Classifying Skin Lesions with Artificial Neural Networks

Juliana Negrini de Araujo
Faculty of Engineering, Environment and Computing, Coventry University
MSc Data Science and Computational Intelligence (ECT104) Stage 1
Coventry, United Kingdom
negrinij@uni.coventry.ac.uk

***Abstract*—Skin cancer is currently the most common type of cancer in the world. In the UK, near 150.000 new cases are registered every year. If diagnosed at Stage I, the patient survival rate is about 100%. With the increase of data and computational power availability, we are closer to build auto-diagnostics tools that provide reliable results to any person that possess a mobile phone. This would facilitate the treatment of the disease by providing easy access to diagnosis to a broader range of the population. Although there are technical and ethical concerns, this possibility is driving researchers and Institutions to the edge of technology. On this work, we present a study comparing three different Artificial Neural Network models to classify skin lesions using the HAM10000 dataset. To date, this dataset is richest set of data of its kind. Our best model achieved an accuracy of 78,6% using Convolutional Neural Networks, while Multilayer Perceptron and the pre-trained network ResNet-50 resulted in 72% and 73% respectively.**

Keywords—Neural Networks; Image Classification; Skin Cancer

I. Introduction

Skin cancer is currently the most common type of cancer in the globe. For the UK, the NHS reports that over a hundred thousand patients are diagnosed with some type of skin cancer each year [1]. According to the Cancer Research UK, melanoma skin cancer is the 20th most common cause of cancer death, being accountable for 2.400 deaths every year [2]. If diagnosed early (Stage I) the survival rate is near to 100%. The chances drop by half if compared with the survival rate of patients where the disease is detected at later stages (Stage III or IV). In contrast, [3] mentions that in the US the survival rate for patients that are diagnosed with an advanced stage of melanoma skin cancer is only 14%.

The study presented by [3] was one of the first attempts to use artificial intelligence to diagnose cancerous skin lesions as benign or malign. With their work, they demonstrated that the deep learning model was capable of detecting skin cancer with the same level of accuracy of 21 certified dermatologists. Since then, research efforts have increased, and better results were achieved [4] [5]. One of the main contributors to models improvement is the fact that more data has become available since the first study was released. The International Skin Imaging Collaboration (ISIC) is now the largest publicly available repository of skin lesions images. An annual challenge is organized by this institution, where the main goal is to promote awareness and advance current technologies in skin cancer detection.

The HAM10000 dataset was first introduced to the wide public during the ISIC 2018 classification challenge, that occurred during the annual MICCAI conference in Spain. The final ranking for the best performing models of this competition can be found in [6]. The first three positions were awarded to the team from MetaOptima Technology Inc. According to the company website, the winning models were trained on HAM10000 dataset combined with proprietary data and ISIC archive images from previous competitions [7]. The top accuracy of 88.5% was achieved by using the average results of 10 different models the researchers developed. The second and third places were awarded to their ensemble model and to the model using the transfer learning from seresneXt-50, with an accuracy of 88.2% and 87.1% respectively. The authors used CNN and pre-trained network architectures for the three submissions. In [8], the team that received 4th place has reported their results for the ISIC 2018 challenge. They compare the performance of 20 pre-trained architectures and one ensemble model. The ensemble is composed of seven different network architectures, using the average prediction of 54 different models. By using this approach accuracy of 85% was achieved. For individual model architectures, the best performance was achieved by SENet154 architecture with 81,7% while ResNet50 achieved an accuracy of 77,9%. Similarly, in [9] they also make use of ensemble methods and pre-trained architectures (Inceptionv4, ResNet-152, and DenseNet-161). The study states that the best result was achieved by using the average prediction of 15 CNN models with normalized multi-class accuracy of 80,3%. With this approach the team achieved 9th place.

For image classification tasks, Convolutional Neural Networks (CNN) are used most often. They are a particular type of Neural Network (NN) and were first introduced by [10], where the LeNet-5 model demonstrated better performance in the handwritten digits dataset than Support Vector Machines and K-nearest Neighbour algorithms. Twenty years later, variations of CNN's swap most of the top positions of several competitions.

On this work, we analyse the performance of three different NN models on the HAM10000 dataset. The work is described as follows. Section II defines the dataset in more details, as well as the models that are used. Experiments and results are demonstrated in Section III. Section IV is discussion and conclusion remarks.

