the ability for the user to see and understand which features are important for solving a particular classification problem, for example, by listing the top $n$ features selected. See, for example, Table 5.2 giving the top 10 features selected for detection of blemish vs. non-blemish in white potatoes.

## 2.3 Example food vision systems

This section summarises some of the major related food vision systems. Many of the described systems are used not only for quality control, but also to monitor the lifecycle of produce, such as the stages of a product being cooked, ripening on a tree, or going bad. In the case of products cooking or ripening, the system could then be used to choose the correct time to use the product because it is properly ripened or cooked. Likewise, a system monitoring the process of going bad could make an optimal choice of when to use a product, such as a potato, before it is no longer fit for use or saleable.

### 2.3.1 Systems for fresh produce

Hasankhani and Navid (2012) presented a per-potato means of classifying potatoes into three grades by size and by colour. For purposes of locating the potato in an image and of identifying the potato size, the image was first reduced to a single grayscale intensity channel to speed up processing. Noise was removed using a Gaussian filter before the boundary of the potato was identified using a Sobel filter. From this information it was possible to accurately measure the area of a potato, which was one factor in choosing a grade for the tuber. Colour based classification was then carried out after manually selecting features by brute force. Each feature set was compared based on the number of pixels it identified as faults compared to the number of pixels an expert had marked as faulty in the same potato. The best correlation involved using a logarithmic transform in the HSV colour space. Finally the potatoes were graded into four classes according to blemish, then regraded according to the qualitative grade and the size. The tests were carried out using one set of 110 potatoes.

Wang et al. (2011) demonstrated the ability to accurately estimate the size of a potato based on a single shape feature. First, the centroid was located followed by the principal axis, which represents the longest straight line through the potato that passes through the centroid. This took 1.38 seconds to calculate in MATLAB, compared to 15.72 seconds to locate the more commonly used Minimum Enclosing Rectangle, found by rotating the potato until the principal axis is horizontal and then fitting a bounding box to the shape. The Centroidal Principal Axis was shown to have a stronger correlation with the major and minor axes than the Minimum Enclosing Rectangle, with a significantly smaller calculation time. Correlation coefficients were given as 0.9656 and 0.9166 for major and minor axis using the CPA, compared to 0.9226 and 0.8807 for the same features using the MER approach.

A support vector machine was used by Dacal-Nieto et al. (2011) to demonstrate a method of using hyperspectral imaging to locate common scab in potatoes. Using a sample set of 234 potatoes, they first produce hypercubes, which are images with 256 colour channels, for each potato. Trial and error was then used to locate the channel best suited to segmenting the potato from the background, which was shown to be the channel representing light at a wavelength of 980nm. After segmenting the potato from the background, each pixel of the potato was then treated as a data point with 256 features consisting of the 256 colour channels. Human markup was used to divide the data points into scab and non-scab sets, then SVM and Random Forest classifiers were trained on both sets, with the SVM providing the highest accuracy at 97.1%

Hyperspectral imaging is still a slow technology compared to traditional 3 channel imaging, but Ariana and Lu (2010) demonstrated a technique using a hyperspectral image as a first step for detecting defects in cucumbers and pickles. Using a similar hypercube to Dacal-Nieto et al. (2011), they used branch and bound feature selection to select features from only four wavebands. By limiting the number of wavebands, it was practical to use real-time imaging with a CCD camera fitted with custom filters while still benefitting from training on the slower system. The resulting image was classified using K-nearest neighbour and with discriminant analysis, producing accuracies of up to 94.7% for cucumbers and 82.9% for pickles.

Samanta et al. (2012) graded potatoes according to scab coverage using a method involving histogram analysis. $k$-means clustering was used to segment the image for background removal, using the Hue component of the HSI colour space. A histogram was then produced of the identified potato pixels using a greyscale version of the original image. This greyscale image used a grey component of $Grey = R \times 0.299 + G \times 0.587 + B \times 0.114$. The potato was then graded according to scab coverage by comparing this histogram to histograms from a ground truth set of potatoes which had been sorted by grade. Accuracy from different test sets ranged from 87.6% to 97.5% with a mean of 93.4%.

Pedreschi et al. (2004) classified potato chips by frying temperature and blanching. Six classes existed in the sample set, blanched and unblanched with a frying temperature of 120 degrees, 150 degrees and 180 degrees, respectively. The system used a mixture of 36 geometric features, including the height, width and roundness, and the ratio of both axes of the fitted ellipse as well as 368 each of red, green, blue and intensity features and 3 L*a*b* features per sample. Overall the system achieved 90% accuracy in assigning the samples' blanching/temperature combination using a simple classifier and selecting the best five features with sequential forward selection. Distinguishing between blanched and unblanched samples at the same temperature produced a 100% accuracy with only the best one or two features and a very simplistic classifier, while using samples blanched at different temperatures and only identifying whether a sample had been blanched, among all temperatures gave a 93% accuracy. The higher accuracy when training and testing using samples blanched at the same temperature shows the benefit of being able to re-train a classifier for a new set of conditions, which is the approach adopted in this thesis.

As described in Section 2.2.1, the use of custom imaging equipment can simplify the detection of blemishes in potatoes. Muir et al. (1999) recorded the intensity of light at 73 different wavelengths in the range of 400-2000 nanometers to demonstrate the different reflective properties of specific diseases at each wavelength. A spectrophotometer covering the frequency range was used to analyse the reflectiveness of several hundred potatoes over different classes of defects as well as unblemished skin. Measurements showed clearly that specific defects reflected differently at specific wavelengths, an example being that blight and healthy potato mostly reflected the same amount of light, with the largest difference occurring at a wavelength slightly under 1200nm. Likewise, although bacterial soft rots and common scab tended to reflect differently, there were frequencies in the 800-1100nm range where they are less distinct. The difference between sensor readings for the different classes are shown in Figure 2.8. Spectrophotometers are slow, but the results attained using the spectrophotometer could then be used to set up a multi-spectral camera, which would capture 16 different channels, chosen based on the results of the spectrophotometer, at a much faster rate. This work was commissioned in part by R.J. Herbert Engineering Ltd. for use in their Upgrader product line.

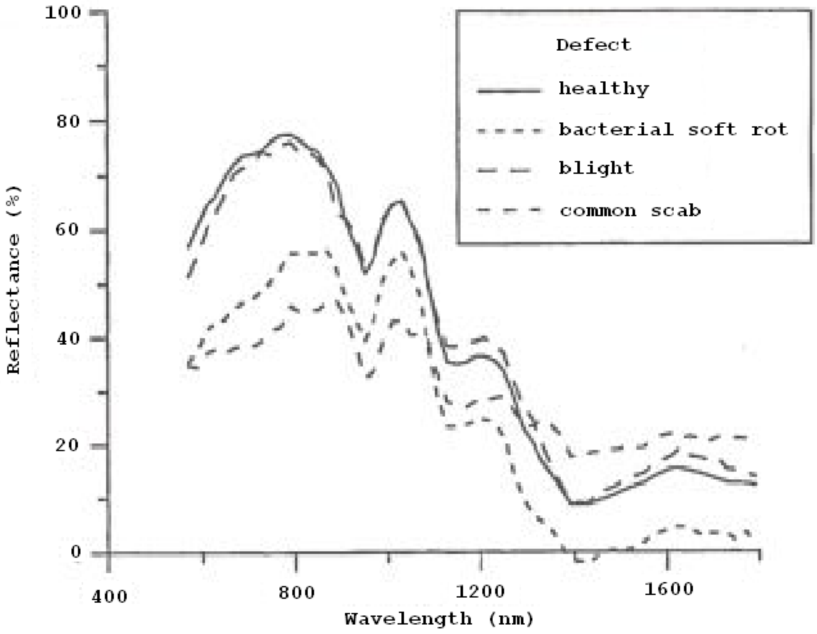


**Figure 2.8:** Output from multispectral scans of different blemishes and unblemished potato skin described in Muir et al. (1999).

Tao et al. (1995) described a method for identifying greened potatoes as well as yellow and green apples using discriminant analysis on histograms produced from each of the hue, saturation and intensity channels in the HSI colour space. The system was trained using 40 good and 40 bad examples of potatoes or apples as appropriate. The best potato classifier correctly classified 95% of good potatoes as good and 87% of greened potatoes as greened. The apple classifier was tested on a smaller sample set and successfully classified 100% of both good and bad apples.

Guannan et al. (2009) detected misshapen potatoes by comparing the local rate of change of the radius of a potato. In addition they detected sprouting using a thresholded value for the difference $G_D$ between the green colour channel $G$ and the average of the three colour channels. If $G_D$ is higher than the threshold $T_D$ then the pixel is deemed to be sprouting. The value of $T_D$ was calculated for a potato variety by comparing the values of $G_D$ for each pixel across a group of good potatoes and placing $T_D$ at a value higher than $G_D$ reaches in these good examples. The system achieved a success rate of 88.0%. We also investigated the use of features that summarise the image content across either the whole potato or a region within the potato. For example, we used the mean of the red colour channel values in a particular region (see Section 4.1.2 for further details).

Rios-Cabrera et al. (2008) presented a method for classifying potatoes by shape, identifying scabs and cracks by shape and detecting greening by colour. The system used a neural network to identify shapes in an image based on the Boundary Object Function, a sequence of distances between the centroid of a region and points on the perimeter. Similarly to Guannan et al. (2009), this allowed the system to identify misshapen potatoes. Once the potato shape had been classified, the same segmentation process used previously to separate the potato from the image background was then run with a higher threshold to separate rough areas of potato skin from the rest of the potato. These areas were classified into scab and crack using the same features used to classify the potato shape, then greening was detected using the hue, since greening is shown to have a hue within a range of 90 to 135. The system achieved a 93.8% success rate for grading potatoes according to the amounts of these three blemishes and tuber shape.

Zhou et al. (1998) developed a similar system, using HSV colour channels to detect green defects (greening and sprouting) in potatoes by counting the number of pixels with hue values in a specific range, chosen based on hand-selected training examples, and classifying those with a high number of pixels in that range as greened. They also analysed the shape of the potato by comparison to an ellipse template and in size and weight by measuring the minor axis and area respectively. This gave an overall success rate of 86.5% with a false positive rate of 57.1%. For the shape classification alone, the results were 90% success, with 17% false positives. This thesis also presents an approach based on counting pixels in Chapter 6.

Mendoza and Aguilera (2004) used nine simple features to identify the ripening stages of bananas. The first three features were the mean of the L*a*b* colour channels across the entire banana, then the next two relate to a pixel-wise detection of brown patches on the banana, followed by the homogeneity, contrast, correlation and entropy of the banana's texture. The background was removed using thresholding and the features above were extracted from the banana. Features were selected using Sequential Forward Selection resulting in a success rate of 98%.

As an alternative to inspecting a potato using multiple images, Noordam et al. (2000) used mirrors to inspect potatoes from all angles as they passed a line-scan camera. To achieve a high speed, the system used 11 dedicated Digital Signal Processor (DSP) boards. The system used segmentation and Neural Networks, perfectly classifying potatoes according to the presence or absence of rhizoctonia. The system also detected growth cracks with 78-87% accuracy and correctly labelled scab blemishes with 70% accuracy.

### 2.3.2 Systems for dairy products

Computer vision has been used in a number of cheese production methods. Wang and Sun (2002) demonstrated the use of an image analysis approach to measure the way that cheese melts. Melting cheese may spread or shrink, and measuring the surface area after melting can prove non-trivial by other methods, while using a machine vision approach this is simply a matter of counting the cheese pixels. The study was undertaken to compare the effects of melting cheese over set time periods and at specific temperatures. Cheeses were stored at $5°C$ for 24 hours before being sliced into 3mm thick samples. The slices were cooked for up to 20 minutes on a pizza tray and removed from the oven periodically to capture images. The images were captured in a dark room with illumination from two fluorescent lights. Both cheeses were shown to melt the most at $130°C$. At $70°C$, cheddar cheese melted by 20% but Mozzarella melted less than 3% and even shrank back down to its original size by the end of the experiment. These results can be used to inform choices for production of foodstuffs involving melted cheese, such as the amount and type of cheese as well as the cooking temperature. Similarly, it could be of interest for this research to consider monitoring the change in area of storage blemishes, such as silver scurf, which spread across potatoes over time in cold stores. By monitoring this change in area, the producer could be warned before a batch of potatoes developed too much coverage of these blemishes to be, allowing those potatoes to be prioritised for sale before the blemish spreads enough to reduce the sale price.

Everard et al. (2007) used colour analysis to monitor the syneresis, or separation, of cheese curd during the cheese-making process, a stage which is usually accompanied by expert monitoring to ensure the quality and yield of the resulting cheese. By observation of the process, it was observed that the milk becomes increasingly yellow throughout the process, so a system was used to monitor colour changes every minute over the 85 minute duration of syneresis. The results indicated a potential for this system to be used for monitoring syneresis, however it was noted that if the milk was stirred too slowly then there was a tendency for the curds to sink. It was suggested that this problem might be resolved with submerged visual probes. If our research were to be adapted to monitor the development of blemishes in storage then visibility might also be a problem, depending on whether the potatoes at the top of a crate are likely to be representative of the whole crate.

### 2.3.3 Systems for meat products

Zheng et al. (2006) produced a system to estimate the level of shrinkage in cooked beef joints during the cooling process. The products were photographed and an ellipse fitted to the shape of the beef joint. After deriving the major and minor axes from this ellipse, the original 2-D joint shape was then divided into numerous parallel cross-sections which were assumed to be cylindrical discs and used to calculate the volume and surface area of the joint. The resulting measurements showed that the volume and surface area reduction due to water loss during cooling could be reliably predicted based on the original volume and surface area of the joint. This method of 3d reconstruction might be of use for estimating blemish coverage in a potato from a flat image.

Jackman et al. (2010) used statistical moments of the red, green and blue raw colour channels, of the whole image, for the purposes of analysing the likeability of meat. Rib-eye steak samples were kept under the same conditions and allowed to “bloom” for an hour to ensure consistent colour development. Five high magnification photographs of different areas of the steaks were then taken, each pixel representing $0.08mm^2$. Five statistical features (mean, variance, skew, kurtosis and interquartile range) were extracted from these images along with features provided by a a form of 2-dimensional wavelet decomposition, chosen by prior experimentation. A classifier was then produced using an evolutionary algorithm, whereby a solution is chosen by testing multiple random options and combining features from the most successful and producing a new random set which include these features, in a method inspired by natural genetics. This classifier described the likeability, tenderness, juiciness and flavour of steaks, from images, with accuracies of 86%, 76%, 69% and 78%, respectively.

### 2.3.4 Other systems

Many computer vision classification systems are developed for experimental purposes. Unay and Gosselin (2006) developed a system to distinguish between blemishes in apples and visible apple stem or calyx features with the intention of removing the stem and calyx data from an apple image that would go on to another system. Images were recorded using custom spectral frequencies, then various features including statistical moments of the light intensities at specific frequencies as well as shape features were used for pattern recognition. These features were then passed to AdaBoost to produce a classifier from. The final AdaBoost classifier provided a true positive rate of 98% for the whole data set with a false positive rate of 9%.

Hepworth et al. (2004) presented a system designed to help experiments aimed at producing a perfect head on a pint of beer. The system worked by measuring the bubble size distribution of what is known as “bubble haze” while the head is still settling. The system was tested on a “synthetic” beer produced for the purpose, with the tests carried out in a tank when it was established that the curvature of a beer glass complicated the process due to image distortion.

## 2.4 Conclusion

Machine vision has much to offer the food industry, where much of the work currently undertaken by humans could be handled more consistently and reliably by automated systems. Computer vision solutions already exist that could be adapted to the task of detecting and identifying blemish diseases in potato images.

Given the cost to industry of potato mis-classification, at a time of rising food costs and with the appearance being the most important factor for consumers in making decisions about the purchase of potatoes, there are clear benefits to a machine learning based potato QC tool.

Having reviewed the literature on machine vision in the food industry, we concluded that there is an opportunity to produce a user-friendly trainable system for blemish detection and identification in potatoes, which can be robust to changes in lighting and in variations between potato varieties and also between different harvests.

The following chapters now describe our approach to the task.

# 3

# Experimental setup

This chapter describes the overall experimental framework common to all of the experiments within this thesis. The details covered include the camera setup and image acquisition, a summary of the data sets and the means by which a set of "ground-truth" accepted correct answers is produced for comparison with the classifier output. Such ground-truth data can be used to evaluate the classifier using performance metrics described below. The code for the system software was implemented in MATLAB (The Mathworks Inc., 2009b), making extensive use of the Image Processing Toolbox (The Mathworks Inc., 2009a).

## 3.1 Image Acquisition

The experimental data in this thesis, consisting of images of potatoes, were acquired using a colour camera (Sony DSLR-A350K) fixed above the tubers, which in turn were placed on a white board. The camera was set to auto-focus at a distance of 60cm from the camera objective to the base on which the subjects were placed, with a focal length of 70mm and an aperture setting of F22. A pixel in such an image covers an area of around $0.02mm^2$ To reduce the effects of shadows and changing light conditions the potatoes were placed inside a white cylinder with daylight bulbs placed around the top. The equipment used to capture these images is shown in Figure 3.1. These factors may change in an industrial application, but given the machine learning approach the only major requirement ought to be that the system uses the same setup for image acquisition, including lighting, when being trained as when in operation.

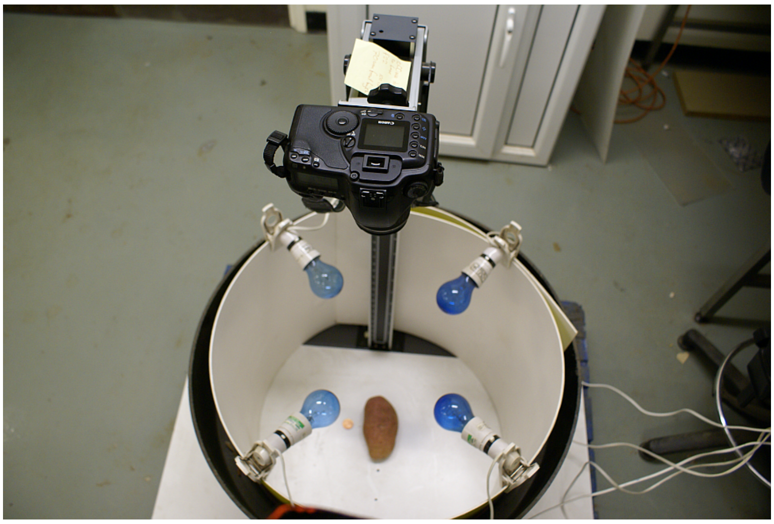


**Figure 3.1:** The camera setup for photographing the potatoes at a constant distance with all-around lighting.

## 3.2 Data Sets

Two sets of data were collected, from potatoes provided by Sutton Bridge Crop Storage Research, part of the Agricultural and Horticultural Development Board. The datasets were for white and red potatoes, respectively, including potatoes affected by different blemishes.

The white potato data set, collected from the harvest of 2007, consisted of 102 images including 19 images containing a single blemish type, 39 with two distinct blemish types, 38 with three and 6 containing more than three blemish types. The most frequently occurring blemishes were black dot and silver scurf, appearing in 69 and 53 images, respectively, while the least frequent were elephant hide and growth cracks, with no more than 3 images of each. Powdery or common scab occurred in 45 images and green blemishes, either greening or sprouting, occurred in 20 images.

The red potato data set, from the harvest of 2008, consisted of 48 images of potatoes specifically selected to represent between 10 and 21 examples each of five blemish types. These included 11 potatoes with black dot, 18 with silver scurf, 21 with scabs and 13 with green blemishes, including both sprouting and greening. Note that greening of red potatoes results in a blackening of the skin rather than the more obvious green colour in white potatoes. Both datasets contain potatoes of multiple varieties.

## 3.3 Ground truth data

Ground truth data are needed in experimental systems both to provide training data and to verify that experimental results correlate with what should be expected, i.e. to provide the machine learning classifier with examples of the desired input-output mapping. A machine learning classifier is said to *generalize* well when the input-output mapping computed by the trained classifier is correct for test data not used in creating or training the classifier; the term "generalization" is borrowed from psychology (Haykin, 1998). Thus it is common practice to divide the experimental data into disjoint subsets, for the purposes of training and testing the classifier, respectively, with a human markup providing the baseline correct classes for both training and testing datasets.

Human-provided ground-truth data are prone to errors arising from the same human inaccuracies that a machine learning classifier might be intended to resolve. Studies have shown discrepancies from 5% (Clark et al., 1998) to 25% (Fletcher-Heath et al., 2001) between human markup, whether from two different humans marking the same image or by the same human using a different interface to mark the same image. In a blemish detection system, areas marked as blemish or non-blemish may be slightly larger or smaller than they should be (Clark et al., 1998), or multiple clustered blemish areas can be accidentally marked as one solid blemish area. Approaches to resolve this issue can include simply detecting the center of an area (Hafiane et al., 2008), using multiple ground truthers and providing automatic feedback on areas which a computer system suspects may be incorrectly marked (Kasturi et al., 2009).

Examples of potatoes with the corresponding ground truth image can be seen in Figure 3.2. The images are colour coded so that the pixel colour, if provided, indicates the correct class in the ground truth data. Black pixels represent background pixels, while pixels of other colours are treated as unmarked. Any unmarked pixels which were found to be of colours used by the ground truth markup were replaced with nearby colour values.

### 3.3.1 Background removal

To train the classifiers and test their performance, the images need to be marked up by hand to provide the ground truth information indicating the correct class of each pixel. The mark up process begins with a semi-automatic method for background removal, using the Magic Wand tool in Photoshop (Adobe, 2010) to label the image region surrounding the potato. This is achieved by manually clicking on the background in Photoshop. The flood fill tool usually succeeds in selecting the whole background from one click due to the fairly uniform colour which was deliberately chosen. In a commercial product, the Herbert Upgrader (R.J. Herbert Engineering Ltd, 2008) also relies on background characteristics to separate potatoes from the bright blue conveyor belt on which they are carried. The use of a blue background would limit the potato varieties that could be handled by this project, especially blue varieties such as Congo or Blue Tomcat, so this approach was not used in this thesis.

Since the flood fill algorithm is prone to missing small areas of background, the system was also made to detect all areas of non-background pixels, setting all but the largest area to background. This approach works so long as the human supervision is detailed enough to ensure that the potato is the largest non-background area in the image, which is a simple requirement. Our image sets included a coin in each image to verify camera settings, but this does not need to be included. This can pose challenges for very small potatoes, or of a commercial environment where a line might include other sizable debris. There is existing work, e.g. Al-Mallahi et al. (2008), on distinguishing between potatoes and the most likely debris to need filtering out of the image.

### 3.3.2 Manual ground truth

In this thesis, the potato area was then hand labelled by an industry expert into regions corresponding to different types of blemishes as well as non-blemish. It is not necessary to label all pixels in an image: some areas of high uncertainty or ambiguity were left unmarked, which is inevitable since even under a microscope it can be impossible to tell that, for example, a spot on a potato has swollen due to an insect digging into the soil and losing a leg beside it. These unmarked pixels are ignored during training of the classifier. Background pixels are also omitted from the subsequent calculations.

The process of manual markup consists of taking the digital image after the background has been removed and marking it using appropriate image editing software (in this case Microsoft Paint). The human markers were asked to outline areas of blemish which were then flood-filled automatically with a solid colour corresponding to a specific blemish type. For non-blemished potato, an area was flood-filled with white pixels after all blemishes had been marked. With practice, using a suitable input device (i.e. a graphics tablet) a marker could typically complete a single potato in a minute.

In addition, to assess the impact of natural variation between human experts involved in providing the ground truth data, we used a combination of markup data obtained from three different humans for a selected subset of the original data for white potatoes. First, 20 different images were selected from the original set. For testing intra-marker consistency, two additional images were also added by taking one of these 20 images and then taking its reflection in $x$ and its tranpose. This resulted in an image datset of 22 images, including 3 images with the same information content but in different guises to allow assessment of the intra-markup consistency for each expert in turn.

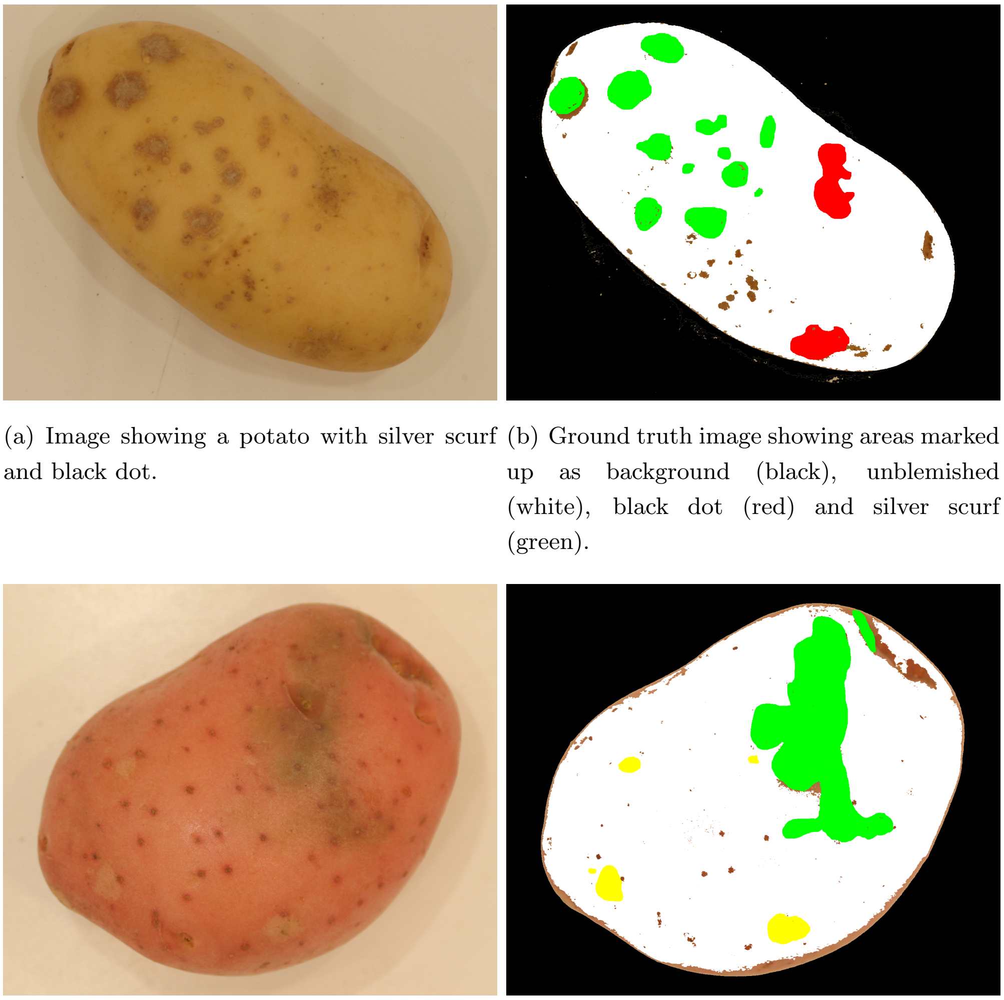
The first marker was the original expert (a potato pathologist at Sutton Bridge Crop Storage Research) who also had access to the original potato samples when marking up the data. The second marker was another potato expert (another employee at Sutton Bridge Crop Storage) and the third marker was a computer vision researcher (from the University of Lincoln). Neither the second or third marker had access to the original potato samples, as these were no longer available due to natural decay – instead, these markers only had access to the image data sets and a list of the blemish categories present in each image.

This panel-based approach can be used to reduce noise by only accepting examples (pixels in this thesis) which are marked as the same class by a majority of markers (i.e. 2 out of 3 markers in our case). This so-called “gold standard” markup was used for assessing inter-marker variation in both the blemish detection and blemish identification experiments. It was found that the marked-up images had up to 6% disagreement between the same marker and up to 12% between different markers. When combined, the voted-upon “gold standard” data set differed from the default, expert-marked data set by 9% among marked pixels.

### 3.3.3 Post-markup resizing

The resolution of the images taken was 3072×2048 pixels. After the background subtraction and manual ground truthing. these images were then scaled down by a factor of

2, to 1536×1024 pixels, in order to reduce the size of the classification task. Initial tests showed that this down-scaling operation did not reduce the accuracy of classification, with accuracy varying by no more than 0.1%.



(a) Image showing a potato with silver scurf and black dot.

(b) Ground truth image showing areas marked up as background (black), unblemished (white), black dot (red) and silver scurf (green).

(c) Image showing a potato with silver scurf and greening.

(d) Ground truth image showing areas marked up as background (black), unblemished (white), siver scurf (yellow) and greening (green).

**Figure 3.2:** Ground truth images of potatoes, with the originals for comparison. More examples can be found in Appendix B.

## 3.4 Training and testing data

Expert marked training data are needed to provide a classifier with examples which are known to be correct, so the classifier can deduce features which relate to the correct classes by example. Likewise, expert marked testing data are required to adequately assess how accurately the system classifies new examples.

When dividing the data set into training and testing data sets, it is necessary to ensure that the human-marked data from any one potato image is only present in either the training set or the testing set, never in both. Once the decision has been made of which images to take training and testing data from, the examples in the training images are then reduced to an equal number $N$ of examples from each class, since an imbalanced number of examples per class can result in the classifier becoming biased in favour of one class over another. For instance if there were 100,000 examples, 90,000 belonging to class A and 10,000 belonging to class B, the classifier could achieve 90% accuracy by classifying everything as class A. AdaBoost compensates for this by adding weight to examples which are misclassified, but a particularly large imbalance can still take several iterations to correct, while pre-selecting the training data can prevent such corrections from being necessary. In this thesis, $N$ is 30,000 per class for blemish detection experiments or 10,000 per class for blemish identification experiments, using a total training set of 60,000 or 50,000 examples per training data set. The number of potato images involved compares reasonably to other studies such as Mendoza and Aguilera (2004), who used data sets of 49 bananas and 6 bunches of up to 13 bananas, respectively, and to Pedreschi et al. (2004) who tested on sets of between 20 and 60 potato chips.

Because pixels near to the boundary of the potato were generally out of focus, the decision was also taken not to process pixels too close to the edge of the potato, defined as having background pixels in the $65 \times 65$ region centered on the pixel. This was done also to reduce the number of features that might be affected by the edge of the potato, which tended to have a large impact on the gradient and edge length features. This approach could be compensated for by using a technique such as that used by the Herbert Upgrader (R.J. Herbert Engineering Ltd, 2008) to summarise the blemish coverage of an entire potato from multiple images taken in different orientations. These omitted pixels represented under 20% of the data set. As in Section 3.3.1, this puts a minimum size on a potato that can be classified, but this could be compensated for by changing camera settings if the system is expecting a batch of small potatoes.

