UNIVERSITÉ DU QUÉBEC À MONTRÉAL

APPLICATION DE MÉTHODES DE CLUSTERING ET D'APPRENTISSAGE PROFOND POUR LE DIAGNOSTIQUE DU CANCER DE LA PEAU

MÉMOIRE
PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INFORMATIQUE

PAR
JOCELYN BÉDARD

JUILLET 2022

ACKNOWLEDGEMENTS

I would like to thank Professor Vladimir Makarenkov, the director of the diplôme d'étude supérieure spécialisée (DESS) en bioinformatique program at the Université du Québec à Montréal (UQAM), for inviting me to do my Master's research with his group. I would also like to thank him for his advice in choosing the topic of my research as well as his support and guidance throughout my studies. In addition, I would like to mention my appreciation for his contribution to my financial support.

I am also grateful to Bogdan Mazoure, a Ph.D.student at McGill University, for allowing me to contribute and participate in the development of the DUNEScan application. I appreciated the discussions we had together during the preparation of the manuscript published recently.

Also, I would like to thank my family for their continued support throughout my studies.

I would also like to thank Prof. Alexandre Blondin Massé , Dr. Guy Francoeur, Dr. George Debay, Dr. Mélanie Lord and Dr. Alix Boc who hired me for teaching assistant positions throughout my studies. These positions allowed be to gain further experience in teaching as well as provided me with additional financial support.

Finally, I would like to thank the Natural Sciences and Engineering Science Council (NSERC) for their generous financial support through the Canada graduate scholarships - Master's program. In addition, I would also like to thank the science faculty of UQAM which also provided financial support through the Bourses de Maîtrise de la Faculté des Sciences program.

CONTENTS

LIST OF FIGURES vi

LIST OF TABLES ix

RÉSUMÉ x

CHAPTER I INTRODUCTION 1

CHAPTER II STATE OF THE ART IN SKIN LESION IMAGE DIAGNOSIS 4

2.1 skin lesion image classification 4

2.2 CAD application developement 8

CHAPTER III RESEARCH PROBLEMATIC AND PROPOSED RESEARCH 10

CHAPTER IV MATERIALS AND METHODS 13

4.1 source of data 13

4.2 preprocessing 15

4.3 input preparation 16

4.4 CNN models 16

4.5 CNN training 18

4.6 K-means analysis 20

CHAPTER V RESULTS 23

5.1 analysis of the full ISIC dataset 23

5.2 seperation of the datasets 30

5.3 unsupervised classification of images 34

5.4 classification in reduced numbers of classes 68

5.5 CAD application for binary skin lesion image classification 74

5.5.1 uncertainty estimation 76

5.5.2 description of the DUNEScan output 77

5.5.3 testing of the application 77

CHAPTER VI DISCUSSION 87

CHAPTER VII CONCLUSION 94

REFERENCES 96

LIST OF FIGURES

Figure Page

5.1 Model size vs. ImageNet accuracy 25

5.2 Mean WACC with the ISIC2019 dataset after increasing numbers of training epochs. 28

5.3 Mean WACC with the BCN and HAM datasets after increasing numbers of training epochs. 33

5.4 Silhouette coefficient distributions for K-means clustering of non-normalized BCN image pixel vectors. 40

5.5 Silhouette coefficient distributions for K-means clustering of Z-score normalized BCN image pixel vectors. 41

5.6 Overlap between K-means image clusters produced from pixel vectors. 42

5.7 Overlap between randomly produced datasets of BCN images. . . 45

5.8 Mean WACC with the BCN random datasets after increasing numbers of training epochs. 46

5.9 Silhouette coefficient distributions and K-means clusters (K-2 and K=3) of non-normalized image pixel vectors reduced with SVD. . 48

5.10 Silhouette coefficient distributions and K-means clusters (K=4 and K=5) of non-normalized image pixel vectors reduced with SVD. . 49

5.11 Silhouette coefficient distributions and clusters for K-means clustering (K=2 and K=3) of MinMax normalized BCN image pixel vectors reduced with SVD. 50

5.12 Silhouette coefficient distributions and clusters for K-means clustering (K=4 and K=5) of MinMax normalized BCN image pixel vectors reduced with SVD. 51

5.13 Silhouette coefficient distributions and clusters for K-means clustering (K=2 and K=3) of non-normalized BCN image RN50avg feature vectors reduced with SVD. 56

5.14 Silhouette coefficient distributions and clusters for K-means clustering (K=4 and K=5) of non-normalized BCN image RN50avg feature vectors reduced with SVD. 57

5.15 Silhouette coefficient distributions and clusters for K-means clustering (K=2 and K=3) of MinMax normalized BCN image RN50avg feature vectors reduced with SVD. 58

5.16 Silhouette coefficient distributions and clusters for K-means clustering (K=4 and K=5) of MinMax normalized BCN image RN50avg feature vectors reduced with SVD. 59

5.17 Silhouette coefficient distributions and clusters for K-means clustering (K=2 and K=3) of non-normalized BCN image RN50max feature vectors reduced with SVD. 60

5.18 Silhouette coefficient distributions and clusters for K-means clustering (K=4 and K=5) of non-normalized BCN image RN50max feature vectors reduced with SVD. 61

5.19 Silhouette coefficient distributions and clusters for K-means clustering (K=2 and K=3) of MinMax normalized BCN image RN50max feature vectors reduced with SVD. 62

5.20 Silhouette coefficient distributions and clusters for K-means clustering (K=4 and K=5) of MinMax-normalized BCN image RN50max feature vectors reduced with SVD. 63

5.21 Overlap between K-means image clusters produced from ResNet50-extracted features. 65

5.22 Overlap between K-means image clusters produced from pixel vectors and those produced from ResNet50-extracted features. 66

5.23 Screenshots of the main features of our DUNEScan web server. . . 78

5.24 Boxplots representing the uncertainty estimates for two malignant lesions 81

5.25 Boxplots representing uncertainty estimates for a first pair of benign lesions 82

5.26 Boxplots representing uncertainty estimates for a second pair of benign lesions 86

LIST OF TABLES

Table Page

4.1 Distribution of the images amongst the different disease classes in the datasets. 14

5.1 Classification results of the ISIC2019 dataset and its sub-datasets into eight classes. 26

5.2 K-means clustering results with the MSK image dataset and subsets of the BCN and HAM datasets. 36

5.3 K-means clustering of pixel vectors produced with the NV and MEL image datasets including BCN, HAM and MSK images. 37

5.4 Silhouette scores obtained for different K-means BCN image clusters based on pixel vectors. 38

5.5 Classification results with K-means BCN image pixel vector clusters into eight disease classes. 43

5.6 Silhouette scores obtained for different K-means BCN image clusters based on RN50 feature vectors. 55

5.7 Classification results with K-means BCN image RN50 feature vector clusters into eight disease classes. 67

5.8 K-means clustering of ResNet50 feature vectors produced with NV and MEL image datasets including BCN, HAM and MSK images. 69

5.9 Classification results with K-means BCN image clusters into two or four disease classes. 72

5.10 Class prediction probabilities obtained for various images by the different CNN models available in DUNEScan. 85

RÉSUMÉ

Le cancer de la peau est un des cancers les plus fréquents dans le monde. Plusieurs types de cancers de la peau existent, parmi lesquels le mélanome est le plus sérieux car il peut, s'il n'est pas diagnostiqué tôt, causer le décès du patient. Des modèles d'apprentissage profond ont été utilisés pour classifier les lésions cutanées causées par les cancers en différentes classes en utilisant des images digitales. Ces modèles d'apprentissage, qui utilisent des architectures de réseaux de neurones à convolution modernes, telle que EfficientNet, furent testés en appliquant l'approche de transfert d'apprentissage (*i.e.* transfer learning). Des méthodes de clustering ont aussi été employées pour tenter de regrouper les images à la base de caractéristiques générales afin de produire des jeux de données plus homogènes qui permettraient d'effectuer la classification supervisée avec une plus grande précision. Certaines méthodes de clustering que nous avons appliquées ont permis d'obtenir des meilleurs résultats que ceux obtenus avec des jeux de données de taille équivalente produits aléatoirement. Cependant, dans tous les cas, les meilleurs résultats ont été obtenus avec les jeux de données complets (*i.e.* sans appliquer le regroupement au préalable). Durant mon projet, j'ai aussi effectué les tests nécessaires pour la mise en place de DUNEScan, une nouvelle application web pour l'analyse d'images de cancers de la peau avec des réseaux de neurones profonds. J'ai également participé à la rédaction d'un article qui décrit cette application.

Mots-clés: skin cancer, skin lesion, deep learning, clustering, convolution neural network, classification, computer vision, diagnosis

CHAPTER I

INTRODUCTION

Skin cancer is the most common and widespread form of cancer in the world ((Lacy et Alwan, 2013),(Karimkhani *et al.*, 2017)). In the recent decades the number of cases has increased significantly due, in most cases, to increased over-exposure to ultraviolet light emitted by the sun. Skin cancer cases are more abundant than the combined counts of breast, lung and colon cancers. Various types of skin cancers exist, among which, melanoma, which affects the melanin-producing skin cells, is the most serious type and can be lethal if not diagnosed early-on. Melanoma, like other types of skin cancers (*e.g.* basal cell carcinoma and squamous cell carcinoma) can either be malignant or benign. These must therefore be correctly diagnosed and followed with close attention to ascertain that they don't develop into a more aggressive form. All forms of skin cancer are characterized by the appearance of skin lesions with an abnormal coloration and irregular shape.

Traditionally initial diagnosis of these skin lesions has been performed by dermatologists with the use of digital dermoscopic images. Dermoscopy is a form of imaging that uses polarized light that generates high resolution images that allow a deeper detailed view of the texture by reducing light reflection (Toader *et al.*, 2017). Some of the lesion traits such as size, shape and texture can be used to differentiate them successfully. However, due to their complex morphological structure and overall similarity in appearance, visual analysis can lead to miss-diagnosis, which may have severe health consequences (Lee *et al.*, 2018). For this reason, the development of computer-assisted diagnosis (CAD) tools to assist experts is highly desirable.

CAD systems are now commonly used in various medical fields where different forms of imaging and radiology are used (Anwar *et al.*, 2018). These applications use computer vision algorithms based on deep learning, a recent advance in the machine learning field. In most cases, these rely on convolutional neural network (CNN) models, a form of artificial neural network that can be trained for image recognition and classification. In this context, convolution is a process used to extract image features and, when used in sequence, several convolution steps transform the images into abstract forms that can be interpreted by a computer to distinguish between different pre-defined classes of images.

Each convolution step uses a specified number of small filters (usually 3x3 or 5x5 pixels) that parse the image to recognize specific forms or features. The image is interpreted as a matrix of pixel values and each filter unit corresponds to a trainable parameter value (or weight). As one filter parses the image, usually by a stride of one pixel at a time, the sum of the multiplications of each filter unit with the corresponding pixel value is returned to produce a feature map. The number of filter maps returned correspond to the number of filters used.

The produced filter maps are then passed on to the next convolutional layer where in turn other filters are used to extract information. At each convolution step the filters correspond to a matrix with a specified x*x size and a depth equivalent to the number of input layers. For example, if a colour image is analyzed this image is interpreted as matrix with three layers, where each layer corresponds to red, green and blue pixel values. A first set of 3x3 convolution filters would actually correspond to 3x3x3 matrices which would allow parsing of the three image layers silmultaneously. Such a filter would therefore contain 27 units or 27 trainable parameters (weights).

As the information is propagated forward, feature maps can be condensed by merging (pooling) information. Finally, when the specified number of steps in the convolution architecture are completed, all values in the final feature maps are transformed into a one-dimensional vector. These final values are interpreted by a classifier to predict the class of the input image. As training of a CNN occurs, the predicted class is compared to the actual class and an error value is obtained. This value is then used by an optimizer function to modify the weights of all the the filters by backpropagation of the error gradient.

Generally, CNN models rely on a very high number of labelled images for training so it learns how to discriminate between image classes and achieve high accuracy predictions. In many cases, although skin cancer is a relatively common disease, the number of quality skin lesion images in local databases is limited. Also, because the skin lesions produced by different types of skin cancers show a low degree of variation, producing a reliable model can be a challenging task.

In the following sections and subsections, I will summarize the current state of the field of CAD system development for skin lesion diagnosis, present the problem setting, our proposed research and the expected impacts, present the material and methods used, the results of my research and a discussion.

CHAPTER II

STATE OF THE ART IN SKIN LESION IMAGE DIAGNOSIS

2.1 skin lesion image classification

Some skin lesion classification methods have been developed based on models that take in consideration the visible attributes of skin lesions such as the diameter, pigmentation, shape and texture, to make predictions(Moura *et al.*, 2019). However, this approach requires segmentation of the lesion images in order to produce a mask that delimits the lesion and allow accurate measurements to be made. Such masks can be produced manually, but again these require the help of an expert, or automatically with a CNN approach. In (Moura *et al.*, 2019) images with available, manually-produced masks were used as source data to develop a model that uses such physical data in combination with features extracted by CNNs to detect melanocytic lesions. However, since the limited set of skin lesion images used in their study is not sufficient to develop and train a CNN model specific for skin lesion classification, they use an alternative approach called transfer learning.

Transfer learning consists of using CNN architectures that have been pre-trained to classify unrelated images and applying them to a new classification task. Since the pre-trained architectures used have already learned to extract features from a vast collection of images of varying objects, they apply the same method to their lesion images to rapidly produce feature sets. Specifically, in the development of their method, the authors tested several state-of-the-art CNN architectures that were all pre-trained on the ImageNet image collection (Deng *et al.*, 2009)(Russakovsky *et al.*, 2015). This collection contains several thousand images of different objects that belong to 1000 different classes. The CNN architectures used included the AlexNet (Krizhevsky *et al.*, 2012), CaffeNet (Jia *et al.*, 2014) and the VGG-16 and VGG-19 (Simonyan et Zisserman, 2014) architectures.

After testing the different CNNs, they selected the two that best differentiated the melanocytic lesions from the others. They combined the features obtained from the final convolutional layers with the obtained physical features to generate a very large multi-feature descriptor of the lesion images. They selectively used only the features that have the best ability to differentiate the images and fed them into a classifier consisting of a few layers of fully connected neurons (a multi-layer perceptron) that achieves 94.9% prediction accuracy with their binary classification problem (melanoma vs. others). However, since their model can only differentiate melanocytic lesions from non-melanocytic lesions, it does not permit to identify the type of cancer that caused the non-melanocytic lesions. In addition, since their approach was developed using a relatively small number of images from two databases, it may not generalize well on images from different sources and non-segmented images.

Several other groups have also recently produced models that rely solely on transfer-learning and fine-tuning to develop multi-class skin lesion classifiers ((Yu *et al.*, 2018), (Dorj *et al.*, 2018),(Han *et al.*, 2018),(Menegola *et al.*, 2017)). In most cases they use similar pre-trained state-of-the-art architectures as those mentioned previously. They use the pre-set parameters of most of the convolution layers of these pre-trained CNNs, and only fine-tune the parameters of the last layer. Fine-tuning is done by further training the CNNs with labelled skin-lesion images in such a way that the parameters are adjusted and adapted to differentiate between different types of skin lesions. Because these new models no longer rely on measured physical traits, segmentation has become a less essential step. This approach has been made possible because a large number of skin lesion images have recently been made publicly available.

In order to carry out fine-tuning, many groups use the human against machine (HAM10000) skin lesion image collection which was used in the 2018 International Skin Imaging Collaboration (ISIC) skin challenge (Codella *et al.*, 2019). In this competition, scientists were asked to develop algorithms than can perform different tasks related to image classification (*e.g.* segmentation, classification). The HAM10000 collection is made up of 10,015 expert-labelled images that represent lesions from seven different types of skin cancers or diseases (Tschandl *et al.*, 2018). These include melanoma, melanocytic nevus, basal cell carcinoma, actinic keratosis, benign keratosis, dermatofibroma and vascular lesions. Other, less large skin lesion image collections have also become available which contain images from other types of skin diseases. As an example, the Dermofit image collection, contains a small number of images from squamous cell carcinoma (88) and pyogenic granuloma (24).

In most cases, images obtained from these databases are preprocessed before being used to train a model. Such preprocessing includes hair removal to reduce noise as well as contrast and brightness normalization. In one noteworthy study (Esteva *et al.*, 2017), a model was developed using a transfer learning approach with a very large dataset of images. Images were pooled from many databases including the ISIC repository images (pre-HAM10000 era), the dermofit library and other large private image collections. Altogether they generated a dataset of 129,450 images, which was composed of normal digital photographic images and only 3,374 dermoscopic images. Using this large dataset, they produced a model using the pre-trained GoogleNet Inception V3 state-of-the-art architecture (Szegedy *et al.*, 2016), for which they fine-tuned every layer of the CNN.

Data-augmentation, but no traditional preprocessing was applied to the images. Data-augmentation is a process through which a dataset can be increased in size by using techniques, such as rotation and flipping of images to generate new images and increase the variability of the dataset. This approach allowed them to generate a model robust to photographic variation. By comparing the prediction of their model with those of a panel of dermatologists in different binary classification tasks, they showed that their model is at least as accurate as the expert panel. In a later study by a different group (Bakkouri et Afdel, 2019), the same model was tested exclusively on the HAM10000 dataset for comparison purposes and achieved an accuracy of 91.6%.

More recently, a novel method was developed that combines fine tuning of state-of-the-art architectures with a multi-layer fusion approach (Bakkouri et Afdel, 2019). Other studies have shown that the initial (lower) layers of a CNN detect general features that can often be used to define various types of images, whereas medium and high-level layers detect more detailed features specific to the different image classes analyzed. Because of this, as a first step in (Bakkouri et Afdel, 2019), they determined how many layers from each of three CNNs should be fine tuned to obtain the best possible classification accuracy. Here they found that both VGG-16 and ResNET-18 (He *et al.*, 2016) performed better when the weights of the first three convolution layers were frozen (not fine-tuned) and the remaining were fine-tuned. In the case of DenseNet-121 (Huang *et al.*, 2017), a more recent and larger CNN architecture, the first four convolutional layers were frozen to obtain optimal performance.

The frozen parts of each of these architectures were then used to feed a novel convolutional fusion unit architecture in which multi-level feature maps from each of these architectures are fused together and further convoluted. Information from different levels of convolution were also used to produce the final image descriptor. This method, to my knowledge obtained the highest level of average classification accuracy of 98.1% with the HAM10000 image collection. There results show that combining the information obtained at multiple levels of CNN can result in improved differentiation between skin lesion images of multiple classes. However, it is important to note that the model used is very complex and training was very time consuming. Nonetheless, once training was completed, obtaining a classification with each test image was quite quick (*i.e.* 4 seconds with a GPU and 7.5 seconds with a CPU).

Clearly, research in classification of skin lesion images for the diagnosis of skin cancers has progressed significantly in the recent years. Many models which achieve a very high accuracy level have been produced using different approaches. However, in general terms most of the best performing models rely to a certain extent, if not solely, on transfer learning.

2.2 CAD application developement

Several applications have been developed to allow self-diagnosis and assist dermatologists in the diagnosis of skin lesions images (Wise, 2018) (Abbott et Smith, 2018). Most apply recent advances in convolutional neural network (CNN) architecture design and allow to obtain a diagnosis with an accuracy that match that obtained by dermatologists. Some recent studies have shown that current CNN models developed for skin cancer diagnosis can even surpass the accuracy of dermatologists in their diagnosis (Brinker *et al.*, 2019). However, several of the currently available applications use proprietary models and require a licence to permit their use (Abbott et Smith, 2018). This limits their availability and also the ability to assess their performance on public skin lesion image datasets.

Also, to our knowledge, all skin cancer diagnosis applications do not allow to estimate the confidence level with which they make their predictions. As a consequence, although large leaps have been made in the development of such CAD applications and this field of research is very competitive, their is still room to produce high quality applications that are more accessible and have additional features.

CHAPTER III

RESEARCH PROBLEMATIC AND PROPOSED RESEARCH

As discussed in the previous section, due to the high prevalence and important effect of skin cancers on global population health, a large amount of effort has been invested to develop high accuracy CAD systems for skin lesion diagnosis. Nonetheless, recent studies based on a large, novel and publicly available skin lesion image dataset, made up of images from different sources, showed that developing models that can generalize well and accurately differentiate images representing multiple disease class is very challenging. Even the winning participants of the ISIC 2019 skin challenge who were charged with this task only achieved a very limited level of classification accuracy (Gessert *et al.*, 2020).

The initial goal of our project is to, by comparing available CNN architectures and deep learning strategies, develop an efficient model that can generalize well and be used to analyze and classify skin lesions images from different sources. The general approach will be to pretrain some of the most recent state-of-the art CNN models available by transfer learning with the same dataset used in the ISIC2019 challenge and assess their performance.

We are particularly interested in investigating the reason behind the low performance achieved by Gessert *et al.*(2020) as compared to that achieved, as previously discussed, by some other groups with the HAM10000 dataset used independently.

For this purpose, when the best performing model for classification of the ISIC2019 dataset is identified, we will proceed to analyze the images originating from different sources as sperate sub-groups with the same model.

We also aim to investigate if, when such a large, diverse skin lesion image dataset is available for model training, a basic unsupervised clustering approach (*e.g.* K-means) may allow us to produce specialized models that achieve higher classification accuracies. Possibly, K-means clustering of images based on different image features such as pixel composition or other general image features may be useful to form more homogenous groups of images. Such image groups may allow us to produce classification models with improved performance.

If such a strategy proves to be successful, a two-step image classification pipeline could be implemented. As a first step, an input image would be classified with a pretrained unsupervised clustering model to determine to which cluster the image belongs to. The result of this first classification step, would allow us to determine which model, amongst a subset of specialized models, would be best suited to, as a second step, accurately determine the disease class of the image.

Finally, in addition, we aim to contribute to the development of a CAD system that carries out high accuracy skin lesion diagnosis. If our previously described attempts to develop a novel classification stategy is fruitful, we would aim to implement this strategy as part of the proposed application. Otherwise, the application will be designed to rely on different pre-trained state-of-the-art CNN architectures.

As previously mentioned, high performance CAD systems can be very useful in assisting specialists in the diagnosis of various types of diseases. The development of an open source, freely accessible CAD application may be of greater use than the currently available expensive, proprietary systems trained on private datasets.

In the future, such an application could be potentially implemented as a mobile device application that could be used to diagnose images obtained with such devices (*e.g.* smartphones, tablets). Such an application would allow users to carry-out a self-assessment of detected skin lesions and hopefully encourage them, when necessary, to look for further medical advice by specialists.

CHAPTER IV

MATERIALS AND METHODS

4.1 source of data

The image dataset used for the ISIC2019 Challenge was used in this study. This dataset contains a collection of 25,331 skin lesion images originating from three separate sources: 12,413 images from the BCN_20000 (BCN) dataset (Combalia *et al.*, 2019), 10,015 from the HAM_10000 (HAM) previously described dataset (Tschandl *et al.*, 2018) and 2,903 images from the MSK dataset(Berseth, 2017).

The HAM dataset contains images of size 600x450 that were centered and cropped around the lesion and histogram corrections were applied to some images by the dataset curators. The BCN images are of size 1024x1024 and contain many uncropped regions or hard to diagnose lesions in uncommon body locations. Finally, the MSK images have various sizes and many images labelled as downsampled.

The dataset contains eight unbalanced classes of skin disease images: melanoma (MEL), melanocytic nevus (NV), basal cell carcinoma (BCC), actinic keratosis (AK), benign keratosis (BKL), dermatofibroma (DF), vascular lesion (VASC) and squamous cell carcinoma (SCC). The previously described HAM dataset was reannotated to produce the new SCC class, which was absent in the original 2018-version of the dataset.

Table 4.1: Distribution of the images amongst the different disease classes in the datasets.
The number of images for the different datasets used in this study in each of the eight disease classs is indicated along with the proportion of the total number of images represented by each class indicated between parentheses. The datasets include the full ISIC2019 dataset, the MSK sub-dataset, the BCN sub-dataset and the HAM sub-dataset. Also, all datasets corresponding to K-means (K=2) clusters generated from pixel vectors and ResNet50 (RN50) extracted feature vectors. The pixel vector-based datasets include those produced with the raw pixel vectors (rawImg), those produced from Z-score normalized pixel vectors (scaledImg) and those produced with minmax-normalized pixel vectors further reduced to two features by SVD (MinMaxSVD). The RN50 feature vector-based datasets were all produced using global average pooled RN50 feature vectors (avg) or global maximum pooled RN50 feature vectors (max). These vectors were either minmax-normalized (MinMax) or non-normalized (NoNorm) and further reduced to two features by SVD. In addition, the distribution of two randomly produced BCN image datasets (random6200_1 and random6200_2) is presented. These two datasets were designed to maintain the same class distribution as the full BCN sub-dataset.

