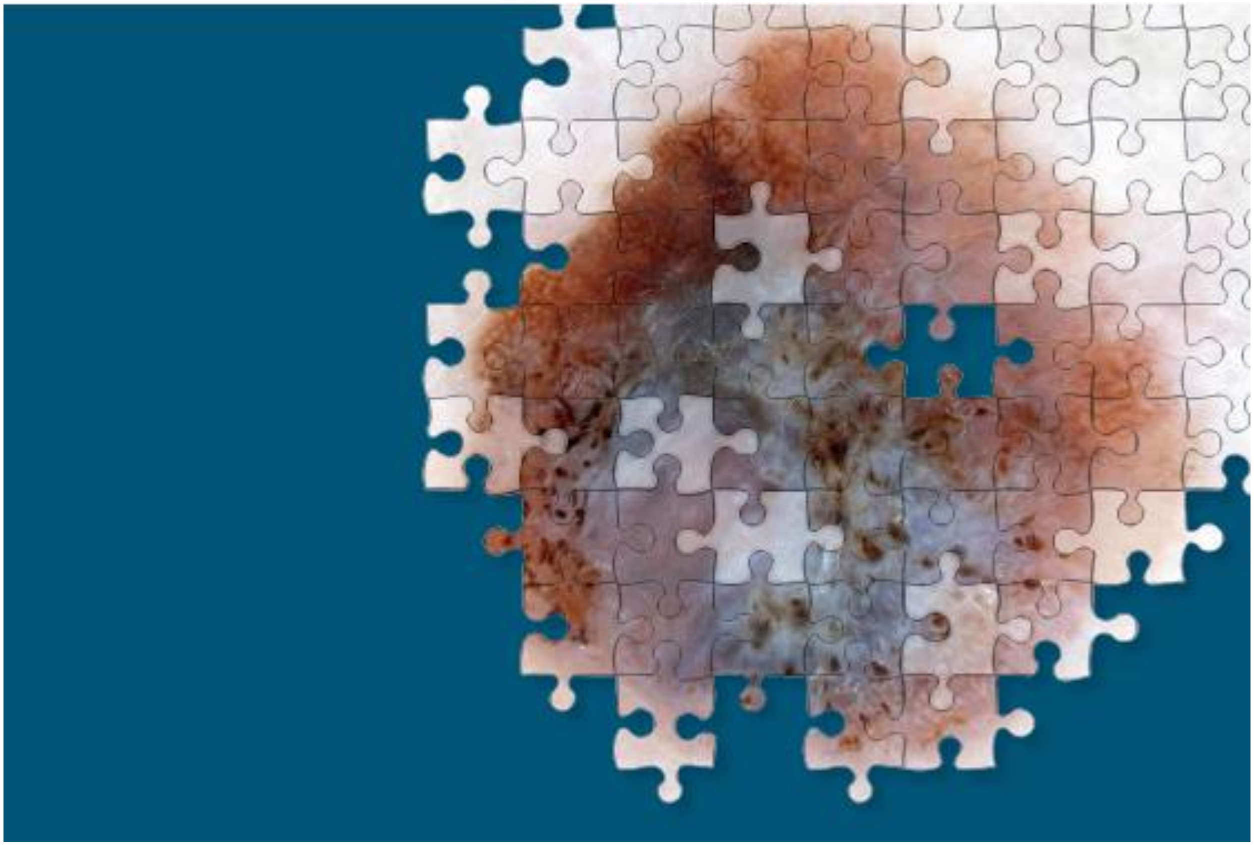
Dermoscopy Image Analysis for Melanoma Classification



DATA7001 – Final Project Report
Master of Data science
The University of Queensland

Ali Shokoohmand (s4596875)
Robert Williamson (s4575121)
Kartik Gupta (s4554135)
Reia Natu (s4563411)

Table of contents

1 Executive Summary

Australia has one of the highest rates of melanoma in the world. It is estimated in 2019 around 16.000 of new cases of melanoma skin cancer will be diagnosed in Australia. Although diagnostic methods have improved over the past few years, detection of melanoma skin lesions at the early stages of the disease is one the major challenges. In the past few yeas dermoscopy, a high resolution imaging method has improved the accuracy of the melanoma detection by skin clinicians and dermatologists, however lack of experience and the complexity of suspicious skin lesion features could hinder the diagnosis and thereby result in inaccurate detection. This in turn delays the treatment process and directly affects the patient outcomes.

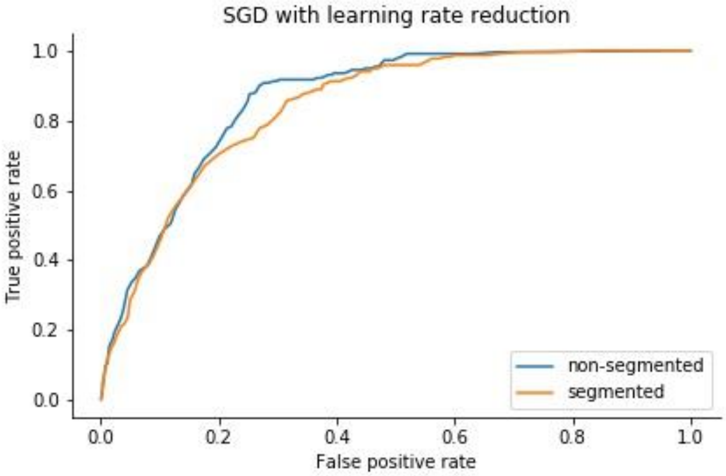
We decided to develop a computer-aided platform that is capable of identifying melanoma from non-melanoma skin lesions with an accuracy higher than skin clinicians and dermatologists when dermoscopy imaging is used. This will provide not only a cheap and fast diagnostic tool for diagnosis of melanoma, but also improve the dignostic accuract by doctors. Moreover, this diagnostic tool could potentially be developed to an mobile application that can be used by individuals for examination of suspicious skin lesions shortly after their appearance on the skin.