Once the training images have been selected, $N$ pixels for each class are then selected. These pixels are evenly distributed between all examples of that class from the training images by taking every $n$th example of the class, where $n$ equals $N$ divided by the number of examples of that class $X_i$. To allow the experiments to be repeated with different training and testing data sets, the first pixel can be offset by up to $n$ and still produce a similarly representative training or testing set. For example, if a class has 100,000 examples in the training set from which the system requires 10,000 examples, those 10,000 could be taken as every tenth example starting with the first example in each file, or the set might be every tenth example starting from the third example, or from anything up to the tenth example.

Testing data are produced in the same manner when testing performance on a pixel-wise level. For those experiments which test performance on a whole potato level, by detecting which blemishes cover more than a certain percentage of the visible potato, every pixel is classified for each test image.

Using the training and testing data with ground-truth labelling, the output of the classifier can be compared to the expected output from ground truth in order to produce various metrics to evaluate the classifier's performance, as discussed next.

## 3.5 Evaluation metrics

Once the classifier has been trained successfully and then tested on a given data set, the actual outputs of the classifier can be compared to the ground truth data to obtain a measure of how well the classifier performs. For the blemish detector, the output of the classifier is a binary image with pixels indicating good potato or blemish. The performance of the system can be measured by comparing the output image to the ground truth information. The following statistics were calculated for each output image:

- TP - true positive, number of pixels that were classified as blemish and matched ground truth;
- FP - false positive, number of pixels that were classified as blemish but did not match ground truth;

- TN - true negative, number of pixels that were classified as good potato and matched ground truth;
- FN - false negative, number of pixels that were classified as good potato but did not match ground truth.

From these statistics we could calculate the following metrics:

- sensitivity = $\frac{TP}{TP+FN}$;
- specificity = $\frac{TN}{TN+FP}$;
- accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$.

We represent the performance of our blemish detection classifier using the most common method based on Receiver Operating Characteristic (ROC) curves, which provide detailed information about the relationship between the sensitivity and specificity with respect to different parameter settings of the system (Fawcett, 2004). Figure 3.3 shows ROC curves produced by varying the value of the parameter $b$ in the AdaBoost algorithm, which is the weighting factor that determines whether the classifier should err on the side of false positives or false negatives. This choice can be important, say in medical systems where it is preferable to err on the side of false positives, referring people for further tests who are actually healthy, than to produce a single false negative, letting a diseased person go undetected. By varying parameter $b$ a little at a time, curves like those shown are produced.

For a multi-class classifier, such as the blemish identifier described in Chapter 6, a popular method of visualising the classifier performance is a confusion matrix (Stehman, 1997). A confusion matrix, as seen in Table 3.1 is a table laid out in such a manner that the correct classification results are shown on the main diagonal, often in bold, with columns representing ground truth classes and rows representing classifier outputs. Cells not on the diagonal represent the *confusion* between two classes, with higher numbers representing more of the column class being wrongly classified as the row class.

As well as confusion matrices, multi-class classifiers can also be summarised using performance metrics designed for binary decisions. In the example summarised in Table 3.1, a summary might be made for the accuracy when classifying blemished skin

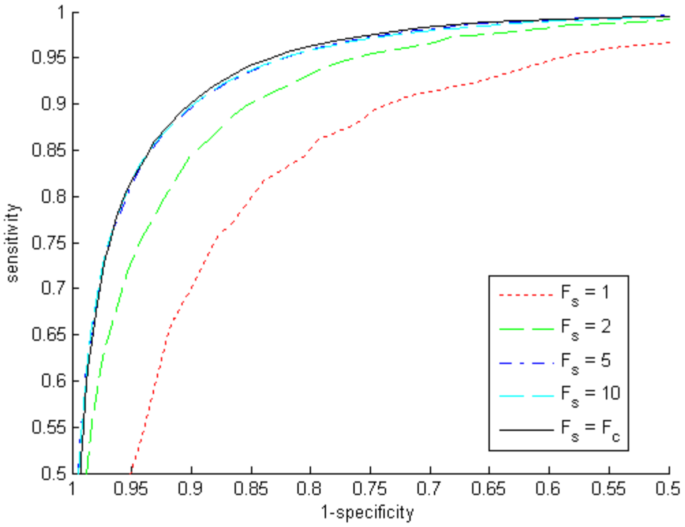


**Figure 3.3:** Examples of ROC curves representing experiments on blemish detection in white potatoes. As can be seen, manually increasing the sensitivity (likelihood of a positive result being classified as positive) results in the specificity (likelihood of a negative result being classified as negative) reducing faster the closer the sensitivity is to 1, and vice-versa. This example compares the performance of Minimalist AdaBoost, as described in Section 4.2.3, with different numbers of unique features ($F_s$) versus all of the available features ($F_c$) in the feature set.

as blemished skin overall, or for the confusion between two classes, such as between silver scurf and black dot. See also Section 6.2.1 where results for blemish identification are presented.

When the experiments are repeated with different sets of training and testing examples, as described in Section 3.4, these results can be summarised using the mean and standard deviation of the results from each training and testing set. A good classifier should not only have a high success rate but also a low standard deviation of its success rate, representing a consistently high success rate.

Finally it is often of interest to view which features are selected, for each class in blemish identification or which features are selected for blemish detection. This can give an insight into the kinds of features which best define a class. Visual inspection can often provide a reasonable explanation of why these features are chosen and prompt the inclusion of other features. As well as simply listing the features selected for each classifier, we carried out a set of controlled experiments to determine the influence of

| | | Classifier output | | | | |
|---|---|---|---|---|---|---|
| | | black dot | silver scurf | scab | green | unblemished |
| Markup | black dot | 83.8 ± 1.3 | 3.7 ± 0.6 | 0.5 ± 0.2 | 2.2 ± 1.2 | 9.8 ± 0.4 |
| | silver scurf | 5.7 ± 1.1 | 86.1 ± 1.0 | 0.8 ± 0.1 | 2.7 ± 0.7 | 4.7 ± 0.4 |
| | scabs | 2.1 ± 0.3 | 2.8 ± 0.6 | 90.5 ± 0.6 | 1.6 ± 0.4 | 3.0 ± 0.2 |
| | green | 1.1 ± 0.4 | 3.3 ± 0.4 | 0.3 ± 0.1 | 94.3 ± 0.5 | 0.9 ± 0.1 |
| | unblemished | 14.5 ± 1.7 | 12.4 ± 0.7 | 1.3 ± 0.2 | 6.5 ± 2.6 | 65.3 ± 2.3 |

**Table 3.1:** Example confusion matrix, showing classification results for red potatoes.

different subsets of features on the overall system; these "lesion experiments" (Kosslyn and Intriligator, 1992) were conducted by removing specific feature sets in turn and assessing the change in performance.

## 3.6 Summary

This chapter has described the experimental setup used to investigate computer vision algorithms for the purpose of detecting and identifying blemishes in potatoes. This includes image acquisition, image pre-processing and the chosen regime for training and testing including the human markup procedure.

The next chapter deals with the machine vision algorithms themselves, then Chapter 5 describes the experiments pertaining to the detection of blemishes and Chapter 6 describes the identification of five different classes of potato skin, including unblemished skin and four blemish types.

# 4

# Feature selection and classification

This chapter describes the main computer vision algorithms investigated in this thesis. The classifier used for blemish detection or identification is produced using a variant of the AdaBoost algorithm (Schapire and Singer, 1999) to select and combine multiple weak classifiers employing simple thresholds within the features described in Section 4.1, to produce a strong "combined" or "ensemble" classifier.

A machine learning solution was chosen due to the variety and evolving nature of potato crops, potato blemishes and lighting conditions. New varieties of potato are developed every year (Guo et al. (2009), Wu et al. (2010), Bizimungu et al. (2013)), and so a system which can be quickly and easily trained to work with a variety which has not been seen before has obvious advantages over one which requires expert intervention to be able to cope with the unexpected. Blemishes may also change over time, with rarer blemishes becoming commonplace or new strains emerging which the system would have to detect. Lighting can change for a number of reasons, with examples including legislation, such as the recent change in the regulation of filament lightbulbs and the turn of the seasons. Being able to retrain quickly also gives the opportunity to apply the approach to other types of produce.

After the images are obtained, they are first processed using a semi-supervised background removal process and scaled down by a factor of 2, as described in Section 3.3. A human expert then manually assigns ground truth markup to each pixel, after which the feature extraction process is begun.

## 4.1 Feature extraction

For these experiments, we used a collection of features for each pixel of interest and for a set of regions centered on the pixel itself. First a selection of pixel statistics were produced for every pixel in the image. At this stage, every pixel ends up being represented by a total of 28 statistics. Each region surrounding the pixel is then represented by five features summarising the pixels within that respective region. This allows the system to identify a blemish based on nearby characteristics as well as the pixel itself. This is of use when blemishes contain individual pixels that are similar but can be distinguished by nearby pixels, such as scab vs. silver scurf wherein similar individual pixels can be distinguished based on nearby pixel gradients. Square regions were used for simplicity, as described in Section 4.1.2.

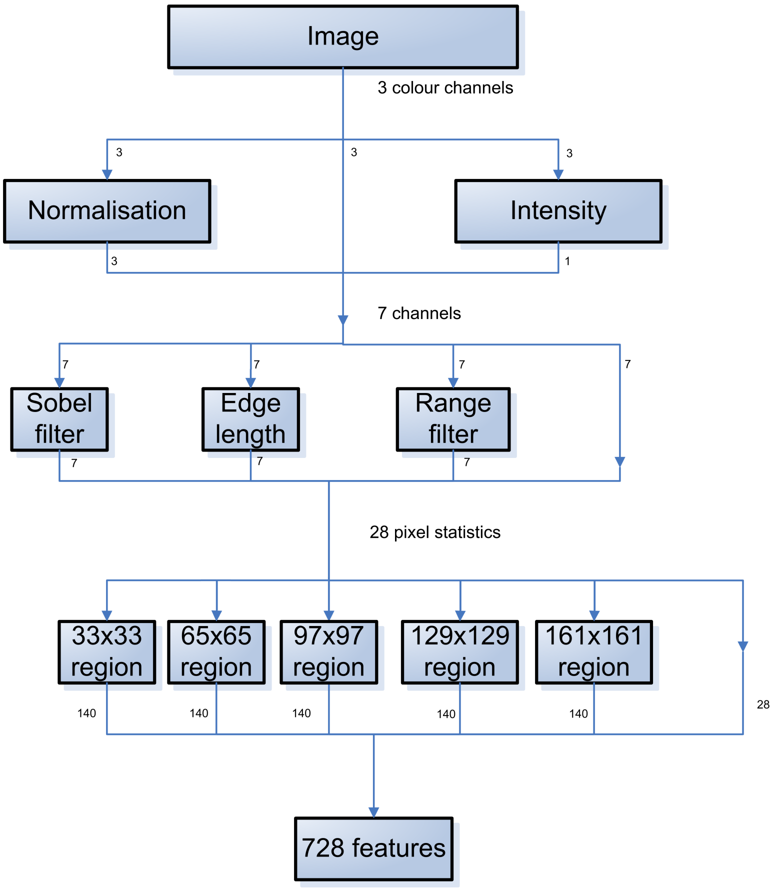


**Figure 4.1:** An overview of the process of extracting 728 features from an image.

As shown in Figure 4.1, we used 7 colour channels, RGB, normalised RGB and intensity × 4 feature types, colour, edge gradient, edge length and range × 5 regional statistics, mean, variance, skewness, minimum and maximum, making 140 statistics for each of 5 regions and 7 colour channels ×4 feature types = 28 features for the pixel itself. All these 728 features are used to provide the candidate feature set. These are described in detail as follows.

### 4.1.1 Pixel features

The proposed system uses the RGB colour space, the original colour format of the camera output, as used in Wijethunga et al. (2009) and Guannan et al. (2009). An alternative solution would be to use the HSV/HSI colour spaces as in Bolle et al. (1996) and Tao et al. (1995), or possibly CIELAB, as used in Mendoza and Aguilera (2004). CIELAB in particular is often chosen for its device-independent nature, but one of our interests in this system is that the learning classifier can easily adapt to changes in hardware, lighting or other conditions. For this reason we decided to try RGB before considering whether it was worth introducing a new colour model. Other systems use more complex hardware set-ups such as customised lighting, as is an option for the Maf-Roda Agrobotic System (Maf Roda Group, 2008) or specific colour filters like in Unay and Gosselin (2006).

Our system uses seven colour channels, which consist of red, green and blue; normalised red, green and blue; and the intensity channel. Normalised red, green and blue are produced by taking the three RGB channels from the original image and turning their absolute values into fractional values as in Equation 4.1.

$$r = \frac{R}{R+G+B},\ g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B},\ I = \frac{R+G+B}{3} \tag{4.1}$$

From these seven colour channels we consider the following image properties:

**Colour channels and intensity.** Intensity is especially of relevance for dark blemishes, e.g. black scurf or skin spot, while the most obvious blemish to be detected by other colour channels would be greening in white potatoes. Each pixel has seven features provided by the colour channels. Figure 4.2 shows an example of colour features highlighting a green blemish.

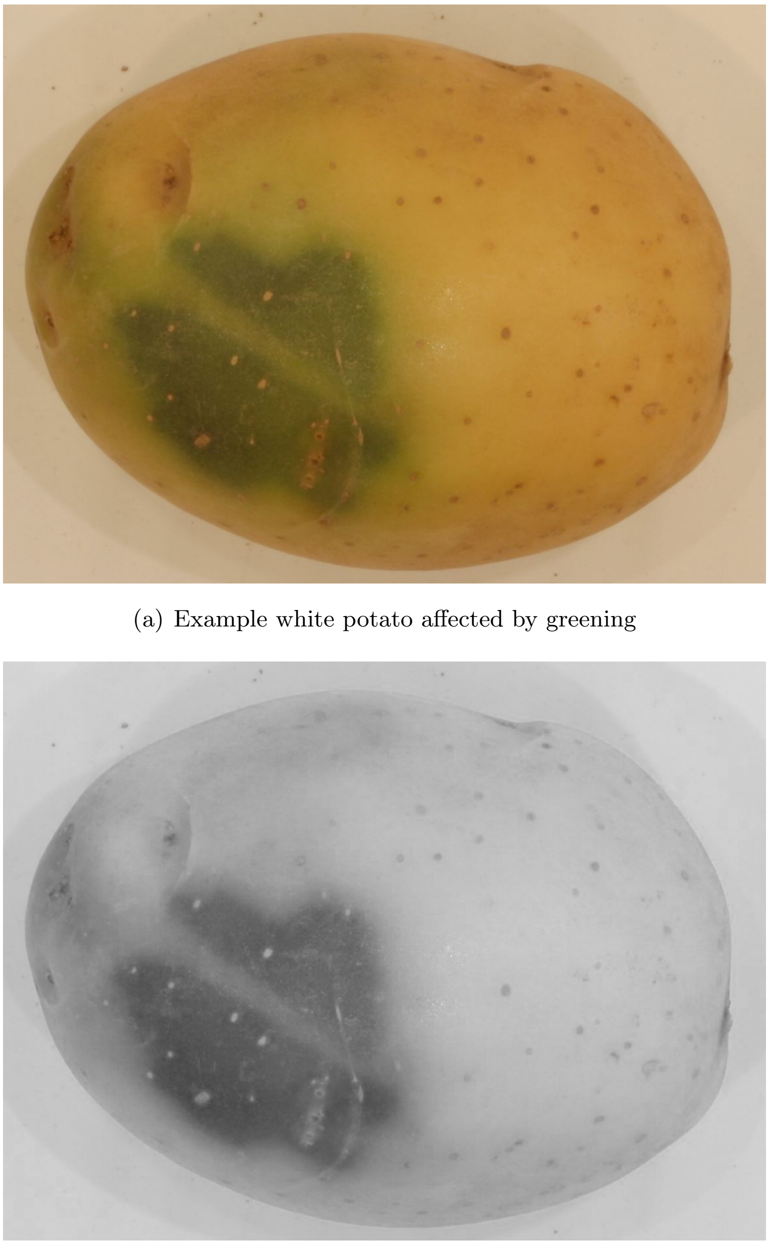
**Gradient filter.** An edge detector determines the rate of change of pixel values in a given neighbourhood in a specific direction. Some blemishes tend to coincide with high rates of change, such as powdery scab when the skin splits. The Sobel edge detector was used in this case with a standard 3 × 3 kernel size. The Sobel filter can be used to detect horizontal or vertical gradients depending on the orientation of the kernel. The gradient formulas for horizontal and vertical gradients, respectively, can be represented as follows:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A, \qquad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A, \tag{4.2}$$

where $*$ is the convolution operator and $A$ is the matrix being processed, in this case one channel (e.g. colour channel) of a given image.

The two orientations can also be combined to give an overall gradient by summing together the absolute output of both orientations $G = |G_x| + |G_y|$. This is of use when seeking rotational invariance since rotating a potato through 90 degrees would otherwise swap the horizontal and vertical gradients. The non-directional gradient filter was run on the same seven colour channels listed above, providing another seven features for each pixel. Figure 4.3 shows an example of gradient features for scabs.

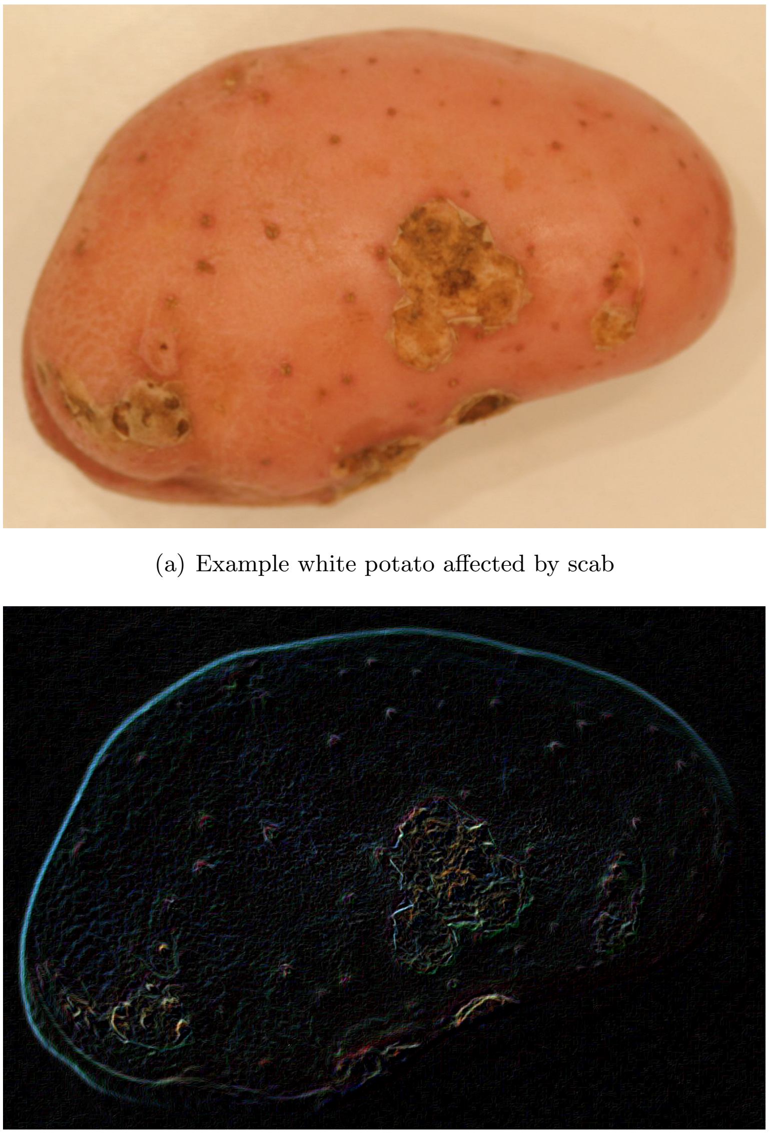
**Edge length.** Using the gradient feature, the edge length is determined by first detecting edges by thresholding the output of the Sobel filter. Pixels are first labelled as edge (high gradient) or non-edge (low gradient), then pixels marked as edge are used to form connected components. Each pixel within a connected region is then given a final value equal to the number of pixels which form that component. Non-edge pixels are given a value of zero. Larger edge components tend to be found around edges in particular. We used MATLAB's default threshold calculation of $T = 4 * \sqrt{mean(gx^2 + gy^2)}$ where $gx$ and $gy$ are the gradients in $x$ and $y$ axes, respectively. This operation provides another seven features per pixel. Figure 4.4 shows an example of edge length features for scabs. It can be seen that, while much of the image is dark, since non-edge pixels all have a value of 0, the longer edges are brighter since they have a higher value corresponding to their larger pixel count.



(a) Example white potato affected by greening

(b) The red colour channel can be isolated to highlight the lack of redness in one area of the image.

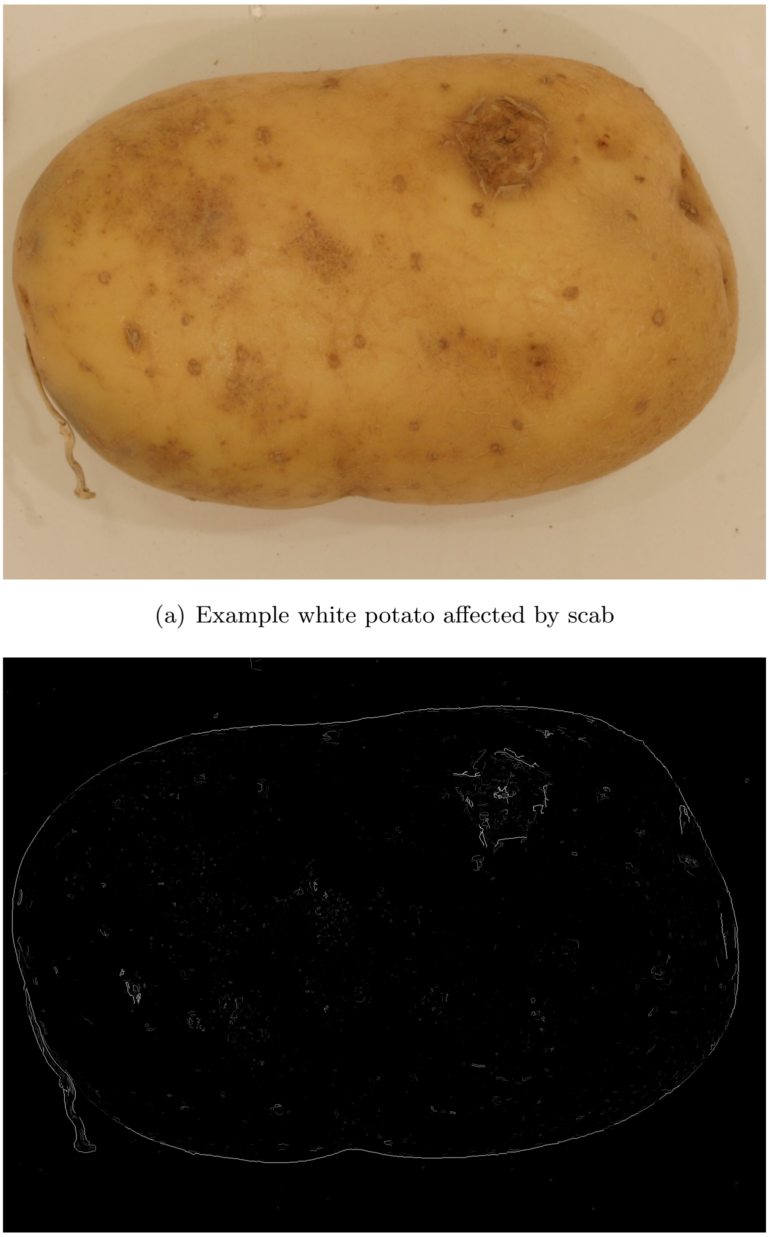
**Figure 4.2:** Colour channels work best on blemishes such as greening.



(a) Example white potato affected by scab

(b) The gradient feature highlighting the scabs.

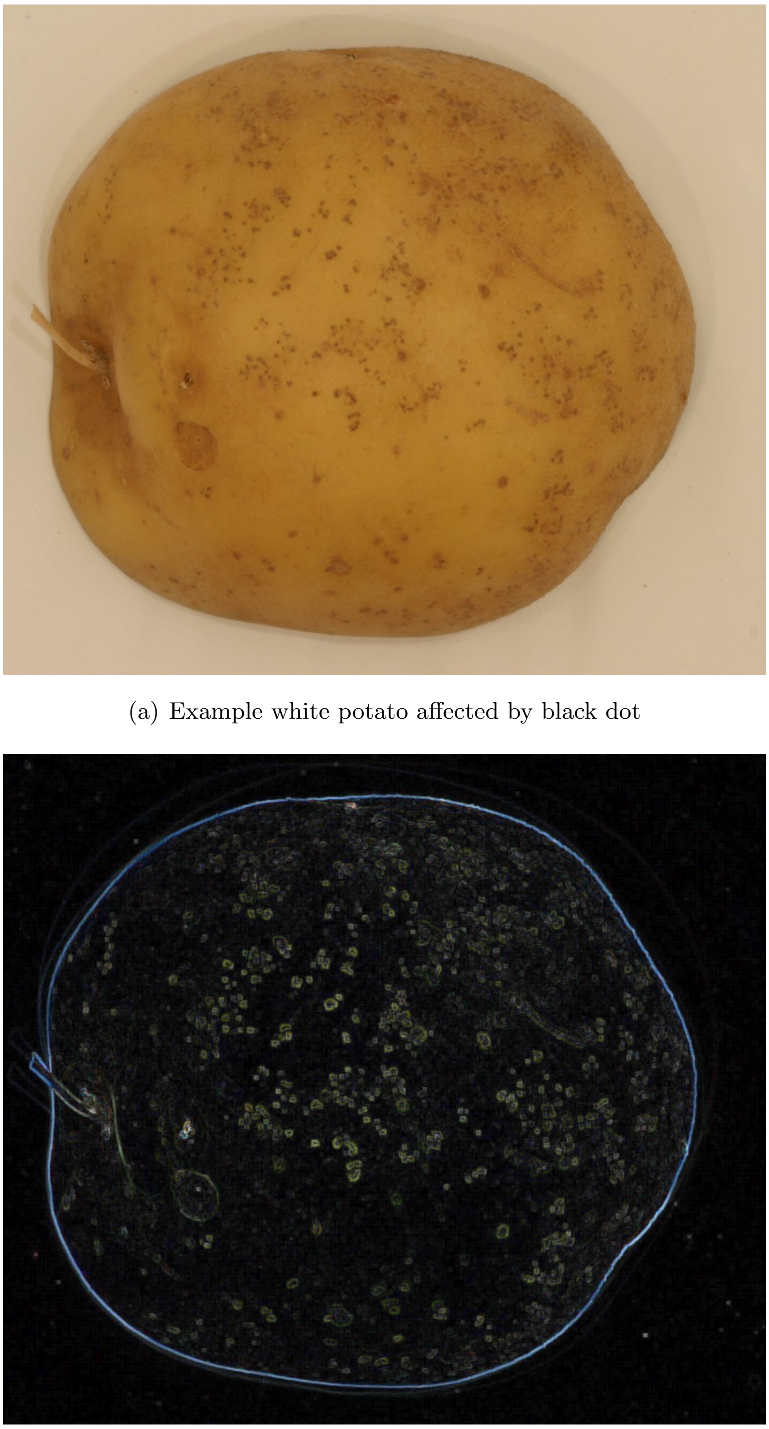
**Figure 4.3:** Gradient filter output for scabs.



(a) Example white potato affected by scab

(b) The edge length feature makes the scabs more visible.

**Figure 4.4:** Edge length feature for scabs.



(a) Example white potato affected by black dot

(b) The output of the range filter makes it easy to spot the individual patches of black dot.

**Figure 4.5:** Range filter output highlighting patches of black dot.

**Range filter.** The range filter is a basic texture filter which determines the maximum difference between pixel values in a given neighbourhood indicating the roughness of the texture. This neighbourhood is relative to each pixel, in this case a 5 × 5 square centered on the pixel. The output of the range filter is a matrix the same size as the input image, with each pixel value replaced with the range of its neighbourhood. Higher values tend to correspond to rougher, potentially damaged areas of the image. The range of the seven colour channels provides another seven features per pixel. Figure 4.5 shows an example of range features highlighting patches of black dot.

### 4.1.2 Regional statistics

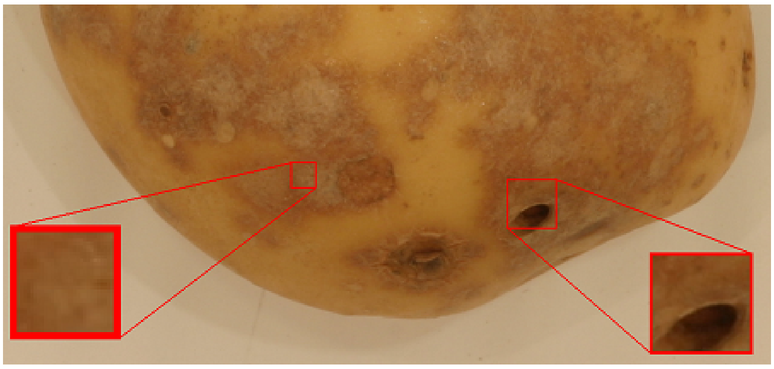


**Figure 4.6:** A section of an image showing regions of 33 × 33 and 65 × 65 on a potato affected by silver scurf, slug damage and scabbing.

Features extracted from a region of interest, such as those used in Kilic et al. (2007) or Savakar and Anamiy (2009) can contain more information than a single pixel. In early experiments, using only pixelwise features, classifiers often confused lenticels, tiny corky patches which allow air into a potato, with silver scurf. Lenticels, however, are very small and surrounded by an area of normal potato skin. Using regional statistics to add spatial context information from surrounding pixels made mis-classifications of this type far less common.

Other systems have summarised regions using only the mean of the region such as Tsai and Tsai (2002) or histograms as in Bolle et al. (1996).

For each pixel, the 28 pixelwise features described in Section 4.1.1 are combined with another five statistics for each of five regions centered on the pixel. These regions are squares of 33 × 33, 65 × 65, 97 × 97, 129 × 129 and 161 × 161. Examples of these regions are shown in Figure 4.6. The question of regions which overlap the image edge is not an issue since only potato pixels are processed and none of these come close enough to the image boundary. However the choice was made to exclude pixels which had been identified as background from regional calculations. The region sizes were chosen as multiples of 32, plus one in order to have a central pixel. Each of these regions was summarised using the mean, variance and skew as well as the minimum and maximum values within the area. Figure 4.6 shows the area of the first two examples in a potato image.

## 4.2 Classification

A classifier is an algorithmic process to assign a data point to one of a set of classes. The classifier will be presented with features pertaining to a datapoint to be classified, in this case a pixel from a potato, and it will return a classification decision, such as which kind of blemish is present if any. A machine learning classifier learns this decision function based on previously presented examples known as training data.