Datasets	MEL	NV	BCC	AK	BKL	DF	VASC	SCC	Total
ISIC2019	4522 (0.179)	12875 (0.508)	3323 (0.131)	867 (0.034)	2624 (0.104)	239 (0.009)	253 (0.010)	628 (0.025)	25331
MSK	552 (0.190)	1964 (0.677)	0	0	387 (0.133)	0	0	0	2903
HAM	1113 (0.111)	6705 (0.670)	514 (0.051)	130 (0.013)	1099 (0.110)	115 (0.011)	142 (0.014)	197 (0.020)	10015
BCN	2857 (0.230)	4206 (0.339)	2809 (0.226)	737 (0.059)	1138 (0.092)	124 (0.010)	111 (0.009)	431 (0.035)	12413
BCN K-means									
Pixel Vectors									
rawImg									
2-0	1534 (0.213)	2952 (0.411)	1446 (0.201)	309 (0.043)	574 (0.080)	69 (0.010)	74 (0.010)	220 (0.031)	7178
2-1	1323 (0.252)	1254 (0.239)	1363 (0.260)	428 (0.082)	564 (0.108)	55 (0.011)	37 (0.007)	211 (0.040)	5235
scaledImg									
2-0	1536 (0.257)	2079 (0.348)	1261 (0.211)	304 (0.051)	528 (0.088)	45 (0.008)	64 (0.011)	163 (0.027)	5980
2-1	1321 (0.205)	2127 (0.331)	1548 (0.241)	433 (0.067)	610 (0.095)	79 (0.012)	47 (0.007)	268 (0.042)	6433
MinMaxSVD									
2_0	1541(0.214)	2954(0.411)	1442(0.201)	310(0.043)	576(0.080)	69(0.010)	74(0.010)	219(0.030)	7185
2_1	1316(0.252)	1252(0.239)	1367(0.261)	427(0.082)	562(0.107)	55(0.011)	37(0.007)	212(0.041)	5228
NoNormSVD									
2_0	1539(0.214)	2958(0.411)	1446(0.201)	310(0.043)	577(0.080)	69(0.010)	74(0.010)	219(0.030)	7192
2_1	1318(0.252)	1248(0.239)	1363(0.261)	427(0.082)	561(0.107)	55(0.011)	37(0.007)	212(0.041)	5221
RN50 Features									
avgMinMaxSVD									
2-0	1508 (0.251)	1566 (0.261)	1501 (0.250)	465 (0.077)	626 (0.104)	62 (0.010)	44 (0.007)	234 (0,039)	6006
2-1	1349 (0.211)	2640 (0.412)	1308 (0.204)	272 (0.042)	512 (0.080)	62 (0.010)	67 (0.010)	197 (0.031)	6407
avgNoNormSVD									
2-0	1510 (0.250)	1624 (0.269)	1488 (0.246)	469 (0.078)	622 (0.103)	64 (0.011)	42 (0.007)	226 (0.037)	6045
2-1	1347 (0.212)	2582 (0.405)	1321 (0.207)	268 (0.042)	516 (0.081)	60 (0.009)	69 (0.011)	205 (0.032)	6368
maxMinMaxSVD									
2-0	1225 (0.204)	2573 (0.428)	1175 (0.196)	248 (0.041)	486 (0.081)	61 (0.010)	62 (0.010)	175 (0.029)	6005
2-1	1632(0.255)	1633 (0.255)	1634 (0.255)	489 (0.076)	652 (0.102)	63 (0.010)	49 (0.008)	256 (0.040)	6408
maxNoNormSVD									
2-0	1683 (0.251)	1713 (0.255)	1730 (0.258)	503 (0.075)	688 (0.103)	67 (0.010)	52 (0.008)	271 (0.040)	6707
2-1	1174 (0.206)	2493 (0.437)	1079 (0.189)	234 (0.041)	450 (0.079)	57 (0.010)	59 (0.010)	160 (0.028)	5706
BCN random									
random6200-1	1427 (0.230)	2101 (0.339)	1403 (0.226)	368 (0.059)	568 (0.092)	62 (0.010)	55 (0.009)	215 (0.035)	6199
random6200-2	1427	2101	1403	368	568	62	55	215	6199

The distribution of the images of the ISIC2019 dataset and each sub-datasets is presented in Table 4.1. As can be observed the most abundant, predominant disease class is the NV class. The MEL class is the second most abundant class in all datasets. The BCN dataset shows a more balanced distribution amongst the MEL, NV, BCC, AK, and BKL image classes than the HAM dataset, whereas in both these datasets some classes such as the DF and VASC are largely under-represented. The MSK dataset is particular as it only contains images from three classes: MEL, NV and BKL.

4.2 preprocessing

The same strategy as used by Gessert *et al.*(2020) was implemented with MatLab to normalize the images. This included image cropping to center the skin lesion and reduce the large black border present in several images from the BCN dataset. Briefly, binarized versions of the images (converted to 0 and 1 pixel values) were produced with a low threshold such that the dermoscopy area (the area containing the actual image) was assigned a 1 value. This allowed us to find the center of mass of the images along with the major and minor axis of an ellipse that has the same second central moment as the inner area of the image. Based on these values a rectangular bounding box was derived and used to crop the relevant field of view.

The necessity for cropping was determined based on a heuristic that tests if the mean intensity inside the bounding box is substantially different from the mean intensity outside the box. After the cropping step, the Shades of Grey color constancy method (Mendoza et Lu, 2015) was used to normalize the intensity of the images. Finally, the images were resized such that the longer side was equal to 600 pixels while preserving their aspect ration.

4.3 input preparation

The images were transformed from their original sizes with a same-size cropping strategy. A random crop of the specified input size is taken from the preprocessed images.

For each experiment, the images are scaled and autoaugmented by applying various transformations at each training epoch. We use random brightness and contrast changes, random flipping, random rotation, random scaling and random shear. Random cut-out is also used where one hole of size 16 is randomly placed in the image.

4.4 CNN models

Most experiments were done using the EfficientNet B4 model. A series of seven EfficientNet models has recently been developed by Tan *et al.*(2019) (Tan et Le, 2019). These models were produced using a compound scaling approach by which in each model a balanced scaling approach was used to increase the CNN depth (number of layers), the layer width (number of channels) and the input image resolution.

The largest model of this series EfficientNet B7 obtained state-of-the-art accuracy when tested on ImageNet dataset (Tan et Le, 2019). The base model of this series, EfficientNet B0, is composed of convolution blocks similar to the residual network blocks of the MobileNetV2 model (Sandler *et al.*, 2018) to which squeeze-and-excitation optimization (SE) (Hu *et al.*, 2018) was added to allow dynamic channel-wise feature recalibration. This model takes as input images of 224x224 resolution. In the case of EfficientNet B4 input images of 380x380 resolution were required.

Residual networks CNNs were first used in the Residual Network (ResNet) models developed by He *et al.* (2016). These residual networks consist in using skip connections between convolution blocks. For each convolution block, in addition to being passed to the next convolution block the output is also passed to the second next block of higher level in the architecture.

This development represented a breakthrough in the design of CNN architectures as it provided a solution to the vanishing gradient problem encountered by very deep CNN models. Previously, the improvement of accuracy of CNN models by increasing their depth was limited. In very deep CNNs the gradient error that is backpropagated through the layers to adjust the weights of the convolution filters diminished progressively and became close to zero. The skip connections of the ResNets preserve the gradient error through a greater number of layers.

Two new variants of ResNet models were tested in our study: SE-ResNet50 (Hu *et al.*, 2018) and ResNext101 (Xie *et al.*, 2017). The SE-ResNet50 is a mid-size traditional ResNet model to which SE was added to all convolution blocks. In contrast RexNext101 is a larger, more recent version of the ResNet models in which, instead of using a single filter at each convolution layer, similar filters are used in multiple branches and the produced feature maps are concatenated. Images of 224x224 resolution were used with both these models.

In addition, the InceptionV3 model (Szegedy *et al.*, 2016) was also tested. Like ResNext models, the Inception models use multiple filters in separate branches at each convolutional layer. InceptionV3 also uses additional strategies such as factorized convolutions, asymmetric convolutions as well as regularization by an auxiliary classifier to improve its performance (*i.e.* favoring accuracy while keeping computational costs relatively low). To evaluate the importance of the auxiliary classifier, InceptionV3 was used either by using the error obtained by the auxiliary classifier compounded with those of the main classifier during training or by ignoring the error of the auxiliary classifier. As recommended images of 299 x 299 resolution were used with this model.

Finally, DenseNet121 (Huang *et al.*, 2017), another popular CNN model was tested. This model also used inter-layer connections such as those used in the residual networks, but in this case feature maps produced by one convolutional layer are shared with all forward layers. This approach is an alternative to the simple shortcut connections of the residual networks to avoid the vanishing gradient problem. Consequently, DenseNet models are said to use densely connected layers and, like ResNets, can be very deep.

4.5 CNN training

The CNN models were implemented with the Python PyTorch library for deep learning. In each experiment a 80:20% split of the image datasets was used for training and validation steps respectively. Most pretrained CNN architectures were obtained from the PyTorch pretrained models module with the exception of efficientNets, which were obtained from the efficientNets PyTorch module.

The models were trained for 60 or, in most cases, 100 epochs with the Adam optimizer. A weighed cross-entropy loss function was used where under-represented class images received a higher weight-based frequency in the training set. The weight of each class was calculated with the $w = Ni/N$ formula where Ni represents the number of images in the class and N the total number of images used for training. In each case a batch-size of 20 images were used and an initial learning rate of 0.000015. The learning rate was halved after every 25 epochs to allow finer adjustments of the CNN weights. The performance is evaluated every 10 epochs on the validation set and the model achieving the best mean sensitivity score is saved.

At every evaluation step a confusion matrix was produced from the classification of the validation image set. From this matrix several metrics that allow to evaluate the efficiency of the classification were calculated. The accuracy, the sensitivity, the specificity and the AUC for each image class was produced in the output. In addition, the F1, the mean accuracy (average of all accuracies) and the balanced (weigted) accuracy were calculated. In most cases the results generated at each evaluation step were saved in a log file in order to allow the generation of plots showing the weighted accuracy (WACC) over an increasing number of training epochs.

The balanced (weighted) accuracy (WACC) is calculated as the sum of True Positives (TP) of each class divided by the number of images of the respcective class ($\sum_{i=1}^{n} \frac{TP_i}{N_i}$) (Grandini *et al.*, 2020). True Positives (TP_i) are correctly classified images for class (i) and N_i is the total number of images of the respective class (i). This metric is equivalent to the mean sensitivity. Since this metric gives an average of how likely images from each class will be classified correctly, we chose to use this metric to evaluate the performance of the models in our different experiments.

In all cases training was done using a Colab-Pro online account at Google Collaboratory whish allocates a single GPU of different type at each runtime session. With a Colab Pro account the GPU type allocated is either an NVIDIA Tesla T4 or Tesla P100.

4.6 K-means analysis

In order to subdivide the datasets into clusters (subgroups), in all cases we used a classic K-means algorithm (Elkan, 2003). The K-means algorithm aims to produce clusters of data points (image vectors) with the smallest inertia (*i.e.* the smallest intra-cluster distance between the data points).

For each iteration of the K-means algorithm, K (number of clusters selected) image vectors are randomly selected as cluster centroids and each member of the sample set is assigned to the closest cluster based on Euclidian distance from the centroid. New mean centroid values are then calculated based on all the data points of the same clusters and the clustering is repeated with these new centroid values. These last two steps (*i.e.* calculation of the new mean centroids and clustering based on these new mean centroids) are repeated until convergence (*i.e.* the centroids of the clusters do not change or the points remain assigned to the same clusters) or the maximum number (300) repetitions is reached.

The random selection of centroid values (K-means algorithm iteration) was repeated ten times in each K-means clustering experiment and the iteration producing the smallest final total inertia for all clusters was preserved. After the best K-means model was identified, the quality of the clustering was assessed with the Silhouette method (Kaufman et Rousseeuw, 2009).

The Silhouette clustering evaluation approach evaluates the quality of the clusters produced by not only taking into account the inertia (cohesion) of the clusters but also the separation between the different clusters. This method measures the cohesion (A) and the sieparation (B) of the clusters by measuring the average distance of each data point (image vector) between all other data points of the same cluster and their average distance between data points of the nearest cluster, respectively.

Silhouette coefficients are produced using those two measures by dividing the difference between A and B by the greater value between A and B (*i.e.* max(A,B)). By evaluating the coefficient of all points an average cluster coefficient is produced (also called the average score). The Silhouette average score produced ranges between -1 and 1, where a score near one indicates that the clusters are dense and well separated (*i.e.* a good quality clustering) and a score near zero indicates that the clusters are overlapping. In contrast a score below zero suggest that points in the clusters may be wrongly assigned.

Different approaches were used to generate input data for the K-means algorithm. First the images were resized to a 150x150 resolution and matrices representing the raw pixel values of all three color channels (RGB) were used to produce linear vectors that could be fed to the K-means algorithm. Resizing (downsampling) of the images was carried out with the Python pillow module using the LANCOS algorithm.

In a second approach normalized (scaled) image pixel vectors were produced by Z-score normalization of the matrix containing all raw image pixel vectors. The Z-scored vectors were also analyzed by K-means clustering.

As a third approach, the pixel vector array, either non-normalized or normalized by the Z-score or MinMax method, were analyzed by SVD or PCA dimension reduction algorithms to produce a small number of new representative features of the image vectors. Again, these reduced vectors were analyzed by K-means.

For each feature (pixel column), Z-score normalization is achieved by subtracting the mean feature value (m) from each feature value (x) and dividing the result by the standard deviation (s) (*i.e.* $Z = (x - m)/s$). The resulting normalized features have a mean of O and a standard deviation of 1.

MinMax normalization is achieved by transforming the feature values as values in the range between 0 and 1. This is done by subtracting the minimum feature value ($min(x)$) from each feature value (x) and dividing the result by the difference between the maximum value ($max(x)$) and the minimum value ($i.e.$ $scaledX = (x - min(x))/(max(x) - min(x))$). With this method the minimum value will be assigned a value of 0, the maximum value will be assigned a value of 1, and all other values will be transformed into intermediate values between 0 and 1.

Principal component analysis (PCA) and singular value decomposition (SVD) are related methods based on eigen values which were used to reduce the dimentionallity of vectors in order to produce a small number (2-3) of features which best represent the variability of the data.

Finally, in order to use potentially more informative data, the images were fed into a ResNet50 (RN50) (He *et al.*, 2016) CNN architecture pretrained on the ImageNet dataset to extract features from the images. The final layer of the CNN architecture was reduced in size by either average pooling or maximum pooling to generate flattened vectors of 2048 features. These non-normalized or normalized (minmax, z-score) vectors were further reduced in size by PCA or SVD to extract principal components before being used for K-means clustering.

CHAPTER V

RESULTS

5.1 analysis of the full ISIC dataset

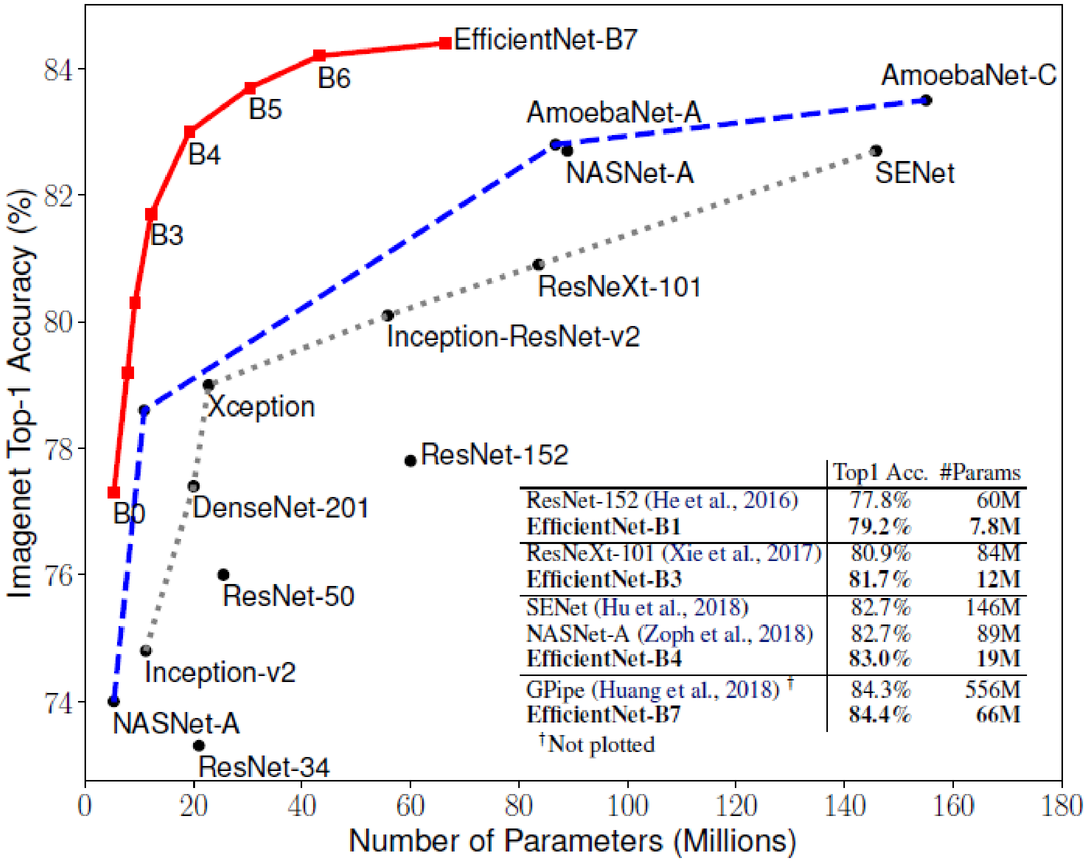
A similar strategy to that used by Gessert *et al.* (2020) was applied to train CNN models with the full ISIC2019 dataset. The group of Gessert *et al.* were the winners of the ISIC 2019 Skin Classification Challenge (Gessert *et al.*, 2020), which aimed to correctly classify the images from the dataset into nine classes corresponding to the assigned labels. Although the ISIC dataset represents only eight classes of labelled skin lesions, the aim of the challenge was to produce a model that could classify images into nine classes in such a way that additional test images, not corresponding to any of the eight official classes, would be classified as "other". In our study we solely aimed to get highly accurate classification into the eight actual classes.

Initially EfficientNet pretrained CNN models were used and assessed (Tan et Le, 2019). A series of EfficientNet models (B0 to B7), which take as input images of increasing sizes and use CNN architectures of increasing depth and width was developed at Google Brain and was found to produce state-of-the-art performance on the ImageNet dataset. The ImageNet dataset contains a collection of millions of images representing 1000 object classes and is commonly used to evaluate the performance of new CNN image classification architectures (Deng *et al.*, 2009)(Russakovsky *et al.*, 2015). The approach of optimizing three key aspects (image size, CNN layer width and CNN depth) of CNN architectures allowed them to produce models that are more efficient and outperform many other recent models (Figure 5.1).

The pretrained EfficientNet and other pretrained models presented below were all pretrained with the ImageNet dataset and then used by transfer learning to classify the ISIC2019 skin lesion images. Since EfficientNet models larger than B4 require significantly greater amounts of computing resources for diminishing amounts of performance (accuracy) improvement (Figure 5.1), only EfficientNet B0 and B4 modesl were used in our study. As shown in Figure 5.1, obtained from (Tan et Le, 2019), the EfficientNet models B0 to B4 show a much greater performance improvement for relatively small increase in resource requirement than the larger EfficientNet models.

By using an esemble of various deep learning models based largely, but not exclusively on the EfficientNet CNN architectures, Gessert *et al.* (2020) achieved an average balanced (weighed) accuracy (WACC) of 0.725 ±0.017 (standard deviation) from five-fold cross-validation results. Trials carried out with various individual models including EfficientNets, ResNext (Xie *et al.*, 2017) variants and SeNET (Hu *et al.*, 2018) produced best average WACCs between 0.653 ±0.008 and 0.688 ±0.007. With individual models, in Gessert *et al.*'s 2020 study the ResNext-WSL model showed the worst average WACC and the best was obtained with the largest EfficientNet model (B6) tested.

In all our classification experiments, in a similar fashion to Gessert *et al.* (2020). the performance of the individual models tested were evaluated by five-fold cross-validation. The hyperparamaters used, as described in the Materials and Methods section, were also the same as those used in Gessert *et al.* (2020). In a prior study



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.2%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.7%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**83.0%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

Figure 5.1: Model size vs. ImageNet accuracy
The figure shows the Top 1 accuracy (accuracy at predicting the correct object) achieved by the various EfficientNet models (B0-B7) and various popular models with the ImageNet dataset plotted against the model size represented as the number of parameters. Since the number of parameters can be interpreted as a computational cost value, the figure shows that the EfficientNet models outperform all other models of comparable size tested. Particular attention can be placed on the ResNet-50, the DenseNet-201 and RexNeXt-101 models as variants of these models were used in this study. In addition, as illustrated, a smaller gain in accuracy for a significantly greater number of parameters is achieved for EfficientNet models larger than the B4 version. This figure was obtained from (Tan et Le, 2019)

Table 5.1: Classification results of the ISIC2019 dataset and its sub-datasets into eight classes.
The best mean WACC and the standard deviation achieved with various models with the full ISIC2019 dataset when classifying the images in all eight labeled disease classes. The results of EfficientNet models with the individual BCN and HAM sub-datasets are also shown as well as those obtained with the combined BCN and HAM datasets. For all experiments the best average WACC along with the standard deviation was calculated from five-fold cross-validation. For all experiments the number of images (#Imgs), the model used (CNN), the number of epochs (#Epochs) of training completed for each cross-validation and the mean best WACC (AvgWACC) ± the standard deviation are shown.

Datasets	#Imgs	CNN	#Epochs	AvgWACC ±Std
ISIC2019	25331	EfficientNetB0	60	0,6193 ±0.0245
ISIC2019	25331	EfficientNetB4	100	0.6632 ±0.0108
ISIC2019	25331	SE-ResNext50	100	0.6156 ±0.0189
ISIC2019	25331	DenseNet121	100	0.6309 ±0.0053
ISIC2019	25331	ResNext101	100	0.6268 ±0.0116
ISIC2019	25331	InceptionV3	100	0.6573 ±0.0062
ISIC2019	25331	InceptionV3(Aux)	100	0.6600 ±0.0119
HAM	10015	EfficientNetB0	60	0.7599 ±0.0202
HAM	10015	EfficientNetB4	100	0.8346 ±0.0129
BCN	12413	EfficientNetB0	60	0.6823 ±0.0194
BCN	12413	EfficientNetB4	100	0.8933 ±0.0096
BCN+HAM	22428	EfficientNetB4	100	0.8747 ±0.0079

by Gessert *et al.* (Gessert *et al.*, 2018), extensive work was done to determine the best parameters to use for transfer learning of pretrained models for the classification of skin lesion images and therfore we chose to use the same parameters in order to be able to reproduce their results.

As seen in Table 5.1, by implementing EfficientNets to classify the full ISIC2019 dataset, as expected, we obtained very similar results to those of Gessert *et al.* (2020). With the smallest version of EfficientNet (B0) we obtained a best average WACC of 0.6193 ±0.0245, whereas with the larger B4 model we obtained a significantly improved best average WACC of 0.6632 ±0.0108. As seen in Figure 5.2, the best WACC with EfficientNet B4 was reached after only 20 epochs of training. This indicates that fine-tunning of the EfficientNet model is achieved rapidly when a large number (0.8*25531=20425) images are used for training.

In comparison, with the same models, Gessert *et al.* (2020) obtained best average WACCs of 0.658 ±0.017 (B0) and 0.678 ± 0.011 (B4). Likely, the superior results obtained by Gessert *et al.* are due to the fact that the dataset they used was supplemented with additional datasets such as the SevenPoint dataset (Kawahara *et al.*, 2018), which contains 4.11 images, and another in-house dataset of unknown size.

Based on Figure 5.1, EfficientNets appear to currently (at the time of writing) be the most efficient and best performing models based on ImageNet classification accuracy result. Nonetheless, we aimed to test a few more recent and popular pretrained models to compare their performance at classifying skin lesion images by transfer learning and see if possibly better results can be achieved.