Outcome

We were able to classify melanoma from non-melanoma with a sensitivity of 91.2% although our precision was low at 27% due to a high rate of false positives. With such a low precision this classifier would only be suitable as an initial indicator of melanoma. With a positive diagnosis requiring expert clinical follow-up. Visual inspection of the false positive images did not provide insight as to why these cases were mis-classified.

Sensitivity	Precision	Specificity	AUC
91.2%	27.0%	70.2%	0.86

Segmentation of the skin lesion decreased classification sensitivity to 88.9%, which indicates that background image features such as skin pigment may provide important information for diagnosis. The figure below compares the receiver operator curves (ROCs) for both the segmented and non-segmented images. The classifier trained and tested with non-segmented images performed 5% better than with the segmented images with an area under the ROC of 0.87 compared to 0.83.



2 Introduction

Melanoma is a type of skin cancer which usually occurs on the parts of the body that have been overexposed to the sun. It is the third most commonly diagnosed cancer in Australia [1]. Australia and New Zealand stand for the highest incidence rate of melanoma worldwide.

Traditionally an invasive biopsy procedure is undertaken, where all or part of the suspicious mole or growth is removed to accurately diagnose melanoma cases by pathological examinations [2]. In recent years, dermoscopy has been adopted and is a non-invasive high resolution (up to 400x) imaging method more widely is used for examinations of skin lesions and detection of melanoma. The use of dermoscopy has shown to improve the ability of primary care physicians to triage lesions suggestive of skin cancer without increasing the number of unnecessary expert consultations [3].

According to meta-analysis of studies performed in clinical setting dermoscopy is more accurate than naked eye examination for the diagnosis of cutaneous melanoma in suspicious skin lesions with the relative diagnostic odds ratios for dermoscopy compared with naked eye examination, to be 15·6 [95% confidence interval (CI) 2·9–83·7, P = 0·016] [4]. Indeed, the accuracy of melanoma detection was increased up to 20% in the case of sensitivity and up to 10% in the case of specificity compared with naked eye examination [5].

In the past few years, computerised methods have been developed to accurately classify dermoscopic images for detection of melanoma. These methods were developed to increase the diagnostic specificity and thereby reducing the frequency of unnecessary surgical excisions. In particular, machine learning methods have provided a promising diagnostic tool with a higher sensitivity and specificity compared to experienced skin clinicians for melanoma detection from dermoscopic images [6]. The areas of digital image processing application increasingly require methods capable of capturing and highlighting the information contained in the image for human interpretation and analysis. Hence, in this project we aimed to develop a computer-aided machine learning platform that combines the most optimal algorithms to capture the key information contained in dermoscopy images for classification of melanoma from non-melanoma cases with high sensitivity better than those provided by experienced skin clinicians.

2.2 Empathy in design

The occurrence rates of melanoma are rising rapidly, which are resulting in higher death rates. However, if the melanoma is diagnosed in Phase I (early stages of the disease), the survival rates will be improved significantly. Dermoscopy has been shown to be a useful and fairly inexpensive tool for melanoma detection. This technique has shown to improve dermatologists and skin clinicians' accuracy to detect melanoma cases while providing a faster and more efficient approach for early diagnosis [8]. Although, as it comes to a human judgment, lack of training, experience and the complexity of skin lesion features could lead to inaccurate detections [9]. We believe an image classifier capable of identifying dermoscopic images of melanoma from non-melanoma more accurately will improve early detection of melanoma and ultimately reducing its mortality rate.

More importantly, such image classifier will not only provide a cheap, easy to use and noninvasive diagnostic tool to doctors and help them to quickly identify high priority patients, but also offers a potential online tool to individuals to check for any suspicious skin lesions that appears on their body. This platform could potentially be developed as a mobile application for analysis of high resolution images captured by modern mobile phone camera technologies. To ensure privacy, the images must be pre-processed and analysed locally and never be uploaded to an external server. We believe the privacy is especially important when it comes to medical data.

2.3 Convolutional Neural Networks

Convolutional neural networks (CNN) are a special architecture of artificial neural networks that uses some features of the brain visual cortex. CNN is one the most popular architecture for image classification and has shown to have the capacity of classifying pigmented skin lesions with a level of competence comparable to expert dermatologists [7].

For this study we used two separate CNNs for segmentation and classification of pigmented skin lesions.

2.4 Segmentation

Segmentation is an essential task in image classification. The objective of segmentation is to isolate an object of interest from the image background. For pigmented skin lesion classification, segmentation is used to separate the lesion from the surrounding skin. Providing only the lesion to the classification model. This assumes that other information in the background image is irrelevant for diagnosis, which may not be the case. To test the effectiveness of segmentation for the classification task each image is classified using both the segmented and non-segmented images.

2.5 Classification

Being novice to CNNs, our knowledge of how to build and train a good classifier was limited, so we chose a small number of hyperparameters to vary to improve our outcome. These are included in Table 1.