II. Methodology

A. Research Data

The HAM10000 dataset is composed of 10.015 dermatoscopic images of pigmented skin lesions. The data was collected from Australian and Austrian patients. Two institutions participated in providing the images: Cliff Rosendahl in Queensland, Australia, and Medical University of Vienna, Austria. Seven classes are defined on this dataset, where some diagnosis were unified into one class for simplicity. A brief description of each class is shown in Table I and further details can be found in [11].

TABLE I - SUMMARY OF THE DATASET.

Class	Description	Number of samples	% of class samples
akiec	Actinic Keratoses (Solar Keratoses) and Intraepithelial Carcinoma (Bowen’s disease) skin lesions. Usually caused by sun damage and treated without the need for surgery.	327	3.27%
bcc	Basal Cell Carcinoma is the most common type of skin cancer. Treatment is required, but it rarely metastasizes.	514	5.13%
bkl	Seborrheic Keratoses, Solar Lentigo and Lichen-planus-like Keratoses (LPLK) benign samples. They usually require biopsy due to similarities with melanoma lesions.	1099	10.97%
df	Dermatofibroma lesions.	115	1.15%
nv	Class contains images extensive data on variants of benign neoplasms called Melanocytic Nevi.	6705	66.95%
mel	Malignant Melanoma. Surgical removal in the early stage of cancer can provide a cure.	1113	11.11%
vasc	This class includes vascular skin lesions such as Cherry Angiomas and Angiokeratomas. It is also a benign type of skin lesion, only treated if it causes discomfort to the patient.	142	1.42%

A sample image of each class is shown in Figure 1. Information regarding patient age, sex, lesion location and diagnosis is also provided with each image. There are blank fields or *unknown* information for some of the records. For our final models, we used only the images and the class to build our model. We did not notice a significant improvement in using additional patient information.

The dataset was divided into three sets: training (70%), validation (10%) and test set (20%). A stratified split was performed to ensure all classes are represented across the three splits. The size of the images was initially 450 x 600 pixels. They were reduced to 75 x 100 pixels due to data processing limitations.

We also make use of online data augmentation process to reduce the chances of overfitting. From [9] and [12] we concluded that the best strategies for this dataset are: vertical and horizontals flip, rotation, shear and scaling. Even though both studies mention the usage of colour variations such as brightness, our preliminary experiments showed that the accuracy was significantly reduced by using this policy.

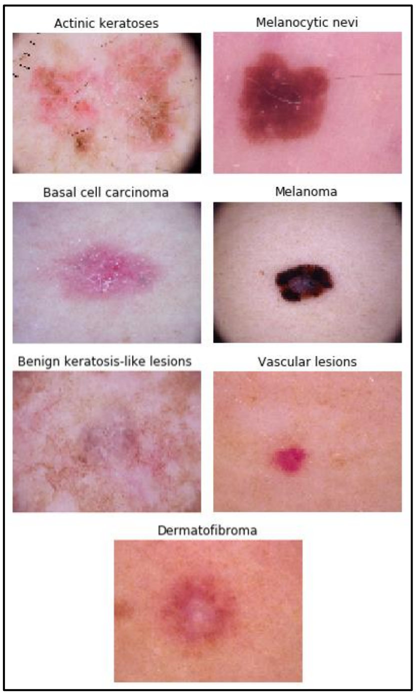


Figure 1 - Sample data from HAM10000 dataset.

B. *Unbalanced Data*

With regards to the unbalanced data, we analysed the usage of a balanced loss function. As mentioned by [8] and [9] this method allows the definition of higher weights for least represented classes. For example, for the *df* and *vasc* classes from Table I, a higher weight is given to correct classifications. This is to influence the model to learn these samples better. From our preliminary experiments, this approach has shown to decrease the model ability to identify the images, decreasing the accuracy dramatically properly. This encouraged us to find other methods.