We chose to test two models based on the residual networks (ResNet) architecture, SE-ResNet50 (Hu *et al.*, 2018) and ResNext101 (Xie *et al.*, 2017). The SE-ResNet50 model is similar to the original ResNet50 model (He *et al.*, 2016) which revolutionized the use of deep CNNs for image classification and other various tasks. However, in a similar-fashion to the EfficientNet models, a squeeze-and-excitation block was added to the convolution blocks to optimize the channels (feature maps) produced at each convolution step. As seen in Table 5.1, this model achieved a best average WACC of 0.6156 ±0.0189 with the ISIC2019 dataset. In comparison, the ResNext101 uses a different convolution layer design where multiple identical branches of convolution are used at each convolution step to produce several (32) feature maps that are afterwards concatenated. This model

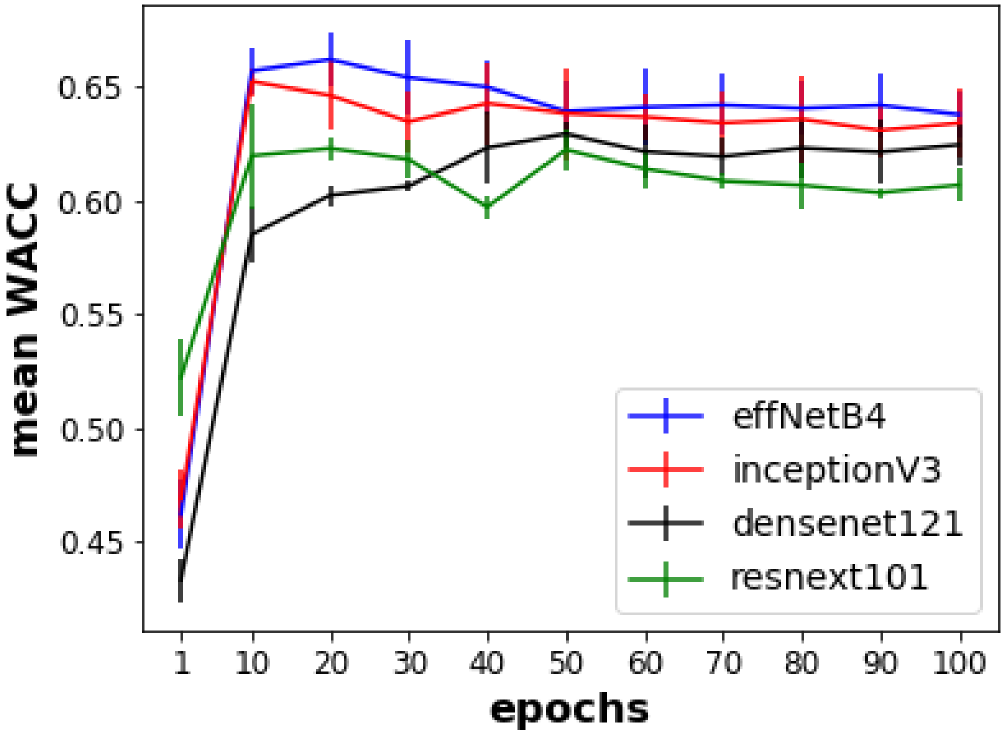


Figure 5.2: Mean WACC with the ISIC2019 dataset after increasing numbers of training epochs.
The mean WACC obtained at the first and every ten epochs for classification of the ISIC2019 dataset into the eight labelled disease classes with various CNN models were calulated on the validation set from cross-validation experiments. For EfficientNet B4 (blue) and InceptionV3 ignoring the auxiliary classifier (red), the mean WACCs were calculated from five-fold cross-validation. For DenseNet121 (green) and ResNext101 (black), the mean WACCs were calculated from three and four-fold cross-validation, respectively. The reduced number of experiments used for densenet121 and resnext101 was due to errors in the generation of the log files required to produce the data. The error bars for each mean WACC value represent the standard deviation.

achieved a slightly improved best average WACC of 0.6268 ±0.0116.

The other two models used, DenseNet121 (Huang *et al.*, 2017) and InceptionV3 (Szegedy *et al.*, 2016) have their own particularities in terms of their CNN architecture design (see Materials and Methods), but have all achieved high accuracy results with the ImageNet dataset. DenseNet121 achieved a best average WACC of 0.6309 ±0.0053 with the ISIC2019 dataset. In the case of InceptionV3, which normally utilizes an auxiliary classifier to regularize the main classifier during training, two training strategies were used. In one case the error obtained with the auxiliary classifier was taken into account and compounded with the error of the main classifier during training, whereas in the second case, the error obtained with the auxiliary classifier was ignored. As can again be seen in Table 5.1, the model where the auxiliary classifier was taken into account (InceptionV3(Aux)) achieved a slightly better best average WACC (0.66 ±0.0119) than the model where the auxiliary classifier is ignored (InceptionV3, 0.6573 ±0.0062).

Again, the increase in average WACC achieved by ResNext101, DenseNet121 and the InceptionV3 model (ignoring the auxiliary classifier) during fine tunning for 100 epochs is shown in Figure 5.2. Like EfficientNet B4, in general these models achieved the best average WACC after a low number (10-20) of epochs. DenseNet121 which uses a large number of inter-connections between its convolution layers produced a smoother curve and reached its peak slightly later after 50 epochs. In general, we can see in Figure 5.2 that, for all the models, after the average WACC peak is reached, a plateau is formed where the its value slightly fluctuates and, in some cases, decreases. However, no severe continuous decrease in the average WACC is observed suggesting that the models to not suffer from strong overfitting.

Based on the results achieved (Table 5.1) we can see that the different models achieve broadly similar best average WACC classification results with the full ISIC2019 image dataset. However, we can see that EfficientNet B4 achieved the best average WACC and is followed closely by the InceptionV3 model where the auxiliary classifier was taken into account. Actually, if we take the standard deviation associated with the best average WACCs achieved by these two models, we cannot say that one model performs significantly better than the other. Nonetheless, for the remainder of our study we chose to use the EfficientNet B4 model which is very efficient and has a relatively low computational cost.

5.2 seperation of the datasets

As mentioned in the introduction, several studies related to skin cancer image classification have been performed in the recent past. Some studies linked to the ISIC2018 skin challenge competition, where the HAM dataset was used for the classification of images into seven classes, are of particular interest. Classification accuracy (WACC) obtained with this limited dataset surpassed the results obtained with the full ISIC2019 dataset. For example, with a similar approach as that described in the previous subsection, the group of Gessert *et al.* (2018), who placed second in this challenge, obtained an average sensitivity (same as WACC) score of 0.795 with the densenet4.1 model and 0.808 with ResNext101.

As also previously mentioned in the introduction, later-on the group of Bakkouri *et al.* (2019) were able to improve these results significantly by using a more complex model and obtained an average sensitivity of 0.934 with the HAM dataset. Here they selected only the later CNN layers of different models (*i.e.* VGG16, ResNet50 and densenet4.1) for fine tunning and used a CNN layer fusion approach with their different models working in parallel to achieve high accuracy classification.

These results strongly suggest that smaller datasets that may contain more homogenous images (such as the HAM dataset) can be more efficiently classified than larger datasets produced by combining different datasets (such as the ISIC2019 dataset). Most likely, the source of the images making up the datasets and the preprocessing strategy applied by the dataset curators have an important effect on classfication accuracy. It is likely that, when several datasets are pooled together to produce a greater sample size, the classification models have a greater difficulty to correctly separate the different skin lesion classes if the images from these classes appear slightly different from one dataset to another.

In order to determine if this is generally true, we evaluated the performance of the EfficientNet models in classifying the two largest subsets of the ISIC2019 dataset (*i.e.* the HAM and the BCN datasets) separately. However, in this case, since the datasets were annotated to produce eight skin lesion classes, we aimed to classify the images into these eigth classes. This is in contrast to the previously described studies where only seven classes were defined.

With the HAM dataset, using the same model configurations than those used with the full ISIC2019 dataset, with the EfficientNet B0 model we obtained an average WACC of 0.7599 ±0.0202, which is significantly greater than that achieved with the full ISIC2019 dataset (Table 5.1). Furthermore, we obtained an even greater average WACC of 0.8346 ±0.0129 with the larger EfficientNet B4 model. This result is better than that obtained with Densenet4.1, the best performing individual model used in Gessert *et al.*'s 2018 study. This further demonstrates that the newer, more recent EfficientNet models (such as B4) can surpass the efficiency of the older models. This is also particularly noteworthy since in this experiment the classification task was more complex than those in Gessert *et al.* (2018) due to the extra image class added.

Significantly, even better classification accuracy results were obtained with the

BCN dataset (Table 5.1). With this slightly larger dataset the EfficientNet B4 model produced an average classification WACC of 0.8933 ±0.0096, which is strikingly greater than that observed with the HAM dataset. This is very surprising since, as mentioned before, this dataset is said to include images of difficultly classified lesions due to their location in uncommon body areas. One would believe that these images could have made the classification task more complex and less accurate with this dataset. This may also be why the smaller less complex EfficientNet B0 model, in contrast to the B4 model, performed less well with the BCN dataset than with the HAM dataset and only achieved a WACC of 0.6823 ±0.0194.

As could be expected, an increased number of epochs were required to reach the best average WACC with these individual datasets due to the reduced number of images (Figure 5.3). In these cases, the best average WACC were obtained after approximately 60 or 70 epochs. However, it is significant that the best average WACC results obtained with these two separate datasets are clearly higher than those obtained with the full ISIC2019 dataset. To determine if such results only occur if these two datasets are separated, we performed a classification experiment with the HAM and BCN datasets combined. In this case EfficientNet B4 achieved a best average WACC of 0.8747 ±0.0079 (Table 5.1).

Although the WACC observed with the combined HAM and BCN datasets is slightly lower than that achieved with the BCN dataset independently, the high classification accuracy suggests that the images in these two datasets are similar. Even if the HAM and BCN datasets have likely been obtained and preprocessed differently, the images representing each lesion class in these datasets most likely share very similar traits and make accurate classification feasible. Importantly, the fact that the WACC obtained with the combined HAM and BCN datasets is

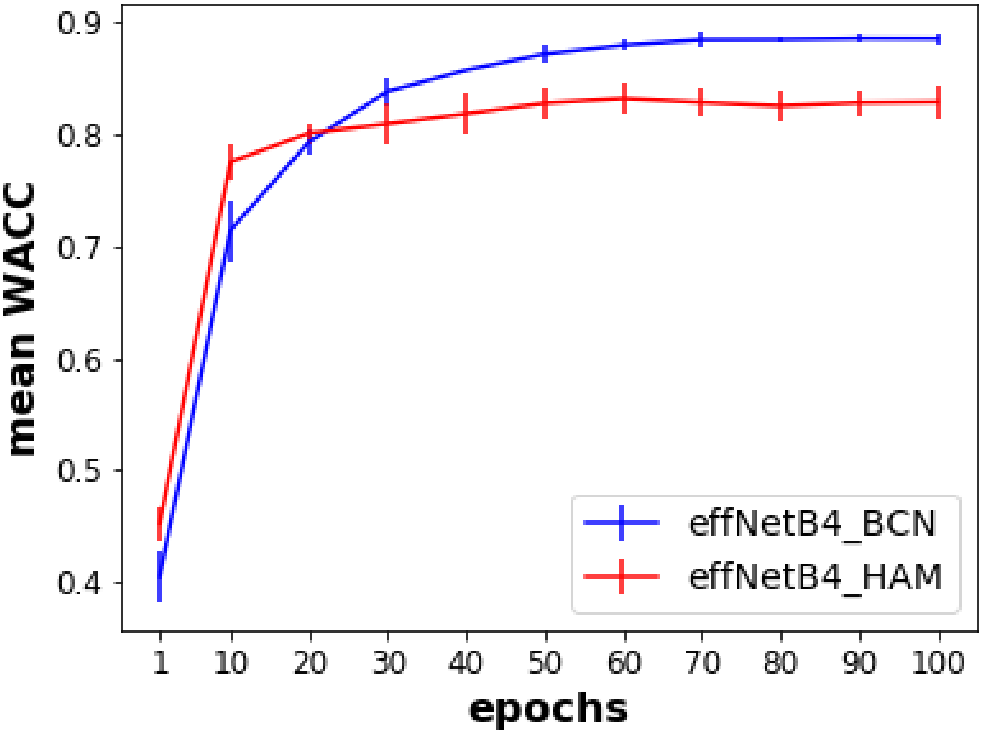


Figure 5.3: Mean WACC with the BCN and HAM datasets after increasing numbers of training epochs.
The mean WACC obtained at the first and every ten epochs for classification of the ISIC2019 sub-datasets (BCN and HAM) into the eight labelled image classes with the EfficientNet B4 model were calculated on the validation sets from cross-validation experiments. The results for the BCN dataset (blue) and the HAM dataset (red) were calculated from three-fold and five-fold cross-validation experiments, respectively. The error bars for each mean WACC represent the standard deviation.

much higher that that obtained with the full ISIC2019 dataset indicates that the presence of the MSK dataset in the latter is responsible for making classification more challenging and less efficient.

It seems that the MSK dataset, the smallest dataset, which contains images of variable sizes, produces noise in the training of the models and makes classification more difficult. It is interesting that, unlike the HAM and BCN datasets, the MSK dataset only represents three of the eight lesion classes present in the ISIC2019 dataset (Table 4.1). Also, it is noteworthy that unlike any images from the HAM or BCN dataset, a large proportion of the MSK images are labeled as being downsampled. However, no details are provided as to how and why these images were downsampled. It seems likely that the images making up this dataset are, in some way, significantly different than those making up the HAM or BCN datasets.

Most liekly some inherant differences between the MSK images and those of the other datasets make the classification task more challenging in the full ISIC2019 dataset.

5.3 unsupervised classification of images

We aimed to find a strategy that would allow us to separate the images according to possible unclear, inherent differences and may allow to improve classification by the CNN models. To achieve this, we initially used the K-means unsupervised classification algorithm (Elkan, 2003) to separate the BCN dataset into subgroups (clusters) of images with similar characteristics. We chose to use this dataset since it contains a more even proportion of images in all of the eight labeled classes.

As a first approach, we produced linear vectors representing all pixel values of each BCN image. In order to produce manageable and comparable vectors, all the images of the BCN dataset were first resized (downsampled) to a 150x150 resolution. The produced linear vectors contained three values for each pixel position representing the three colour channels of the images: red, green and blue.

The resulting large vectors, each containing 67,500 features (pixel values), for each image were fed into the K-means algorithm to produce different number of subgroups. The K-means algorithm (Elkan, 2003) was used to separate the images into two to five clusters (K=2-5) and the image clusters generated were evaluated using the Silhouette method (Kaufman et Rousseeuw, 2009). The K-means algorithm tries to produce clusters with the smallest possible inertia (*i.e.* smallest intra-cluster distance between members of the cluster).

As an initial proof of concept experiment, we selected a small number of images (10) with clearly different visual appearances (five MEL and five VASC). K-means was found to clearly separate these images based on their corresponding raw pixel vectors (data not shown).

Ideally, we would also have liked to show that K-means can also distinguish between images from the MSK dataset and those from the other datasets since it was found that, as mentioned previously, the MSK images may be somehow different than those making up the other datasets. For this purpose, we carried out clustering experiments with a pixel vector dataset (EqualDistImgs) representing all the MSK images and two equal-sized subsets of images from the BCN and HAM datasets (Table 5.2). In this case the BCN and HAM image subsets used contained the same disease class (MEL, NV and BKL) image proportions as those in the MSK dataset. In addition, we further carried out experiments with smaller sub-datasets in which images were separated using two approaches: 1) based on the two most abundant disease classes (MEL and NV) and 2) based on the la-

Table 5.2: K-means clustering results with the MSK image dataset and subsets of the BCN and HAM datasets.
Z-scored (scaledImgVector) and non-normalized (rawImgVector) pixel vectors (Pixel based, left side) were generated with images of the EqualDistImgs dataset, which includes all the MSK images available (2903) and equal-sized subsets of the BCN and HAM datasets. These latter subsets contained the same proportions of each disease class (NV, MEL and BKL) as those found in the MSK dataset. Non-normalized (avgNoNormSVD) and MinMax-normalized global average pooled ResNet50 (RN50) feature vectors (RN50 features, righ side) were also produced with the images of the EqualDistImgs and further reduced to two features by SVD. All vector matrices produced were analyzed by K-means (K=2 or K=3) and the number (#MSK) and proportion (%MSK) of the MSK images in each cluster is presented. The number of images (#Imgs) in the original dataset and in all clusters produced are indicated.

Pixel based				RN50 features			
Dataset	#imgs	#MSK	%MSK	Dataset	#imgs	#MSK	%MSK
equalDistDataset	8709	2903	0.3333	equalDistDataset	8709	2903	0.3333
rawImgVector				avgNoNormSVD			
K-2_0	5239	1688	0.3222	K-2_0	5869	2063	0.4026
K-2_1	3470	4.15	0.3501	K-2_1	2840	840	0.2958
K-3_0	3747	1301	0.3472	K-3_0	3818	1865	0.4885
K-3_1	4.14	72	0.0646	K-3_1	1586	212	0.1337
K-3_2	3848	1530	0.3976	K-3_2	3305	826	0.2499
scaledImgVector				avgMinMaxSVD			
K-2_0	4217	1532	0.3633	K-2_0	2407	760	0.3157
K-2_1	4492	1371	0.3052	K-2_1	6303	2143	0.3400
K3_0	950	56	0.0611	K-3_0	2891	1576	0.5591
K-3_1	3996	1341	0.3356	K-3_1	1570	230	0.1465
K-3_2	3763	1506	0.4002	K-3_2	4248	1097	0.2582

belling MSK images as downsampled and those with no such labelling (Table 5.3).

However, in all the above cases, attempts to separate MSK image pixel vectors from those produced with HAM or BCN images with K-means failed. As seen in Table 5.2 and Table 5.3, these experiments did not even produce clusters where the MSK images were predominant over the others.

Since it is possible that the noise produced by the MSK images in the classification

Table 5.3: K-means clustering of pixel vectors produced with the NV and MEL image datasets including BCN, HAM and MSK images.
Z-scored (scaledImgVector) and non-normalized (rawImgVector) pixel vectors were generated with images from the three parent datasets: BCN, HAM and MSK. The vector matrices produced were analyzed by K-means (K=2 or K=3) and the number (#MSK) and proportion (%MSK) of the MSK images in each cluster are presented. The number of images (#Imgs) in all original datasets and in all clusters produced are also indicated. The EqualDownNV and EqualDownMEL datasets contain all the NV (4.16) or MEL (374) images from the MSK dataset labeled as downsampled, respectively. The EqualOriNV and EqualOriMEL datasets contain all the MSK images of the NV (648) or MEL (178) images from the MSK which are not labeled as downsampled, respectively. All of the EqualDown and EqualOri datasets contain an identical number of images of the same class from each of the BCN and HAM datasets.

Dataset	#Imgs	#MSK	%MSK	Dataset	#Imgs	#MSK	%MSK
equalDownNV	3948	4.16	0.3333	**EqualDownMEL**	5.12	374	0.3333
rawImgVector				rawImgVector			
K-2_0	2428	712	0.2932	K-2_0	206	10	0,0485
K-2_1	5.10	604	0.3974	K-2_1	916	364	0.3974
K-3_0	5.16	705	0.4346	K-3_0	434	138	0.4383
K-3_1	4.17	597	0.3114	K-3_1	487	227	0.0446
K-3_2	405	14	0.0346	K-3_2	201	9	0.3180
scaledImgVector				scaledImgVector			
K-2_0	1747	716	0.4098	K-2_0	536	139	0.2481
K-2_1	2201	608	0.2762	K-2_1	586	235	0.4010
K-3_0	2000	591	0.2955	K-3_0	447	140	0.4315
K-3_1	4.10	718	0.4460	K-3_1	186	9	0.0484
K-3_2	338	7	0.0207	K-3_2	489	225	0.3132
equalOriNV	1944	648	0.3330	**EqualOriMEL**	534	178	0.3333
rawImgVector				rawImgVector			
K-2_0	568	179	0.34.1	K-2_0	405	159	0.3926
K-2_1	1376	470	0.3416	K-2_1	129	19	0.1473
K-3_0	948	283	0.2985	K-3_0	114	11	0.0965
K-3_1	267	45	0.1685	K-3_1	211	93	0.4408
K-3_2	729	520	0.7133	K-3_2	209	74	0.3541
scaledImgVector				scaledImgVector			
K-2_0	908	347	0.3922	K-2_0	255	78	0.3059
K-2_1	1036	301	0.2905	K-2_1	279	100	0.3584
K-3_0	221	32	0.1448	K-3_0	218	95	0.4358
K-3_1	959	293	0.3055	K-3_1	214	73	0.3411
K-3_2	564	323	0.5727	K-3_2	102	10	0.0980

Table 5.4: Silhouette scores obtained for different K-means BCN image clusters based on pixel vectors// Silhouette average scores for K-means clustering into K clusters produced with BCN image pixel vectors. These clusters include those produced with the raw image vectors (rawImg), the Z-sored normalized image vectors (scaledImg) and those produced with the pixel vectors normalized by MinMax (MinMaxSVD) or non-normalized (NoNormSVD) prior to being reduced to two features by SVD. The vectors shown in bold are those which produced the clusters with the best Silhouette scores (for 2 clusters (K=2) in all cases) and were used for analysis in our study. As shown, many permutations of normalization methods (Norm), dimension reductions methods (Red) and number of dimensions (Dim) (features) were tested to obtain the best clustering possible.

Vectors	Norm	Red	Dim	K=2	K=3	K=4	K=5
rawImg	NA	NA	NA	0.3179	0.2354	0.1683	0.1545
scaledImg	Zscore	NA	NA	0.1762	0.1747	0.1502	0.1405
MinMaxSVD	MinMax	SVD	2	0.5194	0.4515	0.3953	0.3764
MinMaxSVDtest2	MinMax	SVD	3	0.4680	0.3865	0.3213	0.3188
MinMaxPCAtest1	MinMax	PCA	2	0.5184	0.4510	0.3961	0.3724
MinMaxPCAtest2	MinMax	PCA	3	0.4675	0.3859	0.3206	0.3185
NoNormSVD	NA	SVD	2	0.5227	0.4540	0.3966	0.3761
NoNormSVDtest2	NA	SVD	3	0.4721	0.3903	0.3264	0.3199
NoNormPCAtest1	NA	PCA	2	0.5214	0.4535	0.4000	0.3700
NoNormZscoreSVDtest1	Zscore	SVD	2	0.3437	0.3742	0.3748	0.3686
NoNormZscorePCAtest1	Zscore	PCA	2	0.3437	0.3742	0.3748	0.3686

of the full ISIC2019 dataset is only caused by a subset of the MSK images, we chose to proceed with our strategy and see if we can achieve better classification with image clusters generated with the BCN dataset.

As shown in Table 5.4, the average Silhouette scores for our initial experiment ranged between 0.32 and 0.15 indicating that the clustering is rather poor. The greater value (0.32) was achieved with two clusters (*i.e.* K=2). Due to the large dimensionality of the pixel vectors used to produce the clustering, it is impossible to produce a visualization of the clusters produced, however the distribution of the Silhouette coefficients for the data points (pixel vectors) of each cluster is shown in Figure 5.4.

In all cases, the fact that the coefficient for a large majority of points sits well below the average score is a visual indicator of poor clustering. As a second approach, we used Z-score normalization to produce a normalized array of pixel vectors in which all features (columns) were scaled. After feeding this normalized array to the K-means algorithm and again evaluating the clustering results by Silhouette, we saw that the clustering is even worst than that obtained without normalization. As seen is Table 5.4 and Figure 5.5, after normalization the average Silhouette scores range between 0.18 and 0.14. In this case also, the highest score was obtained for two clusters.

Although, the clustering results obtained were poor, we examined the distribution of the images for each cluster. As can be seen in Table 4.1, with both approaches (raw (rawImg) and normalized (scaledImg) vectors) the produced clusters were uneven and contained different number of images. With the raw image vectors, clusters of 7178 and 5235 images were produced, and with the normalized vectors, the clusters contained 6433 and 5980 images. As can be seen in Figure 5.6A and 5.6B, the presented Venn diagrams indicate that the images making up the clusters generated with the raw image vectors and those produced with the normalized (scaled) image vectors are very different.

As can also be seen in Table 4.1, the distribution of the images amongst the different image classes, particularly the ones generated with the raw image vectors, is somewhat different from the distribution in the full BCN dataset. However, all clusters still contain images assigned to each disease class. Therefore, the clustering does not seem to separate the images according to their class, but rather, more likely, as desired, based on their general overall pixel composition. To see if the separation produced can improve classification results, datasets were produced for both clusters produced by K-means with each array of vectors (original and

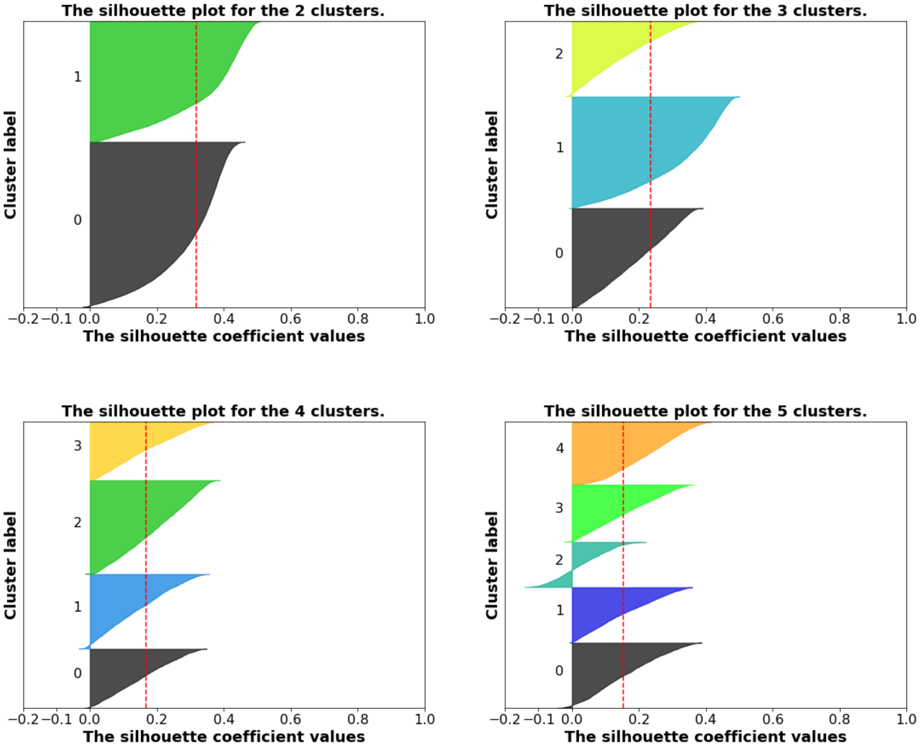


Figure 5.4: Silhouette coefficient distributions for K-means clustering of non-normalized BCN image pixel vectors.
Silhouette method evaluation of the K-means clusters produced with the non-normalized BCN image pixel vectors. The different panels illustrate the Silhouette coefficients with K=2 (top-left), K=3 (top-right), K=4 (bottom-left) and K=5 (bottom-right) where the value of K represents the number of clusters specified to the K-means algorithm. In each panel the area containing the Silhouette coefficients for all the image pixel vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line.