The classifiers used in our experiments are based on the Real AdaBoost algorithm (Schapire and Singer, 1998). In this thesis an extension of the Real AdaBoost algorithm was developed to reduce the number of overall features required to be extracted from every image. This provides an alternative control for reducing the complexity of the AdaBoost-based classifier instead of simply reducing the maximum number of weak classifiers. By reducing the number of unique features used instead of the number of weak classifiers, we achieve a higher speed with a lower reduction in accuracy than would be achieved by reducing the number of weak classifiers.

### 4.2.1 AdaBoost

Real AdaBoost proposed by Schapire and Singer (1998) is a generalisation of the AdaBoost algorithm that provides a lower error rate by allowing weak classifiers to vote by their individual degree of certainty instead of simply voting "yes" or "no". In addition, Real AdaBoost supports a bias so that, for example, if false negative results are considered less desirable than false positive results, then the classifier can be adjusted accordingly. This bias $b$ is also used to generate the ROC curves as described in Section 3.5.

**Algorithm 1** Real AdaBoost learning algorithm.

```
Given a dataset $S = \{(x_1, y_1), ...(x_m, y_m)\}$ where $x_i \in X$ and $y_i \in \{-1, +1\}$, the weak
classifier pool $K$, containing all possible weak classifiers from $F_c$ candidate features,
a specific number of weak classifiers to be chosen $T$
Initialise the sample distribution $D_1(i) = 1/m$
For $t = 1, ...T$
    1. For each weak classifier $h$ in $K$ do:
        a. Partition $X$ into several disjoint blocks $X_1, ..., X_n$
        b. Using the weights in distribution $D_t$ calculate

               $W_l^j = P(x_i \in X_j, y_i = l) = \sum_{i:x_i \in X_j, y_i = l} D_t(i)$

           Where $l = \pm 1$
        c. Set the output of $h$ on each $X_j$ as

               $\forall x \in X_j, h(x) = \frac{1}{2} ln \left( \frac{W_{+1}^j + \epsilon}{W_{-1}^j + \epsilon} \right)$

        d. Calculate the normalisation factor

               $Z = 2 \sum_j \sqrt{W_{+1}^j W_{-1}^j}$

    2. Select the $h_t$, minimising $Z$ i.e.

               $Z_t = \min_{h \in K} Z$

               $h_t = \arg\min_{h \in K} Z$

    3. Update the sample distribution

               $D_{t+1}(i) = D_t(i) exp \left[ -y_i h_t(x_i) \right]$

       and normalise $D_{t+1}$ to give a probability distribution function.
The final strong classifier $H$ is

               $H(x) = sign \left[ \sum_{t=1}^{T} h_t(x) - b \right]$

The confidence of $H$ is defined as

               $Conf_H(x) = \left[ \sum_{t+1}^{T} h_t(x) - b \right]$
```

As described in Algorithm 1, Real AdaBoost applies weights to training examples in order to simulate using a different training set each iteration. Weights are increased for wrongly classified examples and reduced for correctly classified examples. This provides a similar effect to selecting a new training dataset containing more examples which are harder to classify.

As described in the algorithm, the Real AdaBoost algorithm used in this thesis takes a set of training examples and assigns each example an equal weight. These weights are all set to 1, then normalised to produce a probability distribution function (p.d.f.) by dividing each weight by the sum of all weights.

At this point, the system begins its first iteration. The system iterates through each feature in turn, testing potential weak classifiers representing every way the data set can be subdivided using one threshold on that feature. For example, a data set containing the feature values 1, 1, 4, 6 would be tested using thresholds at 2.5 and 5, these numbers being the midpoints between consecutive unique values. For each potential weak classifier, the system calculates the error produced in terms of the weighting of all misclassified examples.

If an exit condition is met, in this case the training error being zero, then this is where the system will exit. If not then the weighting is recalculated as $D_{t+1}(i) = D_t(i) exp\left[-y_i h_t(x_i)\right]$, where $y_i$ is the classification ground truth and $h_t(x_i)$ is the result of classifying the datapoint $x_i$ using the new weak classifier. The weighting is then normalised as before and another iteration is carried out.

When the exit condition is met or a predefined maximum number of iterations has been reached, the final hypothesis is defined as $H(x) = sign\left[\sum_{t=1}^{T} h_t(x) - b\right]$, with the bias value, $b$, defaulting to zero, which can be adjusted up or down if false negatives are considered to be better or worse than false positives. A positive $b$ will tend to result in more false negative results and fewer false positives, while a negative $b$ will have the opposite result.

### 4.2.2 Multiclass AdaBoost

AdaBoost has been used in a variety of methods to produce a multi-class solution. Examples have included adapting binary classifiers to produce a multiclass approach, most commonly either by producing binary trees which use a selection of binary classifiers to progressively reduce the number of classes that can be the correct one before finally reaching a binary decision that only leaves one possible class. Another method involves a "one-against-all" approach, whereby one classifier is produced for each class, with each classifier's output somehow compared to each other classifier's output to find the most likely class (Allwein et al., 2000).

As there are many binary AdaBoost variants, there are also many explicitly multi-class AdaBoost variants. Three of these, AdaBoost.M1, AdaBoost.M2 Freund and Schapire (1996) and SAMME Zhu et al. (2009), each work by testing each available weak classifier's effect on the overall weighted error of the current combined classifier. SAMME and AdaBoost.M1 are both very close to the original AdaBoost algorithm extended to allow for the possibility that a weak classifier now has more than two possible outputs. SAMME features a change in calculation of weights which will allow classifiers to be combined so long as they are better than random guesses, rather than better than 50% accuracy as specified in AdaBoost and preserved in AdaBoost.M1. AdaBoost.M2 is designed to respond to many of the limitations of AdaBoost.M1, including the need for weak classifiers which only output a class decision. Under AdaBoost.M2 each weak classifier returns a vector of confidences, with each representing how likely the weak classifier considers each class as being correct. It is these vectors which are combined to produce a final set of likelihoods for each class, and for calculating the new weighting of training samples between iterations of AdaBoost.M2.

At the time of choosing a multiclass approach, initial tests that were performed on the smaller set of features in use at the time did not find any one multiclass AdaBoost approach to be significantly better than all of the others, with several implementations giving results of around 79% for white potatoes and nothing outperforming that benchmark at the time. From these top performers, the decision was made to continue using a modified version of the blemish detection framework, adapted to use multiple classes by means of a one-against-all implementation of Real AdaBoost, since it required the least modification of our approach. Additionally, the initial tests for this prototype took less than half the time of the next fastest top performer, which was an important consideration.

The one-against-all approach is commonly used with binary classification techniques, comparing the certainties of different binary classifiers to provide a multi-class approach (Bishop and Nasrabadi, 2006). Most classifiers, including Real AdaBoost and other variants produce a "confidence value" which is higher the more likely the result is of being positive and lower the more likely it is negative. This confidence value then allows the decision to be weighted, for instance if a false negative is considered a much worse result than a false positive, adding a constant to the confidence value will increase the required certainty that a result is negative before returning a negative result. In a one-against-all multiclass solution, the multiclass decision involves returning as a result the class associated with the binary classifier that produces the highest confidence value.

### 4.2.3 AdaBoost and classification speeds

The option to limit the number of weak classifiers that AdaBoost combines into a single strong classifier offers a method for increasing the overall speed of a classification process. Several other methods have been developed to speed the training stages of AdaBoost, usually by reducing the size of the training data set, whether by selecting a random subset each iteration, or through means such as Weighted Novelty Selection (Seyedhosseini et al., 2011) which produces a reduced summary with higher initial weights applied to examples representing more datapoints.

In this thesis we observed that over 99% of the processor time involved in the use of the trained system was for the process of feature extraction, so we focussed on approaches that could reduce the number of features used. This included limits on the number of weak classifiers, but also an approach we call “minimalist AdaBoost” whereby the system is allowed to continue to train once it has reached a limited number of features, but can only then select weak classifiers which re-use features already selected.

This approach was inspired by the observation that tests using only ten weak classifiers often used less than ten features. Additionally, AdaBoost had been explored as a feature selection method for Support Vector Machines (SVM) by Littlewort et al. (2006), in research relating to facial expression classification. They observe comparable performance between AdaBoost and a SVM, with some results showing AdaBoost outperforming the SVM and others not. When training the SVM only on features selected by AdaBoost, referred to as AdaSVM, the classification accuracy increases in all experiments. Their research also compares the time and memory requirements of an SVM trained on all features versus an SVM trained on AdaBoost selected features, with the latter being at least 450 times faster and requiring only 3.6% of the memory requirements (90 seconds vs. 0.2 seconds and 90 megabytes vs. 3.3 megabytes, respectively). Our research suggests that it can also be helpful to use AdaBoost to select features for AdaBoost itself.

Since this approach was first published, Mathanker et al. (2011) has shown that a minimalist AdaBoost approach can also give good classification rates for identifying defects in pecan nuts. Rather than simply employing the approach of reducing the pool of weak classifiers to those using features which have already been selected once the feature count reaches $F_s$, Mathanker tries experiments using the features selected using five iterations of one classifier to train another classifier. Ultimately, AdaBoost based systems are shown to produce similar results (92.2% - 92.3%) compared to SVM (90.1% - 92.7%), but ten times faster.

## 4.3 Summary

Now the main computer vision algorithms have been described, looking at the process of turning a potato image into a set of features and the automatic selection of features to create a classifier which can be used to describe the amount of blemish affecting the skins of different potatoes.

The next chapter describes experiments pertaining to the detection and measurement of potato blemishes, with Chapter 6 exploring processes to differentiate between five different blemish types as well as unblemished potato skin.

# 5

# Blemish detection

This chapter describes the experiments undertaken to distinguish potato blemishes from good potato skin. For this purpose, sample images were obtained from which a selection of features were extracted for every pixel. The images were segmented according to an expert markup system and only those pixels known to be either blemish or good potato were used. The potato sets were then divided by images into sets of 10 to 12, with each set in turn used as a testing set for a classifier trained using all the other sets. The overall accuracy of classifying pixels as blemish or non blemish was 91.3% ± 0.2 using white potatoes and 88.9% ± 0.4 using red potatoes. After obtaining this result, experiments were done into the effects of using a reduced number of features, both by the traditional method of restricting the number of iterations used by AdaBoost and by the minimalist AdaBoost approach. In addition, "lesion experiments" were undertaken whereby specific feature types were removed from the feature set. When restricting AdaBoost to a maximum of $n$ features, minimalist AdaBoost outperformed regular AdaBoost. When excluding certain feature types, the best results were obtained using colour and gradient features for white potatoes or colour and range features for red potatoes.

## 5.1 Experiments

These experiments were performed using a hold-one-out strategy described in Devroye et al. (1996) and implemented as follows. The white potato data was divided into 10 groups (9 groups containing 10 images and 1 containing 12 images), while the red potato data set was divided into 4 groups of 12 images. For each group, 3,000 positive and 3,000 negative examples was randomly selected. Then, each group was used in turn as a testing set for a classifier trained on examples from all remaining groups.

The training data for both red and white potato experiments consisted of 27,000 positive and 27,000 negative examples. For the white potato experiments, this consisted of 3,000 positive examples and 3,000 negative examples from each of 9 out of the 10 groups of images, with testing data being 3,000 positive and 3,000 negative examples from the remaining group. For red potatoes the training data consisted of 9,000 positive and 9,000 negative examples from each of 3 out of the 4 groups. This process was carried out once for each groups with a different group being used for testing each time. The decision to use groups of 10-12 images was taken to provide a good selection of blemish types in both training and testing sets for all experiments.

Since this training data involved less than 10% of the total data, it was possible to run the experiment again with different data by applying an offset to the subset of data selected from the total available potato data. The results in Section 5.2 show the mean and standard deviation of experiments using ten different subsets of the available training and testing data.

In this stage of the experimentation we examined the impact of different experimental setups on detecting blemish pixels versus non-blemish pixels using ground truth image data marked up by an operator following the guidance of a potato expert. In terms of the classifier itself we compared the effect of changing the number of iterations of a Real AdaBoost classifier and the number of unique features used by a minimalist Real AdaBoost classifier.

As well as studying the specific features chosen by AdaBoost classifiers, there were also “lesion experiments” to examine the impact of removing specific feature types. A full list of feature categories and the results of testing with each subset can be found in Tables 5.4 and 5.5 for white and red potatoes, respectively.

## 5.2 Results and discussion

The results of these experiments are given in Table 5.1, as well as ROC curves in Figure 5.1. The sensitivity and specificities represented in the ROC curves can also be found, along with graphical representations of the accuracies, in Appendix A.1. These results

| Feature limit | Non-minimalist | | Minimalist | |
|---|---|---|---|---|
| | White | Red | White | Red |
| 1 | 0.807 ± 0.005 | 0.777 ± 0.003 | 0.807 ± 0.005 | 0.777 ± 0.003 |
| 2 | 0.807 ± 0.005 | 0.785 ± 0.004 | 0.869 ± 0.002 | 0.802 ± 0.002 |
| 5 | 0.873 ± 0.006 | 0.821 ± 0.004 | 0.902 ± 0.001 | 0.858 ± 0.004 |
| 10 | 0.896 ± 0.002 | 0.858 ± 0.004 | 0.911 ± 0.002 | 0.886 ± 0.004 |
| 40 | 0.913 ± 0.002 | 0.889 ± 0.004 | 0.913 ± 0.002 | 0.889 ± 0.004 |

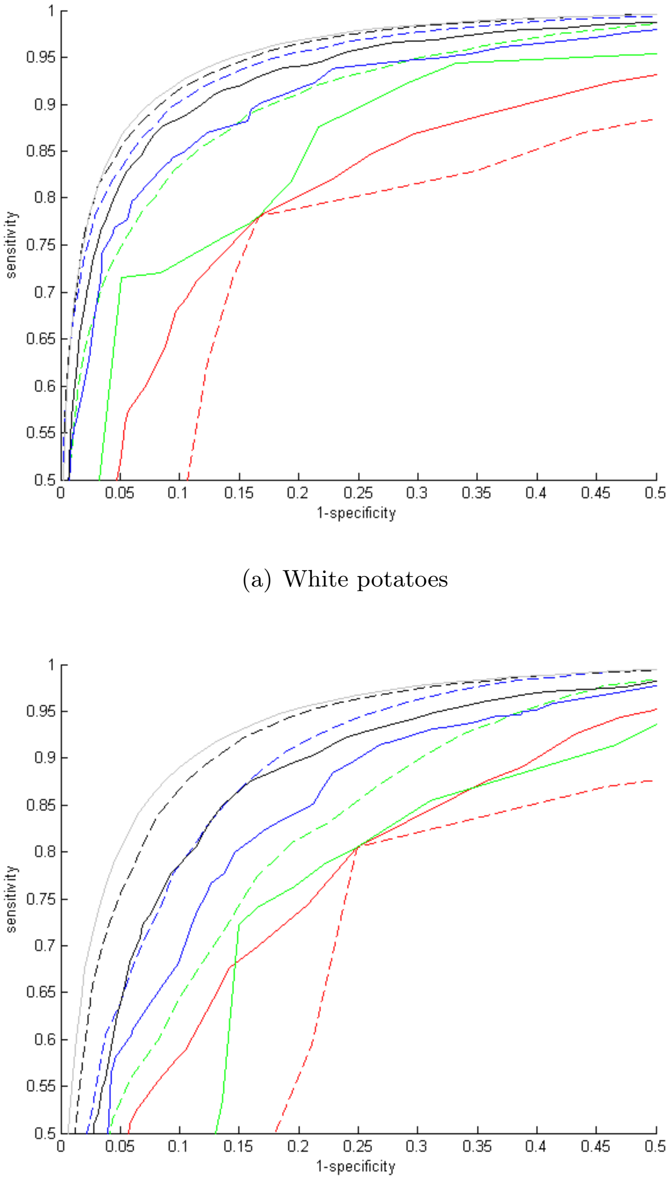
**Table 5.1: Accuracy** of these experiments comparing two methods of restricting the workload of the classifier, by selecting a smaller number of weak classifiers (up to 40) or by selecting 40 weak classifiers with a restricted number of unique features. For the last row the results show no difference since the restriction of 40 unique features does not change the features selected by only 40 weak classifiers.

indicate that, by allowing AdaBoost to re-use features, the accuracy of a classifier using a smaller number of features can be increased over and above that of simply limiting the number of weak classifiers as is standard across AdaBoost variants. This allows a higher accuracy to be obtained at a similar classification speed. The main cost of such an approach would be in the time taken to train the system, which would be of less concern in a real world industrial application than the time saved when classifying potatoes since it would be expected that the system should be trained once at the beginning of a batch and then left for several hours to classify tonnes of produce.

### 5.2.1 Speed

Because of the extra system overhead of running in the MATLAB environment, this prototype system runs much slower than would be required in a commercial system. However it is still of interest to investigate algorithmic approaches that can be used to speed up classification. Table 5.1 and Figure 5.1 already show the effects on classification rates of reducing the number of features. The difference in speeds between the minimalist and non-minimalist AdaBoost approaches is shown in Figure 5.4 for white and red potatoes.

For white potatoes, minimalist AdaBoost using 5 features provides a better classification rate than non-minimalist AdaBoost using 10 weak classifiers, with the minimalist approach achieving a 97% increase in speed (396 pixels per second vs. 201 pixels per



(a) White potatoes

(b) Red potatoes

**Figure 5.1:** ROC curves describing the error rates of blemish detection in white and red potatoes, using the error rates given in Appendix A.1, Tables A.1 and A.2. Dashed lines represent success rates for minimalist AdaBoost, solid lines non-minimalist. Red, green, blue, black and grey represent 1, 2, 5, 10 and 40 features or weak classifiers, respectively. The ROC curves indicate the false positive and false negative rates available by weighting the result. The dashed lines being higher than the solid lines in all but the lowest number of features indicates that minimalist AdaBoost makes more effective use of fewer features than non-minimalist AdaBoost. As discussed in Section 5.2.1, the number of features presents most of the CPU overhead in this system and thus minimalist AdaBoost is shown to produce a better result at an equivalent speed to non-minimalist AdaBoost.

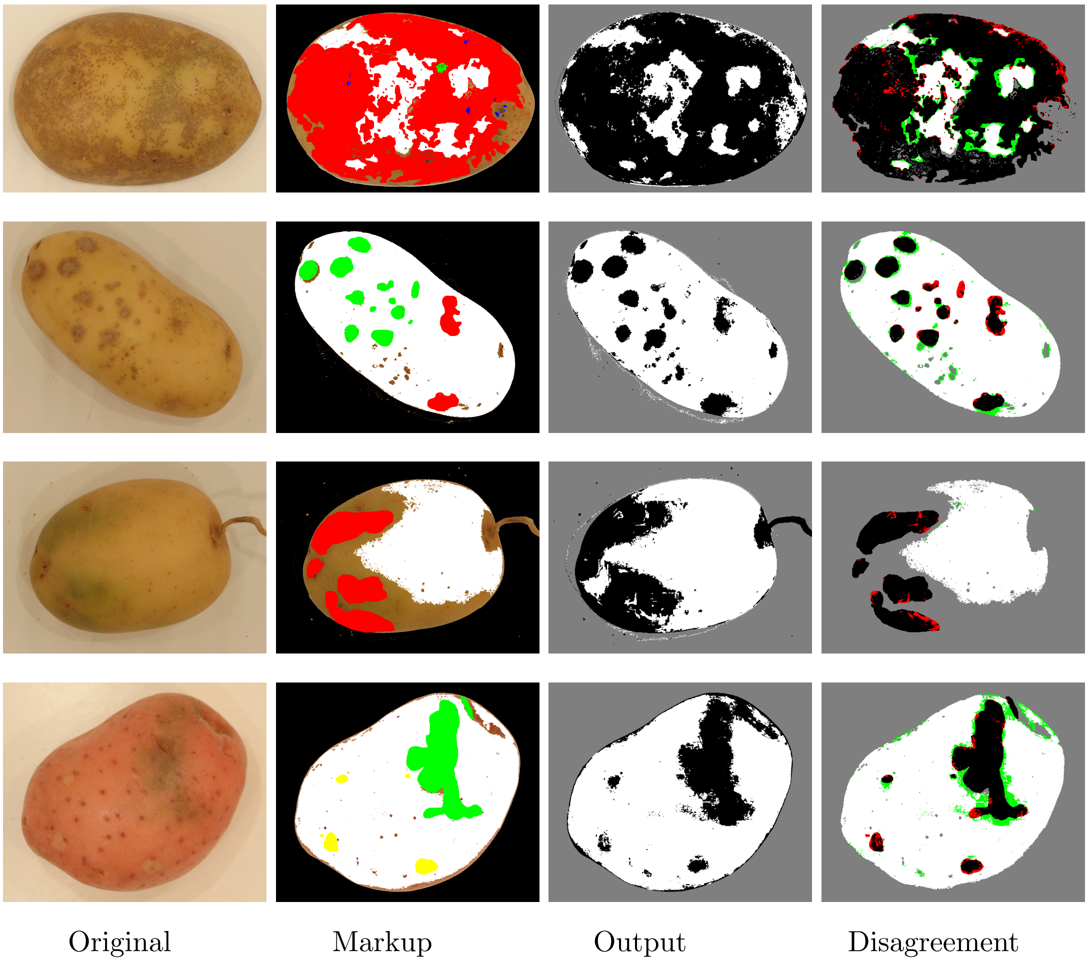
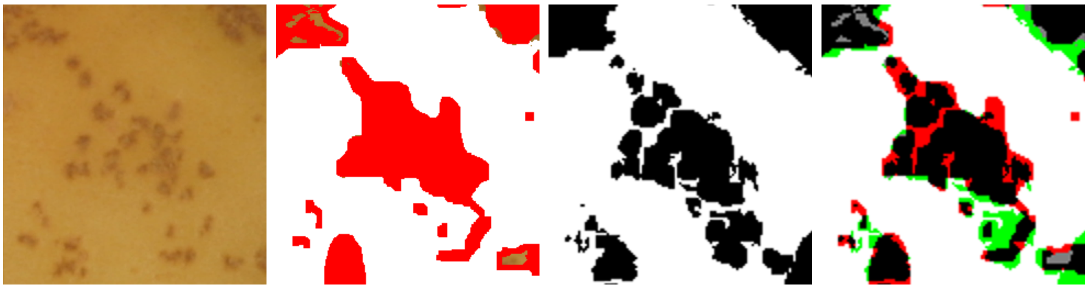
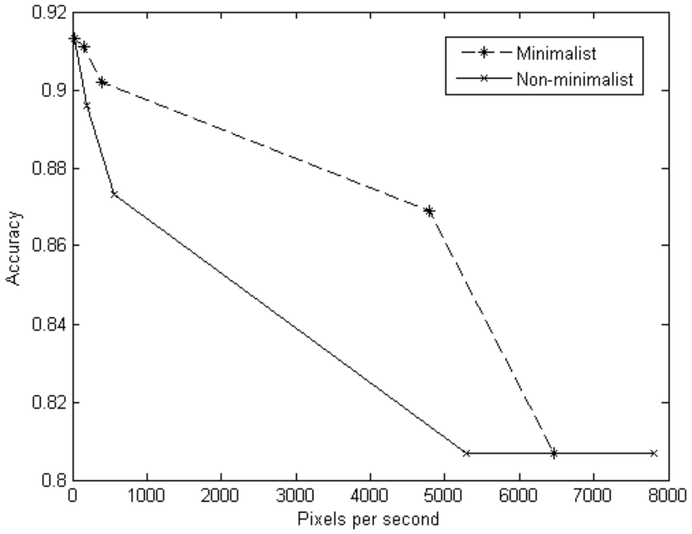


**Figure 5.2:** Images showing (left to right): first an original photograph, then a "ground truth" image labelled manually by a human expert, then the outputs of the trained system, without any human editing, with blemishes in black and good potato in white. The final image shows an error image, showing false positive results in red and false negative results in green, for the detection of blemish.

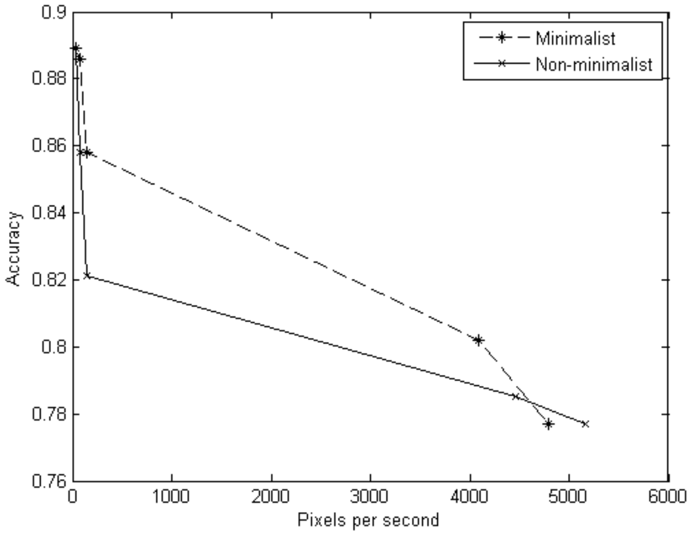


Original Markup Output Disagreement

**Figure 5.3:** Zoomed-in view of the middle of the first image set in Figure 5.2 indicating that the classifier output is possibly a better match to the blemish than the ground truth.



(a) White potatoes



(b) Red potatoes

**Figure 5.4:** Graphs showing the classification accuracy versus the combined speed of feature extraction and classification for minimalist and non-minimalist approaches using (a) white and (b) red potatoes.

second) and minimalist AdaBoost using 2 weak classifiers achieves a speed increase of 748% over non-minimalist AdaBoost using 5 features with no significant drop in accuracy.

For red potatoes, minimalist AdaBoost using 5 features provides a similar classification rate to non-minimalist AdaBoost using 10 weak classifiers, with the minimalist approach achieving a 100% increase in speed (75 vs. 151 pixels per second)

For both red and white potatoes, the difference in accuracy between a minimalist approach and the best performance using 40 weak classifiers is within one standard deviation but the increases in classification speed are 304% for white potatoes and 141% for red potatoes, increasing from 36 to 147 and 31 to 75 pixels per second, respectively. The minimalist approach only becomes worse than the non-minimalist equivalent when restricted to a single feature.

### 5.2.2 Features selected

The selected features listed in Table 5.2 for white potatoes and Table 5.3 for red potatoes show some interesting correlations and obvious distinctions. To begin with, red potatoes use colour as an indicator of blemish a lot more than white potatoes, with twice as many of the top ten selected features being colour data. In both red and white examples, the pixel colour provides the second most important feature, while the most important feature is related to the intensity of the immediate area. In both cases, normalised colours are selected three times. Intensity is used more for red potatoes which may relate to the fact that green blemishes tend to appear black on red potatoes. Classifying the red potato set, the edge length feature is not used, while for classifying white potatoes the range filter does not get used. The regions used tend to be the smaller ones, with only one feature region above 65 × 65 for white potatoes and three for red potatoes. In each case the larger region is 161 × 161.

Tables 5.4 and 5.5 describe the impact of specific feature groups on classification rates. The features are divided up by feature type and subsets of features are used in these lesion experiments. These subsets are referred to by up to four letters, representing the presence of colour (c), range (r), gradient (g) and edge length (l), so the full feature set is referred to as crgl. For both potato varieties, the lowest accuracy of any combination of colour and texture features is higher than the accuracy for colour

| Order selected | Region | Feature type | Channel | Statistic |
|---|---|---|---|---|
| 1 | 33 × 33 | Gradient | Intensity | Mean |
| 2 | Pixel | Colour | Red | |
| 3 | 65 × 65 | Gradient | Normalised Green | Skew |
| 4 | Pixel | Gradient | Red | |
| 5 | 65 × 65 | Colour | Normalised Green | Skew |
| 6 | 33 × 33 | Gradient | Red | Max |
| 7 | 65 × 65 | Colour | Green | Mean |
| 8 | 161 × 161 | Edge Length | Normalised Blue | Var |
| 9 | 33 × 33 | Edge Length | Red | Skew |
| 10 | 65 × 65 | Edge Length | Red | Skew |

**Table 5.2:** The first ten features selected for detecting blemishes in white potatoes.

features which is higher than any combination of only texture features. The highest accuracy for white potatoes is from a combination of colour and gradient features, while the highest accuracy was achieved for red potatoes using colour and range features. Figures A.3 and A.4 show these results as bar charts.

