Developing a Computer-Aided Diagnosis System for Pigmented Skin Lesions in Dermoscopic Images

Thomas Lafferty – C1924454

Supervisor: Paul Rosin

Moderator: Yukun Lai

Final Year Project, BSc Computer Science

School of Computer Science and Informatics, Cardiff University

May 27, 2022

Abstract

The skin is the body's largest organ accounting for roughly 16% of total body weight, and just like any body part, the cells that comprise the skin can develop cancers. Skin cancer is one of the most common cancers, and some variants are life-threatening. Most people have between 20 and 50 pigmented skin lesions, and although most are harmless, roughly 1 in 33,000 skin lesions become cancerous. Early diagnosis is key for cancer survival, and computer-aided diagnosis (CAD) systems intend to save lives by helping speed up the process of obtaining a diagnosis. This report details the design and implementation of a CAD system for pigmented skin lesions. The proposed system can classify dermoscopic images of skin cancer with upwards of 83.3% accuracy using a Random Forest classifier. The best results were achieved with 10-fold cross validation on the HAM10000 dataset; the reported accuracy was 89.56%, the reported F1 score was 0.895, the MCC score was 0.794, and the ROC Area score was 0.962. These results were achieved using the full feature set, including 88 features describing each lesion in terms of its asymmetry, border structure, colour, and differential structures, according to the ABCD dermoscopy algorithm.

Acknowledgements

I would like to thank my supervisor Paul Rosin for his expert guidance throughout the project. His active research also proved invaluable, as I learnt a lot from reading his publications.

I would also like to thank my friends and family for being supportive in so many ways.

Contents

1 Introduction

The skin is the body's largest organ and protects the body from infection and injury while also regulating the body's temperature and creating vitamin D. The epidermis – the outer layer of the skin, is the most at risk of sun damage. Melanocyte cells in the epidermis produce melanin when exposed to sun, but too much exposure can lead to melanoma skin cancer. Melanoma skin cancer is the 5th most common cancer in the UK, with around 50 new cases reported every day; incidence rates have increased by 140% since the early 1990s (Cancer Research UK 2015). While survival rates remain relatively high, more than 6 people die from skin cancer every day in the UK (Cancer Research UK 2015). Early detection of cancerous moles is key; when diagnosed at its earliest stage 100% of patients survive for at least 1 year, which decreases to around 50% when diagnosed at its latest stage (Cancer Research UK 2015).

The inner layers of skin, just like all cells in the body, can also develop cancers. The most common example of non-melanoma skin cancer is basal cell carcinoma, which develops in keratinocytes – cells also located in the epidermis. Non-melanoma skin cancer is even more common than melanoma skin cancer, with 430 new cases reported every day (Cancer Research UK 2018). This incidence rate has increased by 163% since the early 1990s (Cancer Research UK 2018). While the incidence rates are much higher than for melanoma skin cancer, the mortality rates are much lower; there are around 2 deaths every day from non-melanoma skin cancer in the UK (Cancer Research UK 2018).

Diagnosing a lesion that appears on the skin typically involves a physical examination including dermoscopy, where a microscopic image of the lesion is captured, revealing characteristics not visible to the naked eye (DermNet NZ 2004). Some lesions may require further examination by blood tests, microbial swabs, X-ray, CT scan, or even biopsy (DermNet NZ 2008). With increasing rates of incidence and busy health services already struggling with backlogs world-wide (World Health Organization 2022), it is certainly beneficial to speed up the process of obtaining diagnoses for all kinds of diseases, including skin cancer, so that more patients can stand a better chance at survival. In recent years, along with advancements in computer vision and artificial intelligence, researchers have been pushing for automated melanoma recognition using computers. This process involves image analysis of dermatoscopic images by a computer. Several studies found that state-of-the-art machine learning classifiers matched or even outperformed human experts in experiments (Kassem et al. 2021). However, this does not mean that dermatologists have been put out of a job just yet, there are still many problems that this technology must overcome before it can be considered an alternative rather than a supplement to a specialist's diagnosis. For instance a lack of large and representative datasets and standardised frameworks for comparing machine learning models (Kassem et al. 2021).

The primary aim of this project is to create a computer aided diagnosis (CAD) system for pigmented skin lesions. The system will analyse an image of a skin lesion, and predict whether the lesion is benign (harmless) or malignant (dangerous). Through this process, some of the advantages and drawbacks of automated diagnosis will be investigated.

2 Background

This section describes some of the medical and technical concepts that are necessary to understand before any implementation of a CAD system can commence.

2.1 Pigmented Skin Lesions

Skin lesions are areas of skin having abnormal growth or appearance compared to the surrounding skin. Pigmented skin lesions are caused by melanocyte cells in the skin, which produce melanin, making the lesion appear brown, black, or blue in colour. These lesions are most often benign (harmless), and some people are born with them, for example: birthmarks, freckles, and moles. However, some skin lesions can progress over time to become malignant (cancerous); changes in colour, shape, or size of a skin lesion over time is a cause for concern and should be subject to examination by a dermatologist.

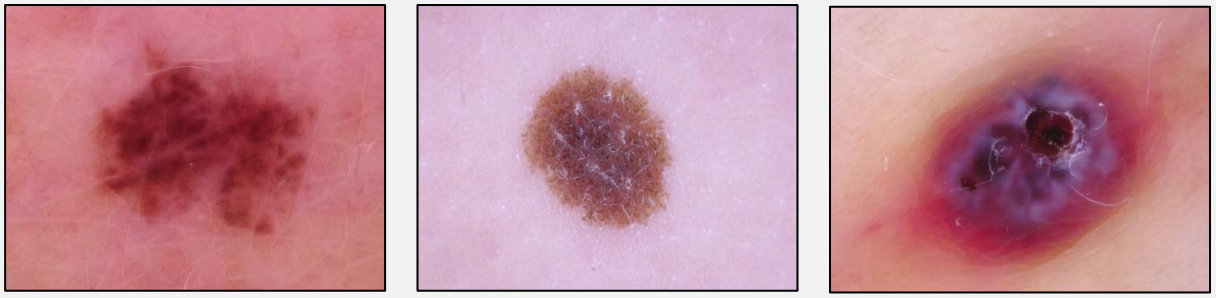


Figure 1: Examples of skin lesions from the HAM10000 skin lesion dataset (Tschandl et al 2018, licensed under CC BY-NC 4.0)

Benign	Malignant
Melanocytic nevus	Melanoma
Dermatofibroma	Vascular lesions
Actinic keratosis	Squamous cell carcinoma
Benign keratosis	
Basal cell carcinoma	

Table 1: Skin lesions included in the HAM10000 dataset categorised into benign / malignant

2.1.1 Dermoscopy

Dermoscopy (dermatoscopy), also called 'epiluminescent microscopy' (ELM) has become one of the most important tools used by a dermatologist in their efforts to evaluate a pigmented skin lesion. It is a non-invasive skin imaging technique; the dermatologist will use a dermatoscope, a handheld device consisting of a 10-20x magnified lens and powerful lighting, to examine the colour and structure of the skin's pigmentation, which are not so visible to the naked eye. In the hands of an expert, the tool increases the diagnostic accuracy by upwards of 10-27% (Argenziano et al. 1998), and therefore can reduce screening errors, meaning fewer benign lesions being excised for further analysis. In other words, the correct detection rate for melanomas and non-melanomas (sensitivity & specificity) is improved.

A dermatoscope can also be used to capture digital images or video clips, which allow for further analysis without needing the patient to be present. While the dermatoscope is a must-have in a dermatologist's toolkit, there are also convenient smartphone attachments available for anybody to purchase for as little as £100. This practice of at-home dermoscopy has the potential to make tracking concerning lesions much easier, as the patient may not need to be present for a physical clinical examination, but rather capture images of their skin at home to send to their healthcare provider or an online service such as the Spot Check clinic (Spot Check 2022) or SkinVision (SkinVision 2018).

2.1.2 Dermatoscopic Features

There are several algorithms used by experts and non-experts that help to diagnose a pigmented skin lesion. Because there are many kinds of benign and malignant lesions, their differences in appearance can often be subtle, particularly to non-experts, which can potentially lead to an incorrect prognosis. For example, seborrheic keratoses are common benign lesions that typically present with age, but highly pigmented variants can resemble malignant lesions (BMJ 2021). For this reason, very specific analysis of dermatoscopic images must be conducted to differentiate the lesions accurately. The 'ABCD' rule, described by Stolz. W in 1994 (Stolz et al. 1994), was the first dermoscopy algorithm created to help the public and clinicians identify features that help to quantitatively differentiate benign from malignant or suspicious lesions in a dermatoscopic image. The four criteria this algorithm consists of are asymmetry (A), border (B), colour (C), and differential structures (D). These memorable criteria are scored individually, from which a total dermoscopy score (TDS) can be calculated using a linear equation, and the lesion can be classified.

The asymmetry score is generated by bisecting the lesion in its most symmetric plane with two axis perpendicular to one other, as shown in Figure 2. If there is no asymmetry with respect to the lesion's structure or distribution of colour (i.e. the structure and colour is symmetrical), then the asymmetry score shall be 0. If asymmetry is present in only one axis, the asymmetry score is 1, and if there is asymmetry in both axis, the score is 2. To evaluate the border score, the lesion is first divided into eight as shown in Figure 3. For each section, if the pigment pattern at the edge of the lesion has an abrupt cut-off, 1 is added to the border score. No score is added for sections where there is no distinct cut-off, and the pigment pattern appears to blend smoothly into the surrounding skin.

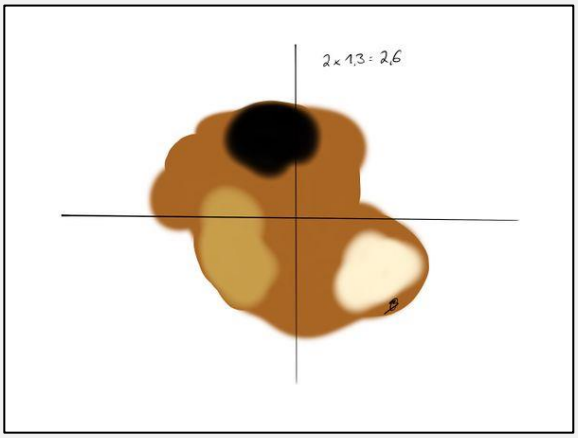


Figure 2: ABCD asymmetry schematic (Ralph Braun 2017, licensed under CC BY-NC-SA 4.0)

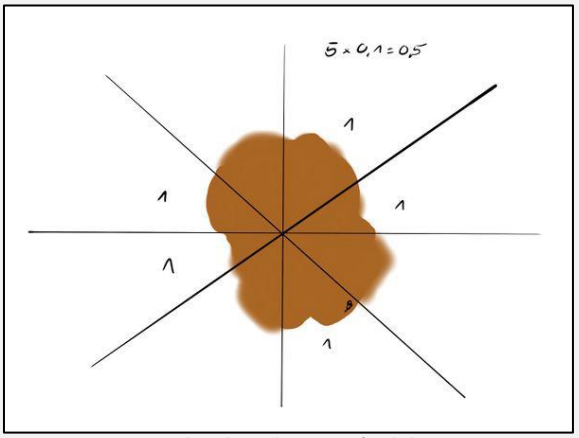


Figure 3: ABCD border schematic (Ralph Braun 2017, licensed under CC BY-NC-SA 4.0)

The colour score is determined by counting the presence of six particular colours: white, red, light brown, dark brown, blue-grey, and black. For each colour present in the lesion, 1 is added to the colour score. The dermoscopic structure score is generated by counting the presence of five particular dermoscopic structures: pigment network, branched streaks, dots, globules, and structureless areas. 1 is added to the score for each structure present.

The scores asymmetry (A), border (B), colour (C), and dermoscopic structure (D) are multiplied by the coefficients 1.3, 0.1, 0.5, and 0.5 respectively to produce a TDS that lies between 1.0-8.9. A TDS of 4.74-5.45 indicates a suspicious lesion and a TDS larger than 5.45 suggests a high likelihood of melanoma.

$$\mathrm{TDS} = (\mathrm{A}\cdot 1.3) + (\mathrm{B}\cdot 0.1) + (\mathrm{C}\cdot 0.5) + (\mathrm{D}\cdot 0.5)$$

This technique claimed to increase diagnostic accuracy to 80% from 64.4% with the naked eye (Stolz et al. 1994). While there are more modern algorithms that may be more appropriate for professional dermatologists, such as the 7-point rule (Argenziano et al. 1998), C.A.S.H (Kopf et al. 2007), CHAOS and clues (Rosendahl et al. 2012), or other kinds of pattern analysis, the ABCD rule provides a simple and encompassing platform for lesion differentiation that can be easily replicated by a computer.

2.2 Computer-Aided Diagnosis

Computer vision and artificial intelligence have been combined with radiology and pathology in clinical settings for over 40 years to support a doctor's diagnosis (Oakden-Rayner 2019). These computer-aided diagnosis (CAD) systems help the doctor interpret medical images faster, and are highly valued in time-sensitive clinical scenarios where imaging techniques such as X-ray or MRI produce large amounts of data that need to be analysed comprehensively.

Early diagnosis is crucial for skin cancer, and increasing incidence rates means that professional human resources and equipment needed for a diagnosis may not be available for every patient. Computer-aided diagnosis (CAD) systems may become this problem, alleviating the dermatologist's workload, and helping achieve a faster diagnosis. A CAD system for skin cancer can analyse the dermoscopic features in a lesion image according to a dermoscopic algorithm, and can attempt to diagnose the lesion if given enough information.

Currently the real-world applicability of CAD for skin cancer is unknown, as the technology only beginning to be used in this particular clinical setting (Topol 2019). There are many challenges slowing the adoption of this technology, including: dataset bias, dataset availability, and the need for a transparent framework from which to assess the quality of a CAD system and the training dataset (Varoquaux and Cheplygina 2022). Despite this, many so-called 'state-of-the-art' CAD systems for skin cancer boast similar diagnostic accuracy when compared to dermatologists (Liu X et al. 2019). Many people look forward to using this technology to obtain a faster diagnosis, enhance patient care, and perhaps one day providing a complete and reliable diagnosis at home.

Most CAD systems, including those for skin cancer, follow this generic pipeline:

1. Image acquisition
2. Pre-processing
3. Segmentation (Region of Interest detection)
4. Feature Extraction
5. Classification (Diagnosis)

2.2.1 Datasets

To be able to perform accurate automatic diagnosis with machine learning in a CAD system, lots of images are needed for training the system's classification model. Historically, creators of CAD systems for skin cancer would have to collaborate with a hospital in order to obtain any meaningful set of skin lesion images. Although primarily used for educational purposes, publicly available dermatology atlases have more recently been used for algorithm training. Sourcing images this way circumvents the need for any researcher to procure the images themselves, along with the financial and regulatory barriers to obtaining such large databases of high-quality lesion images. Large, publicly available datasets are also useful for comparing the effectiveness of different algorithms directly on the same dataset. Because of the increasing interest in CAD systems for early diagnosis, a number of datasets have been made available to the public, in attempts to help progress the technology as much as possible. Nowadays there are many archives of skin lesion images available on the internet, illustrated in Figure 4.

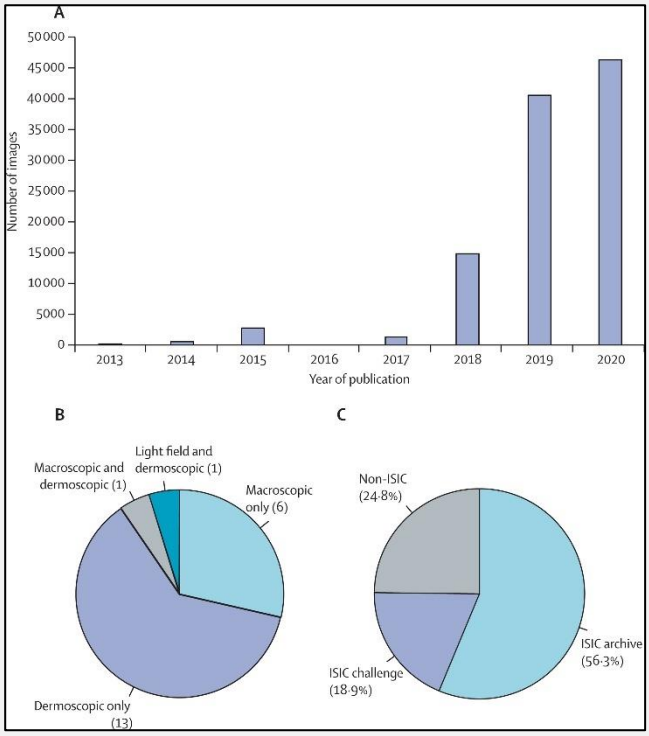


Figure 4: Image publication year, imaging modality, and image source of open access datasets (Wen 2021, licensed under CC BY 4.0)

The International Skin Imaging Collaboration (ISIC) are currently the leading source of dermoscopic images for use in machine learning and medical image analysis research. The academic and industry partnership has created a repository of publicly accessible open-source skin images under Creative Commons licenses, along with gold-standard lesion diagnosis metadata (ISIC 2022). This invaluable resource aims to help reduce melanoma mortality through facilitating the application of digital skin imaging. Since 2016, ISIC has hosted and sponsored annual challenges for CAD skin cancer systems to improve automated dermatologic diagnosis by engaging the computer science community through competitive algorithm performance benchmarking using the ISIC archive. The Human Against Machine 10000 (HAM10000) dataset consists of 10015 dermatoscopic images to be used for academic machine learning purposes, publicly available through the ISIC archive and prevalently used for ISIC challenges for the purposes of improving automated diagnosis (Tschandl et al. 2018). The images included in the dataset are sourced from Austria and Australia.

2.3 Image Acquisition / Pre-processing

The process of acquiring lesion images using a dermatoscope often introduces various visual artefacts captured in the image, such as medical markings, rulers, and the patient's hair, shown in Figure 5. These artefacts can occlude information needed to make a diagnosis. For the purposes of a visual examination by eye, these artefacts are less disruptive, as the dermatologist could physically remove the obstructions before retaking the photo if necessary. Alternatively, and for the purposes of a CAD system, these artefacts can be extracted and removed with software such as DullRazor (Lee et al. 1997).

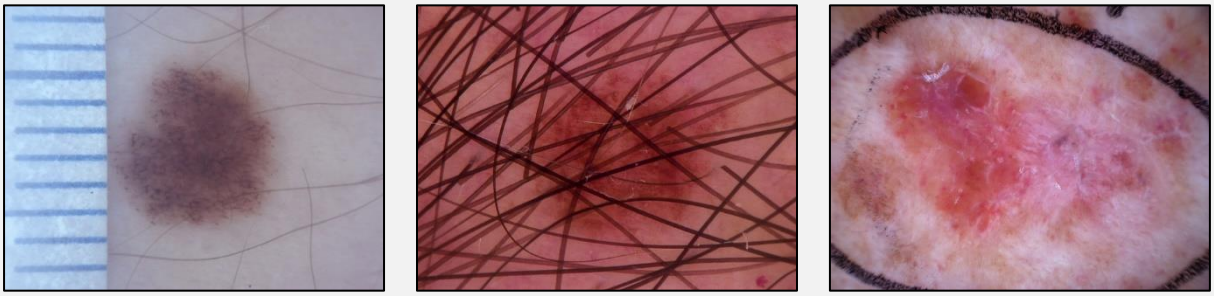


Figure 5: Examples of skin lesions from the HAM10000 skin lesion dataset (Tschandl et al. 2018, licensed under CC BY-NC 4.0)

There are also other characteristics of dermoscopic images that can present issues for automated diagnosis. One problem that frequently occurs in dermoscopic images is a change in the image brightness towards the edges of the image as the natural curvature of the patient's anatomy surrounding the lesion location means the skin moves further away from the lens of the dermatoscope, illustrated in Figure 6. While this characteristic would not usually obstruct a dermatologist's visual examination, it can make it harder for the computer to define the regions of interest within the image. For a CAD skin cancer system, this is typically solved in image pre-processing stages by using contrast enhancement techniques such as histogram equalisation (Schaefer et al. 2011).

There is also the problem of black borders present on many dermoscopic images, as a product of the image acquisition process as shown in Figure 6b and 6c. Again, while this may not affect the work of a dermatologist's visual examination, it can adversely affect the computer's ability to interpret the image properly. Black borders are typically removed from dermoscopic images during the image pre-processing stages of a CAD system.

Figure 6: Examples of skin lesions from the HAM10000 skin lesion dataset (Tschandl et al. 2018, licensed under CC BY-NC 4.0)

2.4 Segmentation

In order for a computer to make any meaningful analysis of any object in the image, it must first know where the object is, hence the region(s) of interest must be defined using segmentation. Image segmentation is a process in which a digital image is partitioned into subgroups called image segments or regions, helping to reduce the complexity of the image, and making further processing or analysis of the image simpler. Techniques such as thresholding, clustering, or edge detection are used to define where a region starts and ends; these techniques detect boundaries within the image based on changes in colour, brightness, or texture. Afterwards, pixels either side of the boundary are assigned labels according to the region in which they lie, thus, the grouped pixels are homogenous with respect to some computed property.

2.4.1 Thresholding

One of the methods of segmentation pertinent to this project is thresholding. Binary thresholding is the simplest method of segmentation. The intensity of each pixel $I_{i,j}$ in the image is examined and compared with a threshold value T. If $I_{i,j}$ is lower than T, the pixel is turned to black, and if $I_{i,j}$ is greater than T, the pixel is turned to white, resulting in a binary image as shown in Figure 7. This method's effectiveness suffers in images that are noisy and have non-uniform lighting.

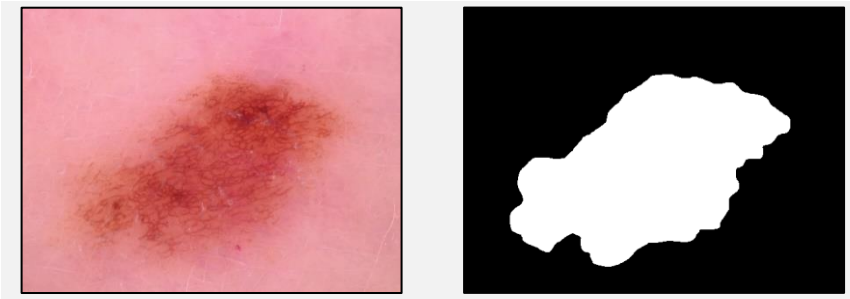


Figure 7: Example lesion image and corresponding binary mask from HAM10000 dataset (Tschandl et al. 2018, licensed under CC BY-NC 4.0)

The threshold value T can be manually chosen, or can be selected automatically by an algorithm such as Otsu's method which exhaustively searches for a threshold that minimises intra-class intensity variance (Otsu's method 2022). Thresholding can be applied 'globally' – to the whole image, or 'locally' by adaptively selecting and applying thresholds in different sections of the image, which can perform better than global thresholding in uneven lighting. Thresholding can also be performed on individual colour channels, which tend to hold more useful information than a grey-scale image. There also exist methods where multiple increasing thresholds T_n are performed, resulting in an image with N classes instead of 2 (binary) classes.

2.4.2 Edge detection

Another popular method of segmentation also pertinent to this project is edge detection. This method operates under the assumption that discontinuities in the image's brightness likely correspond to changes in depth, surface orientation, material properties, or illumination. The change in intensity can then be used to define the edges of objects. In the case of the proposed system, the edges of hairs can be identified for removal. The Sobel operator, presented in 1968 by Irwin Sobel and Gary Feldman, convolves an image with two 3x3 kernels for detecting horizontal and vertical changes in image intensity (Sobel operator 2022). This filter is still popular for edge detection algorithms in image processing and computer vision. The Canny edge detector was developed in 1986 by John Canny, and is an adaptable edge detection algorithm that is still considered state-of-the-art to this day. This algorithm finds the intensity gradients of the image using 4 filters to detect horizontal, vertical, and diagonal edges, before thinning the detected edges with non-maximal suppression and further filtering of weak edges to account for noise (Canny 1986).

2.4.3 Morphological Transformations

Morphology is used in image processing, and in the proposed system, to remove imperfections present in a binary image after thresholding or edge detection. A structuring element is defined and placed at each possible location in the image. At each location, the pixels within the structuring element are adjusted based on the values present in their neighbourhood. The structuring element can be of a certain shape that is more sensitive to particular shapes in the image. There are a multitude of morphological operations for different purposes. Dilation and erosion and the most basic operations, and add or remove pixels to object boundaries respectively, illustrated in Figure 8.

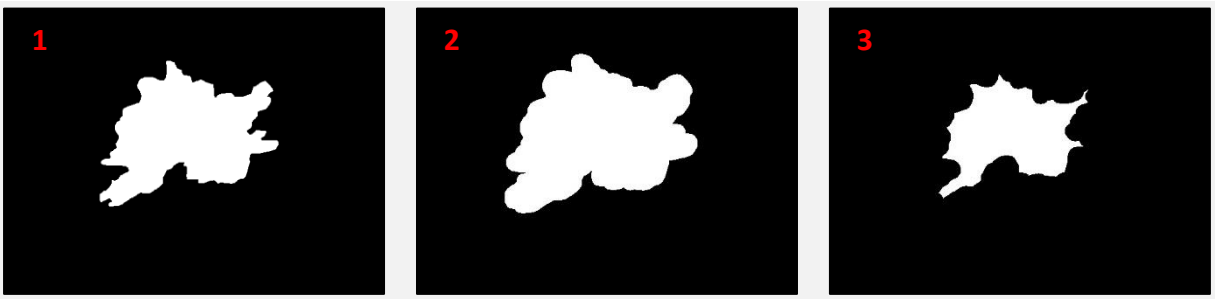


Figure 8: A curated binary mask(1), its morphological dilation output(2), and its morphological erosion output(3) kernel size 21 (Tschandl et al 2018, licensed under CC BY-NC 4.0).

2.5 Region Description

Given an appropriate segmentation, analysis of the regions of interest (ROIs) can be made to generate data that describes the regions. Information about the region's shape, colour and texture can be extracted and used to create useful descriptors for a ROI.

Calculating the perimeter of a region is as simple as counting the pixels along a region's boundary, and counting the pixels inside the region gives the area. To describe the region further, it can be useful to calculate the minimum bounding rectangle and convex hull, as well as their area and perimeter as shown in Figure 9. From these measurements, more meaningful descriptions of the regions can be calculated such as the region's compactness, convexity, and rectangularity. These measurements benefit from being invariant to (unaffected by) the region's scale, rotation, and translation, making them appropriate data to generate in a CAD system for comparing regions of interest and differentiating them from one another.

Figure 9: A curated binary mask with its minimum bounding rectangle, and its convex hull in purple (Tschandl et al 2018, licensed under CC BY-NC 4.0).

2.5.1 Image Moments

Another method for quickly generating meaningful descriptions of a region is by calculating the *image moments* M_{ij}, which are weighted averages of the pixel intensities in a greyscale image calculated using the equation: $M_{ij} = \sum_x \sum_y x^i y^j I(x, y)$

Of the basic moments, a region's area can be calculated using M_{00}, and the region's centroid can be defined as: $\{\bar{x}, \bar{y}\} = \left\{\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}\right\}$

Various moments can be calculated that have useful properties. Central moments are invariant to translation, and Hu showed that moments can be constructed that are invariant to translation, scale, and rotation (Hu 1962). Shape measurements such as circularity can also be derived from the region's moments, as described by Žunić, Hirota and Rosin (Žunić et al. 2010).

2.6 Colour Analysis

In colour images, analysis of a region or pixel's colour can be useful for segmenting and describing a region. Segmentation using colour is called colour clustering, and assumes that homogenous colours represent objects, and splitting the image by groups of colour can be used to define the boundaries between objects.

The properties of different colour spaces and their individual channels can be helpful in revealing more specific information from the region of interest. In a dermoscopic image for instance, a skin lesion is usually represented more intensely in the red colour channel of the RBG colour space, as shown in Figure 10.

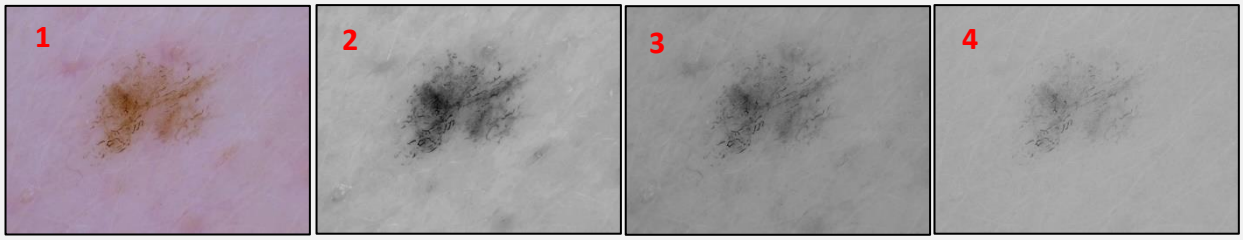


Figure 10: An RGB lesion image(1) from the HAM10000 dataset and its corresponding Red(2), Green(3), and Blue(4) channels (Tschandl et al 2018, licensed under CC BY-NC 4.0)

The RGB colour space represents colours based on the mixing of the three primary colours red, green, and blue. This colour space operates based on the human eye, which is sensitive to the wavelengths associated with the RGB colours. The primary colours are chosen as to stimulate the eye's three photosensitive cones as independently as possible (RGB color space 2022).

The CIEL*a*b* colour space uses a different method of creating colour. The L* channel is used for lightness, and the a* and b* channels are used for red, green, blue, and yellow colours.

For the purposes of a CAD system for skin cancer, analysis of the lesion's colour should be performed as part of differentiating one lesion from another, since the distribution of colours in a lesion is key information pertaining to a diagnosis. This will include analysis over several individual channels in different colour spaces.

2.6.1 Histogram

A colour histogram is one way of representing the colours present in an image, which shows the statistical distribution of colours and the overall tone of an image. The histogram is formed by counting the number of pixels of each intensity 0-255 in each channel. The histogram can be examined to reveal characteristics of the colour in the image.

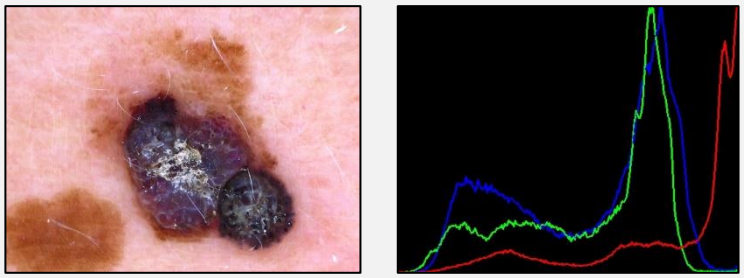


Figure 11: A lesion image from the HAM10000 dataset and its RGB histogram (Tschandl et al. 2018, licensed under CC BY-NC 4.0)

2.7 Texture Analysis

Similarly to colour analysis, analysis of texture in an image can be useful for describing a region. In image processing, texture is considered as the spatial changes of the image pixel's intensities. Particularly in dermoscopic images, the texture of the lesion is key information and can be used to differentiate lesions for diagnosis.

2.7.1 GLCM

One popular method of analysing texture is by using grey-level co-occurrence matrices (GLCM). This method involves comparing the grey-level intensities of two neighbouring pixels and constructing a matrix to store this information. From this matrix, various measures can be extracted including correlation, contrast, energy, homogeneity.