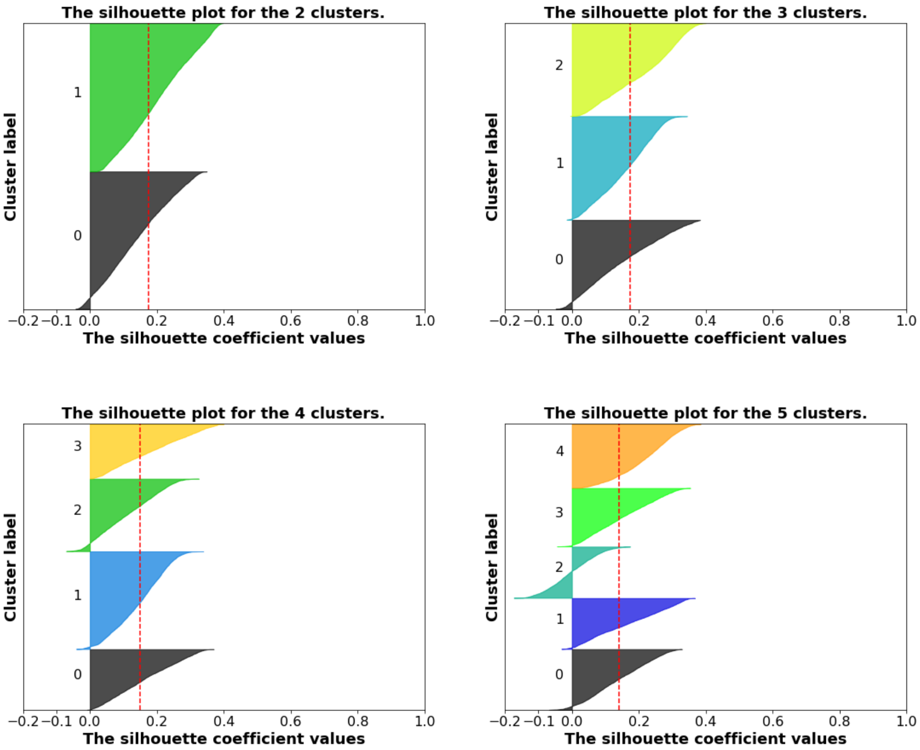


Figure 5.5: Silhouette coefficient distributions for K-means clustering of Z-score normalized BCN image pixel vectors.
Silhouette method evaluation of the K-means clusters produced with the Z-scored normalized BCN image pixel vectors. The different panels illustrate the Silhouette coefficients with K=2 (top-left), K=3 (top-right), K=4 (bottom-left) and K=5 (bottom-right) where the value of K represents the number of clusters specified to the K-means algorithm. In each panel the area containing the Silhouette coefficients for all the image pixel vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line.

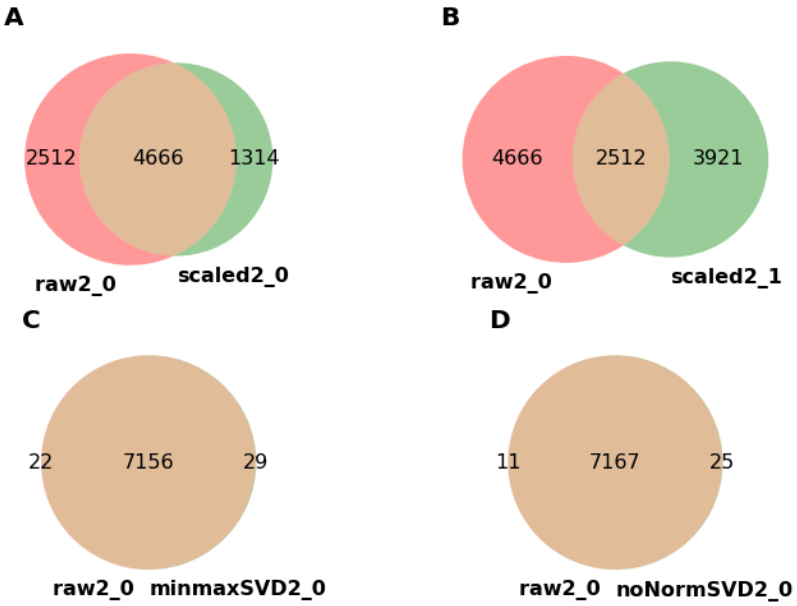


Figure 5.6: Overlap between K-means image clusters produced from pixel vectors. Venn diagrams showing the overlap between pairs of datasets resulting from the K-means clustering (K=2) of the BCN images based on pixel vectors. The first cluster produced from clustering with the non-normalized image vectors (raw2_0) is compared to the first (A) and second (B) cluster produced with the Z-score normalized image pixel vectors, scaled2_0 and scaled2_1, respectively. In C and D, the raw_0 cluster is compared to one cluster produced with MinMax normalized pixel vectors and one cluster produced with the non-normalized pixel vectors prior to being reduced to two features by SVD, minmaxSVD2_0 and noNormSVD2_0.

normalized).

The classification results with the EfficientNet B4 model, shown in Table 5.5, produced WACC results of 0.8164 ±0.0037 (with 7178 images) and 0.7840 ±0.0074 (with 5235 images) with the original vector clusters. The classification results with image clusters produced with the normalized vectors were slightly better and produced WACC results of 0.8259 ±0.0074 (with 5980 images) and 0.8246 ±0.0081 (with 6433 images). Overall, these results are significantly lower than that obtained with the full BCN dataset (WACC 0.8933 ±0.0096), but this may be due in part to the lower number of images used to train the model.

In order to compare the classification efficiency when similar number of images are used, two 6199 image datasets were produced with randomly selected images from each disease class and with a class distribution equal to that of the original BCN dataset (random6200_1 and random6200_2). As seen in Figure 5.7, the two randomly produced datasets are quite different from one another. Classification of these datasets with the EfficientNet B4 model produced best average WACCs of 0.7615 ±0.0130 and 0.8012 ±0.0037 over five cross-validations (Table 5.5). This suggests that indeed the lower classification WACC observed with the clusters is in large part caused by the lower number of images available for training. In addition, these results suggest that with some clusters, those produced with the normalized image vectors, a slightly higher classification accuracy can be obtained than that observed with the random datasets. This therefore suggests that some useful separation of the images is achieved.

It is noteworthy to mention that in these cases where classification was performed on reduced number of images, best average WACC results in each training experiment were reached after a similar number of epochs to that necessary for the full

Table 5.5: Classification results with K-means BCN image pixel vector clusters into eight disease classes.
The mean best WACC and standard deviation achieved with the EfficientNet B4 model with various datasets when classifying the images in all eigth labelled disease classes. The classification results of the full BCN dataset (BCN) previously presented in Table 5.1 is shown for reference purposes. The clusters produced from BCN image pixel vectors include those produced with the raw image vectors (rawImg), the Z-sored normalized image vectors (scaledImg) and those produced wiht the pixel vectors normalized by MinMax and reduced to two features by SVD (MinMaxSVD). For all experiments the K-means cluster (Cluster), the number of images (#Imgs), the number of cross-validations performed (#CVs), the number of epochs (#Epochs) of training completed for each cross-validation and the mean best WACC ± the standard deviation (AvgWACC ±Std) is shown. In addition, the results obtained with the two randomly produced datasets (random6200_1 and random6200_2) are shown for comparision purposes.

DataSet	Cluster	#Imgs	#CVs	#Epochs	AvgWACC ±Std
BCN	NA	12413	5	100	0.8933 ±0.0096
BCN K-means (K=2)					
rawImg	2_0	7178	5	60	0.8164 ±0.0037
	2_1	5235	3	100	0.7842 ±0.0074
scaledImg	2_0	5980	3	100	0.8259 ±0.0074
	2_1	6433	3	100	0.8246 ±0.0081
MinMaxSVD	2_0	7185	5	100	0.8204 ±0.0090
	2_1	5228	5	100	0.7842 ±0.0071
BCN random					
random6200_1	NA	6199	5	100	0.7615 ±0.0130
random6200_2	NA	6199	5	100	0.8012 ±0.0037

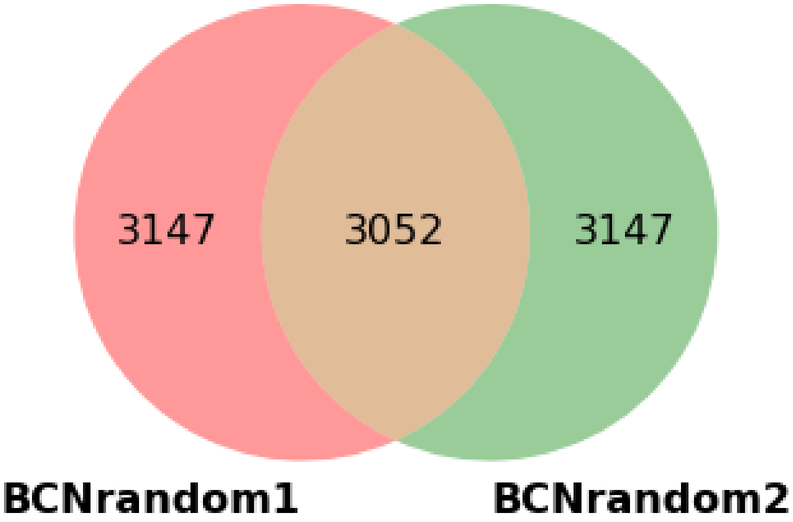


Figure 5.7: Overlap between randomly produced datasets of BCN images.
Venn diagrams showing the overlap between the two randomly procuced BCN image datasets, BCNrandom6200_1 (BCNrandom_1) and BCNrandom6200_2 (BCNrandom_2). Each dataset contains 6199 images in the same proportions for all eight disease classes as those in the full BCN dataset. As depicted, slightly less than half of the images (3042) are present in both datasets.

BCN dataset. As presented in Figure 5.8 we can see from the blue and red curves produced with the random BCN datasets that the WACC increases gradually, in a similar fashion than that observed with the full BCN dataset (Figure 5.3). Generally, the best average WACC was seen after 60 to 100 epochs. Although in some cases (not shown) the best WACC was obtained after the maximal number of epochs (100) performed, a plateau of average WACC was reached after approximately 60 epochs and any increase in WACC after that was minimal.

To determine if the large number of features (67,500) used to produce the clusters in each previous case may impede the performance of the K-means algorithm and limit the quality of the clustering, we used different dimension reduction algorithms to reduce the number of features. We aimed to extract the most representative and important components from the image pixel vector matrices with the singular value decomposition (SVD) and the principal component analysis (PCA) algorithms. In all cases we aimed to obtained new features that best represented the variability in the raw or normalized image pixel vectors.

We tested various approaches where the raw (original) pixel vector array was either not normalized or normalized with the standard-scaler (Z-score) approach or the MinMax approach. The resulting arrays were reduced by SVD or PCA to extract two or three principal components. The resulting reduced vectors for each image were analyzed by K-means clustering in two to five clusters and the clustering was evaluated by Silhouette (Table 5.4).

The best Silhouette scores were obtained when the vectors were not normalized and reduced to two components with SVD (Silhouette scores between 0.5227 to 0.3764 for K=2 to 5 (Table 5.4 and Figures 5.9 and 5.10)) or normalized with the MinMax method and again reduced to two components with SVD (Silhouette scores 0.5194 to 0.3761 with K=2 to 5 (Table 5.4 and Figures 5.11 and 5.12)).

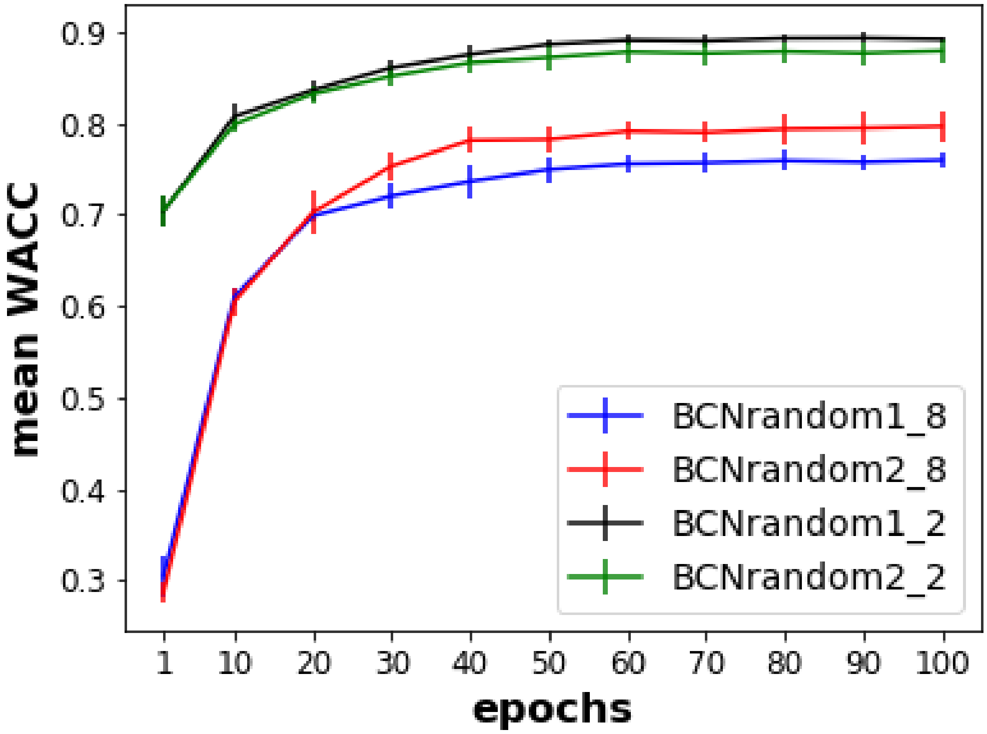


Figure 5.8: Mean WACC with the BCN random datasets after increasing numbers of training epochs.
The mean WACC obtained at the first and every ten epochs for the BCNrandom6200 image datasets with the EfficientNet B4 model were calculated on the validation sets from five-fold cross-validation experiments. The random datasets (BCNrandom1 and BCNrandom2) each contain 6199 images selected randomly from the BCN dataset in such a way that these smaller datasets have the same proportion of images from each disease class as the full BCN dataset. The blue and red lines show the mean WACC obtained from classification of the random datasets (1 and 2, respectively) in the eight labelled disease classes (8). The green and black lines show the mean WACC obtained from classification of the random dataset images (1 and 2, respectively) in two classes (2): MEL or Others. The error bars on for each mean WACC value represent the standard deviation.

In each case, the best Silhouette scores were obtained when the image vectors were separated into two clusters (*i.e.* K=2). As can be seen in Table 5.4, these Silhouette scores are considerably higher than those obtained with the full vectors (without dimension reduction), suggesting that the clusters are more tight and better separated.

In these cases, since only two features (components) were used to cluster the images, we can visualize the clusters on a two-dimensional plot as shown in Figures 5.9 to 5.12. As can be seen in the two cluster images (Figures 5.9 and 5.11, top-right panels) two dense clusters are produced with a large number of data points near the intersection. These latter points must contribute to the relatively low Silhouette score (0.5 out of a maximum of 1). As can also be seen in the other clustering images (K-3 to K-5) in Figures 5.9 to 5.12, using a higher value of K produces clusters that are clearly not well separated.

Again, the image clusters produced with each approach were uneven and had a slightly different distribution of images amongst the different classes than that of the full BCN dataset (Table 4.1). However, both approaches produced clusters with very similar numbers of images (75.1 vs 7185) and (5221 vs 5228) for the rawSVD (NoNormSVD) and MinMaxSVD vectors, respectively, and very similar distributions (Table 4.1). Moreover, the size of the clusters produced were almost identical to those obtained with the full raw (rawImg) non-normalized pixel vectors (7179 and 5235).

As shown in the Venn diagrams in Figure 5.6C and 5.6D, the clusters produced with these reduced vectors (minmaxSVD2_0 and noNormSVD2_0) were almost identical to the clusters produced with the raw original vectors (raw2_0). In fact, more than 99% of the images contained in the larger rawSVD (99.65%) and

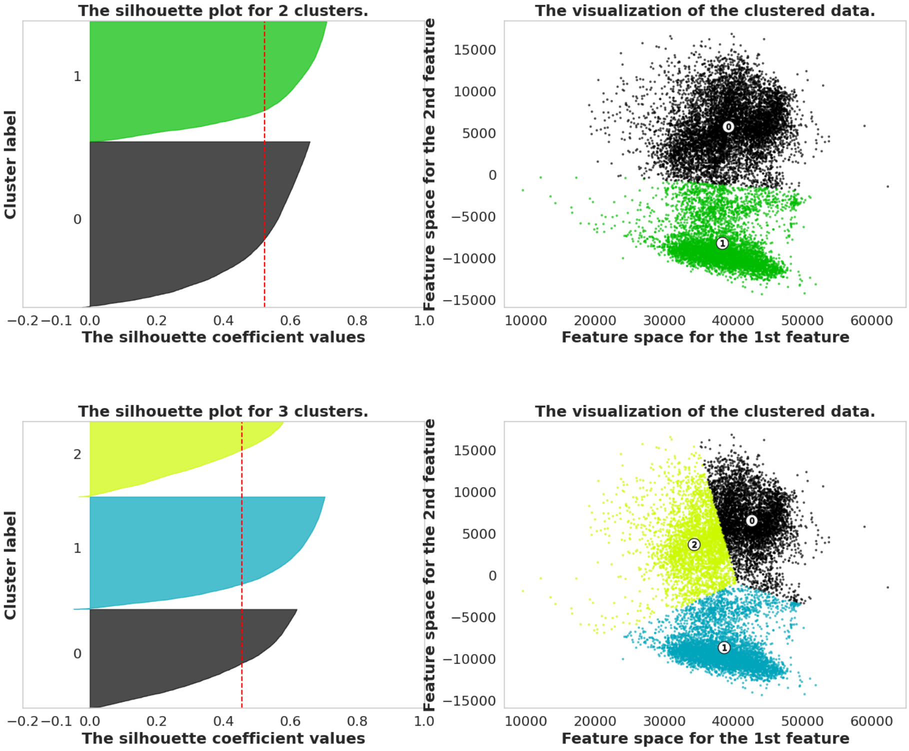


Figure 5.9: Silhouette coefficient distributions and K-means clusters (K-2 and K=3) of non-normalized image pixel vectors reduced with SVD.
Silhouette method evaluation of the K-means clusters produced with the non-normalized BCN image pixel vectors reduced to two features by SVD. The two panels on the left illustrate the Silhouette coefficients with K=2 (top) and K=3 (bottom) where the value of K represents the number of clusters specified to the K-means algorithm. In both panels the area containing the Silhouette coefficients for all the image pixel vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line. The panels on the right illustrate the corresponding clusters where the position of each data point is determined by the value of the two vector features. The same-colour data points make up the different clusters. The centroid and number of each cluster is indicated by numbered white circles.

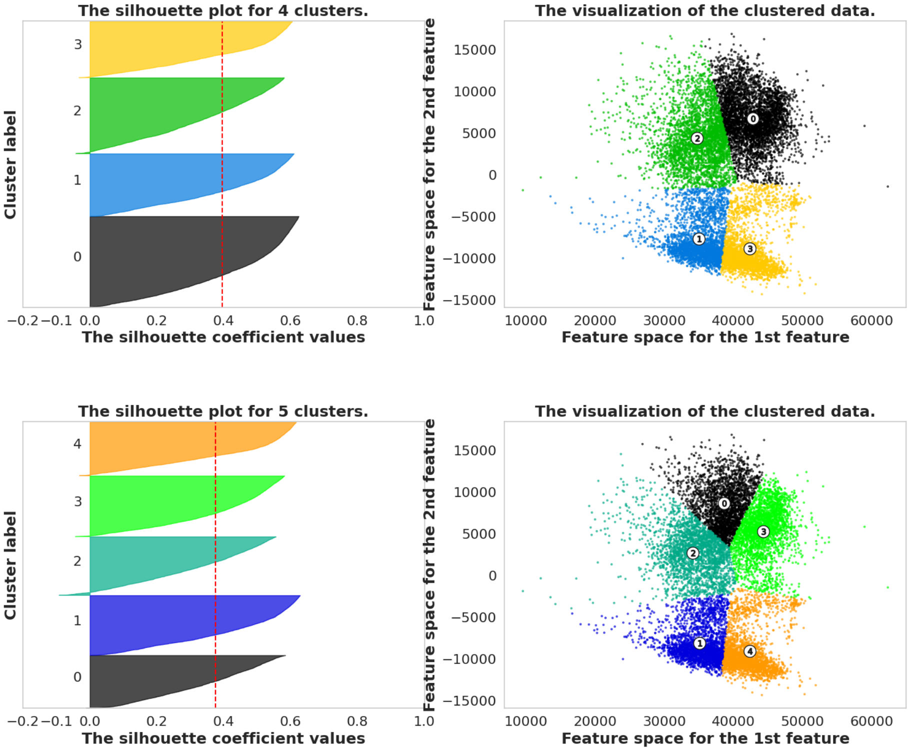


Figure 5.10: Silhouette coefficient distributions and K-means clusters (K=4 and K=5) of non-normalized image pixel vectors reduced with SVD.
Silhouette method evaluation of the K-means clusters produced with the non-normalized BCN image pixel vectors reduced to two features by SVD. The two panels on the left illustrate the Silhouette coefficients with K=4 (top) and K=5 (bottom) where the value of K represents the number of clusters specified to the K-means algorithm. In both panels the area containing the Silhouette coefficients for all the image pixel vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line. The panels on the right illustrate the corresponding clusters where the position of each data point is determined by the value of the two vector features. The same-colour data points make up the different clusters. The centroid and number of each cluster is indicated by numbered white circles.

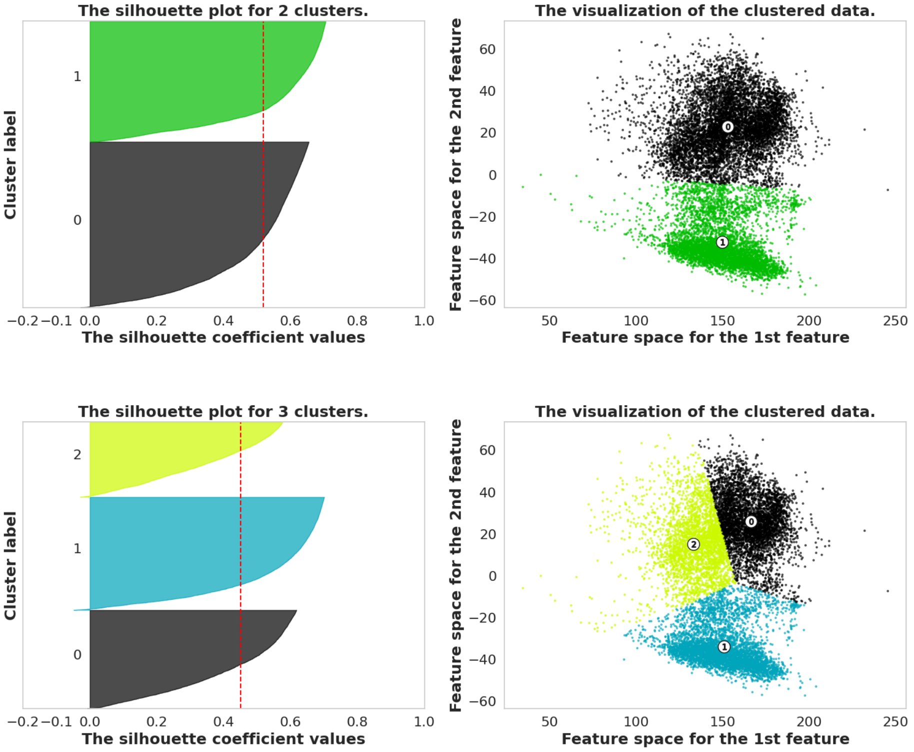


Figure 5.11: Silhouette coefficient distributions and clusters for K-means clustering (K=2 and K=3) of MinMax normalized BCN image pixel vectors reduced with SVD.

Silhouette method evaluation of the K-means clusters produced with the MinMax-normalized BCN image pixel vectors reduced to two features by SVD. The two panels on the left illustrate the Silhouette coefficients with K=2 (top) and K=3 (bottom) where the value of K represents the number of clusters specified to the K-means algorithm. In both panels the area containing the Silhouette coefficients for all the image pixel vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line. The panels on the right illustrate the corresponding clusters where the position of each data point is determined by the value of the two vector features. The same-colour data points make up the different clusters. The centroid and number of each cluster is indicated by numbered white circles.

