THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Image Processing and Machine Learning for Segmentation and Classification of Aluminium Extrusions

by

Josefina Pehrson

A thesis submitted in fulfilment for the
degree of Master of Engineering

in the
School of Engineering

March 2022

Declaration of Authorship

I, Josefina Pehrson, declare that this thesis titled, 'Image Processing and Machine Learning for Segmentation and Classification of Aluminium Extrusions' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Acknowledgements

I would like to thank Dave Slomp at Architectural Profiles Ltd for their funding and giving me an opportunity to learn about computer vision with a practical application. I am grateful to my supervisor, Lee Streeter, for his guidance and support throughout this research project.

I would like to thank my friend Morgan Dally for providing advice from a data science point of view and answering my countless machine learning questions. Finally, I thank my partner Rida Zabad for his support, encouragement and consolement, particularly during lockdowns.

Abstract

Inventory management is one of the most important components of administration in a large-scale manufacturing business. Architectural Profile Limited produces tonnes of aluminium profiles every single day and have over one thousand different extrusions available. Many of the profiles look almost identical to each other and are very difficult for even a highly experienced employee to distinguish. Sending out an incorrect extrusion will cost time and money for both the manufacturer and the customer.

Automation using machine vision is already a prominent tool used in factory environments worldwide. The ability to automate repetitive tasks frees up time for higher skill tasks and reduces manual data entry errors. A real time inventory monitoring system would allow for quick and confident decision making and reduce order problems due to out of date stock numbers.

The aim of this project is to explore whether classification of extrusion profiles from images is possible and what factors ensure an accurate classification system. To achieve the aim, new data was acquired that captures a range of variation such as different angles, positions and lighting.

In this thesis two main paths were explored, segmentation followed by classification of binary images using generic Fourier descriptors, and direct classification from colour images. This thesis presents a qualitative and quantitative analysis of the different approaches, and discusses the challenges and limitations found with the data. The methods are compared and recommendations are highlighted regarding the next steps that aid creation of an automated aluminium extrusion inventory management system.

Presented is a pipeline of extracting features using techniques such as Gabor filters, Gaussian blur, and edge detectors. The output filters are used to train a random forest pixel classifier to segment the extrusions. Perfect reference binary images generated via a semi-manual threshold process are used to train a Logistic Regression classifier, and the test images were automatically segmented and classified. This classifier makes predictions on the segmented images, and the testing accuracy achieved was 88%. Instances

that were misclassified had poor segmentations, in most cases from uneven illumination on the extrusion end. The other promising method is training convolutional neural networks using transfer learning. VGG-19 was trained with only the dataset acquired and achieved a testing accuracy of 89%, while ResNet-50 was trained using data augmentation and achieved a testing accuracy of 90%. The resulting confusion matrices show misclassifications between extrusions that are very similar in shape and size.

Further work will improve the performance of the classifiers by fine-tuning the parameters and carrying out parameter searching. Further investigation should be carried out given more requirements for the application, such as the processors and exact lighting conditions.

Contents

List of Figures vii

1 Introduction 1

2 Literature Review and Background 6
2.1 Image Processing Techniques . . . 6
2.1.1 Median Filter . . . 6
2.1.2 Gaussian Blur . . . 7
2.1.3 Edge Detection Operators . . . 7
2.1.4 Morphological Operators . . . 8
2.1.5 Generic Fourier Descriptor . . . 9
2.2 Features Implemented in Weka . . . 10
2.2.1 Simple Color Histogram . . . 10
2.2.2 Color Layout Descriptor . . . 11
2.2.3 Auto Color-Correlogram . . . 11
2.2.4 Edge Histogram . . . 12
2.2.5 Fuzzy Opponent Histogram . . . 13
2.2.6 Gabor Filters . . . 14
2.3 Machine Learning Algorithms . . . 14
2.3.1 Decision Tree . . . 14
2.3.2 Random Forest . . . 15
2.3.3 Support Vector Machine . . . 16
2.3.4 Mask R-CNN . . . 16
2.3.5 Logistic Regression . . . 18
2.4 Evaluation Metrics . . . 19
2.4.1 Accuracy and Precision . . . 19
2.4.2 Sensitivity and Specificity . . . 20
2.4.3 F1-Score . . . 21
2.5 Review of Applications of Supervised Machine Learning and Deep Learning Approaches . . . 21
2.6 Image Segmentation . . . 25

3 Segmentation of Extrusion End Shapes 28
3.1 Data Acquisition . . . 28
3.2 Data Processing Methods . . . 33
3.2.1 Approach 1: Image Processing Filters . . . 33
3.2.2 Approach 2: Classical Machine Learning . . . 34

3.2.3 Approach 3: Mask R-CNN 35
3.3 Results and Discussion 35
3.4 Classification using Fourier Descriptors on Segmented Extrusions 42

4 Classification and Object Detection of Extrusion End Shapes 45
4.1 Extrusion Classification Methods 45
4.1.1 Classical Machine Learning 45
4.1.1.1 Multi-Class 46
4.1.1.2 One vs Rest 46
4.1.2 Transfer Learning in Matlab 46
4.1.3 Object Detection Neural Networks with Detectron2 48
4.1.3.1 Mask R-CNN 48
4.1.3.2 Faster R-CNN 48
4.2 Results and Discussion 48
4.2.1 Classical Machine Learning 49
4.2.2 Transfer Learning 51
4.2.3 Object Detection 60

5 Discussion and Comparison of Classification Pipelines 64

6 Conclusion and Future Work 66

List of Figures

1.1 Correction of barrel distortion in the initial image dataset. 2
1.2 Images taken inside the warehouse of aluminium extrusions. 3
1.3 Ceiling and lighting above extrusions in the APL warehouse. 4

2.1 An example of a standard structure of a deep learning algorithm. (Reprinted with permission from Abraham, 2005) 17
2.2 The Mask R-CNN framework for instance segmentation (Reprinted with permission from He *et al.*, 2017) 18

3.1 Ceiling and lighting above extrusions in the APL warehouse. 29
3.2 Each of the extrusions and their class number assigned for experiments. . 29
3.3 Camera setup and lighting for data acquisition. 30
3.4 Examples of the four angles captured for each extrusion profile, in the top right corner. 30
3.5 Examples of the eight orientations captured for each extrusion profile, taken at an angle of 10 degrees. 31
3.6 Examples of different positions the extrusions were captured in. 31
3.7 Shade correction performed on a grayscale image to obtain binary image. 34
3.8 Examples of images that did not segment well from the random forest pixel classifier. 37
3.9 Segmented output images from all methods on image flat_corner_83 38
3.10 Segmented output images from all methods on image 30d_middle_31 . . . 39
3.11 Segmented output images from all methods on image 20d_left_corner_44 . 40
3.12 Segmented output images from all methods on image flat_left_corner_81 . 41

4.1 Training Validation Curve for AlexNet. 51
4.2 Each of the extrusions and their class number. 53
4.3 Training Validation Curve for VGG-19. 53
4.4 Training Validation Curve for GoogLeNet. 55
4.5 Training Validation Curve for ResNet-50. 56
4.6 Each of the extrusions and their class number. 57
4.7 Training Validation Curve for ResNet-50 with Data Augmentation. 58
4.8 Examples of Mask R-CNN output failure cases. 61
4.9 Examples of Faster R-CNN output failure cases. 62

Chapter 1

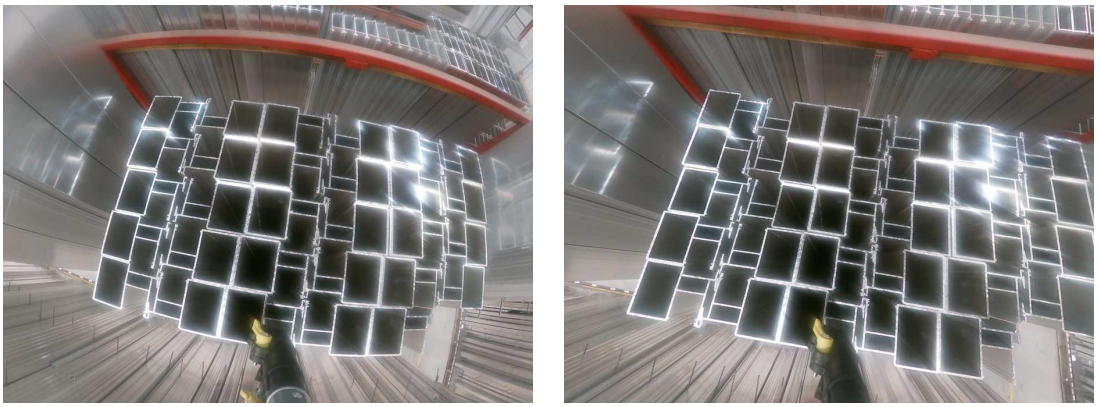
Introduction

Construction is a multi-billion dollar industry in New Zealand, and Architectural Profiles Ltd (APL) is a billion dollar company that manufactures windows, doors and aluminium joinery. APL produces thousands of aluminium extrusions every single day, and even highly experienced staff find difficulty distinguishing between them. Many categories of extrusions are extremely similar, having differences of only millimetres. It is practically impossible for an employee to remember all 1000+ extrusion names and uses.

With the company constantly growing, more profiles are designed and more extrusions are produced. This growth creates increasing complexity to managing the warehouse of inventory. Stock taking is one of the most time consuming and tedious tasks that requires minimal skill. Automating the identification process promises a more efficient and cheaper alternative for the everyday task. Workers introduce factors of error, such as errors due to fatigue, sickness, lack of concentration, and forgetfulness. Inventory management with artificial intelligence will be more efficient, more reliable, smarter and faster than any current system. In the context of extrusion management, automation has yet to be seen.

In conjunction with an automated warehousing system and autonomous guided vehicles for moving product, the total benefits could reach as much as $1.35M annually. These cost reductions primarily come from fewer employees to manage stock and fewer needless back orders.

Computer vision is a large field that allows system automation through cameras. Shape, colour and texture are prime examples or factors that algorithms base decisions on. There are many applications of object detection and classification across industries, robotic fruit harvesters (Jimenez *et al.*, 2000), plant and weed identification (Sharma



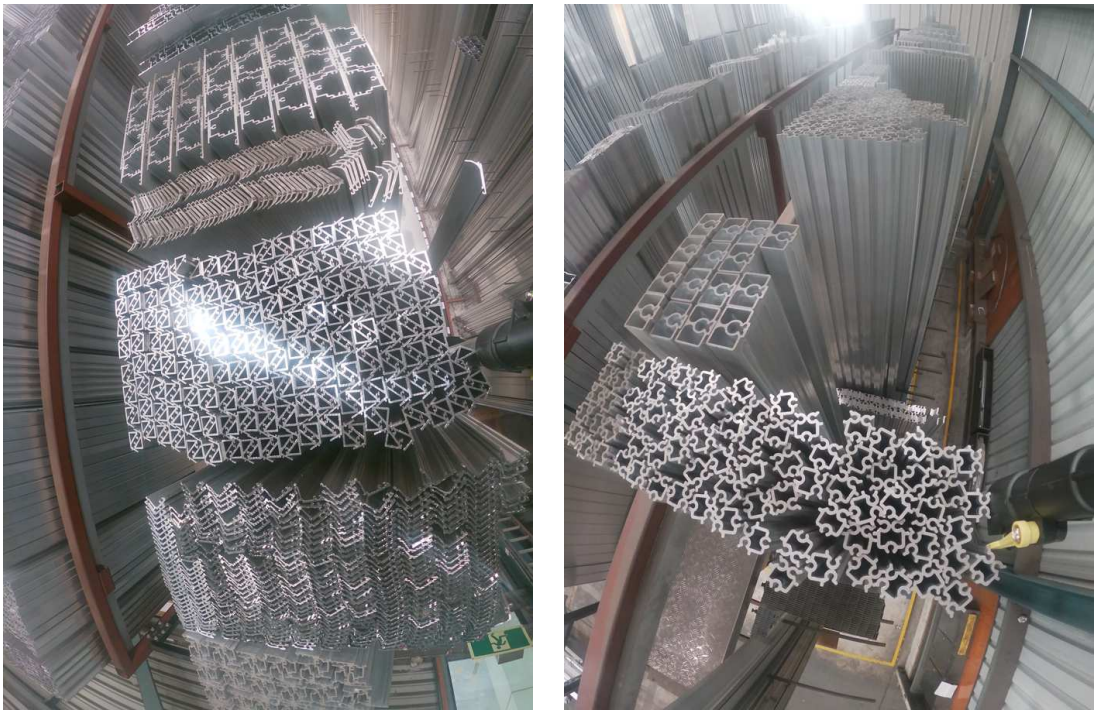
(A) Original image with clear barrel distortion. (B) Image with barrel distortion corrected.

FIGURE 1.1: Correction of barrel distortion in the initial image dataset.

et al., 2020), car make and model recognition (Llorca *et al.*, 2014) and self-driving vehicles (Alam *et al.*, 2020) to name a few. Before any implementation, extensive research must be carried out to explore the most accurate, efficient and cost-effective methods.

An initial dataset of images was provided and was taken from inside the APL warehouse. In the warehouse the profiles are standing in bays for storage in between orders. A GoPro camera on the end of a tall stick captured bays from above. This camera has a fisheye lens which adds barrel distortion to the captures to broaden the field of view: the centre is magnified and decreases as it goes out, causing the image to appear as though it is wrapped around a sphere. The true shape of the profiles in the image is distorted and this effect is undesirable for processing. With the knowledge of the camera used, the lens distortion parameter is easily found. An algorithm available on the MATLAB file exchange performs barrel distortion correction, and the correction can be seen in Fig. 1.1.

It is reasonable to expect that the higher the quality of the images, the better the systems capability to classify accurately. A significant issue with the presented dataset of profile images is saturated pixels from light reflections on the aluminium. The images in Fig. 1.1 show saturation in the top right. The light saturates beyond the edge of the profile, resulting in sections of missing data. Images with missing pixels do not provide adequate learning for the machine, therefore this type of data cannot be used to train a system. This saturation is present in 20% of the images given. The images also indicate the presence of lens flare which occurs easily by sun exposure through any windows or skylights. Understanding the nature of the varying sources of light is helpful in reducing and removing those background effects in the images. These effects cannot always be removed, hence the need to acquire better quality images.



(A) (B)

FIGURE 1.2: Images taken inside the warehouse of aluminium extrusions.

The main factors that differ between images in the initial dataset are lighting, orientation and angle (Figure 1.2). These are therefore factors that need to be included in the acquisition of new data. Given that the extrusions are inside a warehouse, there is rarely natural sunlight present in the images and hence less variation in light.

Upon visiting the warehouse, it was found that there is harsh led lighting on the warehouse ceiling (Figure 1.3), evidently causing the saturation in the images. To overcome this issue, APL provided 11 different sample extrusion ends for new image acquisition. The extrusions are purposefully chosen to be either very different from each other or very similar. If there are issues differentiating between extrusions that are similar during classification, those issues can be identified during the investigation.

The primary aim of the work in this thesis is to investigate methods of classifying aluminium extrusion profiles from images as a part of a system for inventory management. The valuable information to determine is how well different and similar extrusions can be classified and what factors affect the classification performance. An adequate system should be able to distinguish between extrusion ends regardless of whether they are similar or different from each other. To achieve this aim, new data must be acquired without the issues present in the initial dataset. To aid in creating an accurate system, performance limitations must be identified. These limitations may include illuminance in the image, the angle of the extrusions with respect to the camera, how different must

FIGURE 1.3: Ceiling and lighting above extrusions in the APL warehouse.

extrusions be to allow separation, what type of machine learning is able to classify the extrusions, and which algorithms best separate the extrusions.

Chapter 2 presents the background to all algorithms and processes used throughout the project, and a review on literature that discusses the performance of these algorithms in varying applications. These applications are all dependent on lighting and shape information from the object being detected, as this is similar to the problem of extrusion classification.

Chapter 3 begins with a discussion on the quality of the initial image data and the process of acquiring new data. An investigation follows, of three independent pipelines to achieve image segmentation. These pipelines make use of classical image processing techniques, supervised machine learning for pixel classification, and semantic segmentation deep learning models Mask R-CNN and Faster R-CNN. The output segmentations are compared to each other and then classified using extracted Fourier descriptors and machine learning models.

In Chapter 4, multiple methods to classify the colour images are explored. The first section details features extracted from the images using the machine learning package Weka, and subsequent results and discussion on classical supervised machine learning algorithms. The second section discusses transfer learning using Matlab. Matlab hosts many well-known pre-trained deep learning networks that can be trained on a given input dataset. In this tool, data augmentation is also briefly explored, given that deep learning networks typically need very large training datasets. The third section looks into instance segmentation using Mask R-CNN and Faster R-CNN. While Mask R-CNN was previously discussed for its segmenting ability, this chapter discusses its object detection capabilities.

Discussed in Chapter 5 is the comparison between the classical machine learning and deep learning algorithms using either segmented images with Fourier descriptors, or the original images with a set of extracted features. Also outlined is identified limitations and recommendations for a working system based on the results from this research.

Chapter 6 concludes the work done throughout the thesis, highlighting the main findings and addressing possible tasks for future work.

Chapter 2

Literature Review and Background

The goal of this project is to investigate image processing techniques and machine learning algorithms that best classify APLs aluminium profiles. Due to a lack of literature on this application, the scope of the review is widened to shape based object detection and classification. The evaluation metric that is focused on during this review is the classification accuracy.

2.1 Image Processing Techniques

This section provides the theory behind image filters that aided in this research by either extracting useful feature vectors from the images or segmenting an extrusion in the image. A filter is a sliding kernel operation that processes pixels in blocks of designated shapes. The kernel can be designed to perform specific actions, such as noise reduction, edge detection and blurring. Since there is no entry immediately preceding the first and last pixels, the empty pixels copy the neighbouring pixels to fill the window.

2.1.1 Median Filter

The purpose of median filters is removing noise while preserving edges within an image. Median filtering is a digital filtering technique that minimises salt and pepper noise. The median pixel in the kernel neighbourhood becomes the new value of the centre pixel, repeating for every pixel in the image. The size of the neighbourhood can be redefined to suit the needs of the image depending on how much blurring is desired. For larger

kernels, the window is shrunk as it approaches the edges so that every window is full (Huang *et al.*, 1979).

2.1.2 Gaussian Blur

A Gaussian blur achieves blurring on an image by using a kernel sampled from a Gaussian function, to reduce noise and detail in an image.

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.1}$$

Equation 2.1 shows the Gaussian function in two dimensions that calculate the transformation for each pixel. Values from the Gaussian distribution build a convolution matrix, creating the kernel operation. The centre pixel becomes the weighted average of its neighbourhood. Due to the Gaussian function the centre pixel has the largest weighting and the weighting decreases the further the pixel is from the centre (Shapiro and Stockman, 2001).

2.1.3 Edge Detection Operators

Edge detection aims to detect a variety of edges in a grayscale image using a multi-step algorithm. Some common edge operators include Roberts (Eq. 2.2), Sobel (Eq. 2.3), Prewitt (Eq. 2.4) and Scharr (Eq. 2.5).

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.2}$$

(Shrivakshan and Chandrasekar, 2012)

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{2.3}$$

(Sobel *et al.*, 1973)

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \tag{2.4}$$

(Shrivakshan and Chandrasekar, 2012)

$$h'_x(:,:) = \begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix} \tag{2.5}$$

(Watkins and Van Niekerk, 2019)

The kernels of these operators respond to vertical edges, and the horizontal kernels are obtained by a 90 degree rotation. The operators applied on the image create two derivatives, one image that represents the horizontal changes and another image for the vertical changes. Equations 2.6 and 2.7 take the derivative images and compute the gradients magnitude and direction. These more straight forward edge detection methods use a threshold on the gradients to produce the binary image.

$$|G| = \sqrt{{I_x}^2 + {I_y}^2} \tag{2.6}$$

$$\theta(x,y) = \arctan\left(\frac{I_y}{I_x}\right) \tag{2.7}$$

(Canny, 1986)

The first step of the Canny edge detection operator is convolution with a Gaussian kernel to perform noise reduction. The second step is another convolution with a kernel, the same method outlined above with an edge operator to compute the gradient and direction. The kernel is up to the user but usually resembles the Roberts or Sobel operators. Non-Maximum Suppression is performed to thin out all the detected edges. Depending on the strength of the gradient, the edges are split into three categories, strong, weak and non-relevant, using a double threshold. A hysteresis mechanism follows where a weak pixel that is next to a strong pixel is transformed into a strong pixel, otherwise it is transformed to a non-relevant pixel. The final output is a binary image.

2.1.4 Morphological Operators

Dilation is the process of filling small holes and thickening lines in a binary image. Given knowledge of the objects being extracted, a structuring element of a shape defines the neighbourhood. By setting the new value to equal the brightest pixel in the neighbourhood, the shape in the image enlarges based on the shape of the structuring element (Soille, 2004). Erosion performs the opposite steps, where the purpose is to remove floating pixels and thin out lines, by setting the new value to equal the lowest intensity pixel in the neighbourhood.

Closing is an operation that first dilates an image and then erodes the image with the same structuring element for both processes. Using both operators together allows the filling of small holes while preserving the size and shape of larger objects in the image (Dougherty and Lotufo, 2003). Opening performs the opposite steps, where erosion is applied first and then dilation second. This order allows the removal of small objects and detail from an image.