2.8 Skin Type

The Fitzpatrick skin type is a grouping of skin tones into six categories based on the amount of melanin in the skin, and is used to estimate "tanning ability" or risk of disease due to sun exposure. The individual typology angle (ITA) is an estimation of skin tone based on the L* and b* values in a CIEL*a*b* colour image proposed by Chardon et al. in 1991. The ITA value of an image is found using the equation: $\mathrm{ITA}° = \frac{\arctan(L^* - 50)}{b^*(\frac{180}{\pi})}$

Although the two have been used equivalently in the past, it has been shown that they are not equivalent (Osto et al. 2022). Nonetheless, they are still utilised for estimations in photobiologic studies, with caution.

Fitzpatrick skin phototype	**Criteria**
I	Always burn, never tan
II	Always burn, sometimes tan
III	Sometimes burn, always tan
IV	Minimal burns, always tan
V	Rarely burns, always tan
VI	Never burns, always tan

Table 2: Fitzpatrick classification

ITA classification	**ITA° range**
Very light	ITA° > 55
Light	41 < ITA°< 55
Intermediate	28 < ITA° < 41
Tan	10 < ITA° < 28
Brown	-30 < ITA° < 10
Dark	ITA° < -30

Table 3: ITA skin classification (Chardon et al. 1991)

2.9 Classification

Part of a dermatologist's job is to evaluate a patient suspicious skin lesions. After examining the relevant evidence, the dermatologist will try to classify the lesion as a particular condition. In machine learning, classification is the process in which a set of data is categorised. In a CAD system, the categorisation would be based on a diagnosis, or set of possible diagnoses. A classification model is trained on ground truth data, before being able to predict the class of new data.

Binary classification is used to categorise data into two classes, while multi-class classification allows the categorisation into many classes. For the purposes of a CAD system for skin cancer, multi-class classification is favoured, but requires lots of training data and very discerning feature extractors, because lesions of different types can appear very similar to one another. Many CAD systems for skin cancer still classify binarily into malignant or benign, as they support the dermatologist's specific diagnosis.

2.9 Existing Computer-Aided Diagnosis Solutions

As discussed in section 1, there is lots of incentive for the creation of CAD system for skin cancer. There have been great deals of research conducted in this area.

2.9.1 Meta-Analysis

In 2019 a meta-analysis of 70 studies, aiming to review the current literature found that the accuracy of computer-aided diagnosis is comparable to human-experts, however most studies do not represent real-world scenarios and are prone to biases (Dick et al. 2019). The meta-analysis reveals that there is a great need for more standardized and realistic study settings before we are able to fully explore the full potential of CAD systems for skin cancer in clinical practice.

Of the studies included in the meta-analysis, Maglogiannis et al. proposes a set of dot / globule related features that can enhance the performance of classification algorithms that discriminate malignant from benign skin lesions (Maglogiannis et al. 2015a). Since dots and globules appear in both benign and malignant lesions as dark circular structures with differing distributions, Maglogiannis et al. propose segmenting these regions using an algorithm based around non-linear inverse diffusion and creating a set of features based on the dots and globules. The proposed segmentation algorithm achieves 91% and 95% specificity for malignant and non-malignant lesion images respectively. Segmenting this way involves iteratively enhancing the dot and globule structures with inverse non-linear diffusion using differing parameters and returning a circularity function which is then used to segment the structures. The dots / globules that are inside the lesion area are used to create a set of features. The number of dots, the total number of pixels in dots, the mean number of pixels in dots, the variance of number of pixels in dots, and the fraction of the lesion area occupied by the dots are all used as features in the proposed system. The symmetry of the dots' distribution is also analysed, and the following features are calculated: radial asymmetry, angular asymmetry, and asymmetry with respect to the lesion's primary axis. Using the dot features alone, a classification accuracy of 76% was achieved using SVM polykernel. The study also evaluated classification accuracy using the dot features in combination with features that were not dot-related, such as RGB statistics, shape descriptors, and textural features. Using both sets of features improved classification accuracy beyond that of the accuracy using either of the sets alone.

Maglogiannis. I and Delibasis. K also propose 5 hair removal algorithms for dermoscopy images in another of their works, not included in the meta-analysis (Maglogiannis et al. 2015b). Of the proposed algorithms, their bothat based methodology uses a Laplacian filter

to sharpen the image, before using a bottom-hat transformation to increase the contrast of the resulting image, and then performing thresholding using Otsu's method to obtain a hair mask from which the hair pixels in the original image are replaced by interpolation. Their Laplacian based algorithm uses a Laplacian filter to sharpen the image, followed by a Weiner filter to reduce noise. Then the hairs are detected using a Laplacian of Gaussian (LoG) edge detector and the original image is interpolated using the resulting mask after refinement by morphology. An LoG based algorithm is also proposed where a LoG edge detection is performed on the red channel extracted from the original RGB image and the hairs pixels are replaced as before using interpolation. Their Logsobel based algorithm adds the resulting images of LoG and Sobel edge detectors before performing noise reduction to refine the hair mask. Their final proposed algorithm, the Lls based algorithm, operates by strengthening the response of the hairs with a Laplacian filter, which is subtracted from the greyscale input image and fed into further LoG and Sobel edge detectors before noise reduction like before. They compare the performance of their algorithms against one another and against the popular DullRazor software. The authors created images with synthetic hairs that were used as ground truth hair masks for evaluation and achieved accuracies of 80-94% with lower error than DullRazor.

A melanoma detection system based on feature fusion claims 98% sensitivity and 90% specificity using manual segmentations and supervised learning (Barata et al. 2015). This is achieved by extracting only colour and texture descriptors, and no asymmetry or border measures, so as to accommodate for lesion images in which the lesion area is not fully contained. For the global features, colour histograms in three different colour spaces are extracted – HSV, L*a*b and Opponent. As well as amplitude histograms, orientation histograms, and Gabor filters for texture. For the local features, the same methods are used but extracted from regular 40x40 pixel grids sampled from the image containing >50% lesion pixels. When compared to studies that automatically segment the lesion, this study has much higher sensitivity and specificity, indicating that correct segmentation is a significant barrier to an effective CAD system. The feature fusion technique can be performed early during the CAD system, or late. This strategy taken by Barata et al. comprises of two stages of supervised learning. The first stage trains a classifier for each type of extracted feature. The second stage trains a final classifier from the scores of the previous classifiers to predict a diagnosis. However, these maximum classification score were achieved using a single-source dataset (Mendonca et al. 2013), and performance decreased over the multi-source dataset (Lio et al. 2004) to 83% sensitivity and 76% specificity, indicating that there is bias in the methodology towards more homogenous datasets. Nevertheless, the results from the study suggest that the late fusion technique shows promise in CAD systems as multiple descriptors can be used without incurring higher dimensionality in the feature vector.

In 2011, Cavalcanti et al. achieve an accuracy of 96.71% for the classification of skin lesion images taken with standard cameras (Cavalcanti et al. 2011). Their method uses shading attenuation to remove any shadows that could be a barrier to good segmentation. The study segments using an Otsu method-based algorithm and 52 features were extracting based on the ABCD dermoscopy algorithm. The classification was achieved using k-Nearest Neighbours and decision trees. In 2013, Cavalcanti et al. added 12 new features to the

previous study's feature set based on eumelanin and pheomelanin values to achieve a claimed accuracy of 99.34% (Cavalcanti et al. 2013b). An additional 2013 book chapter by Cavalcanti et al. compares a selection of segmentation algorithms: Multi-Direction GVF Snake Method, Alcon et al. Thresholding Method, Thresholding Method on a Multichannel Representation, Otsu's Thresholding Method on the Red Channel, Otsu's Thresholding Method on Grayscale, ICA-Based Active-Contours Method (Cavalcanti et al. 2013a). The authors conclude that the ICA-based active contours method performs most reliably with high average accuracy, with Otsu's thresholding method on greyscale. The features used in this system were the same as in their 2011 publication (Cavalcanti et al. 2011). The classification was done with kNN algorithm with results showing the thresholding technique achieving the highest accuracy.

One study included in the meta-analysis was on automatic detection of melanoma using macroscopic images taken by a conventional digital camera (Ramezani et al. 2014). Instead of using dermoscopic images taken with a dermatoscope, macroscopic images were used to see how effective CAD is for diagnosis when a dermatoscope is not available, as they are not widely available to purchase for cheap. The images are pre-processed using a median filter followed by a sampling and redistribution of pixel intensity in the Value channel of the HSV colour space, in order to counter the effect of nonuniform illumination and shadows that present more dramatically in macroscopic images when compared with dermoscopic images. Colour, intensity, and texture information are extracted using the RGB red colour channel, the CIEluv l channel, and application of the PCA method respectively. The histograms for each the single-channel images are examined, and the image with the maximum distance between the two peaks, corresponding with the lesion and the healthy skin, is selected for thresholding. Four threshold values are calculated across the histogram data of the selected image, and the two that are closest to each other are selected for thresholding. Hairs are removed using bottom-hat and closing morphological transformations before the largest object in the image by pixel count is selected as the lesion mask. Feature extraction is based on the ABCD dermoscopy algorithm, with a total of 187 features: 32 for asymmetry, 34 for border irregularity, 72 for colour, 7 for diameter, and 42 for texture. The study claims significant improvements in the diagnostic accuracy compared to a naked eye examination where the specialist is claimed to have achieved 64% accuracy. Classification was performed using SVM, the authors achieved an accuracy of 81% using all the features, and after application of PCA for the selection of optimal features, an accuracy of 82.2% was achieved using only 13 of the original 187 features.

Ganster. H et al. achieve a classification sensitivity 87% and specificity 92% (Ganster et al. 2001). Their method involves applying a grey-scale morphological closing operation to reduce the hairs appearing as thin dark elongated structures. The resulting images are segmented using dynamic thresholding on the blue channel of RGB colour model and the b channel of the CIE-lab colour model., and 3-D colour clustering. A strategy of segmentation fusion made the algorithm more robust, and 122 global and local lesion features were calculated, which was reduced to 21 features after feature selection.

As should be apparent, there is a great deal of research in the problem area, spanning decades. It is still yet unknown how relevant each study will be for achieving gold-standard classification etc. While many studies report high accuracy, there is little framework for assessing a CAD systems real-world effectiveness, as well as the relevancy of datasets used in the studies.

3 Approach

In order to develop and evaluate an effective CAD system for skin cancer, a clear outline of the system's requirements is needed.

3.0 System Overview

Working backwards from the solution's end – to be able to classify a skin lesion as malignant or benign, there are a number of precursor stages that must first be implemented. To make a diagnostic prediction, the system must be able to assess the characteristics of the skin lesion in question and evaluate it based on a clinical dermoscopic algorithm. To do this, a set of features that describe the lesion in a clinically relevant way must be created. To differentiate the lesion in question from other lesions, and suggest if it is malignant or not, the system needs to 'know' what a malignant and benign lesion looks like. Therefore, a model that can recognize the patterns of malignant and benign lesions must be trained before being applied to the new data. To train the model, the system must analyse a large number of lesion images where the diagnosis has already been established – the ground truth, and create sets of features that describe each one. This way, when a new lesion image is shown to the system it can compare its characteristics with any number of previously seen lesion images. The more similar the new lesion image is to an old one, the more likely it is to be similar diagnostically. After establishing that a diagnosis can be suggested by comparing the features of an unknown lesion to the features of known lesions, the system must be able to first extract the features from a lesion image reliably. To do so, the region of interest must be located within the image, in this scenario the region of interest is the lesion area. For the system to locate the lesion area reliably, some pre-processing of each image must also be performed. Removal of any hairs, black borders, rulers, and other clinical markings is of interest.

3.1 Requirements

The CAD system should be capable of:

1. Acquiring lesion images from a dataset
2. Performing black border and vignette removal on lesion images
3. Performing hair removal on lesion images
4. Segmenting ROIs in a lesion image
5. Evaluating segmented ROIs against available ground truth
6. Extracting features from a ROI
7. Addition / removal of features as needed
8. Storing training data in a file
9. Training a classifier using extracted features
10. Classifying an unseen skin lesion using a training classifier
11. Evaluating the trained classifier
12. Processing more than 20,000 images efficiently

3.2 Tool and Libraries

To be able to develop a system that meets the aforementioned requirements, it was important early on to select the tools and libraries that would enable the development of these particular functional requirements. This section describes why the tools and libraries were chosen.

The project was to be written in the Java language because Java is a relatively simple programming language and the author had experience using it, making the development process efficient. Java is platform-independent and highly portable, meaning that deployment across different systems would be easy, as well as being simple to convert to an Android application so that the system can be used as a dermoscopy skin checking app. Java also has a good selection of popular and open-source computer vision and machine learning tools and libraries.

To be able to perform effective image processing for the following requirements: 2, 3, 4, 5, and 6, it is necessary to carefully select an appropriate library for the job. MATLAB was one candidate considered for the job of image processing; MATLAB is a programming language and computing platform for data analysis, algorithm development, and model creation, and also has bindings that can be called from Java (MATLAB 2022). The Image Processing Toolbox from MATLAB provides image pre-processing analysis, segmentation, and deep learning functionality. MATLAB's Computer Vision Toolbox provides functionality for feature detection and extraction, as well as semantic segmentation. One benefit this toolbox has is that algorithms can easily be accelerated by running them on the GPU. MATLAB was not picked because it is not open source, and lacks good documentation. Fiji was another candidate for image processing in the project. This open source framework provides software for image processing and analysis, along with many community-contributed plugins (ImageJ Wiki 2022). OpenCV was the final candidate for image processing and was chosen because it is open source, available for multiple languages, and has the best documentation due to its popularity. It provides functions for all the important computer vision and image processing tasks, while remaining fast and easy to use. It also has object detection and tracking functions for Android apps, in the case of deploying the CAD system on a mobile phone.

To be able to train a classifier and use it on new data for requirements: 7, 8, 9, and 10, an appropriate machine learning library was needed. JavaML is an open source framework for machine learning and data mining, which provides algorithms for pre-processing, feature selection, classification, and clustering (JavaML 2022). JavaML has good documentation, but does not include any GUI for easy experiments. Weka on the other hand is probably the most popular machine learning library for Java, and includes many functions useful for the project including data pre-processing, visualisation, feature selection, and classification (Weka 2019). Weka is open source, well-documented, and also comes with a handy GUI as well as Java API. Hence, Weka was chosen to be used in the project to enable satisfying the necessary requirements.

3.3 Dataset Selection

The selection of a dataset was an important consideration in the project. The dataset's images will be used to build, train, and evaluate the system, and as such care was taken to select a dataset that provided the ability to satisfy all the requirements.

The dataset used for the project during development and training was the HAM10000 dataset. This dataset was chosen because it comes with high quality metadata, as well as curated ground truth segmentations (Tschandl. et al 2018). The size of the dataset was also a consideration; the HAM10000 dataset includes 10015 images, which provides plenty of training data, while remaining small to download. The included images were of size 600x450, meaning they were relatively fast to process without downscaling and losing potentially useful data. The dataset also has good enough representation of different diseases, providing a good basis for creating a CAD system that can generalise enough to account for the differences. Additional datasets were added later on for more validation and training.

3.4 System Design

Based on the standard pipeline used for CAD systems, the pipeline for the proposed system includes three core stages to building the classification model in the proposed system. The operation will be run once on each image in the training dataset to generate the sets of features describing already diagnosed lesions to feed the classification model. To classify an unseen lesion, the operation will run once on the new lesion image from the validation or test dataset to generate the set of features describing it, before the classifier compares it to the training data and makes a diagnostic prediction.

- Phase 1: Pre-processing – obtrusive visual artifacts will be removed
- Phase 2: Segmentation – the lesion area will be located
- Phase 3: Feature Extraction – data is calculated to describe the lesion

Once the feature sets for each image in the training data have been generated to train the classification model, and the feature set of the new unseen lesion has been generated, the system can proceed to make a judgement on its diagnosis based on the classification outcome.

- Phase 4: Classification – the model will predict the unseen lesion as benign or malignant

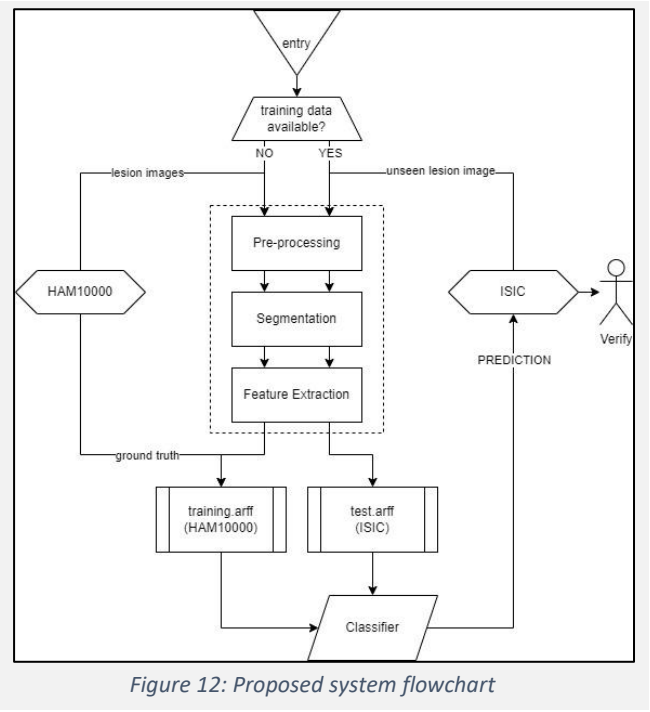


Figure 12: Proposed system flowchart

3.4.1 Pre-processing

Due to the images being taken with different methods, at different locations, and using different equipment, it is beneficial to prepare the image with some generic pre-processing in order to make the following stages easier. As stated, pre-processing is performed on every image in the training set, as well as on each image in the validation or test sets, meaning the image processing must be generalised and versatile to cope with variations between images.

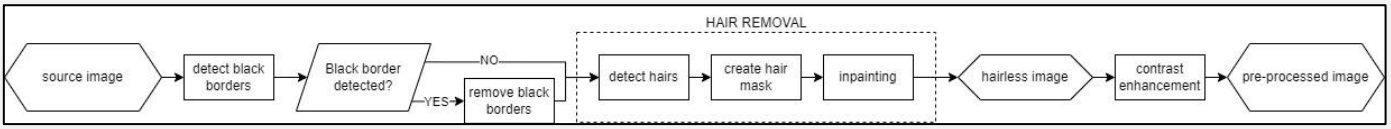


Figure 13: Proposed pre-processing flowchart

Various artefacts captured during image acquisition such as hairs, medical markings and black borders can adversely affect how well the later stages perform (see section 2.3, Figure 5). Occlusion of the lesion by these artefacts increases the difficulty of obtaining a correct segmentation automatically and therefore also, increasing the difficulty in accurately calculating its features. In order to mitigate the negative effects of these artefacts in later stages, hair removal, black border removal, and contrast enhancement is to be performed. Since hairs typically appear in lesion images as intense, dark, elongated structures, to programmatically remove the hairs from the dermoscopic images, first the system will detect these strong responses using filtering and edge detection algorithms, and create a binary mask for the detected structures. From this binary mask, the source image will be repaired by removing the hairs. This pre-processing will also detect and removes clinical markings and rulers present in a lesion image. There are a number of methods commonly used to remove the black borders from the dermoscopic images. Since the black borders

created by the dermatoscope typically appear as full-black circles towards the image edges, the method proposed in this system will use a circle detection algorithm to detect the black borders and replace the pixels in the border so as to remove the negative effect of the dark areas.

3.4.2 Segmentation

When it comes to locating the region of interest – the lesion area, the system needs to segment a lesion image reliably and robustly. This is because thousands of images from varying datasets will be processed, and each lesion can appear differently on the skin. The quality of interpretation of a lesion image depends heavily on the segmentation process, since many features will be calculated directly from the binary mask produced by the segmentation method. The algorithm will have to cope with uneven lighting, varying lesion shape and colour, as well as ensuring that it does not over or under-segment the lesion.

Since the lesions appear as areas of intensity different than that of the surrounding skin, thresholding will be performed to locate the region of interest. Particularly, adaptive thresholding methods will be used as they are robust to uneven lighting that is present in many lesion images. It is also beneficial to apply a medium-sized median blur to the image before thresholding, to reduce the adverse impact of noise in the image. Additional morphology will be used to refine the segmentation produced by thresholding, as to create a binary mask with smoother edges that represents the shape of lesion better.

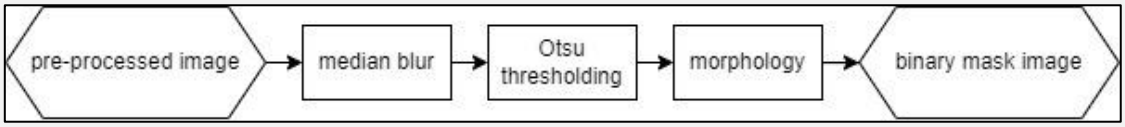


Figure 14: Proposed segmentation flowchart

3.4.3 Feature Extraction

To correctly differentiate between lesions, the system needs to analyse the lesion image using methods that well-represent a real-world physical examination. Discriminating between lesions of different types in a clinically accurate way requires analysis of the lesion across multiple facets according to a dermoscopy algorithm. The features extracted from the lesion image need to be meaningful in order for the lesions to be compared effectively. The features also need to be fast to compute as thousands of images will be processed, each with their own set of features. The features need to also be robust against the limitations of the datasets; dermoscopic images taken with different devices can represent the same lesion slightly differently in terms of colour intensity and apparent size. The features should also be normalised to 0.0-0.1, to reduce complexity in the system.

The ABCD of dermoscopy provides a foundation for classifying lesions based on their physical properties. The proposed system will mimic the ABCD rule to extract the features that are clinically relevant. The following properties of lesion images will be assessed: asymmetry (A), border shape (B), colour (C), and differential structures (D). The asymmetry index will be analysed from the binary mask that represents the lesion area. The border of the lesion will also be analysed using the binary mask; shape measures such as circularity,

compactness, convexity, elongation, and rectangularity will be used to assess the lesion's shape, and the structure of the lesion's border will be assessed using measures such as fractal dimension and radial variance. The colour and texture of the lesion will be analysed from the source image while the surrounding skin is hidden using the lesion's binary mask using GLCM and histogram data. The colour of the surrounding skin will also be analysed using the inverse of the lesion's binary mask.

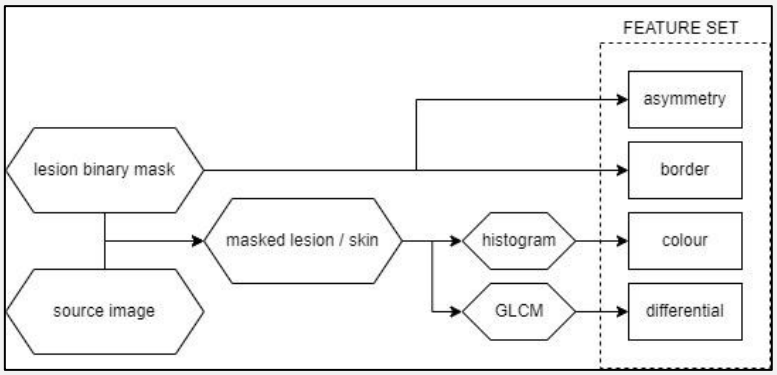


Figure 15: Proposed feature extraction flowchart

3.4.4 Classification

To be able to make a prediction on an unseen lesion, a classification model must be trained. The ground truth diagnoses provided as metadata with the dataset will be grouped so that we can train a model and classify new data binarily. The groups will simply be 'malignant' and 'benign'. Weka provides an easy way to train machine learning classification models from data stored in a text file. The system will write the data generated during the feature extraction stage, as well as the ground truth from the dataset's metadata to a Weka ARFF comma-separated text file for each image in the training set. From this file, the system will use Weka functions to train a machine learning classification model such as K-Nearest Neighbours, Random Forest, or a Bayesian Network. From this newly created model, the system will now be able to classify new data. The new data to be classified is written to another ARFF file. This time, containing no ground truth. The system can then apply the model to this new data and predict the diagnosis. Cross validation will also be used to evaluate the accuracy of classification without needing unseen data, which holds out portions of the training set for testing. For each fold of a 10 fold cross validation procedure, $1/10^{th}$ of the data will be held and evaluated using the other $9/10^{ths}$ as training data, with a different $1/10^{th}$ being held out each fold.

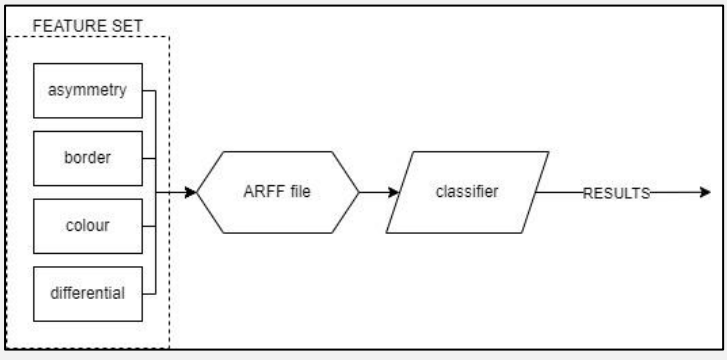


Figure 16: Proposed Classification Flowchart

3.5 Ablation Study

Of the features calculated by the system, some will be more relevant for obtaining an accurate diagnosis than others. It should be considered that some features may not be helpful at all for discriminating one lesion from another, and it would be useful to know which features are contributing and which are not. Other works in this problem area sometimes use a very small number of features. To evaluate the efficacy of certain components in the system, and to find out which features are most relevant to classifying a lesion image, an ablation study will be conducted. This will involve training and testing the classification model on only subsets of the features generated. This method of evaluating the system can also give insight into the system performs on different datasets, and some particularities of the dataset itself. If training and classifying using fewer features, one would expect the classification performance to degrade for all datasets. If this is not the case, that would mean that there is a difference in how each feature extractor performs on different datasets, either due to biases in the dataset or in the way the feature extractor operates. This information can help to refine a system and make it more efficient during future revisions and incremental improvements.

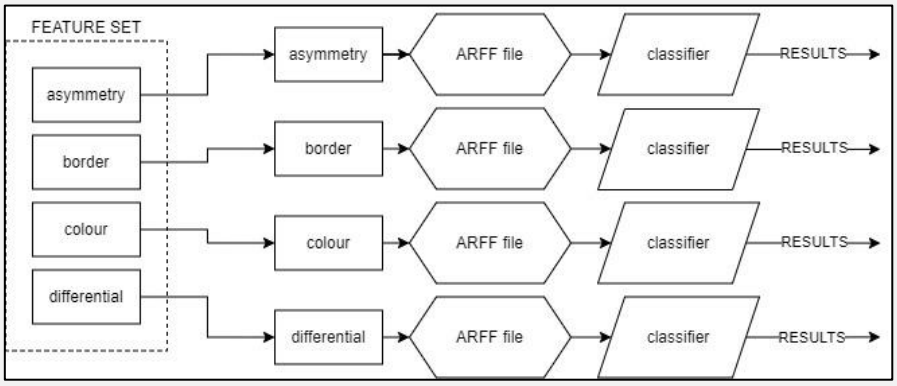


Figure 17: Proposed ablation flowchart

3.6 Skin Type Analysis

Any dataset can be prone to bias. For skin lesion datasets, the individual's skin tone is present in each image. An individual's skin tone is a product of their melanocytes, which can produce different amounts or kinds of melanin, and is directly linked the individual's risk of skin cancer. This unavoidable quality of dermoscopic images can lead to bias in a dataset due to the demographic or geographic locations from which the images are sourced. An effective CAD system should be able to classify skin cancer equally successfully for any skin tone group, but this is difficult to achieve when public datasets are usually sourced from narrow samples of the total geographic population. To discover effectiveness of classification against different skin type groups. The dataset will by split by the calculated skin type (ITA) to create new datasets for each group, and the classification results for each individual group will be compared. This method is only an estimation; as described in section 2.8, ITA and Fitzpatrick skin type are not equivalent, but any difference in performance between groups can still indicate a bias.

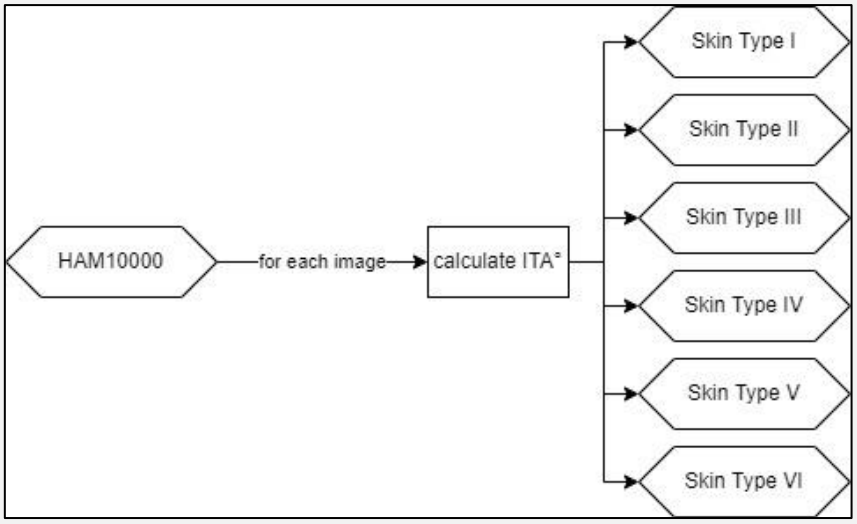


Figure 18: Proposed skin type separation flowchart

4 Implementation

This section provides details on the technical implementation of the CAD system for skin cancer.

Because each stage in the system requires good performance on each of the previous stages, development was iterative at each stage until satisfactory results were achieved. As stated previously, initial development of the CAD system started only with the HAM10000 dataset in mind, which influenced development in various ways. For instance, black borders were not as prevalent as in other datasets, and as such, this particular problem, among others, was not solved until later.

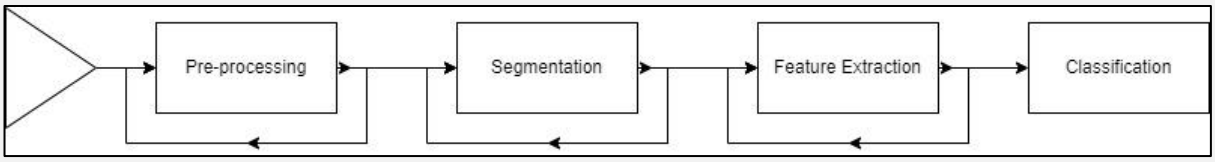


Figure 19: Workflow flowchart

4.0 Image Acquisition

The dataset comes as a zip file of 10015 .jpg images, along with a metadata file including diagnosis. Curated segmentations for every image were also included as .png. The images were stored on a personal hard drive, as they only took up 2.57GB of space, and as such did not need any particular large-scale database storage.

OpenCV provides an Imgcodecs.imread() function to read a .jpg or .png image from a file and store its data in a matrix (Mat), from which further processing can be done. OpenCV also provides an Imgcodecs.imwrite() function to write the matrix back to an image, and a HighGui.imshow() function to display the matrix to the user, with any new changes that had been made. Figure 20 shows how the system loads an image, converts it to grayscale, and shows the resulting image to the user, as well as writing it to a file. From this point, development on the initial image pre-processing was initiated in earnest.

```
// read source image to matrix
Mat src = Imgcodecs.imread("E:\\FYP Dataset\\HAM10000\\HAM10000_images_part_1\\ISIC_0024516");
// create new matrix
Mat gsc = new Mat();
// convert src image to grayscale and put in new matrix
Imgproc.cvtColor(src, gsc, Imgproc.COLOR_BGR2GRAY);
// show the image to the user
HighGui.imshow("grayscale", gsc);
// write the image to a file
Imgcodecs.imwrite("E:\\FYP Dataset\\output\\grayscale.jpg", gsc);
// show image until user input
HighGui.waitKey();
```

Figure 20: Basic OpenCV functions

4.1 Pre-processing

Hair removal was the first problem to solve, as the HAM10000 dataset features many images with hairs occluding information that made the segmentation process nearly impossible.