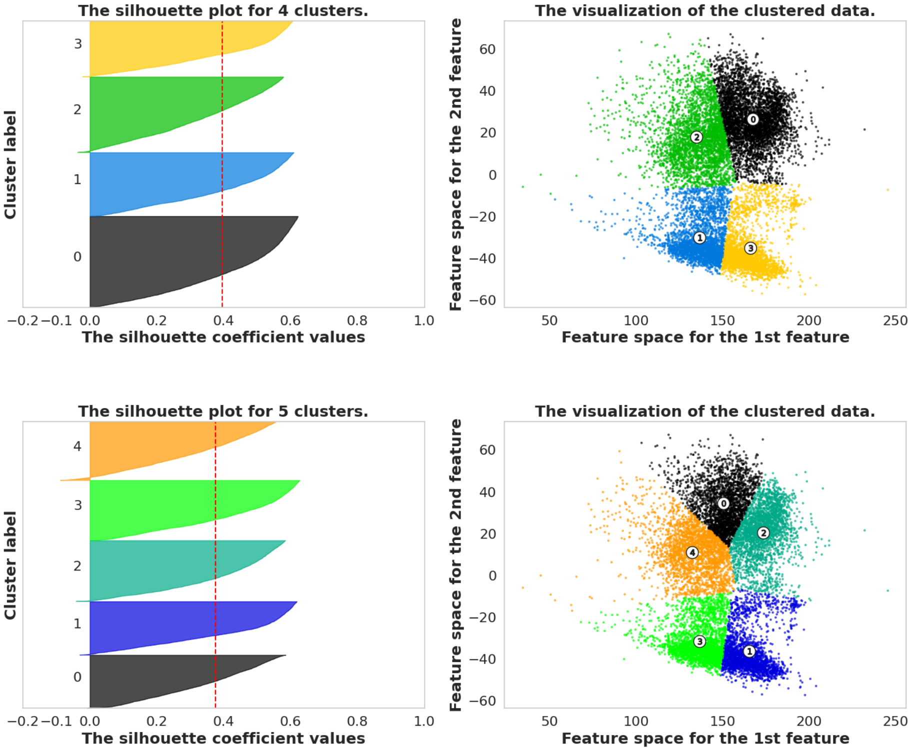


Figure 5.12: Silhouette coefficient distributions and clusters for K-means clustering (K=4 and K=5) of MinMax normalized BCN image pixel vectors reduced with SVD.

Silhouette method evaluation of the K-means clusters produced with the MinMax normalized BCN image pixel vectors reduced to two features by SVD. The two panels on the left illustrate the Silhouette coefficients with K=4 (top) and K=5 (bottom) where the value of K represents the number of clusters specified to the K-means algorithm. In both panels the area containing the Silhouette coefficients for all the image pixel vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line. The panels on the right illustrate the corresponding clusters where the position of each data point is determined by the value of the two vector features. The same-colour data points make up the different clusters. The centroid and number of each cluster is indicated by numbered white circles.

minmaxSVD (99.60%) clusters were also contained in the largest cluster produced with the full raw pixel vectors. By deduction we can conclude that all three methods group the images in a very similar fashion and that the large size of the full pixel vectors does not seem to be detrimental to the clustering. The difference in Silhouette scores obtained for the clusters produced with the full raw pixel vectors and those produced with the reduced vectors appears to be simply caused by the higher dimensionality of the vectors used to calculate the Euclidian distances (both intra and inter cluster).

For this reason, we decided to use only the MinMax normalized and SVD reduced clusters for classification with the EfficientNet B4 model to confirm that the very small difference between the clusters (Figure 5.6C and 5.6D) does not have large effects on classification efficiency. As expected, and as can be seen in Table 5.5, the best average WACC achieved with the two clusters (MinMaxSVD), 0.8204 ±0.0090 with 7185 images and 0.7842 ±0.0071 with 5228 images, is practically identical to the best average WACC values obtained with the full raw pixel (rawImg) vectors (0.8164 ±0.0037 and 0.7842 ±0.0074).

Since pixel values, raw or normalized, may not be the best informative features to efficiently differentiate images into groups with similar characteristics, we chose to explore a different approach. In this case we chose to use a CNN model, ResNet50 (He *et al.*, 2016), pretrained on the ImageNet dataset to extract features from the BCN images. The ResNet50 architecture is a traditional model, one of the first deep CNN neural networks, which has shown to have a very good performance for the classification of the ImageNet images (He *et al.*, 2016)(Koonce, 2021). We anticipated that a model pretrained on the ImageNet images may be able to extract general features that could group the images based on inherant unseen characteristics, rather than sperating them into disease classes.

For this purpose, we fed the BCN images to the pretrained ResNet50 CNN without any dense layers and initially captured the features produced by the model and flattened the output of the last convolutional layer. This produces vectors containing 100,352 features, the result of flattening the information contained in 2048 7x7 feature maps. In order to avoid using very large vectors we chose to add a pooling step at the end of the CNN architecture. Two different pooling approaches were used: global average pooling (GAP) or global max pooling (GMP). These pooling operations produce a single feature from each of the 7x7 feature maps. GAP produces features representing the average of all the values of 7x7 feature maps whereas GMP outputs the maximal value of the feature maps. These operations therefore produce feature vectors containing 2048 values which are smaller-sized representative versions of the full original unpooled vectors.

Again, after extracting the features from the BCN images, we used the K-means algorithm to separate the images into two or more clusters based on the feature vectors. Since these feature vectors are still quite large, we first applied the previously described normalization and dimension reduction algorithms to produce even smaller vectors. With both the GAP and GMP vector arrays the best Silhouette scores were obtained with two clusters (*i.e.* K=2) after either applying SVD to non-normalized vectors (0.5860 with GAP (Table 5.6 and Figures 5.13 and 5.14) and 0.5865 with GMP (Table 5.6 and Figures 5.17 and 5.18)) or SVD after MinMax normalization (0.6087 with GAP (Table 5.6 and Figures 5.15 and 5.16) and 0.5998 with GMP (Table 5.6 and Figures 5.19 and 5.20)) of the vector arrays.

Table 5.6: Silhouette scores obtained for different K-means BCN image clusters based on RN50 feature vectors// Silhouette average scores for K-means clustering into K clusters produced with the BCN ResNet50 (RN50) feature vectors. These clusters were all produced with either global average pooled (GAP) feature vectors or global maximum pooled (GMP) feature vectors. The feature vectors were either further MinMax-normalized (MinMax) or non-normalized (NoNorm) prior to being reduced to two features by SVD. The vectors shown in bold are those which produced clusters with the best Silhouette scores (for 2 clusters (K=2) in all cases) and were used for analysis in our study. As shown, many permutations of normalization methods (Norm), dimension reductions methods (Red) and number of dimensions (Dim) (features) were tested to obtain the best clustering possible.

GAP							
avgMinMaxSVD	MinMax	SVD	2	0.6087	0.4621	0.396	0.3528
avgMinMaxSVDtest2	MinMax	SVD	3	0.4631	0.4637	0.4109	0.3568
avgMinMaxSVDtest3	MinMax	SVD	4	0.3849	0.3779	0.3633	0.3244
avgNoNormSVD	NA	SVD	2	0.5860	0.4383	0.3745	0.3528
avgNoNormSVDtest2	NA	SVD	3	0.4367	0.4708	0.4299	0.3546
avgNoNormSVDtest3	NA	SVD	4	0.3530	0.3768	0.3521	0.2946
avgZscoreSVDtest1	Zscore	SVD	2	0.4658	0.5331	0.5123	0.4567
avgZscoreSVDtest2	Zscore	SVD	3	0.3611	0.4020	0.4041	0.3853
GMP							
maxMinMaxSVD	MinMax	SVD	2	0.5998	0.4561	0.4056	0.3784
maxMinMaxSVDtest2	MinMax	SVD	3	0.4604	0.4491	0.3853	0.3441
maxMinMaxSVDtest3	MinMax	SVD	4	0.3956	0.3776	0.3568	0.3262
maxNoNormSVD	NA	SVD	2	0.5865	0.4923	0.3943	0.3786
maxNoNormSVDtest2	NA	SVD	3	0.4372	0.4202	0.3814	0.3376
maxNoNormSVDtest3	NA	SVD	4	0.3727	0.3499	0.3320	0.3062
maxZscoreSVDtest1	Zscore	SVD	2	0.4647	0.5065	0.4877	0.4375
maxZscoreSVDtest2	Zscore	SVD	3	0.3523	0.3890	0.3834	0.3671

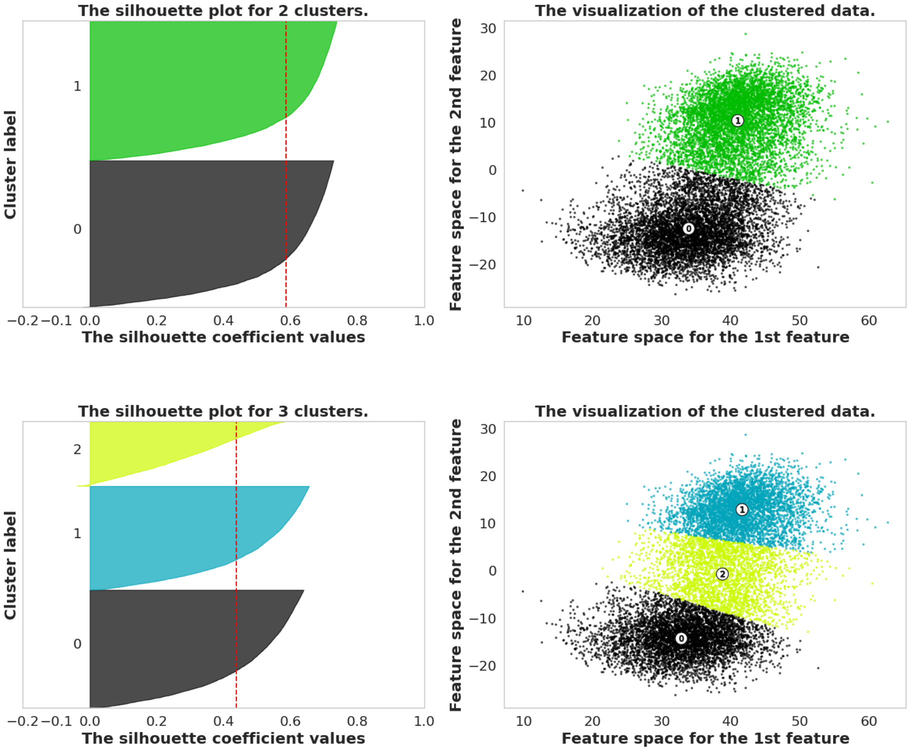


Figure 5.13: Silhouette coefficient distributions and cluster for K-means clustering (K=2 and K=3) of non-normalized BCN image RN50avg feature vectors reduced with SVD.
Silhouette method evaluation of the K-means clusters produced with the BCN image non-normalized ResNet50-extracted and global average pooled (RN50avg) feature vectors reduced to two features by SVD. The two panels on the left illustrate the Silhouette coefficients with K=2 (top) and K=3 (bottom) where the value of K represents the number of clusters specified to the K-means algorithm. In both panels the area containing the Silhouette coefficients for all the image RN50avg feature vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line. The panels on the right illustrate the corresponding clusters where the position of each data point is determined by the value of the two vector features. The same-colour data points make up the different clusters. The centroid and number of each cluster is indicated by numbered white circles.

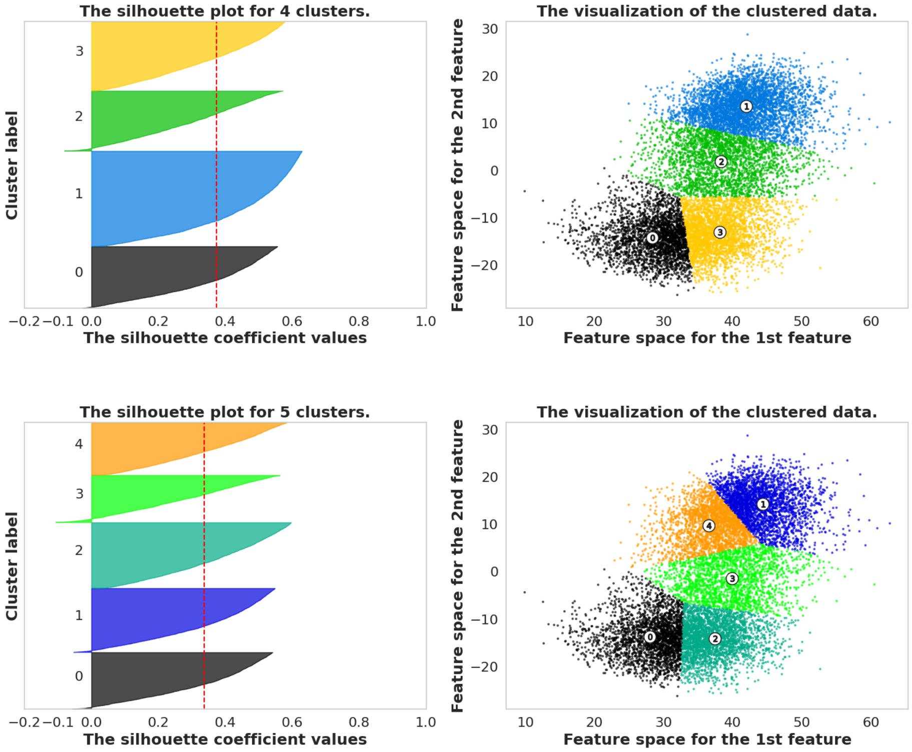


Figure 5.14: Silhouette coefficient distributions and clusters for K-means clustering (K=4 and K=5) of non-normalized BCN image RN50avg feature vectors reduced with SVD.
Silhouette method evaluation of the K-means clusters produced with the BCN image non-normalized ResNet50-extracted and global average pooled (RN50avg) feature vectors reduced to two features by SVD. The two panels on the left illustrate the Silhouette coefficients with K=4 (top) and K=5 (bottom) where the value of K represents the number of clusters specified to the K-means algorithm. In both panels the area containing the Silhouette coefficients for all the image RN50avg feature vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line. The panels on the right illustrate the corresponding clusters where the position of each data point is determined by the value of the two vector features The same-colour data points make up the different clusters. The centroid and number of each cluster is indicated by numbered white circles.

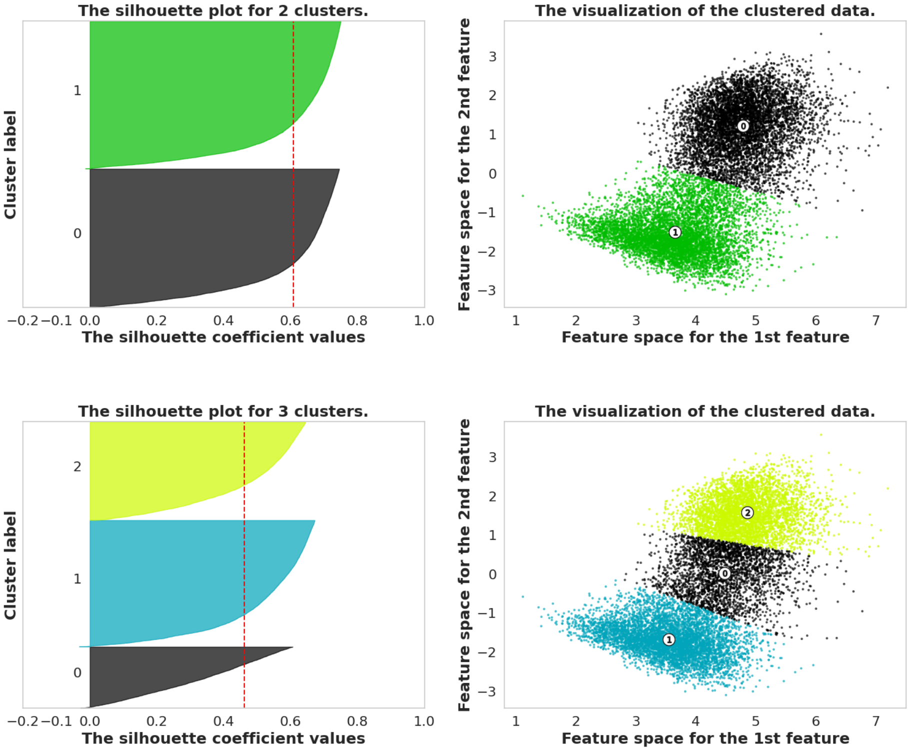


Figure 5.15: Silhouette coefficient distributions and clusters for K-means clustering (K=2 and K=3) of MinMax normalized BCN image RN50avg feature vectors reduced with SVD.
Silhouette method evaluation of the K-means clusters produced with the BCN image MinMax normalized ResNet50-extracted and global average pooled (RN50avg) feature vectors reduced to two features by SVD. The two panels on the left illustrate the Silhouette coefficients with K=2 (top) and K=3 (bottom) where the value of K represents the number of clusters specified to the K-means algorithm. In both panels the area containing the Silhouette coefficients for all the image RN50avg feature vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line. The panels on the right illustrate the corresponding clusters where the position of each data point is determined by the value of the two vector features. The same-colour data points make up the different clusters. The centroid and number of each cluster is indicated by numbered white circles.

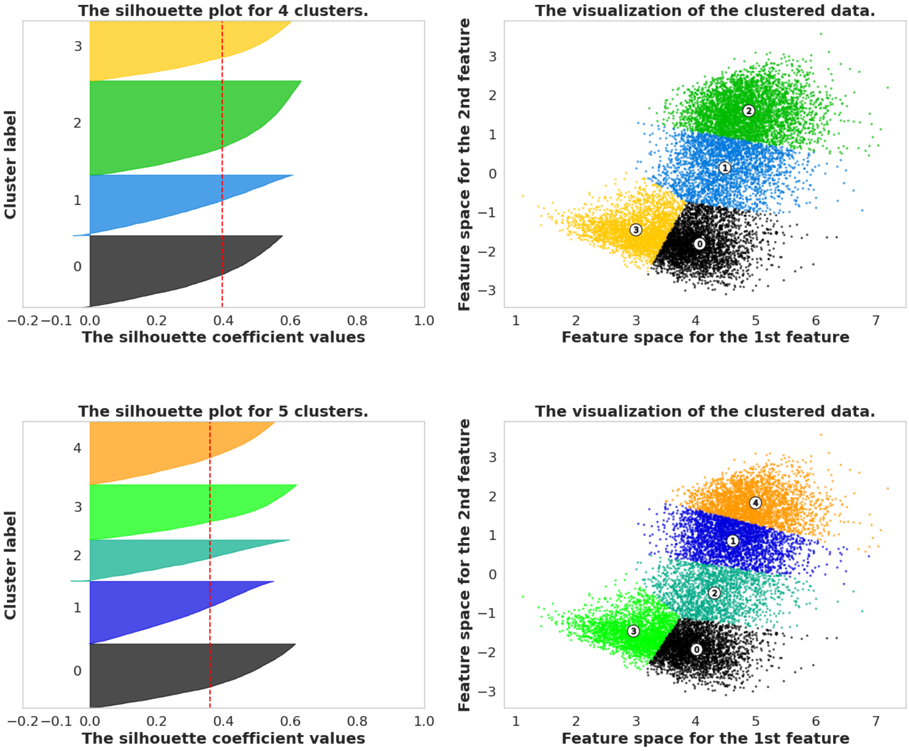


Figure 5.16: Silhouette coefficient distributions and clusters for K-means clustering (K=4 and K=5) of MinMax normalized BCN image RN50avg feature vectors reduced with SVD.
Silhouette method evaluation of the K-means clusters produced with the BCN image MinMax normalized ResNet50-extracted and global average pooled (RN50avg) feature vectors reduced to two features by SVD. The two panels on the left illustrate the Silhouette coefficients with K=4 (top) and K=5 (bottom) where the value of K represents the number of clusters specified to the K-means algorithm. In both panels the area containing the Silhouette coefficients for all the image RN50avg feature vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line. The panels on the right illustrate the corresponding clusters where the position of each data point is determined by the value of the two vector features The same-colour data points make up the different clusters. The centroid and number of each cluster is indicated by numbered white circles.

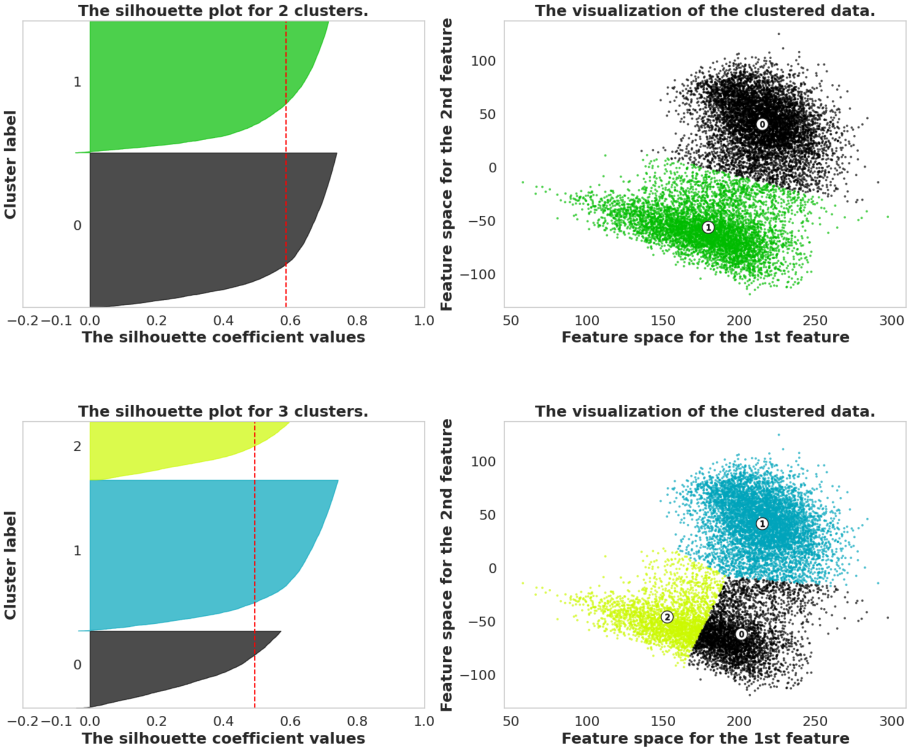


Figure 5.17: Silhouette coefficient distributions and clusters for K-means clustering (K=2 and K=3) of non-normalized BCN image RN50max feature vectors reduced with SVD.
Silhouette method evaluation of the K-means clusters produced with the BCN image non-normalized ResNet50-extracted and global maximum pooled (RN50max) feature vectors reduced to two features by SVD. The two panels on the left illustrate the Silhouette coefficients with K=2 (top) and K=3 (bottom) where the value of K represents the number of clusters specified to the K-means algorithm. In both panels the area containing the Silhouette coefficients for all the image RN50max feature vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line. The panels on the right illustrate the corresponding clusters where the position of each data point is determined by the value of the two vector features. The same-colour data points make up the different clusters. The centroid and number of each cluster is indicated by numbered white circles.

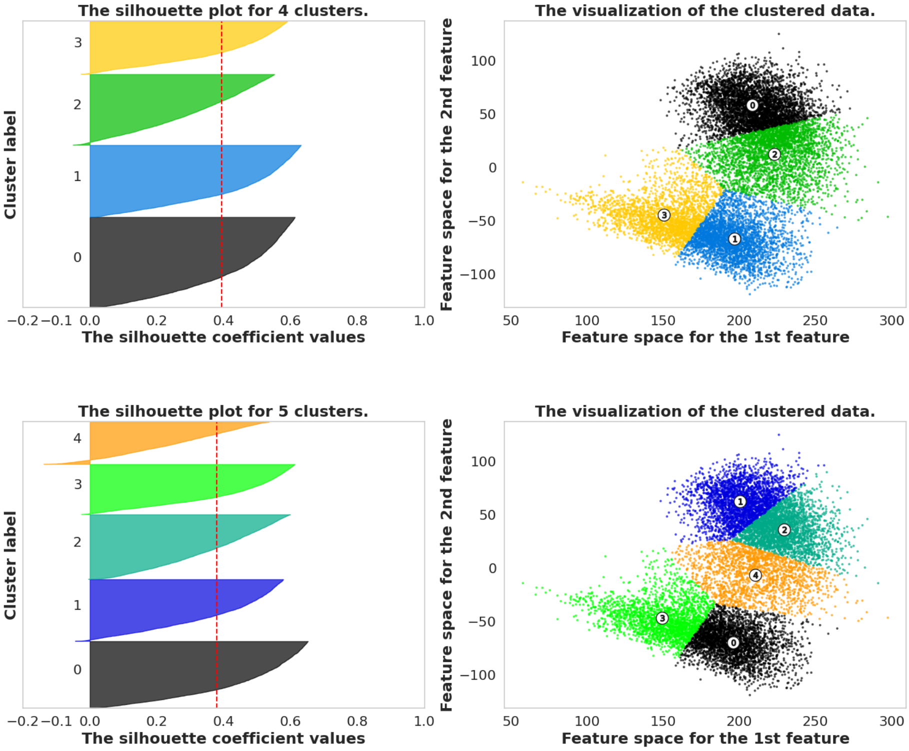


Figure 5.18: Silhouette coefficient distributions and clusters for K-means clustering (K=4 and K=5) of non-normalized BCN image RN50max feature vectors reduced with SVD.
Silhouette method evaluation of the K-means clusters produced with the BCN image non-normalized ResNet50-extracted and global maximum pooled (RN50max) feature vectors reduced to two features by SVD. The two panels on the left illustrate the Silhouette coefficients with K=4 (top) and K=5 (bottom) where the value of K represents the number of clusters specified to the K-means algorithm. In both panels the area containing the Silhouette coefficients for all the image RN50max feature vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line. The panels on the right illustrate the corresponding clusters where the position of each data point is determined by the value of the two vector features. The same-colour data points make up the different clusters. The centroid and number of each cluster is indicated by numbered white circles.