Table 1: Hyperparameters used to improve classification

Hyper-parameter	Description
Batch size	This is the number of training samples used in an iteration of the training algorithm. For efficient use of computer memory it is best to choose a power of 2. E.g.: 2, 8, 16, 32, …, 2^n
Epochs	An epoch is a complete iteration over the whole training set.
Loss function	Measure of how close the model is to the objective outcome. The goal is to minimise this function by adjusting weights associated with each neuron, therefore "training" the model to achieve the desired outcome. For this study we primarily used the binary cross entropy loss function. For segmentation, a Jaccard component is incorporated into the loss function.
Optimizer	Algorithm that adjusts weights in the model with the goal of minimizing the loss function.
Learning rate reduction	Optimizers uses a learning rate which is how large a leap the optimization function takes to find the minimum. Generally, it is good to use a slightly larger learning rate at the start of training to not find a local minima and then slowly reduce this as the loss parameter starts to plateau.

2.6 Metrics

For classification we used sensitivity, specificity, precision and area under the receiver operating curve (AUC) to monitor model training and to assess overall performance.

$$Sensitivity = True\ Positive\ /\ (True\ Positive + False\ Negative)$$

We focused on sensitivity, the true positive rate. This includes false negatives in the denominator. False negatives are missed cases of melanoma which would be catastrophic in real terms. Therefore, it is important to minimize their occurrence.

2.7 Report structure

This report contains the following sections

Hyperparameter	Description
Models	Description of the models used, their structures and how they work
Data Science Approach	Getting the data for training and testing our models Making the data fit for use Making the data confess (results and discussion)
Future Developments	Recommendations on how improve the outcome based on our learnings
References	References
Appendices	Our response to the peer review File register – includes all files included in the submission with a short description for each

3 Models

Convolutional Neural Networks (CNN) are deep artificial neural networks primarily used for image classification. They cluster by similarity and perform object recognition. The basic CNN Architecture begins with feature extraction (alternating convolution and subsampling layers) and ends with classification through dense and final sigmoid layer.

3.1 Segmentation model

State of the art approaches generally use CNN to perform segmentation. [13, 14] use UNet or a variation of UNet. [11] provides pre-built segmentation models including UNet with pre-trained weights for fast training based on new datasets.

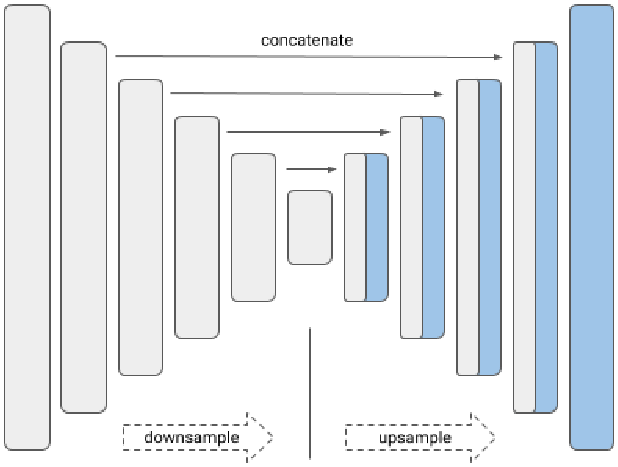


Figure 1: Schematic diagram of UNet architecture

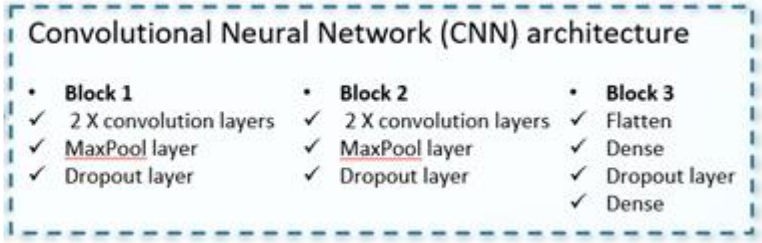
UNet (Figure 1) has two main components, a downsampling side that extracts features from an image and an upsampling side that maps the extracted feature space to the image dimensions. The final layer applies a probability using the sigmoid function to each pixel location of being on the object of interest or not.

Pixels with a probability of 0.5 or greater are predicted as being part of the pigmented skin lesion and the rest is background. This rule is used to create masks as an array with dimensions equal to the image, where the background value is 0 and the foreground (predicted skin lesion) is 1. The masks are then used as a logical filter to set the background value of the images to 0.

The Jaccard index or intersection over union (IOU) is the metric used to assess the performance of the segmentation model. This provides an objective measure of how well our segmentation model performs on the training, validation and test datasets compared to their labels.

3.2 Classification model

The model design for classification is divided into 3 sequential blocks and explained in terms of functionality of each layer as per Figure 2.



Convolutional Neural Network (CNN) architecture

- **Block 1**

✓ 2 X convolution layers
✓ MaxPool layer
✓ Dropout layer

- **Block 2**

✓ 2 X convolution layers
✓ MaxPool layer
✓ Dropout layer

- **Block 3**

✓ Flatten
✓ Dense
✓ Dropout layer
✓ Dense

Figure 2: Classification model architecture

3.2.1 Blocks 1 and 2

Convolution Layer

This layer has a special type of filter having the ability to recognise patterns in an image. The first one considers input shape as an input array. It contains hyperparameters like stride, padding, kernal size and input neuron size.

These are defined as follows:

- stride - steps of running filters in terms of pixels
- padding - assists the decision of padding the new layer that is generated. If padding is valid the size of the new layer is reduced by the kernel size.
- kernel size - the size of the convolution layer
- kernals – or filters are the number of desired feature maps

The convolution layer has an activation function called 'ReLu' that introduces non-linearity to a system that has just been computing linear operations during the convolution layers.