The ISIC database is a live project, where dermatologists from all over the world can contribute and share the images. The data is analysed by the researches for classifying and uploading on their database. Looking at their full archive, we concluded that there were other skin cancer lesion images that were not included in this dataset. In total, we added an extra of 185 images of Melanoma (mel), Seborrheic Keratoses (bkl), Dermatofibroma (df) and Actinic Keratoses (akiec). For some classes, such as Dermatofibroma (df), only twenty images could be added. Special care was taken so that images were not repeated from the existing HAM10000 dataset. We call this dataset with additional images “HAM10000 +”. Our models were trained on both datasets, and their performance evaluated.

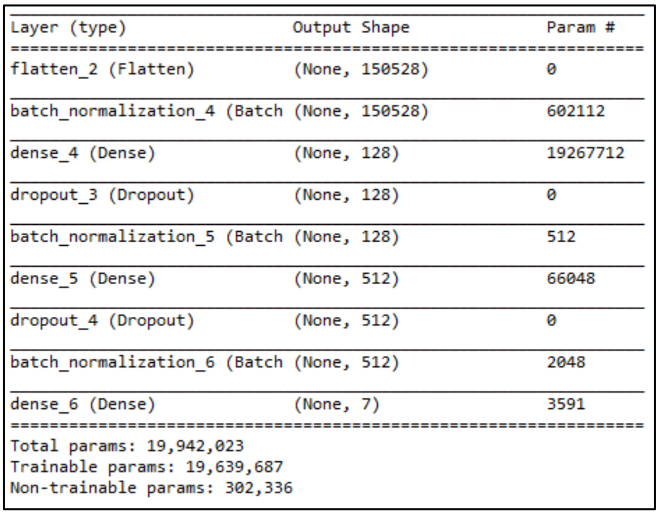
C. *Artificial Neural Networks Models*

1) *Deep Feedforward Networks*

These models represent the essence of deep learning and can also be known as feedforward neural network or multilayer perceptrons (MLPs). The *feedforward* refers to how the information is propagated within the network: from the input *x*, learning a function *f* and resulting in the output *y*. There is no feedback from the output to the model inputs. The *network* stands for the architecture of how the functions of the model are built together. Each function is represented as a layer of the network, the more layers the deeper the network. The first layer is usually called the input layer while the output layer is the final one. The remaining layers of the network are called *hidden layers* because the output of these layers is not given by the training data. Finally, *neural* is because the creation of this algorithm is inspired by how the human brain works. Each layer is formed by several units that can receive multiple inputs from the previous layer and compute its activation function. These units can be seen as neurons, and they define the width of each layer.

The correct combination of parameters of the neural network is crucial to achieve satisfactory results. If compared to other machine learning techniques (SVM or logistic regression, for example) this method requires a more extensive number of design decisions to be made and for this reason, the training and optimisation of the network is a challenging task.

For our MLP models, the following characteristics where trailed: number of layers, number of neurons and optimisers. We also tested the effectiveness of the dropout technique first introduced by [13] to overcome overfitting as well as batch normalisation proposed by [14]. Figure 2 shows the model summary for the best MLP model.



```
Layer (type)                 Output Shape              Param #
=================================================================
flatten_2 (Flatten)          (None, 150528)            0
_________________________________________________________________
batch_normalization_4 (Batch (None, 150528)            602112
_________________________________________________________________
dense_4 (Dense)              (None, 128)               19267712
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
batch_normalization_5 (Batch (None, 128)               512
_________________________________________________________________
dense_5 (Dense)              (None, 512)               66048
_________________________________________________________________
dropout_4 (Dropout)          (None, 512)               0
_________________________________________________________________
batch_normalization_6 (Batch (None, 512)               2048
_________________________________________________________________
dense_6 (Dense)              (None, 7)                 3591
=================================================================
Total params: 19,942,023
Trainable params: 19,639,687
Non-trainable params: 302,336
```

Figure 2 - MLP model summary.

A different number of layers were tested, but we could not see any improvements in additional layers, and it had a higher computational cost. The MLP network is rather simple in its architecture. We also tested with a different combination of neurons (32, 64, 128, 512 and 1024), the best results were with 128 and 512 as seen in Figure 2. With regards to optimisation, we tested Adam and Gradient Descent. Adam optimiser has performed best.