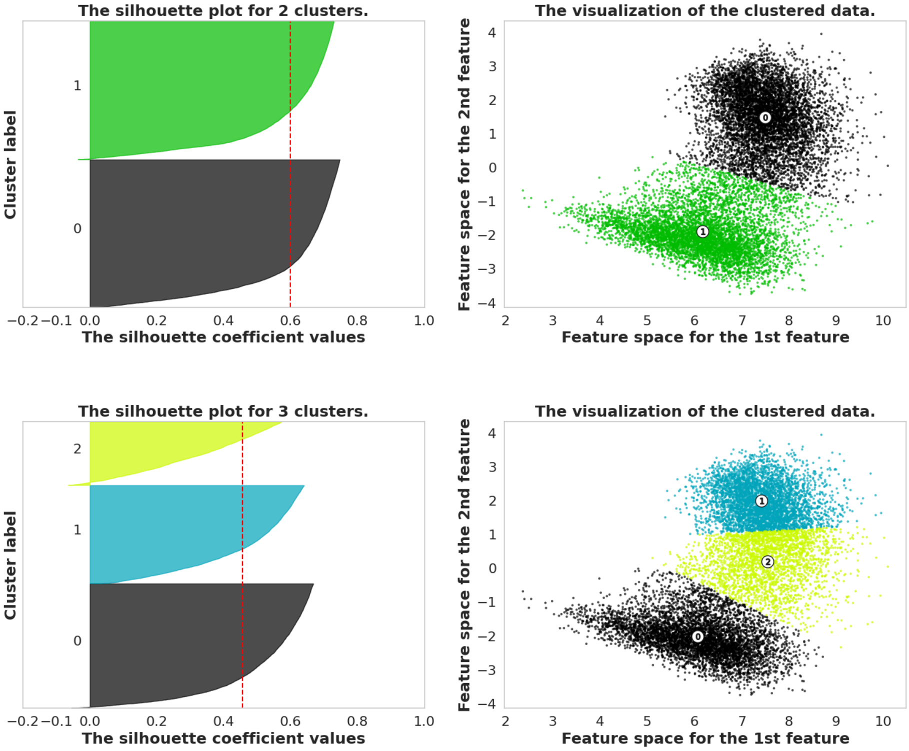


Figure 5.19: Silhouette coefficient distributions and clusters for K-means clustering (K=2 and K=3) of MinMax normalized BCN image RN50max feature vectors reduced with SVD.
Silhouette method evaluation of the K-means clusters produced with the BCN image MinMax-normalized ResNet50-extracted and global maximum pooled (RN50max) feature vectors reduced to two features by SVD. The two panels on the left illustrate the Silhouette coefficients with K=2 (top) and K=3 (bottom) where the value of K represents the number of clusters specified to the K-means algorithm. In both panels the area containing the Silhouette coefficients for all the image RN50max feature vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line. The panels on the right illustrate the corresponding clusters where the position of each data point is determined by the value of the two vector features. The same-colour data points make up the different clusters. The centroid and number of each cluster is indicated by numbered white circles.

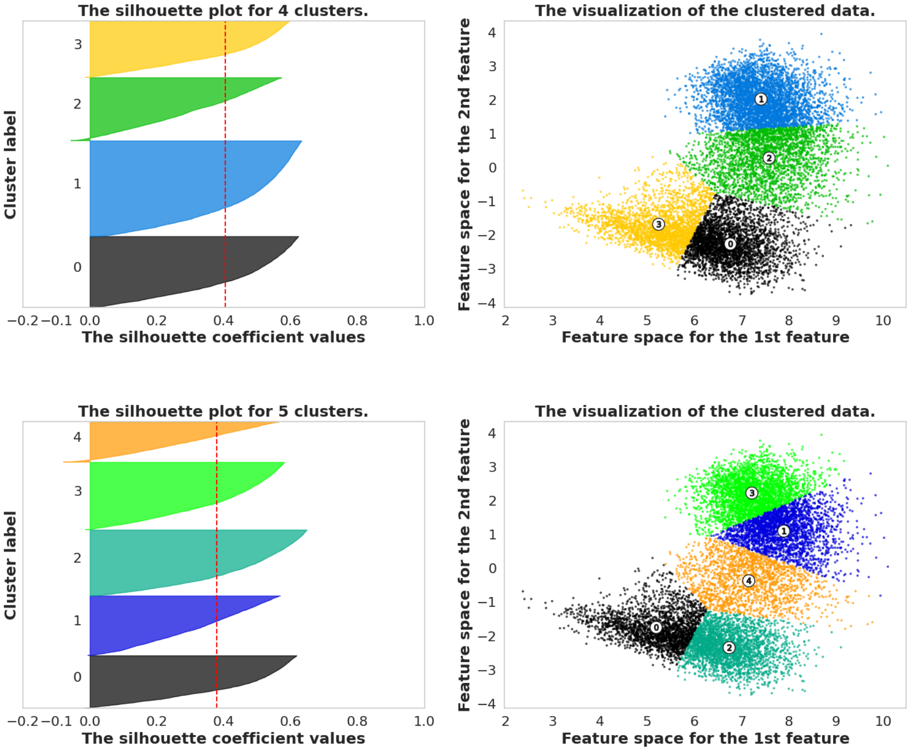


Figure 5.20: Silhouette coefficient distributions and clusters for K-means clustering (K=4 and K=5) of MinMax-normalized BCN image RN50max feature vectors reduced with SVD.
Silhouette method evaluation of the K-means clusters produced with the BCN image MinMax-normalized ResNet50-extracted and global maximum pooled (RN50max) feature vectors reduced to two features by SVD. The two panels on the left illustrate the Silhouette coefficients with K=4 (top) and K=5 (bottom) where the value of K represents the number of clusters specified to the K-means algorithm. In both panels the area containing the Silhouette coefficients for all the image RN50max feature vectors (data points) making up each cluster is filled with a different color. The average Silhouette score for all the clusters is indicated by the red dashed line. The panels on the right illustrate the corresponding clusters where the position of each data point is determined by the value of the two vector features. The same-colour data points make up the different clusters. The centroid and number of each cluster is indicated by numbered white circles.

As seen in Table 5.6 (vs Table 5.4), these approaches produced the highest Silhouette scores observed as of yet in our study. Also as seen in Figures 5.13 to 5.20 (top-right panel for all), although the clusters produced are still not clearly separated, in all cases the most concentrated areas appear to be clearly distinct from one another. However, like the pairs of clusters produced by the non-normalized and MinMax-normalized pixel vectors reduced with SVD (Figures 5.9 and 5.11, top-rigth panels) the clusters with the feature vectors also have many data points near the cluster intersection and are not clearly separated.

However, in these cases the clusters produced contained a more even number of images than those produced with pixel-based vectors (Table 4.1, RN50 Features section). The GMP vectors with normalization produced the two image clusters with the largest size difference where one cluster contained 6797 images and the other 5706 (Table 4.1, maxMinMaxSVD). In all cases the produced clusters again had somewhat different image distributions amongst the different disease classes as compared to that observed in the full BCN dataset. However, again the images did not seem to be segmented based on class as each cluster contains images from all eight classes.

As shown in the Venn diagrams in Figure 5.21, the different pairs of clusters produced by feature extraction are all quite similar to one another, but show some small differences. More importantly, as seen in Figure 5.22 where clusters obtained with MinmMax-normalized and SVD-reduced GAP ResNet50 feature vectors (RN50avgMM2_0 and RN50avgMM2_1) are used as representatives of the clusters produced from ResNet50 features, we can see that these clusters are significantly different to those produced with pixel vectors (raw full pixel vectors (raw2_0) and Z-scored-normalized full pixel vectors (scaled2_0)).

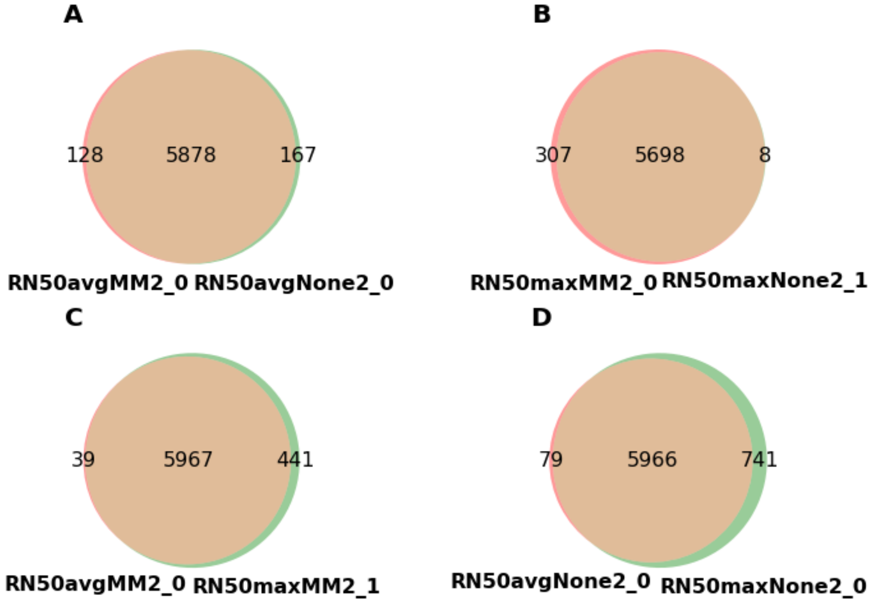


Figure 5.21: Overlap between K-means image clusters produced from ResNet50-extracted features.
Venn diagrams showing the overlap between pairs of datasets resulting from the K-means clustering (K=2) of ResNet50-extracted features. Global average pooled ResNet50 feature vectors (RN50avg) and global maximum pooled ResNet50 feature vectors (RN50max) were used to produce the clusters. In all cases the feature vectors used to produce the K-means clusters were reduced to two features by SVD. The first cluster produced from clustering with MinMax-normalized RN50avg vectors (RN50avgMM2_0) was compared to the first cluster produced with non-normalized RN50avg vectors (RN50avgNone2_0) (A) and the second cluster produced with MinMax-normalized RN50Max vectors (RN50maxMM2_1) (C). The large overlaps show that the clusters produced with MinMax normalization and those produced without normalization of the RN50avg vectors are very similar (A). The same applies for the clusters produced with the MinMax-normalized RN50avg and RN50max vectors (C). In panel B, a cluster produced with MinMax-normalized RN50max vectors (RN50maxMM2_0) is shown to be very similar to a cluster produced with the non-normalized RN50Max vectors (RN50maxNone2_1). Finally, in panel D, a cluster produced with the non-normalized RN50avg vectors (RN50avgNone2_0) is shown to largely overlap with a cluster produced with non-normalized RN50max vectors (RN50maxNone3_0). Overall the figure illustrates that all pairs of image clusters produced with ResNet50-extacted features a generally very similar.

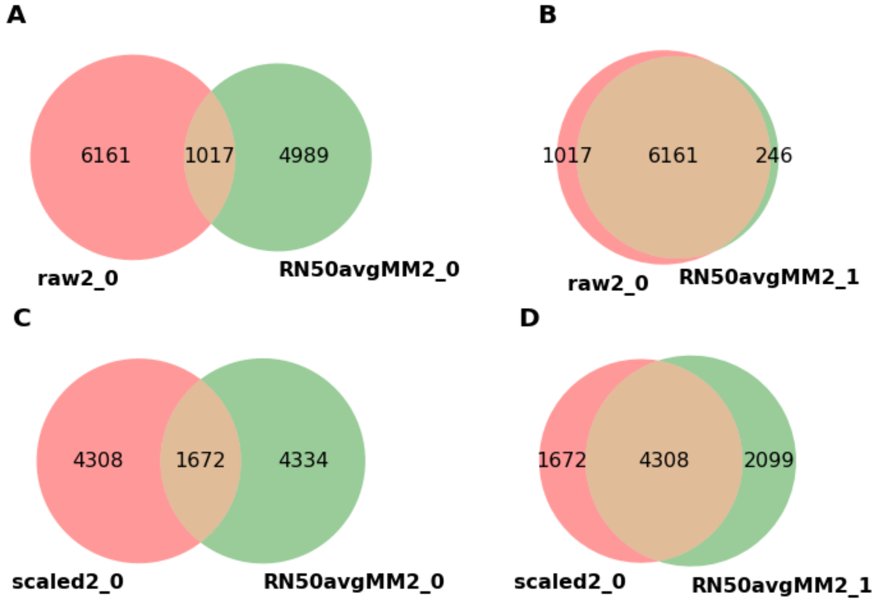


Figure 5.22: Overlap between K-means image clusters produced from pixel vectors and those produced from ResNet50-extracted features.
Venn diagrams showing the overlap between pairs of datasets resulting from the K-means clustering (K=2) of the BCN images based on pixel vectors and ResNet50-extracted features. Global average pooling was used to produce the ResNet50 feature vectors (RN50avg). The first cluster produced with the non-normalized image pixel vectors (raw2_0) (A and B) and the first cluster produced with the Z-score-normalized pixel vectors (scaled2_0) (C and D) are each compared to both clusters produced with the MinMax-normalized RN50avg feature vectors reduced to two features by SVD (RN50avgMM2_0 and RN50avgMM2_1). These diagrams show that overall, the pairs of image clusters produced with the pixel vectors differ significantly from those produced with the ResNet50-extracted feature vectors.

To assess if this approach allowed us to get improved classification when the images from each cluster were analysed separately, we again used the EfficientNet B4 model to classify the images in the eight disease classes. As can be seen in Table 5.7, the best average WACCs achieved with these datasets were in the same range as those achieved with the Z-scored normalized pixel vector (Table 5.5) with some exceptions. In particular one of the MinMax-normalized GAP ResNet50 feature cluster (avgMinMaxSVD2_1) produced the best result obtained with any of the clusters (all methods) with a best average WACC above 0.85 (0.8560 ±0.0037). Also, one of the clusters produced with non-normalized GAP ResNet50 feature vectors (avgNoNormSVD2_1) produced only a slightly lower best average WACC (0.8443 ±0.0113).

The best average WACCs obtained with at least one of the clusters produced by K-means (K=2) with the different SVD-reduced ResNet50 feature vectors are much better than those obtained with the same-sized random BCN datasets (Table 5.7). However, even if clustering can yield better classification efficiency with a reduced number of images than that achieved with the random dataset, it remains that the best classification efficiency was produced with the full BCN dataset in its entirety (Table 5.7). Again this is likely attributed to the fact that this latter dataset is much larger (*i.e* roughly twice as large as the cluster datasets).

Finally, we assessed if K-means clustering with ResNet50 feature vectors could separate MSK images from images originating from the other datasets. We generated extracted features from the various, previously described, balanced datasets containing equal amounts of images from all three ISIC2019 sub-datasets: MSK, BCN and HAM. We generated ResNet50 feature vectors from the images with two of the same approaches used in the previous classification studies. Specifically, non-normalized (avgNoNormSVD) or MinMax-normalized (avgMinMaxSVD) GAP

Table 5.7: Classification results with K-means BCN image RN50 feature vector clusters into eight disease classes.

The mean best WACC and standard deviation achieved with the EfficientNet B4 model with various datasets when classifying the images in all eigth labelled disease classes. The classification results of the full BCN dataset (BCN) previously presented in Table 5.1 is shown for reference purposes. The BCN k-means clusters produced from RN50 feature vectors were all produced with either global average pooled feature vectors (avg) or global maximum pooled feature vectors (max). These feature vectors were either MinMax normalized (MinMax) or non-normalized (NoNorm) prior to being reduced to two features by SVD. For all experiments the K-means cluster (Cluster), the number of images (#Imgs), the number of cross-validations performed (#CVs), the number of epochs (#Epochs) of training completed for each cross-validation and the mean best WACC ± the standard deviation (AvgWACC ±Std) is shown. In addition, the results obtained with the two randomly produced datasets (random6200_1 and random6200_2) are shown for comparision purposes.

DataSet	Cluster	#Imgs	#CVs	#Epochs	AvgWACC ±Std
BCN	NA	12413	5	100	0.8933 ±0.0096
RN50 Features					
avgMinMaxSVD	2_0	6006	5	100	0.7932 ±0.05.1
	2_1	6407	5	100	0.8560 ±0.0037
avgNoNormSVD	2_0	6045	5	100	0.8086 ±0.0053
	2_1	6368	5	100	0,8443 ±0.0113
maxMinMaxSVD	2_0	6005	3	100	0.8228 ±0.0060
	2_1	6408	3	100	0.8270 ±0.0029
maxNoNormSVD	2_0	6707	3	100	0.8363 ±0.0043
	2_1	5706	3	100	0.8092 ±0.05.1
BCN random					
random6200_1	NA	6199	5	100	0.7615 ±0.0130
random6200_2	NA	6199	5	100	0.8012 ±0.0037

ResNet50 feature vectors were produce with all the images and further reduced to two features with SVD.

As can be seen in Tables 5.2 and 5.8, again, no clear speparation was achieved. In all cases with K-means clustering in two or three clusters (K=2 or K=3), images of the MSK dataset were dispersed in the different clusters and no cluster was produced that was predominantly composed of MSK images.

5.4 classification in reduced numbers of classes

Since most of the research carried out to produce CAD models to diagnose skin cancer lesion images aims only to classify the images between malignant and benign lesions (Dick *et al.*, 2019), we were interested to see if our K-means clustering approach may improve classification of images when the classes were more broadly defined. Of the different classes represented in the ISIC2019 dataset, melanoma is by far considered to be the most malignant. However, BCC and SCC can also metastasize and have severe deteriorating effects on patients. For this purpose, we produced new labels for the different datasets that group some of the disease classes together and will allow classification into a reduced number of classes.

In a first case melanoma was labelled as malignant and all other image classes were considered as benign (MEL vs Others). In a second approach we considered MEL, BCC and SCC to be malignant and all other classes to be benign (MBS vs Others). These new labelling strategies allowed us to do binary classification experiments. We also produced a third set of labels in witch four separate classes were produced: MEL, BCC, SCC and others. This last set was produced to see if classification between the four most important classes could yield improved results

Table 5.8: K-means clustering of ResNet50 feature vectors produced with NV and MEL image datasets including BCN, HAM and MSK images.
ResNet50-extracted feature vectors were generated with images from the three parent datasets: BCN, HAM and MSK. Non-normalized (avgNoNormSVD) and MinMax-normalized (avgMinMaxSVD) global average pooled ResNet50 (RN50) feature vectors from the images of the different datasets were reduced to two features with SVD. The vector matrices produced were analyzed by K-means (K=2 or K=3) and the number (#MSK) and the proportion (%MSK) of MSK in each dataset is presented. The number of images (#Imgs) in all original datasets and in all clusters produced is also indicated. The EqualDownNV and EqualDownMEL datasets contain all the NV (4.16) or MEL (374) images from the MSK dataset labeled as downsampled, respectively. The EqualOriNV and EqualOriMEL datasets contain all the NV (648) or MEL (178) images from the MSK dataset which are not labeled as downsampled, respectively. All of the EqualDown and EqualOri dataset contain an identical number of images of the same class from each of the BCN and HAM datasets.

equalDownNV	3948	4.16	0.3333	**EqualDownMEL**	5.12	374	0.3333
avgNoNormSVD				avgNoNormSVD			
K-2_0	1853	379	0.2045	K-2_0	325	19	0.0462
K-2_1	2095	937	0.4472	K-2_1	797	355	0.4454
K-3_0	1085	472	0.4350	K-3_0	552	277	0.5018
K-3_1	1233	180	0.1460	K-3_1	250	13	0.0520
K-3_2	1630	664	0.4073	K-3_2	320	84	0.2625
avgMinMaxSVD				avgMinMaxSVD			
K-2_0	1468	795	0.5416	K-2_0	844	359	0.4254
K-2_1	2480	521	0.2100	K-2_1	278	15	0.0540
K-3_0	1137	103	0.0906	K-3_0	398	166	0.44.1
K-3_1	1299	692	0.5327	K-3_1	244	12	0.0492
K-3_2	4.12	521	0.3446	K-3_2	480	196	0.4083
equalOriNV	1944	648	0.3333	**EqualOriMEL**	534	178	0.3330
avgNoNormSVD				avgNoNormSVD			
K-2_0	1267	434	0.3425	K-2_0	180	32	0.1778
K-2_1	677	214	0.34.1	K-2_1	354	146	0.4124
K-3_0	692	146	0.2110	K-3_0	5.1	44	0.2716
K-3_1	865	409	0.6150	K-3_1	233	108	0.4635
K-3_2	387	93	0.2403	K-3_2	139	26	0.1871
avgMinMaxSVD				avgMinMaxSVD			
K-2_0	1499	542	0.3616	K-2_0	173	30	0.1734
K-2_1	445	106	0.2382	K-2_1	361	148	0.3885
K-3_0	764	344	0.4502	K-3_0	133	34	0.2556
K-3_1	375	85	0.2267	K-3_1	271	123	0.4539
K-3_2	805	219	0.2720	K-3_2	130	21	0.4.15

over those achieved with all eight original disease classes.

Again, we solely focused on the BCN dataset for these experiments as this is the dataset that contains a higher proportion of images in the classes considered malignant (Table 4.1). The MEL vs Others dataset produced was still largely unbalanced between the two classes as the MEL class only contains 2857 images, less than a quarter (proportion = 0.23) of the full dataset, and the others together make up 9556 images (proportion = 0.77). In the case of the MBS vs Others, 6097 images are labelled as malignant and 6316 images were labelled as benign. This dataset is therefore much better balanced, but the malignant class in this labelling strategy is expected to contain more image variation than in the MEL vs Others dataset. Finally, the four-class dataset contains 2857 MEL, 2809 BCC, 431 BCC and 6316 others. This dataset, like the MEL vs Others, also represents an important classification challenge due to the fact that it is largely unbalanced.

We used a modified version of the EfficientNet B4 model to carry out experiments in which images are classified into a reduced number of classes. In these cases, we again used a weighted cross-entropy loss function to counter the unbalanced property of the datasets. Also, we based our evaluation on the average WACC obtained from evaluation on the validation dataset in three-fold cross-validation.

To test if our clustering approach is advantageous for the classification in a reduced number of classes, we first carried out classification experiments with the full BCN dataset (no clustering). This dataset was classified based on the three labelling approaches described above. As can be seen in Table 5.9, as expected, due to the lower complexity of the classification associated with the reduced number of classes, we achieved very high best average WACC (above 0.9) values in all cases.

The best average WACC was obtained with the MEL vs Others classification task

Table 5.9: Classification results with K-means BCN image clusters into two or four disease classes.

The mean best WACC and the standard deviation achieved with the EfficientNet B4 model with various datasets when classifying the images in two or four classes. The classification of the full BCN dataset was done in three separate ways: MEL class images vs. Others (MELvsOthers), MEL+BCC+SCC vs Other (MBCvsOthers) and MEL, BCC, SCC, Others (M, B, S, Others). The BCN K-means (K=2) cluster images were classified only in the MELvsOthers fashion. Only the K-means cluster datasets produced with the ResNet50 (RN50) feature vectors were analyzed. These clusters were all produced with either global average pooled feature vectors (avg) or global maximum pooled feature vectors (max). These feature vectors were either MinMax normalized (MinMax) or non-normalized (NoNorm) prior to being reduced to two features by SVD. For all experiments the K-means cluster (Cluster), the number of images (#Imgs), the classification approach used (Classes). the number of cross-validations performed (#CVs), the number of epochs (#Epochs) of training completed for each cross-validation and the mean best WACC ± the standard deviation (AvgWACC ±Std) are shown. In addition, the results obtained with the two randomly produced datasets (random6200_1 and random6200_2) are shown for comparison purposes.