### 5.2.3 Evaluating the impact of variations in ground truth markup

To assess the impact of natural variation between human experts involved in providing the ground truth data, we used a combination of markup data obtained from three different humans for a selected subset of the original data for white potatoes. The details of the experimental procedure involved were as explained in Section 3.3.2

Table 5.6 shows the overall accuracy of classification obtained using the selected data for blemish detection. Note that the highest result was obtained with the original expert who had access to the original potato samples. The next highest result was obtained with the second potato expert, who only had access to the images. The lowest result was obtained with the computer vision researcher, who only had access to the images and a more limited knowledge of potatoes. The "gold standard" result obtained from the agreement of at least 2 markers on each pixel is similar to the result for the original expert (marker 1). The percentage of pixels in the panel refers to the agreement between the gold standard and the human marker on those pixels for which

| Order selected | Region | Feature type | Channel | Statistic |
|---|---|---|---|---|
| 1 | 33 × 33 | Colour | Intensity | Min |
| 2 | Pixel | Colour | Red | |
| 3 | 65 × 65 | Gradient | Normalised Blue | Skew |
| 4 | 161 × 161 | Range | Intensity | Skew |
| 5 | 33 × 33 | Range | Red | Mean |
| 6 | 33 × 33 | Colour | Normalised Red | Skew |
| 7 | 65 × 65 | Colour | Red | Skew |
| 8 | Pixel | Colour | Normalised Green | |
| 9 | 161 × 161 | Colour | Red | Mean |
| 10 | 161 × 161 | Range | Intensity | Var |

**Table 5.3:** The first ten features selected for detecting blemishes in red potatoes.

| Feature subset | Accuracy | Sensitivity | Specificity |
|---|---:|---:|---:|
| c | 0.891 ± 0.001 | 0.890 ± 0.001 | 0.890 ± 0.001 |
| r | 0.875 ± 0.002 | 0.880 ± 0.002 | 0.880 ± 0.002 |
| g | 0.891 ± 0.002 | 0.889 ± 0.001 | 0.889 ± 0.001 |
| l | 0.841 ± 0.002 | 0.847 ± 0.001 | 0.847 ± 0.001 |
| cr | 0.911 ± 0.002 | 0.910 ± 0.002 | 0.910 ± 0.002 |
| cg | **0.913 ± 0.002** | 0.912 ± 0.001 | **0.912 ± 0.001** |
| cl | 0.904 ± 0.002 | 0.903 ± 0.001 | 0.903 ± 0.001 |
| rg | 0.888 ± 0.002 | 0.890 ± 0.002 | 0.887 ± 0.002 |
| rl | 0.880 ± 0.001 | 0.875 ± 0.002 | 0.885 ± 0.002 |
| gl | 0.890 ± 0.003 | 0.887 ± 0.001 | 0.887 ± 0.001 |
| crg | 0.912 ± 0.002 | **0.915 ± 0.002** | 0.910 ± 0.003 |
| crl | 0.911 ± 0.002 | 0.911 ± 0.002 | 0.911 ± 0.002 |
| cgl | **0.913 ± 0.003** | 0.911 ± 0.002 | 0.911 ± 0.002 |
| rgl | 0.889 ± 0.003 | 0.886 ± 0.002 | 0.886 ± 0.002 |
| crgl | **0.913 ± 0.003** | 0.912 ± 0.002 | **0.912 ± 0.002** |

**Table 5.4:** A comparison of different feature groups for detecting blemishes in **white** potatoes, comprised from colour(c), edge gradient(g), edge length(l) and range(r), where the bold font is used to indicate the best results for each performance metric.

| Feature subset | Accuracy | Sensitivity | Specificity |
|---|---|---|---|
| c | 0.878 ± 0.004 | 0.905 ± 0.003 | 0.851 ± 0.009 |
| r | 0.817 ± 0.003 | 0.843 ± 0.002 | 0.790 ± 0.005 |
| g | 0.820 ± 0.002 | 0.845 ± 0.004 | 0.795 ± 0.005 |
| l | 0.791 ± 0.002 | 0.825 ± 0.005 | 0.757 ± 0.003 |
| cr | **0.889 ± 0.002** | **0.917 ± 0.002** | **0.862 ± 0.006** |
| cg | 0.886 ± 0.004 | 0.912 ± 0.002 | 0.859 ± 0.008 |
| cl | 0.876 ± 0.003 | 0.904 ± 0.003 | 0.848 ± 0.006 |
| rg | 0.824 ± 0.004 | 0.845 ± 0.003 | 0.804 ± 0.008 |
| rl | 0.818 ± 0.002 | 0.848 ± 0.003 | 0.788 ± 0.005 |
| gl | 0.816 ± 0.004 | 0.841 ± 0.006 | 0.791 ± 0.005 |
| cgl | 0.884 ± 0.004 | 0.914 ± 0.003 | 0.855 ± 0.009 |
| crg | 0.886 ± 0.003 | 0.914 ± 0.003 | 0.858 ± 0.006 |
| crl | **0.889 ± 0.004** | 0.915 ± 0.003 | 0.862 ± 0.008 |
| rgl | 0.823 ± 0.003 | 0.850 ± 0.003 | 0.797 ± 0.005 |
| crgl | 0.886 ± 0.003 | 0.914 ± 0.003 | 0.858 ± 0.006 |

**Table 5.5:** A comparison of different feature groups for detecting blemishes in **red** potatoes, comprised from colour(c), edge gradient(g), edge length(l) and range(r), where the bold font is used to indicate the best results for each performance metric.

| Human | Marker 1 | Marker 2 | Marker 3 | Panel |
|---|---|---|---|---|
| Accuracy | 0.913 ± 0.002 | 0.906 ± 0.005 | 0.902 ± 0.003 | 0.911 ± 0.004 |
| Average marked pixels | 134502 | 122323 | 115585 | 135989 |
| Pixels in panel markup | 99.4% | 96.0% | 93.9% | 100.0% |

**Table 5.6:** Accuracy for three different markers and for a "gold standard" mark-up produced from the combination of all three markers, as well as the number of marked pixels per image and the percentage agreement with the gold standard. Marker 1 is the same markup used for the main set of experiments in this thesis.

both list a value. The highest agreement is between the original expert and the gold standard, while the marker with less experience of potatoes has the lowest agreement with the panel.

## 5.3 Conclusions

Since this system required much more time for feature extraction than for classification, the minimalist AdaBoost approach allows a system to provide a higher classifier accuracy at a faster speed compared to a similar non-minimalist system, providing a speed increase of 4 times and 2.5 times for white and red potatoes, respectively. The difference is most apparent when five features are used, indicating that this would be a suitable choice for applications where classification performance and computational cost are both important. The use of minimalist AdaBoost using ten features reduced accuracy by one standard deviation from the use of non-minimalist AdaBoost using 40 weak classifiers.

The most important feature sets overall are a combination of colour and texture features, which always outperform a subset of either colour or texture features alone. The exact choice of feature type is less significant, but the highest performance is always achieved with a combination of colour and texture features, rather than either colour or texture features taken alone. The list of selected features reflects the differences between red and white potatoes, most notably with twice as many colour features being used for white potatoes, since blemishes can cause a greater overall change in the colour of white potatoes than red.

As shown in Figures 5.2 and 5.3, a portion of the error rate is visibly due to disagreements with the exact pixel boundaries of blemishes rather than objective misclassification. This is because the blemishes do not always have clear and precise boundaries, which makes a decision on the "correct" ground truth category quite subjective.

# 6

# Blemish identification

Having succeeded in detecting which areas of potatoes were blemished, the next stage of this research was to distinguish between the blemishes commonly present in our data sets. We decided to separate the blemishes into the subcategories of black dot; silver scurf; green blemishes, which consisted of both pathological greening and sprouting; and scabs. Both powdery scab and common scab were combined into a single class since they are physically similar, often requiring high levels of magnification or chemical analysis to tell the two apart (De Haan and van den Bovenkamp, 2005).

Experiments were also undertaken looking at the trade-offs for time and accuracy with limits on feature numbers. As well as minimalist AdaBoost approaches, experiments were tried using AdaBoost trained on the features previously selected for all classes by minimalist AdaBoost. This provided a higher success rate than a similarly fast minimalist AdaBoost classifier, both of which were several times faster than a general non-minimalist classifier.

## 6.1 Experiments

For these experiments, each data set was divided into two subsets. The white potato images were divided into subsets of 50 and 52 images, respectively, while the red potato images were divided into two subsets of 24 images each. This was done instead of the previous hold-one-out approach in order to ensure sufficient representation of all blemishes in both the training and testing sets. As previously, it was necessary to manually check the white potato set for those cases where the same potato had been photographed more than once, in order to make sure that different potatoes were used for training and testing of the classifier, while the red potato set only contained one image per potato. 34 cases were found where a white potato had been imaged twice from different side. The two groups for each data set were chosen using alternating images to provide a uniform sample for each group. Both groups were then used for training and testing in turn.

When implementing minimalist AdaBoost for multi-class blemish identification, a binary classifier was trained for each class with 10,000 positive examples and 10,000 negative examples, being uniformly sampled from the entire training data set. The 10,000 negative examples were selected from all other blemish types, resulting in 2500 examples per class. The final classification decision was determined by the classifier with the highest confidence, which is also known as a one-against-all approach.

### 6.1.1 Non-minimalist classifier

The use of five standard Real AdaBoost classifiers, trained on all available candidate features for 40 iterations for a one-against-all approach provides a good general performance, albeit the slowest. This is treated as a default, tried and tested classifier approach.

### 6.1.2 Minimalist classifier

As with the detection approach, limiting the number of features selected by a classifier without applying a similar limitation to the number of iterations provides a much better speed to performance ratio than simply limiting the number of iterations. Experiments were run using both 5 and 10 unique features per classifier. In the same way as described in Section 4.2.3, this was achieved by keeping separate counts of the number of features and the number of iterations and, once the number of features exceeded the selected limit, by limiting further iterations to training from those features which had already been used.

### 6.1.3 Preselected features, the best of both worlds?

The use of Minimalist AdaBoost presented an interesting opportunity. While the system restricted the number of features available for speed purposes, this speed gain would be mostly seen in the classification stages. A typical approach using 5 features per classifier would give a total of between 18 and 22 unique features, so by adding another stage to the training process, the system could be trained on a number of features equivalent to 5 per class, while providing a more complete feature set for each class individually.

To implement this feature-sharing approach, the training process had to be extended significantly, which was deemed acceptable since only a small fraction of the system's time would be spent training and the additional time savings extracting features at classification time would amply make up for the loss. The result is a combination of the non-minimalist and minimalist one-against-all classifiers described above. First the minimalist classifier was trained and discarded, keeping only the lists of features used. These lists were then combined and duplicates removed, then the list was used to restrict the features used for training in the same manner as the lists used in the lesion experiments.

For completeness, these lists were used both with a non-minimalist classifier and with a minimalist classifier trained using 10 unique features from the selected list.

After determining that the use of pre-selected features gave comparable performance to minimalist AdaBoost at a significantly higher speed, further lesion experiments were performed, testing the effects of removing entire feature types from the data set. Silver scurf and black dot were best classified using a mixture of colour and texture features, while green blemishes in white potatoes were classified best using only colour, and scabs by using all texture features.

## 6.2 Results

As before, we were interested in both the actual classification rates and the features selected by our AdaBoost-based classifier.

### 6.2.1 Classification rates

Table 6.1, and Figure A.17, show the accuracy and speed of different classification methods, with the most accurate and slowest being the use of a non-minimalist AdaBoost classifier using the one-against-all approach described above. There is no significant

| Classifier | White | | Red | |
|---|---|---|---|---|
| | Accuracy | Pixels/s | Accuracy | Pixels/s |
| Non-minimalist | 86.0 ± 0.5 | 7.1 ± 0.7 | 84.6 ± 0.5 | 6.6 ± 0.5 |
| Minimalist (10) | 84.6 ± 0.4 | 24.4 ± 3.4 | 83.3 ± 0.7 | 16.9 ± 2.0 |
| Minimalist (5) | 82.5 ± 0.9 | 46.6 ± 9.0 | 79.4 ± 1.1 | 33.6 ± 8.4 |
| Selected features | 84.9 ± 0.8 | 46.6 ± 9.0 | 82.6 ± 0.8 | 33.6 ± 8.4 |
| Selected features minimalist | 84.6 ± 0.8 | 46.6 ± 9.0 | 82.5 ± 0.8 | 33.6 ± 8.4 |

**Table 6.1:** Overall accuracy for red and white potatoes using minimalist and non-minimalist AdaBoost, as well as using a minimalist and non-minimalist AdaBoost trained on the features selected by minimalist AdaBoost.

difference between the classification rates for minimalist AdaBoost using 10 weak classifiers per class or non-minimalist AdaBoost using preselected features. As shown in Figures 6.1 and 6.2, the difference between AdaBoost using preselected features and using all features is equally spread across all classes, with the largest difference being for classes with the largest standard deviations in results from experiments using the same classifier.

The highest overall accuracy for both red and white potatoes uses a non-minimalist classifier, achieving accuracies of 86.0% ±0.5 for white potatoes and 84.6% ± 0.5 for red potatoes, respectively.

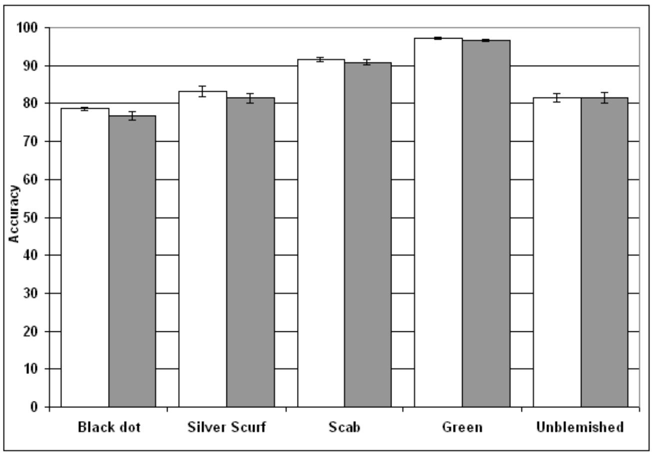


**Figure 6.1:** Bar charts indicating the percentage of each blemish type correctly identified, per pixel, for white potatoes, using minimalist (white) and non-minimalist (grey) classifiers.

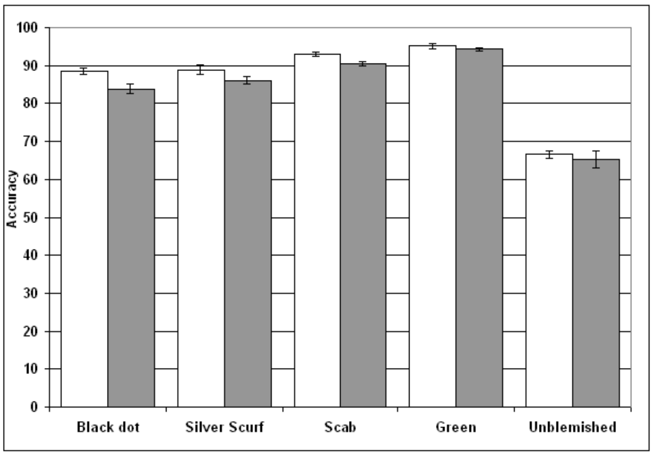


**Figure 6.2:** Bar charts indicating the percentage of each blemish type correctly identified, per pixel, for red potatoes, using minimalist (white) and non-minimalist (grey) classifiers.

| | | Classifier output | | | | |
|---|---|---|---|---|---|---|
| | | black dot | silver scurf | scab | green | unblemished |
| Markup | black dot | **83.8 ± 1.3** | 3.7 ± 0.6 | 0.5 ± 0.2 | 2.2 ± 1.2 | 9.8 ± 0.4 |
| | silver scurf | 5.7 ± 1.1 | **86.1 ± 1.0** | 0.8 ± 0.1 | 2.7 ± 0.7 | 4.7 ± 0.4 |
| | scabs | 2.1 ± 0.3 | 2.8 ± 0.6 | **90.5 ± 0.6** | 1.6 ± 0.4 | 3.0 ± 0.2 |
| | green | 1.1 ± 0.4 | 3.3 ± 0.4 | 0.3 ± 0.1 | **94.3 ± 0.5** | 0.9 ± 0.1 |
| | unblemished | 14.5 ± 1.7 | 12.4 ± 0.7 | 1.3 ± 0.2 | 6.5 ± 2.6 | **65.3 ± 2.3** |

**Table 6.2:** Confusion matrix for red potatoes using features preselected by minimalist AdaBoost

Tables 6.2 to 6.5 are confusion matrices showing how each experiment errs in classifying individual pixels of one class as another class. The rows of each matrix represent the classes according to human markup while the columns represent the classes according to the classifier, the cells along the diagonal that contain bold text are therefore the correctly classified examples, while all off-diagonal cells contain errors. In white potatoes, both minimalist and non-minimalist approaches have the same general error trends, towards the confusion of black dot with silver scurf, both of which tend to be silvery blemishes, as well as the confusion of black dot with unblemished potato in both directions. The confusion matrices for red potatoes, in comparison, show most of the pixel-wise mis-classification being between black dot and unblemished skin.

Intuitively, the lower confusion between silver scurf and black dot on red potatoes

| | | Classifier output | | | | |
|---|---|---|---|---|---|---|
| | | black dot | silver scurf | scab | green | unblemished |
| Markup | black dot | **88.6**% ± **0.8** | 2.9% ± 0.6 | 0.5% ± 0.2 | 1.4% ± 0.3 | 6.7% ± 0.5 |
| | silver scurf | 5.3% ± 0.9 | **88.9**% ± **1.2** | 0.5% ± 0.1 | 2.3% ± 0.5 | 2.9% ± 0.2 |
| | scabs | 1.6% ± 0.3 | 2.4% ± 0.3 | **93.0**% ± **0.5** | 0.8% ± 0.3 | 2.3% ± 0.2 |
| | green | 0.7% ± 0.2 | 3.3% ± 0.6 | 0.1% ± 0.1 | **95.1**% ± **0.6** | 0.8% ± 0.1 |
| | unblemished | 15.6% ± 1.3 | 11.5% ± 0.8 | 1.3% ± 0.1 | 5.1% ± 0.8 | **66.6**% ± **1.0** |

**Table 6.3:** Confusion matrix for red potatoes using non-minimalist AdaBoost trained on all features

| | | Classifier output | | | | |
|---|---|---|---|---|---|---|
| | | black dot | silver scurf | scab | green | unblemished |
| Markup | black dot | **76.7 ± 1.1** | 7.6 ± 0.6 | 5.3 ± 0.7 | 2.6 ± 0.5 | 7.8 ± 0.2 |
| | silver scurf | 8.5 ± 0.4 | **81.4 ± 1.2** | 6.8 ± 0.9 | 0.7 ± 0.3 | 2.6 ± 0.1 |
| | scabs | 1.2 ± 0.2 | 5.9 ± 0.5 | **90.8 ± 0.7** | 1.1 ± 0.5 | 1.0 ± 0.1 |
| | green | 0.8 ± 0.1 | 0.2 ± 0.1 | 0.2 ± 0.1 | **96.6 ± 0.3** | 2.3 ± 0.2 |
| | unblemished | 8.6 ± 0.5 | 3.3 ± 0.2 | 1.4 ± 0.1 | 5.2 ± 1.3 | **81.6 ± 1.4** |

**Table 6.4:** Confusion matrix for white potatoes using features preselected by minimalist AdaBoost

| | | Classifier output | | | | |
|---|---|---|---|---|---|---|
| | | black dot | silver scurf | scab | green | unblemished |
| Markup | black dot | **78.6**% ± **0.5** | 7.0% ± 0.7 | 4.2% ± 0.5 | 2.6% ± 0.3 | 7.6% ± 0.3 |
| | silver scurf | 7.2% ± 0.4 | **83.1**% ± **1.4** | 6.5% ± 1.2 | 0.7% ± 0.2 | 2.4% ± 0.1 |
| | scabs | 1.0% ± 0.1 | 5.6% ± 0.3 | **91.6**% ± **0.5** | 0.8% ± 0.2 | 1.0% ± 0.1 |
| | green | 0.6% ± 0.1 | 0.1% ± 0.1 | 0.1% ± 0.0 | **97.3**% ± **0.3** | 1.9% ± 0.3 |
| | unblemished | 8.8% ± 0.5 | 3.4% ± 0.2 | 2.0% ± 0.2 | 4.4% ± 0.9 | **81.5**% ± **1.1** |

**Table 6.5:** Confusion matrix for white potatoes using non-minimalist AdaBoost trained on all features

compared to white potatoes, is most likely to be because silver scurf generally has a different colour on red potatoes, making it easier to distinguish. As seen in Tables 6.2 and 6.3, the amount of black dot misclassified as silver scurf was 3.7% using preselected features or 2.9% using all features. Likewise in those experiments, silver scurf was misclassified as black dot at a rate of 5.7% and 5.3%, respectively. This compares to the same experiments using white potatoes, as shown in Tables 6.4 and 6.5, where black dot was misclassified as silver scurf in 7.6% and 7.0% of examples and silver scurf was misclassified as black dot in 8.5% and 7.2% of examples when using preselected features and all features, respectively. The confusion between good potato and black dot is at least partly due to the limitations of the human ground truth data, as shown in Figures 5.2 and 5.3 where black dot especially, as a speckled blemish, is prone to being incorrectly marked by hand. The first example classified both by hand and using this classifier shown in Figure 6.3 is a good illustration of this. In this case the human has overlooked the slight darkening to the right of the potato which may have been caused by the black dot blemish, which the classifier reports as the case. Aside from the small patches of scab toward the edge of the potato this can be seen as a successful classification, however it only has a pixel-wise accuracy of 84.1% due to the disagreement over the exact extent of black dot coverage.

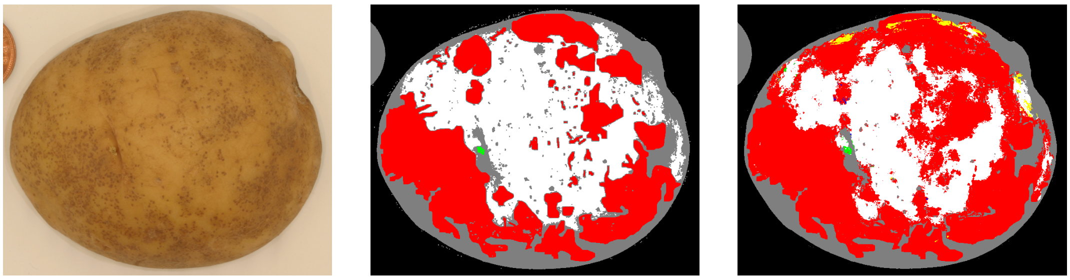
Table 6.6 supports the idea that inaccurate ground truth may be reducing success rates. When detecting the presence or absence of blemishes comprising 10% of the pixels of each potato, the results of the first non-minimalist experiment provides a much higher success rate per image than per pixel. The only class which is worse represented is silver scurf, which suffers from false positives due to the confusion with black dot, which tends to cover very large areas of an affected potato, allowing a smaller amount of confusion to still equate to 10% of the potato. Overall, detecting blemish types per potato provides an accuracy of 93% for white potatoes and 97% for red potatoes, respectively, compared to per-pixel accuracies of 86% and 84%, respectively.

Figure 6.4 shows the performance of the classifier on three more example images. In the first two of these the main causes of errors are the ambiguity of the exact blemish coverage, along with one missed area of greening in the first potato. The third example also mis-classifies a particularly sparse area of black dot as a mixture of scab and unblemished skin.

| | White Potatoes | Red Potatoes |
|---|---|---|
| Black dot | 95% | 100% |
| Silver Scurf | 75% | 92% |
| Scab | 99% | 100% |
| Green | 100% | 94% |
| Good potato | 95% | 100% |
| Overall | 93% | 97% |

**Table 6.6:** Detecting significant (10%) coverage per class is more accurate than detecting blemish in each pixel individually.

Table 6.1 shows the results in terms of accuracy and speed for different classifier types. For both red and white classifiers, a non-minimalist classifier gives the best performance, however a significant speed increase is possible for a relatively small comparative loss of accuracy. Should a final system require such a speed increase, the use of preselected features appears to offer the best combination of speed and accuracy.



(a) Original potato

(b) Ground truth done by hand

(c) Classifier output

**Figure 6.3:** A potato classified by the system, showing the similarity between classifier output and ground truth, with differences in the exact boundaries of blemished areas. Red = black dot, green = green, white = unblemished, black = background

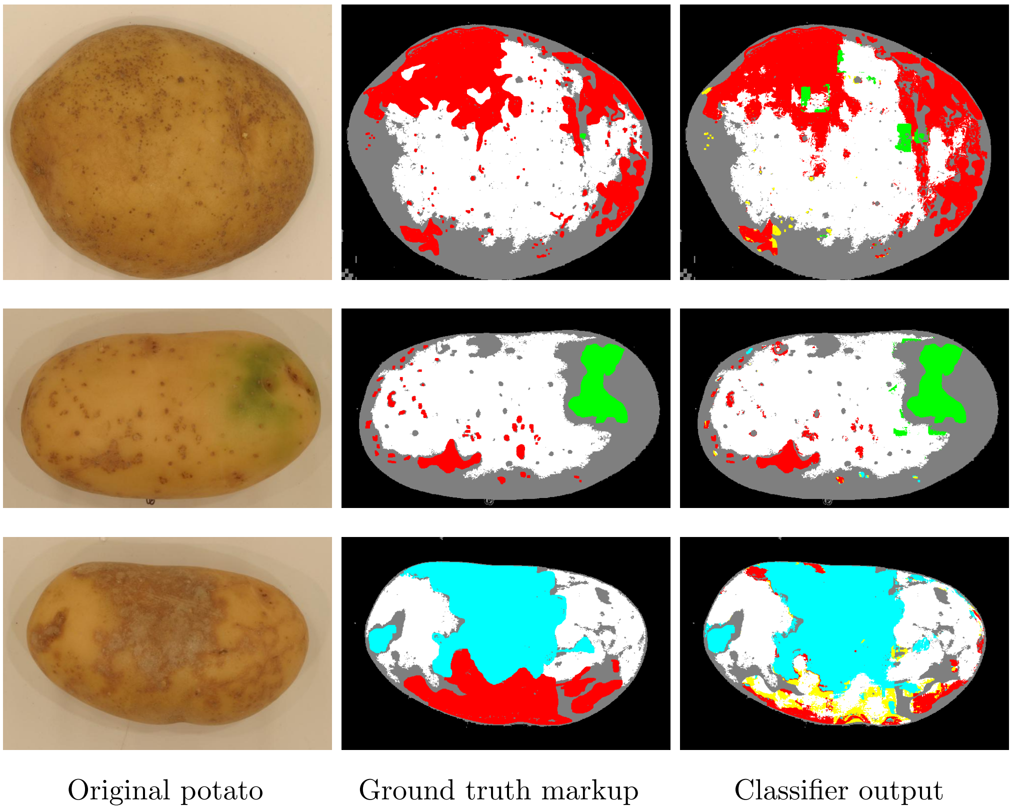


**Figure 6.4:** Three potatoes classified by the system, showing the similarity between classifier output and ground truth, with differences in the exact boundaries of blemished areas. Red = black dot, cyan = silver scurf green = green, white = unblemished, black = background.

### 6.2.2 Speed

As with blemish detection, experiments were run to compare the speed increases achieved by restricting the number of features used by the classifier. For this reason we chose five different sets of parameters.

As well as using non-minimalist AdaBoost, minimalist AdaBoost with 10 and with 5 features per class, experiments were also undertaken using Non-minimalist AdaBoost but restricted to the features selected by minimalist AdaBoost when restricted to 5 features per class.

As shown in Table 6.1, the speed of minimalist AdaBoost restricted to five feature per class and that of non-minimalist AdaBoost when allowed to train each class on any of those five features per class, is effectively the same. The performance of non-minimalist AdaBoost on this reduced feature set, however, is within a standard deviation of that of minimalist AdaBoost using 10 features per class, with a speed increase of 91% for white potatoes and 99% for red potatoes. Compared to the Non-minimalist approach, this represents a speed increase of 556% using white potatoes and 409% using red potatoes.

A further test using minimalist AdaBoost allowing 10 features per class from the 5 selected features gave similar results to the same approach using non-minimalist AdaBoost, within 0.4 standard deviations, with the only noticable difference being in training times that would be insignificant in any real-world application classifying thousands of potatoes.

### 6.2.3 Features selected

Tables 6.7 to 6.16 show the respective choices of features for identifying the class of blemish, or non-blemish in red and white potatoes using a representative set of training data.

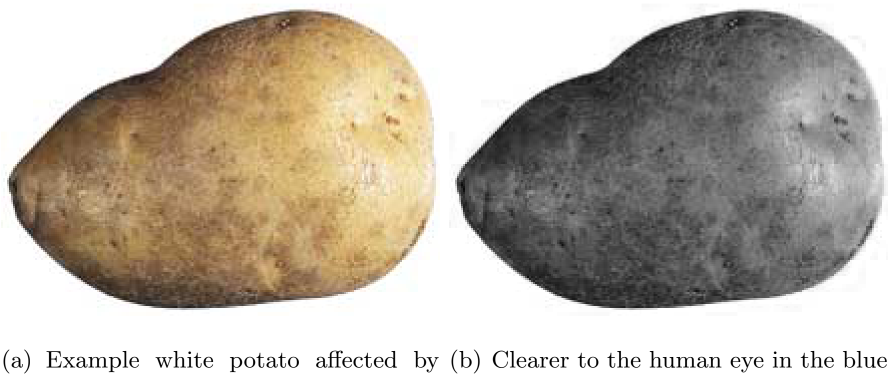
Some of the choices of features are intuitively understandable to a human observer. When classifying black dot in white potatoes and to a lesser extent in red potatoes, features relating to the blue colour channel are prominently featured. As shown in Figure 6.5, black dot is somewhat easier to see with the naked eye in the blue colour channel than in grayscale. Classifying silver scurf mostly uses colour features, which is understandable as it is primarily a silvery blemish, while the feature selection for scabs mostly selects edge and range features. In white potatoes, gradient features are prominently used in detecting scabs. As shown in Figure 6.6, scabs can be clearly seen with the naked eye in the output of an edge detector. In red potatoes, however, the colour and especially the red colour are the first features selected, representing the fact that scabs usually have a very different colour to that of red potato skin. It is quite self-explanatory that the first features selected for green blemishes is the green or normalised green colour channel. In red potatoes, skin greening results in darker, almost black patches which may indicate why the intensity channel is ranked second when detecting green blemishes in red potatoes.

| | Region | Type | Channel | Moment |
|---:|---:|---:|---:|---:|
| 1 | 161 × 161 | colour | red | max |
| 2 | 161 × 161 | gradient | normalised blue | mean |
| 3 | 65 × 65 | edge length | normalised blue | variance |
| 4 | 161 × 161 | colour | normalised blue | variance |
| 5 | 33 × 33 | gradient | intensity | mean |
| 6 | 161 × 161 | gradient | intensity | skewedness |
| 7 | 33 × 33 | colour | normalised red | variance |
| 8 | Pixel | colour | normalised green | - |
| 9 | 161 × 161 | gradient | red | mean |
| 10 | 161 × 161 | edge length | blue | min |

**Table 6.7:** Top ten features selected for identifying **black dot** in **white potatoes**.

| | Region | Type | Channel | Moment |
|---|---|---|---|---|
| 1 | 161 × 161 | colour | blue | mean |
| 2 | 161 × 161 | gradient | intensity | skewedness |
| 3 | 161 × 161 | colour | red | max |
| 4 | 161 × 161 | range | green | max |
| 5 | 129 × 129 | colour | green | max |
| 6 | 161 × 161 | colour | blue | max |
| 7 | 161 × 161 | range | intensity | skewedness |
| 8 | 65 × 65 | range | normalised green | mean |
| 9 | 161 × 161 | colour | blue | skewedness |
| 10 | 161 × 161 | range | blue | min |

**Table 6.8:** Top ten features selected for identifying **black dot** in **red potatoes**.



(a) Example white potato affected by black dot.

(b) Clearer to the human eye in the blue colour channel.