2.1.5 Generic Fourier Descriptor

Generic Fourier descriptors are computed on binary images that contain one object with the centroid of the shape also being the centre of the image (Zhang and Lu, 2002). A segmentation must be performed first, to create the binary image from the original colour image. The object in the image must be centred[1], where x_c and y_c is the location of the centre of the image. Centring is necessary to allow translation invariance between the data samples. The shape image is then converted from Cartesian space to polar space.

$$I = \{f(x, y); 0 \leq x < M, 0 \leq y < N\} \tag{2.8}$$

$$I_p = \{f(r, \theta); 0 \leq r < R_m, 0 \leq \theta < 2\pi\} \tag{2.9}$$

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{2.10}$$

$$\theta = \arctan\left(\frac{(y - y_c)}{(x - x_c)}\right) \tag{2.11}$$

Where x and y define the pixel location in cartesian space, and r and θ define the polar space co-ordinate of the pixel . Performing 2-D Fourier transforms in Cartesian space does not allow rotation invariance as rotated images will have differing Fourier spectra. The polar image in polar space is treated as a normal 2-D rectangular image in Cartesian space. Therefore, the resultant Fourier polar transform has a similar form to the normal 2-D discrete Fourier transform in Cartesian space. Rotated images will have a circular shift in polar space, not affecting spectra distribution. (ρ, ϕ) represents polar co-ordinates in the frequency plane, R and T are radial and angular resolutions.

$$0 \leq \rho < R \tag{2.12}$$

$$0 \leq \phi < T \tag{2.13}$$

[1] https://au.mathworks.com/matlabcentral/fileexchange/52560-centerobject-bw

$$0 \leq i < T \tag{2.14}$$

$$\theta_i = \frac{i2\pi}{T} \tag{2.15}$$

$$F_p(\rho, \phi) = \sum_r \sum_i f(r, \theta_i) e^{-2j\pi(\frac{r}{R}\rho + \frac{2i\pi}{T}\phi)} \tag{2.16}$$

For each pixel in the image, the real and imaginary spectra are computed as;

$$F_r = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) \cos\left(2\pi\rho\frac{r}{R_m} + \phi\theta\right) \tag{2.17}$$

$$F_i = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) \sin\left(2\pi\rho\frac{r}{R_m} + \phi\theta\right) \tag{2.18}$$

The real and imaginary values are summed together independently for each value of ρ and ϕ. The first magnitude (the DC component) for the Fourier descriptor (FD) is computed and used as a normalisation for subsequent calculations, allowing scale invariance.

$$\text{For } (\rho, \phi) = (0, 0)$$

$$\text{FD}(0) = \sqrt{\frac{2{F_r}^2}{{R_m}^2\pi}} \tag{2.19}$$

$$\text{FD} = \sqrt{\frac{{F_r}^2 + F_i^2}{\text{FD}(0)}} \tag{2.20}$$

2.2 Features Implemented in Weka

Weka is the Waikato Environment for Knowledge Analysis [2]. Weka is an open source software package in which is implemented a collection of machine learning algorithms and tools for working with data. Weka includes a range of filters that can be applied to data, including filters for images. This section details some of the image filters that are implemented in Weka.

2.2.1 Simple Color Histogram

The simple histogram comprises three histograms for each of the colour channels, red, blue, and green, in an RGB image. The pixel value range, for each colour, from 0 to

[2] https://www.cs.waikato.ac.nz/ml/weka/

255 and is divided into 32 bins. Similar images will contain similar proportions of each colour, creating comparable descriptor values (Chakravarti and Meng, 2009).

2.2.2 Color Layout Descriptor

The color layout filter describes the spatial distribution of colours, using grid based representation and a discrete cosine transform (Kasutani and Yamada, 2001).

The image is first partitioned into 64 blocks to allow invariance to resolution or scale. A single representative colour is then selected for each block, typically through taking the average of the pixel colours within the block. The result is a tiny 8x8 size image. The image is converted from RGB colour space to YCbCr colour space (luminance and blue and red chrominance). The discrete cosine transform is applied to each of the three colour components (Equation 2.21) (Ohm *et al.*, 2001).

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+q)p}{2N} \tag{2.21}$$

$$0 \leq p \leq M-1, 0 \leq q \leq N-1$$

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 \leq p \leq M-1 \end{cases}$$

$$\alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq q \leq N-1 \end{cases}$$

Lastly, a zigzag scanning is performed that transform the 8x8 matrices into 1-D arrays suitable for a descriptor. A zigzag scan is applied to group the coefficients from low to high frequencies producing a sorted matrix. The scan begins at (0,0) and follows in an anti-diagonal (bottom right corner to upper left corner) direction (Rahman *et al.*, 2017).

2.2.3 Auto Color-Correlogram

The colour correlogram expands upon a histogram (Gonzalez and Woods, 2008) and is used for image indexing and comparison, by obtaining the spatial distribution of colours. The correlogram can also describe the global distribution of local spatial correlation of

colours (Huang *et al.*, 1997). Informally, colour correlogram describes "the probability of finding a pixel of colour c_j at a distance k from a pixel of colour c_i in the image".

The correlogram (γ) for image I is defined in Equation 2.22, giving the probability (Pr) that a pixel at distance k away from the given pixel are of the same colour.

$$\gamma_{ci,cj}{}^{(k)}(I) \triangleq \Pr_{p_1 \in I_{cj}, p_2 \in I}[p_2 \in I_{cj} || p_1 - p_2| = k] \tag{2.22}$$

The first step is to compute how many pixels have an identical colour given that the distance between pixels is k in the neighbourhood (Equation 2.23) and the histogram of the image I (Equation 2.24). These two expressions are manipulated to create the colour correlogram (Equation 2.25).

$$\Gamma_{ci,cj}{}^{(k)}(I) \triangleq |\{p_1 \in I_{ci}, p_2 \in I_{cj} || p_1 - p_2| = k\}| \tag{2.23}$$

$$h_{c_i}(I) \triangleq n^2 \cdot \Pr_{p \in I}[p \in I_{c_i}] \tag{2.24}$$

$$\gamma_{ci,cj}{}^{(k)}(I) = \frac{\Gamma^{(k)}_{ci,cj}(I)}{h_{c_i}(I) \cdot 8k} \tag{2.25}$$

The auto color-correlogram of image I captures spatial information between pixels of identical colour only (Huang *et al.*, 1997). An example of a neighbourhood is the pixels in the positive and horizontal directions, shown in Equations 2.26 and 2.27. Pixel neighbourhoods are specifiable to each correlogram.

$$\lambda_{(x,y)}{}^{c,h} \triangleq |\{(x+i, y) \in I_c; 0 \leq i \leq k\}| \tag{2.26}$$

$$\lambda_{(x,y)}{}^{c,v} \triangleq |\{(x, y+j) \in I_c; 0 \leq j \leq k\}| \tag{2.27}$$

2.2.4 Edge Histogram

MPEG-7 Edge Histogram captures spatial distribution of edges. The input image is divided into 4x4 non-overlapping blocks. Each block is further divided into 2x2 blocks for capturing local edge orientation. The image is resized to be divisible by 4 to achieve the block separation and five types of edge operators are defined. Each edge operator is applied to each 2x2 block. The horizontal and second diagonal operators are 90 degree rotations of the vertical and diagonal blocks in the clockwise direction, respectively.

$$h_{\text{Vertical}} = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \tag{2.28}$$

$$h_{\text{Diagonal 45}} = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & -\sqrt{2} \end{bmatrix} \tag{2.29}$$

$$h_{\text{Isotropic}} = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} \tag{2.30}$$

(Won *et al.*, 2002)

$$p = |\sum_{k=0}^{3} a_k d_k| \tag{2.31}$$

$$[a_k] = 2 \times 2 \text{ sub image}$$

$$[d_k] = \text{edge mask}$$

A p-value is calculated for each operator and block. The maximum p-value out of the five operators is compared to a threshold to find the dominant edge type. How many dominant edges there are for each edge type within the block is recorded as the feature. There are eighty features in total, derived from 4x4 blocks and the 5 operators.

2.2.5 Fuzzy Opponent Histogram

The fuzzy opponent histogram is an opponent colour space histogram that considers the uncertainty of the colour components, due to quantization effects, human perception, and the inherent variability of grey levels in an image (Jawahar and Ray, 1996).

$$O_1 = \frac{R-G}{\sqrt{2}}; O_2 = \frac{R+G-2B}{\sqrt{6}}; O_3 = \frac{R+G+B}{\sqrt{3}} \tag{2.32}$$

The colour data are transformed according to Equation 2.32. The subtraction in the first two channels allows offsets to cancel out if they are equal for colour channels, allowing shift invariance with respect to light intensity.

Fuzzy logic is a form of many valued logic where truth values can be any real continuous number between zero and one. Fuzzy sets are characterised by a membership function. There are boundaries set for numbers of partial truth. The number of bins is defined by the length of the fuzzy membership and the amount of colour channels, giving 64 bins.

For each pixel, the fuzzy membership and the eight neighbouring grey pixel values are calculated and the number of these pixels that are brighter than the pixel of interest. This gives one of nine possible values, resulting in the histogram with length 64x9 (576) (Vertan and Boujemaa, 2000).

2.2.6 Gabor Filters

Gabor filters are used in image recognition tasks involving texture segmentation, edge detection and feature extraction. Gabor filters are band pass filters, meaning they only let through a certain band of frequencies and reject the rest, or also described as a Gabor wavelet applied to an image to produce a Gabor filtered image (Weldon *et al.*, 1996). A Gabor kernel is described by;

$$g(x, y; \sigma, \theta, \lambda, \gamma, \varphi) = e^{-\frac{x^2+\gamma^2 y^2}{2\sigma^2}} e^{i(\frac{x2\pi}{\lambda}+\varphi)} \tag{2.33}$$

The parameters control the shape and size of the filter. The feature is the output of the filter, an image where lines are highlighted if they align with the orientation of the Gabor filter.

2.3 Machine Learning Algorithms

2.3.1 Decision Tree

A decision tree is a supervised machine learning method used in classification and regression applications (Breiman *et al.*, 2017). A decision tree only contains conditional statements that decide which branches to take in a tree-like structure. A decision tree begins at the root node that has some initial function that the instance is evaluated against. Each branch from this function is completely independent and can not have any uncertainty regarding the path the instance follows. The next node in the tree is considered the new root node of a sub-tree, and the process iteratively continues until all instances at a node have the same classification (Freund and Mason, 1999). This node is called the leaf node and is where this part of the tree stops developing.

The information gain method calculates which attribute to split data on at a given node, as it represents how much of the data can be split based on this attribute. An information gain of zero is ideal as it represents a pure daughter node, where there is only one class of information and the branch can stop developing (Witten *et al.*, 2011a).

A daughter node where all the outcomes are equal is where the information gain is maximum. The information gain is based on information entropy and Equation 2.34, where the entropy H, of a given variable X, with possible outcomes x_1 to x_n and their probability of occurrence P.

$$H(X) = -\sum_{i=1}^{n} P(x_i)\log P(x_i) \tag{2.34}$$

The information gain is defined as the difference between the information entropy of the parent node and the average entropy of the daughter nodes. In a more intricate dataset, reaching a pure daughter node without over fitting is far less likely. Often the decision tree stops learning before the pure node is reached and instead measures and accounts for the remaining error.

An alternating decision tree (ADTree) differs from the standard decision tree by the additional features. An instance being classified by an ADTree will follow all branches of the tree in which the condition is satisfied. There are multiple prediction scores calculated and the total of these adds to the final prediction of the class. Decision trees are transparent algorithms where the tree is easily visualized with the thresholds for each condition. Simpler trees are preferred as they are easier to compute and analyse (Kuang *et al.*, 2019).

2.3.2 Random Forest

Random Forest is an ensemble learning method, meaning it consists of multiple decision trees and can therefore also suit multi-class problems. Random forests implement a bootstrap aggregating algorithm for training, also termed bagging. Given a dataset with corresponding output values, bagging generates multiple n new training sets, each the size of one sample from the training dataset, with replacement. Then, n models are fitted through repeatedly sampling the single training sets, and averaging the voted class (Dietterich, 2000).

Trees that grow too deep over fit and form low bias but high variance. Random forests overcome this by averaging the deep trees, reducing the variance with a trade-off of small increase in bias, resulting in random forests being less sensitive to noise compared to a single tree.

The number of trees and other parameters are adjustable to improve the performance. In a classification task, each tree outputs the predicted class, and the class that is predicted the most is the final output (Breiman, 2001).

2.3.3 Support Vector Machine

Support Vector Machines (SVM) are suitable for binary and multi-class problems. Data that is linearly separable may make use of hard margins for classification. To allow SVM to classify data that is not linearly separable, support vector machines make use of soft margins where the margin can move depending on minimisation of the expression in Equation 2.35.

$$\lambda||w||^2 + \left[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(w^T x_i - b))\right] \tag{2.35}$$

The goal in training a support vector machine is to find the hyperplane with the largest margin separating the classes. The observations on the edge and within the soft margin are referred to as the support vectors, and are the most difficult patterns to classify. This defines a soft margin classifier, also known as a support vector classifier. In higher dimensional data the support vector classifier forms a line, a plane, or a hyperplane (James *et al.*, 2013). A larger margin implies better generalisation of the classifier (Hart *et al.*, 2000).

Not all data allows for the simple measures just described. SVMs rely on transforming the low dimensional data into higher dimensional data by a mapping function and the use of a kernel function to systematically find support vector classifiers to best classify the data (James *et al.*, 2013). Examples of the kernel function include polynomial and radial kernels, with the polynomial kernel function described in Equation 2.36.

$$k(\vec{x_i}, \vec{x_j}) = (\vec{x_i} \cdot \vec{x_j} + r)^d \tag{2.36}$$

The higher dimensionality of the feature space increases the generalisation error in support vector machines. However, provided there are sufficient samples the algorithm still performs well. The dominant approach for SVMs to classify multi-class problems is to break the classifiers down to binary classifiers in a one versus rest configuration. Each binary classifier assigns a vote for the class, and the class with the most votes is the predicted class (Duan and Keerthi, 2005).

2.3.4 Mask R-CNN

A convolutional neural network (CNN) is an artificial neural network suitable for imaging applications. They were created based on the neuron connections in a human visual

system (Figure 2.1). There is an input layer with as many neurons as there are pixels in the image. The first hidden layer performs a convolution, giving the algorithm the name and generating a feature map that feeds into the next layer. Each neuron has a weight and training a CNN consists of iteratively adjusting the weights based on the training data (Albawi *et al.*, 2017).

Unlike the classical machine learning algorithms above, deep learning requires a significant size image dataset for training due to the large number of parameters to train. Typically, the more images that are available for training, the better the performance of the classifier. Whereas the classical methods are able to produce accurate results with far less data in comparison.

An advantage of a CNN is a highly accurate black-box solution that performs its own feature extraction and algorithms, given sufficient data to train on. A 7 level network known as LeNet (LeCun *et al.*, 1998) was designed in 1998 to classify digits, understanding that to process higher resolution images, more layers in the network are required. The ImageNet Large Scale Visual Recognition Competition (Russakovsky *et al.*, 2015) has seen and produced some of the most used high performing CNN's in the last decade, with performance evaluated on over 1000 classes and 1 million image dataset. The 2012 winner was an 8 layer network, AlexNet (Krizhevsky *et al.*, 2012). 2014 saw another two deep networks, VGGNet with 19 layers (Simonyan and Zisserman, 2014), and GoogleNet with 22 layers (Szegedy *et al.*, 2015). RestNet came to light in 2015 with 152 layers and showed results outperforming that of humans (He *et al.*, 2016). Each network was considered the deepest at the time and have become the beginning point for any classification and object detection investigation.

Mask R-CNN is a convolutional neural network for object instance segmentation. This method extends from Faster R-CNN. Instance segmentation combines object detection with semantic segmentation, where pixels are classified to the object detected.

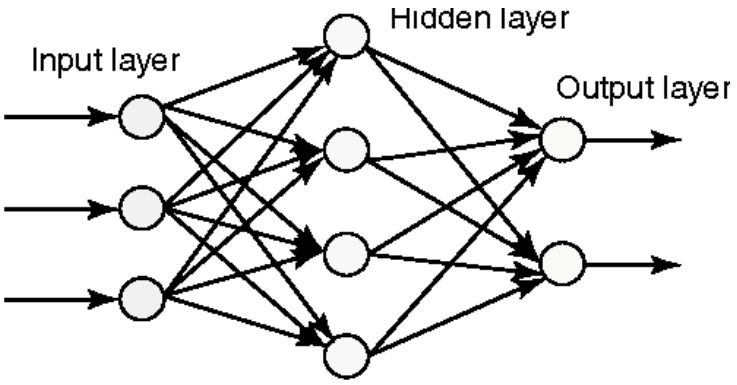


FIGURE 2.1: An example of a standard structure of a deep learning algorithm. (Reprinted with permission from Abraham, 2005)

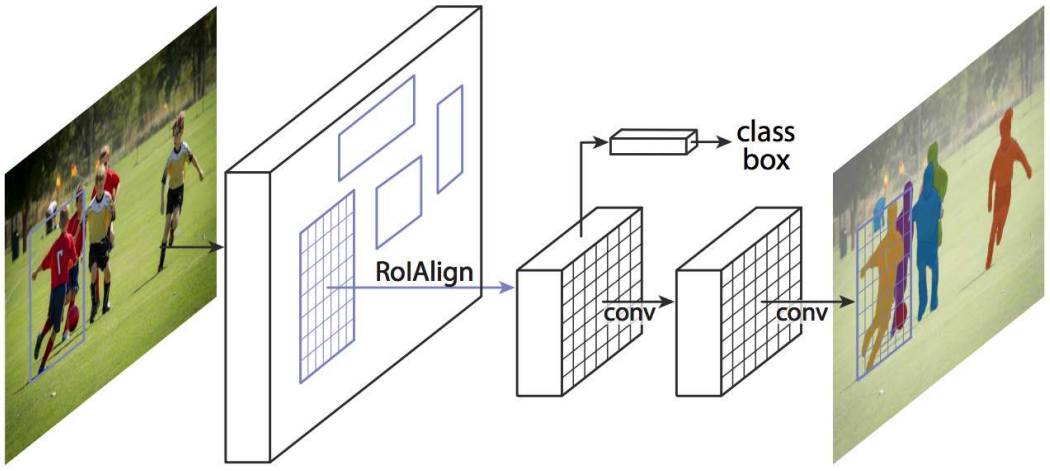


FIGURE 2.2: The Mask R-CNN framework for instance segmentation (Reprinted with permission from He *et al.*, 2017)

The R-CNN architecture begins by scanning the input image for probable objects using a 'selective search' method, generating an abundance of region proposals. Each of the regions are run through the base pre-trained CNN (LeNet, AlexNet, GoogleNet, VGGNet, ResNet). Each output from the CNN becomes the input to a SVM and a linear regressor. SVM classifies the region of object in image and linear regressor tightens the bounding box of the detected object (He *et al.*, 2017).

The Fast R-CNN advances from R-CNN as the input image is fed directly to the CNN to generate a convolutional feature map. Regions are identified from the map and fed into a fully connected layer. Softmax replaces SVM as the final classification layer (He *et al.*, 2017).

Faster R-CNN is a further modified version of Fast R-CNN and makes use of Region Proposal Network (RPN). Image feature maps from the CNN are input to RPN to generate anchor boxes of different shapes and sizes, with their respective probability of being an object. The boxes are passed to the Region of Interest (RoI) pooling layer, that extracts feature maps for each anchor. The feature maps are passed to a fully connected layer which includes softmax and linear regressor layers to classify the object and predict the bounding boxes (Figure 2.2) (He *et al.*, 2017).

Mask R-CNN follows on from Faster R-CNN by also outputting a binary mask for each region of interest in parallel to predicting the class and bounding box.

2.3.5 Logistic Regression

In binomial logistic regression, the two classes are labelled 0 and 1, with 2 probability distributions. The probability of one class is the inverse probability of the other, as the

two probabilities must sum to 1. The class is expressed as a linear combination of the features and trained weights. In multinomial logistic regression, each of the n classes have a separate probability distribution. Still the sum of the probabilities over the n classes must sum to 1. The nominal classes means that the labels are categorical and cannot be ordered in any way that is meaningful (Bishop and Nasrabadi, 2006).

For a multinomial logit model, there are multiple equal methods to create the underlying model. One method is to produce n-1 independent binary logistic regression models. One of the models is delegated as the pivot model, upon which all other models are regressed against. There is a weight for every feature in every class that is determined during training. Where β_k is the weights vector for outcome k, and X_i is the feature vector for observation i. From the matrix of weights and a given sample, the likelihood function gives the probability for each class. The highest probability is the output class.

$$\Pr(Y_i = K) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{\beta_k \dot{X}_i}} \tag{2.37}$$

Another method to formulating multinomial logistic regression is Equation 2.38, a representation of a simple linear probabilistic classifier in parametric form. Where there are K feature functions $f_k(y, x)$, and weight vectors of K length, w_k (Witten *et al.*, 2011b).

$$\Pr(y|x) = \frac{\exp(\sum_{k=1}^{K} w_k f_k(y, x))}{\sum_y exp(\sum_{k=1}^{K} w_k f_k(y, x))} = \frac{1}{Z(x)} \exp\left(\sum_{k=1}^{K} w_k f_k(y, x)\right) \tag{2.38}$$

Matrix vector formulation of multi-class logistic regression is also a widely used softmax function for the final layer of a neural network.