DataSet	Cluster	#Imgs	Classes	#CVs	#Epochs	AvgWACC ±Std
BCN	NA	12413	MELvsOthers	3	100	0.9392 ±0.0057
BCN	NA	12413	MBCvsOthers	3	100	0.9301 ±0.0021
BCN	NA	12413	M, B, S, Others	3	100	0.9105 ±0.0088
BCN Keans						
RN50 Features						
avgMinMaxSVD	2_0	6006	MELvsOthers	3	100	0.9287 ±0.0047
	2_1	6407	MELvsOthers	3	100	0.9078 ±0.04.1
avgNoNormSVD	2_0	6045	MELvsOthers	3	100	0.9180 ±0.0047
	2_1	6368	MELvsOthers	3	100	0.8960 ±0.04.1
maxMinMaxSVD	2_0	6005	MELvsOthers	3	100	0.8710 ±0.0130
	2_1	6408	MELvsOthers	3	100	0.9229 ±0.0037
maxNoNormSVD	2_0	6707	MELvsOthers	3	100	0.9247 ±0.0029
	2_1	5706	MELvsOthers	3	100	0.8941 ±0.05.1
BCN random						
random6200_1	NA	6199	MELvsOthers	5	100	0.8940 ±0.0050
random6200_2	NA	6199	MELvsOthers	5	100	0.8802 ±0.05.1

where we saw an average WACC close to 0.94 (0.9392 ±0.0037). We achieved a slightly lower average WACC of 0.9301 ±0.0021 with the other binary classification task (MBS vs Others). Presumably this is due to the higher variation in the images considered as malignant. Finally, probably due to the higher number of classes, we achieved the lowest best average WACC of 0.9105 ±0.0088 with the four-class classification task (MEL, BCC, SCC vs Others).

As could be expected, in all reduced class number experiments we achieved a higher best average WACC than was observed for classification of the full BCN dataset with all eight labelled classes (0.8933 ±0.0037 (Tables 5.1). This is to be expected since generally classification into multiple classes is more challenging than binary classification or classification with a reduced number of classes. In these latter cases, more broad feature sets allow to predict the class in contrast to cases where a larger number of classes is specified as, in this case, more specific features are required to distinguish between the classes.

Interestingly, binary classification into the MEL vs Others class produced near state-of-the art results (Blundo *et al.*, 2021). The best average WACC achieved with the full BCN dataset is broadly comparable to that achieved by Moura *et al.* (Moura *et al.*, 2019) (0.949) with a smaller and likely less variable dataset.

Next, in order to test our K-means clustering strategy, we used some of the same image clusters produced in the previous part of our study to see if we could achieve higher best average WACCs. Since we achieved the best classification results with the datasets corresponding to the groups produced with the ResNet50 feature vectors (Table 5.7) and the clusters obtained with this strategy had the best Silhouette scores (Table 5.6), we chose to limit our analysis to these datasets. Also, since the best results with the full BCN dataset was achieved with the MEL vs Others binary classification approach and this strategy is the most commonly used in the field, again we limited our study to test only this classification task.

With all the smaller, ResNet50 feature vector-based datasets produced by K-means clustering, as can be seen in Table 5.9, the average WACC achieved was generally slightly lower than that obtained with the full BCN dataset. With the MinMax-normalized GAP clusters (avgMinMaxSVD) we observed best average WACCs of 0.9287 ±0.0047 and 0.9078 ±0.0131 with the two clusters. Comparable results were obtained with MinMax-normalized GMP clusters (maxMinMaxSVD), with which we observed the slightly weaker best average WACCs of 0.8710 ±0.0130 and 0.9229 ±0,0037. Finally, comparable results were produced with the non-normalized GAP and GMP clusters (avgNoNormSVD and maxNoNormSVD, repectively) (Table 5.9). Of course, the similarity observed in the results obtained with these different datasets is expected due to the similarity previously shown between the clusters produced with the different ResNet50 feature vectors (Figure 5.21).

Here, for comparison purposes, we again classified the similar-sized random BCN datasets (random6200_1 and random6200_2). With both these datasets we again obtained weaker best average WACCs of 0.8940 ±0.0050 and 0.8802 ±0.05.1, respectively (Table 5.9). This further supports the conlusion that the K-means clustering with ResNet50 features, can separate the images to a certain extent based on some unknown characteristics to produce better classification results.

However, in this set of experiments we can again see that best classification results are achieved with the full BCN dataset rather than those produced by clustering. Therefore, a larger dataset of homogenous, but possibly slightly more variable images is better for training and classification than smaller size datasets. This again demonstrates the commonly known fact that the size of the dataset, when the data is homogeneous, is a key factor in obtaining better trained models and best accuracy results.

5.5 CAD application for binary skin lesion image classification

A new application, Deep Uncertainty Estimation of Skin Cancer (DUNEScan), that addresses some of the drawbacks of currently available skin cancer diagnosis applications, has been developed as a collaborative effort by Mazoure *et al* (2022). The application uses six state-of-the-art CNN-based publicly available models to give a prediction of a skin lesion image being a malignant or benign cancer. Four of these models, Inceptionv3 (Szegedy *et al.*, 2016), ResNet50 (He *et al.*, 2016), MobileNetv2 (Sandler *et al.*, 2018), EfficientNet (Tan et Le, 2019), are traditional models trained in a supervised fashion with labeled images, whereas the other two, BYOL (Grill *et al.*, 2020) and SwAV (Caron *et al.*, 2020), are self-supervised models and were trained with unlabelled images.

Although recent self-supervised learning models can match the performance of supervised learning models, no skin cancer detection applications have integrated self-supervised models in their pipelines so far. The major advantage of self-supervised methods is the ability to leverage large amounts of unlabeled data to pretrain the latent representation, which can then be used to train a simple classifier, matching the accuracy of fully supervised methods (Grill *et al.*, 2020).

Also, in contrast to models used in currently available applications, all the above models were trained using publically available image datasets obtained from the International ISIC archive. Furthermore, different approaches are used by the application to evaluate the uncertainty of the predictions, including Grad-CAM (Selvaraju *et al.*, 2017), UMAP and binary dropout techniques. Importantly, this application is publicly available (*i.e.* does not require a licence) on a web-server that can be reached at https://www.dunescan.org

We contributed to the preparation of a manuscript describing DUNEScan, which was recently published in the Scientific Reports journal (Mazoure *et al.*, 2022). Since the development and implementation of DUNEScan was carried out by our collaborator, we refer you to the published article (Mazoure *et al.*, 2022) describing the application for detailed information on the materials and methods used. In addition to contributing to the editing of the manuscript as a whole, more importantly we were responsible for the testing of the application, the preparation of the test-related table, the related figures and the corresponding text which is presented in the "testing of the application" section of the summarized results below.

In the following sections we present the key sections of the results presented in the published manuscript. These include an explanation of the strategies used to estimate the uncertainty of the predictions made by the different models, an explanation of the general output made by the DUNEScan application and the results of the tests we performed on the application.

5.5.1 uncertainty estimation

In risk-sensitive fields such as medical imaging, where a false negative prediction can make a difference between life and death, it is crucial to quantify the confidence level of a given model. DUNEScan uses the technique proposed by (Gal et Ghahramani, 2016), randomly disabling parameters of the classifier in an independent set of replicates, and thus achieving an approximate Bayesian posterior over the possible estimates of the model for a given skin lesion image.

The DUNEScan user can select the number of random replicates to be used for a given model. DUNEScan provides uncertainty estimates for each classifier through a boxplot (see Figure 5.23b). If the prediction probabilities with the replicates are tightly concentrated around the mean, this implies that the classifier is confident in its class prediction for the input image and the prediction is trustworthy. In contrast, if the prediction probabilities for the benign and malignant image classes are dispersed and their confidence intervals overlap, this implies that the classifier is not confident and hence, the prediction is not trustworthy.

In addition to the boxplots described above, a classification manifold is also produced with the trained MobileNetv2 model, the fastest of the six available models (see Figure 5.23d). This plot provides an alternative illustration of the confidence of the MobileNetv2 classifier obtained for the input image class prediction.

In the classification manifold graph, each green dot represents a benign skin lesion image used for training, and each red dot represents a malignant one (see Figure 5.23d). If the input image, represented by a blue dot, is located close to the middle of the benign (green) cluster - then the MobileNetv2 model is confident that the lesion is benign, but if it is located close to the middle of the malignant (red) cluster - then the MobileNetv2 model is confident that the lesion is malignant. However, if the blue dot is located close to the boundary of the green and red clusters, then the model exhibits uncertainty in the prediction.

5.5.2 description of the DUNEScan output

DUNEScan first produces and presents the output plot of Grad-CAM (Selvaraju *et al.*, 2017) that highlights the regions of high importance on the input image detected by the MobileNetv2 model (see Figure 5.23c). The above described MobileNetv2 classification manifold is then presented, followed by the uncertainty estimate boxplot for each model selected to analyze the input image (see Figure 5.23b).

Moreover, the output contains a bar-graph showing the average prediction probabilities of both classes obtained with each model used (see Figure 5.23a). By providing the classification probabilities together with means to assess the confidence of these predictions, the DUNEScan server allows practitioners to quickly evaluate the probability that a given skin lesion is benign or malignant. This probability is computed by passing a given skin lesion image through one of the six available models, which outputs a vector of 2 real values (*i.e.* logits). These values are passed through the softmax function, which maps them onto the probability simplex. Hence, all probabilities computed in the paper are of the form: P[malignant|skin lesion image].

5.5.3 testing of the application

Our application was tested by using images from the HAM10000 dataset (Tschandl *et al.*, 2018). This original version of the dataset was used as source data for the ISIC 2018 challenge (Codella *et al.*, 2019). As described in the introduction, this dataset includes images of skin lesions corresponding to seven different classes: actinic keratosis (AK), basal cell carcinoma (BCC), benign keratosis (BKL), dermatofibroma (DF), melanocytic nevi (NV), melanoma (MEL) and vascular lesions (VASC). In contrast to the newer version of the HAM10000 dataset incorporated into the ISIC2019 dataset, no images were labelled as representing the squamous cell carcinoma (SCC) disease class.

Amongst these, MEL and BCC are considered to be malignant skin diseases, whereas the other lesion types are considered as benign. Again, as previously mentioned, the class labels assigned for more than 50% of the images were confirmed by histopathology, while for the others the labels were derived from expert consensus or confirmed by in-vivo confocal microscopy. Selected images were analyzed using 50 replicates with all six CNN models available in DUNEScan to give an overall classification prediction.

MEL and NV images, the most common malignant and benign classes of lesions in the dataset, representing 11% and 67% of the dataset, respectively, were used to assess the performance of the application. In general, the prediction average and the confidence in the prediction vary between the different algorithms. However, in most cases they broadly tend to agree on the prediction with some exceptions.

For example, for the MEL1 image (ISIC_24482) presented in Figure 5.24a, all the algorithms, except BYOL, give a malignant prediction with a probability greater than 0.80 (Table 5.10; for improved readability, it is expressed in percentages in Figures 5.23 to 5.26). As also illustrated in Figures 5.24 to 5.26, half of the algorithms (ResNet50, EfficientNet and SwAV) are highly confident in their predictions as they all output low-variance probability distributions. The MobileNetv2 and InceptionV3 models also yield reliable predictions, but the spread of their approximate posterior distribution is noticeably larger. However, the BYOL model generally provided low-confidence predictions for the images analyzed, and thus should be used with caution.

In the case of the MEL2 image (ISIC_24751), most algorithms yield a high probability of malignancy (above 0.90) with the exception of InceptionV3 and BYOL, which suggest that the lesion is benign with a probability of 0.74 and 0.93, respectively (see Table 5.10 and Figure 5.24b). Although the confidence intervals produced by InceptionV3 do not overlap, they are considerably larger than those produced by the other models. Therefore, the results produced by InceptionV3 and BYOL are less reliable than the consensus prediction obtained with the rest of the models for the MEL2 image.

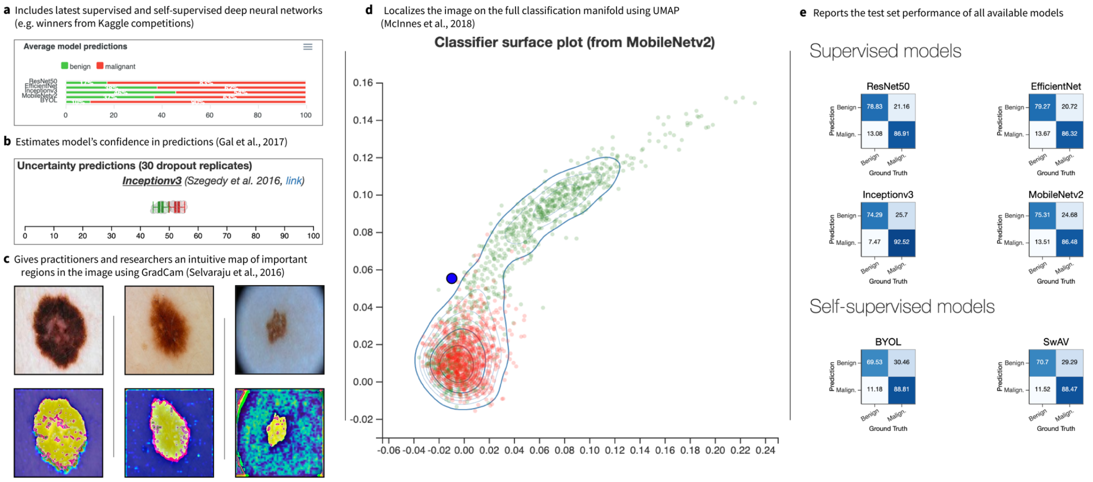

a Includes latest supervised and self-supervised deep neural networks (e.g. winners from Kaggle competitions)
Average model predictions
benign
malignant
ResNet50
EfficientNet
Inceptionv3
MobileNetv2
BYOL
17%
83%
38%
62%
46%
54%
37%
63%
10%
90%
0
20
40
60
80
100
b Estimates model's confidence in predictions (Gal et al., 2017)
Uncertainty predictions (30 dropout replicates)
Inceptionv3 (Szegedy et al. 2016, link)
0
10
20
30
40
50
60
70
80
90
100
c Gives practitioners and researchers an intuitive map of important regions in the image using GradCam (Selvaraju et al., 2016)
d Localizes the image on the full classification manifold using UMAP (McInnes et al., 2018)
Classifier surface plot (from MobileNetv2)
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-0.02
-0.06 -0.04 -0.02 0.00 0.02 0.04 0.06 0.08 0.10 0.12 0.14 0.16 0.18 0.20 0.22 0.24
e Reports the test set performance of all available models
Supervised models
ResNet50
Prediction
Benign
Malign.
78.83
21.16
13.08
86.91
Benign
Malign.
Ground Truth
EfficientNet
79.27
20.72
13.67
86.32
Inceptionv3
74.29
25.7
7.47
92.52
MobileNetv2
75.31
24.68
13.51
86.48
Self-supervised models
BYOL
69.53
30.46
11.18
88.81
SwAV
70.7
29.29
11.52
88.47

Figure 5.23: Screenshots of the main features of our DUNEScan web server.
(a) Average model predictions for a given skin lesion image (malignant or benign) provided by the six available CNN models, (b) boxplots showing uncertainty of model predictions, (c) Grad-CAM gradient saliency plot of most important lesion features, (d) classification manifold from the MobileNetv2 features, (e) confusion matrices computed over the test set for all six CNN models.

Table 5.10: Class prediction probabilities obtained for various images by the different CNN models available in DUNEScan.
The predictions are represented as probability of malignancy, p(malignancy). The probability of benignancy can be obtained by 1-p(malignancy). The names in the Image field are short-hand acronyms used for referencing. The Image Identifier corresponds to the ISIC image identifier. EffNet, Incept and MobNet are abbreviations for EfficientNet, InceptionV3 and MobileNetV2, respectively.

Image	Image Identifier	ResNet50	EffNet	Incept	MobNet	SwAV	BYOL
MEL1	ISIC_0024482	0.95	0.81	0.81	0.81	0.96	0.01
MEL2	ISIC_0024751	1.00	0.91	0.26	0.95	0.96	0.07
NV1	ISIC_0024320	0.00	0.48	0.27	0.12	0.36	0.03
NV2	ISIC_0024334	0.02	0.27	0.66	0.36	0.03	0.06
NV3	ISIC_0024307	0.45	0.43	0.53	0.58	0.03	0.10
BKL1	ISIC_0024337	0.30	0.09	0.16	0.12	0.05	0.47

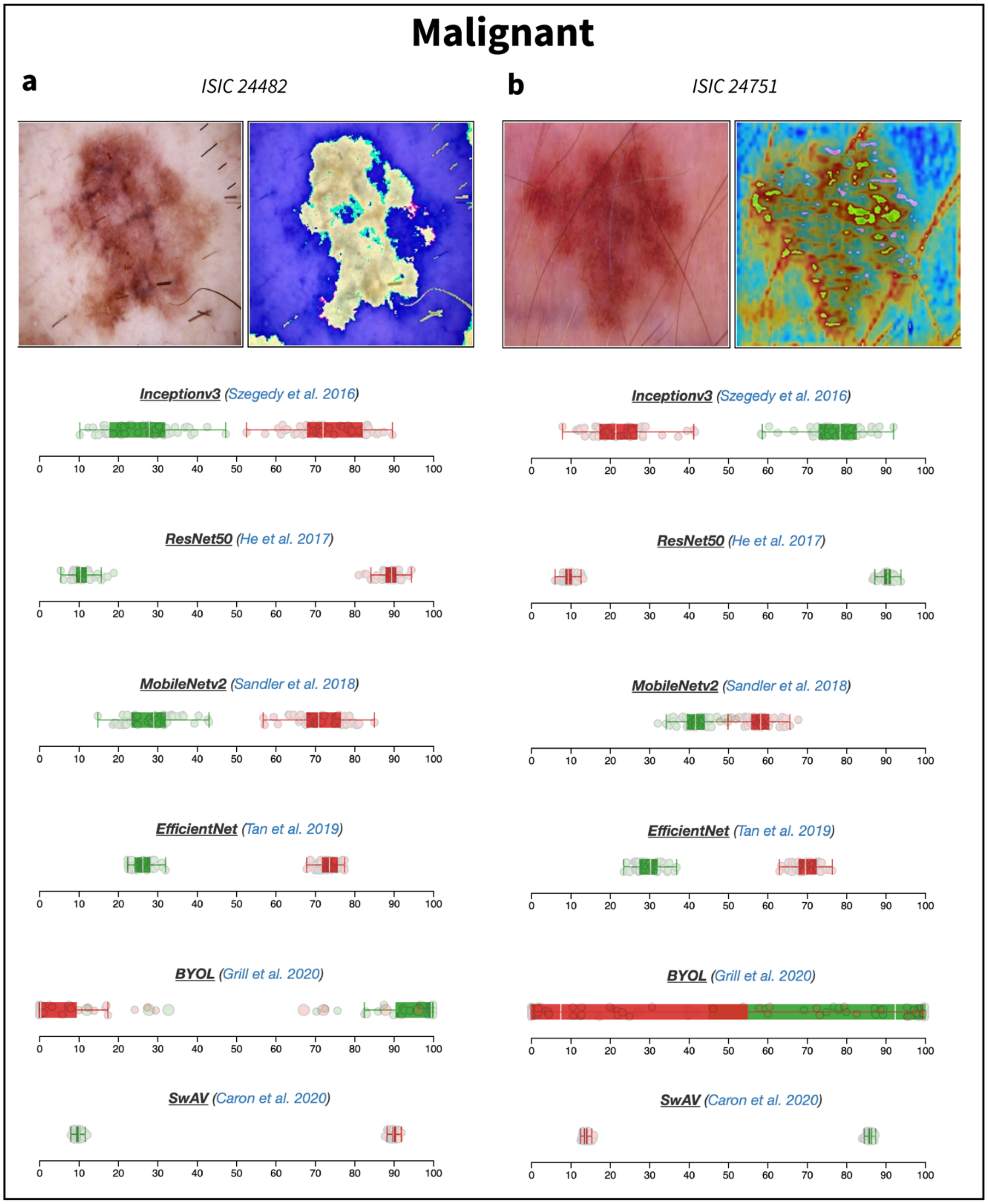


Figure 5.24: Boxplots representing uncertainty estimates for two malignant lesion. The original images of the malignant MEL skin lesions ISIC_0002482 (MEL1, a) and ISIC_0024751 (MEL2, b) are presented at the top-left of each panel. The Grad-CAM output is presented on the top-right part of the panels. In both panels the boxplots provided by the six CNN models available on DUNEScan are presented bellow for the corresponding skin lesion images.

Interestingly, the InceptionV3 model again produces an outlier result with the NV2 image (ISIC_24334, see Figure 5.25b). In this case, all other algorithms predict that the lesion is likely benign (all producing a probability of malignancy below 0.36), whereas InceptionV3 predicts that the lesion is malignant with a probability of 0.66 (see Table 5.10). In this case, the two models predicting the lesion to be benign with the highest probabilities, ResNet50 (0.98) and SwAV (0.97), have the tightest prediction distribution, whereas those of both InceptionV3 and MobileNetv2 are broad and overlapping (see Figure 5.25b). The distributions of the prediction probabilities obtained with EfficientNet are intermediate in size, but do not overlap. Based on these results, by relying on the models producing predictions with higher confidence (ResNet50, SwAV and EfficientNet), one could conclude that the image is indeed benign.

From the sample of NV images tested, it seems that the models have a difficulty producing a consensus benign prediction with high probability and confidence. Nevertheless, for most NV images, such as NV1 (ISIC_24320), an overall convincing set of benign prediction probabilities (all 0.52 or greater) are obtained from all models (see Table 5.10). The EfficientNet which produces the 0.52 probability is clearly unable to assign the lesion image to one class over the other. This is clearly illustrated by the fact that all replicate prediction probabilities for both benign and malignant classes overlap with a mean near 0.50 (see Figure 5.25a). All other models, which give higher benign prediction probabilities have varying levels of confidence based on the corresponding boxplots (see Figure 5.25a).

Since MEL and NV lesions often appear to be visually similar, this may explain why in some cases most of the models have a difficulty in favoring one class over the other. For example, with the NV3 image (ISIC_24307, Figure 5.26a) most models output predictions close to 0.50 for both classes (see Table 5.10). Interestingly,

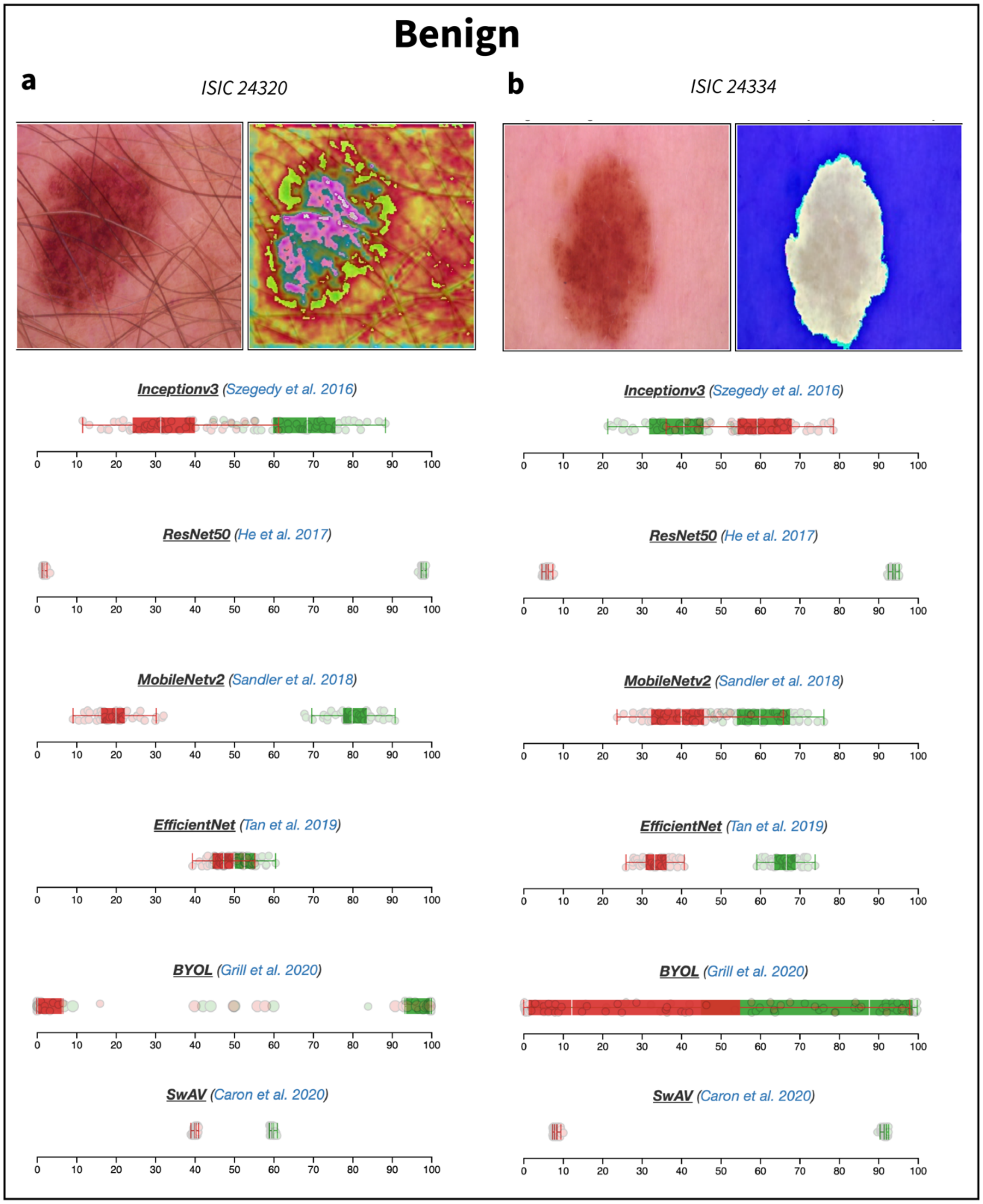


Figure 5.25: Boxplots representing uncertainty estimates provided for a first pair of benign lesions.