**Figure 6.5:** Example illustrating how the black dot blemish can be easily seen in the blue colour channel, explaining the blue colour channel being selected for the top two features in Table 6.7.

| Order selected | Region | Type | Channel | Moment |
|---|---|---|---|---|
| 1 | 161 × 161 | colour | intensity | mean |
| 2 | Pixel | colour | normalised green | - |
| 3 | 65 × 65 | range | normalised green | variance |
| 4 | 33 × 33 | colour | normalised red | variance |
| 5 | 33 × 33 | gradient | red | variance |
| 6 | 33 × 33 | edge length | normalised red | variance |
| 7 | 97 × 97 | colour | normalised green | skewedness |
| 8 | 161 × 161 | gradient | normalised blue | mean |
| 9 | 97 × 97 | edge length | normalised blue | min |
| 10 | 161 × 161 | gradient | intensity | variance |

**Table 6.9:** Top ten features selected for identifying **silver scurf** in **white potatoes**.

| Order selected | Region | Type | Channel | Moment |
|---|---|---|---|---|
| 1 | 161 × 161 | gradient | normalised blue | variance |
| 2 | 65 × 65 | colour | normalised blue | variance |
| 3 | 33 × 33 | colour | green | mean |
| 4 | 161 × 161 | gradient | normalised blue | skewedness |
| 5 | 161 × 161 | colour | blue | variance |
| 6 | 65 × 65 | colour | normalised green | variance |
| 7 | 33 × 33 | colour | green | variance |
| 8 | 161 × 161 | colour | normalised blue | min |
| 9 | Pixel | colour | green | - |
| 10 | 161 × 161 | edge length | intensity | skewedness |

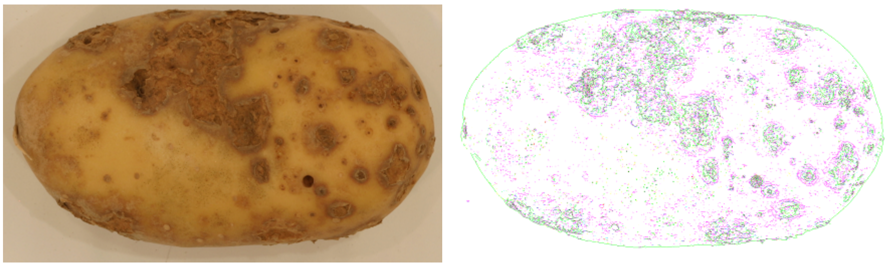
**Table 6.10:** Top ten features selected for identifying **silver scurf** in **red potatoes**.

| Order selected | Region | Type | Channel | Moment |
|---:|---:|---:|---:|---:|
| 1 | 33 × 33 | gradient | intensity | mean |
| 2 | 161 × 161 | gradient | blue | variance |
| 3 | 161 × 161 | edge length | blue | min |
| 4 | 161 × 161 | range | blue | min |
| 5 | 161 × 161 | gradient | normalised blue | min |
| 6 | 161 × 161 | edge length | intensity | variance |
| 7 | 33 × 33 | edge length | blue | min |
| 8 | 129 × 129 | colour | normalised green | skewedness |
| 9 | 161 × 161 | range | blue | mean |
| 10 | 33 × 33 | range | red | min |

**Table 6.11:** Top ten features selected for identifying **scab** in **white potatoes**.

| Order selected | Region | Type | Channel | Moment |
|---:|---:|---:|---:|---:|
| 1 | 33 × 33 | colour | normalised red | max |
| 2 | 161 × 161 | colour | intensity | variance |
| 3 | 97 × 97 | range | red | mean |
| 4 | 65 × 65 | gradient | blue | skewedness |
| 5 | Pixel | colour | normalised blue | - |
| 6 | 161 × 161 | gradient | blue | mean |
| 7 | Pixel | colour | normalised green | - |
| 8 | 97 × 97 | edge length | red | mean |
| 9 | 161 × 161 | gradient | normalised blue | variance |
| 10 | 33 × 33 | range | blue | skewedness |

**Table 6.12:** Top ten features selected for identifying **scab** in **red potatoes**.



(a) Example white potato affected by scab

(b) edge detector output can be clearly seen to correlate with the presence of scabs in a potato image, explaining the prevalence of edge features among the selected features in tables 6.11 and 6.12

**Figure 6.6:** The classifier uses edge features extensively for detection of scabs, these edges can be seen with the human eye.

| Order selected | Region | Type | Channel | Moment |
|---:|---:|---:|---:|---:|
| 1 | 65 × 65 | colour | green | min |
| 2 | 129 × 129 | gradient | normalised red | skewedness |
| 3 | Pixel | gradient | blue | - |
| 4 | 129 × 129 | colour | green | skewedness |
| 5 | 161 × 161 | colour | blue | mean |
| 6 | 65 × 65 | gradient | green | min |
| 7 | 33 × 33 | colour | blue | skewedness |
| 8 | 33 × 33 | colour | intensity | min |
| 9 | 161 × 161 | edge length | green | min |
| 10 | 161 × 161 | edge length | blue | min |

**Table 6.13:** Top ten features selected for identifying **green blemishes** in **white potatoes**.

| Order selected | Region | Type | Channel | Moment |
|---|---|---|---|---|
| 1 | 65 × 65 | colour | normalised green | skewedness |
| 2 | 161 × 161 | edge length | intensity | variance |
| 3 | Pixel | colour | red | - |
| 4 | 97 × 97 | gradient | normalised blue | skewedness |
| 5 | 161 × 161 | colour | normalised blue | max |
| 6 | 129 × 129 | colour | green | variance |
| 7 | 97 × 97 | colour | normalised red | variance |
| 8 | 129 × 129 | gradient | normalised green | max |
| 9 | 33 × 33 | gradient | normalised red | variance |
| 10 | 65 × 65 | range | normalised blue | skewedness |

**Table 6.14:** Top ten features selected for identifying **green blemishes** in **red potatoes**.

| Order selected | Region | Type | Channel | Moment |
|---|---|---|---|---|
| 1 | Pixel | colour | red | - |
| 2 | 33 × 33 | range | blue | mean |
| 3 | 65 × 65 | colour | normalised green | skewedness |
| 4 | 33 × 33 | edge length | normalised red | skewedness |
| 5 | 65 × 65 | colour | green | min |
| 6 | Pixel | gradient | red | - |
| 7 | 161 × 161 | gradient | red | mean |
| 8 | 33 × 33 | range | blue | skewedness |
| 9 | 33 × 33 | colour | intensity | mean |
| 10 | 161 × 161 | colour | intensity | skewedness |

**Table 6.15:** Top ten features selected for identifying **unblemished skin** in **white potatoes**.

| Order selected | Region | Type | Channel | Moment |
|---|---|---|---|---|
| 1 | 33 × 33 | colour | intensity | min |
| 2 | 129 × 129 | gradient | normalised green | skewedness |
| 3 | Pixel | colour | red | - |
| 4 | 33 × 33 | colour | normalised red | skewedness |
| 5 | 161 × 161 | edge length | intensity | skewedness |
| 6 | Pixel | colour | normalised green | - |
| 7 | 97 × 97 | colour | normalised green | mean |
| 8 | 97 × 97 | colour | intensity | skewedness |
| 9 | 65 × 65 | edge length | normalised blue | skewedness |
| 10 | 161 × 161 | range | red | variance |

**Table 6.16:** Top ten features selected for identifying **unblemished skin** in **red potatoes**.

As before, tests were also performed to investigate the impact of the main feature types available to the classifiers. Table 6.17 presents the overall accuracy for each subset of features, with Tables 6.18 to 6.19 providing breakdowns according to class. The overall results are also presented as bar charts in Appendix A.2, Figures A.5 and A.6, with the breakdowns according to class represented in Figures A.7 to A.16. Notably, while all classes have a lower accuracy with colour channels removed, green blemishes fare the worst in both red and white potatoes, while using only colour features in white potatoes only reduces the accuracy for green blemishes by half a standard deviation. Similarly, silver scurf on red potatoes is such a distinctive colour compared to all other classes means that using only colour features reduces the accuracy even less. In common with the results from blemish detection, detecting unblemished potato tends not to change much so long as the feature set contains colour features and some texture-based features. Scabs, being mostly marked out with broken skin, are the blemish least affected by the removal of colour features in white potatoes, although in red potatoes this is not the case due to the marked difference between skin and flesh colours. Overall, the best classification rate for red potatoes occurs with the combination of colour and gradient features (cg), while the best classification rate for white potatoes is from the combination of colour, range and edge length features (crl).

### 6.2.4 Evaluating the impact of variations in ground truth mark-up

To assess the impact of natural variation between human experts involved in providing the ground truth data, we used a combination of markup data obtained from three different humans for a selected subset of the original data for white potatoes. The details of the experimental procedure involved were as explained in Section 3.3.2

Table 6.20 shows the overall accuracy of classification obtained using the selected data for blemish identification. Note that the highest result was obtained with the original expert who had access to the original potato samples. The next highest result was obtained with the second potato expert, who only had access to the images. The lowest result was obtained with the computer vision researcher, who only had access to the images and a more limited knowledge of potatoes. The "gold standard" result obtained from the agreement of at least 2 markers on each pixel is similar to the result for the original expert (marker 1). The percentage of pixels in the panel refers to the agreement between the gold standard and the human marker on those pixels for which

| Feature subset | red | white |
|---|---|---|
| c | 0.806 ± 0.012 | 0.793 ± 0.005 |
| r | 0.636 ± 0.012 | 0.731 ± 0.011 |
| g | 0.647 ± 0.010 | 0.739 ± 0.014 |
| l | 0.496 ± 0.008 | 0.642 ± 0.009 |
| cr | 0.826 ± 0.013 | 0.843 ± 0.004 |
| cg | **0.828 ± 0.009** | 0.843 ± 0.010 |
| cl | 0.822 ± 0.006 | 0.805 ± 0.004 |
| rg | 0.650 ± 0.009 | 0.726 ± 0.014 |
| rl | 0.633 ± 0.009 | 0.707 ± 0.006 |
| gl | 0.646 ± 0.008 | 0.704 ± 0.006 |
| crg | 0.823 ± 0.008 | 0.834 ± 0.006 |
| crl | 0.825 ± 0.012 | **0.849 ± 0.004** |
| cgl | **0.828 ± 0.009** | 0.845 ± 0.007 |
| rgl | 0.636 ± 0.009 | 0.712 ± 0.005 |
| crgl | 0.825 ± 0.008 | 0.846 ± 0.008 |

**Table 6.17:** Overall accuracy for all classes using different feature sets, as defined in Section 5.2.2, where the bold font is used to indicate the best results for each blemish type.

| Feature subset | Black dot | Silver scurf | Scab | Green | Unblemished |
|---|---|---|---|---|---|
| c | 0.807±0.013 | 0.856±0.008 | 0.965±0.007 | 0.909±0.002 | 0.859±0.006 |
| r | 0.637±0.013 | 0.484±0.033 | 0.935±0.003 | 0.384±0.066 | 0.719±0.009 |
| g | 0.646±0.021 | 0.534±0.023 | 0.933±0.004 | 0.423±0.039 | 0.685±0.011 |
| l | 0.473±0.015 | 0.324±0.029 | 0.824±0.012 | 0.155±0.018 | 0.676±0.013 |
| cr | 0.834±0.002 | 0.858±0.015 | 0.936±0.005 | 0.923±0.002 | **0.874±0.006** |
| cg | 0.843±0.002 | 0.856±0.010 | 0.968±0.002 | **0.932±0.001** | 0.863±0.005 |
| cl | 0.831±0.009 | **0.861±0.008** | 0.964±0.004 | 0.924±0.002 | 0.867±0.004 |
| rg | 0.631±0.021 | 0.546±0.026 | 0.939±0.004 | 0.415±0.037 | 0.704±0.010 |
| rl | 0.617±0.017 | 0.529±0.026 | 0.948±0.004 | 0.329±0.031 | 0.721±0.010 |
| gl | 0.660±0.019 | 0.513±0.019 | 0.949±0.005 | 0.373±0.030 | 0.715±0.009 |
| crg | **0.844±0.011** | 0.847±0.012 | 0.967±0.003 | 0.922±0.018 | 0.869±0.006 |
| crl | 0.839±0.016 | 0.857±0.008 | 0.968±0.002 | 0.930±0.021 | 0.862±0.006 |
| cgl | **0.844±0.015** | 0.858±0.011 | 0.970±0.003 | 0.930±0.016 | 0.864±0.007 |
| rgl | 0.801±0.010 | 0.724±0.008 | 0.951±0.002 | 0.663±0.015 | 0.733±0.006 |
| crgl | 0.838±0.016 | 0.857±0.005 | **0.971±0.002** | 0.927±0.020 | 0.862±0.005 |

**Table 6.18:** Accuracy using different feature sets, as defined in Section 5.2.2, for red potatoes, where the bold font is used to indicate the best results for each blemish type.

| Feature subset | Black dot | Silver scurf | Scab | Green | Unblemished |
|---|---|---|---|---|---|
| c | 0.817±0.004 | 0.841±0.005 | 0.876±0.009 | 0.942±0.009 | 0.882±0.004 |
| r | 0.691±0.017 | 0.701±0.015 | 0.896±0.006 | 0.560±0.060 | 0.801±0.008 |
| g | 0.702±0.010 | 0.688±0.005 | 0.882±0.007 | 0.595±0.070 | 0.823±0.007 |
| l | 0.745±0.005 | 0.547±0.009 | 0.793±0.023 | 0.386±0.040 | 0.729±0.006 |
| cr | 0.863±0.006 | 0.886±0.004 | 0.917±0.006 | 0.945±0.010 | 0.898±0.003 |
| cg | 0.865±0.006 | 0.884±0.006 | 0.914±0.007 | 0.945±0.020 | 0.901±0.003 |
| cl | 0.856±0.004 | 0.840±0.006 | 0.867±0.008 | 0.940±0.011 | 0.890±0.003 |
| rg | 0.698±0.015 | 0.693±0.013 | 0.893±0.007 | 0.518±0.070 | 0.821±0.005 |
| rl | 0.742±0.010 | 0.732±0.013 | 0.873±0.010 | 0.375±0.024 | 0.799±0.005 |
| gl | 0.738±0.015 | 0.726±0.010 | 0.869±0.009 | 0.364±0.019 | 0.812±0.008 |
| crg | 0.868±0.004 | **0.892±0.004** | 0.911±0.005 | 0.943±0.012 | 0.894±0.004 |
| crl | 0.870±0.005 | 0.885±0.006 | **0.922±0.007** | 0.949±0.008 | **0.902±0.002** |
| cgl | 0.870±0.005 | 0.885±0.009 | 0.912±0.007 | **0.950±0.008** | 0.900±0.002 |
| rgl | 0.834±0.004 | 0.836±0.005 | 0.915±0.003 | 0.687±0.015 | 0.825±0.004 |
| crgl | **0.872±0.003** | 0.886±0.007 | 0.916±0.012 | 0.945±0.006 | 0.899±0.002 |

**Table 6.19:** Accuracy using different feature sets, as defined in Section 5.2.2, for white potatoes, where the bold font is used to indicate the best results for each blemish type.

| Human | Marker 1 | Marker 2 | Marker 3 | Panel |
|---|---|---|---|---|
| Accuracy | 86.0% ± 0.5 | 84.3% ± 0.6 | 84.0% ± 0.5 | 85.2% ± 0.3 |
| Average marked pixels | 134502 | 122323 | 115585 | 135989 |
| Pixels in panel markup | 90.8% | 95.6% | 88.1% | 100% |

**Table 6.20:** Accuracy for three different markers and for a “gold standard” mark-up produced from the combination of all three markers, as well as the number of marked pixels per image and the percentage agreement with the gold standard. Marker 1 is the same markup used for the main set of experiments in this thesis.

both list a value. These are lower than in Chapter 5 because there are more classes to disagree on. The highest agreement is between the second potato expert and the gold standard, while the marker with less experience of potatoes has the lowest agreement with the panel.

## 6.3 Conclusions

With five binary AdaBoost classifiers we were able to compare the signed confidences of each classifier to choose the class which a pixel belonged to. By using minimalist AdaBoost to select five features per class then training all five classifiers on the combination of the five subsets, the speed of classification was improved by a factor of 6.5 for white potatoes and 4.1 for red potatoes. This provided a higher accuracy than using minimalist AdaBoost itself to reduce the number of features to the same amount with no loss of speed. For red potatoes using minimalist AdaBoost to extract a larger number of features, ten per class, provided a higher classification rate than using fewer preselected features, for white potatoes the higher classification rate is from the smaller number of preselected features. In both cases, the use of ten features per class was around half as fast as using five per class.

As shown in Figure 6.4, the error rates are still inflated by ambiguity in human markups, although the third example shows a sparse area of black dot being partly missed, probably due to being sparser than other examples, with part of it being misclassified as scab. This is a challenge posed by blemishes with diverse appearances.

The per-potato accuracies listed in Table 6.6 compare favourably with those obtained by Samanta et al. (2012) for scabs as well as Guannan et al. (2009) for green blemishes and Rios-Cabrera et al. (2008) for both scabs and green blemishes. Also it should be noted that the system presented in this thesis not only outperforms these existing approaches, but also offers much greater flexibility because it can be easily retrained to work with many different blemish types.

# 7

# Conclusions and future work

## 7.1 Conclusions

This thesis presents a machine learning approach for blemish detection and identification in potatoes. Potatoes differ greatly, variety and from one season to the next, as well as by factors such as the amount and timing of rainfall changing the development of a tuber in the field, or the length of time since harvesting. These factors can all change the appearance of certain storage blemishes such as black dot as well as the overall texture of good potato skin. This presents a clear motivation for a machine learning based system to detect, measure and identify blemishes in potatoes. Additionally, the trainable nature of the system allows extension into fields such as blemish treatment, where a user might only be interested in differences in the coverage of a single blemish on treated and untreated potatoes. In such an application, the binary classifier could be retrained to only detect that blemish vs. not that blemish.

In order to speed up the classification process, minimalist AdaBoost can provide an alternative way of reducing the number of features used by AdaBoost when using simple weak classifiers such as decision stumps. When feature extraction takes a large portion of the processing time, as in this research, the more traditional method of simply reducing the number of weak classifiers has been shown to require more processing time to produce comparable accuracy. In particular, using minimalist AdaBoost restricted to ten unique features resulted in a mean accuracy within one standard deviation of the mean accuracy of not reducing features at all, with a speed increase of 304% and 151% for white and red potatoes, respectively.

Some disagreements between ground truth and classification results can be seen in Figure 5.2 to be around the edges of blemishes where ground-truthing may be less accurate. This ambiguity as to the exact extent of a blemish between human markup and classifier output is partly symptomatic of the problem this research sets out to solve, that of human inaccuracy in the grading process. Ultimately, a machine learning based system could be developed to combine the best of both worlds, with the computer's consistency in locating blemishes and the human ability to intuitively spot when something is wrong. The results per-potato support this idea since there is a much higher accuracy for identifying the significant blemish areas in individual potatoes compared to identifying each pixel.

Different feature sets work better for different blemishes and also for different potato varieties. For example, scabs are better classified with texture features while greening is better classified using colour.

To the best knowledge of the author, the system presented in this thesis is currently the only available system in the literature capable of learning many different blemish categories from examples.

## 7.2 Future work

The use of square regions is very much as a proof of concept only. Since rotation-invariance is desirable, circular regions might provide an alternative. Some investigation has also been done into superpixels to provide alternative regions. As shown in Figure 7.1, superpixels are a method of dividing an image into small areas of similar pixels, thus providing guidelines for ground truth and natural areas to classify in one go, reducing the problem of ambiguous blemish boundaries and also reducing the processing time involved in extracting features, which could be extracted per-superpixel instead of per-pixel.

Real-time operation is a necessary development in this software if it is to be used in an industrial application. Superpixels can help to streamline the code. It can also be sped up by using a more efficient runtime written in, for example, C++, with much of the repetitive number crunching taking place on a Graphics Processing Unit (GPU, a processor designed to process large numbers of simple mathematical operations in parallel) for further speed improvements.

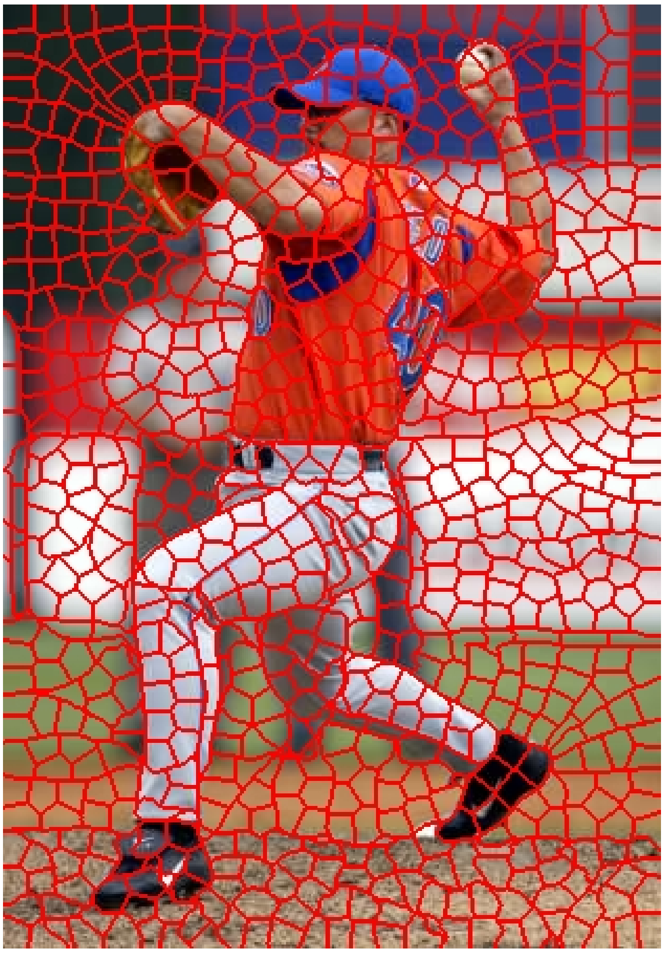


**Figure 7.1:** An image of a baseball player divided into superpixels, from Mori (2005). Every superpixel can be seen to only contain a single type of content, be that content leg, arm, hat, shoe, grass etc.

While the minimalist AdaBoost approach demonstrably provides faster processing with a lower cost to accuracy than a non-minimalist approach, there could be room for development of an algorithm which will prefer features that have a lower overhead, or consider prerequisites. For instance, to obtain the skewedness of the gradient of the normalised green colour channel in a $66 \times 66$ region requires calculating the following prerequisites:

1. the normalised green colour channel.

2. the gradient of the normalised green colour channel.

3. the mean of feature 2 in a 66 × 66 region.

4. the variance of feature 2 in a 66 × 66 region.

Any of these can then provide additional features, which require no extra processing time to extract.

The research presented in this thesis utilised the RGB colour space almost exclusively, while using normalised RGB and the intensity channel to allow the system to judge chromaticity and intensity features independently. Other colour spaces, such as HSI or YCrCb (Gonzalez et al., 2004) ought to be tested at a later date. In addition, other possible feature sets might include the use of shape features, especially if superpixels are to be used to locate areas of blemish.

For this research, a pixelwise approach was taken in order to identify the amount of a potato which is covered with blemish. No attempt is currently made to translate pixels into surface area, which can be approached in a variety of ways. One approach which is used in industry is to take multiple images of a single potato, possibly omitting areas which are close to the potato borders, then average the series of classified images to gain an overview of the entire potato (R.J. Herbert Engineering Ltd, 2008). Alternative approaches include using 3D sensors, such as photonic mixer devices, or imperceptible structured light to reconstruct the 3D surface of the potato.

These experiments were run separately on both red and white potato varieties. At a later stage it would be possible to run them on other varieties or on multicoloured potatoes such as King Edward, which is both red and white, or Catriona, which is a white and blue variety.

In developing this system, only the most common blemishes in our feature sets were used. This was a deliberate choice since it was not plausible to include a blemish such as elephant hide, of which we had three examples in the entire white data set and none in the red data set. Since it is impractical to produce an exhaustive sample set of all possible potato blemishes, some decision will need to be made about how to handle unknown blemishes.

There would also need to be longer-term tests, probably in an industrial setting, to assess the performance of the system over a whole harvest if not multiple harvests. Questions to explore will include how quickly potato features change from harvest to harvest and how frequently it is necessary to retrain the system.

Finally, a complete product based around this research would require an accessible interface allowing a semi-skilled user to quickly and easily mark up new training data and modify classification parameters as well as monitoring the classification process.

# References

Adobe. *Using Adobe Photoshop CS4*. Adobe, 10 2010.

A Al-Mallahi, T Kataoka, and H Okamoto. Discrimination between potato tubers and clods by detecting the significant wavebands. *Biosystems engineering*, 100(3): 329–337, 2008.

R. F. Alexander, G. B. Forbes, and E. S. Hawkins. A fatal case of solanine poisoning. *British Medical Journal*, 4575:518, 1948.

Erin L. Allwein, Robert E. Schapire, Yoram Singer, and Pack Kaelbling. Reducing multiclass to binary: A unifying approach for margin classifiers. 2000.

Diwan P Ariana and Renfu Lu. Hyperspectral waveband selection for internal defect detection of pickling cucumbers and whole pickles. *Computers and electronics in agriculture*, 74(1):137–144, 2010.

Michael Barnes, Tom Duckett, Grzegorz Cielniak, Graeme Stroud, and Glyn Harper. Visual detection of blemishes in potatoes using minimalist boosted classifiers. *Journal of Food Engineering*, 98(3):339 – 346, 2010. ISSN 0260-8774. doi: DOI:10.1016/j.jfoodeng.2010.01.010. URL http://www.sciencedirect.com/science/article/B6T8J-4Y646GH-1/2/d2068debd28b75ee4b75ccd1dfb59d3b.

Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

B. Bizimungu, L.M. Kawchuk, J. Wahab, D. Waterer, and R.J. Howard. Sentinel: An early yellow-fleshed potato cultivar suitable for fresh market and chip processing. *American Journal of Potato Research*, pages 1–9, 2013.

J Blasco, S Cubero, S Alegre-Sosa, J Gómez-Sanchís, and E Moltó. Short communication. automatic inspection of the pomegranate (punica granatum l.) arils quality by means of computer vision. *Spanish Journal of Agricultural Research*, 6(1):12–16, 2008.

R.M. Bolle, J.H. Connell, N. Haas, R. Mohan, and G. Taubin. Veggievision: A produce recognition system. In *Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision (WACV '96)*, page 244, Washington, DC, USA, 1996. IEEE Computer Society. ISBN 0-8186-7620-5.

Peter Bowbrick. The economics of grades. *Oxford Development Studies*, 11(1):65–92, 1982.

Tadhg Brosnan and Da-Wen Sun. Improving quality inspection of food products by computer vision—-a review. *Journal of Food Engineering*, 61(1):3–16, 2004.

M.C. Clark, L.O. Hall, D.B. Goldgof, R Velthuizen, F.R. Murtagh, and M.S. Silbiger. Automatic tumor segmentation using knowledge-based techniques. *Medical Imaging*, 17:187–201, 1998.

Angel Dacal-Nieto, Arno Formella, Pilar Carrión, Esteban Vazquez-Fernandez, and Manuel Fernández-Delgado. Common scab detection on potatoes using an infrared hyperspectral imaging system. In *Image Analysis and Processing–ICIAP 2011*, pages 303–312. Springer, 2011.

Eisse G. De Haan and Ge W van den Bovenkamp. Improved diagnosis of powdery scab (spongospora subterranea f.sp. subterranea) symptoms on potato tubers (solanum tuberosum l.). *Potato Research*, 48(1-2):1–14, 2005.

Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. New York: Springer, 1996.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* 2nd edition edition, 2000.

C. D. Everard, D. J. OCallaghan, C. C. Fagan, C. P. ODonnell, M. Castillo, and F. A. Payne. Computer vision and color measurement techniques for inline monitoring of cheese curd syneresis. *Journal of Dairy Science*, 90:3162–3170, 2007.

Tom Fawcett. Roc graphs : Notes and practical considerations for researchers. *ReCALL*, 31(HPL-2003-4):1–38, 2004. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.9777&rep=rep1&type=pdf`.

Marie Fiers, Catherine Chatot, Vronique Edel-Hermann, Yves Hingrat, AbelYanougo Konate, Nadine Gautheron, Emmanuel Guillery, Claude Alabouvette, and Christian Steinberg. Diversity of microorganisms associated with atypical superficial blemishes of potato tubers and pathogenicity assessment. *European Journal of Plant Pathology*, 128(3):353–371, 2010. ISSN 0929-1873. doi: 10.1007/s10658-010-9657-2. URL `http://dx.doi.org/10.1007/s10658-010-9657-2`.

Lynn M. Fletcher-Heath, Lawrence O. Hall, Dmitry B. Goldgof, and F. Reed Murtagh. Automatic segmentation of non-enhancing brain tumors in magnetic resonance images. *Artificial Intelligence in Medicine*, 21:43–63, 2001.

Food and Agriculture Organisation. FAO Statistics. Online, 2011. (http://faostat.fao.org).

D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

Y. Freund and R.E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14 (5):771 – 780, September 1999.

Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. 1996.

Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. *Digital Image Processing Using MATLAB*. Prentice Hall, 2004.

Alistair Gorman, David William Fletcher-Holmes, and Andrew Robert Harvey. Generalization of the lyot filter and its application to snapshot spectral imaging. *Optics Express*, 8(6):5602–5608, 2010.

Zheng Guannan, Tan Yuzhi, Zhang Junxiong, and Le Wei. Automatic detecting and grading method of potatoes with computer vision. *Nongye Jixie Xuebao / Transactions of the Chinese Society of Agricultural Machinery*, 40(4):166–168+1, 2009.

Long Guo, Zhang ShaoRong, Wang ChaoHai, Cao Xi, Chai XiangXi, Nie ZongPing, Nie ShaoKe, et al. Breeding of bishu 3, a new potato variety. *Guizhou Agricultural Sciences*, 10:8–10, 2009.

Ricardo Gutierrez-Osuna and Andreas Hierlemann. Adaptive microsensor systems. *Analytical Chemistry*, 3:255–276, 2010.

A. Hafiane, F. Bunyak, and K. Palaniappan. Clustering initiated multiphase active contours and robust separation of nuclei groups for tissue segmentation. *19th International Conference on Pattern Recognition*, pages 1–4, 2008.

Roya Hasankhani and Hosein Navid. Potato sorting based on size and color in machine vision system. *Journal of Agricultural Science*, 4(5):p235, 2012.

S. Haykin. *Neural Networks - A Comprehensive Foundation.* Pearson Education, NY, 1998.

Tobias Heimann and Hans-Peter Meinzer. Statistical shape models for 3d medical image segmentation: A review. *Medical image analysis*, 13(4):543–563, 2009.

Peter Henrici. *Applied and Computational Complex Analysis, Discrete Fourier Analysis, Cauchy Integrals, Construction of Conformal Maps, Univalent Functions.* Wiley, 1993.

N.J. Hepworth, J.R.M. Hammond, and J. Varley. Novel application of computer vision to determine bubble size distributions in beer. *Journal of Food Engineering*, 61: 119–124, 2004.

IBM. Image information system group technical agenda. Online, 2012. URL `http://www.research.ibm.com/networked_data_systems/9auc/research.html`. retrieved 2012-01-18.

Patrick Jackman, Da-Wen Sun, Paul Allen, Karen Brandon, and Anna-Marie White. Correlation of consumer assessment of longissimus dorsi beef palatability with image colour, marbling and surface texture features. *Meat Science*, 84:564–568, 2010.

Matthew Johnson and Jamie Shotton. Semantic texton forests. In *Computer Vision*. Springer Berlin / Heidelberg, 2010.

R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and Zhang Jing. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31: 319–336, 2009.