Pooling Layer

The second important layer is the pooling (MaxPool2D) layer. It acts as a downsampling filter. It considers neighbouring pixels and picks the maximal value. It's aim is to reduce the size of the image so that it can be easily replaced with a convolution layer with the same stride as the corresponding pooling layer.

It is used to reduce computational cost as well as the problem of overfitting.

A combination of these first 2 layers combine local features and learn more global features of the image.

3.2.2 BLOCK 3

Dropout Layer

Dropout acts as a regularization method. It randomly ignores a proportion of nodes by setting their weights to zero for each training sample. It helps provide right classification. For example, even if some of the activations are dropped out, it ensures that the network isn't getting too fitted to the training data. Besides, this layer is used only during training and not during testing.

Flatten

This layer has an activation function of sigmoid (specially for binary classification).

Dense

A dense layer thus is used to change the dimensions of the feature vector. Mathematically speaking, it applies a rotation, scaling, translation transform the feature vector.

4 Data science approach

This section follows the data science process and details our approach from obtaining our datasets to training and testing our models.

4.1 Getting the data

Two datasets were required for this study. The first for training and testing our segmentation model and the second for training and testing our classification model.

Image data consists of pixels with intensity values for red, green and blue (RGB). The RGB values range from 0 to 255.

For computational purposes images are effectively 3-dimensional arrays, where the first two dimensions are the coordinates of each pixel and the third dimension contains the colour channels.

4.1.1 Segmentation data

Segmentation data was obtained from [12] and contains 2,594 dermoscopic skin lesion images and their corresponding masks as ground truth, where the masks represent the lesion boundary. Ground truth masks were generated by either human experts or computer aided processes approved by human experts [12].

4.1.2 Classification data

This is a separate large collection of multi-source dermoscopic pigmented images, including 10,015 melanoma and non-melanoma images obtained from Kaggle [13].

Along with the images is a csv containing the metadata for each image including:

- lesion_id – unique identifier of each lesion
- image_id – unique identifier of each image
- dx – class of lesion
- dx_type – stage of diagnostic process
- age of the patient
- sex of the patient
- localization – where the lesion occurred on the patient.

The lesions are labelled into 7 different classes including both melanoma and non-melanoma cancerous types and benign types. We chose to simplify the study into a binary classification problem and re-classify the images more generally as either melanoma or not.

4.2 Making the data fit for use

Image data needs to be converted into a set of training, validation and testing arrays for our classification model. The classification object is usually segmented from the background of an image.

For the segmentation and classification training algorithms to work the pixel values also have to be normalised to the order of 10^{-1}.

4.2.1 Segmentation images

Images and masks are provided as JPGs and PNGs in a range of sizes and resolutions.

To convert the images into a valid set of features and labels for training and testing the segmentation model, the images must have square dimensions and be concatenated into arrays as in Figure 3.

Finally, the array values were normalised for the optimization function to work effectively.

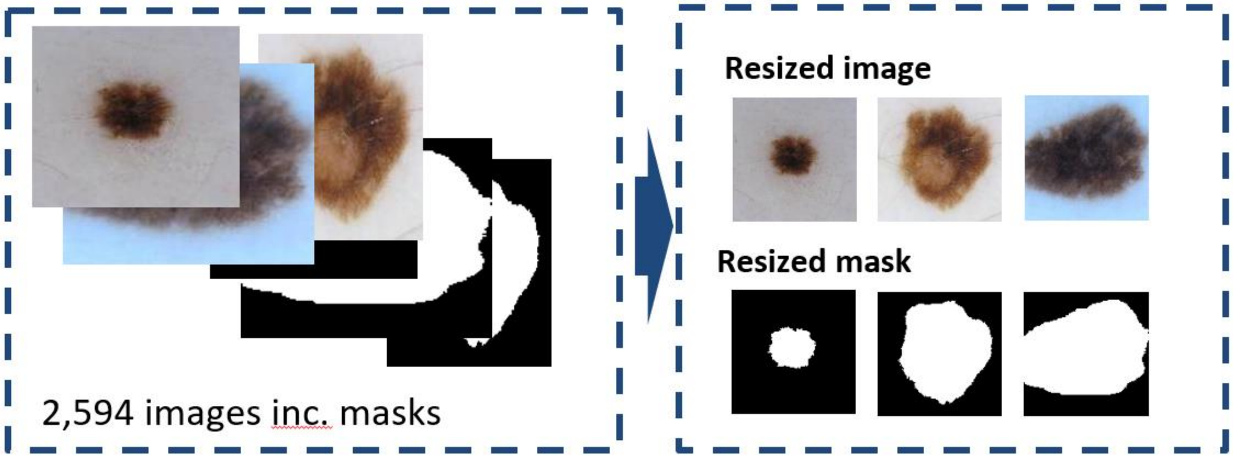


Figure 3: Rezising the images

4.2.2 Classification images

The resolution of these images is 600x450 which is far too big for training the classification model due to limited computing processing resources.

Images were cropped and resized down to 128x128 pixels to enable training of the model without reaching the computing resource limit and crashing the kernel.

The resized images were then concatenated into a single 4-dimensional array and the values were normalised to between +/- 0.5.

The labels that are provided in the metadata file were converted into a binary target vector of non-melanoma (0) and melanoma (1).

Finally, the feature array and target vectors were split into training (60%), validation (20%) and test sets (20%).

Segmentation

We used the segmentation model to segment the classification dataset to see if this would improve our results.

Class imbalance

Reclassifying our images as melanoma and non-melanoma resulted in a large class imbalance.