2.4 Evaluation Metrics

Models require metrics to prove how well they predict data. There are many metrics that represent this, and this section discusses and describes those metrics. A value of 100% for any of these metrics discussed indicates perfect performance, with zero misclassifications, but is however very difficult to achieve.

2.4.1 Accuracy and Precision

The metric that is used most frequently among the literature is classification accuracy. It is defined as the ratio between the number of correct classifications to the total number of samples (Tharwat, 2020).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.39}$$

Accuracy is not a suitable metric for binary problems where the data is unbalanced, as the accuracy could be very high when the outcome is actually terrible. For example, in a 100 sample data set with 10 true positives, if 99 are classified as negative then the accuracy will be 91%, even though only 1 out of the 10 true positives was classified correctly. However for balanced data, accuracy is good to use as an approximate overall metric. Therefore, there are more suitable metrics for binary classification, though for multi-class classification accuracy is a good rough metric to use. An accuracy above 90% is considered very good, but often depends on what is being classified. An indication that a model isn't contributing to the classification is if the accuracy is 100 divided by the number of classes. For example, in a 100 sample data set with 10 samples of each of the 10 classes, an accuracy of 10% indicates that the model predicts no better than random classification.

Precision defines the proportion of predicted positive cases that are actual positive samples (Powers, 2011). Precision is a good evaluation metric to use when the cost of a false positive is very high and the cost of a false negative is very low.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.40}$$

2.4.2 Sensitivity and Specificity

Sensitivity is also known as the true positive ratio. It is defined by the ratio between the number of correctly classified positive samples to the total number of positive samples (Powers, 2011). It is often termed as recall as well. Recall is a good metric to use when the cost of a false positive is very low and the cost of a false negative is very high, opposite to precision.

$$\text{TPR} = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{2.41}$$

Specificity is also known as the true negative rate. It is defined by the ratio between the number of correctly classified negative samples to the total number of negative samples (Powers, 2011).

$$\text{TNR} = \frac{TN}{FP + TN} = \frac{TN}{N} \tag{2.42}$$

Specificity and sensitivity are good metrics to use in single class problems, such as binary problems or in image segmentation. The metrics provide more insight than accuracy, as they better represents how well both the positives and negatives are being classified independently.

2.4.3 F1-Score

The F1-Score is a metric that represents a harmonic mean between precision and recall (Chicco and Jurman, 2020). For a full picture, both precision and recall must be evaluated together, as improving one typically reduces the other. Hence, for the F1-score to measure high, both precision and recall must also be high values.

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{2.43}$$

2.5 Review of Applications of Supervised Machine Learning and Deep Learning Approaches

A search of the literature returned no examples on image based aluminium profile classification. The literature search was broadened to other applications that share similar characteristics with aluminium profile classification. Machine learning, a subfield of artificial intelligence, is commonly used for object detection and classification. Machine learning algorithms can be broadly divided into two group, supervised and unsupervised methods. Supervision in machine learning means that there are class labels attached to the data, where the true class is known before building our classification model (Singh *et al.*, 2016).

Deep learning is a form of neural networks based on the neuron connections in the human brain (Yegnanarayana, 2009). Similar to how humans constantly gather data through their sight, neural networks require constant feedback from their output. This process starts with training the network with as many images as possible. Preferably datasets have thousands, if not millions of images available to train the system. This requirement is not very achievable in this scope and therefore more classical methods will also be researched.

Examples of existing applications of supervised machine learning of images in the literature that are comparable to aluminium classification include disease detection in medical imaging (Çinarer and Emiroglu, 2019; Yassin *et al.*, 2017) and plants (Sharma *et al.*, 2020), classifying cracks in concrete (Kim *et al.*, 2019), and object recognition for

autonomous vehicles (Alam *et al.*, 2020). The processes proven to be effective in the literature provide a good starting point for this project and suggest which methods are worth investigating.

In an application for image classification of handwritten digits and facial recognition (Das, 2017), a k-nearest neighbour (KNN) classifier is compared against perceptron trained using back-propagation. It is found that while learning time for the back propagation method is long, the recognition time is short making it more suitable for real time recognition. With the large number of dimensions in the sample, back-propagation outperforms KNN in classification ability, irrespective of speed. The sets of images used from the online databases, MNIST[3] and ORL[4], contained 70,000 and 400 images, respectively. These images are across the 10 different digits and 40 different faces.

For classification of plant leaf diseases, Sharma *et al.* (2020) segmented the leaf from the background of 20,000 images and performed the machine learning algorithms logistic regression (LR), KNN, support vector machine (SVM) and a convolutional neural network (CNN). In this case, the CNN outperformed the other algorithms with a classification accuracy of 98%. Logistic Regression produced an accuracy of 66.4%, followed by KNN at 54.5%, and most poorly SVM with 53.4%.

To classify cracks and crack-like non-cracks in concrete, Kim *et al.* (2019) compares classification models SVM and Faster R-CNN. The features derived include image binarization, edge detection and morphological operations to create an accurate segmentation. The CNN method performed significantly higher compared to SVM based on the F1-score and accuracy.

Ongoing research for autonomous vehicles heavily involves object classification. Alam *et al.* (2020) compares decision trees (DT) to deep learning. The datasets are manually labelled with polygons as opposed to boxes to allow higher accuracy in pixel labelling. All experiments were carried out on an Aziz supercomputer. The analysis reveals that the two methods compared provide a similar accuracy, however deep learning computed 30% faster than the decision tree method.

Using a general purpose image database of 500 JPEG images for object classification, Kumar *et al.* (2012) analyses the performance of SVM, DT, and KNN classifiers. Based on the accuracy, precision and recall results, SVM is found to perform the best out of the three techniques.

In an effort to classify brain tumours from magnetic response image features, Çinarer and Emiroglu (2019) tested KNN, random forest (RF), SVM and linear discriminant

[3]http://yann.lecun.com/exdb/mnist/
[4]https://cam-orl.co.uk/facedatabase.html

analysis (LDA) machine learning algorithms. The SVM algorithm produced the highest classification accuracy of 90%. They used the REMBRANDT[5] dataset that contains a total of 660 images, with the images undergoing feature evaluation to triple the amount of data.

In an investigation of methods for crop discrimination, Peña-Barragán *et al.* (2014) evaluates decision trees (DT), logistic regression (LR), support vector machine (SVM) and multilayer perceptron (MLP) neural network methods for nine crop classes. Tests are carried out as single classifiers and as a combination in a hierarchical classification of nine summer crops in remote images. Independently, the MLP and SVM classifiers achieve an overall accuracy of 88%, with LR just below at 86% and DT at 79%. A combination of two SVM classifiers minorly increases accuracy to 89%. A ten-fold cross validation is performed to give confidence in these results.

An informative study by Lawrence and Moran (2015) performed statistically rigorous comparisons between six different machine learning methods for classification. The algorithms were tested across 30 different multispectral datasets, with consistency behind the procedures. Three out of the six methods outperformed the other methods by at least one metric on at least one of the datasets, indicating that no one method will suit all applications. They found that the method with the highest overall mean classification accuracy was RF at 73.19%, and was the best classifier for 18 out of the 30 datasets. Compared to RF, the overall mean accuracy for SVM was 9% lower at 62.28% and was never proven to be the best classifier across the 30 datasets.

A more recent algorithm for categorising images is isolation forests (Liu *et al.*, 2008). This method is based on random forests, however is unsupervised. It is commonly used for anomaly detection in hyper spectral images (Liu *et al.*, 2008). Li *et al.* (2019) state that iForest outperforms random forests, and particularly in large datasets, which is demonstrated in experimental results.

The applications of image classification described above have widely different applications, meaning the results are difficult to compare to each other. Support vector machines seem to be a favoured method to review in medical imaging (Yassin *et al.*, 2017), however in a more general sense RF has been proven to outperform SVM (Lawrence and Moran, 2015). In some classification applications, CNN has been proven to give the highest accuracy (Sharma *et al.*, 2020). These methods are examples of algorithms that may provide a solution to the classification problem presented.

Facial Recognition and Car Model Classification are two applications in the literature that are comparable to this project, in regard to identifying one type of 'object' that

[5]https://wiki.cancerimagingarchive.net/display/Public/REMBRANDT

has vast variation in shape. There are plenty of examples in the literature on these two applications.

Car model classification that encompasses every model requires hundreds of independent classes. The system by Llorca *et al.* (2014) uses Vehicle Manufacturer and Model Recognition (VMMR) models using rear view images of cars instead of the typically used front view. A Licence plate recognition (LPR) model provides a means of normalising the images. The approach involves learning the geometry and appearance of car emblems independently for each model. Using Histogram of Gradient (HoG) features, a set of binary discriminative SVM classifiers are applied to model the appearance of emblems for each car model (Llorca *et al.*, 2014). The probability for each model is carried out on an image, with the highest probability inferring the most probable model of the car. 28 different car models were used from 910 training images and the system achieved an overall accuracy of 93.75%.

An interesting research topic is car make and model recognition (CMMR) at night. The limited lighting conditions result in features missing from the images. Boonsim and Prakoonwit (2017) propose a method that deals with the missing features. Feature extraction is performed from the tail light and licence plates of the rear view of the car, as the size, distances, and angles between them are often unique between car models. SVM, DT and KNN are used as one class classification systems. A one class method is where there is a model for each car class, with the car class being the positives and every other car class being the negatives. The majority results of the three classifiers verifies the target model in the classification process (Boonsim and Prakoonwit, 2017). The overall accuracy achieved was 93%.

Many methods for CMMR rely on licence plate recognition. Hsieh *et al.* (2013) applied a symmetric scale invariant feature detector and descriptor (SURF) to a make and model recognition system to address the problems using normal SURF. Symmetrical points are found from a grid-based scheme and extracted to train various SVM algorithms, before integrating the learning algorithms into an ensemble classifier (Hsieh *et al.*, 2013). The average classification accuracy was 99.07%, without relying on licence plate recognition.

A large subject in computer vision is face recognition for security measures. To address the issues regarding recognition with pose and lighting variations, Julina and Sharmila (2017) evaluates classification using Histogram of Oriented Gradients (HOG) with SVMs. The HOG descriptor generates feature vectors from the images and comprise the training dataset for the SVMs. The image dataset includes a variety of poses and lighting conditions to verify the performance. The overall accuracy was 90.2%.

Chen and Jenkins (2017) perform a comparison of Principal Component Analysis (PCA), a popular feature selection method, with linear discriminant analysis, KNN, and SVM. For the facial recognition problem, SVM was found to have the highest accuracy and KNN had the fastest running speed.

Research for facial recognition for mobile phone use was carried out by Kremic and Subasi (2016) . Detection was performed based on skin colour and histograms prior to classification using SVM and RF. The SVM kernel was optimised and a classification accuracy of 97.94% was achieved, similar to the RF accuracy of 97.17%.

Each approach to solve a problem works well given the specific variation being studied (Tolba *et al.*, 2005). When another form of variation is present then the performance quickly degrades. Through reviewing facial recognition literature, Tolba *et al.* observes that there does not appear to be any feature, or any combination of features that is invariant to all variations an image of a face may present. It is common to need to recognise image features under different variations, such as lighting from the time of day and/or external light sources. Another observation is that techniques will generally perform better with more training images. This knowledge is transferable to other applications of image classification.

2.6 Image Segmentation

Segmentation is the delineation of objects in an image. With the option to produce binary images of the extrusion cross sections via. segmentation, the range of feasible classification methods broadens. Methods used for letter and number classification become applicable as the extrusion profiles are of similar nature in regard to shape. Hence a review is conducted to explore methods for accurate image segmentation. There are many uses and applications of image segmentation in computer vision. There are deep learning algorithms that can perform instance segmentation on colour images. The PASCAL VOC challenge 2011 saw FCNs implemented on the segmentation data set and achieve a pixel classification accuracy of 90.3% (Everingham *et al.*, 2010). The performance of any segmentation pipeline is highly dependent on the quality of the images (Mehul *et al.*, 2014). Ghosh *et al.* (2019) discusses building deep learning models for instance segmentation, addressing pros and cons for various convolutional algorithms.

Segmentation is a technique popular in hand-writing and text recognition methods. The ability to remove noise and normalise an image produces a clean image that is easier to binarize. Documents are written on a single colour sheet, with a separate colour used for the writing so that the words easily stand out. These factors are the perfect set up

for segmentation as the document is easily binarized based on the colour information. The ability to separate the objects in the image, in this example being letters and words, directly affects the recognition performance (Mehul *et al.*, 2014).

Another field that applies segmentation is medical imaging. Chowdhary and Acharjya (2020) discuss many techniques for segmentation in tumour detection, which assists medical professionals in diagnosis and treatment plans for patients. Thresholding, edge detection and a list of other techniques were developed many years ago but are still in use today. Recent techniques are artificial neural networks, clustering and segmentation using machine learning (Chowdhary and Acharjya, 2020). The author goes on to discuss some popular filtering techniques for feature extraction in medical imaging, such as local binary pattern, Gabor texture features, and dimension reduction techniques principal component analysis (PCA) and independent component analysis (ICA).

This short review will detail the more traditional filtering methods for segmenting objects from images, due to the known quality of the initial images. Adams *et al.* (2020) identifies issues similar in segmenting plants, to factors present in the extrusion images. Any reflective surfaces in the scene produce a mimic information about the target object. The extrusions themselves are reflective objects and add complexity to the problem. Neighbouring extrusions add reflections of the target extrusion, adding difficulty to segmentation.

A simple scenario is where the background in all acquired images are the same. Frame differencing is a popular method that allows the use of a reference image of the background, given the background never changes (Choudhury *et al.*, 2018). A simple subtraction between frames removes pixel information that doesn't relate to the target object. Thresholding methods across different colour spaces are also often considered in image segmentation problems. Thresholding involves setting all pixel intensities that are above a certain threshold as one class, and every pixel intensity below that threshold as another class, producing a binary image. The intensities come from the grayscale image.

Adams *et al.* also discusses the use of morphological operations, opening and closing, as post processing techniques to remove unwanted background noise. The scene is engineered to ensure that these methods are feasible solutions to the image segmentation problem, to obtain accurate pixel labels.

Supervised machine learning approaches linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), random forest, support vector machines (SVM) and neural networks (NN) were tested. A reduced dataset was used for training due to limited available RAM to carry out the task. The results showed that random forest and neural networks performed well compared to the thresholding method and the other machine

learning methods. Though the neural network was favoured due to the computational feasibility (Adams *et al.*, 2020).

The literature suggests that preparing the scene to allow the use of simple tools and filtering is a good start to determining how well objects can be segmented from images.

Chapter 3

Segmentation of Extrusion End Shapes

A significant problem in this project is accurate segmentation of extrusion ends from the images. A clear segmented image will expose the highly detailed extrusion end, and allow meaningful feature extraction to best classify the extrusion. This segmentation problem is a supervised classification problem, where the ground truth is found from semi-automated processing.

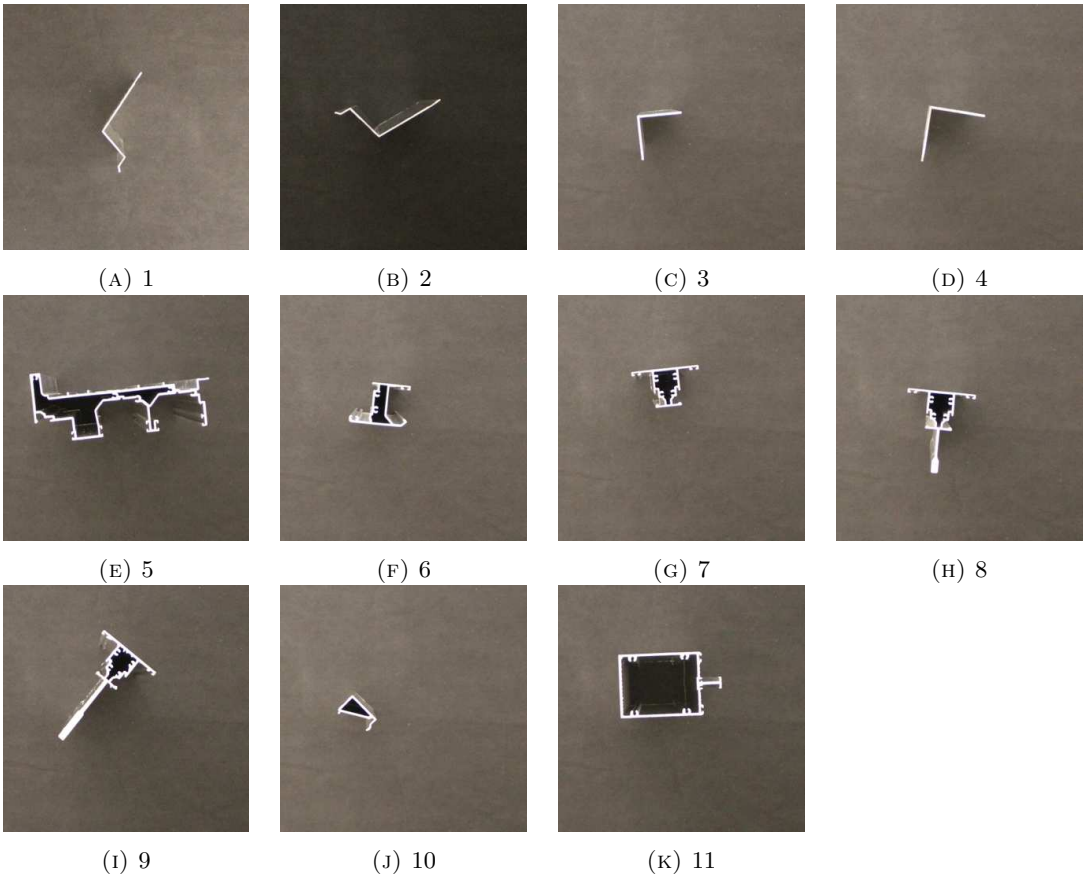
Discussed are factors relating to image quality and an investigation on three independent processing pipelines to segment extrusion ends. These pipelines all make use of image processing techniques and differ by the machine learning approaches. Firstly without a machine learning algorithm, secondly with a classical supervised machine learning algorithm, and thirdly using a deep learning algorithm.

3.1 Data Acquisition

To overcome lighting issues due to the warehouse environment, APL provided 11 different sample extrusion ends to acquire new data that were purposefully chosen to be either very different from each other or very similar (Figure 3.2).

The newly acquired data was taken in the optics lab at the University of Waikato. There are no windows in this room, therefore all lighting is artificial, (Figure 3.3). There are two primary light sources, one for each section in the room. To avoid saturation, the lighting directly above the scene was turned off and the main light source was the lights from the second section of the room. This setup allowed for enough lighting to see everything clearly without over saturating the image and losing pixel information. Having overhead

FIGURE 3.1: Ceiling and lighting above extrusions in the APL warehouse.

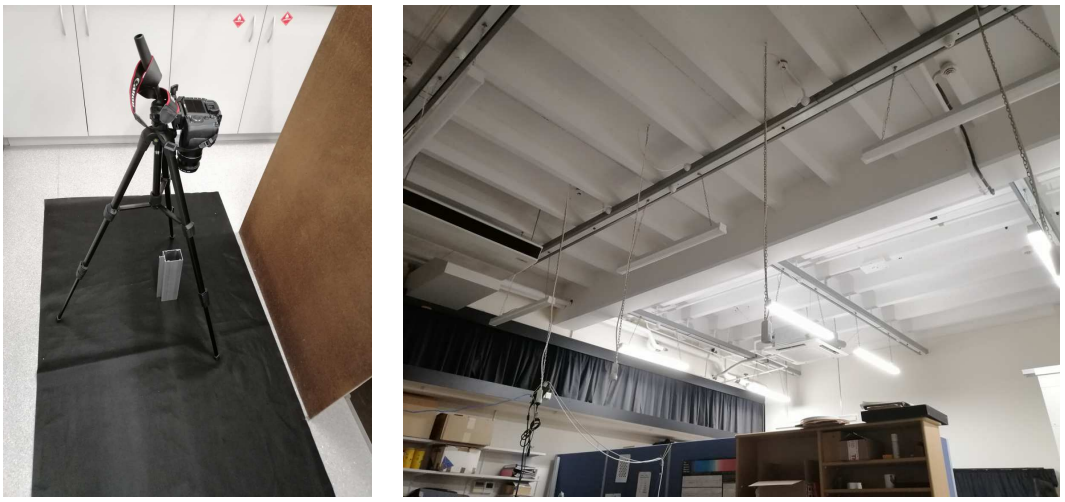


(A) 1 (B) 2 (C) 3 (D) 4

(E) 5 (F) 6 (G) 7 (H) 8

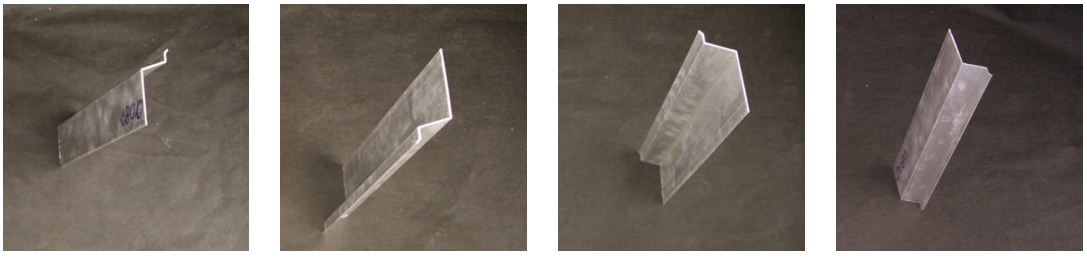
(I) 9 (J) 10 (K) 11

FIGURE 3.2: Each of the extrusions and their class number assigned for experiments.