Kivanc Kilic, Ismail Hakki Boyaci, Hamit Koksel, and Ismail Kusmenoglu. A classification system for beans using computer vision system and artificial neural networks. *Journal of Food Engineering*, 78:897–904, 2007.

Stephen M Kosslyn and James M Intriligator. Is cognitive neuropsychology plausible? the perils of sitting on a one-legged stool. *Journal of Cognitive Neuroscience*, 4(1): 96–105, 1992.

Akhlesh Lakhtakia. Would brewster recognize today's brewster angle? *Optics News*, 15(6):14–18, 1989.

Gwen Littlewort, Marian Stewart Bartlett, Ian Fasel, Joshua Susskind, and Javier Movellan. Dynamics of facial expression extracted automatically from video. *Image and Vision Computing*, 24(6):615 – 625, 2006. ISSN 0262-8856. doi: 10.1016/ j.imavis.2005.09.011. URL `http://www.sciencedirect.com/science/article/pii/S0262885605001654`. ¡ce:title¿Face Processing in Video Sequences¡/ce:title¿.

Maf Roda Group. Meeting with Maf-Roda personnel, October 2008. (http://www.maf-roda.com/).

S.K. Mathanker, P.R. Weckler, T.J. Bowser, N. Wang, and N.O. Maness. Adaboost classifiers for pecan defect classification. *Computers and Electronics in Agriculture*, In Press, Corrected Proof, 2011. doi: DOI:10.1016/j.compag.2011.03.008.

F. Mendoza and J.M. Aguilera. Application of image analysis for classification of ripening bananas. *Journal of Food Science*, 9, 2004.

E. Mismi, J.R. Mathiassen, and U. Erikson. Computer vision-based sorting of atlantic salmon (salmo salar) fillets according to their color level. *Journal of Food Science*, 72:31–35, 2007.

Greg Mori. Guiding model search using segmentation. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1417 – 1423, 2005.

Andrew J Muir, David W Ross, Calum J Dewar, and Duncan Kennedy. Defect and disease detection in potato tubers. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 3543, pages 199–207, 1999.

P. Munkevik, G. Hall, and T. Duckett. A computer vision system for appearance-based descriptive sensory evaluation of meals. *Journal of Food Engineering*, 78:246 – 256, 2007.

V.G̃. Narendra and K.S̃. Hareesh. Quality inspection and grading of agricultural and food products by computer vision- a review. *International Journal of Computer Applications*, 2(1):43–65, 2010.

J.C. Noordam, G.W. Otten, T.J. Timmermans, and B.H. van Zwol. High-speed potato grading and quality inspection based on a color vision system. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3966 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 206–217, March 2000.

F. Pedreschi, D. Mery, F. Mendoza, and J.M. Aguilera. Classification of potato chips using pattern recognition. *Journal of Food Science*, 69(6), 2004.

Pavel Pudil, Jana Novovičová, and Josef Kittler. Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125, 1994.

T. Pun, M. Lefebvre, S. Gil, D. Brunet, J. Dessimoz, and P. Gugerli. The potato operation: Computer vision for agricultural robotics. In *Proc. SPIE Conf. on Advances in Intelligent Robotic Systems*, November 1991.

Roberto Quevedo, Oscar Diaz, Arnaldo Caqueo, Betty Ronceros, and J.M. Aguilera. Quantification of enzymatic browning kinetics in pear slices using non-homogenous l* color information from digital images. *Food Science and Technology*, 42:1367–1373, 2009.

R. Rios-Cabrera, I Lopez-Juarez, and Hsieh Sheng-Jen. Ann analysis in a vision approach for potato inspection. *Journal of Applied Research and Technology*, 6(2): 106–119, 2008.

R.J. Herbert Engineering Ltd. Meeting with representatives of R.J. Herbert Engineering Ltd., Marshland St. James, Cambridgeshire, UK, October 2008. (http://www.rjherbert.co.uk/).

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

Debabrata Samanta, Prajna Paramita Chaudhury, and Arya Ghosh. Scab diseases detection of potato using image processing. *International Journal of Computer Trends and Technology*, 3:109–113, 2012.

Dayanand G. Savakar and Basavaraj S. Anamiy. Recognition and classification of food grains, fruits and flowers using machine vision. *International Journal of Food Engineering*, 5, 2009.

R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 80–91, 1998.

R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, December 1999.

Mojtaba Seyedhosseini, Antonio R. C. Paiva, and Tolga Tasdizen. Fast adaboost training using weighted novelty selection. In *International Joint Conference on Neural Networks*, pages 1245–1250, August 2011.

Stephen V. Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1):77 – 89, 1997.

Christos Stergiou and Dimitrios Siganos. Neural networks. online, 2012. URL `http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html`. retrieved 2012-01-18.

Y Tao, P H Heinemann, and Z Varghese. Machine vision for color inspection of potatoes and apples. *Transactions of the American Society of Agricultural Engineers*, 38(5): 1555–1561, 1995.

The Mathworks Inc. Image Processing Toolbox - MATLAB, 2009a. URL `http://www.mathworks.com/products/image/`.

The Mathworks Inc. MATLAB, 2009b. URL `http://www.mathworks.com/`.

D. Tsai and Y. Tsai. Rotation-invariant pattern matching with colour-ring projection. *Pattern Recognition*, 35:131–141, 2002.

L. Tsror, O. Erlich, M. Hazanovsky, B. Ben Daniel, U. Zig, and S. Lebiush. Detection of dickeya spp. latent infection in potato seed tubers using pcr or elisa and correlation with disease incidence in commercial field crops under hot-climate conditions. *Plant Pathology*, 61(1):161–168, 2012. ISSN 1365-3059.

D. Unay and B. Gosselin. Stem and calyx recognition on 'jonagold' apples by pattern recognition. *Journal of Food Engineering*, 78:597 – 605, 2006.

P. Uthaisombut. Detecting defects in cherries using machine vision. Master's thesis, Michigan State University, 1995.

Chenglong Wang, Xiaoyu Li, Wei Wang, Liu Jie, Hailong Tao, and Dongdong Wen. Detection of potatos size based on centroidal principal axis. *African Journal of Agricultural Research*, 6(17):4140–4148, 2011.

Hai-Hong Wang and Da-Wen Sun. Melting characteristics of cheese: analysis of effects of cooking conditions using computer vision technology. *Jounal of Food Engineering*, 51:305–310, 2002.

P. Wijethunga, S. Samarasinghe, and D. Kulasiri. Towards a generalized colour image segmentation for kiwifruit detection. In *24th International Conference Image and Vision Computing New Zealand*, 2009.

Huan-zhang Wu, Zhao-juan Guo, and Huan-li Chen. A new potato varietyzhengshu no. 9. *China Vegetables*, 22:020, 2010.

Chaoxin Zheng, Da-Wen Sun, and Cheng-Jin Du. Estimating shrinkage of large cooked beef joints during air-blast cooling by computer vision. *Journal of Food Engineering*, 72:56–62, 2006.

L Zhou, V Chalana, and Y Kim. Pc-based machine vision system for real-time computer-aided potato inspection. *International Journal of Imaging Systems and Technology*, 9(6):423–433, 1998.

Ji Zhu, Hui Zuo, Saharon Rosset, and Trevor Hastie. Multi-class adaboost. *Statistics and Its Inheritance*, 2:349–360, 2009.

## Declaration

I herewith declare that I have produced this paper without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This paper has not previously been presented in identical or similar form to any other British or foreign examination board.

The thesis work was conducted from 2007 to 2010 under the supervision of Tom Duckett and Grzegorz Cielniak at the University of Lincoln, Lincoln, UK.

Lincoln, UK

# Appendix A

# Further representation of results

## A.1 Blemish detection

| Feature limit | Non minimalist |  | Minimalist |  |
|---|---|---|---|---|
|  | White | Red | White | Red |
| 1 | 0.824 ± 0.004 | 0.763 ± 0.007 | 0.824 ± 0.004 | 0.763 ± 0.008 |
| 2 | 0.824 ± 0.004 | 0.843 ± 0.006 | 0.882 ± 0.002 | 0.760 ± 0.009 |
| 5 | 0.865 ± 0.011 | 0.793 ± 0.006 | 0.901 ± 0.001 | 0.814 ± 0.004 |
| 10 | 0.890 ± 0.005 | 0.840 ± 0.006 | 0.911 ± 0.003 | 0.858 ± 0.006 |
| 40 | 0.914 ± 0.006 | 0.917 ± 0.010 | 0.914 ± 0.006 | 0.917 ± 0.010 |

**Table A.1: Specificity** for blemish detection in red and white potatoes, used to produce the ROC curves in Figure 5.1.

| Feature | Non minimalist | | Minimalist | |
|---|---|---|---|---|
| limit | White | Red | White | Red |
| 1 | 0.790 ± 0.008 | 0.790 ± 0.008 | 0.790 ± 0.008 | 0.790 ± 0.008 |
| 2 | 0.790 ± 0.008 | 0.728 ± 0.009 | 0.856 ± 0.005 | 0.844 ± 0.006 |
| 5 | 0.881 ± 0.004 | 0.850 ± 0.006 | 0.901 ± 0.003 | 0.903 ± 0.004 |
| 10 | 0.901 ± 0.005 | 0.876 ± 0.004 | 0.913 ± 0.003 | 0.914 ± 0.003 |
| 40 | 0.911 ± 0.003 | 0.862 ± 0.004 | 0.911 ± 0.003 | 0.862 ± 0.004 |

**Table A.2: Sensitivity** for blemish detection in potatoes, used to produce the ROC curves in Figure 5.1.

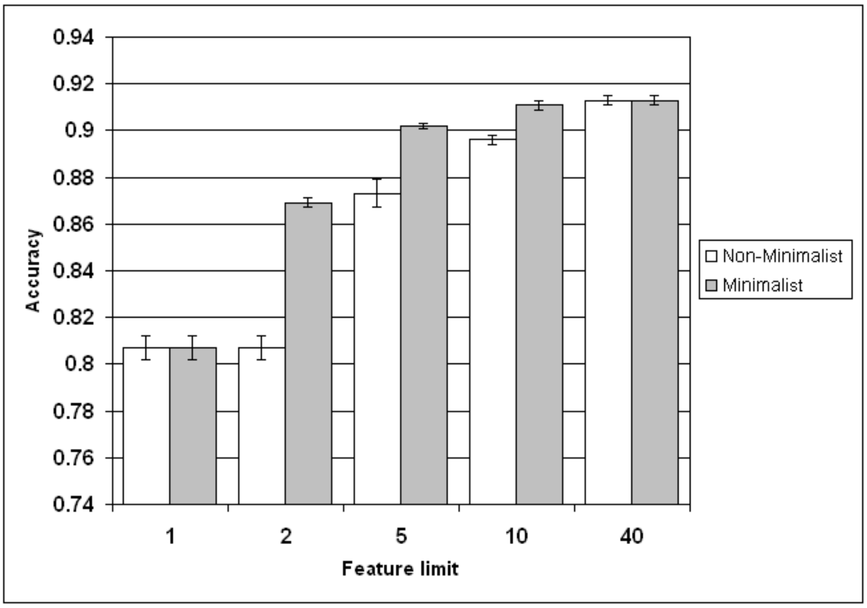


**Figure A.1:** A bar chart describing the accuracy of blemish detection in white potatoes, as described in Table 5.1.

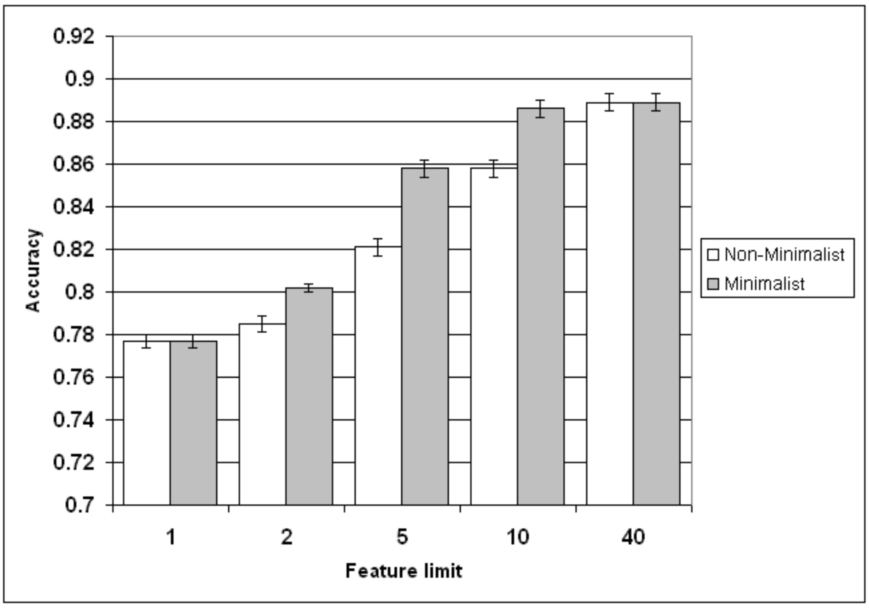


**Figure A.2:** A bar chart describing the accuracy of blemish detection in red potatoes, as described in Table 5.1.

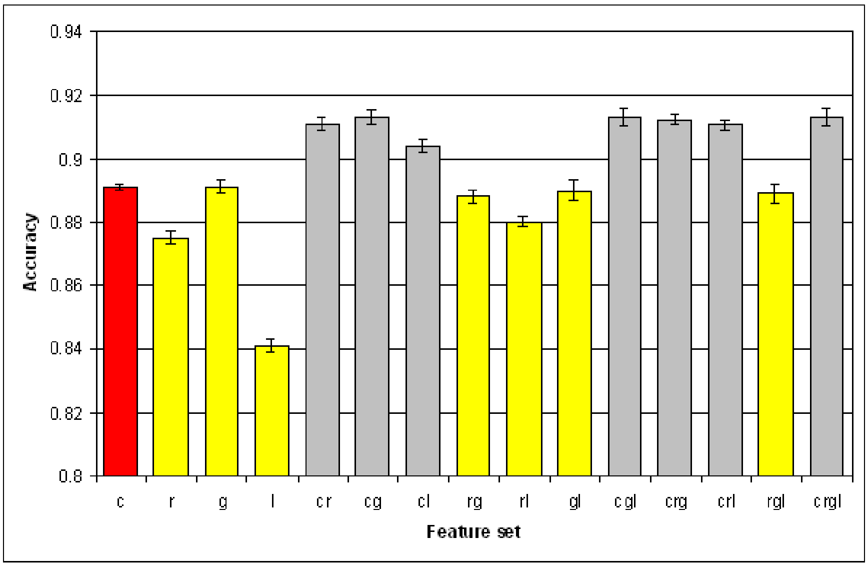


**Figure A.3:** A bar chart describing the accuracy of blemish detection in white potatoes using restricted feature sets, as described in Table 5.4. Colours indicate sets involving only colour (red), only texture (yellow) and both (grey)

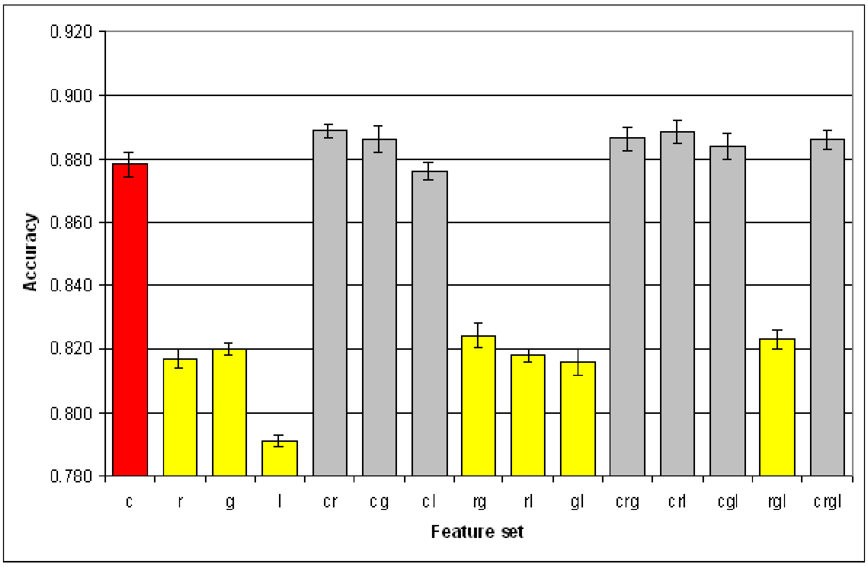


**Figure A.4:** A bar chart describing the accuracy of blemish detection in red potatoes using restricted feature sets, as described in Table 5.4. Colours indicate sets involving only colour (red), only texture (yellow) and both (grey)

## A.2 Blemish idenfication

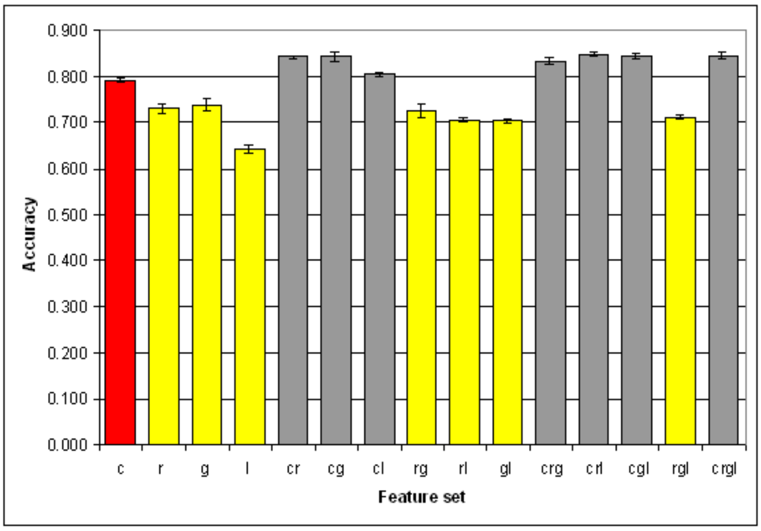


**Figure A.5: Overall** accuracy rates for correct identification of blemish types in **white potatoes** using AdaBoost trained with preselected features from different feature sets.

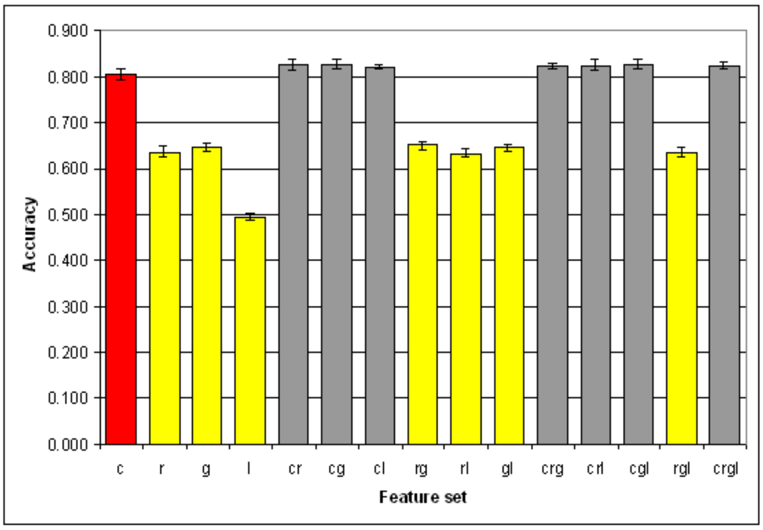


**Figure A.6: Overall** accuracy rates for correct identification of blemish types in **red potatoes** using AdaBoost trained with preselected features from different feature sets.

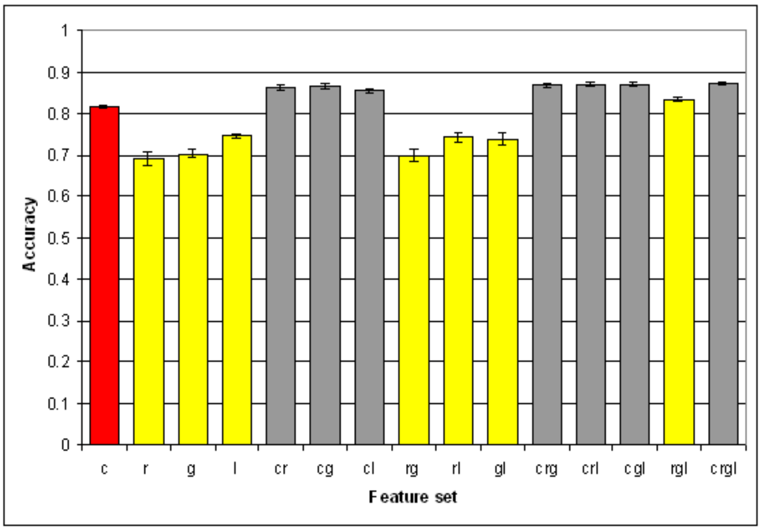


**Figure A.7:** Accuracy rates for detection of **black dot** in **white potatoes** using AdaBoost trained with preselected features from different feature sets.

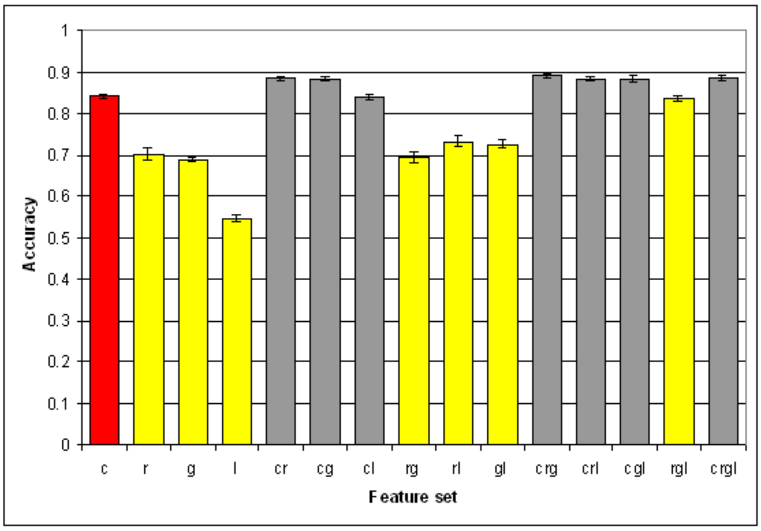


**Figure A.8:** Accuracy rates for detection of **silver scurf** in **white potatoes** using AdaBoost trained with preselected features from different feature sets.

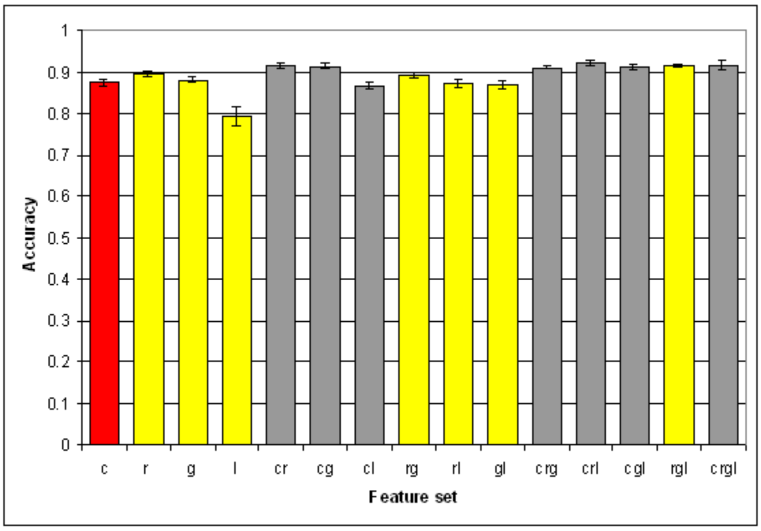


**Figure A.9:** Accuracy rates for detection of **scabs** in **white potatoes** using AdaBoost trained with preselected features from different feature sets.

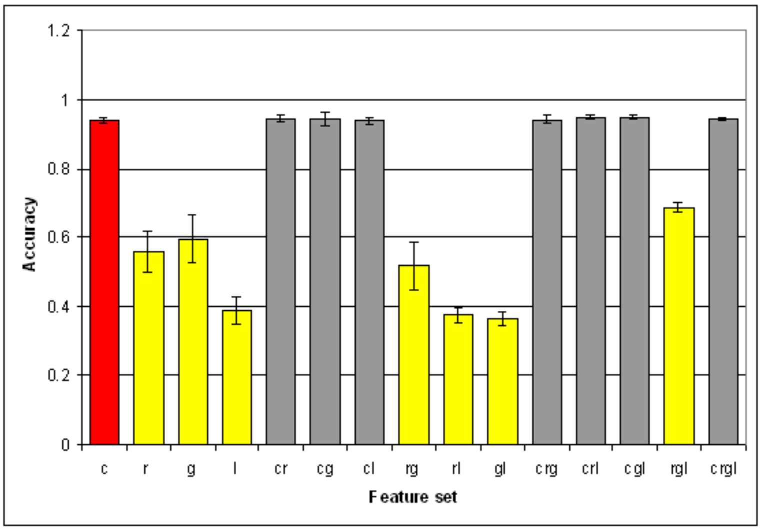


**Figure A.10:** Accuracy rates for detection of **green blemishes** in **white potatoes** using AdaBoost trained with preselected features from different feature sets.

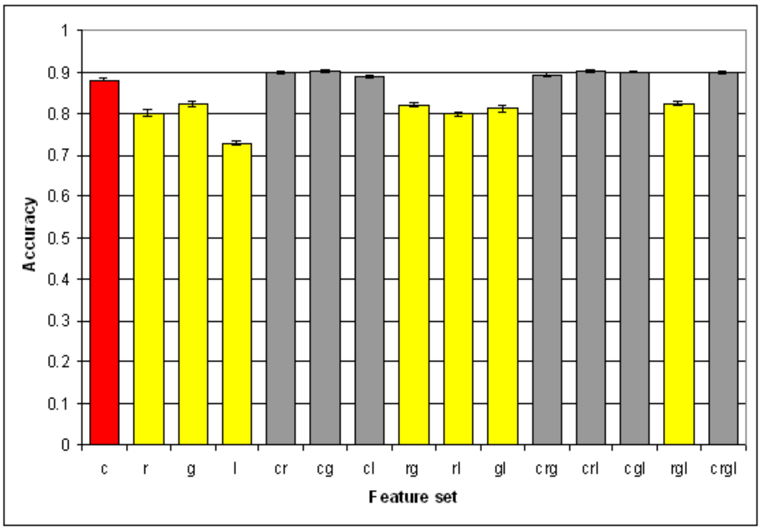


**Figure A.11:** Accuracy rates for detection of **good potato skin** in **white potatoes** using AdaBoost trained with preselected features from different feature sets.

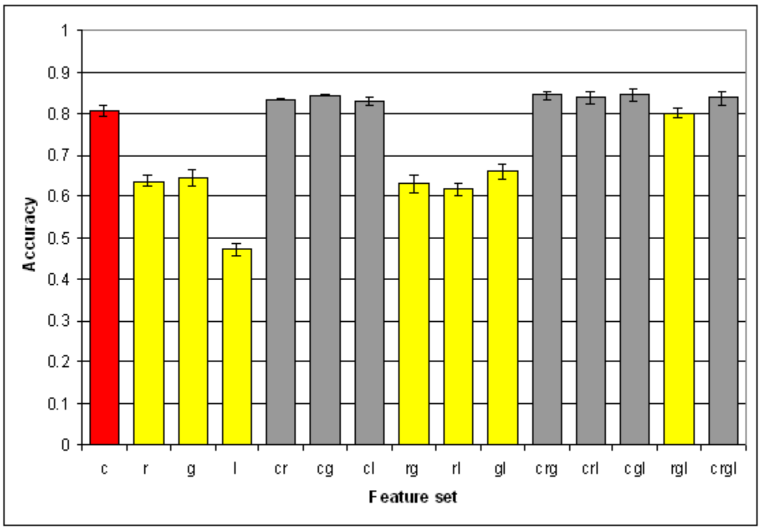


**Figure A.12:** Accuracy rates for detection of **black dot** in **red potatoes** using AdaBoost trained with preselected features from different feature sets.

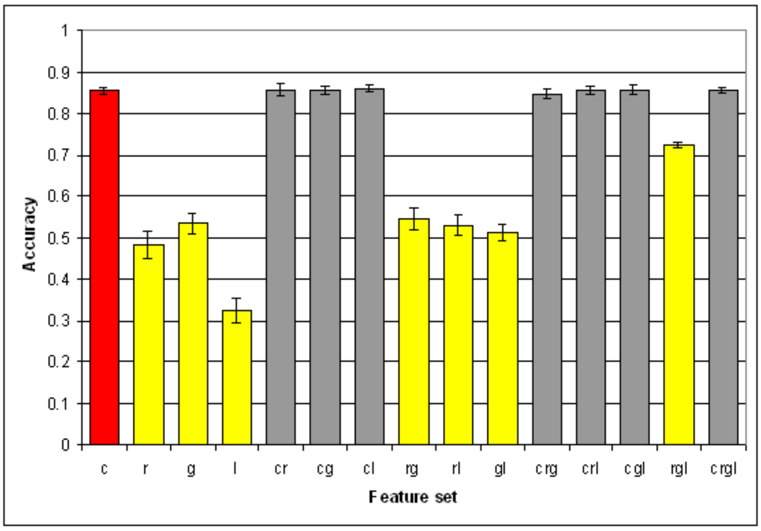


**Figure A.13:** Accuracy rates for detection of **silver scurf** in **red potatoes** using AdaBoost trained with preselected features from different feature sets.

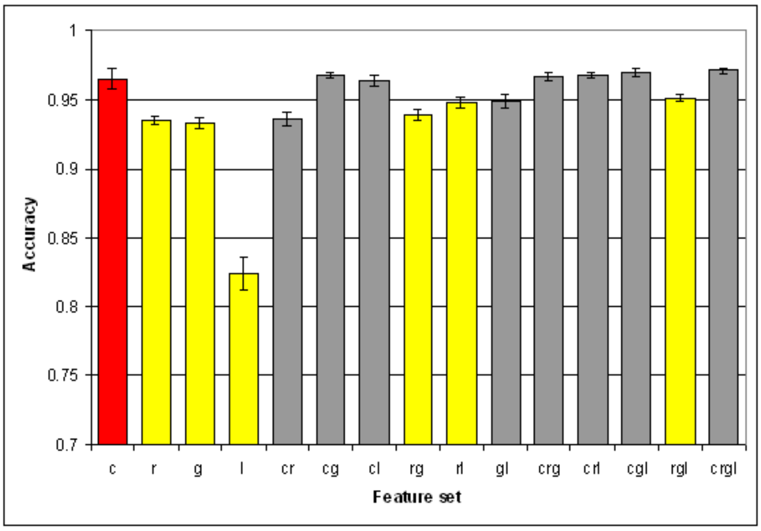


**Figure A.14:** Accuracy rates for detection of **scabs** in **red potatoes** using AdaBoost trained with preselected features from different feature sets.