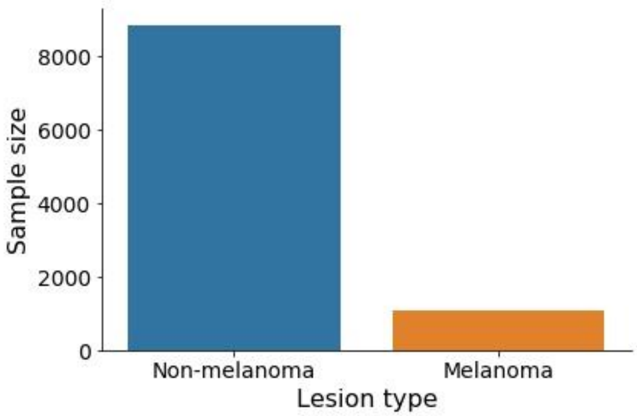


Figure 4. Class sample sizes

Initial testing resulted in poor training performance due to the imbalance in the training set. To counter this weighting was in the loss function to "focus" on the melanoma and improve the results. This is provided as a built-in feature in the Keras Model class.

The training set contained 5,335 non-melanoma and 674 melanoma images. Divided the non-melanoma sample size by the melanoma sample size gives a class weight of 7.92.

4.3 Making the data confess

Table 2 summarises scenarios tested to train the classification model.

Table 2: Modelling scenarios

Scenario	Inputs	Optimizer	Learning rate	Learning rate reduction	Batch size	Epochs
1	Segmented images	SGD	0.001	No	16	10
2		SGD		Yes		
3		Adam		No		
4		Adam		Yes		
5	Non-segmented	SGD		No		
6		SGD		Yes		
7		Adam		No		
8		Adam		Yes		

4.3.1 Segmentation results

The segmentation model achieved an overall Jaccard index of 82% on the test set. Figure 5 shows good and bad examples taken from 10 randomly sampled images with a Jaccard of 94% and 65% respectively. A Jaccard of 82% was achieved overall on the test set.

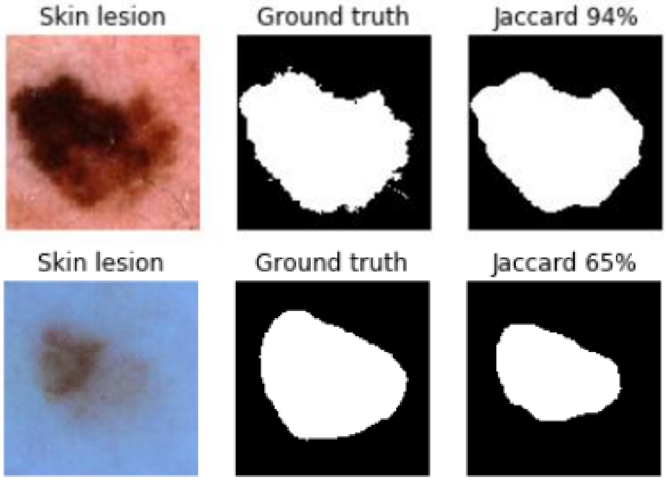


Figure 5: good and bad examples of segmentation results

The segmentation model was then used to segment the classification dataset.

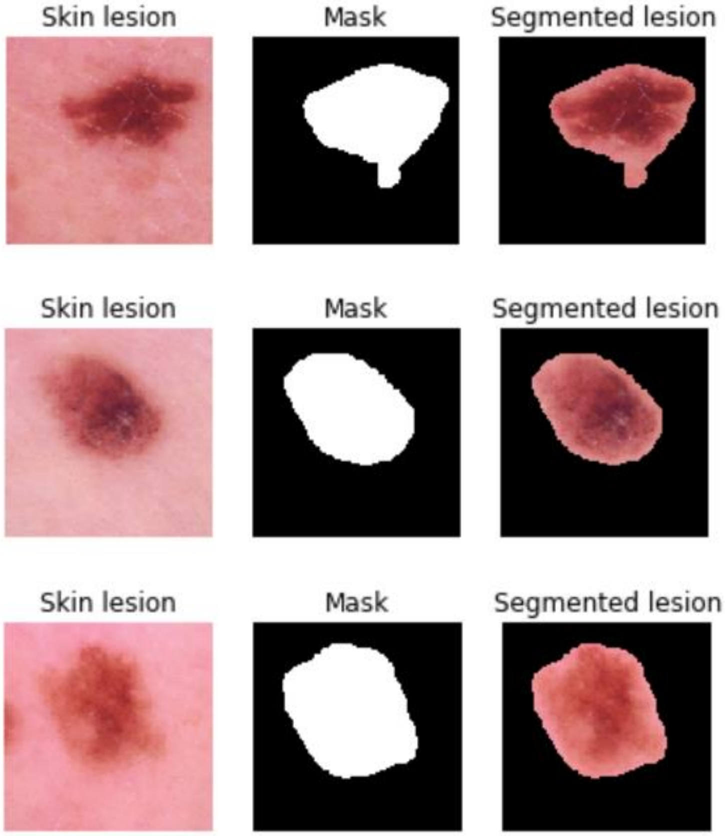


Figure 6: Sample of segmented images from the classification dataset

4.4.1 Classification results

The highest sensitivity achieved was 91.2% with the non-segmented images using the SGD optimizer with learning rate reduction.