The original images of the benign NV skin lesions ISIC_00024320 (NV1, a) and ISIC_0024334 (NV2, b) are presented at the top-left of each panel. The Grad-CAM output is presented on the top-right part of the panels. In both panels the boxplots provided by the six CNN models available on DUNEScan are presented bellow for the corresponding skin lesion images.

with this image, only SwAV classifies the lesion as benign with a high probability (0.97) and confidence (see Table 5.10 and Figure 5.26a).

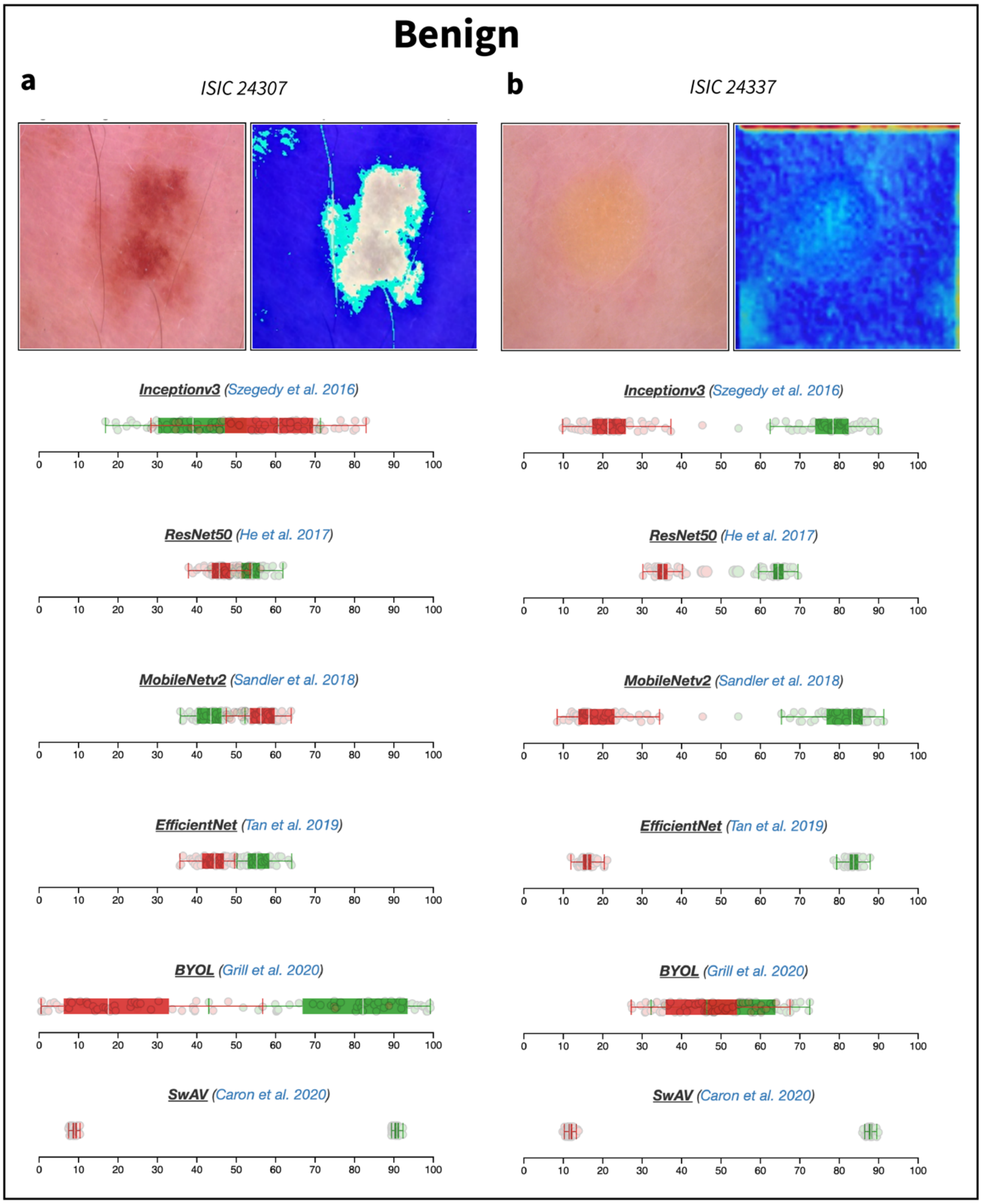

Benign
a
ISIC 24307
b
ISIC 24337
Inceptionv3 (Szegedy et al. 2016)
ResNet50 (He et al. 2017)
MobileNetv2 (Sandler et al. 2018)
EfficientNet (Tan et al. 2019)
BYOL (Grill et al. 2020)
SwAV (Caron et al. 2020)
0
10
20
30
40
50
60
70
80
90
100

Figure 5.26: Boxplots representing uncertainty estimates for a second pair of benign lesions.
The original images of the benign NV skin lesion ISIC_00024307 (NV3, a) and the benign BKL skin lesion ISIC_0024337 (BKL1, b) are presented at the top-left of each panel. The Grad-CAM output is presented on the top-right part of the panels. In both panels the boxplots provided by the six CNN models available on DUNEScan are presented bellow for the corresponding skin lesion images.

Finally, we present the results obtained with the benign BKL1 image (ISIC_24337, Figure 5.26b), which has a clearly different appearance to those of NV and MEL lesions. In this case, all models except BYOL predict the lesion to be benign with a probability of 0.70 or greater (Table 5.10). We obtain dispersed (but non-overlapping) replicate prediction probability distributions with the InceptionV3 and MobileNetv2 models (see Figure 5.26b), suggesting that the overall predictions that the lesion is benign (0.84 and 0.88, respectively, see Table 5.10), may not be highly accurate. However, based on Figure 5.26b, the predictions from all other models, except BYOL, appear to be trustworthy. The results presented in Table 5.10 provide a representative sample for a handful of malignant and benign skin lesions. The confusion matrices for all six models computed for the entire test set can be found in Figure 5.23.

Overall, it appears that most models, with the exeption of BYOL, are capable of giving valuable predictions in most test cases. In the case of BYOL, which has difficulty giving accurate malignancy predictions, additional training with adjusted parameters is likely required to obtain better performance. Nonetheless, when ignoring BYOL predictions, the application may be useful in assisting practitioners in the analysis of skin lesion images.

CHAPTER VI

DISCUSSION

Classification of skin cancer images may be considered a challenging computer vision task because the characteristics of some images belonging to the different classes do not appear to be clearly distinct (*i.e.* to the naked eye images from one class can be very similar looking to those of another). In the recent years, relatively large, novel annotated skin lesion image datasets (*e.g.* ISIC2019) have become publicly available. In the case of the ISIC2019 dataset the classification is made more difficult due to the large imbalance between the number of images representing each class. Some classes, such as the DF, VASC and SCC, are significantly under-represented in the dataset.

Class imbalance generally results in the most abundant classes having more influence on the weight adjustment in the convolution steps of CNN models during training and therefore these become better recognized by the classifier. In contrast, under-represented classes become less accurately differentiated from one another. In order to overcome the problem of class imbalance, a similar approach to that used by Gessert *et al.* (2020) was used. A weighted cross-entropy loss function was used where a greater weight is placed on images of poorly represented classes, and, as a consequence, miss-classification of one of these images has more impact on the fine-tuning of the CNN by back-propagation than miss-classification of an image from the more abundant classes.

After testing some of the more recent state-of-the-art CNN architectures pre-trained on the ImageNet dataset, we observed that all the models tested achieved broadly similar results. All models achieved best average WACCs between 0.61 and 0.66 with the ISIC2019 dataset (Table 5.1), which represents rather poor overall classification accuracy. From those tested, the best performing models were EfficientNet B4 and Inception V3. These models achieved best average WACCs of 0.6632 ±0.0108 and 0.6600 ±0.0119. It is likely that better results could have been achieved by one of the larger versions of the EfficientNet series, but the possible benefit of such models would have been offset by greater demands on computing resources. We therefore selected to use the EfficientNet B4 model for the remainder of our experiments.

As shown in this study, it appears that classification can be made even more difficult if images originate from different sources and may not have been preprocessed in an identical fashion by the dataset curators (*i.e.* some forms of batch effect). This is clearly illustrated by the fact that a much greater best average WACC can be achieved with the BCN (0.8933 ±0.0096) and HAM (0.8346 ±0.0129) datasets when they are analysed separately or together (0.8747 ±0.0079) as compared to that achieved when the MSK dataset is included (Table 5.1). This clearly shows that, although the number of images used for training is generally considered to be an important factor determining how well a CNN will perform in a classification task, variation in some general underlying composition of the images can have a large negative effect in a classification task.

A key feature of a good CAD classification model to separate skin lesion images into different disease classes is to be able to generalize well across images originating from different sources. In order to achieve this, perhaps a better preprocessing pipeline applied prior to training and classification would be required. This would be an avenue worth investigating. In addition, in this study the weights of the pretrained CNN architectures tested were simply readjusted by using batch normalization when re-training the networks with skin lesion images. It may be worthwhile to apply a more classical transfer-learning strategy by which the weights of a proportion of the convolution layer blocks are re-adjusted by fine-tunning. This may likely improve the performance of models by helping them to identify features that better separate the skin lesion image classes. However, in our study we chose to investigate if we could separate images based on some general traits or inherent characteristics into subgroups of images in order to improve the classification efficiency. To test this we chose to focus our study on the BCN image dataset since this dataset contains the largest number of images amongst the three sub-datasets composing the ISIC2019 dataset. In addition, the rarer disease classes are better represented in this dataset (*i.e.* the classes represent a greater proportion of the dataset) than in the HAM dataset, the second largest sub-dataset.

In all attempts made to subdivide the BCN dataset we used the K-means algorithm to produce clusters based on the Euclidian distance metric between image vectors produced by different strategies. With this algorithm, we can argue that the best results were obtained with vectors of features extracted from the images by the ResNet50 CNN architecture pretrained on ImageNet. With this approach, when classifying the images into eight classes with the EfficientNet B4 model, we achieved higher best average WACC scores with some clusters (best 0.8560 ±0.0037, Table 5.7) than with datasets of the same size composed of randomly selected BCN images (best 0.8012 ±0.0037, Table 5.7). However, none of the clustering approaches used were able to produce a subgroup of images with which we could outperform models trained with the full BCN dataset (0.8933 ±0.0096, Table 5.1).

Similar results were obtained when we tried classifying the images in a smaller number of classes (Table 5.9). In this case we used the same clusters generated by K-means with the ResNet50 extracted feature vectors as datasets to do binary classification between the MEL disease class and all other classes combined. EfficientNet B4 was found to achieve better best average WACCs with some clusters (best 0.9297 ±0.0047) than with the same-sized random BCN datasets (best 0.8940 ±0.0050). However, in the same classification approach we again achieved a significantly better best average WACC with the full BCN dataset (0.9392 ±0.0057).

These data clearly indicate that in the case of the BCN dataset, maintaining a higher number of images used for training is more important than separating the images based on what are likely relatively small general differences. This suggests that since the BCN images were likely all obtained from a common source and, perhaps more importantly, were all preprocessed in a similar fashion by the dataset curators, the images are likely generally homogenous. It may also be possible, however, that the methods used simply do not allow us to detect the best characteristics to segregate the images. In support of this view, it is important to note that none of the selected approaches used to produce image clusters were able to efficiently differentiate between MSK images and those from the HAM or BCN datasets (Tables 5.2, 5.3 and 5.8).

Since the various K-means clustering approaches used were unable to segregate the MSK images from those of the other datasets, it is not yet clear exactly why this dataset appears to create noise in training and classification (Table 5.1). As described previously, the MSK images seem to be the odd ones out and reduce the general classification accuracy when they are included in a classification task along with the HAM and BCN datasets. Therefore, one would expect that these images have underlying characteristics that make them, in some way, generally different from the HAM and BCN images and cause the EfficientNet B4 model to underperform.

However, it is also possible that only a sub-set of the MSK images are problematic either by being outliers for particular disease classes or by having different inherent characteristics not distinguished by the clustering approaches used. It is possible that other clustering algorithms such as those based on density (*e.g.* HDBscan (McInnes *et al.*, 2017), OPTICS (Ankerst *et al.*, 1999)) rather than distance (*e.g.* K-means) may help to segregate the images in a more efficient manner.

Nonetheless, since some visible improvement in classification accuracy was observed with the image clusters produced by K-means with ResNet50 extracted feature vectors, it would be interesting to apply a similar approach to a larger dataset. For example, if this approach was used on the combined BCN and HAM datasets, larger clusters similar in size to the BCN or HAM datasets (*i.e.* 10000 images) would be produced and, potentially, results obtained for classification of these larger clusters may surpass the accuracy observed with either of these individual original datasets.

It may also be worthwhile to implement the approach used by Bakkouri *et al.* (2019) with the ISIC2019 dataset. As described previously their approach consisted in using several older fine-tuned CNN models (*i.e.* VGG16, ResNet18 and DenseNet121). Feature maps produced by these different models were fused together and further convoluted. The resulting model trained with the HAM dataset yielded an average accuracy of 0.981. It is likely that the model produced may not generalize very well with images from other sources than the HAM dataset (*i.e.* it may be overspecialized), but by training a similar model with the ISIC2019 dataset we may achieve better results than those obtained with the different models tested individually in this study. Also, by using a similar approach with more recent state-of-the-art CNN models we may be able to produce a better performing and better generalizing model.

Interestingly, recently Tian *et al.*(2021) used a broadly similar approach to what we tried to do with the a completely different dataset. They used a self-supervised approach to extract features from a very large dataset containing many classes of images and used K-means to cluster the images into subgroups based on the features produced by the self-supervised model. Expert classifiers were then trained and learned to classify images from the different subgroups produced by K-means. Finally, the expert classifiers produced with the different subgroups were distilled (fused) together to produce a model that achieves state-of-the art results with the ImageNet dataset (Tian *et al.*, 2021).

Intuitively, using a self-supervised approach may be a better strategy than using a pretrained CNN (*e.g.* ResNet50) to detect underlying differentiating characteristics between images from different datasets or within a dataset when these characteristics are unknown or poorly defined. Self-supervised learning learns representations from unlabelled images (Kolesnikov *et al.*, 2019) and these representations may then be more informative that the ImagNet-learned features extracted by a pretrained network. In a similar fashion to the extracted features, the image representations could be clustered together into a potentially small number of sub-groups that would allow more accurate classification with a traditional classifier (*i.e.* supervised learning).

Interestingly, DeepCluster (Caron *et al.*, 2018) and Deep K-means (Fard *et al.*, 2020), new, recently repotted clustering methods, were specifically developed to cluster data based on the parameters and features assigned by a neural network These methods can be implemented jointly with unsupervised CNN models and assist them in learning useful representations. These methods are likely to yield greater success than the simple clustering strategies used in this study and the implementation of such methods to skin lesion image classification is another avenue which would clearly be worth exploring.

Finally, in the last part of our study we describe DUNEscan (Mazoure *et al.*, 2022), a new CAD application for skin lesion images. This new application addresses some drawbacks of other currently available applications: 1) it is an open-access application which utilizes models solely trained on publicly available skin lesion images, and 2) it implements methods that evaluate the uncertainty of the predictions made by the various models used. This latter feature allows users to get a clear statistical metric that indicates how trustworthy is the prediction made by each model.

The DUNEscan application is a valuable tool to assist specialists in the diagnosis of skin lesion images. However, as shown in our results, the fact that in some cases, the different models used give conflicting predictions, can make these hard to interpret. For this reason, it would be helpful to investigate if using an ensemble approach to determine if the predictions made by some of the models could be combined to produce a single, more accurate prediction. Such a strategy would make the results produced by the application much easier to interpret and thus make the use of the application more attractive.

CHAPTER VII

CONCLUSION

In conclusion, this study has shown that not only the sample size is important to produce highly accurate CNN based classifiers, but the homogeneity of the data also plays an important part. Optimal preprocessing may be able to improve the homogeneity of the data, but it remains that segregating the data based on unseen inherent characteristics may be able to improve the results. However, the detriment to such an approach is that the sample size becomes smaller in all subgroups produced.

The methods used in this study to separate the data by K-means using image pixel vectors or feature vectors produced by feature extraction with a model pretrained on ImageNet, show signs that clustering may be a useful approach. However, other approaches including, the use of alternative unsupervised classifiers to K-means, Deep K-means and/or using self-supervised feature extraction are avenues worth exploring and may yield better results.

As presented in the last section the some of the pretrained models tested in this study, although they do not perform very well with the ISIC2019 dataset as a whole, can still be useful when considered together to produce an application (*i.e.* DUNEScan) that should help medical specialists to classify skin lesion images.

The main feature of DUNEScan is an intuitive estimation and visualization of uncertainty for the predictions made by the selected state-of-the-art classifiers used. Uncertainty estimates are reported via boxplots of dropout replicates, Grad-CAM highlighting of regions of interest on the input image, as well as the projection of the input image onto the MobileNetv2 classification manifold. Thus, DUNEScan provides valuable information for bioinformaticians, dermatologists and health practitioners, looking for an accurate skin cancer diagnosis.

REFERENCES

Abbott, L. M. et Smith, S. D. (2018). Smartphone apps for skin cancer diagnosis: Implications for patients and practitioners. *Australasian Journal of Dermatology*, *59*(3), 168–170.

Ankerst, M., Breunig, M. M., Kriegel, H.-P. et Sander, J. (1999). Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, *28*(2), 49–60.

Anwar, S. M., Majid, M., Qayyum, A., Awais, M., Alnowami, M. et Khan, M. K. (2018). Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, *42*(11), 226.

Bakkouri, I. et Afdel, K. (2019). Computer-aided diagnosis (cad) system based on multi-layer feature fusion network for skin lesion recognition in dermoscopy images. *Multimedia Tools and Applications*, 1–36.

Berseth, M. (2017). Isic 2017-skin lesion analysis towards melanoma detection. *arXiv preprint arXiv:1703.00523*.

Blundo, A., Cignoni, A., Banfi, T. et Ciuti, G. (2021). Comparative analysis of diagnostic techniques for melanoma detection: a systematic review of diagnostic test accuracy studies and meta-analysis. *Frontiers in medicine*, *8*.

Brinker, T. J., Hekler, A., Enk, A. H., Klode, J., Hauschild, A., Berking, C., Schilling, B., Haferkamp, S., Schadendorf, D., Holland-Letz, T. *et al.* (2019). Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task. *European Journal of Cancer*, *113*, 47–54.

Caron, M., Bojanowski, P., Joulin, A. et Douze, M. (2018). Deep clustering for unsupervised learning of visual features. Dans *Proceedings of the European conference on computer vision (ECCV)*, 132–149.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P. et Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*.

Codella, N., Rotemberg, V., Tschandl, P., Celebi, M. E., Dusza, S., Gutman, D., Helba, B., Kalloo, A., Liopyris, K., Marchetti, M. *et al.* (2019). Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*.

Combalia, M., Codella, N. C., Rotemberg, V., Helba, B., Vilaplana, V., Reiter, O., Carrera, C., Barreiro, A., Halpern, A. C., Puig, S. *et al.* (2019). Bcn20000: Dermoscopic lesions in the wild. *arXiv preprint arXiv:1908.02288*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. et Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. Dans *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dick, V., Sinz, C., Mittlböck, M., Kittler, H. et Tschandl, P. (2019). Accuracy of computer-aided diagnosis of melanoma: a meta-analysis. *JAMA dermatology*, *155*(11), 1291–1299.

Dorj, U.-O., Lee, K.-K., Choi, J.-Y. et Lee, M. (2018). The skin cancer classification using deep convolutional neural network. *Multimedia Tools and Applications*, *77*(8), 9909–9924.

Elkan, C. (2003). Using the triangle inequality to accelerate k-means. Dans *Proceedings of the 20th international conference on Machine Learning (ICML-03)*, 147–153.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M. et Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, *542*(7639), 115–118.

Fard, M. M., Thonet, T. et Gaussier, E. (2020). Deep k-means: Jointly clustering with k-means and learning representations. *Pattern Recognition Letters*, *138*, 185–192.

Gal, Y. et Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. Dans *international conference on machine learning*, 1050–1059. PMLR.

Gessert, N., Nielsen, M., Shaikh, M., Werner, R. et Schlaefer, A. (2020). Skin lesion classification using ensembles of multi-resolution efficientnets with meta data. *MethodsX*, *7*, 100864.

Gessert, N., Sentker, T., Madesta, F., Schmitz, R., Kniep, H., Baltruschat, I., Werner, R. et Schlaefer, A. (2018). Skin lesion diagnosis using ensembles, unscaled multi-crop evaluation and loss weighting. *arXiv preprint* *arXiv:1808.01694*.

Grandini, M., Bagli, E. et Visani, G. (2020). Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G. *et al.* (2020). Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Han, S. S., Kim, M. S., Lim, W., Park, G. H., Park, I. et Chang, S. E. (2018). Classification of the clinical images for benign and malignant cutaneous tumors using a deep learning algorithm. *Journal of Investigative Dermatology*, *138*(7), 1529–1538.

He, K., Zhang, X., Ren, S. et Sun, J. (2016). Deep residual learning for image recognition. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hu, J., Shen, L. et Sun, G. (2018). Squeeze-and-excitation networks. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

Huang, G., Liu, Z., Van Der Maaten, L. et Weinberger, K. Q. (2017). Densely connected convolutional networks. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. et Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. Dans *Proceedings of the 22nd ACM international conference on Multimedia*, 675–678.

Karimkhani, C., Green, A. C., Nijsten, T., Weinstock, M., Dellavalle, R. P., Naghavi, M. et Fitzmaurice, C. (2017). The global burden of melanoma: results from the global burden of disease study 2015. *British Journal of Dermatology*, *177*(1), 134–140.

Kaufman, L. et Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.

Kawahara, J., Daneshvar, S., Argenziano, G. et Hamarneh, G. (2018). Seven-point checklist and skin lesion classification using multitask multimodal neural nets. *IEEE journal of biomedical and health informatics*, *23*(2), 538–546.

Kolesnikov, A., Zhai, X. et Beyer, L. (2019). Revisiting self-supervised visual representation learning. Dans *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1920–1929.

Koonce, B. (2021). Resnet 50. In *Convolutional Neural Networks with Swift for Tensorflow* 63–72. Springer.

Krizhevsky, A., Sutskever, I. et Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Dans *Advances in neural information processing systems*, 1097–1105.

Lacy, K. et Alwan, W. (2013). Skin cancer. *Medicine*, *41*(7), 402–405.

Lee, H. D., Mendes, A. I., Spolaor, N., Oliva, J. T., Parmezan, A. R. S., Wu, F. C. et Fonseca-Pinto, R. (2018). Dermoscopic assisted diagnosis in melanoma: Reviewing results, optimizing methodologies and quantifying empirical guidelines. *Knowledge-Based Systems*, *158*, 9–24.

Mazoure, B., Mazoure, A., Bédard, J. et Makarenkov, V. (2022). Dunescan: a web server for uncertainty estimation in skin cancer detection with deep neural networks. *Scientific Reports*, *12*(1), 1–10.

McInnes, L., Healy, J. et Astels, S. (2017). hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, *2*(11), 205.

Mendoza, F. et Lu, R. (2015). Basics of image analysis. In *Hyperspectral imaging technology in food and agriculture* 9–56. Springer.

Menegola, A., Fornaciali, M., Pires, R., Bittencourt, F. V., Avila, S. et Valle, E. (2017). Knowledge transfer for melanoma screening with deep learning. Dans *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, 297–300. IEEE.

Moura, N., Veras, R., Aires, K., Machado, V., Silva, R., Araújo, F. et Claro, M. (2019). Abcd rule and pre-trained cnns for melanoma diagnosis. *Multimedia Tools and Applications*, *78*(6), 6869–6888.

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

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. et Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. et Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. Dans *Proceedings of the IEEE international conference on computer vision*, 618–626.

Simonyan, K. et Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition (2014). *arXiv preprint arXiv:1409.1556*, *3*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. et Wojna, Z. (2016). Rethinking the inception architecture for computer vision. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Tan, M. et Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. Dans *International Conference on Machine Learning*, 6105–6114. PMLR.

Tian, Y., Henaff, O. J. et Oord, A. v. d. (2021). Divide and contrast: Self-supervised learning from uncurated data. *arXiv preprint arXiv:2105.08054*.

Toader, M. P., Esanu, I. M., Taranu, T. et Toader, S. V. (2017). Utility of polarized dermoscopy in the diagnosis of cutaneous lupus erythematosus and morphea. Dans *2017 E-Health and Bioengineering Conference (EHB)*, 583–586. IEEE.

Tschandl, P., Rosendahl, C. et Kittler, H. (2018). The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, *5*, 180161.

Wise, J. (2018). Skin cancer: smartphone diagnostic apps may offer false reassurance, warn dermatologists.

Xie, S., Girshick, R., Dollár, P., Tu, Z. et He, K. (2017). Aggregated residual transformations for deep neural networks. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500.

Yu, Z., Jiang, X., Zhou, F., Qin, J., Ni, D., Chen, S., Lei, B. et Wang, T. (2018). Melanoma recognition in dermoscopy images via aggregated deep convolutional features. *IEEE Transactions on Biomedical Engineering*, *66*(4), 1006–1016.