As stated in section 3.4.1, edge detection algorithms were the primary candidate for detecting the hairs within the lesion image. This is because most hairs appear as thin, dark areas that contrast heavily to the surrounding skin, intuitively providing an abrupt change in brightness for the edge detection methods to find. Initial tests were conducted to see which edge detection techniques could reliably detect the hairs, whilst also not detecting other features in the image such as dark spots or streaks in the lesion or surrounding skin. Because dermoscopic images are taken with high magnification, there can be lots of noise present in the images, which the edge detection algorithms were detecting.

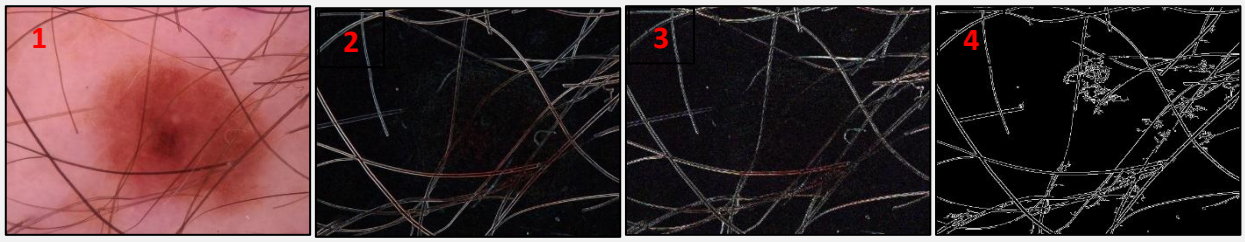


*Figure 21: Lesion image(1) from HAM10000 and Sobel(2), Laplacian(3), and Canny(4) edge detection outputs **without** blurring or denoising (Tschandl et al. 2018, licensed under CC BY-NC 4.0)*

Thus, experiments were conducted using gaussian blur and median blur to see which could weaken the effect of noise whilst retaining enough information in the image for detection of the hair. At this stage, a gaussian blur with kernel size 3 alongside a denoising filter with kernel size 7 seemed to be helping the most. Sobel, Canny, and Laplacian edge detectors were tested, however without much luck; Figures 21 and 22 show that the edge detection algorithms struggle to give good responses to all the hairs, particularly over the lesion area.

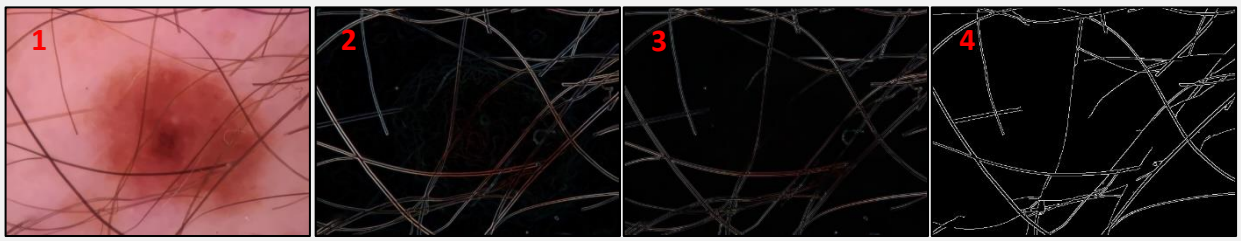


*Figure 22: Lesion image(1) from HAM10000 and Sobel(2), Laplacian(3), and Canny(4) edge detection outputs **after** blurring and denoising (Tschandl et al. 2018, licensed under CC BY-NC 4.0)*

The DullRazor dermoscopy hair removal program (Lee at al. 1997) performs hair detection using a morphological closing operation, so this technique was also experimented with at this stage. Unfortunately, this method was also not delivering ideal hair removal; using a kernel size of 5 removed most of the hairs, and using a kernel size of 7 removed the hairs more effectively but blurs the lesion area more, which is not desired.

Figure 23: Lesion image(1) from HAM10000 and morphological closing operation outputs with kernel sizes 5 and 7 (images 2 and 3 respectively) (Tschandl et al 2018, licensed under CC BY-NC 4.0)

Maglogiannis et al. propose five hair removal algorithms in their 2015 paper 'Hair Removal on Dermoscopy Images', and include detailed pictorial descriptions of the algorithms at work, as well as results for ground truth images. This helped to glean some insight into how the hairs in lesion images could be better identified and segmented. As such, the first full solution to the hair removal problem was based on their LI's based algorithm.

The proposed system's implementation of Maglogiannis et al. LI's based algorithm involved deepening the contrast of the hairs in a grayscale image using a Laplacian filter with kernel size 3. The resulting image, with responses correlating to hairs, is subsequently subtracted from the greyscale source image to strengthen the responses to hairs, before applying Sobel detector with kernel size 3, and LoG edge detection with kernel size 3 to define the hairs again. The two resulting images are added together to amplify the responses given by the edge detectors. The resulting image is thresholded with an adaptive implementation of Mean-C using a block size of 3 to achieve a hair mask, which is subsequently refined using morphological dilation & erosion with a circular filter with kernel sizes 9 and 11 respectively.

4.1.1 Hair Removal *V1*

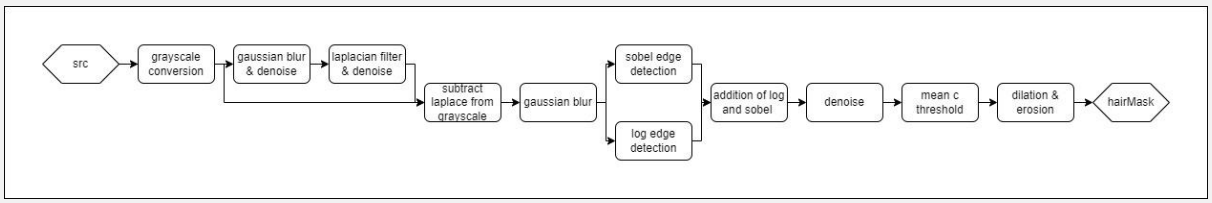


Figure 24: Hair Removal V1 flowchart

This technique was a good start, and allowed for the segmentation stage to commence. However, it was clear that erroneous features were being detected that would sully the attempts at segmentation in the next stage. Additionally, the refinement morphology was merging areas of the hair mask incorrectly, leading to a mask area that far exceeded the area of the image actually covered by hair, causing errors when inpainting (see Figure 25, image 12). Many tuneable variables were changed in attempts to reduce the false detection rate, including differing kernel shapes and sizes for gaussian blur, denoising, and morphological transformations.

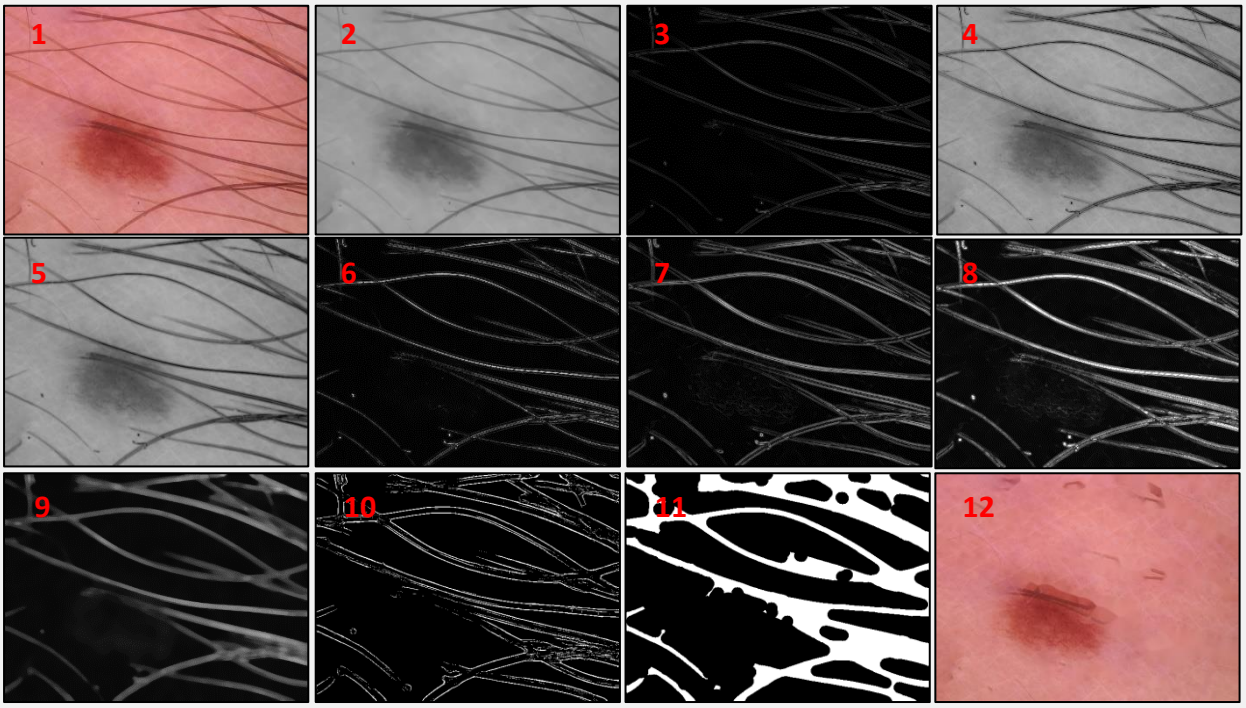


Figure 25: Hair Removal V1 per-operation output

source image(1), greyscale conversion & gaussian blur (2), Laplacian filter(3), Laplacian subtracted from greyscale(4), gaussian blur(5), LoG edge detection(6), Sobel edge detection(7), addition of LoG and Sobel output(8), blur & denoise(9), mean C threshold(10), morphology(11), fast marching inpainted(12) (Tschandl et al. 2018, licensed under CC BY-NC 4.0)

As suggested by the project's supervisor Rosin, P. the technique used to detect hairs in version 2 of the algorithm was Hough lines detection. Hough lines detection maps edge points from the input image onto cosine curves in the Hough space. Intersections of these curves in the Hough space are counted, and if the count exceeds the given threshold, a line is detected in the image space. Because hairs in lesion images generally appear as lines that are mostly straight, this technique can be used to detect them, with the right parameters. Canny edge detection with an aperture size of 3 was used to detect the hair edges, and a morphological closing operation with circular kernel of size 5 was performed before the resulting image was fed into the Hough lines detector with a vote threshold of 20, minimum line length of 25, and maximum line gap of 10. Another morphological close was applied to the resulting detected line image using kernel size 2 for refinement, resulting in the binary hair mask segmentation as shown in Figure 27. After experimenting with threshold parameters, the solution was demonstrating better sensitivity and specificity, compared to V1. In other words, hairs were being detected accurately with fewer erroneous features detected.

*4.1.2 Hair Removal **V2***

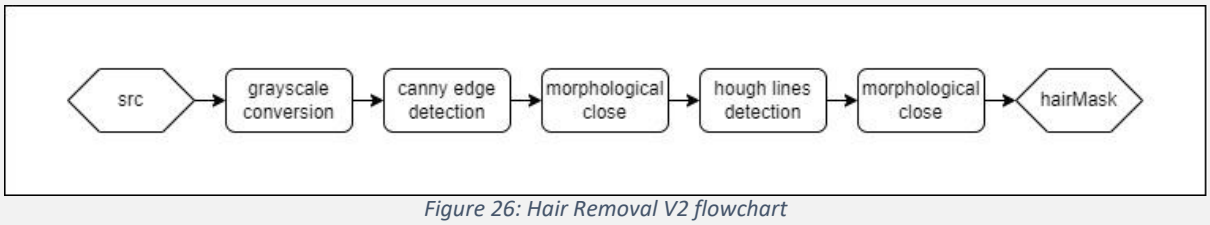


Figure 26: Hair Removal V2 flowchart

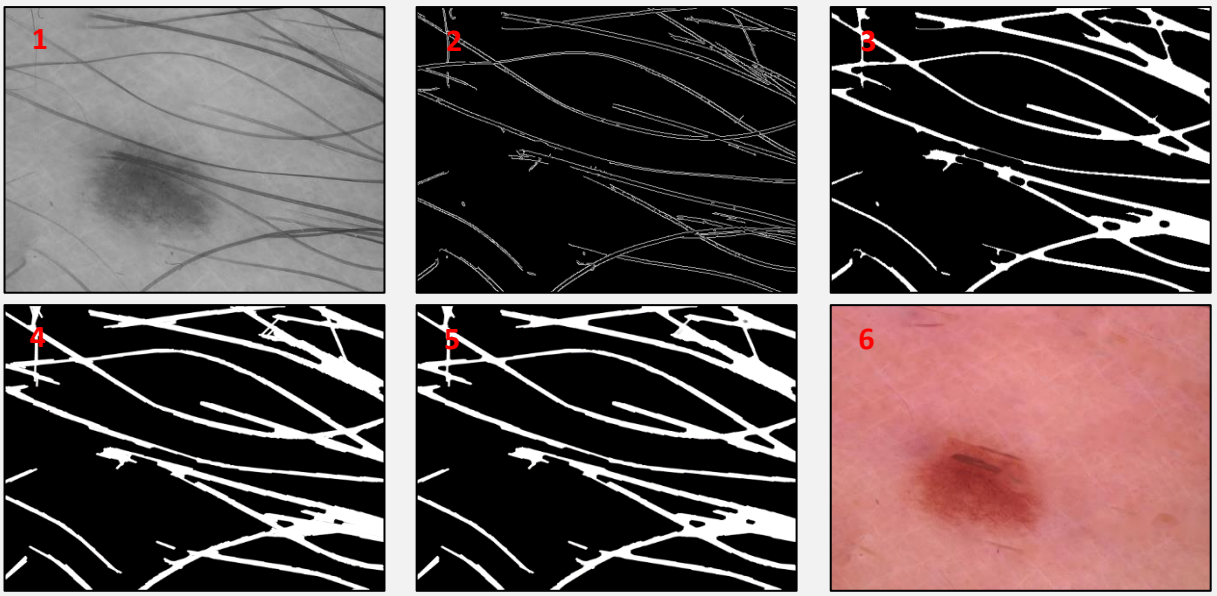


Figure 27: Hair Removal V2 per-operation output

greyscale conversion(1), Canny edge detection output(2), morphological closing(3), Hough lines detection output(4) additional morphology(5), fast marching inpainted(6) (Tschandl et al. 2018, licensed under CC BY-NC 4.0)

To further improve the performance of the hair detection, the V2 Hough lines based algorithm was extended to incorporate the improved edge detection from the V1 Ll's based algorithm. The red channel from the RGB image was extracted, as it shows the hairs best. A gaussian blur kernel size 3 is used as a precursor to a Laplacian filter with kernel size 3 to deepen the contrast of the hairs. The resulting image is subtracted from the original grayscale conversion to produce an image with strong responses for the hairs, which is then fed into the Canny edge detector with aperture size 3, as before. This new algorithm again resulted in an improved algorithm which could more accurately detect hairs, while detecting fewer erroneous features.

4.1.3 Hair Removal V3

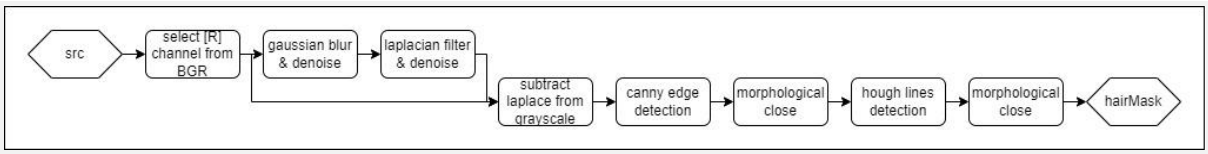


Figure 28: Hair Removal V3 flowchart

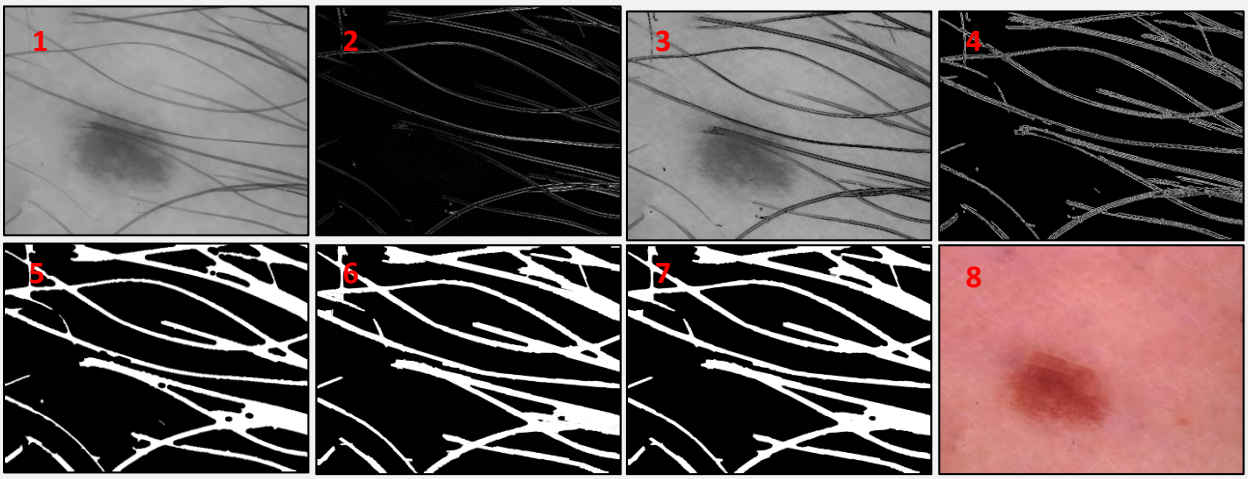


Figure 29: Hair Removal V3 per-operation output

greyscale conversion with gaussian blur(1), Laplacian filter output(2), subtraction of Laplacian from greyscale(3), Canny edge detection(4) morphological closing(5), Hough lines detection(6), additional morphology(7), fast marching inpainted(8) (Tschandl et al. 2018, licensed under CC BY-NC 4.0)

In order to evaluate the hair removal algorithms before proceeding, it would have been preferential to have ground truth hair masks that had been manually marked for comparison. However, the dataset used in the project did not include any ground truth for hair masks, and creating them manually for many images would have become very time consuming to do. Instead, attempts were made at simulating random hairs on hairless images, effectively creating ground truth data to compare to. This was done by making a large image 4 times the size of the original image, and drawing many thin curvilinear spline structures in different shades of dark brown. From this image, regions were selected at random, and further random transformations were performed to the selected region before it was imposed on top of a hairless image as shown in Figure 30. The tests that commenced showed good results for all 3 algorithms, however this did not provide satisfactory basis for a meaningful comparison and evaluation, as it was established that the simulated hairs did not well-represent real hairs found in lesion images. Thus, evaluation of the hair removal algorithms was done by visual examination over multiple images, and comparisons against Lee et al.'s DullRazor software were made to bolster the visual evaluations, as shown in Figure 31.

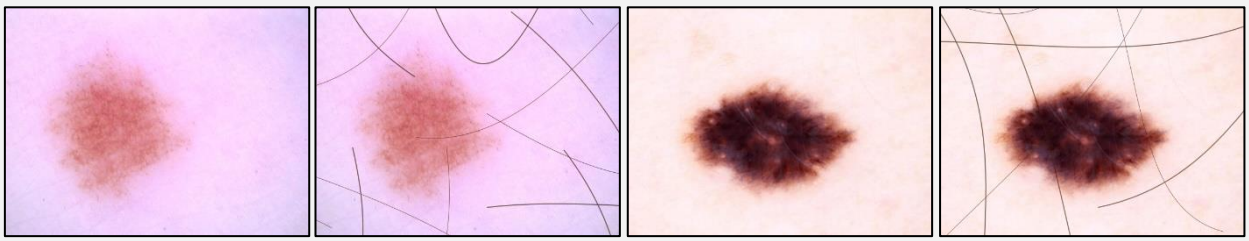


Figure 30: Original lesion images without hairs, alongside their corresponding simulated hair image (Tschandl et al. 2018, licensed under CC BY-NC 4.0)

To repair the original images based on the calculated hair masks, the original images are inpainted using OpenCV's fast-marching method based on Alexandra Telea's method (Telea 2004). The produced hair mask denotes the region from which the inpainting method repairs the source image as shown in Figure 31. The method starts at the region boundary, and for each pixel along the boundary, it replaces the pixel with a normalised weighted sum of its neighbouring pixels. After all the boundary pixels have been replaced, the method repeats, replacing the pixels that are near known pixels first, until all pixels have been replaced.

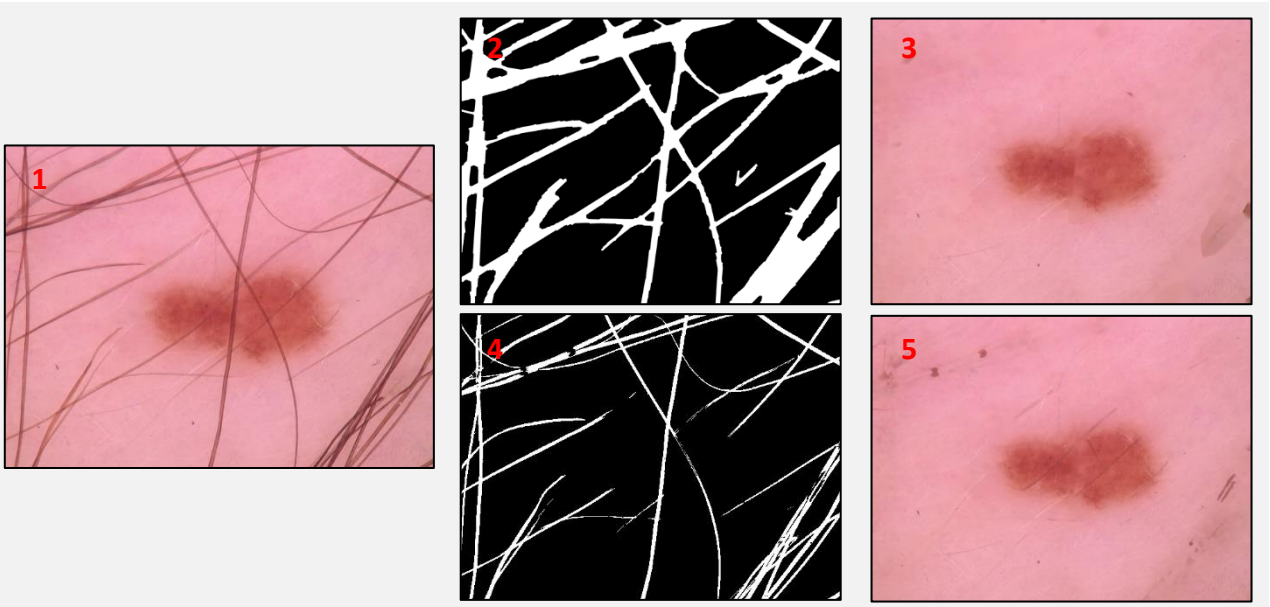


Figure 31: Original lesion image(1), Hair Removal V3 hair mask(2), V3 inpainted(3), DullRazor hair mask(4), DullRazor inpainted(5) (Tschandl et al. 2018, licensed under CC BY-NC 4.0) (Lee et al. 1997)

4.2 Segmentation

Once a satisfactory hair removal algorithm was in place, and the images were free from artefacts that prevent a good segmentation, development on the lesion segmentation stage was initiated. Much like with the hair removal algorithms, the segmentation algorithms underwent multiple revisions before they were satisfactory. This is because in order to calculate the various features in the next step, an accurate representation of the lesion area as a binary mask is necessary. This task proved difficult due to uneven lighting across the skin and black borders at the edges of some images.

To evaluate the segmentation algorithms, the Jaccard index (Intersection over Union) is calculated for the proposed system's segmentation and the ground truth segmentation, supplied with the dataset. This ground truth was manually marked by an expert dermatologist (Tschandl 2018) and should be a relatively good target for the system to try to obtain. To calculate the Jaccard index, the area of intersection is divided by the area of union. Hence, the ground truth segmentation and the recently created segmentation are

placed on top of each other, and the number of pixels in the area of overlap are counted as the area of overlap, and the number of pixels in either of the segmentation areas is counted as the area of union as shown in Figure 32.

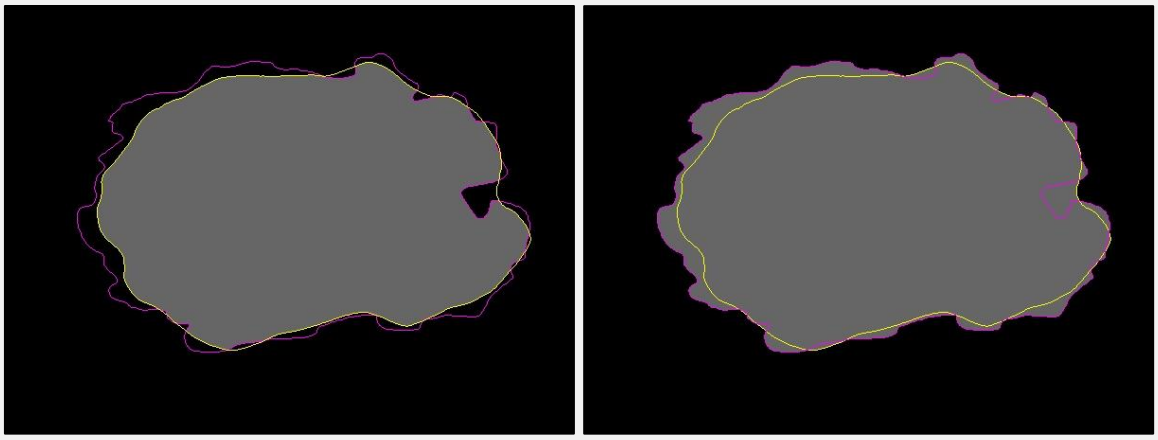


Figure 32: Example intersection(left), and union(right) of the proposed and curated segmentations (Tschandl 2018, licensed under CC BY-NC 4.0)

The obvious first choice for a segmentation method was Otsu's method. This method has seen success in many previous works in the problem field and was simple to implement. The first solution involved converting the pre-processed image to greyscale and applying a median blur of kernel size 9. Otsu's adaptive thresholding was used to segment the lesion, and a morphological closing operation of 7x7 ellipse kernel was used to refine the binary mask. Lastly, the largest contour found in the image by area was picked as the final segmentation. This is because the largest contour in the binary image typically surrounded the lesion area.

4.2.1 Segmentation *V1*

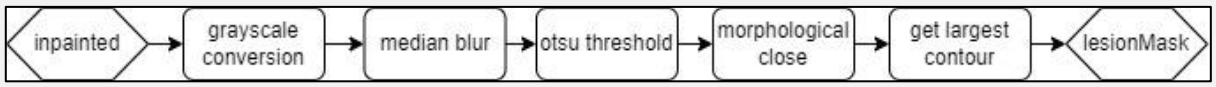


Figure 33: Segmentation V1 flowchart

This technique was a good start to segmenting the lesions properly, as the blur worked to remove any remaining noise, and the adaptive thresholding worked to counter some of the uneven lighting in the images. However, the algorithm was under-segmenting, causing resulting segmentation boundaries that did not well-represent the actual lesion area. Furthermore, the thresholding did not have enough generality to give correct segmentations when the background colour and lesion colour had little contrast between one another. To quickly and quantitively evaluate the effectiveness of the first version of the segmentation algorithm, 100 random images were selected from the dataset, and compared against the ground truth. A mean Jaccard index of 0.56 was calculated, indicating that only about half of the lesion area is being correctly identified, according to the ground truth. This consolidates the need for an improved algorithm before moving onto feature extraction.

The next iteration of the algorithm involved converting the image to 2 different colour spaces and thresholding separately and choosing the best one. The blue channel of the RGB colour space was chosen as the first channel to threshold because it appeared to provide more information for discriminating the lesion from the skin, compared to the green and red channels, as it segmented more consistently in experiments. The b channel of the CIEL*a*b* colour space was chosen as it also tended to reveal more information than the other two channels. Otsu thresholding was performed on both of the two selected channels and in each case the largest contour was selected as a segmentation. In some cases, the lesion was segmented well in one colour channel but not in the other, and the opposite occurred in other cases. Therefore, a check is performed to see which of the two segmentations were better. In the cases that the lesion was under-segmented, the detected boundaries tended to appear at the outer regions of the image, and in cases touching the image edges. The check that was performed in this version iterates over the image edges to see whether either segmentation is at the boundaries of the image. If the segmentation did not pass this check, it was discarded. If both segmentations pass the check, they are added together by method of OpenCV's 'bitwise_or' function. In other words, the resulting segmentation would be the area of both segmentations combined.

4.2.2 Segmentation V2

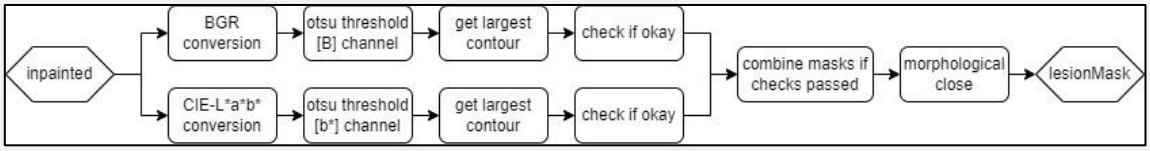


Figure 34: Segmentation V2 flowchart

This combination technique improved the accuracy of lesion segmentation significantly, as the added safety net by using two segmentations from two different channels allowed mistakes to be made in thresholding in one channel, as thresholding the other channel could still deliver a good segmentation. It can be noted that the selected CIEL*a*b* channel was delivering good segmentations more frequently than the selected RGB channel.

However, this method was still delivering some unsatisfactory segmentations, particularly in outlier cases where the colour of the lesion was close in colour to that of the surrounding skin, or when the structure of the lesion was non-uniform. In order to refine the algorithm further, experiments were conducted to see where the algorithm still fails, and in which of these cases further checks could be added to refine the choices of segmentations. The typical cases in which V2 would fail is when there were black borders around the image, indicating that a black border removal tool was necessary to implement in the system. However, solving this problem elegantly became time-consuming, so it was left aside until later.

To quantitatively compare the effectiveness of each version of the segmentation algorithm, the Jaccard index was calculated over all 1000 lesion images, comparing both versions of the system's segmentations to the ground truth. The mean Jaccard index of V1 segmentation algorithm was 0.584, similar to before, and the mean of V2 was 0.707, indicating that the

changes made in V2 did indeed improve performance, leading to more correct identifications of the region of interest. The standard deviation between version also decreased from 0.269 to 0.249, indicating that the new V2 algorithm was slightly more reliable.

	V1	V2
Mean	0.584	0.707
Min	0	0
Max	0.973	0.986
Median	0.688	0.801
Standard Deviation	0.269	0.249

Table 4: Segmentation IoU results against curated segmentations (Tschandl 2018, licensed under CC BY-NC 4.0)

4.3 Feature Extraction

Given a reasonable segmentation of the lesion, the next stage was to use the segmentation to extract particular information that characterise the particular lesion – the features. The information contained in the segmentation itself, given that the segmentation is an accurate representation of the lesion area, can be used to calculate a variety of measures that describe the shape of the lesion. The uniformity in the shape of a lesion is typically indicative of malignancy, and therefore this data is pertinent to a diagnosis. The texture and colour information in the source image itself is also pertinent to a diagnosis, as malignant lesions often have a more irregular distribution of colours and subsurface structures. This information is extracted by masking the source image with the segmentation, so that either just the lesion or skin is revealed as shown in Figure 35. From this masked image, measures can be made in terms of colour and texture that could indicate malignancy.