Table 3: Summary of results

Scenario	Description	Sensitivity	Precision	Specificity	AUC
1	Segmented images with SGD optimizer	89.4%	21.8%	61.3%	0.83
2	Segmented images with SGD optimizer and learning rate reduction applied	88.9%	22.3%	62.6%	0.83
3	Segmented images with Adam optimizer	Scenarios did not work			
4	Segmented images with Adam optimizer and learning rate reduction applied	Scenarios did not work			
5	Non-segmented images with SGD optimizer	83.3%	28.3%	74.4%	0.86
6	Non-segmented images with SGD optimizer and learning rate reduction applied	91.2%	27.0%	70.2%	0.86
7	Non-segmented images with Adam optimizer	88.9%	25.5%	68.7%	0.87
8	Non-segmented images with Adam optimizer and learning rate reduction applied	84.7%	29.5%	75.5%	0.86

Receiver operator curves (ROC)

Figure 7 through Figure 10 show the ROCs for each of the scenarios tested and compare the performance between the segmented and non-segmented images.

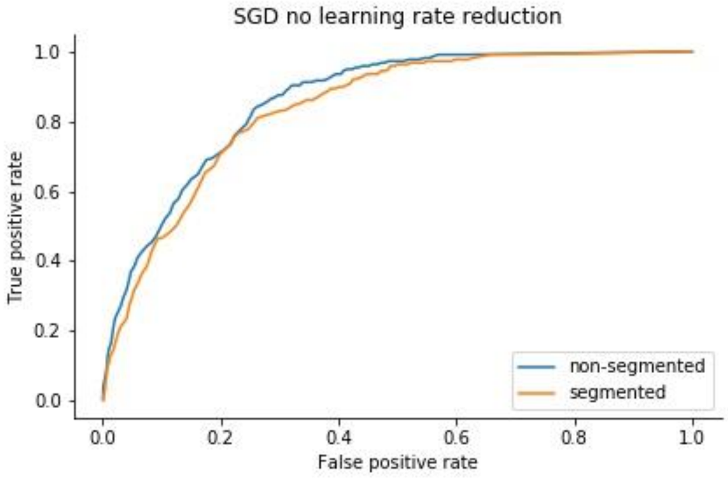


Figure 7: ROC from training with SGD for segmented and non-segmented images

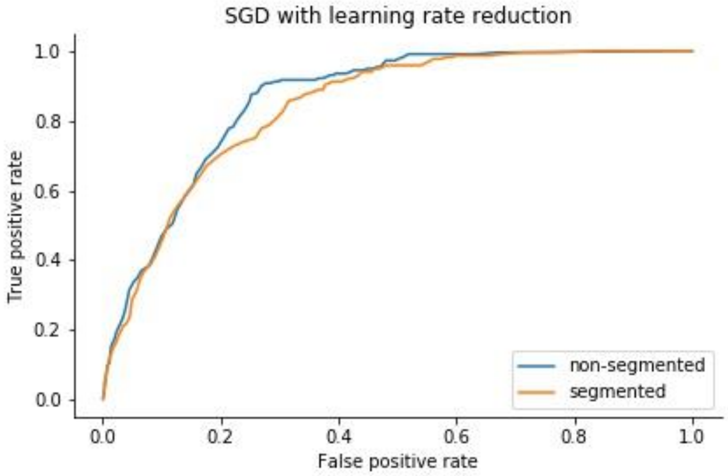


Figure 8: ROC from training with SGD and learning rate reduction for segmented and non-segmented images

It is clear from Figure 9 and Figure 10 that the combination of the Adam optimizer and the segmented images did not work. Unfortunately, we were unable to discover why.

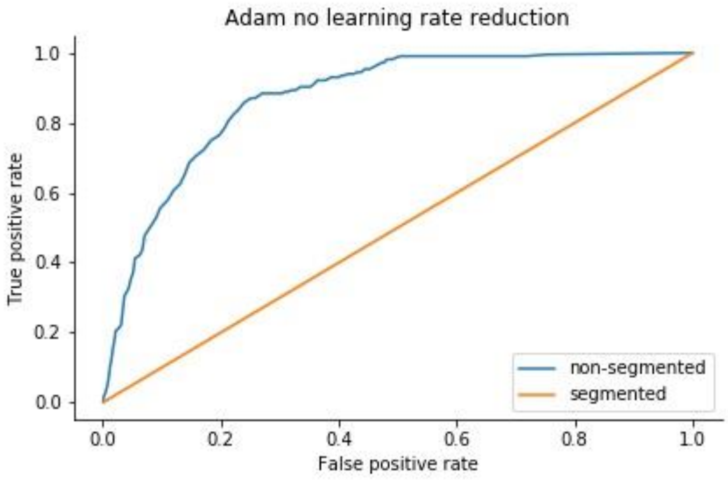


Figure 9: ROC from training with Adam for segmented and non-segmented images

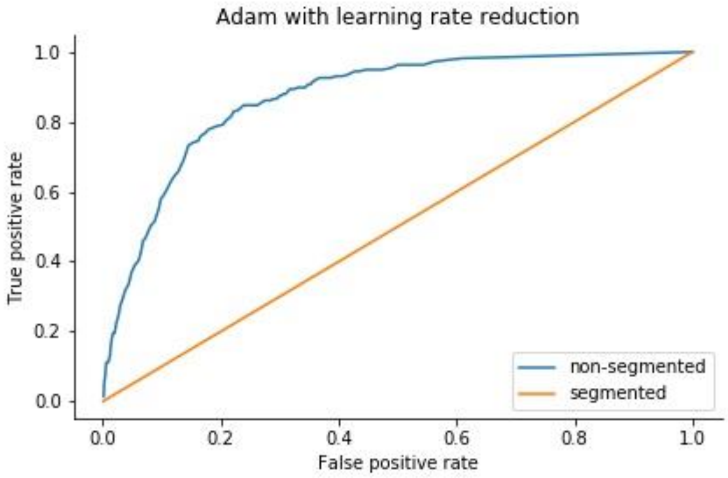


Figure 10: ROC from training with Adam and learning rate reduction for segmented and non-segmented images

4.4.2 Discussion

Surprisingly the non-segmented images performed better than the segmented images which suggests that segmentation (or our segmentation model) removed too much information from the image relevant to the classification of melanoma.