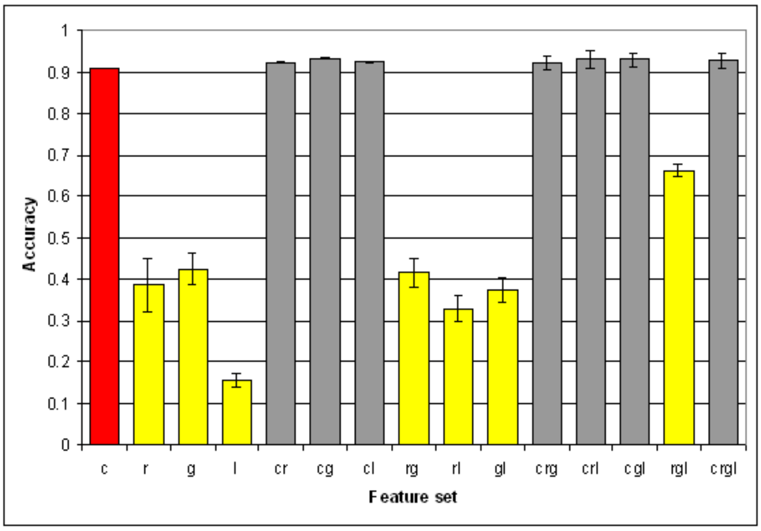


**Figure A.15:** Accuracy rates for detection of **green blemishes** in **red potatoes** using AdaBoost trained with preselected features from different feature sets.

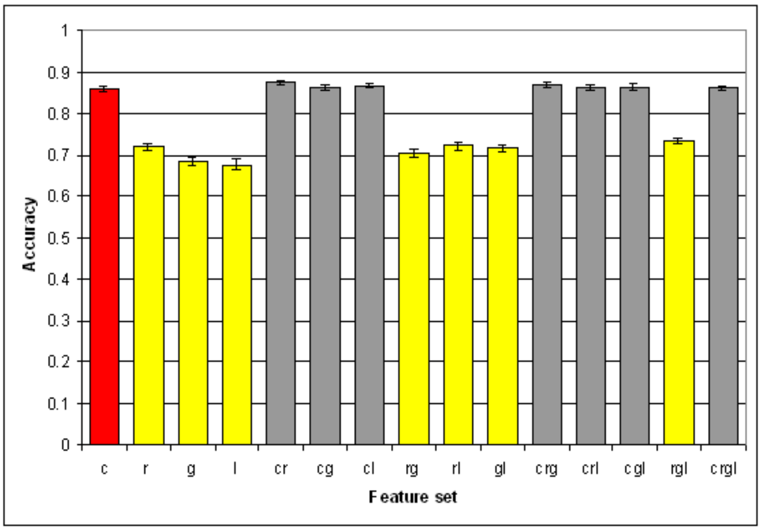


**Figure A.16:** Accuracy rates for detection of **good potato skin** in **red potatoes** using AdaBoost trained with preselected features from different feature sets.

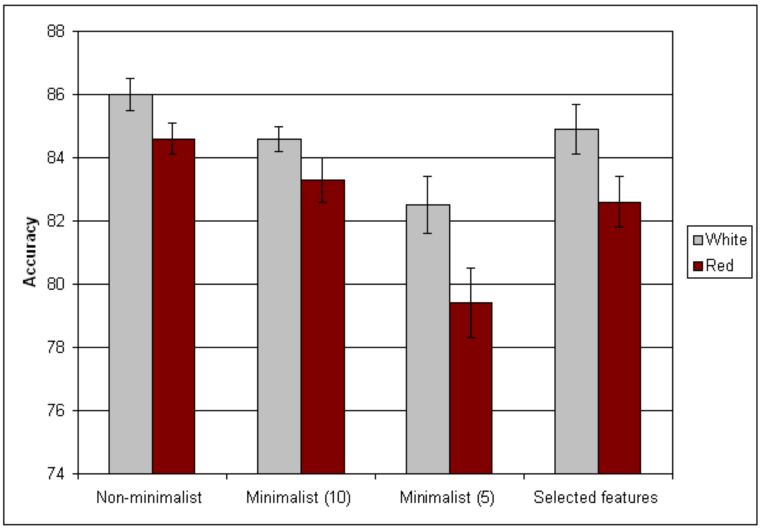
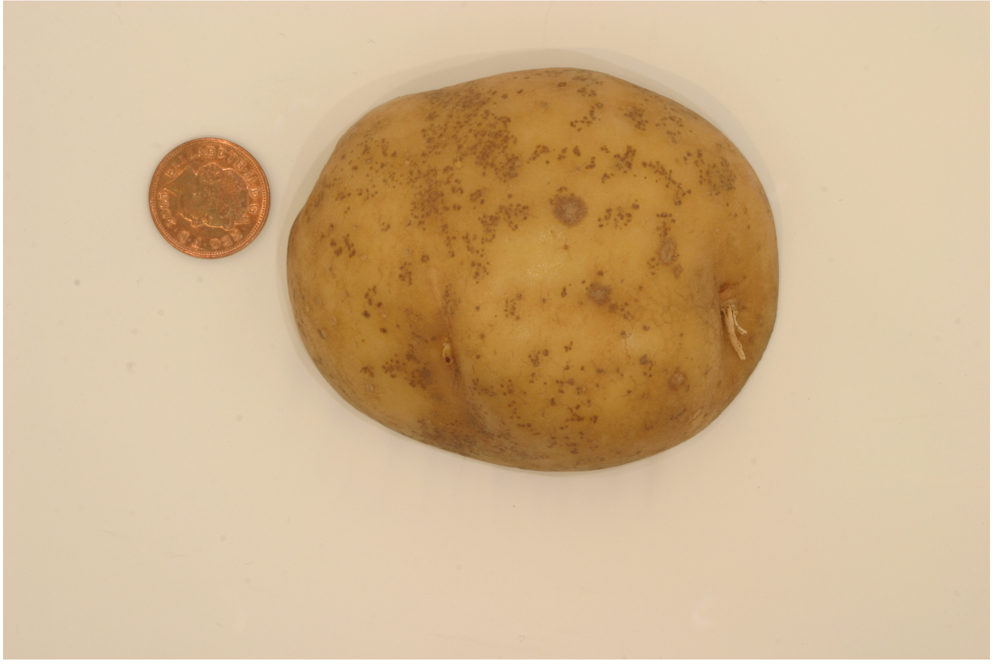


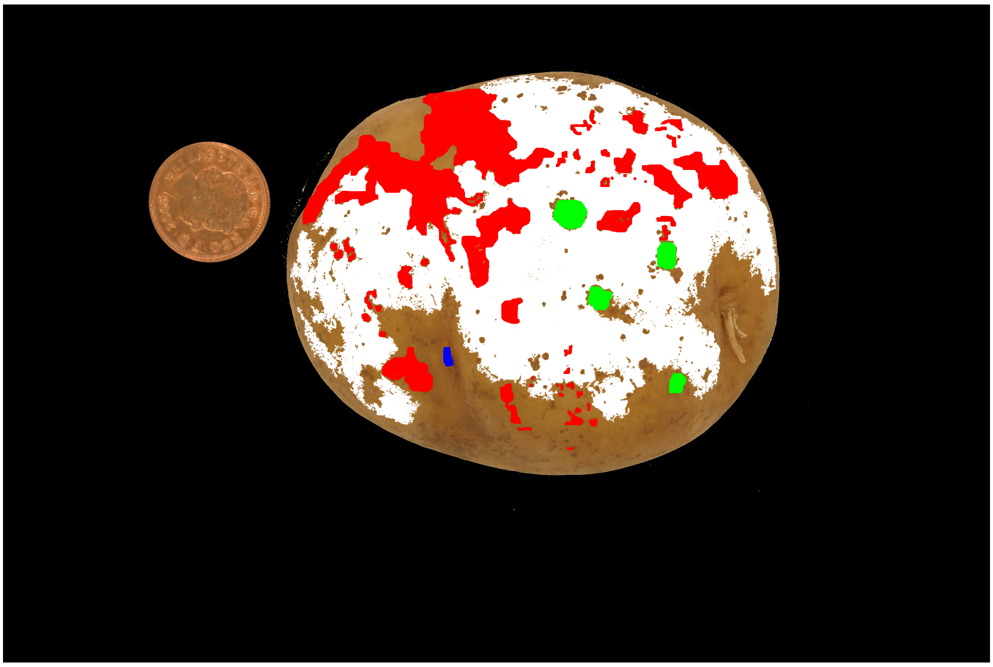
**Figure A.17:** Comparison of results for different types of minimalist and non-minimalist classification using red and white potatoes

# Appendix B

# Example images from data sets

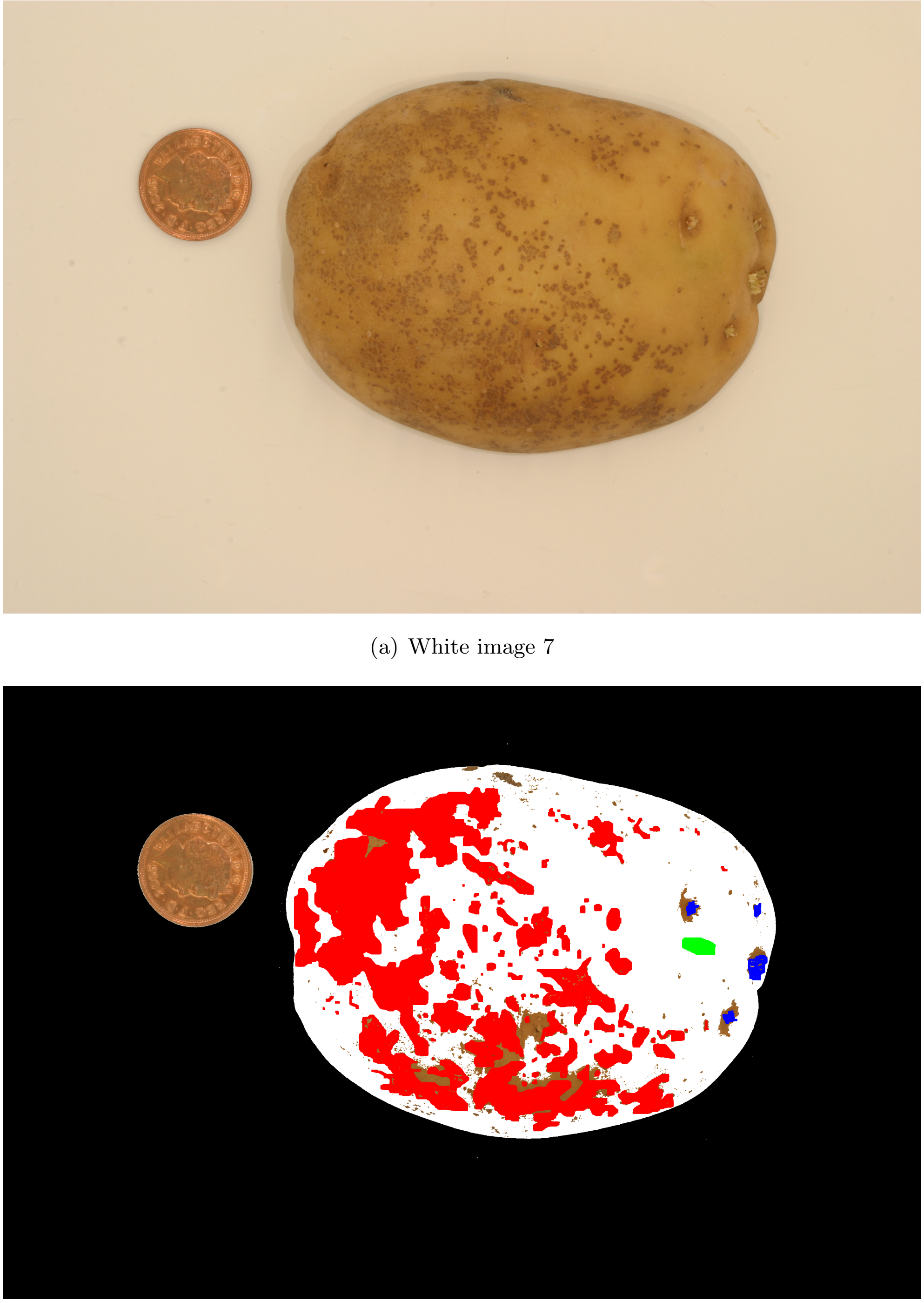


(a) White image 4



(b) Ground truth of White image 4

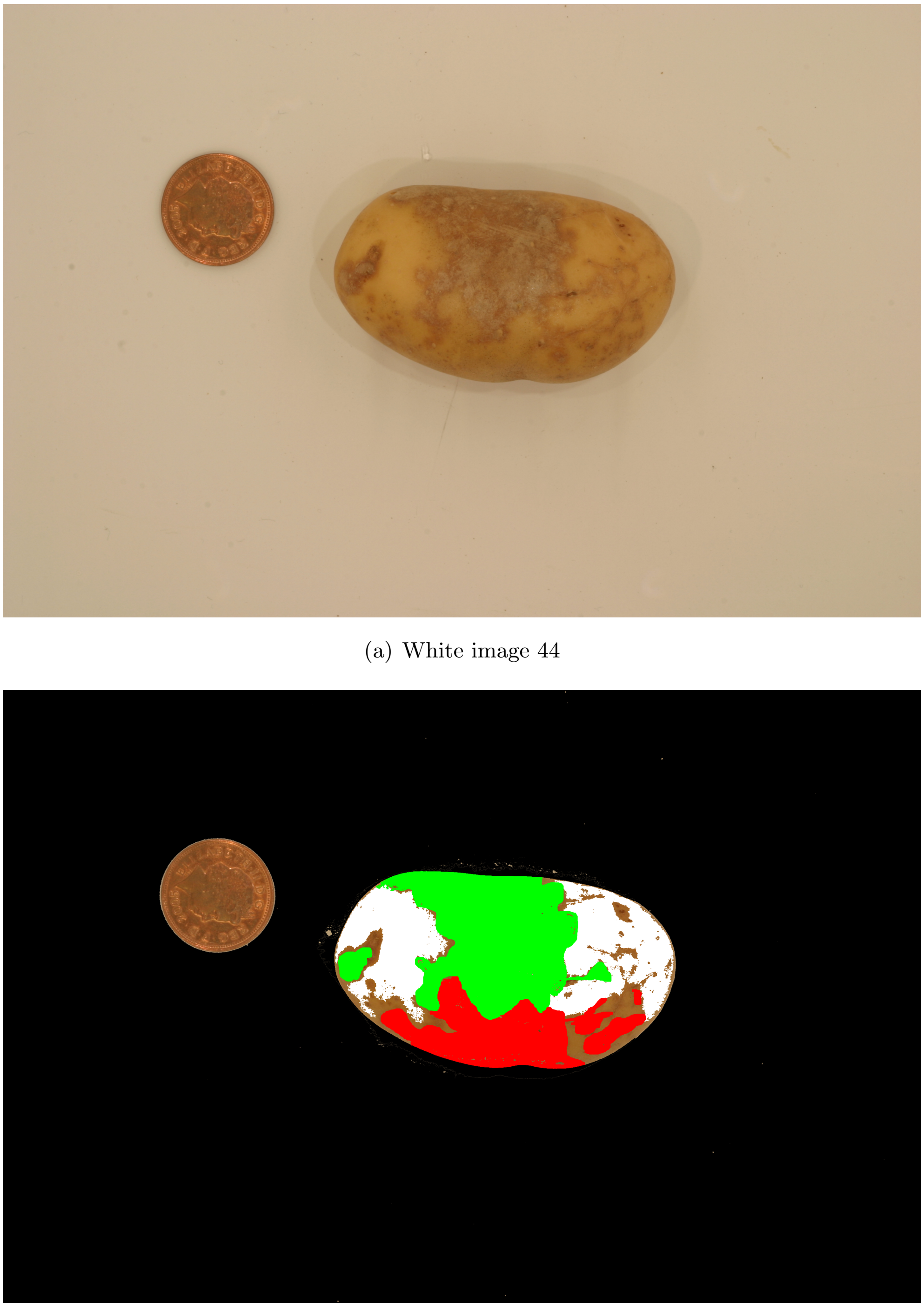
**Figure B.1:** Image 4 from the white data set. Red = Black dot, Green = Silver scurf, Blue = sprouting



(a) White image 7

(b) Ground truth of White image 7

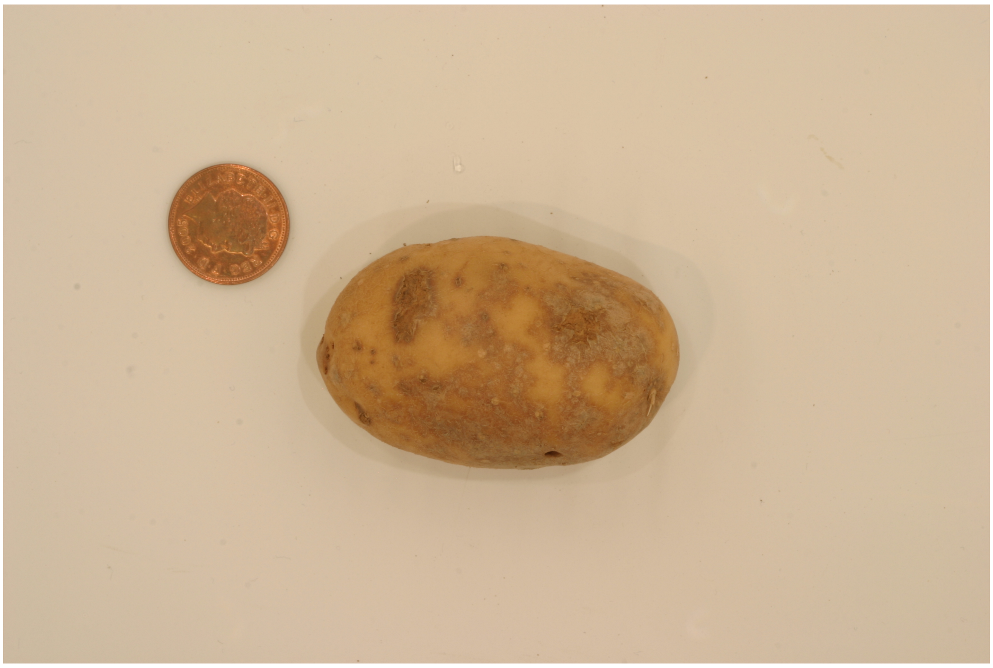
**Figure B.2:** Image 7 from the white data set. Red = Black dot, Green = greening, Blue = sprouting



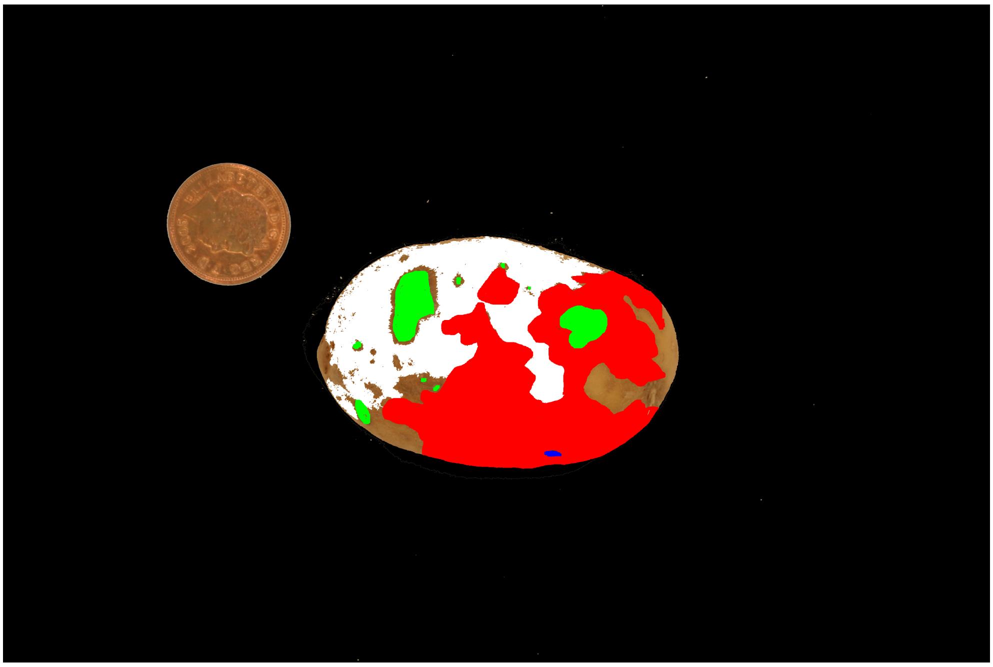
(a) White image 44

(b) Ground truth of White image 44

**Figure B.3:** Image 44 from the white data set. Red = Black dot, Green = silver scurf

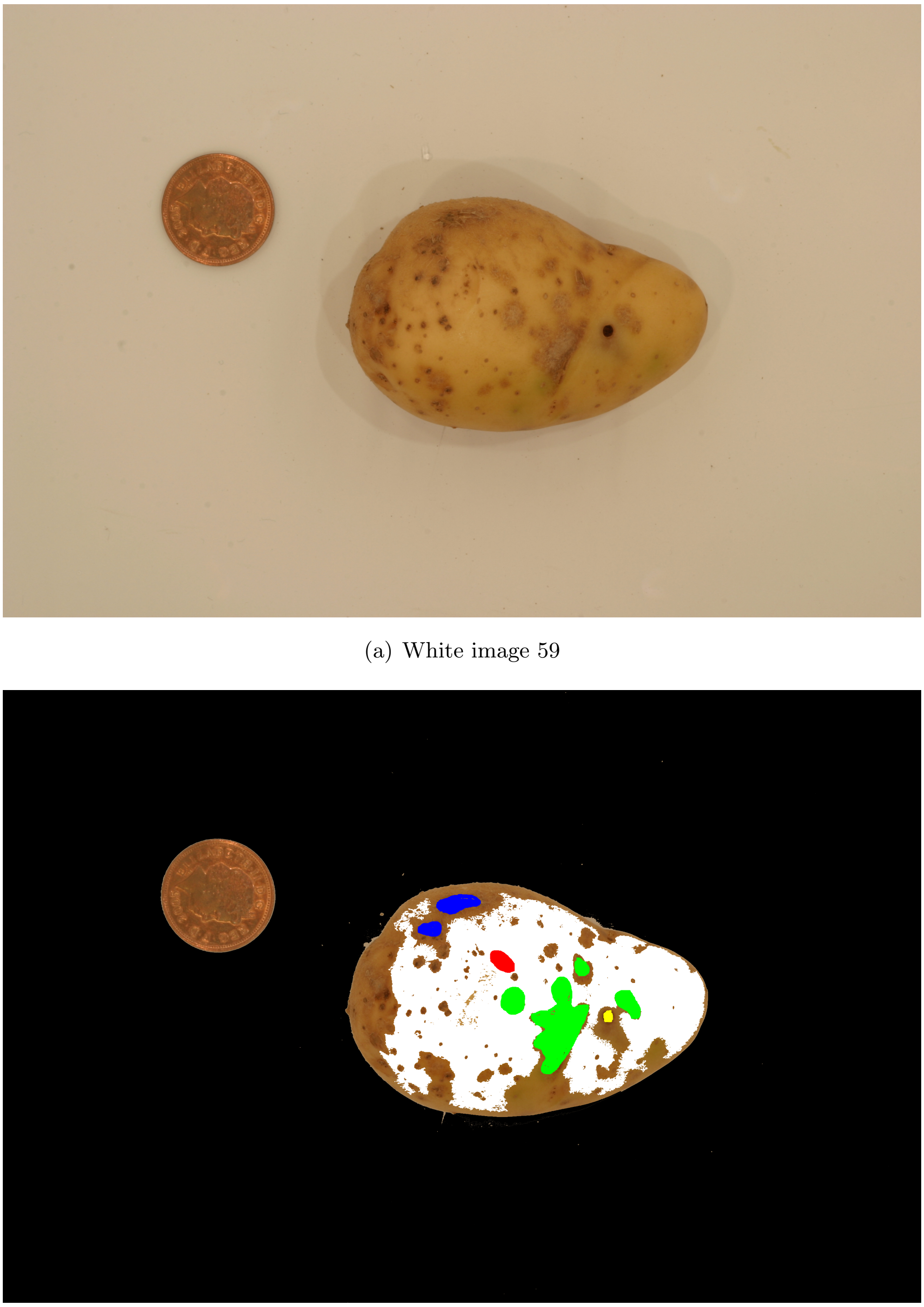


(a) White image 47



(b) Ground truth of White image 47

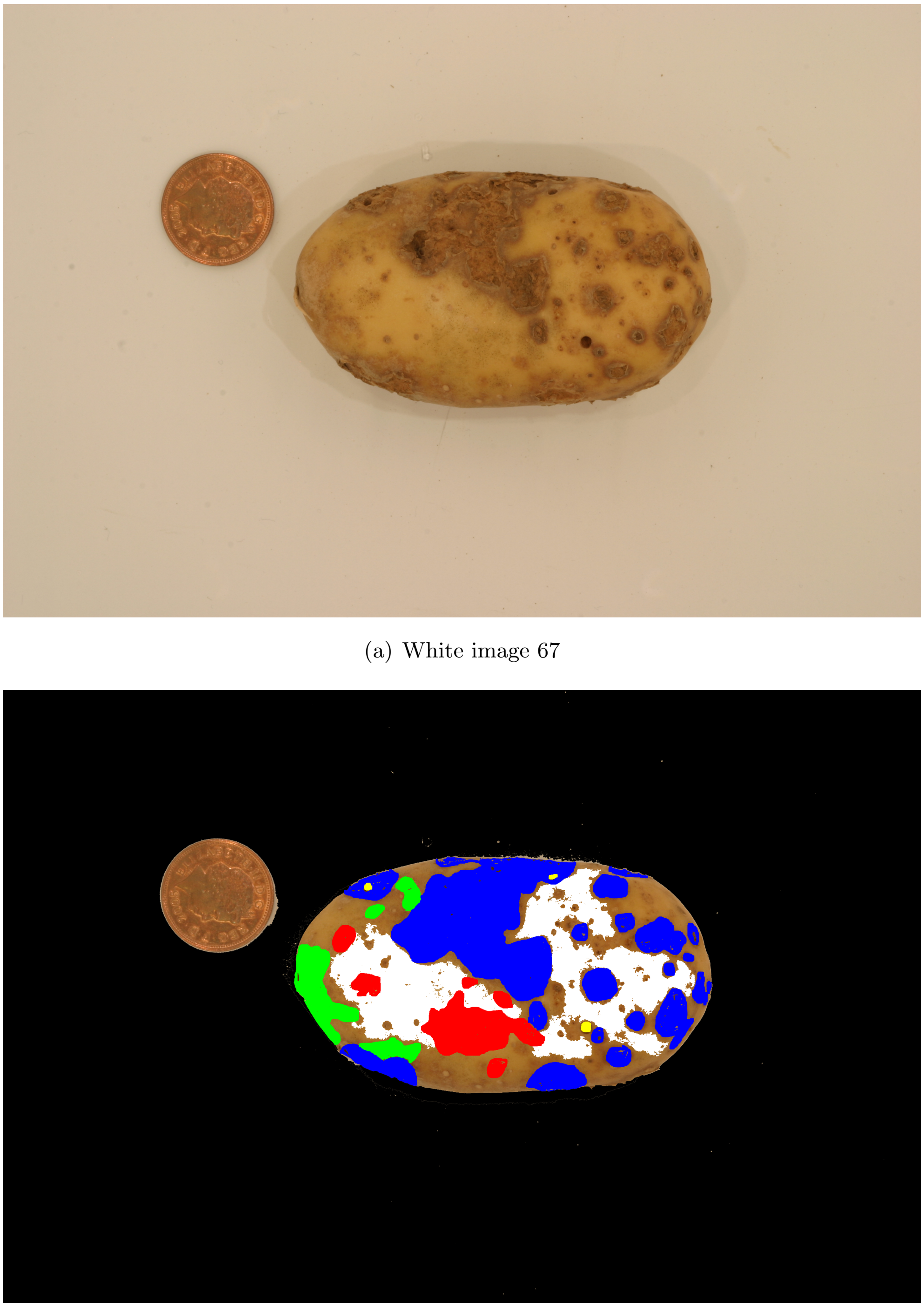
**Figure B.4:** Image 47 from the white data set. Red = silver scurf, Green = scab, Blue = damage



(a) White image 59

(b) Ground truth of White image 59

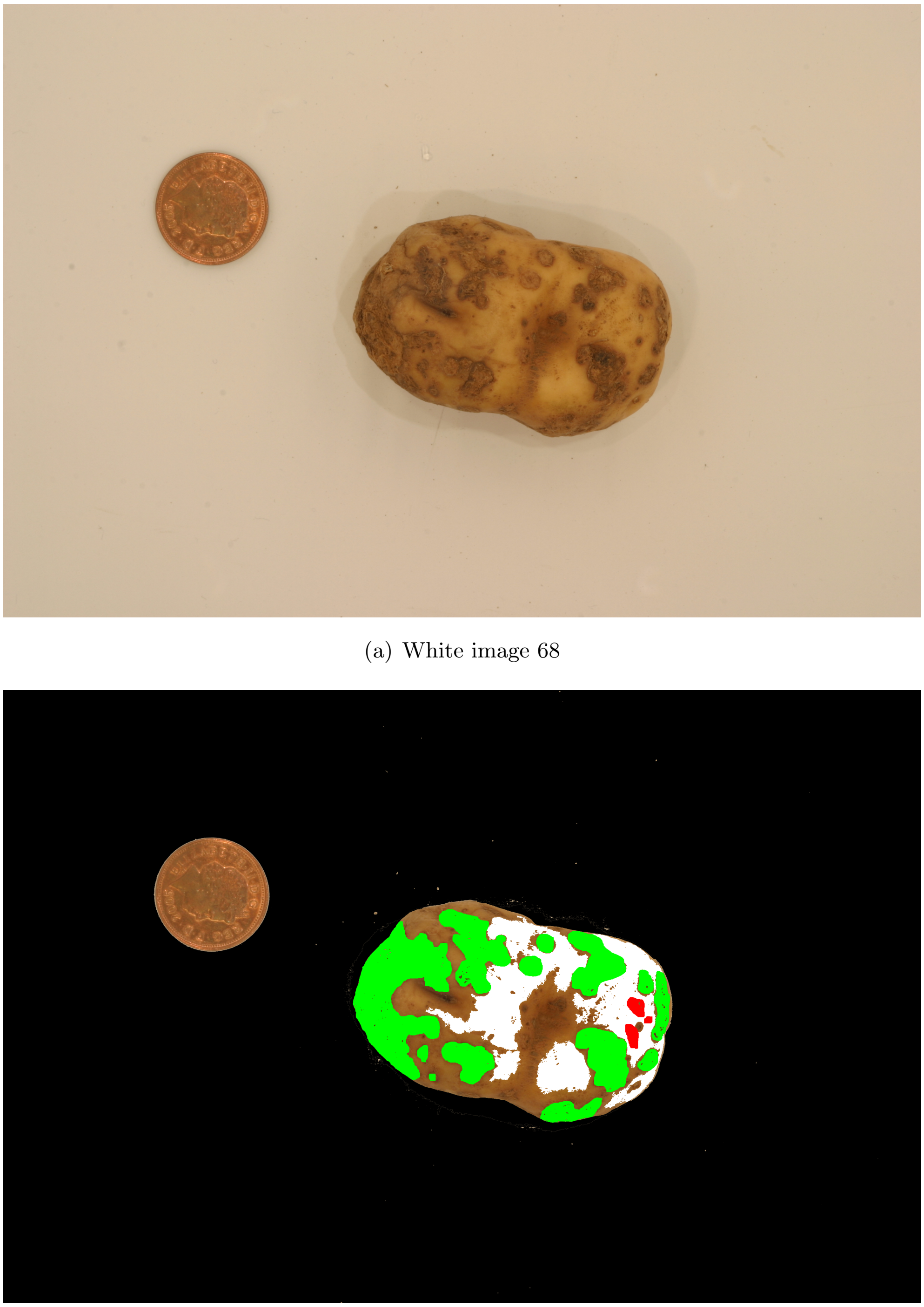
**Figure B.5:** Image 59 from the white data set. Red = silver scurf, Green = scab, Blue = damage