(A) (B)

FIGURE 3.3: Camera setup and lighting for data acquisition.



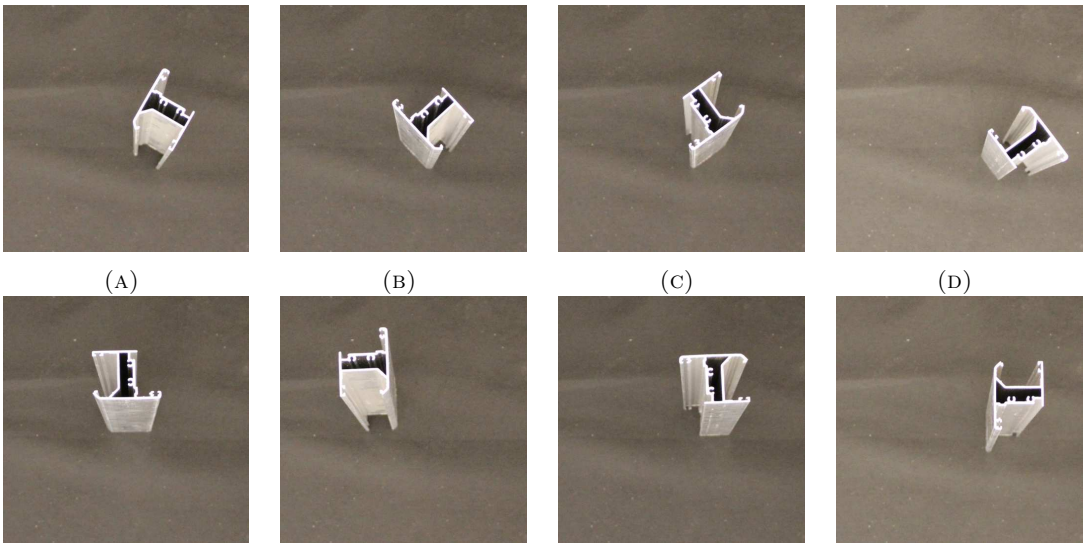
(A) flat (B) 10 degrees (C) 20 degrees (D) 30 degrees

FIGURE 3.4: Examples of the four angles captured for each extrusion profile, in the top right corner.

lighting allows the profile of the extrusion to receive the most light and be the brightest part of the image, hopefully simplifying the problem of segmentation.

In the initial images, the largest angle between the camera and the extrusions is determined to be 30 degrees. There are 4 angles chosen to acquire new images at, 0, 10, 20 and 30 degrees (Figure 3.4). Zero degrees is when the camera lens is pointing directly down, facing an upright extrusion. There are countless positions in the scene that extrusions are in. To simplify, an extrusion was placed in the middle of the image, and the top outermost left and right corners. The last factor to incorporate is the orientation. For simplicity, 8 orientations were chosen. Five out of the 11 extrusions are not symmetrical, and extrusions are not necessarily placed into the bays the same way up. At each end, 4 orientations are chosen, each rotated 90 degrees, with the placement of the opposite end shifted by 45 degrees allowing a difference for the symmetrical extrusions (Figure 3.5).

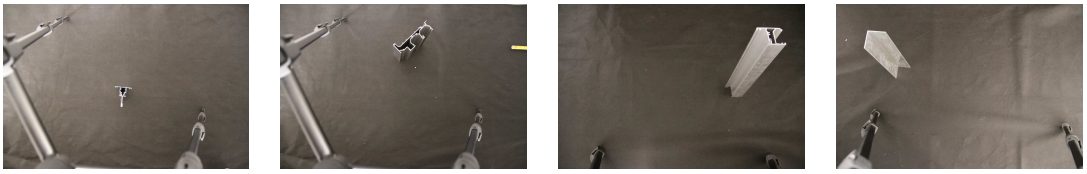
To begin with, all of the images at 0 degrees were taken first, including a set of images



(A) (B) (C) (D)

(E) (F) (G) (H)

FIGURE 3.5: Examples of the eight orientations captured for each extrusion profile, taken at an angle of 10 degrees.



(A) Centre (B) Middle Top (C) Right Corner (D) Left Corner

FIGURE 3.6: Examples of different positions the extrusions were captured in.

where the extrusion was placed in the middle at the top, Figure 3.6 (a) and (b). Following that, all positions and orientations for each angle were captured. Initially, the images in the top right corner (c) were augmented to produce images in the top left corner. However, early in the investigation this showed signs of over fitting the data, so those positions were then captured in the camera (d). The angle of the camera was measured using an accelerometer phone app, whereas the other parameters, position and orientation were approximated by eye, allowing some typical variation between images. The naming convention for the image sets begins with the angle of the camera in degrees (flat in which the lens points directly to the ground, 10 degrees, 20 degrees and 30 degrees), followed by where in the image scene the extrusion is (right corner, left corner, centre and middle top). Image file names end with an identification number for the set.

The camera settings are kept the same throughout each set of images taken. However, variation crept through with the lens slipping in between shoots, and these changes were unfortunately unnoticed during the process. Table 3.1 presents the different values from the camera associated for each set. The digital view finder was used to shoot the 20d_left_corner set, which unknowingly uses a different shutter speed and resulted in a

TABLE 3.1: Canon EOS 7D camera with canon EFS 18-135mm lens and ISO 3200, settings for each set of images in order of time taken.

Image set	F-stop	Exposure time (s)	Focal length (mm)	ISO
flat_centre	f/3.5	1/13	18	3200
flat_right_corner	f/3.5	1/3	18	3200
flat_middle_top	f/3.5	1/5	18	3200
10d_centre	f/3.5	0.6	18	3200
10d_right_corner	f/3.5	1/2	18	3200
20d_centre	f/5	1/2	18	3200
20d_right_corner	f/3.5	1/2	18	3200
30d_centre	f/3.5	1/3	18	3200
30d_right_corner	f/3.5	1/20	18	3200
flat_left_corner	f/3.5	1/2	18	3200
10d_left_corner	f/3.5	1/2	20	3200
20d_left_corner	f/3.5	1/2	20	3200
low_light_centre	f/3.5	1/30	20	1000
high_light_centre	f/3.5	1/30	20	2500

set of darker images. Though the data from the image did not present a difference in the setting numbers. During the last set of images in the left corner, the camera lens was taped to minimise changes, allowing the shutter speed and focal length to remain the same between sets on that day. Most camera settings can be viewed from the original unedited image dataset. Another two sets of images were taken, where the lights directly above the extrusions were turned on, and every other light turned off. One of the sets was taken in snapshot view, to add the same effect as was in the darker set, and the final set was taken through the optical view finder. The higher lighting adds some saturation to the profile ends, which adds some variation to the overall data set.

The first step for every raw image was to crop the image to 1500x1500, to only the area of the image that the extrusion is present in. Many machine learning algorithms and devices available are limited by memory. Therefore, all images were reduced to 500x500. These resultant images became the overall dataset.

Prior to testing any filters or algorithms, a gold standard set of segmented images were produced to be the reference set. These images became the reference for computing accuracy for any following segmentation. Most of the extrusion pixels were able to be segmented using a threshold, since the light was manipulated to give the profiles the most light, resulting extrusion pixels being the brightest in the images. Due to the minor differences between images, such as other things in the area moving and introducing shadows, slight changes in camera and extrusion positioning and angling, the same threshold cannot be used for all images in the dataset. Images within the same set

also require different thresholds. The reference set was created from manually adjusting the threshold as needed for each image, and manual filling in image editing software (MS paint) to join together parts of the profile. This manual method requires labour and time, giving motivation to find more automated methods. It was found during this process that the image set 30d_corner could not be remotely segmented using any simple filters, as the lightest part of the images aren't the extrusion ends. Hence, these images were excluded from the dataset used for investigating segmentation. To ensure that the results are always comparable, designated training/testing/validation sets were created. Both the test and validation set contain 110 images, 10 of each of the 11 classes. These sets were generated by choosing 10 random images within each class. These sets were configured in this way with the knowledge that machine learning algorithms perform best when the classes are balanced.

3.2 Data Processing Methods

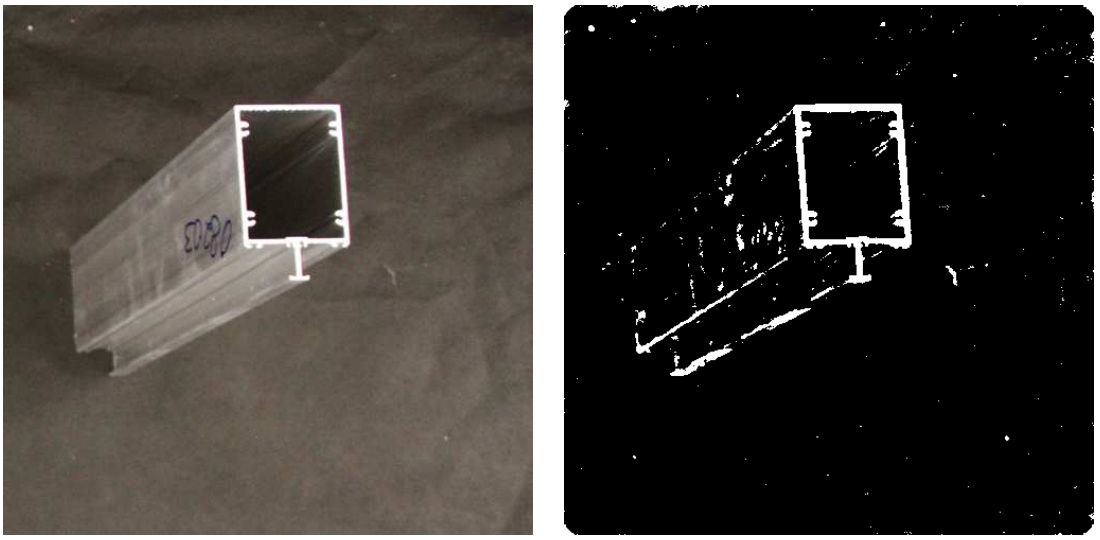
3.2.1 Approach 1: Image Processing Filters

The first important step is to convert the image from colour space to grayscale. The second step is termed shade correction, beginning with resizing the image to 75% of the original size. A median filter with kernel size of 15x15 is applied to the resized image. The filtered image is resized back to the original size. The process blurs the detail in the image leaving an estimate of the background shading. This filtered image is subtracted from the original image (Fig. 3.7), removing the background of the image and leaving pixel information primarily for the extrusion profile. The shade corrected image is still a grayscale image. A threshold is applied to binarise the shade corrected image.

It is common to find many random white pixels left in the binary image that do not contribute to the extrusion profile. To remove these, the final step is to apply an algorithm that finds all the shapes in the image (regions of white pixels) and removes all shapes except the largest one, the extrusion profile. The algorithm is called `bwpropfilt` on Matlab [1]. This is an important step as there can only be one object in the image to compute the generic Fourier descriptor for classification.

Unfortunately, any means of creating an adaptive threshold or adapting images to a threshold, drastically reduced the accuracy of the segmentation. Hence, these methods were not used in the final segmentation pipeline.

[1] https://au.mathworks.com/help/images/ref/bwpropfilt



(A) Original grayscale image. (B) Shade corrected and binarised image.

FIGURE 3.7: Shade correction performed on a grayscale image to obtain binary image.

3.2.2 Approach 2: Classical Machine Learning

Now that the perfect reference images exist, every image in the dataset has a corresponding image of pixel labels, exactly what is needed for a supervised machine learning algorithm. For every image in the training set, the pipeline computes a set of filtered images with the same mask image. Each filtered image makes up a feature of that image. The features list contains the following: the original grayscale pixel values, 32 Gabor filters, the shade corrected image, Gaussian filter with $\sigma = 3$, Gaussian filter with $\sigma = 7$, median filter, and edge detection descriptors Canny, Roberts, Sobel, Scharr and Prewitt. Initial testing showed that out of the available machine learning algorithms in Sci-kit learn[2], the random forest classifier produced the highest accuracy when trained as a segmentation classifier. Hence, all testing of training was carried out using the random forest classification algorithm.

Ideally, given the set of 1077 images, approximately 700 of these images would make up the training set. However, that would require the computer to have loaded in 700 images and each filter set in memory at the same time, which is well beyond the memory capacity. Scikit-learn does not support partial learning for the random forest algorithm. However given the opportunity and resources, training with the full set should be investigated in the future to fully explore this path.

Initially, the random forest algorithm was trained singularly on five images from different image sets. Following on, the algorithm was trained on all 5 images, and then 11 images, one of each class, including the previous 5 images. Lastly, the algorithm was trained on

[2]https://scikit-learn.org/stable/

an additional 11 images, totalling 22 images, with 2 images from each class. The images for the training set were chosen using a random number generator until one image from each class was selected to ensure there was variety in the training set. A randomly selected image was skipped if there were already 2 images from the same image set in the training data, to also ensure variety in the image scene. All models were predicted on the designated test set to validate the accuracy.

The output from the random forest classifier is not suitable for computing the generic Fourier descriptor from as there can only be one object in a binary image. Applied to the images is the morphological closing operation, which performs a dilation and erosion using the same structuring element, to remove any small holes and to allow the chance for two very close objects to connect.

3.2.3 Approach 3: Mask R-CNN

The input format required for the Mask R-CNN algorithm is a separate file for annotations[3]. This file contains the information on where the images are located, a polygon describing the pixel locations of each object in each image, and the class that each annotation is. Each polygon has identification numbers for which image it is in and which annotation it is, as in object classification tasks there is the possibility of multiple annotations per image. In this case, there is only one object per image so the image ID and annotation ID are the same. The polygon is generated from the reference binary image. A polygon is calculated based on the perimeter of the object, regardless of whether the extrusion is hollow or not. 967 images are input as training images with the remaining 110 images left in reserve for post-training testing.

3.3 Results and Discussion

To evaluate the difference in performance between the trained models, the average F1-Score is compared.

Training the segmentation of extrusion ends using a singular image performs well given that all the images have roughly the same brightness. Training using any image that isn't from the darker set (20d_left_corner), does not segment any image from the darker set at all. And vice versa, training using an image from the darker set will not segment extrusions from any other set of images. Given that illumination on the extrusions

[3]https://www.immersivelimit.com/tutorials/create-coco-annotations-from-scratch Date accessed 07/10/2021

TABLE 3.2: Variation between random forest models trained with different number of images.

	1 image	5 images	11 images	22 images
Average F1-Score	0.87	0.92	0.92	0.55
F1-Score Standard Deviation	0.25	0.12	0.13	0.17

TABLE 3.3: T-Tests between the testing results from random forest training with different number of images.

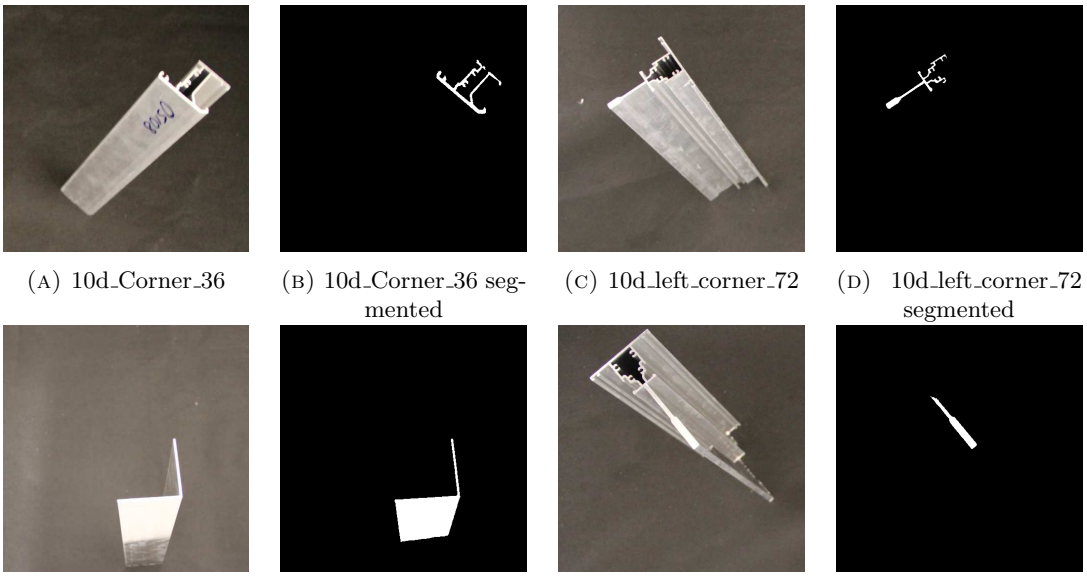
1 vs 5	1 vs 11	5 vs 11
0.065	0.060	0.96

change throughout the day due to sunlight, it is desirable that the model can handle changes in brightness between images.

Training the pixel classifier using a singular image shows the largest standard deviation due to the darker set not segmenting at all (Table 3.2). Whereas the pixel classifier trained with 5 or 11 images both were able to perform segmentations on the darker set. T-tests between the training of 1vs5 and 1vs11 show that there is very close to being a significant difference between them (Tables 3.3).

Training using five or eleven images produced much the same results as each other, with no significant difference based on t-testing, and an improvement compared to training using a singular image. Between image sets, including the darker set, there is no significant difference in the f1-score of the segmentations. The last model, that was trained on 22 images, performed the worst with the lowest average f1-score. The reason is unclear, it is expected that using more training data will increase the model's performance, however these results suggest otherwise. Training was also quickly carried out with a mix of other images and varying numbers of images but all achieved the same poor results. A t-test between the 22 image and any other number of image results output a t-value of zero, confirming training the pixel classifier using 22 images produced statistically the worst model.

The pipeline follows on using the model trained with 11 images. In the random forest pipeline, images where segmentation performed poorly did so for obvious reasons. For example, in Figure 3.8 (a) demonstrates a blurry image, where even to the human eye the pixels that are the extrusion aren't clear. (c) many of the pixel intensities are not distinguishably different on the profile compared to the body of the extrusion, due to uneven illuminations on the aluminium. (e) the side of the extrusion is reflecting the



(A) 10d_Corner_36 (B) 10d_Corner_36 segmented (C) 10d_left_corner_72 (D) 10d_left_corner_72 segmented

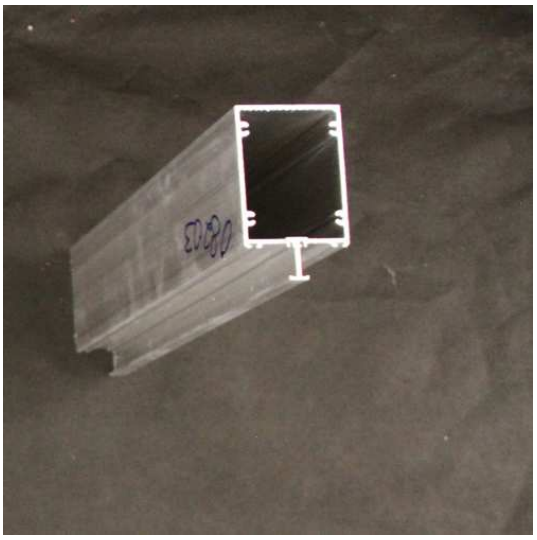
(E) 30d_Middle_24 (F) 30d_Middle_24 segmented (G) Flat_left_corner_78 (H) Flat_left_corner_78 segmented

FIGURE 3.8: Examples of images that did not segment well from the random forest pixel classifier.

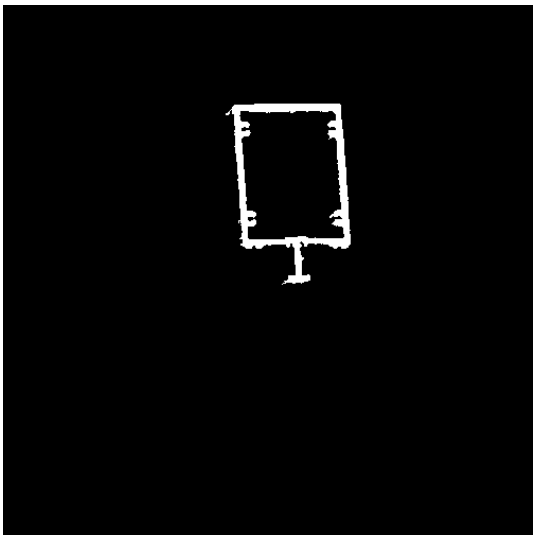
linoleum floor (the floor is out of frame), giving it the same brightness as the profile pixels. (g) there is a clear gradient of illumination across the profile and some parts are even saturated. These failure cases highlight limitations for the classification system. Based on the resultant segmented image from the machine learning pipeline, the most significant factor in achieving accurate image segmentation is even illumination on the profile end. The pathway to improve robustness is to find ways to ensure evenness of illumination on the profile ends.