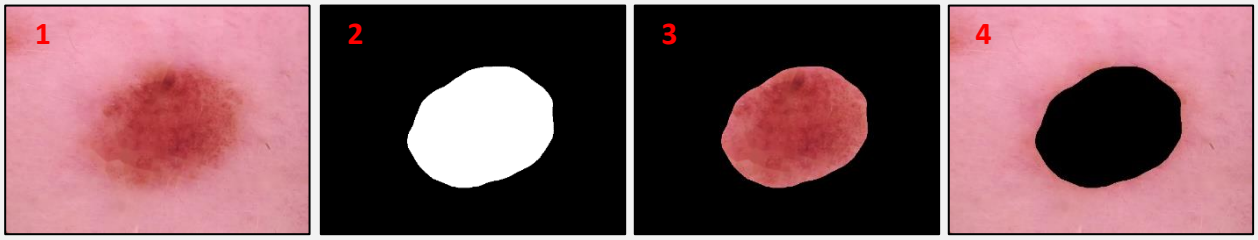


Figure 35: Inpainted image(1), its segmentation(2), lesion masked(3), skin masked(4) (Tschandl 2018, licensed under CC BY-NC 4.0)

Although there is plenty of available information to analyse in the images produced by the system so far, much of this data can be considered irrelevant in terms of obtaining a diagnosis. As such, the features extracted should be carefully selected according to an established dermoscopic algorithm, in order to avoid calculating redundant features, and to maintain a fast computation time, as many thousands of images are to be processed. The ABCD dermoscopy algorithm was chosen as the basis for extracting the features, as it has seen success in many previous works and is largely easy to implement. Hence, the asymmetry of the lesion's region, the region's border structure, and the lesion's colour and

texture are analysed. As opposed to calculating a total dermoscopy score (TDS) based on scores from each of the ABCD features, a feature vector will be created for each image analysed from the calculated features, which can be fed into the feature space of a machine learning classifier, which is then used for discrimination and classification of lesion images. For machine learning classifiers, it is preferential to use features that are normalised between 0.0-1.0, so care was taken to compute measurements this way. Otherwise, results that were not normalised *could* be normalised afterwards.

4.3.1 Asymmetry

Melanoma is often non-uniform in shape compared to other non-cancerous lesions, making symmetry a rather important analysis to make. Instead of bisecting the lesion into two perpendicular planes like the ABCD algorithm describes, an estimation of its asymmetry can be made in a number of different ways. In the dermoscopic ABCD algorithm, asymmetry is also measured in terms of the distribution of colour and texture, however these were not calculated in this section of the feature extraction, as these characteristics were hopefully captured in subsequent colour and texture analysis using histograms and GLCMs.

In the proposed system, the asymmetry index is calculated by flipping the axis aligned segmentation with area L around its major axis to obtain its symmetry $\mathrm{S_L}$. The non-overlapping areas of the L and $\mathrm{S_L}$ are denoted FS – false symmetry, and A is the area that includes lesion area L and area FS. This single feature is the first feature in the feature vector calculated for each image.

$$A = L \cup S_L \qquad FS = L \,\Delta\, S_L \qquad Asymmetry = 1 - \frac{FS}{A}$$

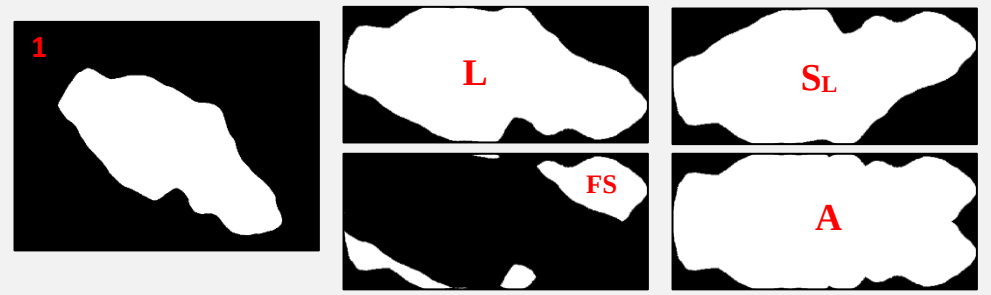


Figure 36: Segmentation(1), major axis aligned(L), major axis flipped(SL), false symmetry(FS), segmentation + false symmetry(A)

4.3.2 Border

The segmentation provides key diagnostic information in the form of shape measures, and by measuring the irregularity in the border of the shape itself. As opposed to splitting the lesion into eight sections and analysing the abruptness of the pigment's cut-off in each section, as described in the ABCD dermoscopy algorithm, the shape and structure of the segmentation are more simply analysed in the form of standard shape and contour measures. From the segmentation, a set of 8 features in the range 0.0-1.0 are extracted that hope to capture diagnostic information that could discriminate the different skin conditions. As described in section 2.5, the following features are also invariant with respects to translation, rotation, and scaling. This aspect is vital in assuring lesion images are assessed objectively, so meaningful comparisons can be made for discriminating between each

lesion. The perimeter and area of the produced segmentation's shape are calculated simply by counting the pixels that made up the border, and the pixels that are inside the shape, respectively.

The first of the border features calculated was circularity. Benign lesions typically present as a rounded and uniform shape, and malignancy tends to disfigure the shape and introduce extruding and intruding irregularities. Žunić et al. propose a Hu invariant circularity measure using 0^{th} and 2^{nd} order moments (Žunić et al. 2010). Because this measure is area-based, it is less prone to local noise and sharp intrusions in the shape than the standard circularity measure based on area and perimeter: $\frac{4*\pi*Area}{Perimeter^2}$. This proves to be a more robust method of measuring the lesion's true circularity, particularly when the segmentation it is calculated from has some incorrect irregularities. The shape irregularities not captured by this measure are to be captured in the solidity and fractal dimension measures later on in this section.

$$Circularity\ (0.0\text{-}1.0) = \frac{m00^2}{2*\pi*(m20+m02)}$$

To gain access to more shape measures, the minimum bounding rectangle of the shape was computed, as shown in Figure 9; from this the shape's rectangularity can be computed. Although rectangularity is not intuitively indicative of any particular skin condition, a lesion shape that is more rectangular also means there is a more distinct deviation from the round and uniform shape typically found in benign lesions.

$$Rectangularity\ (0.0\text{-}1.0) = \frac{Lesion\ Area}{Bounding\ Rectangle\ Area}$$

A rough elongation measure can also be computed using the minimum bounding rectangle. Again, this measure can help to differentiate between lesion conditions, as more elongated lesions are indicative of malignancy; malignant cancer can quickly grow outwards from the lesion's origin in any particular direction. This measure does not perform well in curved regions, however, a typical lesion area appears as one homogenous area, not in a curved fashion, so this measure can be used in earnest.

$$Elongation\ (0.0\text{-}1.0) = \frac{Bounding\ Rectangle\ Width}{Bounding\ Rectangle\ Height}$$

Computing the convex hull of the shape, like shown in Figure 8, allows for further shape measures to be calculated, one of which is solidity, an estimation of the shape's 'density'. The solidity measure can be helpful in identifying lesions with small or large intrusions, again indicating possible malignancy due to the non-uniformity.

$$Solidity\ (0.0\text{-}1.0) = \frac{Lesion\ Area}{Convex\ Hull\ Area}$$

The convex hull information also allows for the estimation of the convexity of the shape, which is equivalent to the ratio of the skin lesion perimeter to the perimeter of the convex hull. Once again, this measure is useful in indicating malignancy, as the convexity of a convex object equates to 1.0, and any concave intrusions in the shape will decrease this score towards 0.0.

$$Convexity\ (0.0\text{-}1.0) = \frac{Convex\ Hull\ Perimeter}{Lesion\ Perimeter}$$

The Polsby-popper test is a well-known shape measure for shapes. In this case, as circularity is already calculated, and it is equivalent to compactness, the convex hull perimeter is used, which excludes local irregularities of the shape while accounting for irregular (non-circular) shapes in general. The result for this measure is a maximum of 1 for a fully compact object, which decreases to 0.0 for any lack of compactness.

$$Compactness\ (0.0\text{-}1.0) = \frac{4 * \pi * Area}{Convex\ Hull\ Perimeter^2}$$

To measure the structure of the border, an estimation of the border's fractality is computed. In the proposed system, the Minkowski-Bouligand dimension is the method used for estimating this characteristic, and involves covering the border of the shape with square boxes of increasing size r on a d-dimensional fixed grid, and counting the number of boxes $N(r)$ it takes to cover the boundary completely. In the proposed method, the initial box size ε is 1 pixel, which doubles each iteration to 2x2 pixels, then 4x4 and so on for 8 iterations. A linear regression formula over the results gives an estimation of the fractal dimension between 0.0-1.0. A higher value for the fractal dimension indicates a more complex border pattern. This technique is useful as it does not require smoothing operations on the border like in other techniques.

$$Fractal\ dimension\ (0.0\text{-}1.0) = \lim_{r \to 0} \frac{logN(r)}{log(1/r)}$$

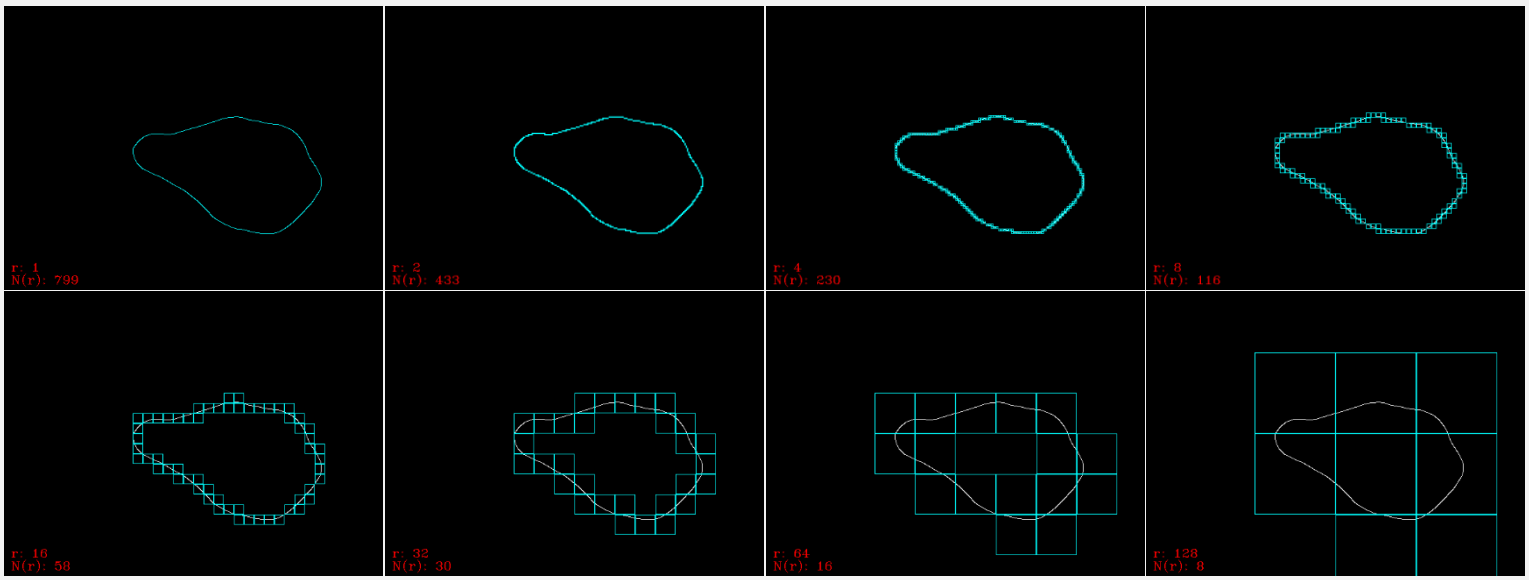


Figure 37: Intermediary stages of the box counting method

A lesion with irregular borders tends to have a large variance in the radial distance, to compute this measure, the variance of the radial distance distribution is computed. d is the distance between the centroid and boundary point C, and m is the average distance from the centroid and each point along the boundary. This average distance is used to constuct a circle of the same radius, from which a ratio of the circle's area against the surface of the lesion is made.

$$Radial\ variance\ (0.0\text{-}1.0) = \frac{\frac{1}{P}\Sigma\, P \in C\ (d(p,G)-m)^2}{m^2}$$

4.3.3 Colour

Malignant melanoma is a skin cancer that develops in the pigment-producing melanocyte cells. Hence, a change in pigment, and therefore colour of the skin, is commonly associated with malignancy. The traditional ABCD dermoscopy algorithm counts the number of different colours present in a lesion. This method is ambiguous for computer vision, as digital colours are part of a spectrum that would need grouping in order to 'count' colours.

The proposed system's method extracts colour features from two areas in the lesion image: the lesion area itself, and the surrounding skin area. By masking either the lesion itself or the surrounding skin using the segmentation, the colour features are extracted using RGB histogram cubic bins of size 256. Instead of extracting relative colour based on surrounding skin, the mean, median, and standard deviations of the histogram data for both the lesion and the skin are extracted for each of the 3 channels of the RGB colour space, resulting in 18 new features to add to the feature vector for each image.

4.3.4 Differential

To estimate the variability of the differential structures present in a lesion, a normalized grey level co-occurrence matrix (GLCM) was constructed to extract descriptors that could differentiate lesions in terms of their texture. A GLCM was computed for the segmented lesion area for orientations 0°, 45°, 90°, and 135°. From these matrices, a set of averaged statistical measures are extracted, resulting in an additional 8 new features added to the feature vector for an image. Figures 38 part 1 and 2 show how the GLCM is constructed to store the co-occurrences of pixel intensity at each angle.

```
// changing offset for each angle
switch (angle) {
	case 0:
		endX = endX - 1;
		jChangeX = jChangeX + 1;
		break;
	case 45:
		startY = startY + 1;
		endX = endX - 1;
		jChangeY = jChangeY - 1;
		jChangeX = jChangeX + 1;
		break;
	case 90:
		startY = startY + 1;
		jChangeY = jChangeY - 1;
		break;
	case 135:
		startY = startY + 1;
		startX = startX + 1;
		jChangeY = jChangeY - 1;
		jChangeX = jChangeX - 1;
		break;
}
```

Figure 38: GLCM implementation part 1

```
// loop through pixels
for (int y = startY; y < endY; y++) {
	for (int x = startX; x < endX; x++) {
		// get origin pixel
		int i = (int) img.get(y, x)[0];
		// ignore black (empty) pixels
		if (i != 0) {
			// get offset pixel
			int j = (int) img.get(y + jChangeY, x + jChangeX)[0];
			// ignore black (empty) pixels
			if (j != 0) {
				// store the co-occurrences in the matrix gl
				double[] count = gl.get(i, j);
				count[0]++;
				gl.put(i, j, count);
			}
		}
	}
}
```

Figure 38: GLCM implementation part 2

The following features were calculated from the normalised version of the newly created GLCM, and their equations are shown below. Figure 39 shows how the features listed are calculated in the code.

P_{ij} = element i,j of the normalised GLCM $\quad N = 256$ (number of grey levels)

$\mu = \sum_{i,j=0}^{N-1} iP_{ij}$ = GLCM intensity mean $\quad \sigma^2 = \sum_{i,j=0}^{N-1} P_{ij}(i-\mu)^2$ = GLCM intensity variance

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{P_{ij}}{1+(i-j)^2}$$

$$Energy = \sum_{i,j=0}^{N-1} (P_{ij})^2$$

$$Entropy = \sum_{i,j=0}^{N-1} -ln(P_{ij})P_{ij}$$

$$Contrast = \sum_{i,j=0}^{N-1} P_{ij}(i-j)^2$$

$$Correlation = \sum_{i,j=0}^{N-1} P_{ij} \frac{(i-\mu)(j-\mu)}{\sigma^2}$$

```
// loop through GLCM for asm, contrast, homogeneity, entropy sums and means of i,j
for(int i=0;i<256;i++) {
	for(int j=0;j<256;j++) {
		asm = asm + glcm.get(i,j)[0] * glcm.get(i,j)[0];
		contrast = contrast + (i-j) * (i-j) * glcm.get(i,j)[0];
		homogeneity = homogeneity + glcm.get(i,j)[0] / (1 + ((i-j)*(i-j)));
		if(glcm.get(i,j)[0] != 0) {
			entropy = entropy - glcm.get(i,j)[0] * Math.log10(glcm.get(i,j)[0]);
			if(glcm.get(i,j)[0] > maxProb) {
				maxProb = glcm.get(i,j)[0];
			}
		}
		iMean = iMean + (i*glcm.get(i,j)[0]);
		jMean = jMean + (j*glcm.get(i,j)[0]);
	}
}
// loop through GLCM again for variance of i,j
for(int i=0;i<256;i++) {
	for(int j=0;j<256;j++) {
		iVariance = iVariance + (i-iMean) * (i-iMean) * glcm.get(i,j)[0];
		jVariance = jVariance + (j-jMean) * (j-jMean) * glcm.get(i,j)[0];
	}
}
// loop through GLCM again for correlation sum
for(int i=0;i<256;i++) {
	for(int j=0;j<256;j++) {
		correlation = correlation + (((i-iMean) * (j-jMean))/Math.sqrt(iVariance*jVariance)) * glcm.get(i,j)[0];
	}
}
energy = Math.sqrt(asm);
mean = (iMean+jMean)/2;
variance = (iVariance+jVariance)/2;
```

Figure 39: GLCM feature calculations

After calculating the 35 features based on the ABCD algorithm, the resulting feature vector becomes a characterisation of the lesion image to the best of the system's ability. Weka provides an ARFF (Attribute-Relation File Format) text file that describes a list of instances sharing a set of attributes. There are two sections to an ARFF file, the first being a header, containing a list of attribute names and types, including the class attribute. The second section contains the data for the attributes in comma separated lists for every instance, containing entries for values of each attribute, as well as an entry for class values. Pertinent to the proposed system, after processing all of the images in the dataset and inputting their feature vectors into the ARFF file, the data section of the ARFF file contains 10015 rows, one for each image, and each row contains a comma separated list of results for the 35 features, as well as the known class found in the dataset's metadata, which is either 'MALIGNANT' or 'BENIGN'. An example of which is shown in Figure 40 for clarity.

```
@relation 'example-ham-lesions'

@attribute symmetry numeric
@attribute circularity numeric
@attribute fractalDimension numeric
@attribute radialVariance numeric
@attribute contrast numeric
@attribute correlation numeric
@attribute energy numeric
@attribute entropy numeric
@attribute homogeneity numeric
@attribute lesionBMean numeric
@attribute lesionGMean numeric
@attribute lesionRMean numeric
@attribute lesionBMedian numeric
@attribute lesionGMedian numeric
@attribute lesionRMedian numeric
@attribute lesionBStd numeric
@attribute lesionGStd numeric
@attribute lesionRStd numeric
@attribute class {MALIGNANT,BENIGN}

@data
0.883,0.957,0.982,1.186,15.863,0.989,0.033,3.103,0.342,85.657,103.319,72.277,0.178,0.214,69.657,131.993,141.776,103.14,BENIGN
0.829,0.893,0.971,1.359,15.055,0.971,0.043,2.91,0.329,70.613,57.116,37.781,14.727,4.017,0,122.482,103.424,89.046,BENIGN
0.816,0.678,0.986,1.781,62.349,0.949,0.024,3.405,0.206,65.345,74.686,122.108,0.111,1.772,247.57,116.635,114.037,141.679,BENIGN
0.848,0.883,0.979,1.34,8.576,0.985,0.043,2.848,0.361,67.068,69.606,56.354,0.143,0,32.073,130.164,124.255,105.164,BENIGN
0.777,0.786,1.042,1.509,80.453,0.947,0.022,3.486,0.186,65.115,105.555,80.981,29.874,147.799,73.101,103.074,135.397,102.545,MALIGNANT
0.918,0.96,1.004,1.201,76.851,0.904,0.029,3.294,0.203,70.951,54.115,68.847,23.889,2.543,136.389,105.882,94.646,109.984,MALIGNANT
0.951,0.962,0.974,1.198,15.41,0.968,0.04,2.962,0.306,55.839,61.86,42.065,2.136,28.592,0,104.855,108.631,98.305,BENIGN
0.832,0.933,0.974,1.261,29.231,0.946,0.036,3.076,0.269,46.923,59.119,80.378,0.162,0.306,280.913,99.246,112.05,128.532,BENIGN
0.777,0.86,0.959,1.375,7.141,0.956,0.069,2.523,0.379,30.864,29.846,22.304,0,0,0,83.976,82.271,71.42,BENIGN
0.906,0.854,0.98,1.452,18.401,0.959,0.038,2.994,0.267,66.953,57.256,42.868,45.968,11.495,0,116.209,102.633,98.797,BENIGN
0.786,0.816,0.985,1.527,40.203,0.94,0.034,3.121,0.257,72.004,69.323,53.814,3.984,35.985,117.601,119.302,119.001,110.46,MALIGNANT
0.873,0.921,1,1.282,21.063,0.972,0.035,3.072,0.304,78.991,93.894,66.741,4.703,11.181,0.111,127.873,142.301,123.212,BENIGN
0.962,0.976,0.999,1.151,18.811,0.971,0.041,2.968,0.313,39.295,37.1,76.841,0.304,0,299.11,87.25,80.039,108.806,BENIGN
0.893,0.909,0.979,1.29,32.227,0.961,0.031,3.19,0.25,89.767,74.798,69.411,153.369,91.58,384.372,127.23,120.47,122.771,MALIGNANT
0.751,0.532,1.018,2.15,3.635,0.919,0.101,2.146,0.497,18.742,19.801,16.437,0,0,0,61.766,70.793,65.913,BENIGN
```

Figure 40: Example Weka ARFF file

The ARFF file generated by the system after feature calculation is used hereafter in the system pipeline, as opposed to calculating the features every time they are needed. The ARFF file can be modified manually or programmatically, and pre-processing such as filtering, class balancing, clustering, visualisation, and classification can be performed on the file using Weka functions.

4.4 Classification

As stated in section 2.1.2, classifying lesions in the ABCD algorithm is done using the total dermoscopy score which lies between 1.0 and 8.9, and a TDS of 4.74-5.45 indicates a suspicious lesion and a TDS larger than 5.45 suggests a high likelihood of melanoma.

In the proposed system, the feature vector is used directly for machine learning classification. Once the ARFF file has been generated for the training set, Weka provides functions for using this to build classification models and obtain classification scores based on a supplied test set, cross-validation, or a train/test split of the dataset. To obtain a classification summary on the training data produced thus far by the system, the ARFF file is

loaded into Weka, from which classification can be performed directly. For the purposes of testing, the Weka GUI was utilised to perform pre-processing and obtain summaries of results quickly. 10-fold cross validation was chosen to evaluate the training data using the following classifiers, as it provides a way to obtain an estimate of the model's performance without providing a test set.

4.4.1 kNN

As stated in section 2.9, the k-Nearest Neighbours is an unsupervised machine learning algorithm frequently used for classification in many CAD systems, including those for skin cancer. The algorithm assumes that similar things exist in close proximity to one another in the feature space. Hence, for new data, the algorithm searches for nearby data in the feature space, by calculating the Euclidian distance between these points, and the most frequent class label assigned to the k-nearest neighbours is considered the predicted class for the new data.

Using a kNN rule of thumb, a value of k was selected as $k = \frac{\sqrt{n}}{2}$, where n is the number of samples in the dataset. Hence, $k = 45$. Using 10-fold cross validation, the classification summary for the training set is shown in Table 5.

Correctly Classified Instances:	85.6%
Incorrectly Classified Instances:	14.4%
MAE:	0.216
RMSE:	0.333

	TP rate	**FP rate**	**Precision**	**Recall**	**F1**	**MCC**	**ROC Area**
MALIGNANT	0.015	0.001	0.647	0.015	0.030	0.083	0.751
BENIGN	0.999	0.985	0.857	0.999	0.922	0.083	0.751
Weighted avg.	0.856	0.842	0.826	0.856	0.793	0.083	0.751

Table 5: kNN classification summary (HAM10000)

4.4.2 Bayes Net

A Bayesian network is a probabilistic graphical model often used for unsupervised machine learning, based on Bayes' theorem, which specifies joint conditional probability distributions, and uses Bayesian inference to evaluate the probability of a particular outcome. In the case of classification, the Bayesian network predicts the probability that a particular instance belongs to a particular class. Performing classification on the training set with a Bayesian network yields the results shown in Table 6.

Correctly Classified Instances:	74.9%
Incorrectly Classified Instances:	25.1%
MAE:	0.259
RMSE:	0.46

	TP rate	FP rate	Precision	Recall	F1	MCC	ROC Area
MALIGNANT	0.543	0.216	0.298	0.543	0.385	0.261	0.727
BENIGN	0.784	0.457	0.910	0.784	0.842	0.261	0.727
Weighted avg.	0.749	0.422	0.821	0.749	0.776	0.261	0.727

Table 6: Bayes Net classification summary (HAM10000)

4.4.3 Random Forest

Random forest is a machine learning algorithm commonly used for classification. The random forest model consists of multiple decision trees. Decision trees are non-parametric supervised learning models where the data is split into subsets using binary recursive partitioning, otherwise known as 'divide and conquer'. The result of a classification in a random forest classifier is decided as the class selected by the most trees. Performing classification on the training set with a random forest classifier yields the results shown in Table 7.

Correctly Classified Instances:	86.1%
Incorrectly Classified Instances:	13.9%
MAE:	0.215
RMSE:	0.322

	TP rate	FP rate	Precision	Recall	F1	MCC	ROC Area
MALIGNANT	0.090	0.008	0.649	0.090	0.158	0.205	0.801
BENIGN	0.992	0.910	0.865	0.992	0.924	0.205	0.801
Weighted avg.	0.861	0.779	0.834	0.861	0.813	0.205	0.801

Table 7: Random Forest classification summary (HAM10000)

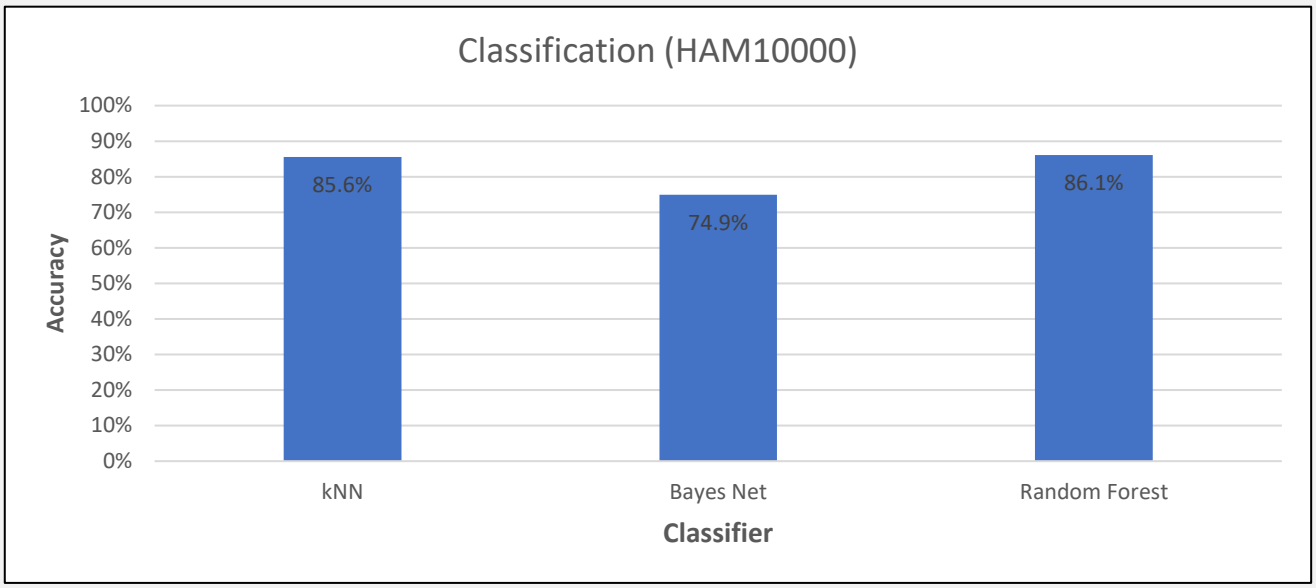


Figure 41: HAM10000 classification accuracy comparison

4.4.4 Class Balancing

It was noted that for each of the classifiers tested above, the F_1 score for the malignant class was much lower than that for the benign class. The F_1 score is calculated from the precision and recall measures, and is a measure of the test's accuracy itself. This suggests that the malignant class is not represented well in the dataset, compared to the benign class. Due to the fact that there is a significant imbalance in the classes present in the training set, it was necessary to perform class balancing, in hopes to better train the classification models and obtain improved results, as it is important to be able to classify either class with equal accuracy.

There are 8563 lesions in the HAM10000 dataset that are considered benign, and only 1452 that are malignant, which roughly equates to a ratio of 5:1. To remedy the issue, either the majority class could be undersampled, or the minority class could be oversampled, or a combination of the two. Weka provides a variety of functions for resampling the training data in the ARFF file. SMOTE is one technique included that is used to oversample the minority class. SMOTE stands for synthetic minority oversampling technique, and instead of simply duplicating data in the minority class it generates new instances that are plausible by interpolating between existing instances that lie near to each other in the feature space (Chawla et al. 2002).

After oversampling the minority class using SMOTE, there were now 8563 benign instances, as well as 8563 malignant instances, equalling 17126 total training instances. The classifiers were run again, with the results listed in Tables 8, 9, and 10.

kNN:

Accuracy:	74.8%
Inaccuracy:	25.2%
MAE:	0.35
RMSE:	0.41

	TP rate	FP rate	Precision	Recall	F1	MCC	ROC Area
MALIGNANT	0.899	0.403	0.691	0.899	0.781	0.520	0.844
BENIGN	0.597	0.101	0.855	0.597	0.703	0.520	0.844
Weighted avg.	0.748	0.252	0.773	0.748	0.742	0.520	0.844

Table 8: kNN classification summary after SMOTE class balancing (HAM10000)

Bayes Net:

Accuracy:	88.3%
Inaccuracy:	11.7%
MAE:	0.12
RMSE:	0.32

	TP rate	FP rate	Precision	Recall	F1	MCC	ROC Area
MALIGNANT	0.880	0.114	0.885	0.880	0.883	0.766	0.949
BENIGN	0.886	0.120	0.881	0.886	0.883	0.766	0.949
Weighted avg.	0.883	0.117	0.883	0.883	0.883	0.766	0.949

Table 9: Bayes Net classification summary after SMOTE class balancing (HAM10000)

Random Forest:

Accuracy:	87.1%
Inaccuracy:	12.9%
MAE:	0.26
RMSE:	0.32

	TP rate	FP rate	Precision	Recall	F1	MCC	ROC Area
MALIGNANT	0.906	0.163	0.847	0.906	0.876	0.744	0.951
BENIGN	0.837	0.094	0.899	0.837	0.867	0.744	0.951
Weighted avg.	0.871	0.129	0.873	0.871	0.871	0.744	0.951

Table 10: Random Forest classification summary after SMOTE class balancing (HAM10000)

The results obtained after class balancing are greatly improved from before; the F1 scores for both classes are much closer to one another, therefore both classes can be classified with roughly equal amounts of confidence, contrary to before. The MCC score is generally used to estimate the model's performance class-wise, and this measure also increased after class balancing, showing that the classifier is indeed more often correctly predicting *both* classes. Additionally, the ROC (Receiver Operator Characteristic) area score is increased, showing that the model is better at distinguishing between classes.