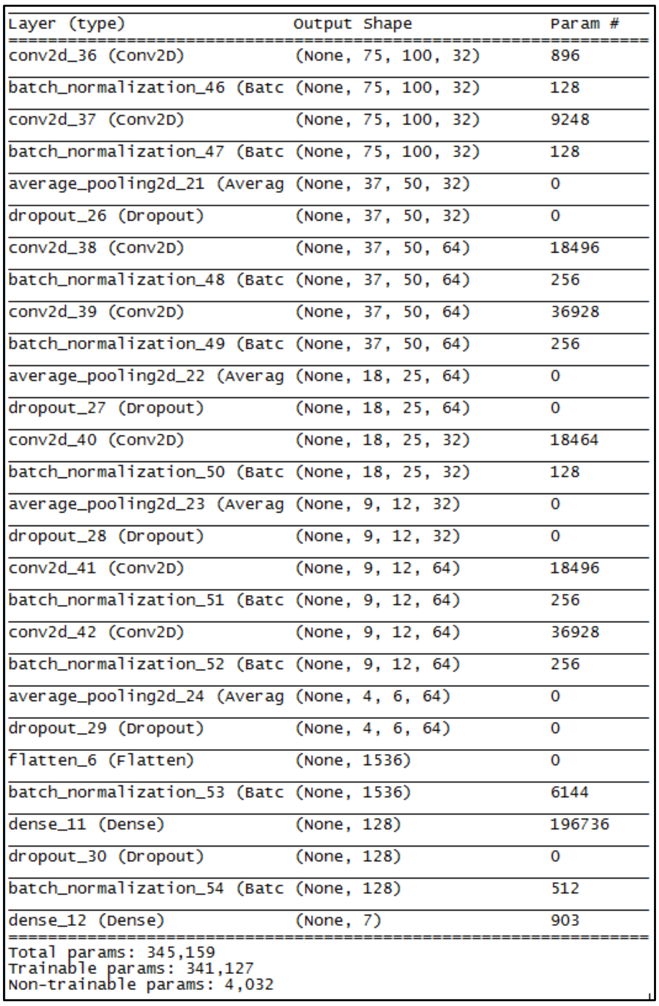
2) *Convolutional Neural Networks*

The Convolutional Neural Network is a specialised type of neural network, ideal for data that can be represented as a grid. CNN is most commonly used for image recognition tasks since this input can be perceived as a 2D grid of pixels. As described by [15], the CNN are neural networks that use in at least one of their layers the convolution operation. CNN's can handle larger inputs due to their unique characteristics. The convolution kernel is smaller than the input size, meaning that many input units interact with the kernel at the same time (sparse interaction). Common NN's have a one by one interaction, which results in thousands or millions of interactions if we consider an image processing task. CNN's also makes use of parameter sharing, i.e. instead of learning the weights between every element of the network, it learns a set of parameters that can be used across all input nodes.

Another typical operation performed in Convolutional Neural Networks is called pooling. The output of the pooling layer can be described as a summarised version of the input. Further information regarding pooling operation is described in [15]. A comparison between different pooling techniques is performed by [16] where they concluded that Max Pooling achieves a better performance than average pooling and attention pooling.

For our CNN model, we focused on the following parameters: number of convolutional layers, pooling strategies and tested different optimizers. We also implemented the use of reducing the learning rate if the plateau is achieved within 3 epochs .