The algorithm handles lighting variation from image to image, and issues in the segmentation only arise due to uneven illumination or the failure cases described.

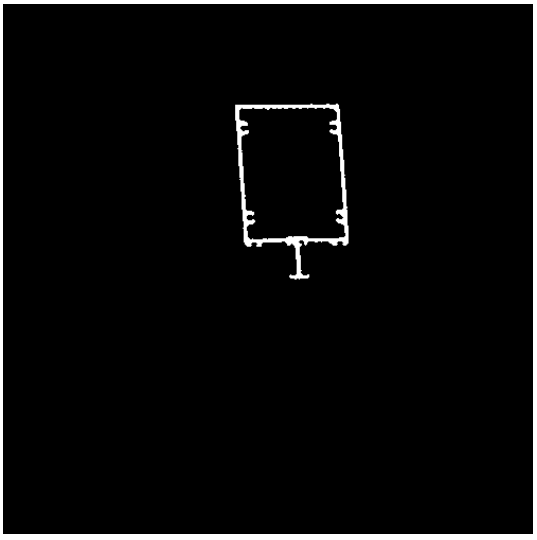
A main problem with the Mask R-CNN approach is the configuration of the annotations. Since the annotations have to be solid, the input masks include the dark pixels that make up the hollow parts of the extrusion profiles. In some of the extrusions, particularly extrusion 11, there are more pixels of the hollow middle than there are of the actual profile. Therefore, there are many dark pixels in the hollow areas that have been labelled as the extrusion profile. The algorithm learns that there are both very light and dark pixels that create a profile, causing great difficulty in accurately segmenting an extrusion. Extrusion 11 (Figure 3.9) has the largest area of dark pixels in the centre due to its square shape so the segmentation is generally the closest to its reference polygon annotation. The protruding piece of the profile is however never a part of the resultant segmentation.



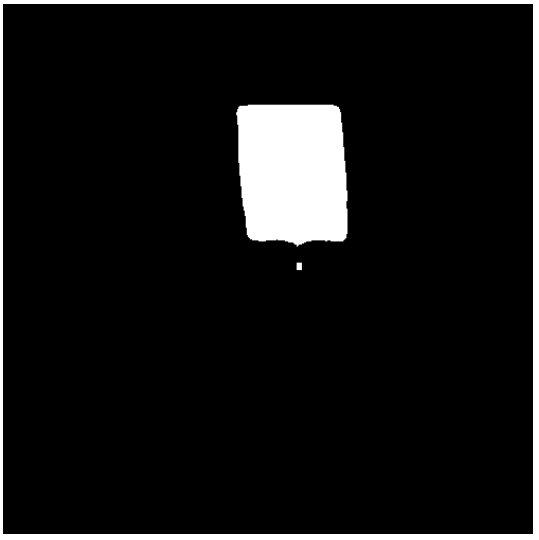
(A) Original grayscale image.



(B) Segmentation using image processing filters, f1-score of 0.77.



(C) Segmentation using random forest machine learning algorithm, f1-score of 0.95.

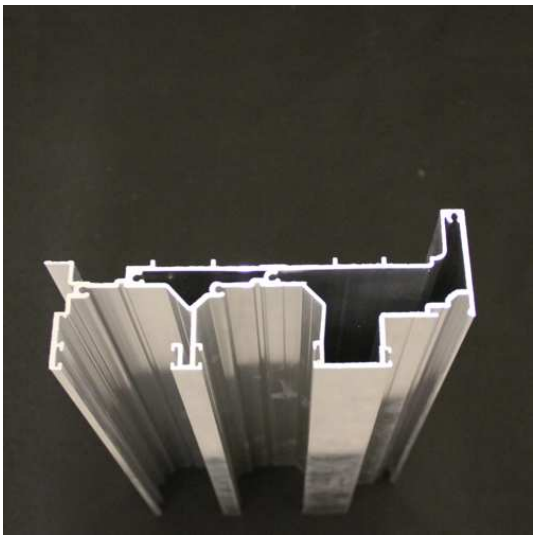


(D) Segmentation from Mask R-CNN, f1-score of 0.23.

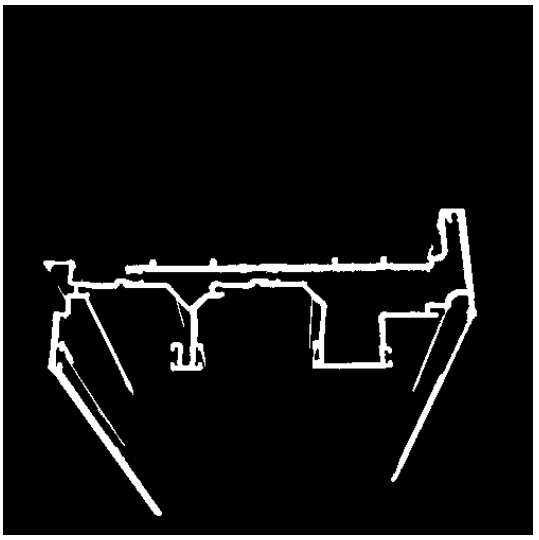
FIGURE 3.9: Segmented output images from all methods on image flat_corner_83

Since the algorithm learns that the centre of extrusions are dark, any extrusion that does not 'close' and therefore doesn't have a dark centre is not identified in the image. Without an object being identified there is no segmentation to create. Extrusions 1-4 do not close and were not segmented in any of the images. Extrusions 8 and 9 are variations of extrusion 7. Similar to how extrusions 1-4 were unable to be segmented, the long pieces on extrusions 8 and 9 are poorly segmented, where the segmentation appears as two separate objects.

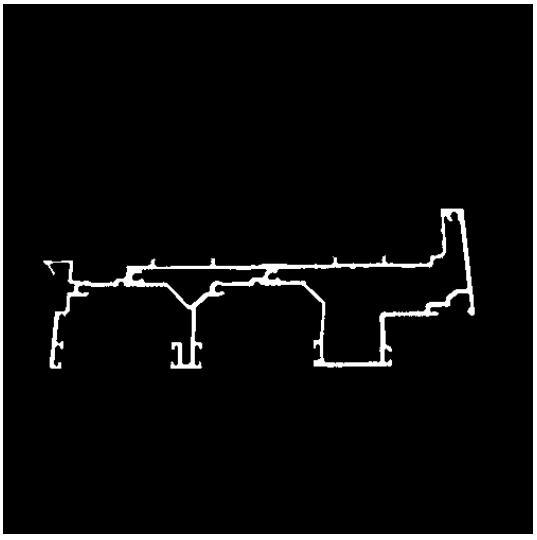
Since annotations cannot be coded to be hollow, these results show that Mask R-CNN is the poorest at segmenting extrusions from images out of the three methods investigated. However, with consideration of the length of time used to implement a working Mask



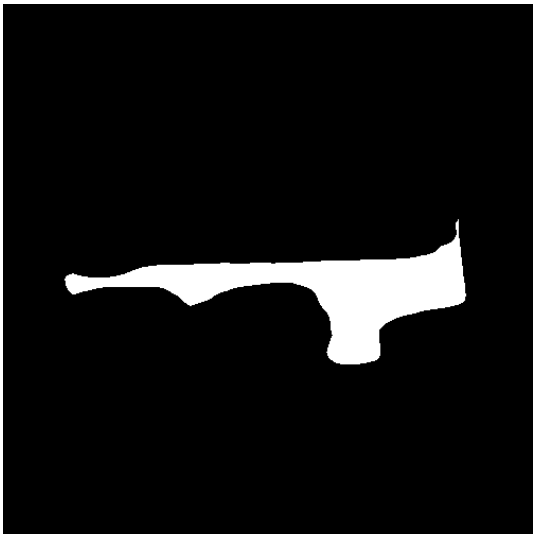
(A) Original grayscale image.



(B) Segmentation using image processing filters, f1-score of 0.67.



(C) Segmentation using random forest machine learning algorithm, f1-score of 0.96.



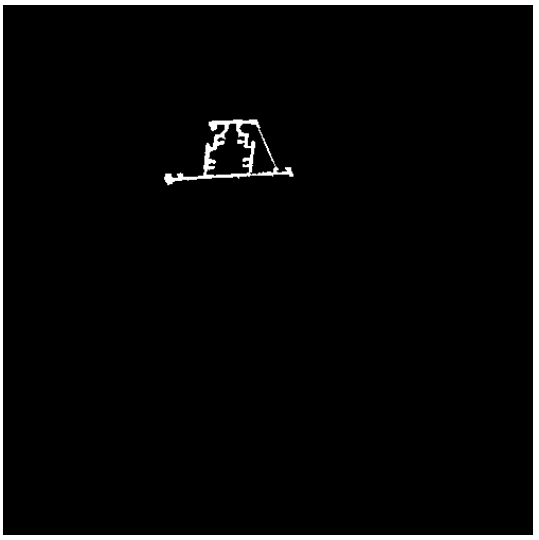
(D) Segmentation from Mask R-CNN, f1-score of 0.28.

FIGURE 3.10: Segmented output images from all methods on image 30d_middle_31

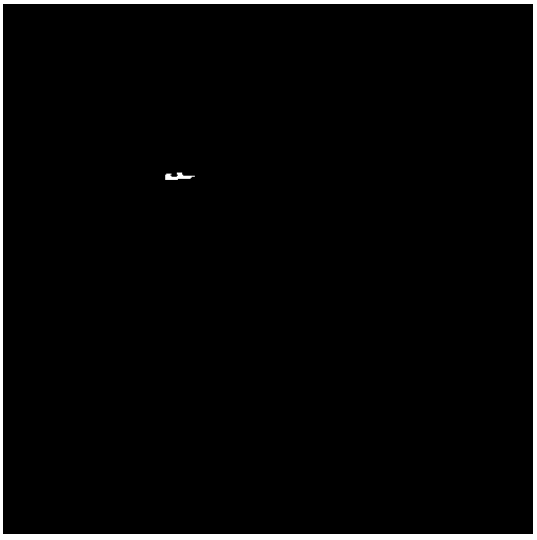
R-CNN, there was little time to adjust the parameters or implement validation during training to optimise results. Further time and improvement to the training system would benefit the investigation.

Figures 3.9, 3.10, 3.11, 3.12, compare the segmentations between the three methods on four different cases. Figure 3.9 is a perfect example of the annotation style from the Mask R-CNN approach, where based on the reference image many false positives were predicted. Both approaches 1 and 2 present good segmentations, where visually the extrusions look identical, with approach 2 producing the highest f1-score. Figure 3.10 presents a common issue with approach 1 where pixels of the side of the extrusion as well as the profile are segmented due to their high brightness. Again approach 2 produces

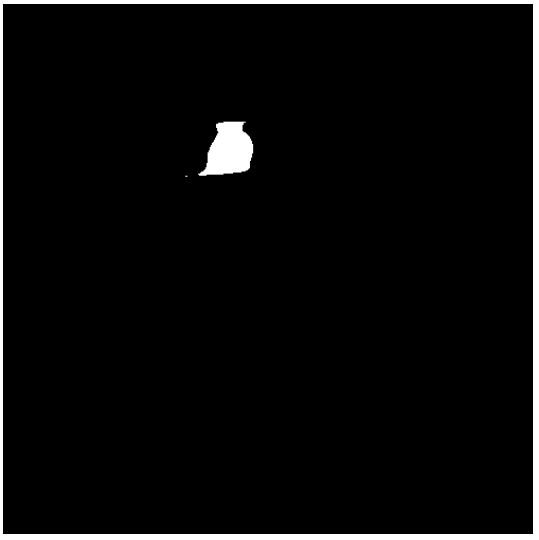
(A) Original grayscale image.



(B) Segmentation using image processing filters, f1-score of 0.85.



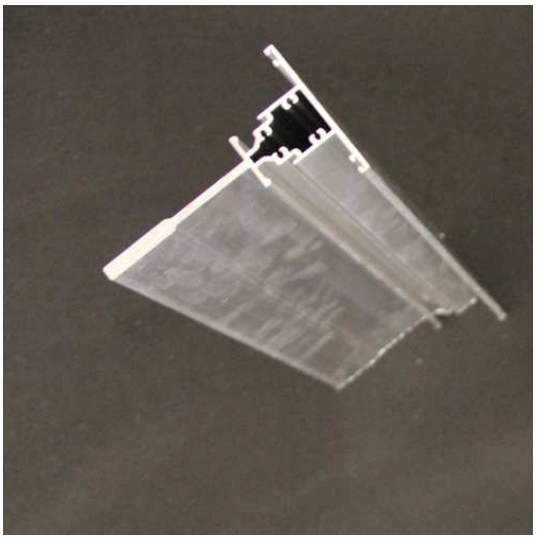
(C) Segmentation using random forest machine learning algorithm, f1-score of 0.27.



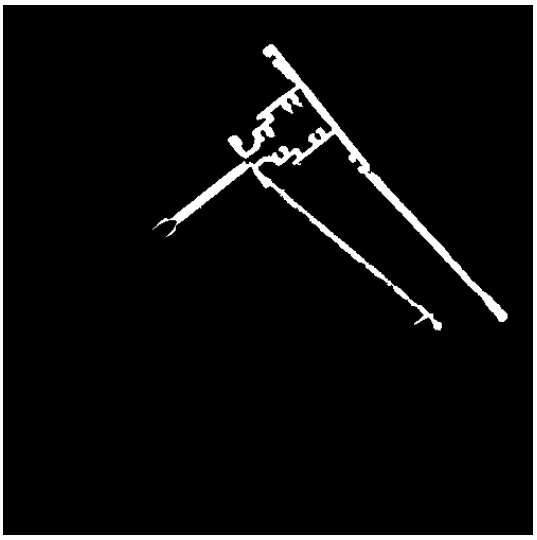
(D) Segmentation from Mask R-CNN, f1-score of 0.28.

FIGURE 3.11: Segmented output images from all methods on image 20d_left_corner_44

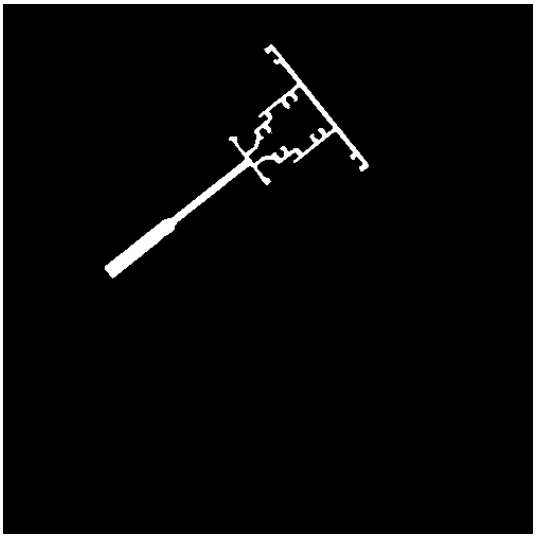
the highest f1-score. The extrusion in Figure 3.11 has uneven illumination on the profile. The effect can be minimised in approach 1 due to the shade correction step allowing a meaningful segmentation, which produced the highest f1-score. However approach 2 did not overcome this, resulting in very few pixels successfully segmented. Figure 3.12 shows similar results to Figure 3.10, except approach 1 where the end portion was missed. It is unclear as to why this part was not segmented as the pixels are the same brightness as pixels that were segmented correctly. Again here, the highest f1-score was produced by approach 2. The average results across the entire validation set are displayed in Table 3.4, and suggest that the best approach is the random forest model due to the consistent high values between the 3 methods and producing the highest f1-score. The predicted image results also suggest that on average, approach 2 performs the most



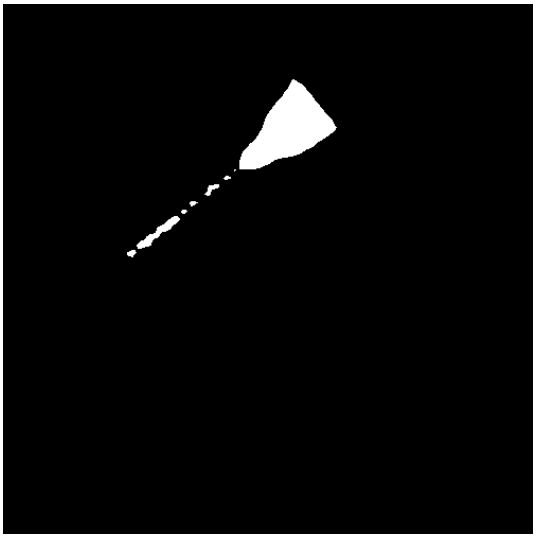
(A) Original grayscale image.



(B) Segmentation using image processing filters, f1-score of 0.59.



(C) Segmentation using random forest machine learning algorithm, f1-score of 0.90.



(D) Segmentation from Mask R-CNN, f1-score of 0.40.

FIGURE 3.12: Segmented output images from all methods on image flat_left_corner_81

accurate segmentation. While this method produces some flawed segmentations like the other two methods, the problems that arose can be overcome by more careful data acquisition. Whereas the results of Mask R-CNN show that the algorithm is not suitable for the type of segmentation desired. T-testing between the f1-score sets between the 3 approaches all produce a t-value of zero, further indicating that the difference between the f1-scores results are significant, allowing claim that approach 2 performed the best.

TABLE 3.4: Average of results from segmented images from the three main pipelines.

Metric	Traditional	Random Forest	Mask R-CNN
Sensitivity	0.91	0.97	0.36
Specificity	0.99	0.99	0.98
Precision	0.54	0.94	0.18
F1-Score	0.65	0.92	0.23

TABLE 3.5: Standard deviation of results from segmented images from the three main pipelines.

Metric	Traditional	Random Forest	Mask R-CNN
Sensitivity	0.17	0.12	0.29
Specificity	0.0053	0.0041	0.027
Precision	0.14	0.14	0.16
F1-Score	0.13	0.14	0.19

3.4 Classification using Fourier Descriptors on Segmented Extrusions

The generic Fourier descriptor (GFD) algorithm requires that the input images have only one object. To be sure, `bwpropfilt` is applied to every image prior to feature extraction to remove all objects except the largest. In some cases this can mean that extrusion pixel information is lost. This step can be improved given more time to carry out further research. The generic Fourier descriptor is computed for the full binary training set of 967 reference images. Three radial frequencies and twelve angular frequencies were computed over the image. Giving a Fourier descriptor length of 52 (R+1)(T+1). The 52 descriptors make up the feature set for training classifiers in Weka. Many classifiers are available in Weka, hence a variety was tried to gauge accuracy achieved across different types of algorithms. The decision for choosing classification algorithms to test is based on what is prominent in the literature for multi-class problems

A range of classifiers were tried (Table 3.6) and a few were supplied with a test set to calculate the validation accuracy to confirm the training results. The model that produced the highest accuracy and f-measure results was Logistic Regression.

TABLE 3.6: Results for 10-fold cross validation of various GFD trained models using Weka.

Model	F-measure	Accuracy
SVM	0.81	82
Naive Bayes	0.87	87
Local KNN	0.89	89
Optimized Forest	0.91	91
Random Forest	0.91	91
Cost-Sensitive Forest	0.91	91
Logistic Regression	0.92	92

TABLE 3.7: Confusion matrix for cross validation of a GFD trained logistic regression model.

Classified as	1	2	3	4	5	6	7	8	9	10	11
1	64	23	0	0	0	0	0	0	0	0	0
2	30	58	1	0	0	0	0	0	0	0	0
3	0	2	60	26	0	0	0	0	0	0	0
4	0	1	38	49	0	0	0	0	0	0	0
5	0	0	0	0	88	0	0	0	0	0	0
6	0	0	0	0	0	88	0	0	0	0	0
7	0	0	0	0	0	0	88	0	0	0	0
8	0	0	0	0	0	0	1	87	0	0	0
9	0	0	0	0	0	0	0	1	87	0	0
10	0	0	0	0	0	0	0	0	0	88	0
11	0	0	0	0	0	0	0	0	0	0	88

Table 3.6 presents an accuracy of 92% and an f-measure of 0.92 for the logistic regression classifier.

TABLE 3.8: Confusion matrix for post training testing (approach 2) on the GFD trained logistic regression model.

Classified as	1	2	3	4	5	6	7	8	9	10	11
1	8	1	0	1	0	0	0	0	0	0	0
2	2	8	0	0	0	0	0	0	0	0	0
3	0	0	9	0	0	0	0	0	0	0	1
4	0	0	1	8	0	0	0	0	0	0	1
5	1	0	0	0	9	0	0	0	0	0	0
6	0	0	0	0	0	9	1	0	0	0	0
7	0	0	0	0	0	1	9	0	0	0	0
8	0	0	0	0	0	0	0	9	1	0	0
9	0	0	0	0	0	0	0	1	9	0	0
10	0	0	0	0	0	0	0	0	0	10	0
11	0	0	0	0	0	0	0	0	0	1	9

The total correct predictions in Table 3.8, 97, over the total number of predictions, 110, indicates the accuracy of the model on the testing data, 88%. Among the images that were classified incorrectly as a completely different extrusion and not a remotely similar one, all the extrusions had been poorly segmented. Extrusions 1 and 2 and extrusions 3 and 4 were expected to be misclassified as each other as they are very close to being the same extrusion, only a difference of millimetre thickness. However, 80% accuracy within each of the four extrusions is a great performance compared to the other algorithms.