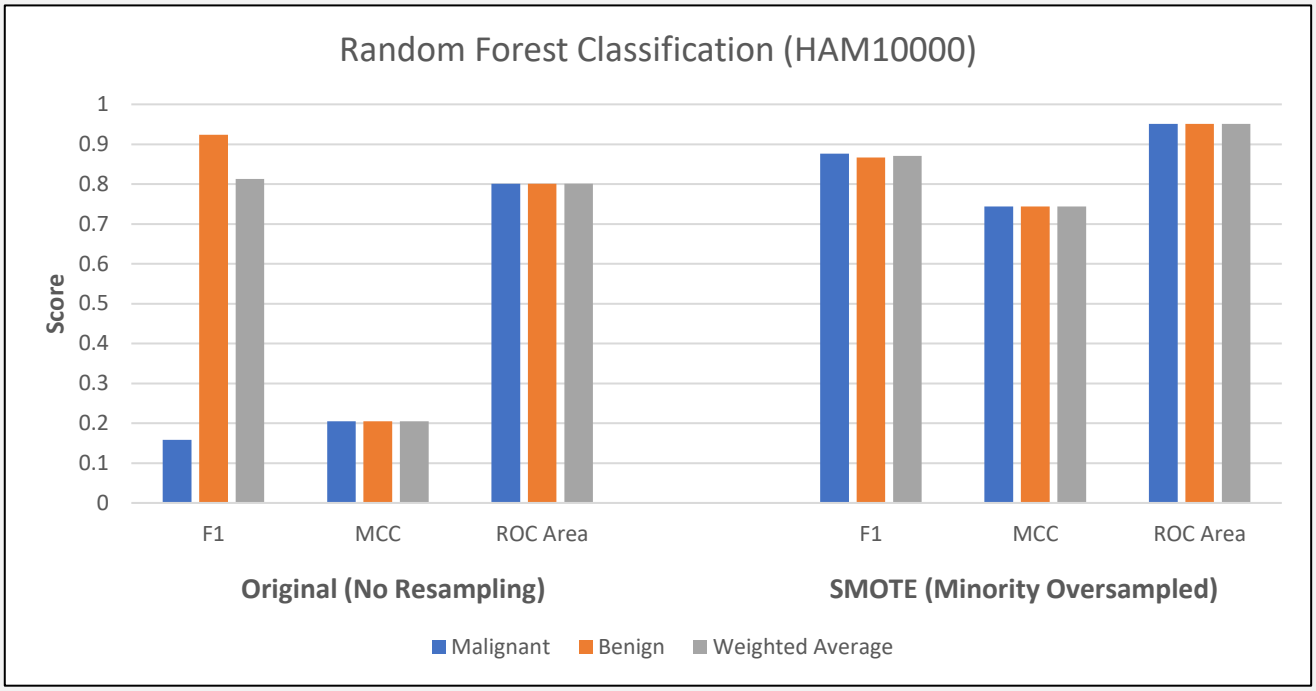


Figure 42: HAM10000 classification measures (F1, MCC, and ROC Area) comparison for non-resampled and SMOTE resampled data

4.4.5 Test / Validation Datasets

To evaluate the system more critically, and assess how well each part of the system works across a larger variety of images, two additional datasets were added into the project: BCN20000 (Combalia et al. 2019) and MSK (Codella et al. 2018).

	Benign	**Malignant**	**Total**
HAM10000	8563	1452	10015
BCN20000	9014	3399	12413
MSK	2170	536	2706
Total	**19747**	**5387**	**25134**

Table 11: Class (benign / malignant) counts for the datasets HAM10000, BCN20000, MSK

The ISIC 2019 challenge training dataset includes the HAM10000 dataset, as well as the BCN20000 and MSK datasets. Seeing as these individual datasets were combined for the ISIC challenge, it seemed sensible to use this additional data for further analysis of the system built with the HAM10000 dataset in mind. As mentioned in section 2.2.1, the dermoscopic images included in the HAM10000 dataset were sourced from hospitals in Austria and Australia over a period of 20 years. The images in the BCN20000 dataset were sourced from a hospital in Barcelona, Spain over a 6-year period, and the images in the MSK dataset were sourced from the Memorial Sloan-Kettering Cancer Center in New York, USA. It is well known that geographical bias in a dataset can skew classification and produce optimistic results (Bissoto et al. 2019), and that CAD systems trained on biased datasets are less able to generalise, and can perform significantly worse when using a different dataset (Wen et al. 2022). Given this, it would be interesting to see how classification accuracy is affected when training and testing on the different datasets. If it were found that classification accuracy

differs significantly between datasets, it could shed light on where the CAD system lacks ability to generalise, as the system will be confronted with data that is slightly different due to biases inherent to each individual dataset.

To verify that the SMOTE class balancing technique also provides better classification results for the newly added datasets, the same tests were run as with the HAM10000 datasets, with the results shown in Figures 43 and 44 – confirming the positive effect of the technique.

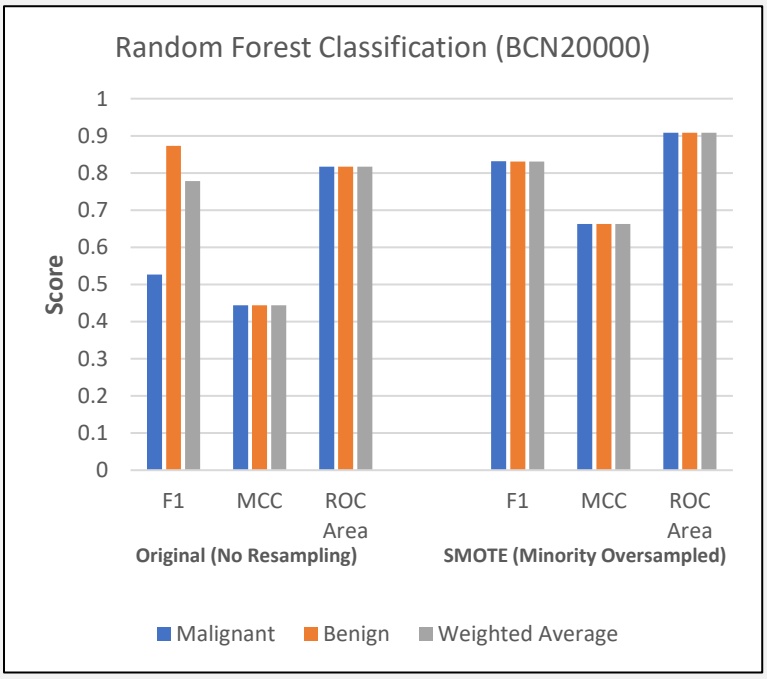


Figure 43: BCN20000 classification measures (F1, MCC, and ROC Area) comparison for non-resampled and SMOTE resampled data

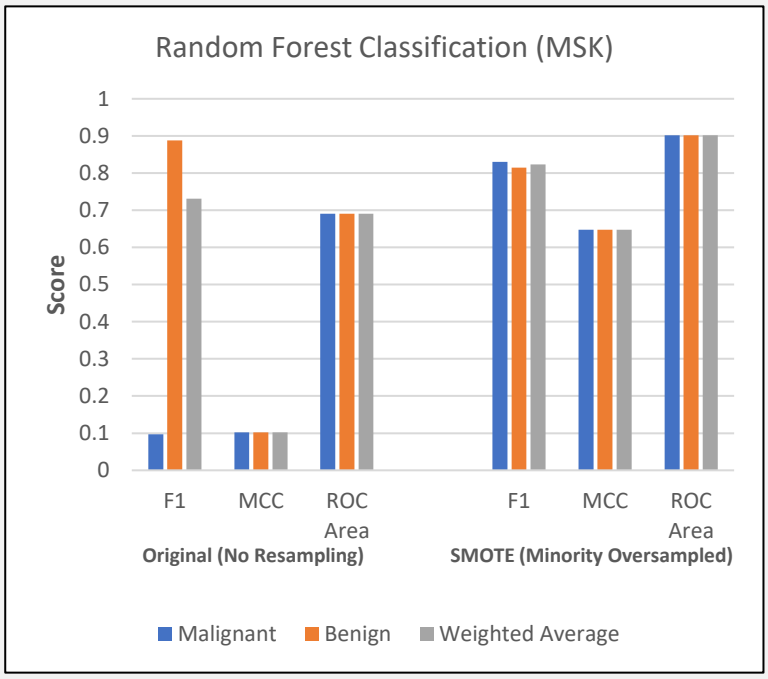


Figure 44: MSK classification measures (F1, MCC, and ROC Area) comparison for non-resampled and SMOTE resampled data

4.5 Additional Improvements

4.5.1 Pre-processing: Black Border Removal

Black borders appear at the corners of a number of lesion images, because of the nature of the dermatoscope lens, and they appear as a circular border with the lesion present inside as shown in Section 2.3, Figure 6. These black borders can be a barrier to achieving a good segmentation, and the V2 segmentation algorithm fails in most cases where black borders occur, because the thresholding detects the much darker region. The addition of the two extra datasets prompted the creation of this tool, as black borders appear much more frequently in BCN20000 and MSK, than in HAM10000. Because of the circular property of the black borders, a circle detecting algorithm would be appropriate for detecting them, as a pre-requisite to their removal.

Four regions of interest of size 25x25 are defined in the four corners of the image, as shown in Figure 45, from which the average pixel intensity is sampled. If the average pixel intensity is smaller than 15, it means that it is quite likely that a black border is present. A median blur is applied to the image with a kernel size of 75 to make the black border appear more

clearly against the rest of the image. Afterwards, a Hough circles detection is performed on the image to detect the circle made by the black border. Any circles detected of radius smaller than a half of the image's height were discarded, and as such the largest circle detected in the image denotes the edge of the borders, shown in Figure 46.

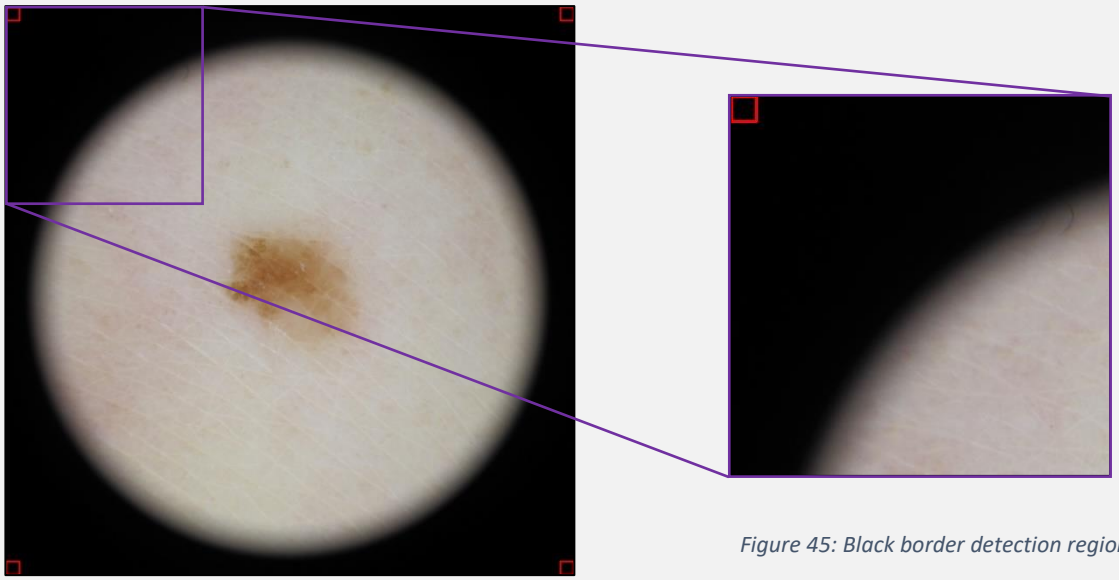


Figure 45: Black border detection region

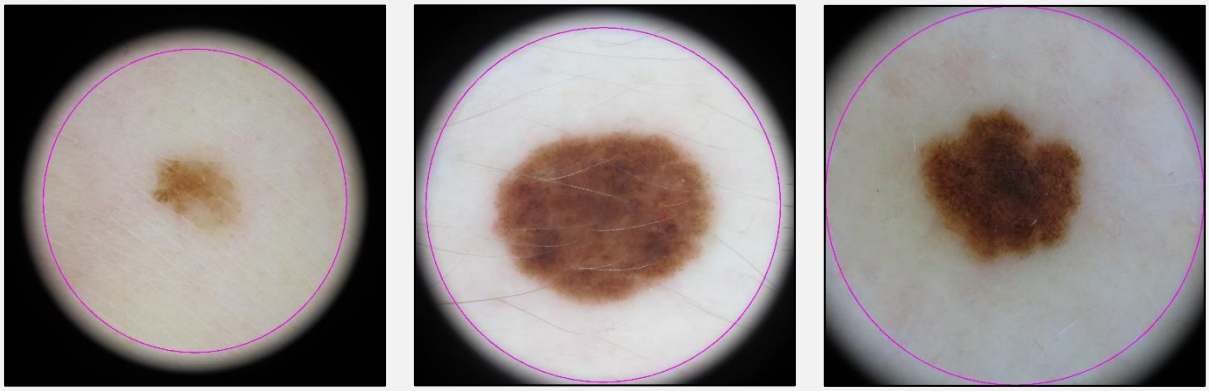
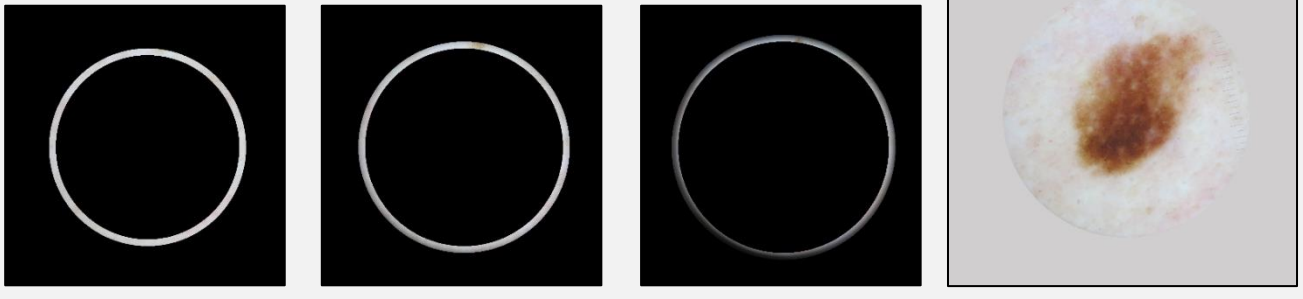


Figure 46: Largest detected circle using Hough circles

Concentric circles are created with the same centroid C of the circle denoting the detected black border, each with increasing radius. The region between each adjacent pair of circles is sampled for the mean pixel intensity, starting at the inner-most region, shown in Figure 47. For each region with a mean pixel intensity lower than that of its inner region, the pixel values are replaced with the greater mean pixel intensity. The process is repeated until the black borders have been removed.

Figure 47: Circular regions for incremental average pixel replacement, and resulting image



4.5.2 Segmentation: V3

The motive behind implementing V3 of the segmentation algorithm was that the average Jaccard index of the V2 algorithm was 0.71, according to the curated ground truth. While this result is firmly positive, it means there is ~30% error on average, meaning there was clearly room for improvement. An accurate segmentation is a critical requisite to obtaining accurate feature results, particularly for the asymmetry and border features, which makes up half of the ABCD algorithm, and leaves the 30% segmentation error as a prime target to reduce in hopes of improving the overall effectiveness of the CAD system. There were also numerous visible outliers in the feature vectors as a result, where the calculated feature would equal 0.0 because of an incorrect segmentation that effects the measurement taken for the calculation.

It was noted that in the cases where the V2 segmentation fails, it was usually for lesion images where the colour values of the skin and lesion were close to each other, and therefore difficult to discriminate. The idea behind V2 of the algorithm was to threshold on two different colour channels, and ruling out any that were clearly incorrect, providing a single extra chance of finding a good segmentation. This technique worked to an extent, but clearly the algorithm was limited in that only two colour channels were analysed, so if neither held enough information to segment well, the resulting segmentation was bad.

This idea was extended for V3 of the segmentation algorithm. The new method essentially thresholds across several colour channels, as well as colour clustering with varying parameters, and compares the resulting segmentations with one another to find any 'agreement' between them. In other words, if different techniques identify a lesion area very similar to one another, it is more likely that they are both identifying the subject of the image – the lesion, in theory.

In V3 of the segmentation algorithm, thresholding and colour clustering is performed on each channel of the RGB and CIEL*a*b* colour spaces. The resulting binary masks are searched for their contours, and any contours smaller than $1/50^{th}$ of the image size, or any that extend to the image border are discarded. The remaining contours from each method are added to a list containing all of the segmentation candidates. A pair-wise search of this list is done, and each pair is compared by shape matching, based on the hu-moment values of the shape of each candidate segmentation. The pair that have the best match is assumed as correct, and they are combined and returned as the final segmentation.

4.5.3 Features: Revised

The features were extended in hopes to try to capture more information that could lead to improved classification accuracy. Of the ABCD feature set, only the colour and differential features were added to. The initial colour features were histogram statistics calculated from the skin and lesion areas on each of the RGB colour channels. To extend this, histogram statistics were also calculated for each channel of the CIEL*a*b* colour space. The initial differential (texture) features were calculated as averages over each of the RGB colour channels. The revised differential feature set now includes the same statistics but separated out, instead of averaged over the colour channels. The new feature set also includes the same statistics calculated from the CIEL*a*b* colour channel as well.

4.5.4 Refactoring

After the critical system functionality was implemented, a refactor of the codebase was underway. The reasoning behind this was to improve maintainability, readability, and extensibility of the system. More specifically, I/O (input/output) capabilities were extended to improve efficiency in diagnosing and evaluating the system, for instance writing intermediary working images to the disk and outputting the results of tests and Weka classification with various parameters.

Starting from the very top level, the Startup class simply instantiates the Coordinator class and calls its run() function with string arguments input by the user. These arguments state which folders are to be processed by the system and which folders are to be used for any output from the system.

The Coordinator class provides the coordination of the system. The Coordinator.run() function passes the folder names from Startup to the Coordinator.processFolder() function, which will process every image in the specified input folder, and output any results to the specified output folder. From within the Coordinator.processFolder() function, the FeatureProcessor() class is instantiated with a list of features to be calculated. From there, the FileListProvider() is instantiated to fetch the list of files to be processed, and the FeatureProcessor.getResults() function is called, and instantiates the ImagesFactory, which uses the various Image Processors to create the pre-processed and segmented images for use in feature extraction. The FeatureFactory class is then used to instantiate the feature processors, and their getResult() function is called to return the results which are finally written to the Weka ARFF file using the arffWriter() class. After the ARFF file is written for all images in the input folder, the MachineLearningController() class is used to perform pre-processing, feature selection, and classification.

4.6 Ablation Study

To discover which of the calculated features are most important for obtaining correct classifications, ablation was used on the feature set. Moreover, with the recent addition of 52 new features, the ‘curse of dimensionality’ was a new cause for concern. As the feature vector increases in size without the inclusion of additional training samples, the feature space can quickly become more sparse, which can lead to classifiers that rely on pairwise Euclidian distance in the feature space, such as kNN, to overfit.

The CAD system’s feature set can be considered to consist of four parts, one for each step of the ABCD dermoscopy algorithm. Weka allows for the ARFF file to be split up by features, so 4 new ARFF files were created, with subsets A, B, C, and D of the total features from the SMOTE class balanced data. The classification accuracy was then compared between each of the feature subsets to understand which subsets contributed most in terms of correctly classifying a lesion image. A Random Forest classifier was used to compare the feature sets, as it provided the most reliable scores post-class balancing in previous tests. ANOVA was also used to statistically evaluate any differences found between the groups. The results of these tests are revealed in section 5.2.2.

4.7 Skin Type Analysis

The motivation behind analysing the performance of the system against different skin types is to understand how biases present in skin lesion datasets in terms of skin tone representation can influence the development of a CAD skin cancer system, and its diagnostic effectiveness. Ideally, the CAD system should be able to classify accurately for any skin tone, but this is not always the case.

As stated in section 3.6, this grouping is only an *estimation* and as such, no absolute conclusion can be drawn from this analysis. Skin type was estimated using ITA (individual topology angle), and each dataset was split into groups according to this measure. The ITA was calculated for the skin only, and not the lesion area, therefore, each image had to be segmented using the CAD system's V3 segmentation algorithm to separate the skin from the lesion area (see Section 4.3, Figure 35). Since the segmentation algorithm is not 100% accurate, there are some discrepancies in the grouping which are somewhat mitigated by the large sample size.

5. Results and Evaluation

5.1 Segmentation

The HAM10000 dataset included curated segmentations courtesy of (Tschandl). These can be used to quantitatively compare the algorithms using a Jaccard index (IoU). In this case, the Jaccard index can be considered as the algorithm's accuracy at locating the correct lesion area, and is calculated for all 10015 lesion images in the dataset. The maximum value for the Jaccard index is 1.0, and this score is only achieved if and only if the segmentation is pixel-wise identical to the curated segmentation, and any incorrect pixels will decrease the score.

5.1.1 Jaccard index

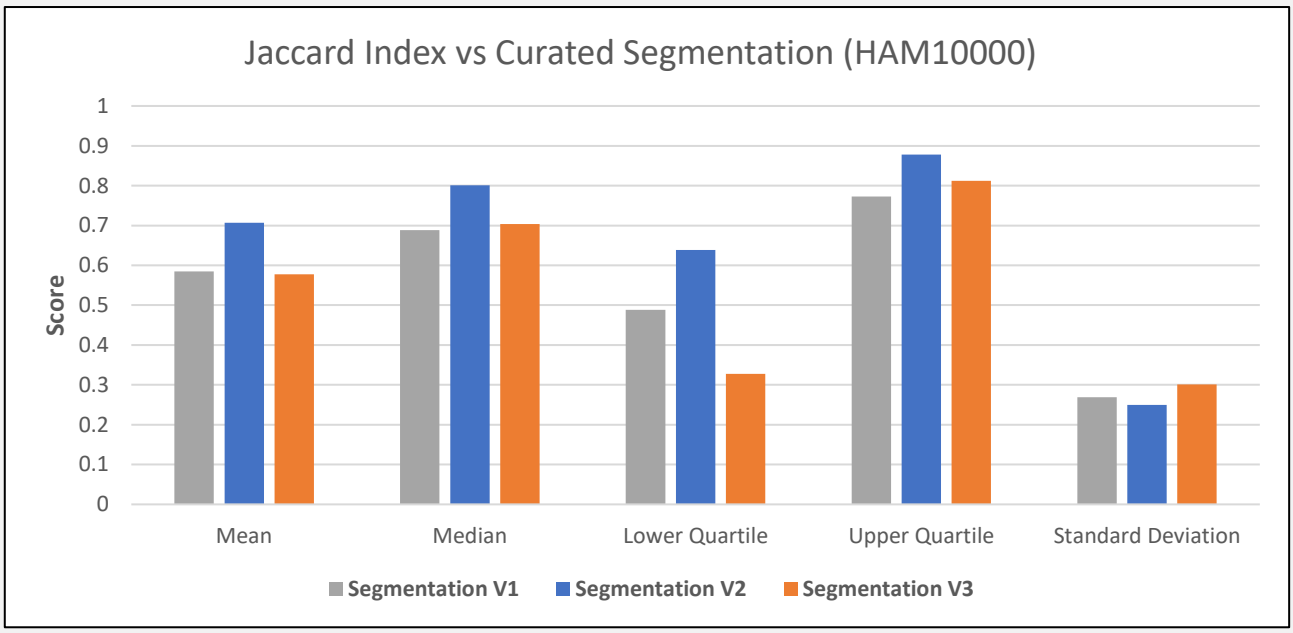


Figure 48: Jaccard Index comparison, Segmentation V1 vs V2 vs V3 (HAM10000)

From the results shown in Figure 48, it is easy to see that the Segmentation V2 algorithm outperformed V1 and V3 in all aspects in terms of the Jaccard index against the curated segmentations. As well as V2 having significantly improved accuracy over the mean, median, and lower and upper quartiles, the results for V2 also have less variance when compared to the alternative algorithms. From this information, it is *possible* to conclude that the V2 algorithm should be used instead of V1 or V3, as it the most accurate and reliable of the three. The V2 algorithm provided an additional ~20% accuracy over the newer V3 algorithm (70.1% vs 57.7%). However, this evaluation contradicted the assumptions made when developing the V3 algorithm. In detail, the improvements made between V2 and V3 were intended to reduce the likelihood of errors in segmentation, by providing ten-fold more options for finding a good segmentation, also involving discarding many segmentations that V2 could have ended up picking.

To verify that the difference in results is statistically significant, and not the result of noise, one-way ANOVA (analysis of variance) was performed on the Jaccard index results for V1 against V2, and V2 against V3. The one-way ANOVA revealed that there was indeed a statistically significant difference in mean Jaccard index between V1 and V2, and between V2 and V3.

For V1 vs V2, the F value (1122.953), was much greater than the F-critical value (3.842), indicating a significant statistical difference. This is consolidated by checking the P value, which was exceptionally small (1.358×10^{-239}).

$$\text{Jaccard Index (V1 vs V2):} \quad F(1, 20028) = [1122.953], p = 1.358 \times 10^{-239} < 0.05)$$

$$F\ crit = [3.842]$$

For V2 vs V3, the case was much the same; the F value (1102.664), was considerably greater than the F-critical value (3.842), indicating a significant statistical difference, consolidated by the very small P value (2.044×10^{-235}).

$$\text{Jaccard Index (V2 vs V3):} \quad F(1, 20028) = [1102.664], p = 2.044 \times 10^{-235} < 0.05)$$

$$F\ crit = [3.842]$$

5.1.2 Classification Accuracy

To further inquire into the effectiveness of the segmentation algorithms, ahead of picking the one used for final classification evaluations, it seemed sensible to compare the algorithms in terms of their ability to classify correctly.

F1, MCC, and ROC Area were the metrics used to evaluate the classification results. Because of the way different classifiers treat the same data slightly differently, the classification metrics were compared for Random Forest, Bayes Net, and kNN.

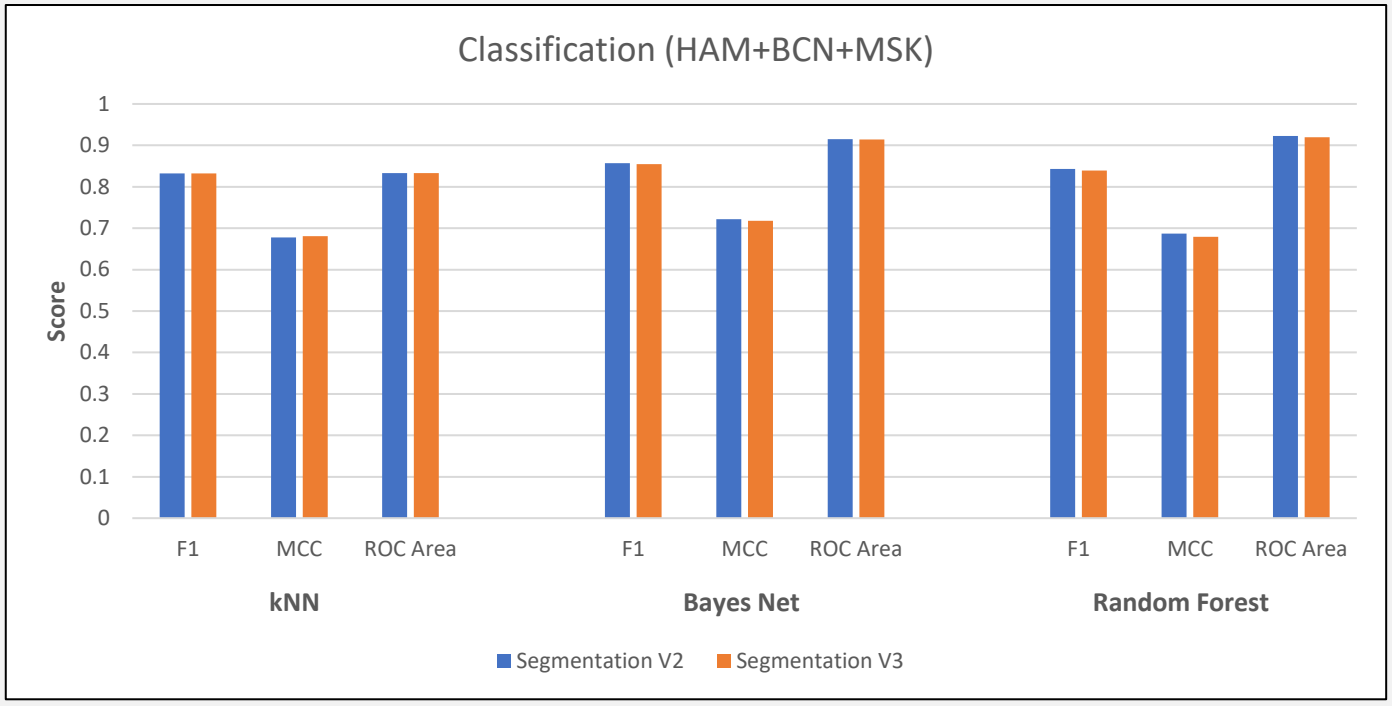


Figure 49: Classification comparison using F1, MCC, and ROC Area measures to compare segmentation performance for all 3 datasets combined

From these results, it can be noted that the two segmentation algorithms perform similarly in terms of classification accuracy. While the V2 technique provides marginal improvements to the accuracy of classification, the difference in the mean Jaccard index between V2 and V3 was around 20%, and this degree of difference is not present when comparing any of the classification metrics.

The classification's accuracy, F1-measure, MCC, and ROC area measures were scrutinised for each classifier using ANOVA testing to determine if the V2 segmentation algorithm performs better than the V3 algorithm in terms of classification efficacy. The ANOVA tests were performed for each classifier scored above (kNN, Bayes Net, and Random Forest). The input for each ANOVA test was the model's evaluation measures (accuracy, F1, MCC, ROC) from each fold of a 10-fold cross validation.

Measure	Classifier	F1	P-value	F crit
Accuracy	kNN	0.319	0.86	4.414
Accuracy	Bayes Net	0.758	0.395	4.414
Accuracy	Random Forest	1.551	0.229	4.414
F1	kNN	0.001	0.971	4.414
F1	Bayes Net	0.628	0.438	4.414
F1	Random Forest	1.562	0.227	4.414
ROC Area	kNN	0.012	0.916	4.414
ROC Area	Bayes Net	0.808	0.381	4.414
ROC Area	Random Forest	1.554	0.228	4.414
MCC	kNN	0.69	0.971	4.414
MCC	Bayes Net	1.612	0.22	4.414
MCC	Random Forest	1.506	0.236	4.414

Table 12: ANOVA test results for accuracy, F1, ROC Area, and MCC measures

For all of the ANOVA tests conducted, there was no significant statistical difference in the results between V2 and V3 of the segmentation algorithm, indicating that both algorithms perform about the same. The F crit value for the following tests was 4.414, which was not reached by any of the test's F scores. Additionally, the P-values for each ANOVA test were all higher than the significance factor 0.05. In conclusion, either segmentation algorithm could be used interchangeably to obtain virtually the same classification accuracies, according to ANOVA testing. For the purposes of evaluation from here in the report onwards, the V3 algorithm was used, as it produced fewer outliers (incorrect calculations) when calculating the border features in the feature extraction stages (see Table 13).