Below are 10 examples each of correctly classified melanoma (true positives) and missed melanoma (false negatives). Visually it is not obvious why the false positive results were missed.

True positives

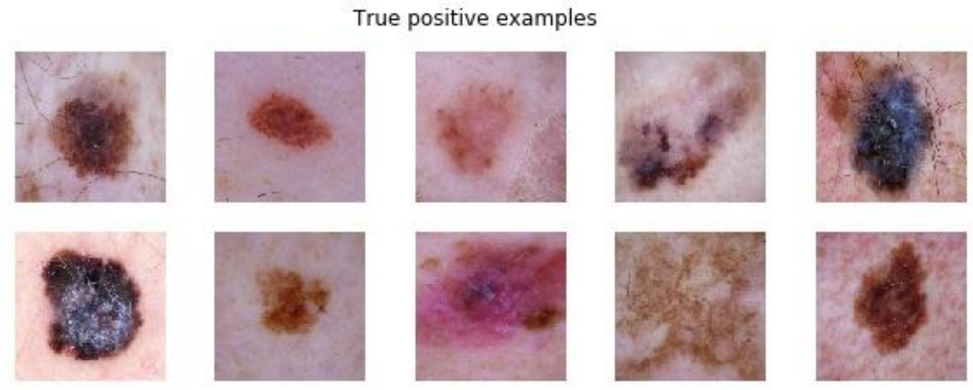


Figure 11: Correctly classified samples

False positives

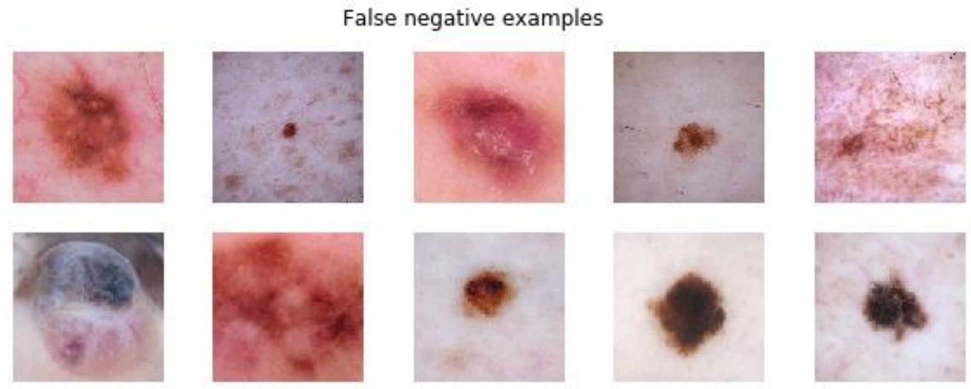


Figure 12: Incorrectly classified samples

5 Future Developments

The following developments in the future may improve our classification objective.

5.1 Running more epochs

It is clear from Figure 13 that the training and validation metrics were far from converging. Unfortunately, due to limited computational resources we decided that 10 epochs would provide us with results without the kernel crashing.

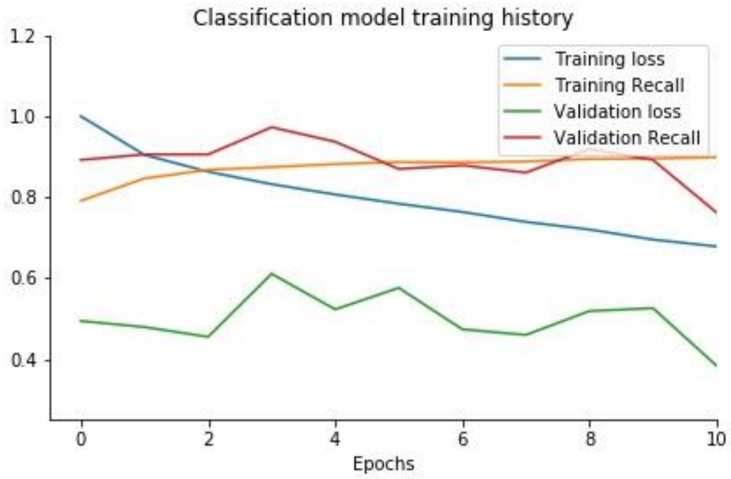


Figure 13: Plot of training and validation metrics

Obtaining a more reliable source of processing power using cloud services such as Google Cloud or Amazon Web Services would have helped, but proved to be beyond our capability in a short amount of time.

5.2 Using structures with fully connected layers – DenseNet

DenseNet is a fully connected architecture that is established as a state-of-the-art image classifier. Using an established structure, rather than starting from scratch may improve the outcome.

5.3 Image pre-processing

Removing artifacts such as hairs, markers, stickers and shadow from the image may improve the classification result.

Classifying the images at a higher resolution may also improve results.