In the cases of extrusions 5,7 and 11, the one image in each of these classes that were misclassified show uneven illumination on the profile end in the original images, reiterating that uneven illumination is the largest contributor to failure cases. These 3 images were however all from the darker set of images, suggesting that the level of illumination was not high enough to produce even light cover. The resulting recommendation is to provide sufficient light that the profile end is fully lit, but not enough that there is over saturation or reflections in the images. The level of illumination would be further investigated given knowledge of where exactly the system will be implemented and the conditions of the environment.

Chapter 4

Classification and Object Detection of Extrusion End Shapes

Direct extrusion classification from colour images is a very useful tool for inventory management. Extrusion identification with high accuracy is difficult even for an experienced user. Automating this process to a high accuracy would achieve more reliability as it reduces human error. The primary aim is to determine how well different and similar extrusions can be classified and what factors affect the classification performance.

Discussed herein are the various machine learning approaches for direct classification from the original colour image dataset. There are three main approaches in this investigation with feature extraction for each approach. Firstly, classical supervised methods with both multi-class and pooled binary classifiers, secondly with a variety of pre-trained convolutional neural network classifiers, and thirdly using deep learning object detection models. Feature extraction gives classical machine learning algorithms the information to best separate the classes by, whereas convolutional neural networks perform feature extraction within the algorithm and do not require pre-processing.

4.1 Extrusion Classification Methods

4.1.1 Classical Machine Learning

Implemented in Weka are a set of 10 image filters. Each of the filters were applied to the image training set. With the new features, the data were classified using the

default random forest algorithm. This output was used to gauge the effectiveness for each image filter, and also a combination of the filters. Out of all the image filters, the correlogram features allowed the highest classification performance as a solo descriptor. Additional filters were added where any filter that cause the classification accuracy to further increase was kept as a feature. Through this testing, it was found that the combination of the following features: auto-colour correlogram, simple color histogram, color layout filter, fuzzy opponent histogram, edge histogram, and Gabor filters allow the best performance for the random forest classifier. This conclusion is based on the summary of results in Weka, which showed the highest accuracy, f-measure score and desirable confusion matrices. This exploratory work is used to determine appropriate features for image classification.

4.1.1.1 Multi-Class

Now that the appropriate input data has been determined, several classifiers were compared against each other. Classifiers were tested that were able to handle multi-class data. With the best classifier determined, the model was tested on the test set and a confusion matrix was computed to ensure there was minimal over fitting.

4.1.1.2 One vs Rest

One vs. rest is a method where we set one extrusion at a time as the positive class, and every other extrusion the negative class. This method allows testing of binary classifiers on a multi-class problem. Initially, the ADTree model is trained and the resultant tree shows that the classifier only used 9 to 11 features out of the hundreds derived from the image filters. The feature reduction allows identification of the features that contribute the most meaningful information for classification. The reduced features from all 11 binary ADTree models were kept and the rest discarded. Majority of the kept features were from the correlogram filter. The correlogram produces the largest feature vector, and also produced the highest accuracy when solely used during training out of all the image filters available. With the reduced features and based on the results from the multi-class problem, SVM models were trained.

4.1.2 Transfer Learning in Matlab

In this project there are too few data samples to train a convolutional neural network (CNN) algorithm from scratch. Transfer learning is the process of taking a pre-trained algorithm and reusing the model for a different task, while only partially training the

network. Pre-trained models have already learnt how to classify objects from obvious features such as edges and shapes.

To perform transfer learning, the first step was to acquire a pre-trained convolutional neural network. The final convolutional layer was replaced with an untrained layer. The pre-trained model provides initial weights for training the new model. The model was then trained on the extrusion images, with the pre-trained layers having a low learning rate and the new layer having a higher learning rate. During training the weights are updated to fine-tune the model to the task.

The following training options were set for AlexNet, GoogLeNet, ResNet-50 and VGG-19 networks: initial learning rate of 0.0001, validation frequency of 20, maximum number of epochs of 50 and mini batch size of 16. 50 epochs was chosen to allow the network sufficient time to fully train the network. In every epoch the network works through the entire training dataset. The mini batch size parameter defines the number of samples that the networks trains on in an iteration before updating the weights.

It is easy to overfit deep learning models. Over training a deep learning algorithm creates a model that fits to the training data so well that it poorly classifies data it wasn't trained on. The training data is split 70/30 to produce a validation set to evaluate the model while it is training. To avoid over-fitting, the validation curves during training were observed to gauge where the training and validation accuracies diverge from each other. The divergence is the generalisation error, and minimal error indicates a smaller likelihood that overfitting is occurring. A common stopping point is when the validation accuracy decreases yet the training accuracy is still increasing. The performance can vary a lot during training, meaning that the first instance of the validation accuracy decreasing is not necessarily the stopping point.

Data augmentation is considered a useful tool that creates new data based on the existing data to increase the amount of training data. Theoretically, the more data to train an algorithm with, the better the classification performance. Deep learning networks however have millions of parameters to train and therefore require a significant training dataset. To be consistent with the previous steps and produce results that can be compared fairly, the CNNs were first trained without data augmentation and then again with data augmentation. The augmentation features used were: random reflection about the horizontal and vertical axes, random rotation of 90 degree in both positive and negative directions, and random rescaling ranging from the original size to 200%.

4.1.3 Object Detection Neural Networks with Detectron2

Detectron2 is a Python object detection library, built by Facebook AI Research to support computer vision research. It provides detection and segmentation algorithms. Detectron2 was used to test both Mask R-CNN and Faster R-CNN.

4.1.3.1 Mask R-CNN

This section carries on from segmentation using Mask R-CNN in Chapter 3.2.3. It is the same method and experiment described in the previous chapter to achieve segmentation. There is a second output of the network that shows the bounding box around the object detected in the image and also the classification score as a percentage.

4.1.3.2 Faster R-CNN

Mask R-CNN is derived from Faster R-CNN. The arrangement of input data and code for training Faster R-CNN is the same for Mask R-CNN. The parameters were not changed between the two detection models, though the only output expected is the detected objects with bounding boxes and confidence scores.

4.2 Results and Discussion

Extrusions that are considered similar are extrusions that contain an almost identical region of the profile. For example, extrusions 1 and 2, and 3 and 4 are nearly exactly the same pairs of extrusions, with a difference of merely millimetres. Extrusions 7, 8 and 9 are considered similar as a region of extrusions 8 and 9 are identical to extrusion 7. Every other extrusion in this project is considered different as they are unique.

4.2.1 Classical Machine Learning

TABLE 4.1: Results for 10-fold cross validation of various image filter trained models using Weka.

Model	F-measure	Accuracy (%)
Cost-Sensitive Forest [1]	0.633	63.81
Optimized Forest [2]	0.652	65.67
Random Forest[3]	0.657	66.18
Logistic Regression [4]	0.691	69.18
Support Vector Machine [5]	0.780	77.77

TABLE 4.2: Confusion matrix for 10-fold cross validation trained SVM model.

Classified as	1	2	3	4	5	6	7	8	9	10	11
1	52	22	3	8	0	0	1	0	0	1	0
2	19	53	6	7	0	0	1	0	0	1	1
3	4	5	61	13	0	4	0	0	0	1	0
4	7	11	12	58	0	0	0	0	0	0	0
5	0	1	0	0	82	0	1	2	0	0	2
6	0	1	2	1	0	66	12	3	0	3	0
7	1	0	0	0	0	9	68	5	0	4	1
8	0	1	0	0	1	3	5	69	7	0	2
9	0	1	0	0	1	1	0	6	79	0	0
10	3	0	2	0	0	0	3	0	0	80	0
11	0	1	0	0	0	0	0	3	0	0	84

Based on the results in Table 4.1, there is an almost 10% difference between the top two performing classifiers. SVM produced the highest F1-score and accuracy results, because there is a good margin of separation between the data of extrusions 5-11. The features extracted for extrusions 1-4 are very similar so there is not a clear margin between the classes, and they are misclassified as each other more than other extrusions are misclassified. Tables 4.2 and 4.3 show that the majority of each class is classified correctly. The

[1] https://github.com/zislam/CSForest/
[2] https://github.com/zislam/OptimizedForest/
[3] https://link.springer.com/article/10.1023/A:1010933404324
[4] http://liblinear.bwaldvogel.de/
[5] https://www.csie.ntu.edu.tw/ cjlin/libsvm/

tables also show that a lot of the extrusions were misclassified as completely different extrusions.

TABLE 4.3: Confusion matrix for post training testing on the weka filter trained SVM model, with 0.48 f-measure and 75% accuracy.

Classified as	1	2	3	4	5	6	7	8	9	10	11
1	6	2	1	1	0	1	0	0	0	0	0
2	0	8	0	1	0	1	0	0	0	0	0
3	0	0	6	3	0	1	0	0	0	0	0
4	1	1	4	4	0	0	0	0	0	0	0
5	0	0	0	0	9	0	0	1	0	0	0
6	0	0	0	0	0	10	0	0	0	0	0
7	0	0	0	1	0	1	8	0	0	0	0
8	0	0	0	1	0	0	1	8	0	0	0
9	0	0	0	1	0	0	2	1	6	0	0
10	0	0	0	0	0	0	1	0	0	9	0
11	0	0	0	0	1	1	0	0	0	0	8

The results from the test set (Table 4.3) align with the results from training, with a drop in the f-measure and accuracy of roughly 3%.

TABLE 4.4: Confusion matrix of most probable extrusion given all 11 validations from the one versus rest models, with 35% accuracy.

Classified as	1	2	3	4	5	6	7	8	9	10	11
1	2	0	2	0	0	1	0	0	0	0	0
2	2	0	0	0	0	1	0	0	0	0	0
3	0	0	2	0	0	1	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0	1
5	0	0	0	0	9	0	0	0	0	0	0
6	0	0	0	0	0	2	0	0	0	0	0
7	0	0	0	0	0	0	2	0	0	1	1
8	0	0	0	0	0	0	0	3	3	0	0
9	0	0	0	0	0	0	0	2	6	0	0
10	0	0	0	0	0	0	0	1	0	10	0
11	0	0	0	0	0	0	0	1	0	0	7

The binary classifier models classify an instance as the extrusion the model was fit to, or classify that the instance is not that extrusion. However, an instance can be classified as negative by all of the models, leaving an unclassified instance. Hence, not all of the rows in the confusion matrix (Table 4.4) add to 10.

Due to the datasets for each binary model being largely negative samples and the majority of the negative samples classified correctly, the accuracy for each model is very high. So even though the validation accuracy for each binary model is above 90%, this isn't an accurate indicator of how well the model performs, as the training dataset is heavily unbalanced.

For example, let's consider the first model of extrusion 1. Considering extrusion 1 and 2 are almost identical, the first model has feature vectors for both positive and negative data points that have extremely similar features. Hence, majority of extrusions 1 and 2 were classified negative by every model. This problem affects most of the binary models, leaving only several instances that were predicted correctly and producing very poor results. The accuracy from this method is already significantly lower than the multi-class methods and so will not be recommended.

4.2.2 Transfer Learning

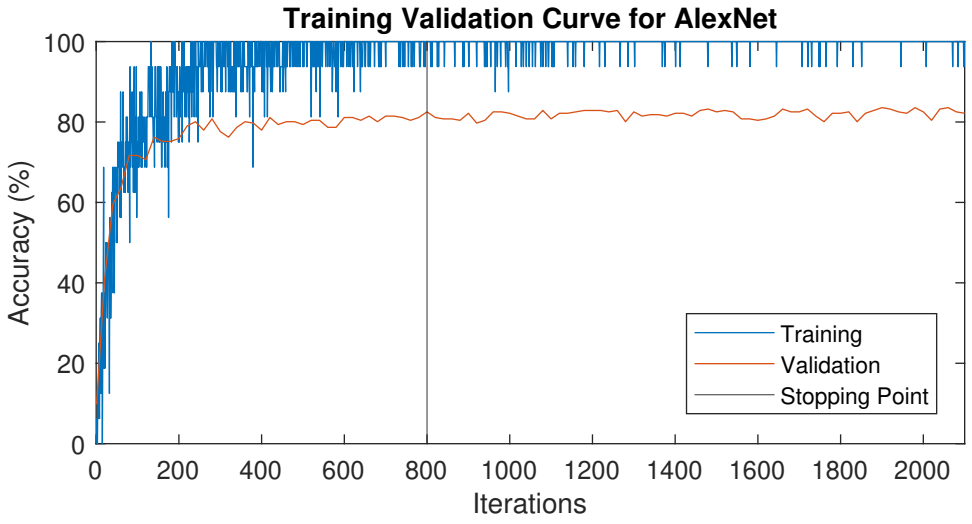


FIGURE 4.1: Training Validation Curve for AlexNet.

The training validation curve for AlexNet in Figure 4.1 shows a validation accuracy that is steadily increasing until roughly iteration 800 (epoch 19), where the validation accuracy hovers around 83%. The accuracy does fluctuates randomly from this point,

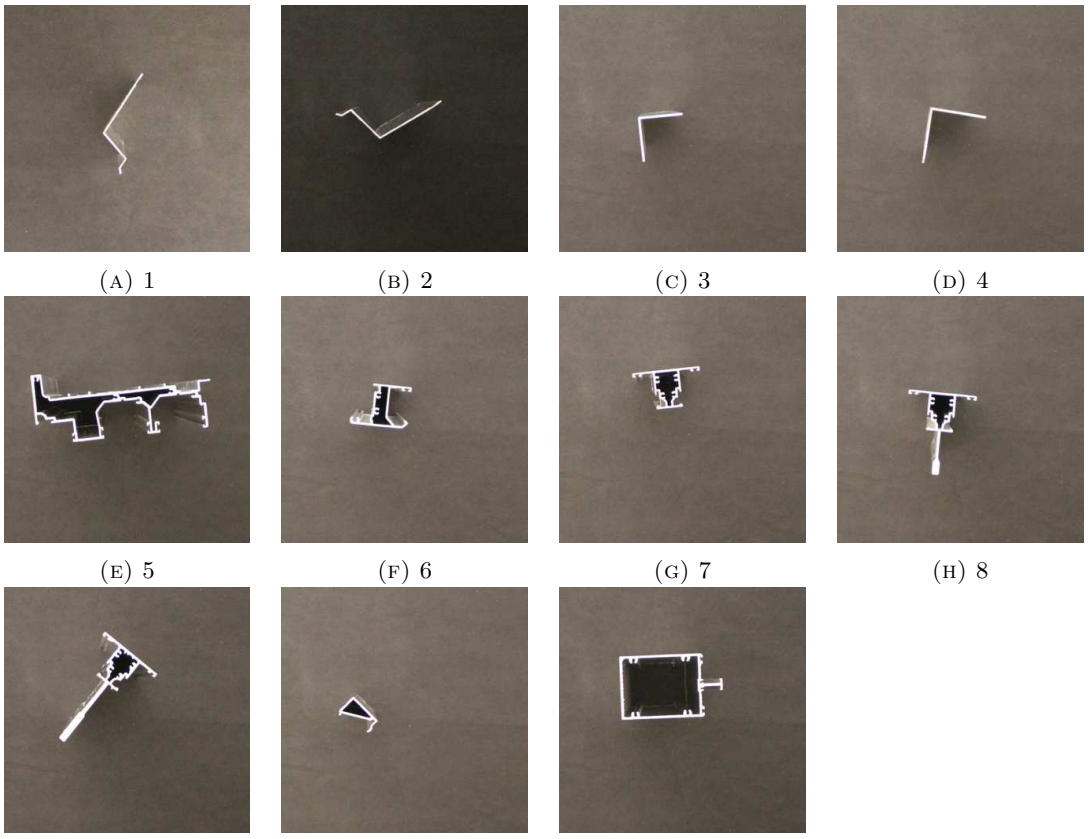
between 81% and 86%. The variation in the accuracies is not very large, suggesting that the validation set is not excessively small.

There starts to be a clear separation between training and validation accuracy after epoch 4. However, both the training and validation accuracy continues to increase. At epoch 4 the validation accuracy is 73%, approximately 10% lower than the average training accuracy reached. Whilst the increasing gap between the training and validation accuracy is typically used as a stopping point trigger, the validation accuracy is still increasing. The increase indicates that the model is still training, or that the validation set is not independent of the training data.

Across the training of the different algorithms, the validation accuracy did not begin to decrease. Instead, no change was observed. The lack of change is a more subtle trigger that the model may be starting to overfit at this point, and is a good place to stop the training. True evaluation is determined through prediction on the designated test set, data that the model has not seen before. The confusion matrix in Table 4.5 gives an accuracy of 75%. The predictions on the test set were also performed on the 50 epoch long models, for all the trained models. For AlexNet, the accuracy was 79%. The difference between the two accuracies is the largest compared to the other three models. The approximately 4% difference accounts for four more correctly classified instances. The instances that were classified differently between the two models were always one of the extrusions that has another similar extrusion (1 and 2, 3 and 4, and 7, 8 and 9, Figure 4.2).

TABLE 4.5: Confusion matrix of post training testing of AlexNet model.

Classified as	1	2	3	4	5	6	7	8	9	10	11
1	6	4	0	0	0	0	0	0	0	0	0
2	4	1	0	5	0	0	0	0	0	0	0
3	1	1	5	3	0	0	0	0	0	0	0
4	2	0	2	6	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0
6	0	0	0	0	0	10	0	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0	0
8	0	0	0	0	0	0	1	8	1	0	0
9	0	0	0	0	0	0	1	2	7	0	0
10	0	0	0	0	0	0	0	0	0	10	0
11	0	0	0	0	0	0	0	0	0	0	10



(A) 1 (B) 2 (C) 3 (D) 4

(E) 5 (F) 6 (G) 7 (H) 8

(I) 9 (J) 10 (K) 11

FIGURE 4.2: Each of the extrusions and their class number.

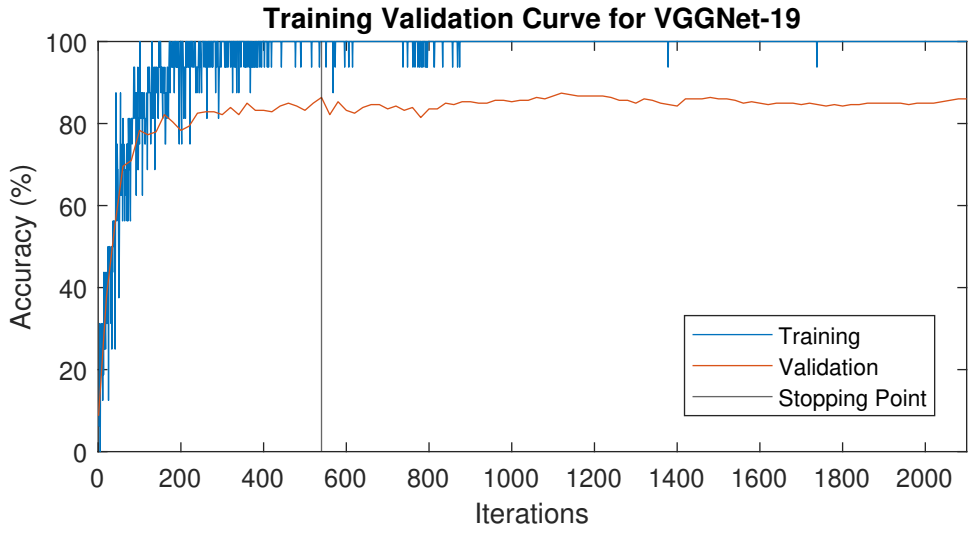


FIGURE 4.3: Training Validation Curve for VGG-19.

Similar trends in the data are seen in the remaining training curves as well. Each deep learning model reaches an accuracy plateau at different epochs and reach varying accuracies. Each of the training curves tend towards 100% accuracy and the validation accuracy never goes beyond 90%, indicating that the model may be over fitting to the

training data with increasing iterations. The validation accuracy after 50 epochs is 86% (Figure 4.3). The validation accuracy first reaches 86% at epoch 13, iteration 540.

VGG-19 presents the highest test accuracy out of all four networks, at 89%, with corresponding confusion matrix in Table 4.6. From epoch 13 the curves are steady with little variation throughout the remaining epochs. The test accuracy after 50 epochs also produces the same result, 89%. This result confirms that an appropriate stopping point was chosen, as the accuracy values for training, validation, and testing did not change from epoch 13. There are 10 instances in which these two models did not predict the same class as each other. Again, the extrusions in these instances are the ones with very similar profiles.

TABLE 4.6: Confusion matrix of post training testing of VGG-19 model.