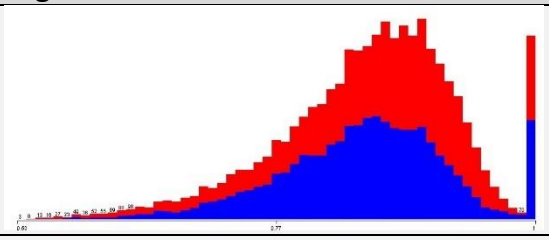
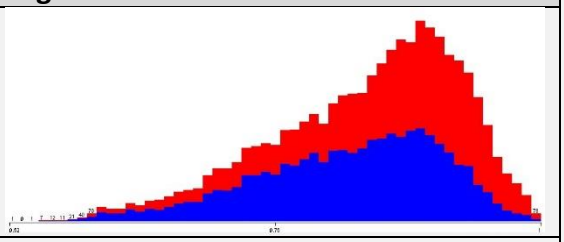
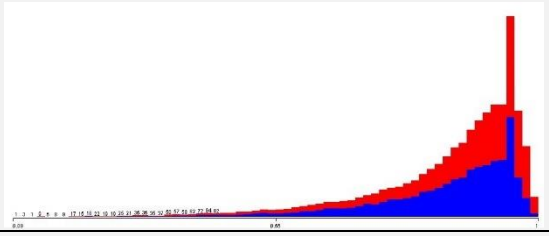
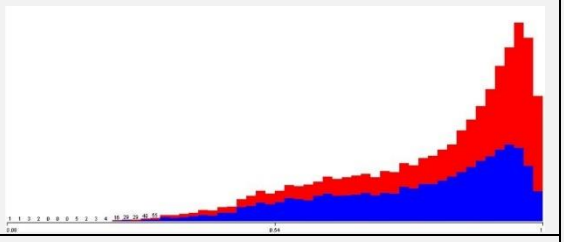
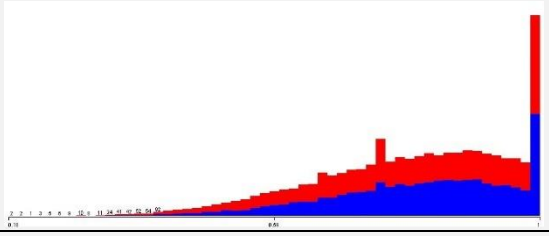
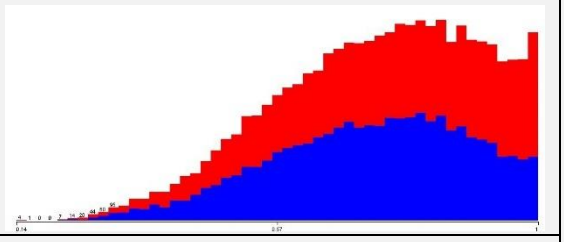
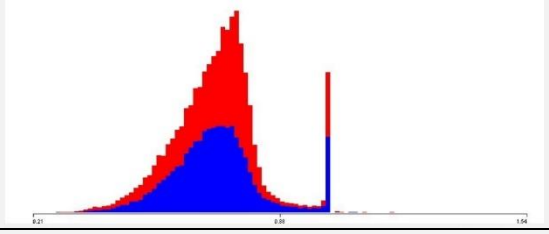
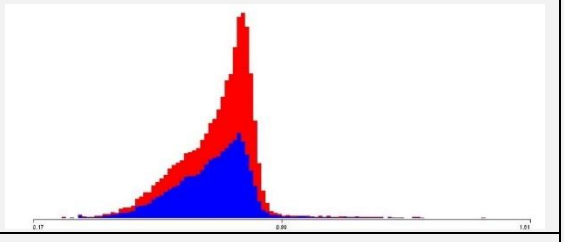
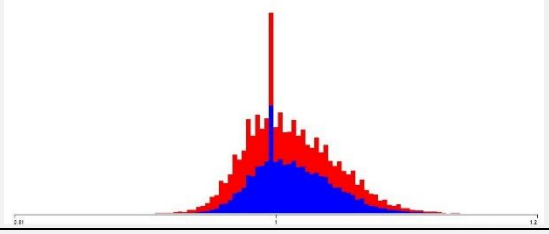
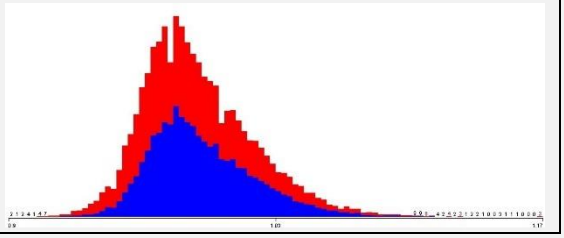
	Segmentation V2	**Segmentation V3**
Symmetry		
Circularity		
Elongation		
Rectangularity		
Fractal Dimension		

Table 13: Feature result histograms showing outliers present for V2 and not present for V3

5.2 Features

5.2.1 Old vs New Feature Set

The implementation of additional features in the system prompted an evaluation as to how these new features affect the efficacy of the CAD system's classification. Therefore, classification performance was measured over all images from each of the three datasets.

For the following tests, the segmentation V3 algorithm was used to locate the lesion area and produce the binary mask, which was analysed according to each version of the feature vector separately. After processing the images and extracting the features, the old feature vector contains 35 attributes characterising the lesion in terms of A,B,C, and D. The new feature vector contains 87 attributes covering the same characteristics, but in a little more detail. The two resulting Weka ARFF files were used for Random Forest classification with 10-fold cross validation. Random Forest classification was used as it gives reliable performance all-round for previous tests.

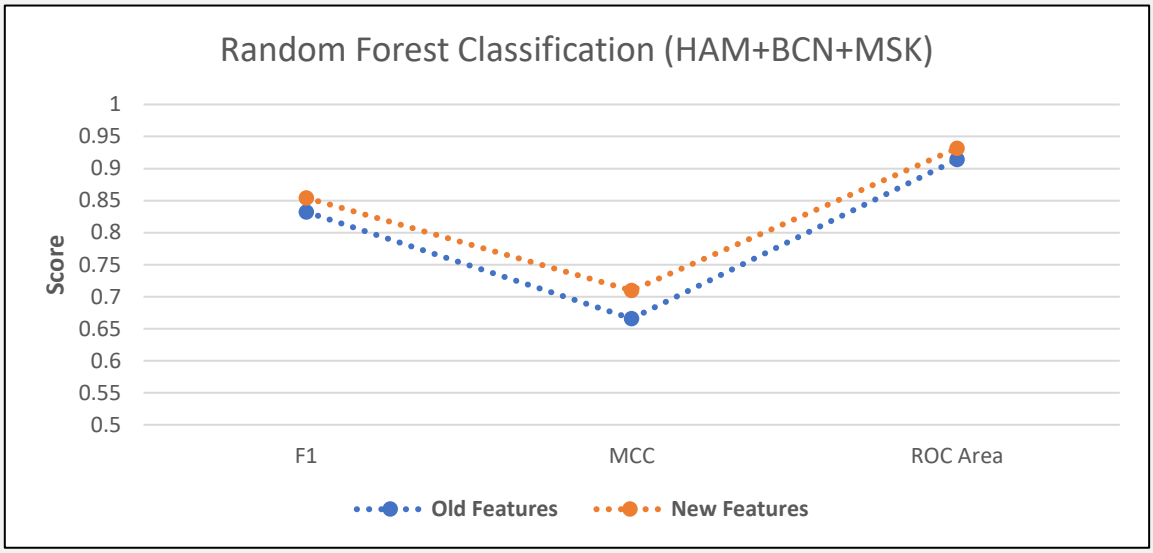


Figure 50: Classification measures for old and new features

An ANOVA test was conducted to determine if the difference in performance between the new and old features was statistically significant. The input for the ANOVA test was the classification accuracy for each fold of the 10-fold cross validation, for the new and old features. The mean accuracy for classification using the old features was 83.253, and the mean accuracy using the new features was 85.436. The ANOVA test revealed that there was a significant difference between the two groups.

$$\text{Accuracy (Old vs New):} \quad F(1, 18) = [69.379],\ p = 1.373 \times 10^{-7} < 0.05)$$

$$F\ \text{crit} = [4.414]$$

From the results gained through comparing classification scores and verifying the difference with ANOVA, it was decided that the new features were to be used for final classification evaluations.

5.2.2 Feature Ablation & Selection

By investigating how the CAD system operates when only using a subset of the available features, it can lead to insight into which features are most important for correct classification. In combination with attribute selection techniques provided with Weka, the goal was to reduce the size of the feature vectors without compromising on classification accuracy, if possible.

To be able to compare the scores of classification using any feature subset, a baseline score was needed. As stated in section 4.6, a Random Forest classifier was used for comparison, and the baseline score for the following tests, shown in Table 14, were obtained by 10-fold cross validation on the complete feature set.

	Size	Accuracy	F1	ROC Area	MCC
A+B+C+D	87	85%	0.854	0.932	0.708

Table 14: Classification measures using the full feature set

The first test involved splitting the feature set into subsets of features based on the A,B,C, and D sections of the ABCD dermoscopy algorithm. The feature subset A consists of a single feature – asymmetry. Feature subset B consists of 8 features characterising the lesion's border. Subset C consists of 36 features that describe the colour of the lesion and the surrounding skin, and subset D consists of 42 features from the analysis of the lesion's texture. The results were again obtained by 10-fold cross validation using the Random Forest classifier.

	Size	Accuracy	F1	ROC Area	MCC
A	1	85%	0.845	0.875	0.727
B	8	74%	0.741	0.824	0.483
C	36	83%	0.829	0.909	0.658
D	42	86%	0.858	0.93	0.715
A+B+C+D	**87**	**85%**	**0.854**	**0.932**	**0.708**

Table 15: Classification measures for A,B,C, and D feature subsets

The green shaded areas show where the classification measure was better or equal to the baseline score, and the red shaded areas show where the measure was worse. This data shows that there are cases where classifying using only a subset of the original features leads to improvements in classification measures. In fact, using only subset D for feature extaction produces results that are almost indistinguishable compared to the baseline; 3 of the 4 key measures were marginally improved, and the remaining measure was marginally worse. Curiously, subset A, only consisting of a single feature, was able to match or exceed 2 of the 4 baseline measures. These results definitively show that application of feature selection can be used to reduce the size of the feature vector, while maintaining, or even improving the classification accuracy.

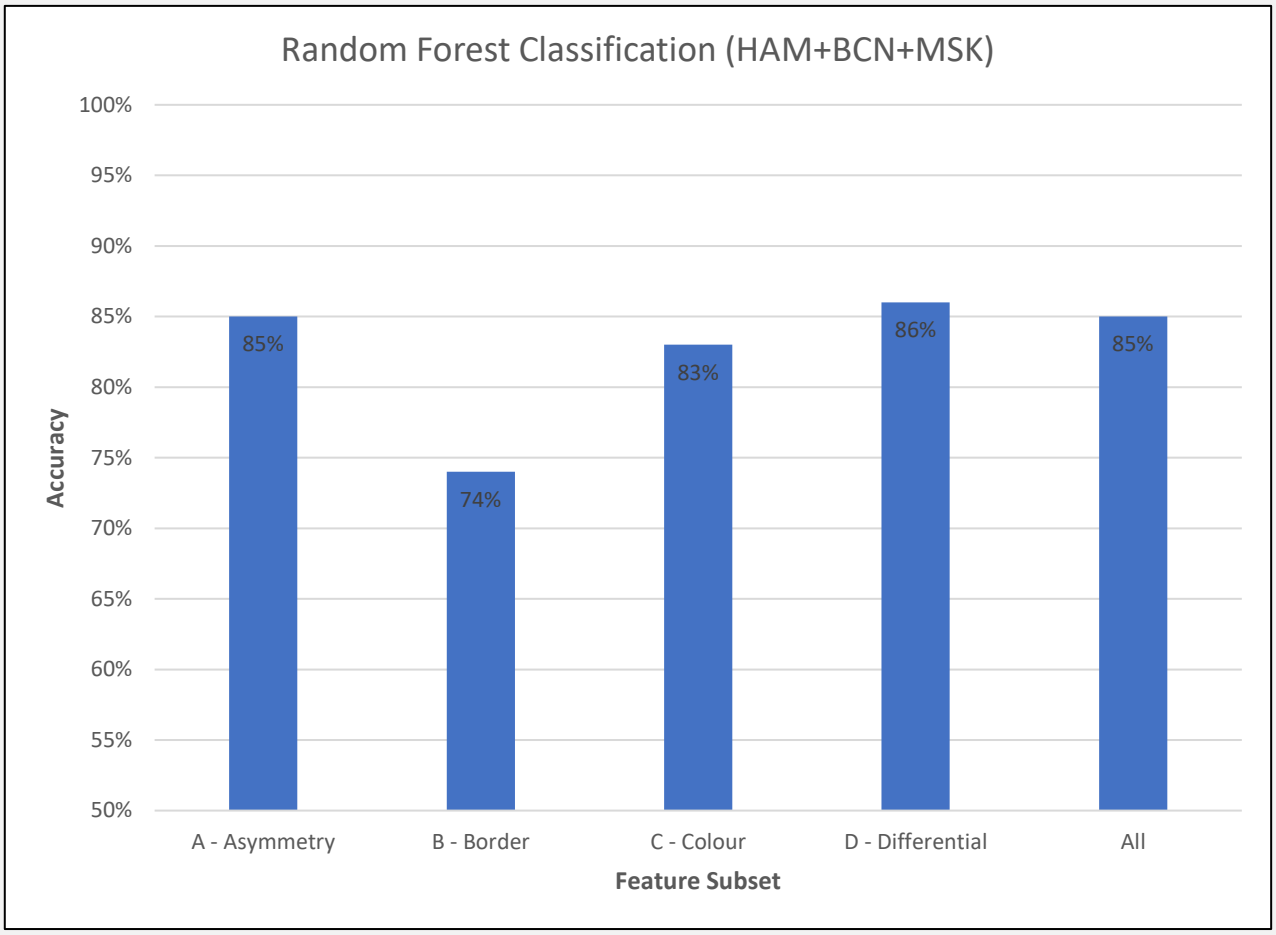


Figure 51: Random Forest classification accuracy for feature subsets A,B,C,D, and full feature set

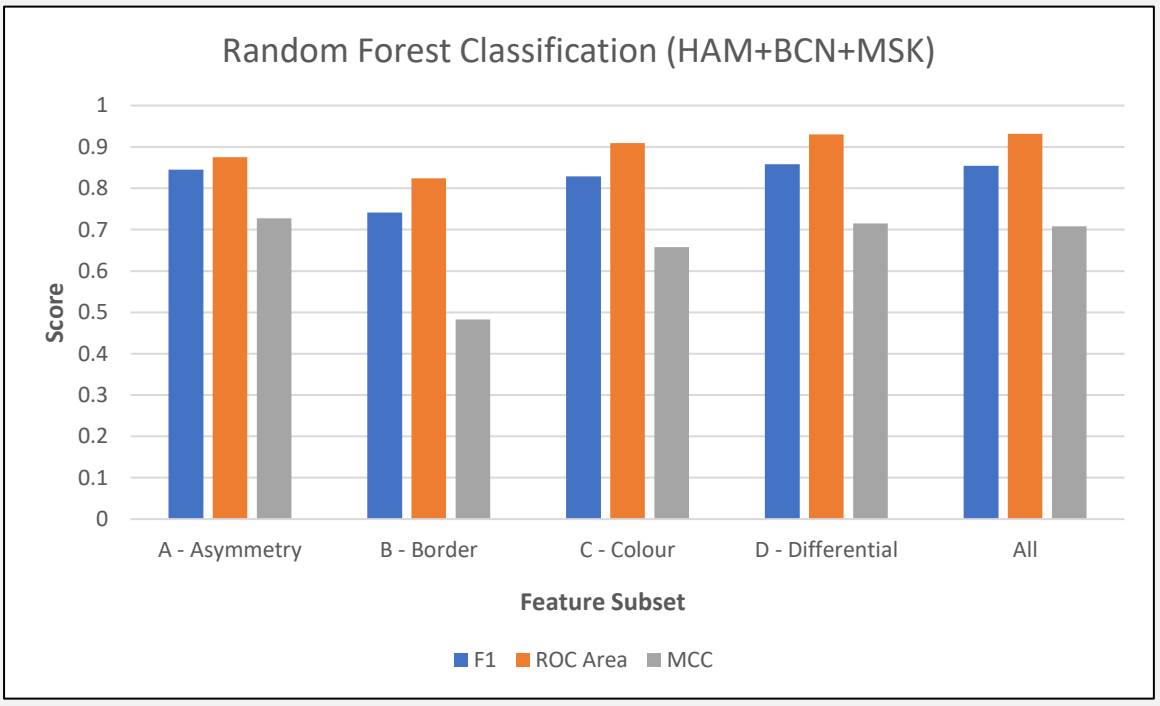


Figure 52: Random Forest classification measures for feature subsets A,B,C,D, and full feature set

To further examine the features, and to confirm or deny the theory that using only a subset of the total features can give equivalent classification accuracy, Weka's attribute selection tools were utilised. 'OneRAttributeEval' evaluates the worth of an attribute using a OneR (One Rule) classifier, which transforms the numerical features to categorical, before generating rules for each feature by constructing frequency tables, and predicts using the rule with the smallest error. 'InfoGainAttributeEval' evaluates the worth of an attribute by measuring the information gain with respect to the class, and 'GainRatioAttributeEval' evaluates an attributes worth by the information gain *ratio* with respect to the class. Each of these three evaluators rank the features slightly differently, the top 20 ranked features for each evaluator are colour-coded and shown in Table 16.

		OneRAttributeEval		InfoGainAttributeEval		GainRatioAttributeEval	
Key	Rank	# Feature	Score	# Feature	Score	# Feature	Score
A	1	8 fractalDimension	85.162	8 fractalDimension	0.476	69 lEnergy	0.070
	2	1 asymmetry	84.519	69 lEnergy	0.444	48 bEnergy	0.070
	3	68 lCorrelation	84.256	55 gEnergy	0.434	55 gEnergy	0.068
	4	47 bCorrelation	84.066	76 aEnergy	0.433	76 aEnergy	0.068
	5	7 solidity	84.058	48 bEnergy	0.428	8 fractalDimension	0.067
B	6	71 lHomogeneity	84.020	68 lCorrelation	0.418	75 aCorrelation	0.064
	7	50 bHomogeneity	83.970	47 bCorrelation	0.415	54 gCorrelation	0.064
	8	54 gCorrelation	83.970	83 bbEnergy	0.414	62 rEnergy	0.063
	9	85 bbHomogeneity	83.906	62 rEnergy	0.406	83 bbEnergy	0.063
	10	75 aCorrelation	83.871	75 aCorrelation	0.399	68 lCorrelation	0.061
C	11	64 rHomogeneity	83.853	54 gCorrelation	0.399	47 bCorrelation	0.061
	12	57 gHomogeneity	83.820	61 rCorrelation	0.361	7 solidity	0.060
	13	78 aHomogeneity	83.711	1 asymmetry	0.333	82 bbCorrelation	0.059
	14	83 bbEnergy	83.646	82 bbCorrelation	0.317	1 asymmetry	0.059
	15	62 rEnergy	83.633	7 solidity	0.304	61 rCorrelation	0.057
D	16	6 rectangularity	83.582	6 rectangularity	0.288	6 rectangularity	0.054
	17	82 bbCorrelation	83.577	4 convexity	0.269	71 lHomogeneity	0.051
	18	5 elongation	83.557	71 lHomogeneity	0.263	50 bHomogeneity	0.050
	19	55 gEnergy	83.504	50 bHomogeneity	0.256	4 convexity	0.050
	20	61 rCorrelation	83.468	2 circularity	0.213	2 circularity	0.049

Table 16: Top 20 ranked attributes by 3 different attribute evaluators

From Table 16, it is clear that the subset D's features are the most frequently occurring, which correlates with the classification scores obtained using only these features. It can also be noted that no colour features appear in the top 20 ranked features for any of the evaluation methods, despite the feature subset C performing only 2% worse in terms of classification accuracy compared to the full ABCD feature set. Conversely, 4 or more border features are ranked in the top 20 for each of the three selected methods, despite feature subset B performing 11% worse than the full feature set in terms of accuracy. Curiously, the Radial Variance and Compactness features from subset B are not ranked in the top 20 for any of the three evaluators. One can assume that the discrepancy in these cases is because the total worth of a particular subset of features is not always equal to the sum worth of the subset's individual features. In other words, as features interact with one another in the model, they may work with or against each other to produce positive or negative results.

$$\text{worth}(B_{i\ldots n}) \quad \not\equiv \quad \text{worth}(B_i) + \text{worth}(B_{i+1}) + \text{worth}(B_{i+2}) + \ldots + \text{worth}(B_n)$$

From Table 16, OneRAttributeEval and InfoGainAttributeEval rank the fractal dimension feature as the single feature with the most worth for classification, and for GainRatioAttributeEval, the feature is also ranked in the top 5. Similarly, the Asymmetry feature is also present in the top 20 ranked features for all 3 evaluators. These feature subsets were tested in combination with each other to further understand which features were contributing most.

	Size	Accuracy	F1	ROC Area	MCC
A	1	85%	0.845	0.875	0.727
B	8	74%	0.741	0.824	0.483
C	36	83%	0.829	0.909	0.658
D	42	86%	0.858	0.93	0.715
A+B+C+D	**87**	**85%**	**0.854**	**0.932**	**0.708**
B_{df}	1	85%	0.851	0.861	0.739
D+B_{df}	43	86%	0.856	0.929	0.712
A+D+B_{df}	44	85%	0.848	0.925	0.696
A+D	43	85%	0.850	0.926	0.700

Table 17: Classification measures for Bdf and combined feature subsets. B_{df} = Fractal Dimension feature

Table 17 still shows that while improvements can be made by using feature selection, the improvements are marginal and from the configurations tested so far, and no feature subset outperformed the full feature set across all the key classification measures. The best example of this is the B_{df} (fractal dimension) subset, which despite only consisting of a single feature, sees a greatly improved MCC score compared to the baseline and similar accuracy and F1 scores also. However, the ROC Area score is substantially lower than the baseline.

The D feature subset, containing only textural analysis features from the GLCM method, performs better than the baseline in all but one measure, which trails only a small fraction behind. This indicates that the differential features are the most meaningful of the ABCD algorithm for lesion discrimination in the proposed CAD system. To try and extend the abilities of subset D, additional features were combined that performed well on their own. The table shows that addition of the fractal dimension feature to the D feature subset ($D+B_{df}$) decreases the scores slightly from subset D alone, but not by a meaningful amount. The further addition of the Asymmetry feature decreases the scores again ($A+D+B_{df}$).

Of the configurations tested thus far, it appears as though the subset D alone produces the best and most reliable scores, followed by $D+B_{df}$ and A+B+C+D. Notably, of the features in subset D, only 3 of the GLCM measures ranked in the top 20 for the three attribute evaluators: correlation, energy, and homogeneity. Therefore, another configuration $D_{C,E,H}$ was tested using only these features.

	Size	Accuracy	F1	ROC Area	MCC
A	1	85%	0.845	0.875	0.727
B	8	74%	0.741	0.824	0.483
C	36	83%	0.829	0.909	0.658
D	42	86%	0.858	0.930	0.715
A+B+C+D	**87**	**85%**	**0.854**	**0.932**	**0.708**
B_{df}	1	85%	0.851	0.861	0.739
D+B_{df}	43	86%	0.856	0.929	0.712
A+D+B_{df}	44	85%	0.848	0.925	0.696
A+D	43	85%	0.850	0.926	0.700
$D_{C,E,H}$	18	85%	0.853	0.920	0.712

Table 18: Classification measures for B_{df} and combined feature subsets. $D_{C,E,H}$ = Correlation, Energy, and Homogeneity GLCM features

One-way ANOVA testing was performed on the 10-fold classification accuracy scores for subset D, D+B_{df}, and A+B+C+D. The test found that there was no significant statistical difference in accuracy between the groups. Pairwise ANOVA testing on the same groups also did not reveal any statistical difference between any pair of groups.

$$\text{Accuracy (D vs D+B}_{df}\text{ vs A+B+C+D):} \qquad F(2, 27) = [0.807],\ p = 0.456 > 0.05)$$

$$F\ \text{crit} = [3.354]$$

Another Weka function used to evaluate the features was CfsSubsetEval, which selects a subset of features that are most correlated with the class, while also having low intercorrelation. The 20 features chosen as the best feature subset is shown in Table 19 in no particular order. In contrast to the three previous evaluators, the resulting subset from CfsSubsetEval includes features from all 4 of the ABCD feature subsets, indicating that a combination of different features may lead to an improved model. Classifying using the features selected with this evaluator gave somewhat good results, but did not outperform most other configurations, shown in Table 20.

	Size	Accuracy	F1	ROC Area	MCC
A	1	85%	0.845	0.875	0.727
B	8	74%	0.741	0.824	0.483
C	36	83%	0.829	0.909	0.658
D	42	86%	0.858	0.930	0.715
A+B+C+D	**87**	**85%**	**0.854**	**0.932**	**0.708**
B_{df}	1	85%	0.851	0.861	0.739
D+B_{df}	43	86%	0.856	0.929	0.712
A+D+B_{df}	44	85%	0.848	0.925	0.696
A+D	43	85%	0.850	0.926	0.700
$D_{C,E,H}$	18	85%	0.853	0.920	0.712
CfsSubsetEval	20	78%	0.784	0.867	0.572

Table 20: Classification measures for CfsSubsetEval features

1	asymmetry
2	circularity
4	convexity
6	rectangularity
7	solidity
8	fractalDimension
24	skinRMedian
33	lesionBbMedian
42	skinBbMedian
47	bCorrelation
48	bEnergy
54	gCorrelation
55	gEnergy
62	rEnergy
68	lCorrelation
69	lEnergy
75	aCorrelation
76	aEnergy
82	bbCorrelation
83	bbEnergy

Table 19: CfsSubsetEval output

5.3 Classification

To rigorously evaluate the effectiveness of the CAD system for classification, multiple classifiers are used, as well as different configurations, to try to obtain the best results. The results detailed in this section were all obtained *after* using SMOTE to balance the classes equally because it produces better results, as established in section 4.4.4. Additionally, as no feature selection was performed in this section as no feature subset was found that clearly outperformed the full feature set. Hence, the following results were obtained using all 87 features.

As stated in section 4.4, classification was performed using K-Nearest Neighbours, Random Forest, and Bayesian Network classifiers with Weka. These classifiers were chosen as they have been used in previous studies relating to CAD skin cancer systems, and provided the best results in experiments while maintaining low computation time.

5.3.1 HAM10000

The first evaluations are made on the original dataset – HAM10000. This dataset includes 10015 lesion images, 8563 of which are cases of benign lesions, and the remaining 1452 are malignant cases. After SMOTE class balancing, there are 8563 instances of each class, totalling to 17126 training instances. 10-fold cross validation was used to obtain classification metrics that will assist in evaluating the CAD system's effectiveness. In terms of accuracy, the Bayesian Network classifier narrowly beats the Random Forest classifier by 0.4%, which beats the kNN classifier by a slightly larger margin of 0.9%. Overall, the accuracies between classifiers are similar enough that additional metrics are needed to decide which is best.

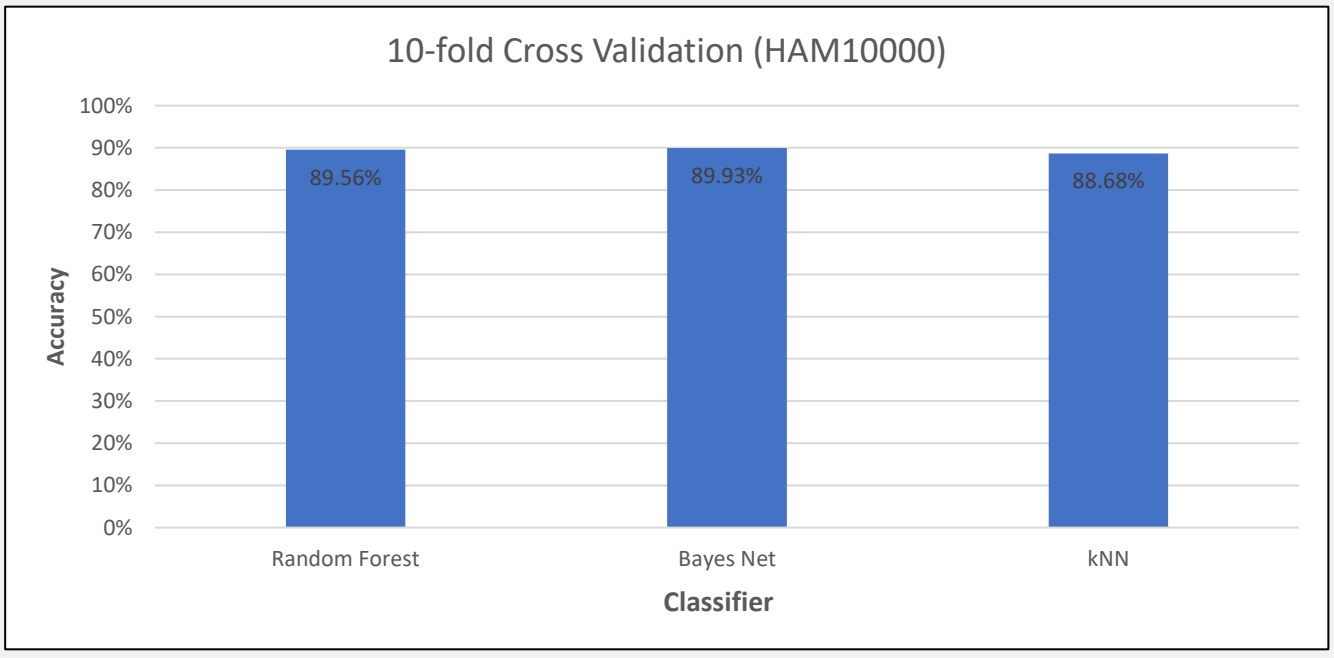


Figure 53: Classification accuracy over 10-fold cross validation for 3 classifiers using HAM10000

The results for the weighted F-Measure show much the same story as for the accuracy, and the Bayesian Network outperforms again in this metric by 0.5%. For the Matthews Correlation Coefficient, the Bayesian network leads by less than 0.1%. For the ROC Area metric however, the Random Forest classifier has the highest score of the three, showing that this classifier was 1.1% better at distinguishing between the two classes than using the Bayesian Network. Once again, it would be difficult to decide which classifier is truly best at this point.