6 References

1. Understanding Melanoma, Cancer Council Australia. 2017. Last medical review of source booklet, January 2017. https://www.cancer.org.au
2. Morton, D. L., Thompson, J. F., Cochran, A. J., Mozzillo, N., Elashoff, R., Essner, R., ... & Reintgen, D. S. (2006). Sentinel-node biopsy or nodal observation in melanoma. New England Journal of Medicine, 355(13), 1307-1317.
3. Argenziano, G., Puig, S., Iris, Z., Sera, F., Corona, R., Alsina, M., ... & Massi, D. (2006). Dermoscopy improves accuracy of primary care physicians to triage lesions suggestive of skin cancer. Journal of Clinical Oncology, 24(12), 1877-1882.
4. Vestergaard, M. E., Macaskill, P. H. P. M., Holt, P. E., & Menzies, S. W. (2008). Dermoscopy compared with naked eye examination for the diagnosis of primary melanoma: a meta-analysis of studies performed in a clinical setting. British Journal of Dermatology, 159(3), 669-676.
5. Bono, A., Bartoli, C., Cascinelli, N., Lualdi, M., Maurichi, A., Moglia, D., ... & Marchesini, R. (2002). Melanoma detection. Dermatology, 205(4), 362-366.
6. Celebi, M. E., Kingravi, H. A., Uddin, B., Iyatomi, H., Aslandogan, Y. A., Stoecker, W. V., & Moss, R. H. (2007). A methodological approach to the classification of dermoscopy images. Computerized Medical imaging and graphics, 31(6), 362-373.
7. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115.
8. Stanganelli, I., Longo, C., Mazzoni, L., Magi, S., Medri, M., Lanzanova, G., ... & Pellacani, G. (2015). Integration of reflectance confocal microscopy in sequential dermoscopy follow-up improves melanoma detection accuracy. British Journal of Dermatology, 172(2), 365-371.
9. Mishra, N. K., & Celebi, M. E. (2016). An overview of melanoma detection in dermoscopy images using image processing and machine learning. arXiv preprint arXiv:1601.07843.
10. https://scikit-image.org/docs/dev/auto_examples/segmentation/plot_morphsnakes.html#sphx-glr-auto-examples-segmentation-plot-morphsnakes-py
11. https://github.com/qubvel/segmentation_models
12. The international skin collaboration (ISIC) (2018, April 2). Skin Lesion Analysis Towards Melanoma Detection. *https://challenge2018.isic-archive.com*.
13. Kaggle (2018 October). Skin Cancer MNIST: HAM10000. https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000.
14. Guth F. & DeCampos T. E. (2018). Skin lesion segmentation using U-Net and good training strategies
15. Motsch A., Motsch S. & Saguet T. (2018). Lesion Segmentation using U-Net Network

7 Appendix

7.1 Response to feedback

Feedback	Response no.	Response
Proportion of testing data is 1:4. It would be better to use 1:10 or k-fold cross validation	1	As we are resource constrained, we did not want to have to re-run the models. Likewise, using k-fold would take a considerable amount of time
Could you please group 2,594 images into different groups and see if small group with same pattern has different AUC value.	2	We did initially have the images classified into smaller groups, thought we thought from a story telling perspective a binary approach would be better
Try some different optimization techniques to improve sensitivity and improve the class balance.	3, 4	We did eventually apply class weights to improve sensitivity. Two optimisation functions were tried.
Try to use F1 score and choose a threshold value the suits the requirement.	5	We thought that balancing Sensitivity and Precision would suffice
More detail on how to build a segmentation model	7	We used an “off the shelf” version and focused on the classifier
Include images of melanoma and non-melanoma	8	Refer section 4.4.2
What processes can be put in place to not misdiagnose melanoma	9	This tool in its current form is only meant as an initial indicator, we could set a very low threshold to ensure that we pickup all TP, however this will increase the occurrence of FP.
It would also be useful to see the false positive rate and estimate of the cost/benefits of unnecessary interventions vs correct diagnoses	10	This is interesting however due to lack of time will be ignored.
Can you divide samples into more classifications and analyze them respectively	11	We did this initially, but thought we should simplify the problem if we convert it to a binary classification task

7.2 Key tools that we used

Tools	Purpose
Google Colab	Google's cloud computing platform for running Jupyter Notebooks
Jupyter notebook and python	Processing images, training and testing models
Pillow	Ingesting images
Numpy and pandas	Data handling and manipulation
Keras and tensorflow	Convolutional neural network framework
Matplotlib and seaborn	For plotting and displaying images

7.3 Attached files

File	Type	Description
Segmentation images	Zip folder containing: Images (jpg) Masks (png)	Sample of images and masks used to train and test the segmentation model
Classification images	Zip folder containing images (jpg)	Sample of images used to train and test the classifier
HAM10000_metadata.csv	CSV	Metadata of the classification images
ResizeImagesMasks.ipynb	Jupyter notebook	Resize the images and masks for training and testing the segmentation model
ExploritoryDataAnalysis.ipynb	Jupyter notebook	Calculate class sizes visualisation
PreProcessImages.ipynb	Jupyter notebook	Resize and convert classification images into numpy arrays for training and testing the classification model
TrainUNetSegmentor.ipynb	Jupyter notebook	Train UNet
TestSeg.ipynb	Jupyter notebook	Test the segmentation model
SegmentImages.ipynb	Jupyter notebook	Segment the classification images
TrainTestSplit.ipynb	Jupyter notebook	Split classification dataset into training (60%), validation (20%) and testing (20%)
Classifier.ipynb	Jupyter notebook	Train classifier with various training hyper parameters
TestClassifier.ipynb	Jupyter notebook	Test the classifier variations