From our preliminary tests, it was concluded that Adam optimiser provided better results if compared to Gradient Descent and RMSprop optimisers. A (2,2) and (5,5) spatial dimension for the pooling operations were tested, smaller (2,2) window performed better. For this dataset, Average Pooling provided better results than Max Pooling.



```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_36 (Conv2D)           (None, 75, 100, 32)       896
_________________________________________________________________
batch_normalization_46 (Batc (None, 75, 100, 32)       128
_________________________________________________________________
conv2d_37 (Conv2D)           (None, 75, 100, 32)       9248
_________________________________________________________________
batch_normalization_47 (Batc (None, 75, 100, 32)       128
_________________________________________________________________
average_pooling2d_21 (Averag (None, 37, 50, 32)        0
_________________________________________________________________
dropout_26 (Dropout)         (None, 37, 50, 32)        0
_________________________________________________________________
conv2d_38 (Conv2D)           (None, 37, 50, 64)        18496
_________________________________________________________________
batch_normalization_48 (Batc (None, 37, 50, 64)        256
_________________________________________________________________
conv2d_39 (Conv2D)           (None, 37, 50, 64)        36928
_________________________________________________________________
batch_normalization_49 (Batc (None, 37, 50, 64)        256
_________________________________________________________________
average_pooling2d_22 (Averag (None, 18, 25, 64)        0
_________________________________________________________________
dropout_27 (Dropout)         (None, 18, 25, 64)        0
_________________________________________________________________
conv2d_40 (Conv2D)           (None, 18, 25, 32)        18464
_________________________________________________________________
batch_normalization_50 (Batc (None, 18, 25, 32)        128
_________________________________________________________________
average_pooling2d_23 (Averag (None, 9, 12, 32)         0
_________________________________________________________________
dropout_28 (Dropout)         (None, 9, 12, 32)         0
_________________________________________________________________
conv2d_41 (Conv2D)           (None, 9, 12, 64)         18496
_________________________________________________________________
batch_normalization_51 (Batc (None, 9, 12, 64)         256
_________________________________________________________________
conv2d_42 (Conv2D)           (None, 9, 12, 64)         36928
_________________________________________________________________
batch_normalization_52 (Batc (None, 9, 12, 64)         256
_________________________________________________________________
average_pooling2d_24 (Averag (None, 4, 6, 64)          0
_________________________________________________________________
dropout_29 (Dropout)         (None, 4, 6, 64)          0
_________________________________________________________________
flatten_6 (Flatten)          (None, 1536)              0
_________________________________________________________________
batch_normalization_53 (Batc (None, 1536)              6144
_________________________________________________________________
dense_11 (Dense)             (None, 128)               196736
_________________________________________________________________
dropout_30 (Dropout)         (None, 128)               0
_________________________________________________________________
batch_normalization_54 (Batc (None, 128)               512
_________________________________________________________________
dense_12 (Dense)             (None, 7)                 903
=================================================================
Total params: 345,159
Trainable params: 341,127
Non-trainable params: 4,032
```

Figure 3 - Summary of the CNN model.

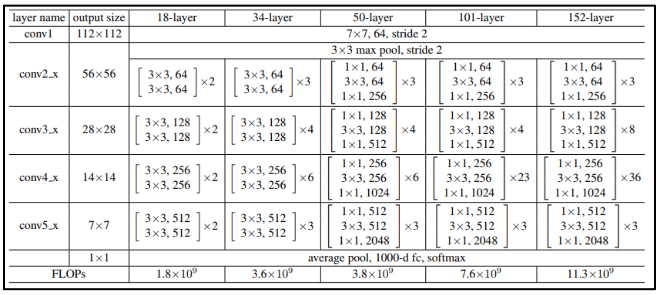
It is interesting to note the difference between the total number of parameters of the MLP and CNN model. From Figure 2 and Figure 3 we can see that the number of trainable parameters for the MLP is 50 times the CNN model.

3) *ResNet-50*

The idea behind the residual networks (ResNet) is an attempt to overcome a problem faced by many researchers when working with deep models. The training error starts to increase as more layers are added to the network. The explanation that this behaviour is caused by *overfitting* is not valid since training error should reduce in overly fitted models. Another hypothesis is that accuracy degradation occurs in deeper models because they are harder to optimize. The introduction of deep residual learning was done by [17]. On the same year, this proposed architecture won first place on the most prestigious image recognition competitions, ILSVRC 2015 (ImageNet) and COCO 2015.

While the NNs mentioned so far try to learn *H(x)*, the ResNet tries to learn the residual function denoted by *F(x) = H(x) – x*. The hypothesis is that the residual function could be easier to learn than *H(x)*. With this approach, they were able to stack more Convolutional layers without the accuracy degradation mentioned earlier.

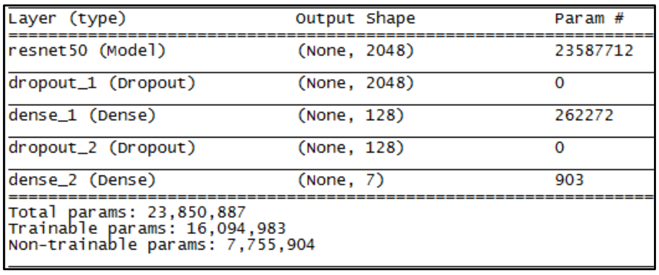
From our literature review, it was noticed that ResNet was always among the pre-trained architectures used on the ISIC 2018 challenge [6] [7] [8] [9]. We chose as our model the ResNet-50, which uses 50 convolutional layers. The architecture of ResNet-50 is shown in Figure 4.