Classified as	1	2	3	4	5	6	7	8	9	10	11
1	7	3	0	0	0	0	0	0	0	0	0
2	2	6	0	2	0	0	0	0	0	0	0
3	0	0	7	3	0	0	0	0	0	0	0
4	0	0	1	9	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0
6	0	0	0	0	0	10	0	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0	0
8	0	0	0	0	0	0	0	10	0	0	0
9	0	0	0	0	0	0	0	1	9	0	0
10	0	0	0	0	0	0	0	0	0	10	0
11	0	0	0	0	0	0	0	0	0	0	10

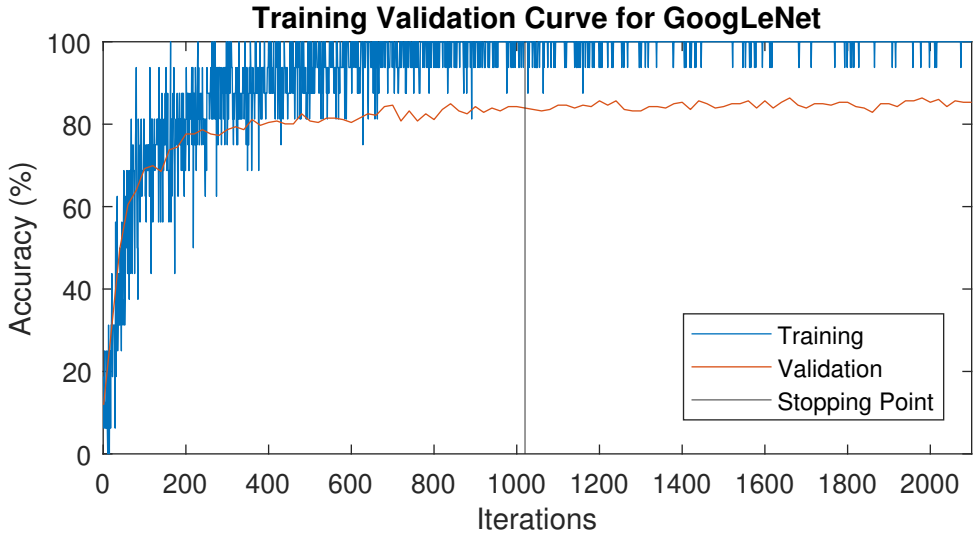


FIGURE 4.4: Training Validation Curve for GoogLeNet.

For the first 4 epochs while training GoogLeNet the training and validation accuracy hover around the same value as each other (Figure 4.4). From this point the training accuracy continues to increase until it is in the 90% range, whereas the validation accuracy slows down and begins to plateau in the 80% range. A similar trend is seen in all the curves.

At the end of 50 epochs, the validation accuracy reached 85%. This accuracy number is reached far earlier than epoch 50, at epoch 24, iteration 1020. Before this point the validation accuracy is steadily increasing. From this point in the training curve the validation accuracy plateaus, with the remaining validation scores within 2% for the rest of the epochs. Hence, the latest stopping point for training is epoch 24. The test accuracy at epoch 24 was 82%, while the test accuracy at epoch 50 was 83%, presenting a difference of 1%. This difference isn't large enough to suspect that there's a significant difference as it indicated that only one more instance was correctly classified. The extrusion in this instance was extrusion 2, and was misclassified as extrusion 1, an almost identical extrusion (Table 4.7).

TABLE 4.7: Confusion matrix of post training testing of GoogLeNet model.

Classified as	1	2	3	4	5	6	7	8	9	10	11
1	4	5	1	0	0	0	0	0	0	0	0
2	4	6	0	0	0	0	0	0	0	0	0
3	0	0	8	2	0	0	0	0	0	0	0
4	0	0	3	7	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0
6	0	0	0	0	0	10	0	0	0	0	0
7	0	0	0	0	0	0	9	1	0	0	0
8	0	0	0	0	0	0	0	9	1	0	0
9	0	0	0	0	0	0	0	3	7	0	0
10	0	0	0	0	0	0	0	0	0	10	0
11	0	0	0	0	0	0	0	0	0	0	10

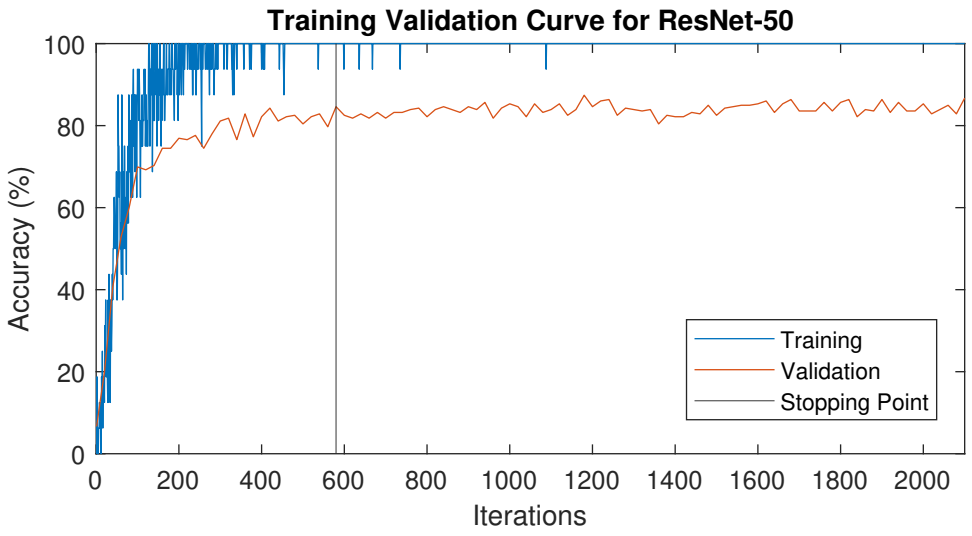
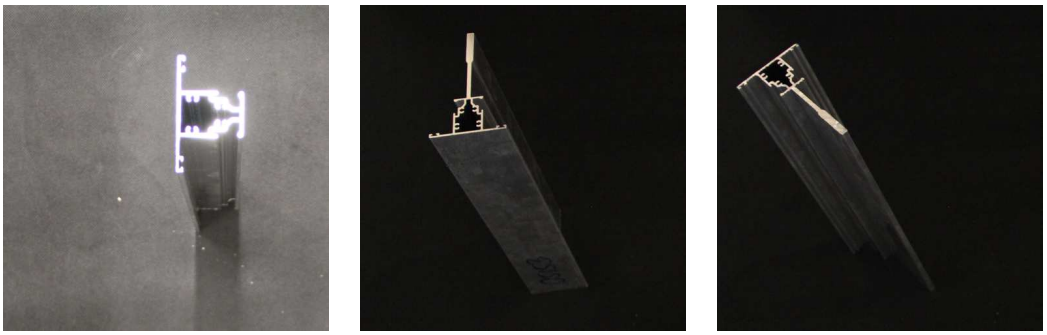


FIGURE 4.5: Training Validation Curve for ResNet-50.

The point on the ResNet training curve (Figure 4.5) where the accuracy begins to flatten is at epoch 14, iteration 580. From this point the training accuracy has supposedly already reached 100%, and the validation accuracy is 85%. After 50 epochs the validation accuracy is 85%. The validation curve is slightly more jagged than the previous VGG graph. The validation accuracy floats between 80% and 87%. The test accuracy at epoch 14 is 79%, and 80% at epoch 50. The 1% difference is one more correct instance, an image of extrusion 7 that was previously misclassified as extrusion 8 (Table 4.8).



(A) 7 (B) 9 (C) 9

FIGURE 4.6: Each of the extrusions and their class number.

TABLE 4.8: Confusion matrix of post training testing of ResNet model.

Classified as	1	2	3	4	5	6	7	8	9	10	11
1	7	3	0	0	0	0	0	0	0	0	0
2	4	4	0	2	0	0	0	0	0	0	0
3	1	0	7	2	0	0	0	0	0	0	0
4	0	2	1	7	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0
6	0	0	0	0	0	9	1	0	0	0	0
7	0	0	0	0	0	0	8	1	0	0	1
8	0	0	0	0	0	0	0	10	0	0	0
9	0	0	0	0	0	0	2	3	5	0	0
10	0	0	0	0	0	0	0	0	0	10	0
11	0	0	0	0	0	0	0	0	0	0	10

An image of extrusion 7 that was misclassified by two networks has saturation on the pixels of the profile (Figure 4.6 a). Two images of extrusion 9 that were mis-classified by three networks are images from the darker set. The dark lighting may be the reason they were misclassified, however they are the only two instances from the darker set that were misclassified. These three examples were all correctly classified by the VGG-19 model, and the instance of extrusion 7 was also classified correctly by AlexNet.

The training accuracy for all four networks tends towards 100%. The curves become smooth with little variation, indicating that the model has over fit. However, the test accuracy confirmed in all cases that the validation accuracy is an accurate value. An interesting find would be whether these results improve given more data.

TABLE 4.9: Accuracy (%) of pre-trained deep learning models on test set

	AlexNet	GoogLeNet	VGG-19	ResNet-50
Stopping Point	75.45	81.82	89.09	79.09
Epoch 50	79.09	82.73	89.09	80.00
Augmented Data	81.82	80.00	88.18	90.00

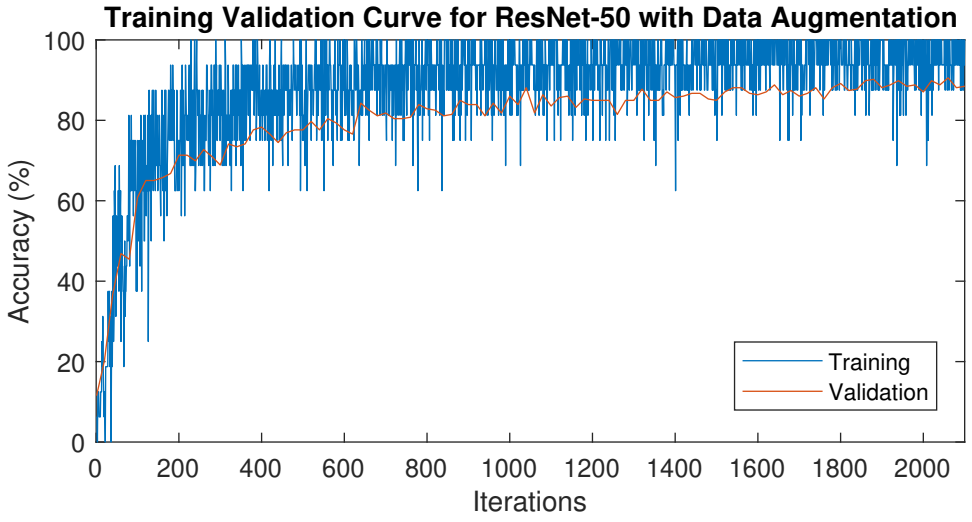


FIGURE 4.7: Training Validation Curve for ResNet-50 with Data Augmentation.

Based on the confusion matrices, AlexNet and ResNet have a higher tendency for extrusions 3 and 4 to be misclassified as extrusions 1 and 2, and vice versa, whereas this is a lot less common with GoogLeNet and VGG. All the networks have cases of misclassifying extrusions 1 and 2 as each other, and extrusions 3 and 4. That type of misclassification is present in all the classification approaches, so is not a problem exclusive to deep learning. Extrusion 5, 6, 10 and 11 are unique extrusions so often they are classified correctly 100% of the time. There is one instance by ResNet where an image of extrusion 6 is misclassified as extrusion 7. Within the family of extrusions 7, 8 and 9, both AlexNet and GoogLeNet misclassified 5 instances, and ResNet 7. VGG-19 only has one instance of this case that was mentioned earlier.

From these results and analysis without data augmentation, VGG-19 performs the best classification out of the four neural networks (Table 4.9). After training these networks again with augmented data, the test results show that using data augmentation of the training data affected the performance of each network differently. ResNet showed the greatest change in performance, an increase in test accuracy of 10%. GoogLeNet and VGG-19 performed slightly worse with the addition of data augmentation, and AlexNet performed only slightly better. Between the models with and without data augmentation, there was between 76% and 85% of the instances classified the same.

In the training curves for models with data augmentation (Figure 4.7), the training accuracy did not smooth out at 100% like in the previous training curves shown, indicating that the model is likely not overfitting. It is also possible that these models would have continued to increase with more training, however this would also increase the likelihood of overfitting the model.

TABLE 4.10: Confusion matrix of post training testing of ResNet model trained with augmented data.

Classified as	1	2	3	4	5	6	7	8	9	10	11
1	6	4	0	0	0	0	0	0	0	0	0
2	2	8	0	0	0	0	0	0	0	0	0
3	0	1	8	1	0	0	0	0	0	0	0
4	0	0	0	10	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0
6	0	0	0	0	0	9	1	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0	0
8	0	0	0	0	0	0	2	8	0	0	0
9	0	0	0	0	0	0	0	0	10	0	0
10	0	0	0	0	0	0	0	0	0	10	0
11	0	0	0	0	0	0	0	0	0	0	10

The ResNet-50 model trained with data augmentation is the only model that predicted all 10 instances of extrusion 4 correctly in the test set (Table 4.10). Extrusions 2 and 3 were predicted correctly 80% of the time, and extrusion 1 with 60%. Compared to other models, this ResNet model classifies these extrusions correctly more often. The confusion matrix is very comparable to the VGG-19 test confusion matrix, without data augmentation.

However, t-tests between the test predictions across all the networks with and without data augmentation on the training data, show that there is no significant difference between any of the transfer learning results. So there is not one network that is proven to be the most statistically accurate. The networks with the highest accuracies will be refered to loosely as the best performing networks solely based on the confusion matricies and testing accuracies, given the small size of the test image set.

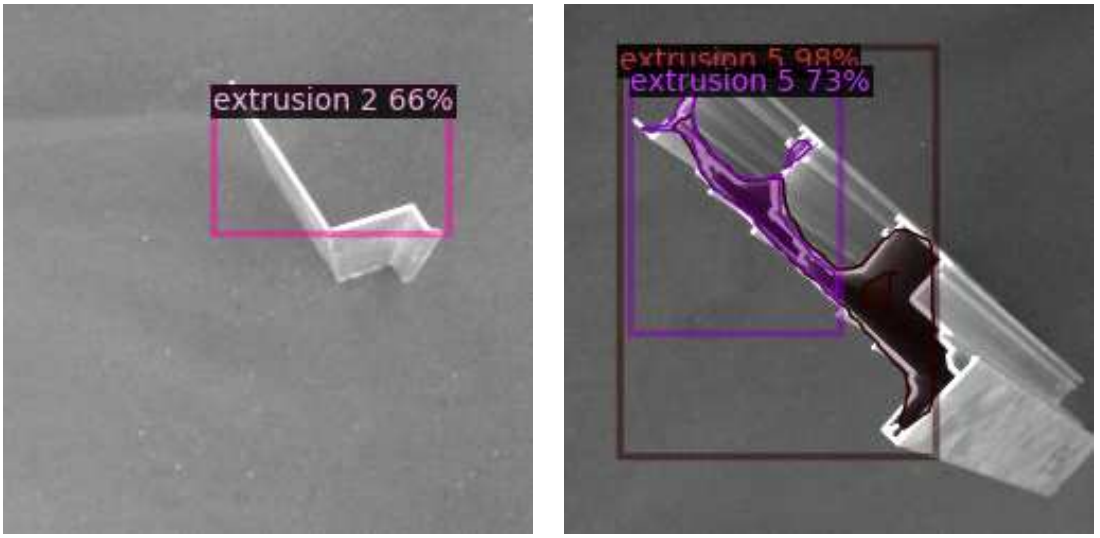
4.2.3 Object Detection

TABLE 4.11: Confusion matrix for post training testing of Mask R-CNN.

Classified as	1	2	3	4	5	6	7	8	9	10	11
1	0	4	0	0	0	0	0	0	0	0	0
2	0	6	0	0	0	0	0	0	0	0	0
3	0	0	0	4	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0
6	0	0	0	0	0	10	0	0	0	1	0
7	0	0	0	0	0	0	10	0	0	0	0
8	0	0	0	0	0	0	2	9	0	0	0
9	0	0	0	0	0	0	2	3	9	0	0
10	0	0	0	0	0	1	0	0	0	10	0
11	0	0	0	0	0	0	0	0	0	0	10

The same reason that the segmentation from Mask R-CNN did not work well in Chapter 3.3, applies to many of the misclassifications seen in the object detection output. The algorithm learns that extrusions have a dark centre, since that is present in 7 out of the 11 profiles and is a part of the annotations. However, some of extrusions 1-4 were detected with a bounding box (Figure 4.8 a), even when no segmentation was extracted. The test accuracy from the confusion matrix in Table 4.11 is 67%. For 3 extrusions (1, 3 and 4), zero of the 10 samples the test set were detected correctly.

Mask R-CNN produces multiple region proposals at the input stage. Regions are passed through the pipeline and some regions have information of only part of an extrusion. In the case of extrusion 5, a region of only part of the extrusion provides enough information to correctly classify this extrusion. In half of the test set images of extrusion 5, the extrusion was detected twice in the image (Figure 4.8 b). Between the two objects detected in the same image, the correct region has a higher confidence value than the smaller extra region. Increasing the confidence threshold of what is determined as the object will remove the extra regions. However, in many images across all the extrusions there are positive matches that are correct but have a low confidence score. These objects would be removed with a higher threshold.



(A) An instance of extrusion 1 that was misclassfied as extrusion 2.

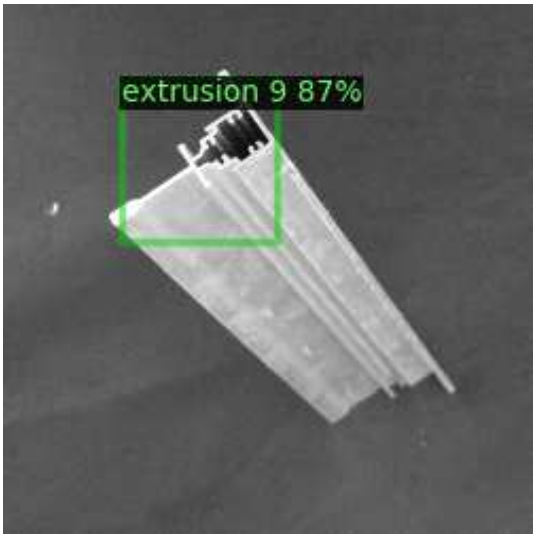
(B) An instance of extrusion 5 that was detected twice.

FIGURE 4.8: Examples of Mask R-CNN output failure cases.

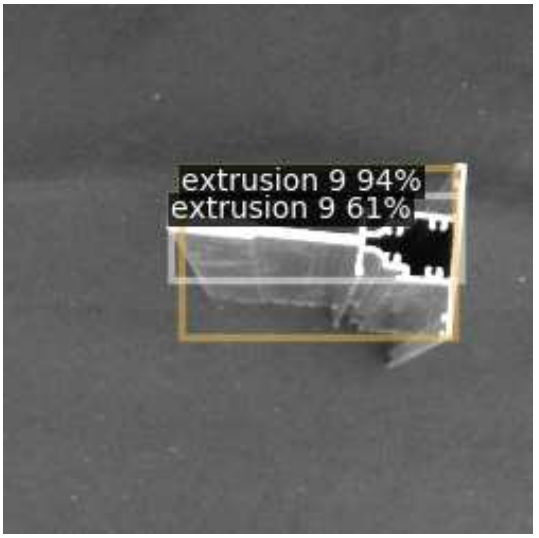
TABLE 4.12: Confusion matrix for post training testing of Faster R-CNN.

Classified as	1	2	3	4	5	6	7	8	9	10	11
1	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	4	2	0	0	0	0	0	0	0
4	0	0	2	2	0	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0
6	0	0	0	0	1	10	0	0	0	1	0
7	0	0	0	0	0	0	10	0	0	0	0
8	0	0	0	0	0	0	0	0	9	0	0
9	0	0	0	0	0	0	2	0	9	0	0
10	0	0	0	0	0	1	0	0	0	10	0
11	0	0	0	0	0	0	0	0	0	0	10

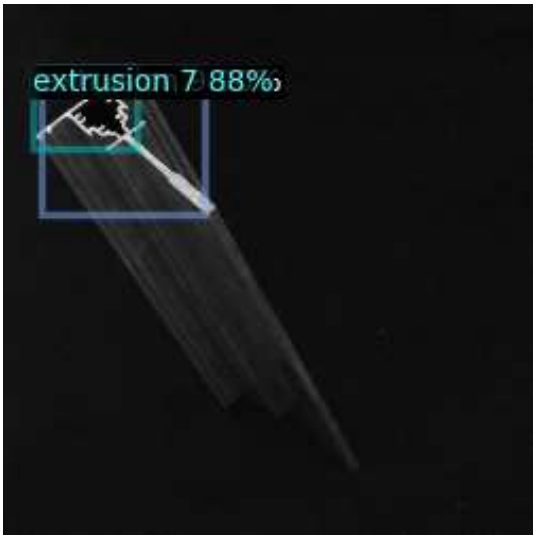
The test accuracy from the Faster R-CNN confusion matrix (Table 4.12) is 60%. All instances in the test set of extrusion 8 were misclassified as extrusion 9 (Figure 4.9), indicating that Faster R-CNN was not able to clearly separate the feature information between the two extrusions. To the human eye, these extrusions can be distinguished easily, as one is clearly longer than the other.



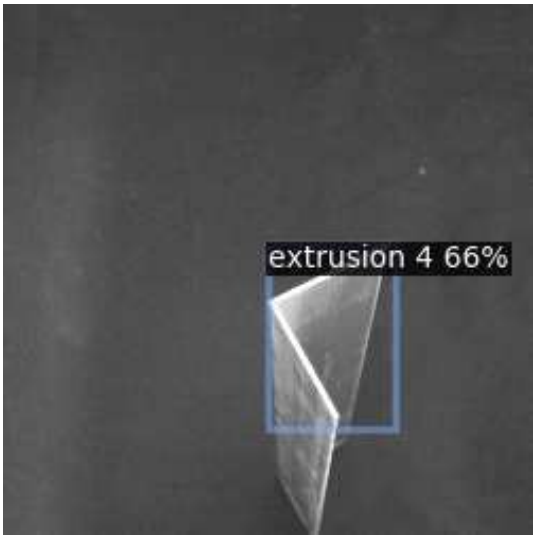
(A) An instance of extrusion 8 that was misclassified as extrusion 9.