(a) White image 67

(b) Ground truth of White image 67

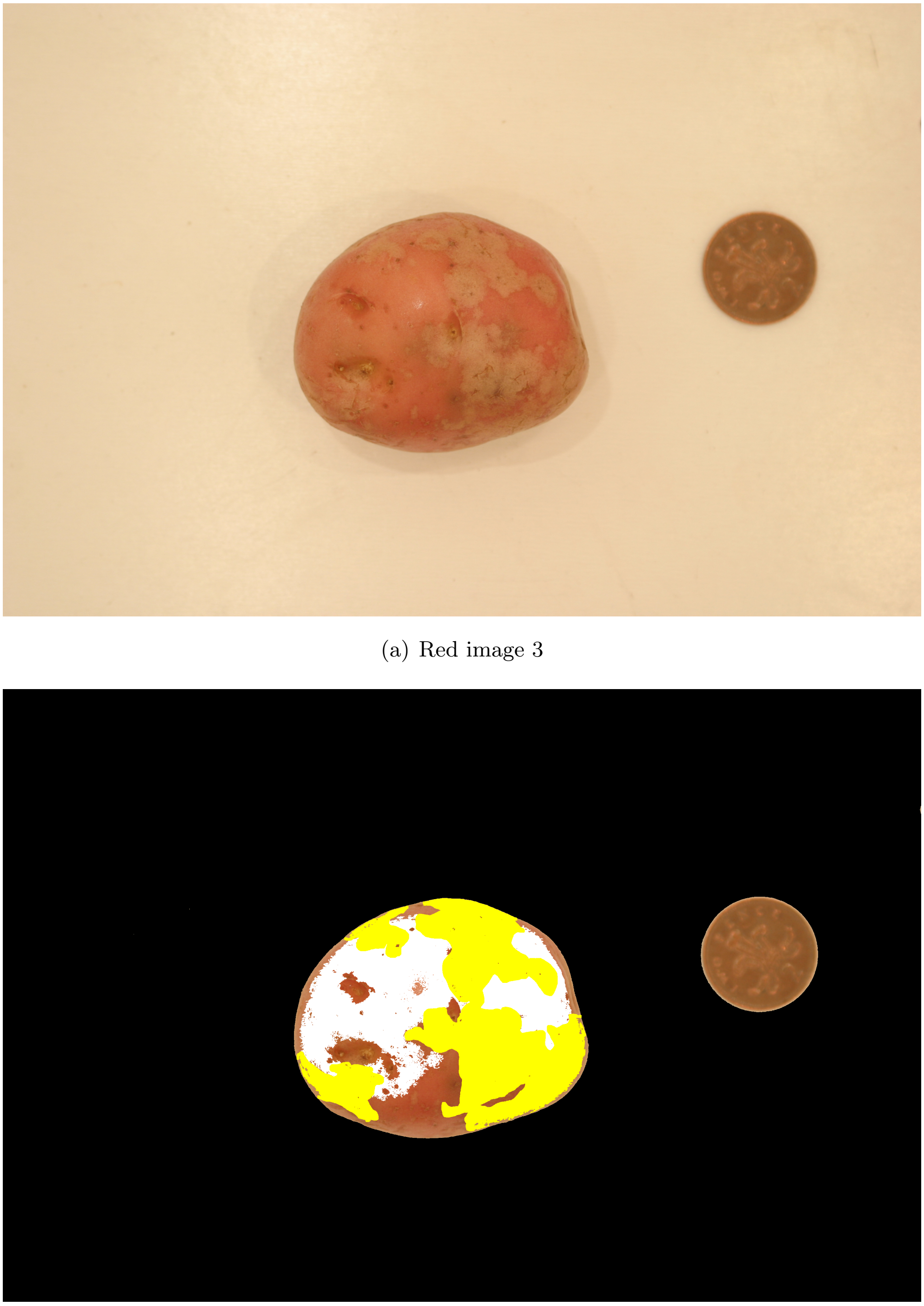
**Figure B.6:** Image 67 from the white data set. Red = black dot, Green = silver scurf, Blue = common scab, Yellow = damage



(a) White image 68

(b) Ground truth of White image 68

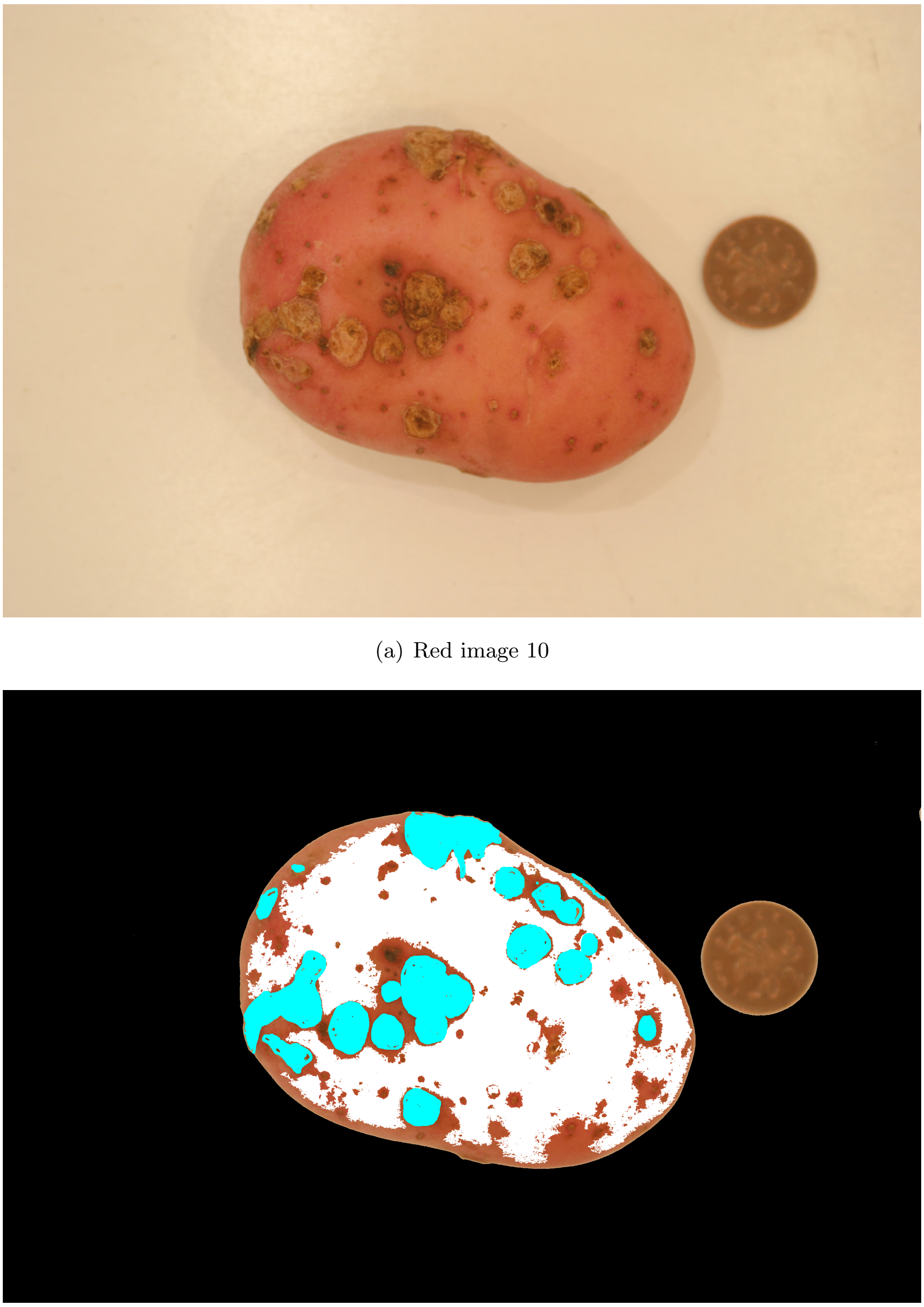
**Figure B.7:** Image 68 from the white data set. Red = black dot, Green = common scab



(a) Red image 3

(b) Ground truth of Red image 3

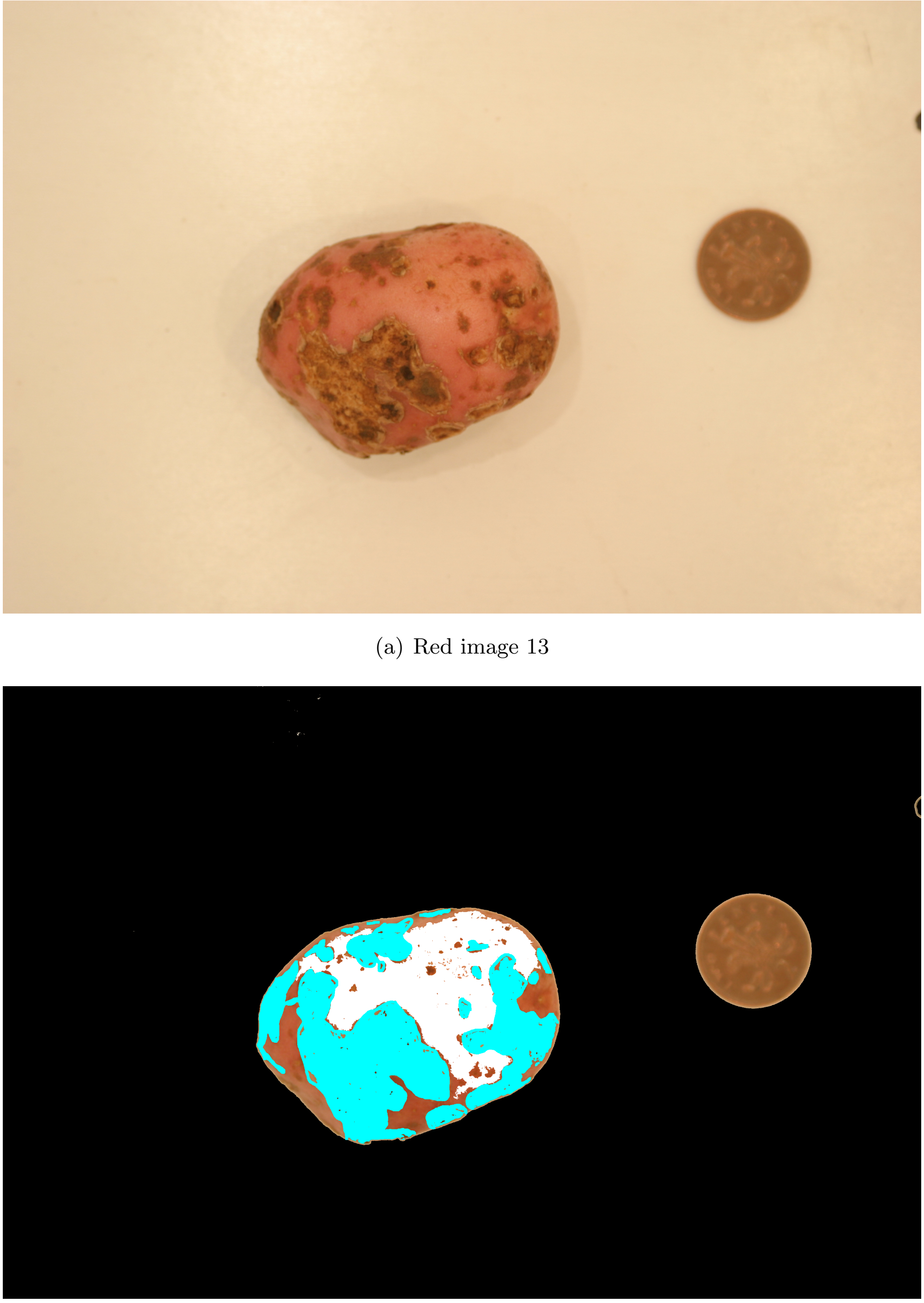
**Figure B.8:** Image 3 from the red data set. Blue = black dot, Green = greening, Yellow = silver scurf, Cyan = scab



(a) Red image 10

(b) Ground truth of Red image 10

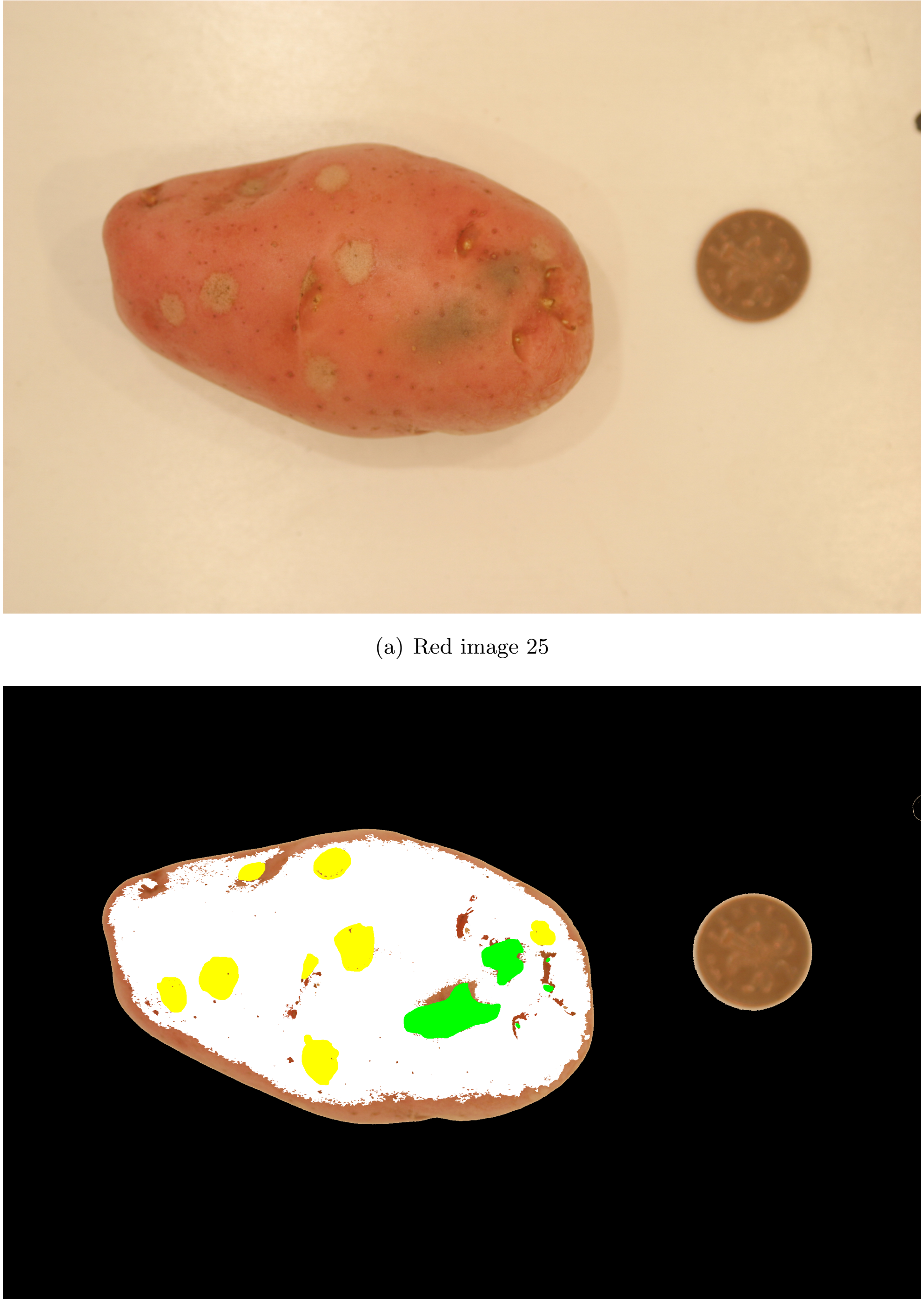
**Figure B.9:** Image 3 from the red data set. Blue = black dot, Green = greening, Yellow = silver scurf, Cyan = scab



(a) Red image 13

(b) Ground truth of Red image 13

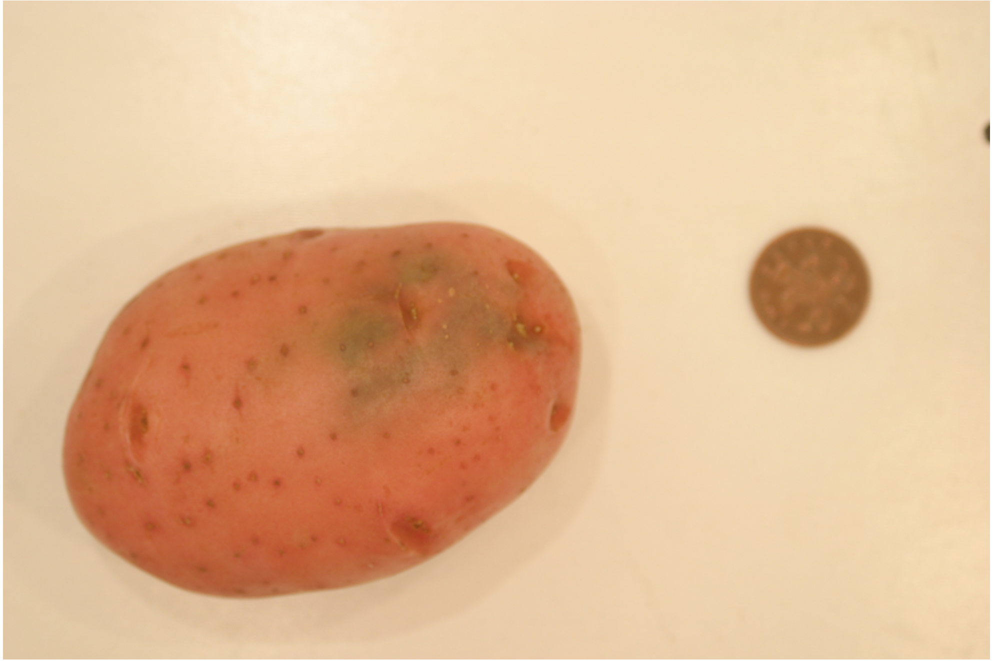
**Figure B.10:** Image 13 from the red data set. Blue = black dot, Green = greening, Yellow = silver scurf, Cyan = scab



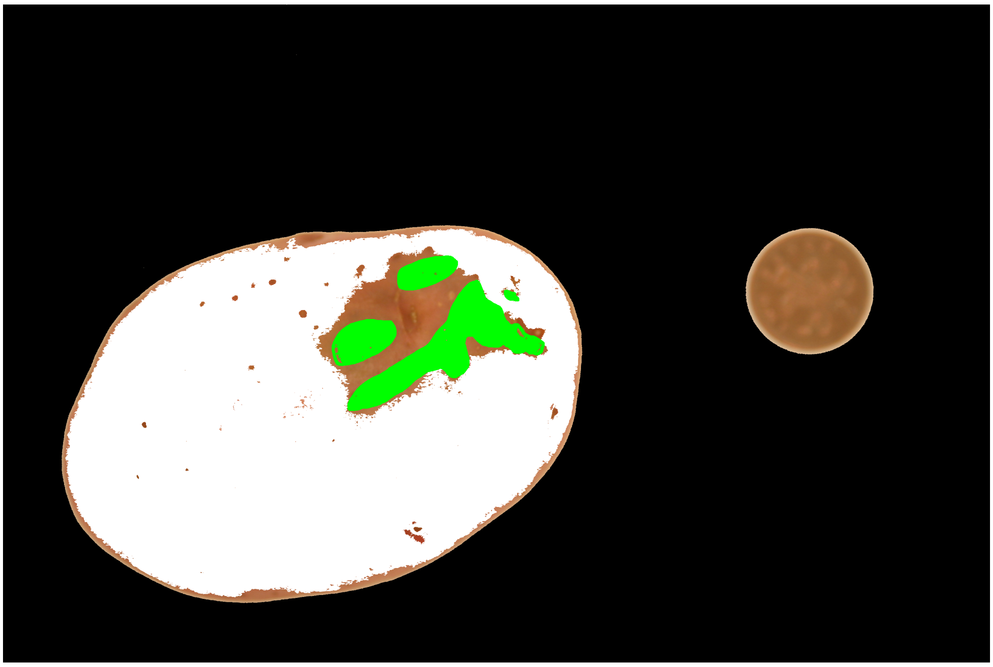
(a) Red image 25

(b) Ground truth of Red image 25

**Figure B.11:** Image 25 from the red data set. Blue = black dot, Green = greening, Yellow = silver scurf, Cyan = scab

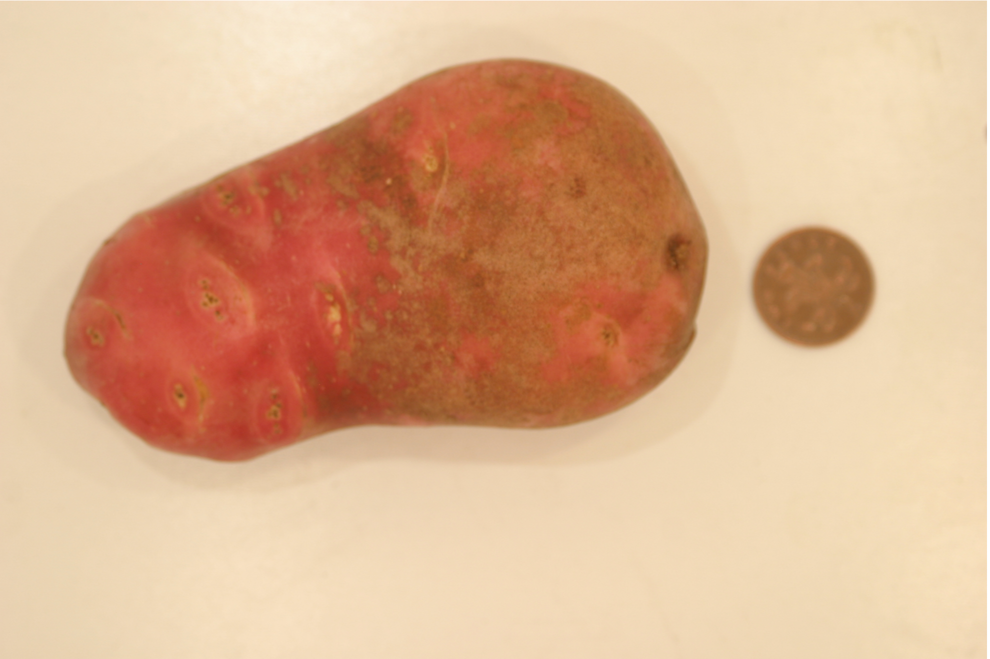


(a) Red image 27

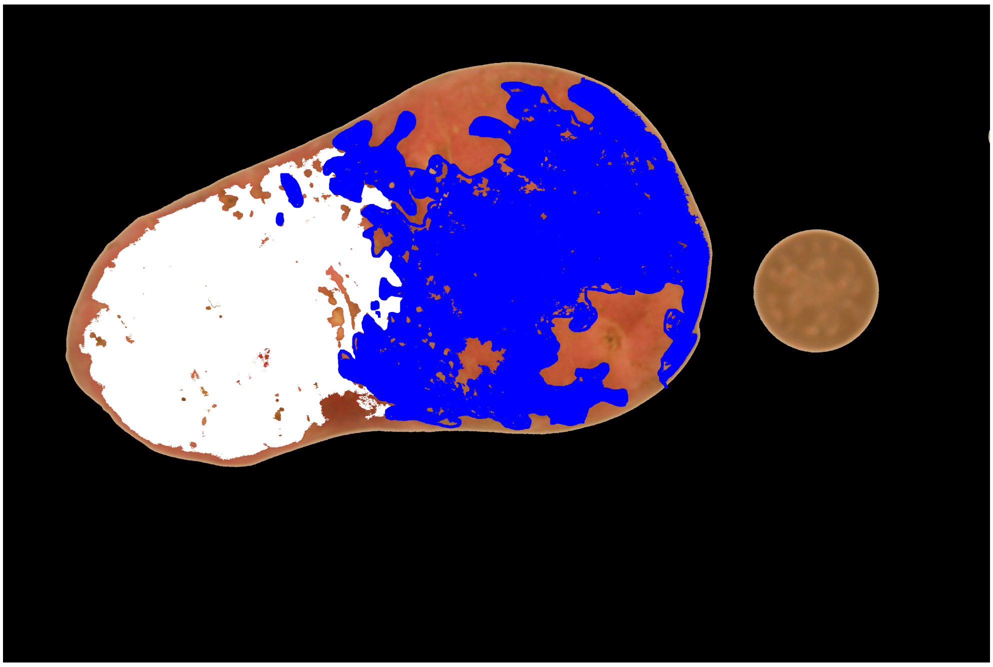


(b) Ground truth of Red image 27

**Figure B.12:** Image 27 from the red data set. Blue = black dot, Green = greening, Yellow = silver scurf, Cyan = scab

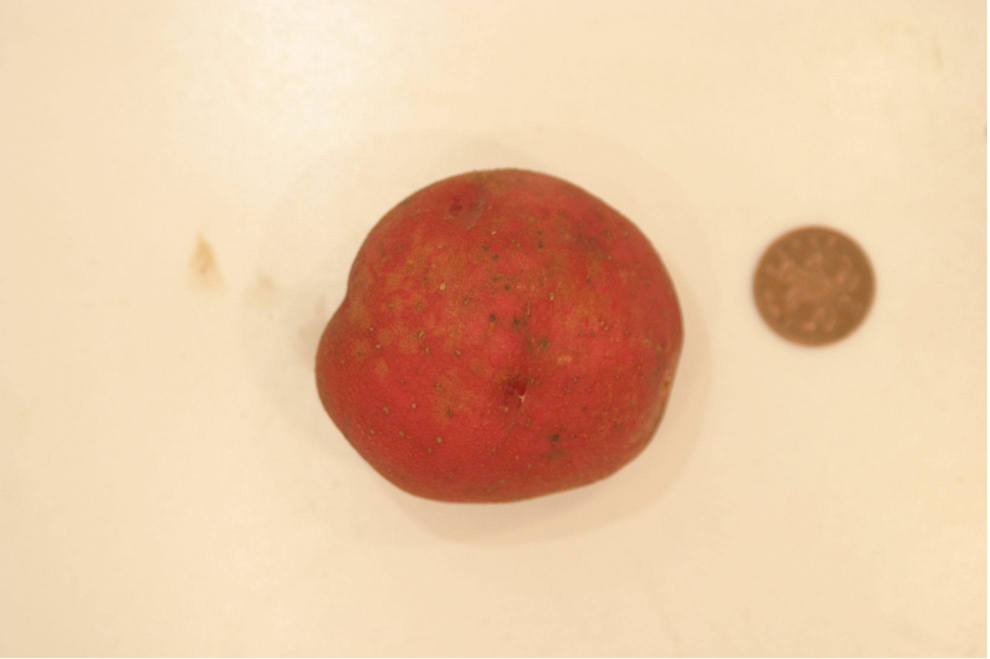


(a) Red image 38

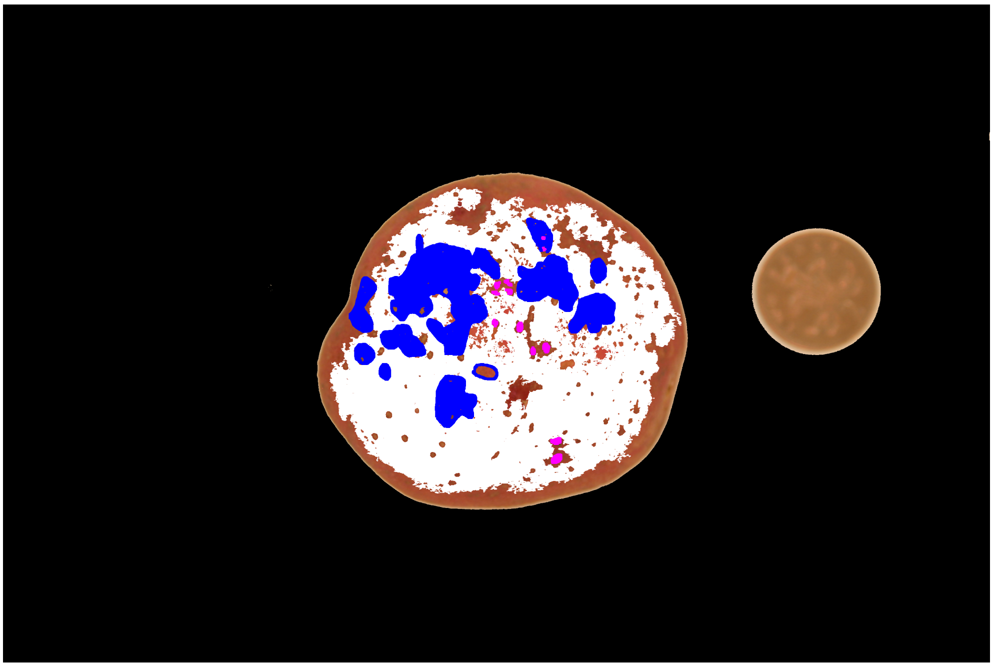


(b) Ground truth of Red image 38

**Figure B.13:** Image 38 from the red data set. Blue = black dot, Green = greening, Yellow = silver scurf, Cyan = scab



(a) Red image 47



(b) Ground truth of Red image 47

**Figure B.14:** Image 47 from the red data set. Blue = black dot, Green = greening, Yellow = silver scurf, Cyan = scab