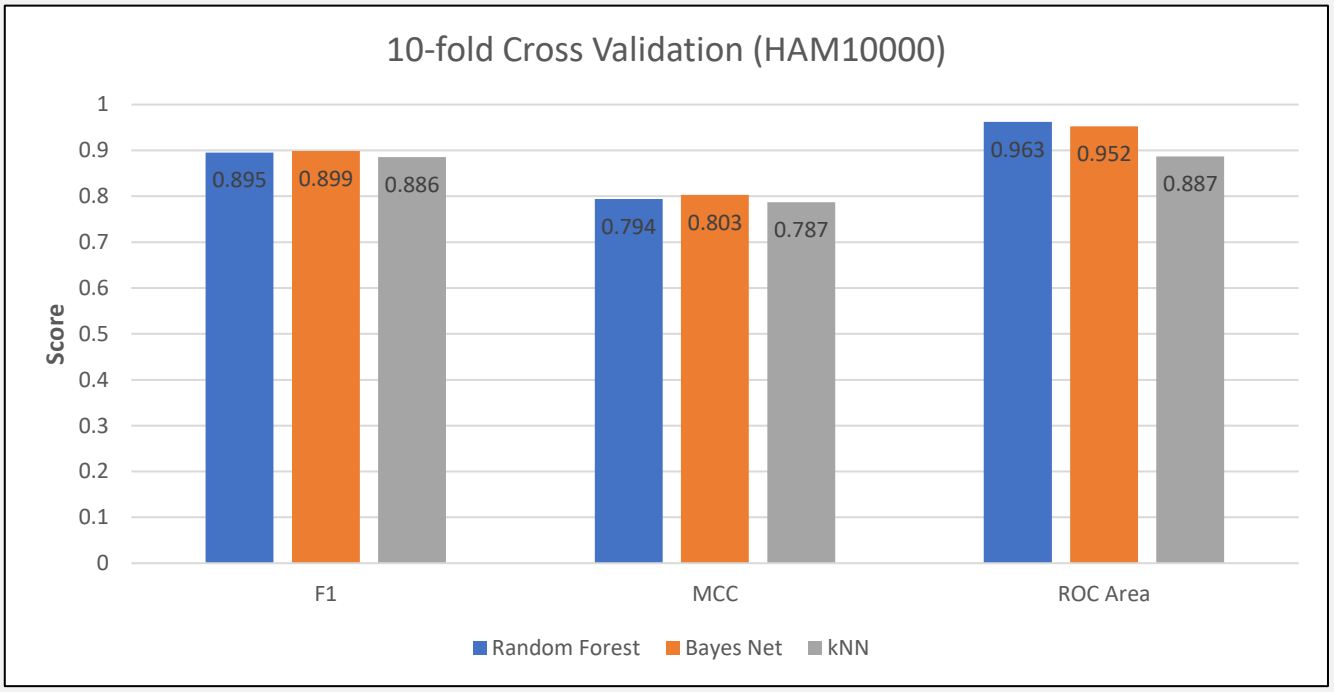


Figure 54: Classification measures F1, MCC, and ROC Area over 10-fold cross validation for 3 classifiers using HAM10000

Another consideration, particular to the problem of distinguishing between benign and harmful skin lesions, is whether the CAD system is mis-diagnosing benign lesions more frequently than malignant lesions. A practical example of where this could be an issue is an occurrence of the CAD system diagnosing a benign lesion as malignant, which may cause unnecessary distress to the patient and inevitably lead to follow-up examinations and further testing before it would be confirmed as truly benign. On the other hand, it could be much riskier to diagnose a malignant lesion as benign, for obvious reasons. The confusion matrices from the 10-fold cross validation, as well as the per-class precision and recall metrics, can highlight any occurrence of this class bias.

Random Forest | 10xVal		
classified as ->	MALIGNANT	BENIGN
MALIGNANT	8019	544
BENIGN	1244	7319

	MALIGNANT	BENIGN
Precision:	0.866	0.931
Recall:	0.936	0.855

Bayes Net | 10xVal		
classified as ->	MALIGNANT	BENIGN
MALIGNANT	7246	1317
BENIGN	408	8155

	MALIGNANT	BENIGN
Precision:	0.947	0.861
Recall:	0.846	0.952

kNN (1) | 10xVal		
classified as ->	MALIGNANT	BENIGN
MALIGNANT	8389	174
BENIGN	1765	6798

	MALIGNANT	BENIGN
Precision:	0.826	0.975
Recall:	0.98	0.794

Table 21: Confusion matrices and precision and recall scores for 3 classifiers over 10-fold cross validation

The confusion matrix for the Random Forest classification reveals that there are fewer malignant lesions classified as benign, than there are benign lesions classified as malignant. This is also reflected in the per-class recall statistics which show an 8.1% difference between classes. This is the preferred bias if equal recall rates cannot be achieved across the classes. The same is the case for the kNN classifier, but with a larger difference of 18.6% in per-class recall. However, the opposite happens when using the Bayes Net classifier, the results show that malignant lesions are 10.6% more likely to be misdiagnosed as benign, than for the other way around. In this use case, where misclassifying a malignant lesion is more dangerous than misclassifying a benign lesion, the Random Forest or kNN classifier would be preferred.

5.3.2 BCN20000

The BCN20000 dataset includes 12413 lesion images, 9014 of which are benign cases, and the remaining 3399 are malignant cases. After SMOTE class balancing, there are 9014 cases for each class, totalling 18028 training instances. Again, 10-fold cross validation was used to obtain classification metrics that will assist in evaluating the CAD system's effectiveness.

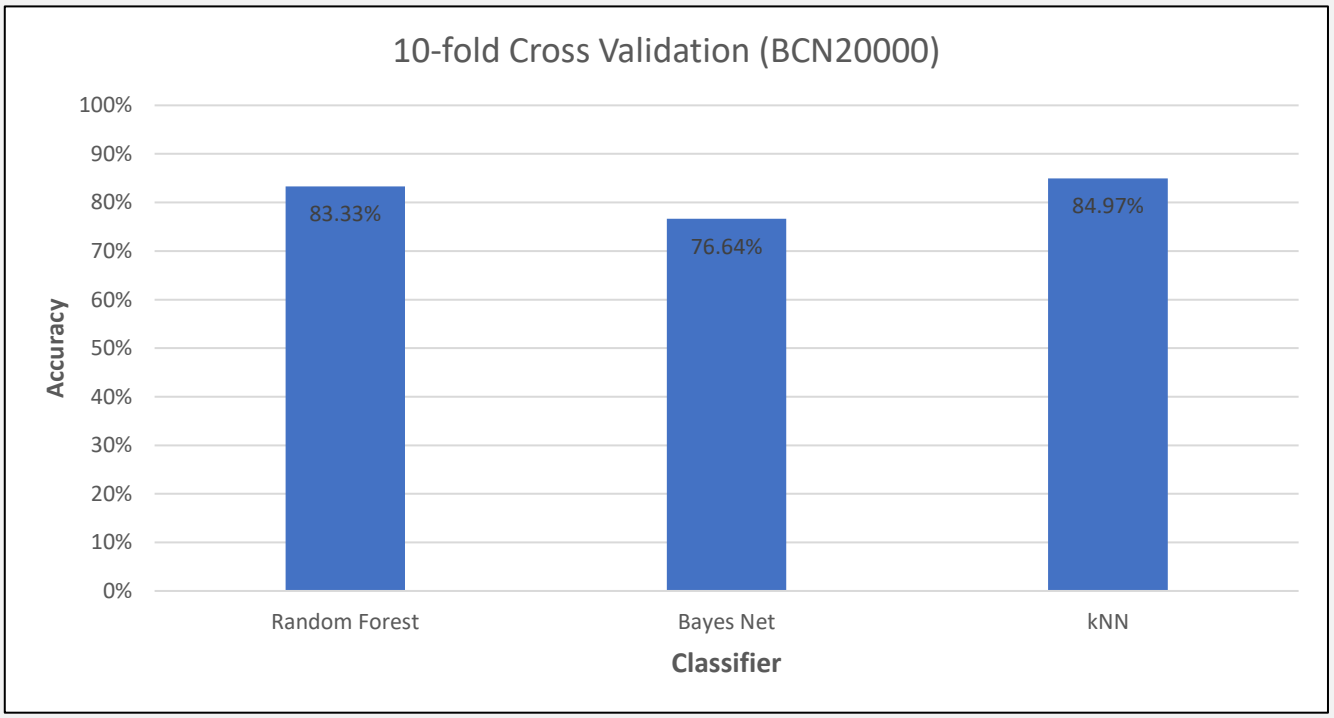


Figure 55: Classification accuracy over 10-fold cross validation for 3 classifiers using BCN20000

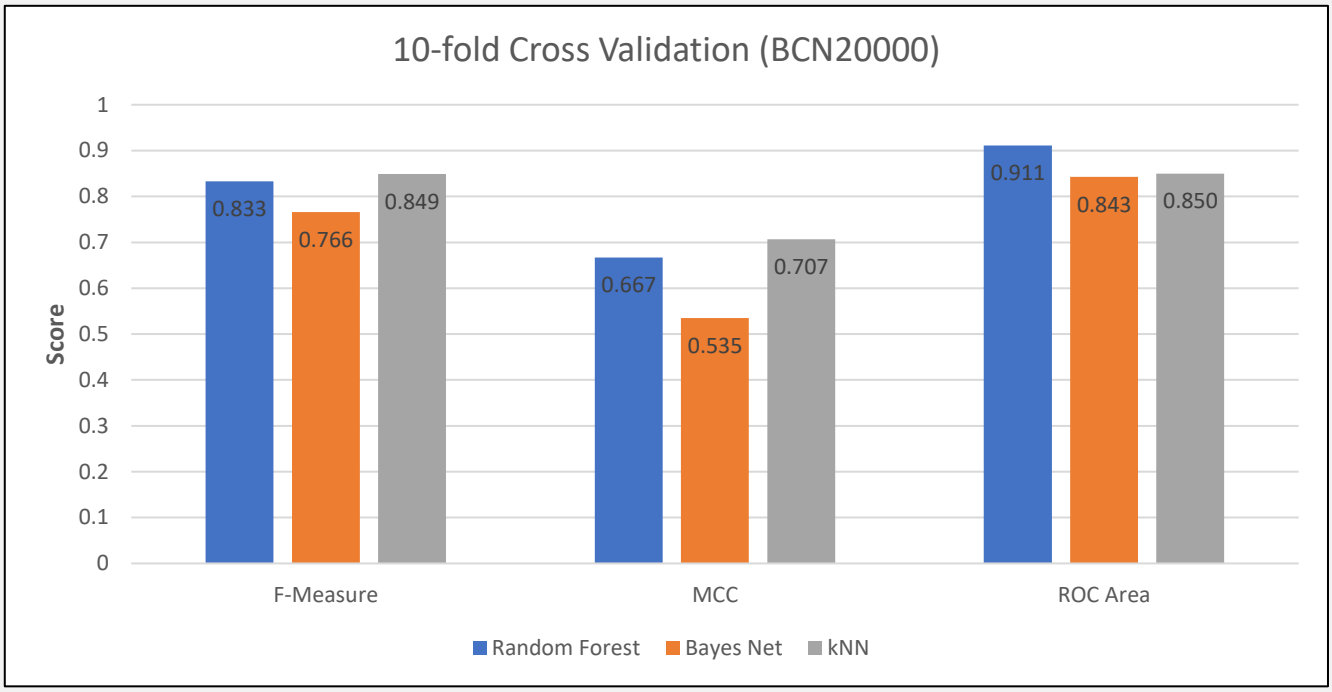


Figure 56: Classification measures F1, MCC, and ROC Area over 10-fold cross validation for 3 classifiers using BCN20000

The results over 10-fold cross validation for the BCN20000 dataset show that the Bayesian Network classifier performs 6.69% worse in terms of accuracy than the Random Forest classifier. And the kNN classifier outperforms Random Forest by only 1.6%. In terms of the other key model performance measures: F-measure, MCC, and ROC Area, the Bayesian Network classifier also sees the worst performance of the three classifiers. Overall, the kNN classifier has the edge over the Random Forest classifier for the BCN20000 dataset, which is in contrast to the evaluation for the HAM10000 dataset, where it performed the worst of the three. Notably, the variance in scores between classifiers is greater than when using the HAM10000 dataset. This could indicate that the quality of the HAM10000 data is greater than that of the BCN20000 dataset.

The confusion matrices and precision and recall metrics reveal any bias in the model towards a particular class. The results from the Bayesian Network classifier report that there is slight bias towards misclassifying malignant lesions as benign rather than the other way around, with an 8.5% difference in recall between the classes. The kNN classifier's results show the opposite case to a high degree, with a 14% difference in per-class recall. The Random Forest classifier has a remarkably even precision and recall metrics, with less than a 1% difference per-class for either metric. For the purposes of a CAD system for skin cancer, the Random Forest classifier is the best of the three for this dataset, as it can classify malignant and benign lesions equally.

Random Forest | 10xVal		
classified as ->	MALIGNANT	BENIGN
MALIGNANT	7552	1462
BENIGN	1536	7478

	MALIGNANT	BENIGN
Precision:	0.831	0.836
Recall:	0.838	0.83

Bayes Net | 10xVal		
classified as ->	MALIGNANT	BENIGN
MALIGNANT	6523	2491
BENIGN	1722	7292

	MALIGNANT	BENIGN
Precision:	0.791	0.745
Recall:	0.724	0.809

kNN (1) | 10xVal		
classified as ->	MALIGNANT	BENIGN
MALIGNANT	8315	699
BENIGN	2011	7003

	MALIGNANT	BENIGN
Precision:	0.805	0.909
Recall:	0.922	0.777

Table 22: Confusion matrices and precision and recall scores for 3 classifiers over 10-fold cross validation

5.3.3 MSK

The MSK dataset consists of 2706 lesion images, 2170 of which are benign cases, and the remaining 536 are malignant cases. After SMOTE class balancing, there are 2170 training instances for each class. The following results were achieved using 10-fold cross validation.

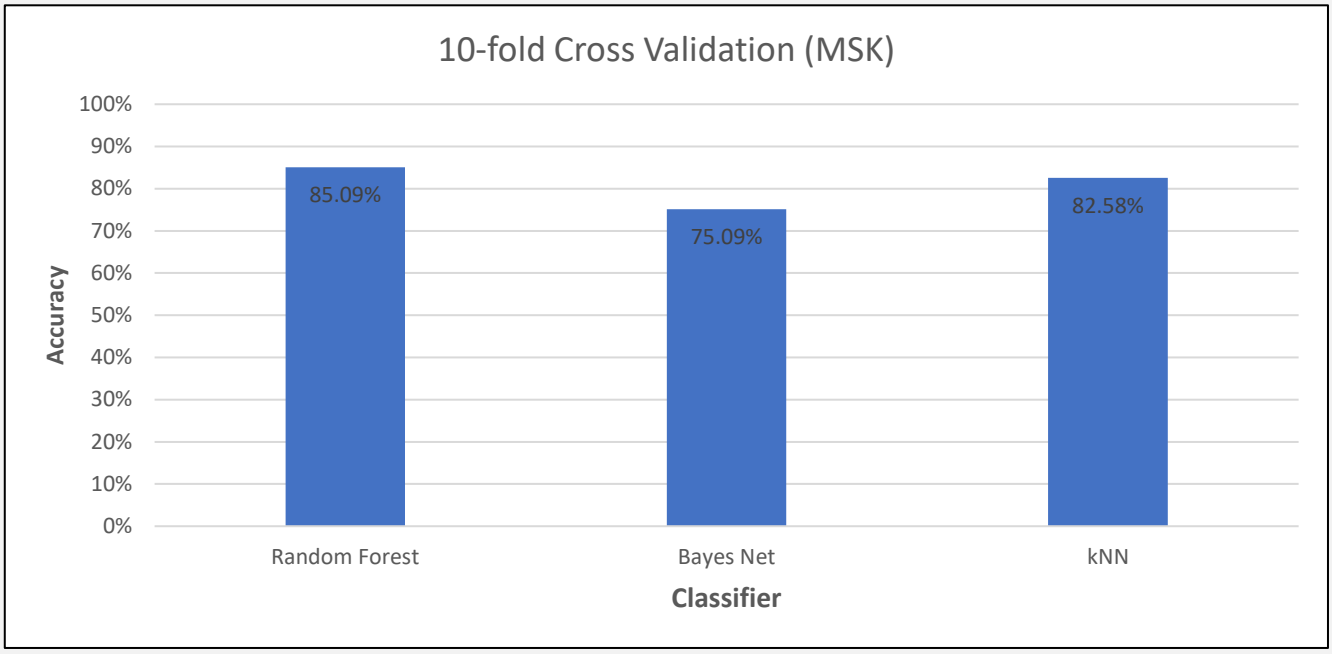


Figure 57: Classification accuracy over 10-fold cross validation for 3 classifiers using MSK dataset

The per-classifier accuracy results show the Bayesian Network classifier performing the worst of the three again, trailing 7.5% behind the kNN classifier. The Random Forest classifier shows the best accuracy of the three classifiers for the MSK dataset, it was 2.5% more accurate than using kNN.

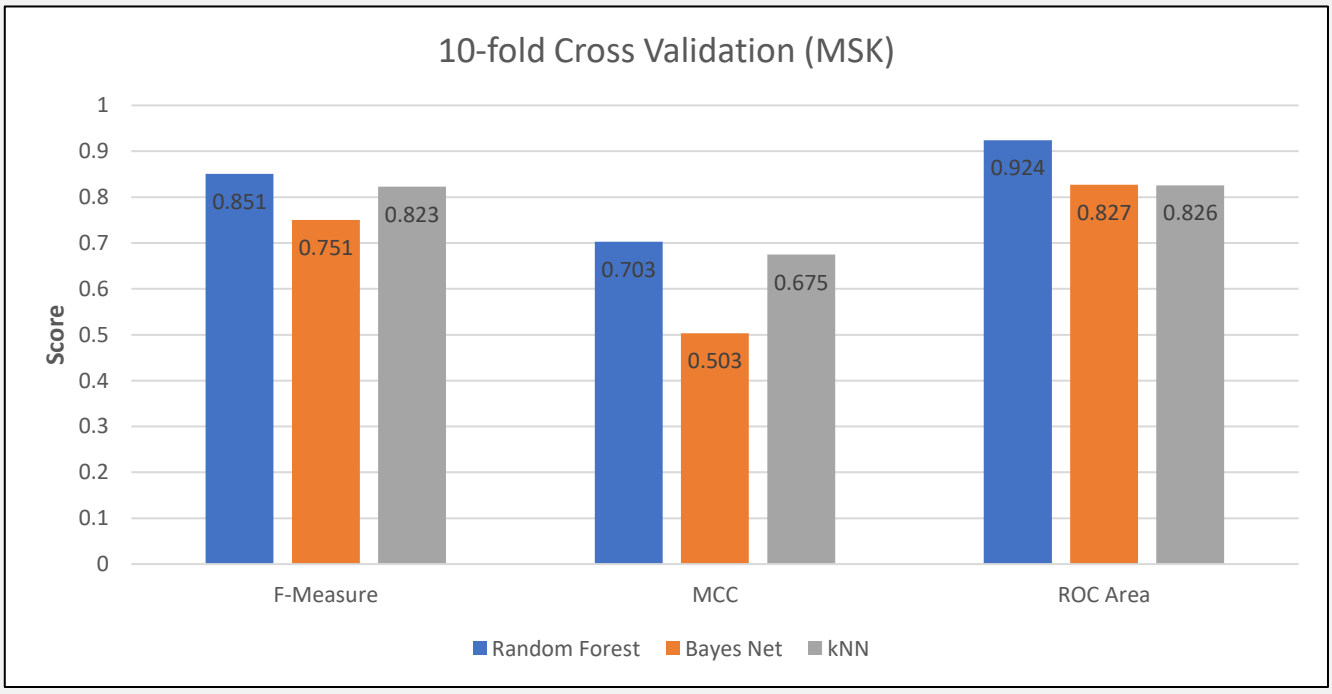


Figure 58: Classification measures F1, MCC, and ROC Area over 10-fold cross validation for 3 classifiers using MSK dataset

The cross validation results' F-Measure, MCC, and ROC Area metrics show that the Random Forest classifier decisively outperforms the other Bayes Net and kNN classifiers on the MSK dataset. The results also show that the Bayes Net classifier has the worst performance of the three. Similarly to with the BCN20000 dataset, the variance in scores between classifiers is greater than for the HAM10000 dataset, perhaps indicating that the data is of lower quality, and therefore more difficult for particular classifiers to perform reliably.

5.3.4 Classification Matrix

The datasets were compared against each other in attempts to understand how adaptable each dataset is for training data when tested against data sourced from different geographical locations, and from slightly different imaging methods. The results were achieved using the Random Forest classifier and SMOTE class balancing the training data, because this has previously been established as the most reliable configuration for the use case based on the results from section 4.4.4, and 5.3. Table 23 shows the accuracy, F1, MCC, and ROC Area measures for each training and test configuration.

		TEST											
		HAM		BCN		MSK		HAM+BCN		HAM+MSK		BCN+MSK	
TRAIN	HAM	accuracy:	100%	**Accuracy:**	**67.40%**	**Accuracy:**	**71.29%**	accuracy:	-	accuracy:	-	**Accuracy:**	**68.09%**
		f1:		F1:	0.654	F1:	0.708	f1:	-	f1:	-	F1:	0.663
		mcc:		MCC:	0.098	MCC:	0.069	mcc:	-	mcc:	-	MCC:	0.094
		roc area:		ROC Area:	0.606	ROC Area:	0.583	roc area:	-	roc area:	-	ROC Area:	0.601
	BCN	**Accuracy:**	**79.79%**	accuracy:	100%	**Accuracy:**	**67.15%**	accuracy:	-	**Accuracy:**	**77.10%**	accuracy:	-
		F1: r2	0.803	f1:	1	F1:	0.688	f1:	-	F1: r3	0.779	f1:	-
		MCC:	0.225	mcc:	1	MCC:	0.084	mcc:	-	MCC:	0.192	mcc:	-
		ROC Area:	0.69	roc area:	1	ROC Area:	0.548	roc area:	-	ROC Area:	0.652	roc area:	-
	MSK	**Accuracy:**	**73.61%**	**Accuracy:**	**61.89%**	accuracy:	100%	**Accuracy:**	**67.12%**	accuracy:		accuracy:	-
		F1: r4	0.761	F1:	0.625	f1:	1	F1:	0.685	f1:	-	f1:	-
		MCC:	0.173	MCC:	0.078	mcc:	1	MCC:	0.123	mcc:	-	mcc:	-
		ROC Area:	0.648	ROC Area:	0.585	roc area:	1	ROC Area:	0.617	roc area:	-	roc area:	-
	HAM+BCN	accuracy:	-	accuracy:	-	**accuracy:**	**70.81%**	accuracy:	100%	accuracy:	-	accuracy:	-
		f1:	-	f1:	-	f1:	0.711	f1:	1	f1:	-	f1:	-
		mcc:	-	mcc:	-	mcc:	0.098	mcc:	1	mcc:	-	mcc:	-
		roc area:	-	roc area:	-	roc area:	0.573	roc area:	1	roc area:	-	roc area:	-
	HAM+MSK	accuracy:	-	**Accuracy:**	**66.79%**	accuracy:	-	accuracy:	-	accuracy:	100%	accuracy:	-
		f1:	-	F1:	0.657	f1:	-	f1:	-	f1:	1	f1:	-
		mcc:	-	MCC:	0.117	mcc:	-	mcc:	-	mcc:	1	mcc:	-
		roc area:	-	ROC Area:	0.612	roc area:	-	roc area:	-	roc area:	1	roc area:	-
	BCN+MSK	**Accuracy:**	**79.49%**	accuracy:		accuracy:	-	accuracy:	-	accuracy:	-	accuracy:	100.00%
		F1: r1	0.803	f1:		f1:	-	f1:	-	f1:	-	f1:	
		MCC:	0.24	mcc:	-	mcc:	-	mcc:	-	mcc:	-	mcc:	
		ROC Area:	0.697	roc area:	-	roc area:	-	roc area:	-	roc area:	-	roc area:	

Table 23: Train / test classification matrix for different combinations of datasets

The classification matrix shows the most accurate and / or reliable results highlighted in green. The best performance was achieved using the BCN20000 and MSK datasets for training and testing on the HAM10000 dataset (r1). Despite its 79.49% accuracy being 0.3% lower than when training with the BCN20000 dataset alone, this configuration gave the highest MCC score of 0.24, indicating that this was the best setup in terms of discriminating between classes. The second best configuration was training using the BCN20000 dataset alone and testing on the HAM10000 dataset (r2), resulting in a slight decrease in scores

compared to training using the BCN and MSK datasets in combination. The third best configuration was training using the BCN20000 dataset and testing on the HAM10000 and MSK datasets together (r3), resulting in 77.1% accuracy, just 1.5% lower than the second best configuration. The remaining highlighted box (r4) shows the classification performance when training using the MSK dataset, and testing on the HAM10000 dataset. The performance of this particular configuration is surprising as the MSK dataset has the fewest total samples available for training, even after SMOTE class balancing; 2170 samples for each class totalling 4340 training samples, which is less than half the size of the HAM10000 dataset.

Overall, it would appear as though HAM10000 is the easiest dataset of the three to test against, most likely in part due to the fact that the original system was created with only the HAM10000 dataset mind. Additionally, the quality of the HAM10000 dataset was higher, in that the images included were selected by the curators according to strict requirements. For instance, only lesion images taken with a particular dermoscopic device were extracted from medical records for the dataset. Furthermore, an expert dermatologist performed histogram correction for underexposed images or images with an unbalanced hue. It is not known whether the curators of the BCN20000 or MSK datasets performed the same level of technical validation.

5.3.5 HAM10000 + BCN20000 + MSK

For completeness, the CAD system's performance was tested using all 3 datasets in combination, in addition to SMOTE class balancing, for as many training samples as possible. Again, the evaluations were made using 10-fold cross validation with the Random Forest classifier.

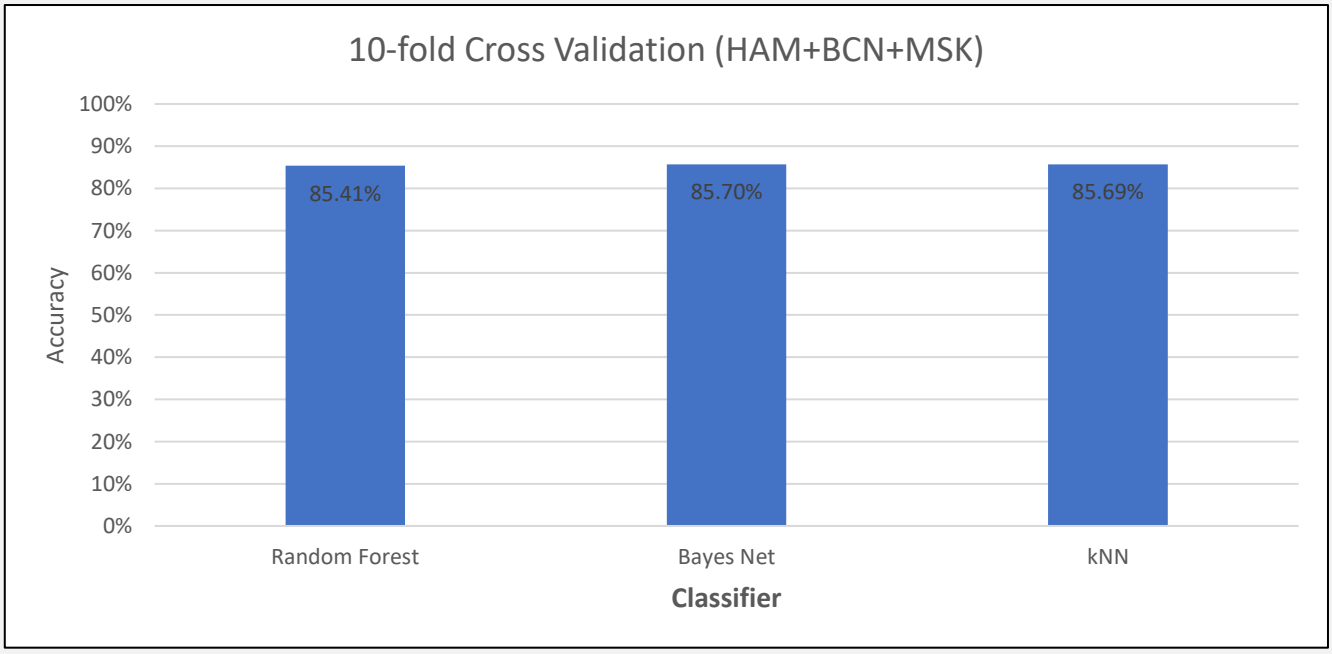


Figure 59: Classification accuracy over 10-fold cross validation for 3 classifiers using all 3 datasets combined

The average accuracy over the 10-fold cross validation shows that each classifier performs very similarly to one another, with the Random Forest classifier classifying just 0.28% less accurately than kNN. ANOVA testing revealed that there was no significant statistical difference in accuracy between the groups.

$$\text{Accuracy(Random Forest vs Bayes Net vs kNN):}\quad F(2, 27) = [1.093],\ p = 0.35 > 0.05)$$

$$F\ \text{crit} = [3.354]$$

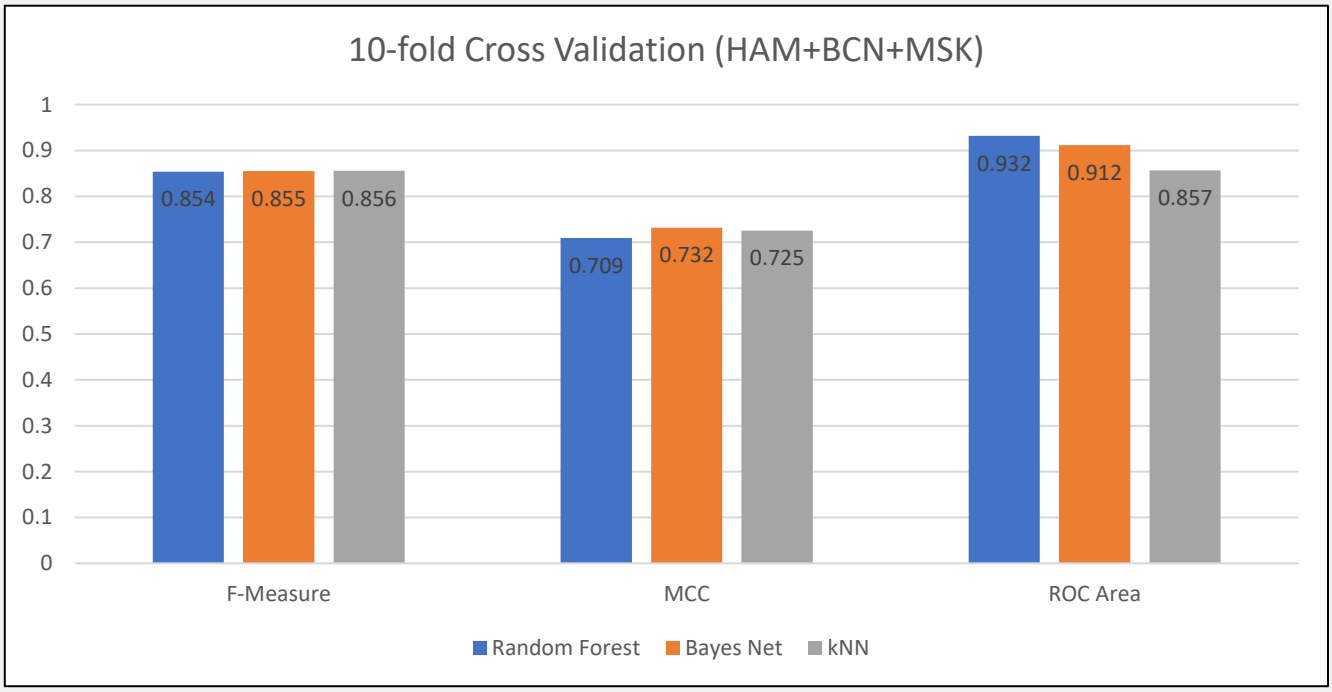


Figure 60: Classification measures F1, MCC, and ROC Area over 10-fold cross validation for 3 classifiers using all 3 datasets combined

Notably, the variance in scores between each classifier is much smaller than when cross-validating using the BCN20000 or MSK datasets alone, similar to using just the HAM10000 dataset.