(B) An instance of extrusion 9 that was detection twice.



(C) An instance of extrusion 9 where extrusion 7 was also detected.



(D) An instance of extrusion 4 that was correctly classified.

FIGURE 4.9: Examples of Faster R-CNN output failure cases.

Similar to how multiple instances of extrusion 5 are detected within one image, the same occurs within the images of extrusion 9. One instance of extrusion 9 showed the extrusion was detected twice, again one detection had a distinctly lower confidence score and can be removed with a lower threshold. Since the main part of these two extrusions are identical to extrusion 7, 2 instances of extrusion 7 were also detected. The detection of 7 in these images sometimes have higher confidence scores than other correct detections. For example, the instance of extrusion 7 detected in an image of extrusion 9 had a confidence score of 88%, just 2% lower than the correctly detected extrusion 9 with 90%. Whereas a correctly classified instance of extrusion 4 had a confidence score of 66%, lower than the false detection of extrusion 7.

The two networks performed similarly to each other for the rest of the extrusions. These results therefore show that Mask R-CNN produced a more accurate model for detecting the extrusions compared to Faster R-CNN. Mask R-CNN is more developed algorithm and is an improvement from Faster R-CNN, as evident in the results.

Even though the accuracy for the multi-class testing in Weka produced a higher value compared to the Mask R-CNN approach, Mask R-CNN better separated the obviously different classes. Overall, the transfer learning algorithms produced the highest performance scores across the classification methods investigated, with or without the aid of data augmentation. In particular VGG-19 with only the original data set, and also ResNet-50 with the use of data augmentation. The performance difference in accuracy between these two models was 1%. Object detection models were found unsuitable for this application due to the requirements of the input data for the algorithms. The Mask R-CNN and Faster R-CNN approaches could be re-evaluated with a varying approach for future investigation.

Chapter 5

Discussion and Comparison of Classification Pipelines

Perfect classification of extrusions 5, 6, 10 and 11 are almost expected since they are the most distinct extrusions in this sample set. Based on the results, it appears to be easy to extract meaningful features that separate these extrusions from the rest. Many of the algorithms identified all of these extrusions in the test set correctly 100% of the time. Consequently we most critically examine how well the rest of the extrusions are classified, since they are not distinct. The odd misclassification within extrusions 7, 8 and 9 is present across every classifier. From visual inspection, it is not obvious why many of the extrusions are misclassified. A possible explanation is that the derived features are too similar to each other to separate the data.

Extrusions 1-4 were best classified by computing the generic Fourier descriptors (GFD) from segmented images and using a Logistic Regression classifier, where at least 80 or 90% of the extrusions were classified correctly. Compared to the rest of the classifiers, the GFD method however misclassified one instance of extrusions 5, 6 and 11. As discussed before, these misclassifications were due to poor segmentations. With that method, a solution to the remaining misclassifications is easily identifiable as the problem is due to the quality of the image. In future work, the image acquisition step can be manipulated to optimise lighting and maximise the accuracy of segmentation.

The data augmented trained ResNet-50 model predicted one more instance correctly compared to the GFD method. However, ResNet-50 is a deep learning network and requires significant computer performance to perform predictions quickly. Compared to a random forest and logistic regression method, a deep learning method has greater complexity and less transparency.

A future step for this project is to determine what processor would be used in a real system. This choice is dependent on cost, budget and further research. Simpler algorithms and methods are preferred as they require a less sophisticated computing system and are easier to understand and therefore, easier to fix when problems arise. An example option processor is the MeLE computer stick, that offers comparable processing ability to the desktop computer that the vast majority of this research was carried out on. Compute sticks are considered mini computers and can plug into any screen with an HDMI port. The MeLE PCG02 computer stick retails for 239.99NZD on Amazon[1].

A classifier is more impressive if it is able to separate the almost identical extrusions. The GFD method confusion matrix suggests that the GFD algorithm is able to extract features that better separate these harder to classify extrusions compared to the other methods.

TABLE 5.1: Summary of top results for classifying extrusion end shapes.

Model	Accuracy (%)
Faster R-CNN	60
Mask R-CNN	67
SVM with Weka filters	78
AlexNet with Augmented Data	82
GoogLeNet	82
GFD Logistic Regression	88
VGG-19	89
ResNet-50 with Augmented Data	90

The methods that stand out as the best performing classification pipelines is GFD trained Logistic Regression model on segmented images, VGG-19 on the original dataset, and ResNet-50 on a dataset that utilises data augmentation (Table 5.1). Through t-tests, there is no significant difference between these three methods results. Further investigation of these methods with parameter searching algorithms, a chosen processor and knowledge on the physical environment of a working system will allow for a more concrete recommendation. The top three performing classification pipelines have very similar performance. Most critically, a much larger scale study will determine if any of the small differences in the classification power between the top performing classification methods are genuine, or if they ultimately have the same performance.

[1]https://www.amazon.com/MeLE-J4125-128G-PCG02-GLE/dp/B08LYRQZ59?th=1

Chapter 6

Conclusion and Future Work

In conclusion, multiple pipelines for classifying aluminium extrusion profiles were presented with accuracy results. The top approaches involved Fourier descriptors with logistic regression (88%) and deep learning algorithms VGG-19 (89%) and ResNet-50 (90%).

Data was captured that represented the possible variation between images. A subset of 11 extrusion samples were provided and the lighting was used in a controlled environment that provided suitable image intensity. The extrusions were chosen so that some were very similar to others, and some of them were very distinct from the rest. A perfect mask image set was created through varying threshold and manual pixel filling, which became the reference data for evaluation.

The Fourier approach involved firstly performing pixel classification on grayscale images. Three image segmentation methods were explored. The first method was to apply image filters such as shade correction, thresholding, and morphological operators to binarize the images without any machine learning. This simple method was not adaptable enough between images. Parts of the extrusion that aren't the profile were also segmented in the process, and were not easily removable with post-processing. Classical filters such as edge detectors, Gabor filters, and Gaussian blurs were used to extract features from the images. The filtered images became the training data for a random forest classifier. This involved testing various sizes of input training sets to achieve the best possible outcome. The deep learning model Mask R-CNN was also used for segmentation. Due to the requirements of the input annotation file, this method did not segment solid extrusions. Annotations for this algorithm cannot be hollow, therefore the dark centres of some extrusions were included in the positive pixel labels despite containing pixels that are not a part of the profile. The large contrast in intensity between these two sets of pixels did not provide clearly separable data for Mask R-CNN to create accurate masks.

Considering the ratio of background to profile is unbalanced, accuracy isn't indicative of performance for segmentation, and thus the primary metric evaluated was the f1-score. The random forest method produced the highest results consistently compared to the other two methods, segmentation using image processing techniques and Mask R-CNN. These results provided sufficient information to determine that the machine learning random forest method produced the best segmentations, with an f1-score of 0.922.

A logistic regression algorithm was trained using the perfect images and generic Fourier descriptors to classify the extrusions. The segmented images from the random forest algorithm were then tested on the trained classifier and produced a test accuracy of 88%. Classification directly from colour images was also investigated using varying image filters to extract features in Weka, transfer learning in MATLAB, and object detection algorithms Mask R-CNN and Faster R-CNN in Python. Out of the four neural networks tested with transfer learning, VGG-19 and ResNet-50 presented the highest test accuracies. VGG-19 produced an accuracy of 89% without any augmented data, and ResNet-50 produced an accuracy of 90% using data augmentation.

The choice of which algorithm to use is dependent on, for example, the processor that will be used, if the system is intended to work in real time, the amount of additional training data that can be collected and the lighting environment of the scene where the cameras are set up. From each method investigated, it is evident that sufficient and even illumination upon the extrusion profile end is crucial in segmentation and classification approaches. There must be enough illumination to highlight the entire extrusion end without oversaturation.

A range of view positions and angles were used in the testing set, and the instances of poor segmentation or classification were often due to poor image quality. These results indicate that even illumination upon the profile allows the highest likelihood of correct classification. It is likely that the flatter the angle of the extrusion the easier it is to achieve even illumination, however the lighting can be manipulated to achieve this at a range of angles.

The dataset acquired of 1077 images is relatively small for machine learning exploration. For future investigations on this project, acquiring significantly more data with additional controlled variation will allow a greater exploration into the challenges of the problem. There are hundreds of different extrusion profiles manufactured. Widening the selection of extrusions for testing in subsequent research will expose further shape factors and new information.

Next steps in this research would involve exploring object detection and classification on images with multiple extrusions that are the same and also images with a variety

of profiles present. The methods that have performed well in this research are not guaranteed to perform the same on a cluster of different or the same profiles. This means investigating ways to prepare the images to use the methods researched, or to explore additional methods.

Though segmentation and the use of the generic Fourier descriptor prove useful in classifying the singular images, additional steps of object detection to separate multiple extrusions in an image have to occur prior to classification. The additional complexity would have to be compared to the other methods to determine what is more suitable for the application.

It is seen in the initial dataset of images that even to the human eye, the pixels that represent separate extrusions within the same image are difficult to differentiate, due to the shapes of the profiles and the way they are stacked next to each other in the bays. While the contents of this thesis show that classification of the profile shapes is possible with meaningful results, the additional investigation will contribute significantly to recommendations for a final system.

There are many pairs of extrusions that are almost identical, as they come from the same family of extrusions. An entire category of extrusions could be classified as a single extrusion, as seen with extrusions 1 and 2 in this project. A possible method to avoid this is a two-step classification process. The first classifier identifying the category of the extrusion, and a second classifier that identifies which extrusion it is in that given category.

Bibliography

Abraham, A. *Handbook of Measuring System Design*, chapter Artificial Neural Networks, pp. 901–908. John Wiley & Sons, Ltd, Chichester, UK (2005).

Adams, J., Y. Qiu, Y. Xu, and J. Schnable. Plant segmentation by supervised machine learning methods. *The Plant Phenome Journal*, **3(1)**, pp. 1–11 (2020).

Alam, F., R. Mehmood, and I. Katib. *Smart Infrastructure and Applications: Foundations for Smarter Cities and Societies*, chapter Comparison of Decision Trees and Deep Learning for Object Classification in Autonomous Driving, pp. 135–158. Springer International, Cham, Switzerland (2020).

Albawi, S., T. A. Mohammed, and S. Al-Zawi. Understanding of a convolutional neural network. In: *International Conference on Engineering and Technology (ICET)*, pp. 1–6. Antalya, Turkey (2017).

Bishop, C. M. and N. M. Nasrabadi. *Pattern Recognition and Machine Learning*, volume 4, chapter Linear Regression, pp. 206–209. Springer, New York (2006).

Boonsim, N. and S. Prakoonwit. Car make and model recognition under limited lighting conditions at night. *Pattern Analysis and Applications*, **20(4)**, pp. 1195–1207 (2017).

Breiman, L. Random forests. In: *Machine Learning*, volume 45, pp. 5–32 (2001).

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*, chapter Regression Trees, pp. 216–265. Chapman & Hall/CRC, Boca Raton, Florida (2017).

Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-8(6)**, pp. 679–698 (1986).

Chakravarti, R. and X. Meng. A study of color histogram based image retrieval. In: *6th International Conference on Information Technology: New Generations*, pp. 1323–1328. Las Vegas, Nevada (2009).

Chen, J. and W. K. Jenkins. Facial recognition with PCA and machine learning methods. In: *IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 973–976. Boston, MA (2017).

Chicco, D. and G. Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, **21(1)**, pp. 1–13 (2020).

Choudhury, S. D., S. Bashyam, Y. Qiu, A. Samal, and T. Awada. Holistic and component plant phenotyping using temporal image sequence. *Plant Methods*, **14(35)**, pp. 1–21 (2018).

Chowdhary, C. L. and D. Acharjya. Segmentation and feature extraction in medical imaging: A systematic review. *Procedia Computer Science*, **167**, pp. 26–36 (2020).

Das, T. Machine learning algorithms for image classification of hand digits and face recognition dataset. *International Research Journal of Engineering and Technology (IRJET)*, **04(12)**, pp. 640–649 (2017).

Dietterich, T. G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, **40(2)**, pp. 139–157 (2000).

Dougherty, E. and R. Lotufo. *Hands-on Morphological Image Processing*, chapter Binary Opening and Closing, pp. 25–38. SPIE Optical Engineering Press, Bellingham, Washington (2003).

Duan, K.-B. and S. S. Keerthi. Which is the best multiclass SVM method? an empirical study. In: *International Workshop on Multiple Classifier Systems*, pp. 278–285. Seaside, CA (2005).

Everingham, M., L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, **88(2)**, pp. 303–338 (2010).

Freund, Y. and L. Mason. The alternating decision tree learning algorithm. In: *International Conference on Machine Learning (ICML)*, volume 99, pp. 124–133 (1999).

Ghosh, S., N. Das, I. Das, and U. Maulik. Understanding deep learning techniques for image segmentation. *ACM Computing Surveys (CSUR)*, **52(4)**, pp. 1–35 (2019).

Gonzalez, R. C. and R. E. Woods. *Digital Image Processing*, chapter Morphological Reconstruction, pp. 120–144. Prentice Hall, Upper Saddle River, NJ, third edition (2008).

Hart, P. E., D. G. Stork, and R. O. Duda. *Pattern Classification*, chapter Support Vector Machines, pp. 259–265. John Wiley and Sons, New York, second edition (2000).

He, K., G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969. Venice, Italy (2017).

He, K., X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778. Las Vegas, Nevada (2016).

Hsieh, J.-W., L. Chen, D.-Y. Chen, and S.-C. Cheng. Vehicle make and model recognition using symmetrical SURF. In: *10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 472–477. Krakow, Poland (2013).

Huang, J., S. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 762–768. Puerto Rico (1997).

Huang, T., G. Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **27(1)**, pp. 13–18 (1979).

James, G., D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R*, chapter Support Vector Machines, pp. 367–402. Springer, New York (2013).

Jawahar, C. and A. Ray. Incorporation of gray-level imprecision in representation and processing of digital images. *Pattern Recognition Letters*, **17(5)**, pp. 541–546 (1996).

Jimenez, A., R. Ceres, and J. L. Pons. A survey of computer vision methods for locating fruit on trees. *Transactions of the American Society of Agricultural and Biological Engineers*, **43(6)**, pp. 1911–1920 (2000).

Julina, J. K. J. and T. S. Sharmila. Facial recognition using histogram of gradients and support vector machines. In: *International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pp. 1–5. Chennai, India (2017).

Kasutani, E. and A. Yamada. The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In: *Proceedings of the International Conference on Image Processing*, volume 1, pp. 674–677. Thessaloniki, Greece (2001).

Kim, H., E. Ahn, M. Shin, and S.-H. Sim. Crack and non crack classification from concrete surface images using machine learning. *Structural Health Monitoring*, **18(3)**, pp. 725–738 (2019).

Kremic, E. and A. Subasi. Performance of random forest and SVM in face recognition. *International Arab Journal of Information Technology*, **13(2)**, pp. 287–293 (2016).

Krizhevsky, A., I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, volume 25, pp. 1097–1105. Lake Tahoe, Nevada (2012).

Kuang, Y. C., L. Streeter, M. J. Cree, and M. P.-L. Ooi. Evaluation of deep neural network and alternating decision tree for kiwifruit detection. In: *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6. Auckland, New Zealand (2019).

Kumar, S., Z. Khan, and A. Jain. A review of content based image classification using machine learning approach. *International Journal of Advanced Computer Research*, **2(3)**, pp. 55–60 (2012).

Lawrence, R. and C. Moran. The americaview classification methods accuracy comparison project: A rigorous approach for model selection. *Remote Sensing of Environment*, **170**, pp. 115–120 (2015).

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86(11)**, pp. 2278–2324 (1998).

Li, S., K. Zhang, P. Duan, and X. Kang. Hyperspectral anomaly detection with kernel isolation forest. *IEEE Transactions on Geoscience and Remote Sensing*, **58(1)**, pp. 319–329 (2019).

Liu, F. T., K. M. Ting, and Z.-H. Zhou. Isolation forest. In: *8th IEEE International Conference on Data Mining*, pp. 413–422. Pisa, Italy (2008).

Llorca, D. F., D. Colás, I. G. Daza, I. Parra, and M. A. Sotelo. Vehicle model recognition using geometry and appearance of car emblems from rear view images. In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 3094–3099. Qingdao, China (2014).

Mehul, G., P. Ankita, D. Namrata, G. Rahul, and S. Sheth. Text-based image segmentation methodology. *Procedia Technology*, **14**, pp. 465–472 (2014).

Ohm, J.-R., L. Cieplinski, H. J. Kim, S. Krishnamachari, B. Manjunath, D. S. Messing, and A. Yamada. The MPEG-7 color descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, **24**, pp. 658–665 (2001).

Peña-Barragán, J. M., P. A. Gutiérrez, C. Martínez, J. Six, R. Plant, and F. López-Granados. Object-based image classification of summer crops with machine learning methods. *Remote Sensing*, **6**, pp. 5019–5041 (2014).

Powers, D. M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *International Journal of Machine Learning Technology*, **2(1)**, pp. 37–63 (2011).

Rahman, M. M., S. Ahammed, and M. R. Ahmed. A semi blind watermarking technique for copyright protection of image based on DCT and SVD domain. *Global Journal of Research in Engineering*, **16**, pp. 9–15 (2017).

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.* Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, **115(3)**, pp. 211–252 (2015).

Shapiro, L. and G. Stockman. *Computer Vision*, chapter Gaussian Filtering and LOG Edge Detection, pp. 166–170. Pearson, Upper Saddle River, NJ, first edition (2001).

Sharma, P., P. Hans, and S. C. Gupta. Classification of plant leaf diseases using machine learning and image preprocessing techniques. In: *10th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pp. 480–484. Noida, India (2020).

Shrivakshan, G. and C. Chandrasekar. A comparison of various edge detection techniques used in image processing. *International Journal of Computer Science Issues (IJCSI)*, **9(5)**, pp. 269–276 (2012).

Simonyan, K. and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In: *International Conference on Learning Representations (ICLR) 2015*, p. 14. San Diego, CA (2014).

Singh, A., N. Thakur, and A. Sharma. A review of supervised machine learning algorithms. In: *3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 1310–1315. New Delhi, India (2016).

Sobel, I., R. Duda, and P. Hart. *Pattern Classification and Scene Analysis*, chapter Sobel-Feldman Operator, pp. 271–272. John Wiley and Sons, New York (1973).

Soille, P. *Morphological Image Analysis: Principles and Applications*, chapter Erosion and Dilation, pp. 63–103. Springer, Berlin, Heidelberg (2004).

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9. Boston, MA (2015).

Tharwat, A. Classification assessment methods. *Applied Computing and Informatics*, **17(1)**, pp. 168–192 (2020).

Tolba, A., A. El-Baz, and A. El-Harby. Face recognition: A literature review. *International Journal of Signal Processing*, **2(2)**, pp. 88–103 (2005).

Vertan, C. and N. Boujemaa. Using fuzzy histograms and distances for color image retrieval. *Challenge of Image Retrieval*, **6**, p. 6 (2000).

Watkins, B. and A. Van Niekerk. A comparison of object-based image analysis approaches for field boundary delineation using multi-temporal sentinel-2 imagery. *Computers and Electronics in Agriculture*, **158**, pp. 294–302 (2019).

Weldon, T. P., W. E. Higgins, and D. F. Dunn. Efficient Gabor filter design for texture segmentation. *Pattern Recognition*, **29(12)**, pp. 2005–2015 (1996).

Witten, I. H., E. Frank, M. A. Hall, and C. Pal. *Data Mining: Practical machine learning tools and techniques*, chapter Divide-and-Conquer: Constructing Decision Trees, pp. 105–113. Elsevier, Amsterdam, Netherlands, fourth edition (2011a).

Witten, I. H., E. Frank, M. A. Hall, and C. Pal. *Data Mining: Practical machine learning tools and techniques*, chapter Multiclass Logistic Regression, pp. 396–397. Elsevier, Amsterdam, Netherlands, fourth edition (2011b).

Won, C. S., D. K. Park, and S.-J. Park. Efficient use of MPEG-7 edge histogram descriptor. *Electronics and Telecommunications Research Institute (ETRI) Journal*, **24(1)**, pp. 23–30 (2002).

Yassin, N., S. Omran, E. El Houby, and H. Allam. Machine learning techniques for breast cancer computer aided diagnosis using different image modalities: A systematic review. *Computer Methods and Programs in Biomedicine*, **156**, pp. 25–45 (2017).

Yegnanarayana, B. *Artificial neural networks*, chapter Basics of Artificial Neural Networks, pp. 15–39. Prentice Hall of India, New Delhi (2009).

Zhang, D. and G. Lu. Shape-based image retrieval using generic Fourier descriptor. *Signal Processing: Image Communication*, **17(10)**, pp. 825–848 (2002).

Çinarer, G. and B. G. Emiroglu. Classification of brain tumors by machine learning algorithms. In: *3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 1–4. Ankara, Turkey (2019).