Random Forest | 10xVal		
classified as ->	MALIGNANT	BENIGN
MALIGNANT	17341	2406
BENIGN	3364	16383
	MALIGNANT	BENIGN
Precision:	0.838	0.872
Recall:	0.878	0.83

Bayes Net | 10xVal		
classified as ->	MALIGNANT	BENIGN
MALIGNANT	14750	4997
BENIGN	655	19092
	MALIGNANT	BENIGN
Precision:	0.957	0.793
Recall:	0.747	0.967

kNN (1) | 10xVal		
classified as ->	MALIGNANT	BENIGN
MALIGNANT	18687	1060
BENIGN	4593	15154
	MALIGNANT	BENIGN
Precision:	0.803	0.935
Recall:	0.946	0.767

Table 24: Confusion matrices and precision and recall scores for 3 classifiers over 10-fold cross validation

The confusion matrices show that the Bayes Net classifier remains biased towards the malignant class in terms of high precision and low recall, which is not suitable for a CAD skin cancer system. On the other hands, kNN is biased towards the benign class in terms of high precision and low recall, which is preferred in this scenario over the Bayes Net solution. The Random Forest classifier is the most balanced towards the individual classes and therefore should be considered the classifier with the best overall performance with respects to the proposed CAD system and the datasets used.

HAM+BCN+MSK | Random Forest								mal	ben
								19747	19747
fold	% Accuracy	% Inaccuracy	Kappa	F-Measure	MCC	ROC Area		total:	39494
1	85.291139	14.708861	0.705823	0.85266	0.708239	0.932533			
2	85.544304	14.455696	0.710886	0.855333	0.711971	0.930216			
3	84.151899	15.848101	0.683038	0.841466	0.683492	0.922663			
4	85.924051	14.075949	0.718481	0.859154	0.719367	0.932205			
5	85.287415	14.712585	0.705751	0.85281	0.706376	0.930076			
6	85.110154	14.889846	0.702208	0.85096	0.703552	0.929195			
7	86.148392	13.851608	0.722971	0.861414	0.723707	0.941091			
8	85.414029	14.585971	0.708277	0.85406	0.709047	0.932615			
9	85.464675	14.535325	0.70929	0.854541	0.710314	0.934823			
10	85.717903	14.282097	0.714356	0.85715	0.714643	0.933016			
mean	**85.4053961**	**14.5946039**	**0.7081081**	**0.8539548**	**0.7090708**	**0.9318433**			

	TP	FP	Precision	Recall	F-Measure	MCC	ROC	PRC	Class
	0.878	0.17	0.838	0.878	0.857	0.709	0.931	0.928	MALIGNANT
	0.83	0.122	0.872	0.83	0.85	0.709	0.931	0.931	BENIGN
Weighted Avg.	**0.854**	**0.146**	**0.855**	**0.854**	**0.854**	**0.709**	**0.931**	**0.93**	

classified as ->	MALIGNANT	BENIGN
MALIGNANT	**17341**	2406
BENIGN	3364	**16383**

Table 25: Complete Random Forest classification summary for 10-fold cross validation, including per-fold key measures

5.4 ITA Skin Type

As stated in section 4.7, the datasets were split into 6 groups based on the calculated ITA of the skin area in the lesion image (Chardon et al. 1991). Based on the calculations made using the segmentation V3 algorithm to mask the skin, it was found that 99% of the images in the HAM10000 dataset, and 96% of the MSK dataset existed in the 'Light' and 'Intermediate' categories of the ITA skin classification. The BCN20000 dataset appears more balanced – 75% of the images were categorised as 'Light' or 'Intermediate'. However, the actual distribution of the BCN20000 ITA skin classification shows that the IV group – 'Tan' skin is under-represented compared to its 2 adjacent groups. This quirk of the distribution could be because of bad segmentations, and therefore it is not possible to conclude that the distribution is purely inherent to the particular dataset. Overall, and despite the calculations being only an estimation, it is easy to see from Figure 61 that all 3 datasets are not balanced across the full variety of skin tones. This is to be expected, as the geographical location that a dataset is sourced from heavily influences the distribution of skin tone, as well as the fact that risk of skin cancer due to sun damage is directly linked to the amount of melanin in an individual's skin.

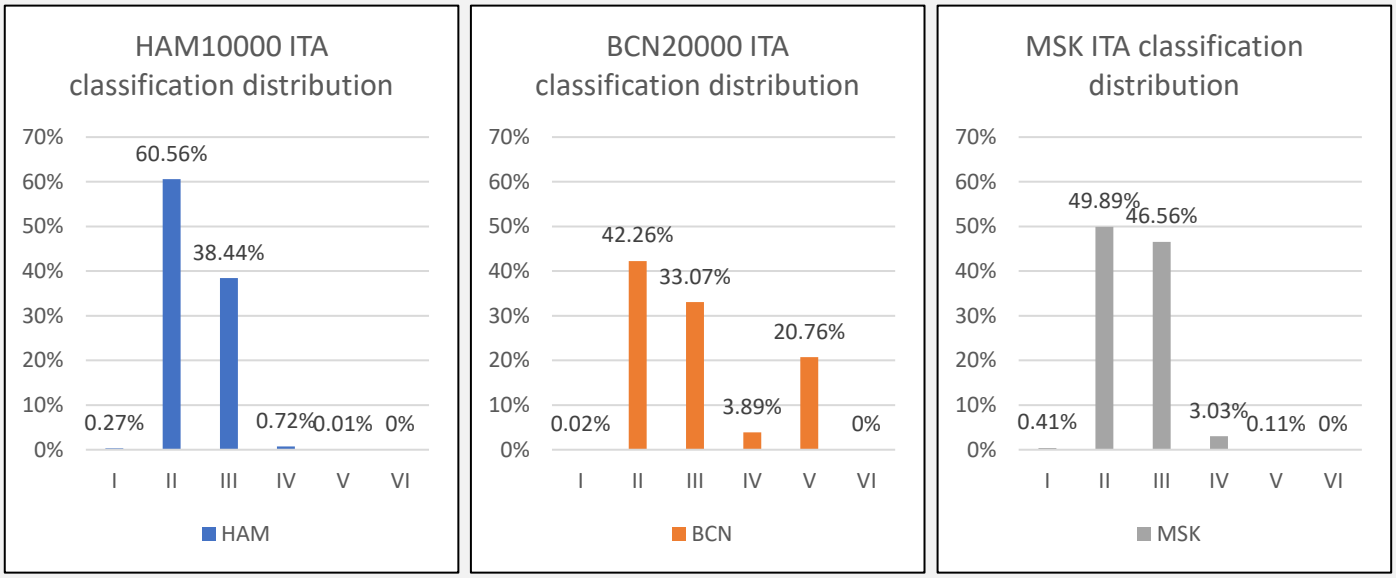


Figure 61: ITA skin classification distribution for each of the 3 datasets

The datasets were combined and the classification metrics using each individual skin group were compared against once another using Random Forest classification and 10-fold cross validation. The skin classification distribution of the combined dataset is shown in Figure 62.

ITA classification	ITA° range
Very light (I)	ITA° > 55
Light (II)	41 < ITA°< 55
Intermediate (III)	28 < ITA° < 41
Tan (IV)	10 < ITA° < 28
Brown (V)	-30 < ITA° < 10
Dark (VI)	ITA° < -30

Table 3: ITA skin classification (Chardon et al. 1991)

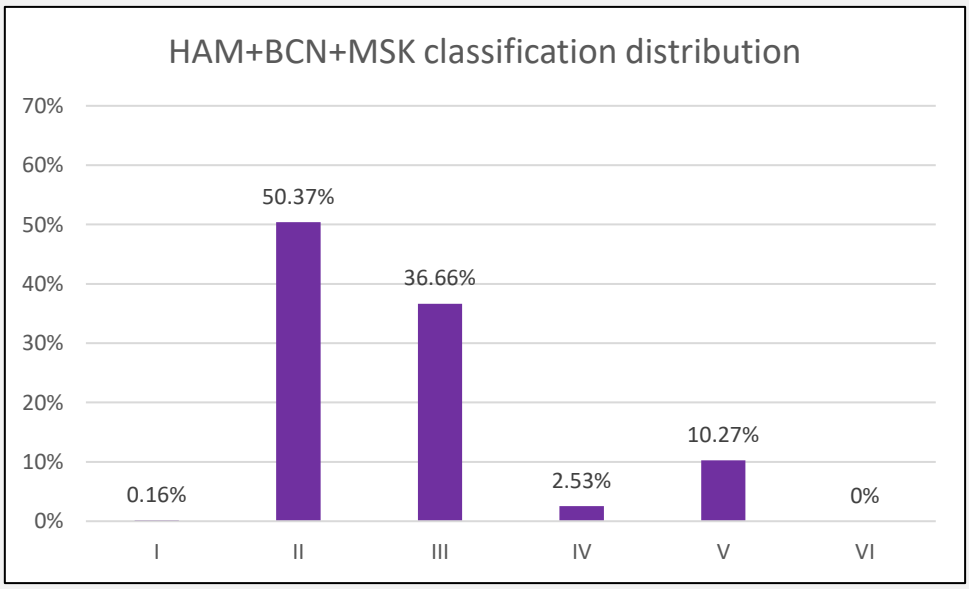


Figure 62: ITA skin classification distribution for all 3 datasets combined

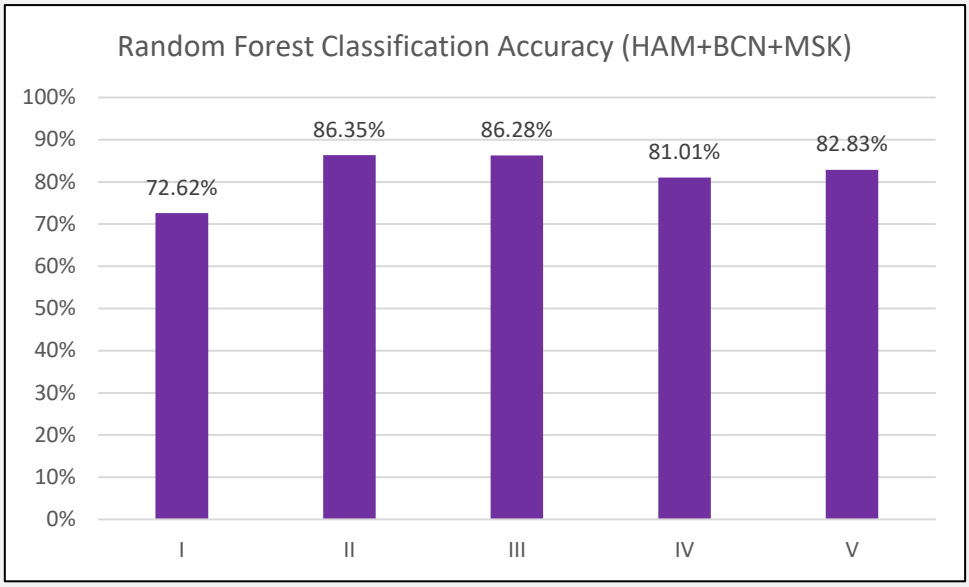


Figure 63: Classification accuracy for each ITA skin group

Over the 10-fold cross validation, the highest accuracy was achieved with the most populous ITA skin group – 86.35%, closely followed by the 2nd most populous group – 86.28%. In fact, the classification accuracy for each group correlates with the representation of each group in the dataset, this data is shown in Figure 63. Because of this correlation, it is not possible to say that the proposed CAD system treats skin type groups differently. In fact, the classification accuracy remains satisfactory even for the under-represented skin groups. For example, group I contains only 40 total instances, so for each fold of a 10-fold cross validation 36 of these instances are used for training and 4 are used as test data. ANOVA testing revealed that there was no statistical different in the accuracy between groups II and III over each fold of the 10-fold cross validation. This makes sense as the maximum accuracy achieved using all the skin groups was 85.4%, which indicates that this could be roughly the best score the CAD system can achieve under optimum conditions.

$$\text{Accuracy (II vs III):} \quad F(1, 18) = [0.037],\ p = 0.849 > 0.05), \qquad F\ \text{crit} = [4.414]$$

6. Future Work

Overall, the project achieved the primary objectives that were set out at the beginning. However, along the way there were many additional components that were thought of that were not possible to complete because they were complex, and any available time had to be wisely spent. One example of a system component that was abandoned was the hair removal evaluation tool, where hair is simulated on a lesion image with no hair, in turn creating ground truth for comparison of different hair segmentation algorithms. This component was abandoned because it was taking too much development time. In detail, the attempts at simulating hairs were not sufficient enough for a meaningful comparison for evaluation; hairs in lesion images appear in many forms, and constantly vary in size, colour, and contrast against the background. Digitally simulating these various properties of hairs in a way that accurately represents their natural distribution is nearly impossible. If more time was available, and this tool had reached completion, it would have been very helpful for quantitatively evaluating the hair removal algorithms, enabling more significant improvements to the algorithms themselves through iterative development. This could have improved the effectiveness of the overall system as a result because the success of each stage in the system is quite heavily dependent on the success of the previous stages.

In addition, there were also amendments to the feature set that would have been desirable to implement. For instance, the ABCD dermoscopy algorithm evaluates the symmetry of the lesion with respects to the distribution of the shape, as well as colour and differential structures. The CAD system only assesses symmetry in terms of the lesion's shape. Despite this simple feature performing well on its own in terms of classification accuracy (see Table 15), extending this feature subset by analysing the symmetry of colour and differential distribution could have led to significant improvements to classification accuracy, as these features were not truly analysed anywhere else in the feature set. To implement this in the CAD system would not have been difficult, but at the time, there were more important features of concern. This analysis could be done by segmenting the lesion area into 4 regions based on its primary symmetry axes, which were already computed, and comparing the results of colour and differential descriptors for each region. If more time was available, and additional features were implemented, use of SFFS and SFBS (sequential floating forward / backward selection) would have also been used to reduce the dimensionality of the feature set in hopes of better classification. Another system component that did not reach completion was using a convolutional neural network for segmentation of the hair and lesion itself. Given that the perceived weakness of the CAD system as a whole lies in the segmentation algorithms, and the algorithms already underwent several iterations resulting in only small incremental improvements, use of a neural network could have ended up improving the CAD system's final classification scores significantly. Studies have shown that the use of neural networks for this purpose can achieve classification accuracies of 94% (Liu et al. 2021). Given enough time, it would have been interesting to compare the results of the implemented computer vision segmentation algorithms against a neural network tasked to do the same thing. A neural network could also have been used to segment and remove the hair, as well as features extraction and classification.

7. Conclusions

The final CAD system, evaluated using 10-fold cross validation using the Random Forest classifier on all 3 datasets combined, achieves a respectable accuracy of 85.4%, shown in Section 5.3.5. For the HAM10000 dataset alone, which the CAD system was initially built to use, the CAD system achieved 89.56% average accuracy over a 10-fold cross validation using the Random Forest classifier, shown in Section 5.3.1. The CAD system achieved an accuracy of 83.3% using the BCN20000 dataset for evaluation, and an accuracy of 85.1% using the MSK dataset. Although the scores are not competitive with state-of-the-art studies, no neural networks were used, and the CAD system's implementation was done using only computer vision techniques and typical machine learning classifiers. It is of the opinion of the author that the system could be useful in a clinical setting to support a doctor's diagnosis, given that the images were of a similar quality to those included in the datasets used in this project, i.e. taken with similar equipment, and by the same standards.

The classification scores detailed above were obtained using the full feature set, which includes 88 features describing a lesion in terms of its asymmetry, border structure, colour, and differential structures. It was shown in Section 5.2.2 that the most useful features were the differential structure features. This shows that the GLCM texture (differential) features are good discriminators of lesion conditions in dermoscopic images. Section 5.2.2 also shows that the asymmetry and fractal dimension features were also some of the more useful features for accurate classification.

The final segmentation algorithm (V3) achieves an average Jaccard index of 57.7% against the HAM10000 curated segmentations (Tschandl 2018), which is close to being considered only half accurate. However, this metric can only be considered an estimation of the segmentation performance with respects to the system as a whole. The segmentations were curated by Tschandl. P alone through a combination of segmentation techniques and manual marking, and therefore it is impossible to say if these are truly the best segmentation for obtaining an accurate classification. It was shown through ANOVA testing that there was no statistical difference between the classification scores between the segmentation V2 and V3 algorithms, despite V2 being 13% more accurate than V3 in terms of the average Jaccard index against the curated ground truth.

Biases in datasets, and datasets that do not fully represent real-world scenarios make evaluating and comparing algorithms difficult. Skin lesion datasets are skewed towards particular skin tones, as they are sourced from particular geographical locations. Most datasets also do not fully represent the typical clinical setting – the HAM10000 dataset for instance includes very few artefacts such as clinical markings. After developing and training the system with the HAM10000, then using the system to evaluate a different dataset (BCN20000) that included more hard-to-diagnose images, it was discovered that the system performs very differently (see Section 5.3.4). It was as though the HAM10000 dataset only represented cases that were mostly easy for a computer to diagnose, which does not represent the typical clinical setting as well as the BCN20000 dataset. A framework is needed to be able to properly compare the many techniques used in CAD systems and evaluate the best methods, as well as the datasets that were used in the study. Currently, as

many works in the problem field use different, or even proprietary datasets, it is hard to recognise if any proposed system would perform well diagnosing lesions from datasets sourced from different geographical locations.

The requirements set out in Section 3.1 were satisfactorily met, as the final system is functional from beginning to end. The details of each requirement's implementation are shown in Table 26.

The system should be capable of:	Requirement met?	In which class?	Notes
Acquiring lesion images from a dataset	Yes	FileListProvider.java	Section 4.0
Performing black border and vignette removal on lesion images	Yes	BlackBordersImageProcessor.java	Section 4.5.1
Performing hair removal on lesion images	Yes	HairMaskImageProcessor.java	Section 4.1
Segmenting ROIs in a lesion image	Yes	SegmentationImageProcessorV3.java	Section 4.2
Evaluating segmented ROIs against available ground truth	Yes	SegmentationComparison.java	Section 5.1 - 57.7% accuracy
Extracting features from a ROI	Yes	FeatureProcessor.java	Section 4.3 - 88 features extracted
Addition / removal of features as needed	Yes	FeatureFactory.java	
Storing training data in a file	Yes	arffWriter.java	
Training a classifier using extracted features	Yes	MachineLearningController.java	Section 4.4
Classifying an unseen skin lesion using a training classifier	Yes	MachineLearningController.java	Section 5.3.4
Evaluating the trained classifier	Yes	MachineLearningController.java	Section 5.3 - with 85% accuracy
Processing more than 20,000 images efficiently	Yes	Coordinator.java	~4 seconds per image

Table 26: System requirements

8. Reflection on Learning

Through the process of undertaking a project of this scale, I learnt a lot about what it takes to produce an effective end-to-end computer-aided diagnosis system. Typically this would be taken on by a team of developers, so I am proud to have achieved a working CAD system, and I will be attempting to purchase a dermatoscope smart phone attachment so I can diagnose my friends' and family's concerning skin lesions. This project had me working consistently near-daily to keep up-to-schedule and allow myself enough time for writing the report. The process taught me how to balance the workload in terms of not spending too long on a single problem; many times I found myself up late at night working on an algorithm to little or no avail, and had to cut my losses and go to sleep, and move on to the next problem the following day. By delving deep into this particular problem field, I found that it was much easier to enjoy the work, particularly for a project that has very practicable and potentially life-saving applications. This aspect of the project meant that I also learnt a lot about skin cancer, its risks, and how machine learning can actually help to save lives. Thoroughly reading many technical articles from a variety of medical journals as a vital part of the project directly improved my understanding of computer vision, artificial intelligence, and biomedical topics. By the end of the project, I ended up finding some small amount of entertainment in reading research articles, as I developed an understanding of the problem field that extended more than surface-deep. Not only this, but reading research papers helped me to refine my report writing style, as I found myself naturally trying to use similar styles of writing as I was reading in the articles. The scale of the report also helped to improve my writing skill, as I had to plan my writing meticulously based on the available time, and I had to be selective about the content I was writing in order to not breach the word count.

Table of Figures

References

Argenziano, G., Fabbrocini, G., Carli, P., De Giorgi, V., Sammarco, E. and Delfino, M. 1998. Epiluminescence Microscopy for the Diagnosis of Doubtful Melanocytic Skin Lesions: Comparison of the ABCD Rule of Dermatoscopy and a New 7-Point Checklist Based on Pattern Analysis. *Archives of Dermatology* 134(12), pp.1563–1570. doi:10.1001/archderm.134.12.1563.

Barata, C., Celebi, M.E. and Marques, J.S. 2015. Melanoma detection algorithm based on feature fusion. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* 2015, pp. 2653-2656. doi:10.1109/embc.2015.7318937.

Bissoto, A., Fornaciali, M., Valle, E. and Avila, S. 2019. (De)Constructing bias on skin lesion datasets. I*n Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 0-0). doi:10.48550/arXiv.1904.08818.

BMJ. 2021. *Seborrhoeic keratosis - Symptoms, diagnosis and treatment | BMJ Best Practice*. Available at: https://bestpractice.bmj.com/topics/en-gb/617 [Accessed: 21 April 2022].

Cancer Research UK. 2015. *Melanoma Skin Cancer Statistics*. Available at: https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/melanoma-skin-cancer [Accessed: 21 April 2022].

Cancer Research UK. 2018. *Non-melanoma skin cancer statistics*. Available at: https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/non-melanoma-skin-cancer [Accessed: 21 April 2022].

Canny, J. 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8(6), pp.679–698. doi:10.1109/tpami.1986.4767851.

Cavalcanti, P.G. and Scharcanski, J. 2011. Automated prescreening of pigmented skin lesions using standard cameras. *Computerized Medical Imaging and Graphics* 35(6), pp.481–491. doi:10.1016/j.compmedimag.2011.02.007.

Cavalcanti, P.G. and Scharcanski, J. 2013a. Macroscopic Pigmented Skin Lesion Segmentation and Its Influence on the Lesion Classification and Diagnosis. In: Celebi, M.E., Schaefer, G. eds. *Color Medical Image Analysis, Lecture Notes in Computational Vision and Biomechanics* vol.6, 2013. Netherlands: Springer, pp.15-39.

Cavalcanti, P.G., Scharcanski, J. and Baranoski, G.V.G. 2013b. A two-stage approach for discriminating melanocytic skin lesions using standard cameras. *Expert Systems with Applications* 40(10), pp.4054–4064. doi:10.1016/j.eswa.2013.01.002.

Chardon, A., Cretois, I. and Houseau, C. 1991. Skin colour typology and suntanning pathways. *International Journal of Cosmetic Science* 13(4), pp.191–208. doi:10.1111/j.1467-2494.1991.tb00561.x.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16(16), pp.321–357. doi:10.1613/jair.953.

Codella, N.C.F., Gutman, D., Celebi, M.E., Helba, B., Marchetti, M.A., Dusza, S.W., Kalloo, A., Liopyris, K., Mishra, N., Kittler, H. and Halpern, A. 2018. Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC). *2018 IEEE 15th international symposium on biomedical imaging* ISBI 2018, pp. 168-172. doi:10.48550/arXiv.1710.05006.

Combalia, M., Codella, N.C.F., Rotemberg, V., Helba, B., Vilaplana, V., Reiter, O., Carrera, C., Barreiro, A., Halpern, A.C., Puig, S. and Malvehy, J. 2019. BCN20000: Dermoscopic Lesions in the Wild. *arXiv preprint arXiv:1908.02288*. doi:10.48550/arXiv.1908.02288.

DermNet NZ. 2004. *Dermoscopy.* Available at: https://dermnetnz.org/topics/dermoscopy [Accessed 21 April 2022].

DermNet NZ. 2008. *Examination of the skin*. Available at: https://dermnetnz.org/cme/principles/examination-of-the-skin [Accessed: 21 April 2022].

Dick, V., Sinz, C., Mittlböck, M., Kittler, H. and Tschandl, P. 2019. Accuracy of Computer-Aided Diagnosis of Melanoma. *JAMA Dermatology* 2019;155(11):1291-1299. doi:10.1001/jamadermatol.2019.1375.

Ganster, H., Pinz, P., Rohrer, R., Wildling, E., Binder, M. and Kittler, H. 2001. Automated melanoma recognition. *IEEE Transactions on Medical Imaging* 20(3), pp.233–239. doi:10.1109/42.918473.

Henning, J.S., Dusza, S.W., Wang, S.Q., Marghoob, A.A., Rabinovitz, H.S., Polsky, D. and Kopf, A.W. 2007. The CASH (color, architecture, symmetry, and homogeneity) algorithm for dermoscopy. *Journal of the American Academy of Dermatology* 56(1), pp.45–52. doi:10.1016/j.jaad.2006.09.003.

ImageJ Wiki. 2022. Introduction. Available at: https://imagej.net/learn/ [Accessed: 23 April 2022].

ISIC. 2022. *ISIC Archive | About ISIC*. Available at: https://www.isic-archive.com/# [Accessed: 22 April 2022].

JavaML. 2022. Java Machine Learning Library (Java-ML). Available at: http://java-ml.sourceforge.net/ [Accessed: 23 April 2022].

Kassem, M.A., Hosny, K.M., Damaševičius, R. and Eltoukhy, M.M. 2021. Machine Learning and Deep Learning Methods for Skin Lesion Classification and Diagnosis: A Systematic Review. *Diagnostics* 11(8), 1390. doi:10.3390/diagnostics11081390.

Lee, T., Ng, V., Gallagher, R., Coldman, A. and McLean, D. 1997. Dullrazor®: A software approach to hair removal from images. *Computers in Biology and Medicine* 27(6), pp.533–543. doi:10.1016/s0010-4825(97)00020-6.

Lio, P.A. and Nghiem, P. 2004. Interactive Atlas of Dermoscopy. Journal of the American Academy of Dermatology, 50(5), pp.807–808. doi:10.1016/j.jaad.2003.07.029.

Liu, L., Tsui, Y.Y. and Mandal, M. 2021. Skin Lesion Segmentation Using Deep Learning with Auxiliary Task. *Journal of Imaging* 7(4), p.67. doi:10.3390/jimaging7040067.

Liu, X., Faes, L., Kale, A.U., Wagner, S.K., Fu, D.J., Bruynseels, A., Mahendiran, T., Moraes, G., Shamdas, M., Kern, C., Ledsam, J.R., Schmid, M.K., Balaskas, K., Topol, E.J., Bachmann, L.M., Keane, P.A. and Denniston, A.K. 2019. A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. *The Lancet Digital Health* 1(6), pp.e271–e297. doi:10.1016/s2589-7500(19)30123-2.

Maglogiannis, I. and Delibasis, K. 2015b. Hair removal on dermoscopy images. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* 2015:2960-2963. doi:10.1109/embc.2015.7319013.

Maglogiannis, I. and Delibasis, K.K. 2015a. Enhancing classification accuracy utilizing globules and dots features in digital dermoscopy. *Computer Methods and Programs in Biomedicine* 118(2), pp.124–133. doi:10.1016/j.cmpb.2014.12.001.

MathWorks 2022. MATLAB - MathWorks. Available at: https://ch.mathworks.com/products/matlab.html [Accessed: 23 April 2022].

Mendonca, T., Ferreira, P.M., Marques, J.S., Marcal, A.R.S. and Rozeira, J. 2013. PH2 - A dermoscopic image database for research and benchmarking. *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* 2013, pp. 5437-5440. doi:10.1109/embc.2013.6610779.

Ming-Kuei, Hu . 1962. Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory* 8(2), pp.179–187. doi:10.1109/tit.1962.1057692.

Nachbar, F., Stolz, W., Merkle, T., Cognetta, A.B., Vogt, T., Landthaler, M., Bilek, P., Braun-Falco, O. and Plewig, G. 1994. The ABCD rule of dermatoscopy. *Journal of the American Academy of Dermatology* 30(4), pp.551–559. doi:10.1016/s0190-9622(94)70061-3.

Oakden-Rayner, L. 2019. The Rebirth of CAD: How Is Modern AI Different from the CAD We Know? *Radiology: Artificial Intelligence* 1(3), p.e180089. doi:10.1148/ryai.2019180089.

OpenCV. 2019. OpenCV library. Available at: https://opencv.org/ [Accessed: 23 April 2022].

Osto, M., Hamzavi, I.H., Lim, H.W. and Kohli, I. (2021). Individual Typology Angle and Fitzpatrick Skin Phototypes are Not Equivalent in Photodermatology. *Photochemistry and Photobiology* 2022 98(1), pp.127-129. doi:10.1111/php.13562.

Otsu's method. 2022. *Wikipedia*. Available at: https://en.wikipedia.org/wiki/Otsu%27s_method [Accessed: 24 April 2022].

Ralph Braun. 2017. *ABCD asymetry schematic*. Available at: https://dermoscopedia.org/File:ABCD_asymetry_schematic.jpg [Accessed: 22 April 2022].

RGB color spaces. 2022. Wikipedia. Available at: https://en.wikipedia.org/wiki/RGB_color_spaces [Accessed 25 April 2022].

Rosendahl, C., Cameron, A., McColl, I., Wilkinson, D. 2012. Dermatoscopy in routine practice - 'chaos and clues'. *Aust Fam Physician* 2012;41:482-7. PMID: 22762066.

Schaefer, G., Rajab, M.I., Emre Celebi, M. and Iyatomi, H. 2011. Colour and contrast enhancement for improved skin lesion segmentation. *Computerized Medical Imaging and Graphics* 35(2), pp.99–104. doi:10.1016/j.compmedimag.2010.08.004.

SkinVision. 2018. *SkinVision | Skin Cancer Melanoma Detection App | Check Your Skin*. Available at: https://www.skinvision.com/ [Accessed: 21 April 2022].

Sobel operator. 2022. Wikipedia. Available at: https://en.wikipedia.org/wiki/Otsu%27s_method [Accessed: 24 April 2022].

Spot Check. 2022. *Online consulting: send a photo of your spot to a skin cancer doctor | Spot Check Clinic*. Available at: https://www.spotcheck.clinic/online-consulting [Accessed: 21 April 2022].

Telea, A. 2004. An Image Inpainting Technique Based on the Fast Marching Method. *Journal of Graphics Tools* 9(1), pp.23–34. doi:10.1080/10867651.2004.10487596.

Topol, E.J. 2019. High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine* 25(1), pp.44–56. doi:10.1038/s41591-018-0300-7.

Tschandl, P., Rosendahl, C. and Kittler, H. 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data* 5(1). doi:10.1038/sdata.2018.161.

Varoquaux, G. and Cheplygina, V. 2022. Machine learning for medical imaging: methodological failures and recommendations for the future. *npj Digital Medicine* 5(1), pp.1–8. doi:10.1038/s41746-022-00592-y.

Weka. 2019. Weka 3 - Data Mining with Open Source Machine Learning Software in Java. Available at: https://www.cs.waikato.ac.nz/ml/weka/ [Accessed: 23 April 2022].

Wen, D., Khan, S.M., Xu, A.J., Ibrahim, H., Smith, L., Caballero, J., Zepeda, L., Perez, C. de B., Denniston, A.K., Liu, X. and Matin, R.N. 2021. Characteristics of publicly available skin cancer image datasets: a systematic review. *The Lancet Digital Health* 0(0). doi:10.1016/S2589-7500(21)00252-1.

World Health Organization. 2022. *Third round of the global pulse survey on continuity of essential health services during the COVID-19 pandemic. Interim report.* Available at: https://www.who.int/publications/i/item/WHO-2019-nCoV-EHS_continuity-survey-2022.1 [Accessed 21 April 2022].

Žunić, J., Hirota, K. and Rosin, P.L. 2010. A Hu moment invariant as a shape circularity measure. *Pattern Recognition* 43(1), pp.47–57. doi:10.1016/j.patcog.2009.06.017.