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 4 - Available architectures of ResNet. In our work, ResNet-50 was used. Adapted from [17].

We reuse the weights of the pre-trained network and remove the output layer to adapt to our dataset. We achieved the best results when the last 25 layers were re-trained with our dataset. The final model summary is shown in Figure 5.



```
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
dropout_1 (Dropout)          (None, 2048)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               262272
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 903
=================================================================
Total params: 23,850,887
Trainable params: 16,094,983
Non-trainable params: 7,755,904
_________________________________________________________________
```

Figure 5 - ResNet-50 model summary.

D. *Training Routine*

The models were written in python using keras and tensorflow libraries. The batch size and number of epochs were experimentally defined. All models performed better with a batch size of 10, and the accuracy does not seem to improve after 50 epochs.

III. Experimental Results

Accuracy, Sensitivity (recall) and Precision are the metrics selected to measure the performance of our models. The usage of a single metric can be misleading since the classes are unbalanced. The final model results are defined as the average of three runs. Equations 1 through 3 shows the formulas for each metric.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Our results are shown in Table II. The models MLP, ResNet-50 and CNN were trained on the original HAM10000 dataset. The model CNN HAM10000 + was trained on the dataset with additional images. CNN's models have outperformed the other two models (MLP and ResNet-50) on all metrics. We can also see the improvements resulted from the usage of HAM10000+ dataset. Tables III through VI displays the results segmented by class for each of our models. Specificity and precision metrics are shown.

Table II – Results.

Metric	**Model**			
	MLP	**ResNet-50**	**CNN**	**CNN HAM10000 +**
Accuracy	72.81%	73.37%	77.96%	78.68%
Sensitivity	39.67%	38.57%	50.57%	59.19%
Precision	54.52%	53.90%	56.57%	68.43%

Table III – Sensitivity and Precision of the MLP model by class.

Class	**Sensitivity**	**Precision**
Actinic keratoses	33.3%	66.00%
Basal cell carcinoma	33.67%	53.67%
Benign keratosis-like	34.33%	54.33%
Dermatofibroma	20.00%	36.00%
Melanocytic nevi	94.33%	78.67%
Melanoma	19.00%	41.00%
Vascular Lesions	43.00%	52.00%

Table IV – Sensitivity and Precision of ResNet-50 model by class.

Class	**Sensitivity**	**Precision**
Actinic keratoses	9.33%	56.33%
Basal cell carcinoma	45.33%	60.33%
Benign keratosis-like	32.00%	47.33%
Dermatofibroma	0.00%	0.00%
Melanocytic nevi	95.67%	79.67%
Melanoma	28.33%	55.33%
Vascular Lesions	59.33%	78.33%

TABLE V – SENSITIVITY AND PRECISION OF CNN MODEL BY CLASS.

Class	Sensitivity	Precision
Actinic keratoses	37.33%	55.67%
Basal cell carcinoma	55.33%	66.67%
Benign keratosis-like	59.67%	54.00%
Dermatofibroma	6.67%	8.33%
Melanocytic nevi	95.00%	85.00%
Melanoma	21.00%	61.33%
Vascular Lesions	79.00%	65.00%

TABLE VI – SENSITIVITY AND PRECISION OF CNN HAM10000+ MODEL BY CLASS.

Class	Sensitivity	Precision
Actinic keratoses	51.33%	53.67%
Basal cell carcinoma	59.00%	76.33%
Benign keratosis-like	50.67%	59.67%
Dermatofibroma	52.33%	72.33%
Melanocytic nevi	95.00%	84.67%
Melanoma	32.67%	53.33%
Vascular Lesions	73.33%	79.00%

IV. DISCUSSION AND CONCLUSION

The possibility to obtain a reliable diagnose from a picture and app of the mobile phone is not yet our reality, but the usage of artificial intelligence is certainly helping the development of new tools to improve medical diagnosis. Although there are several ethical issues regarding reliability and privacy, the possibility to bring state of the art diagnosis to a never seen scale of patients is exciting.

With our work, we wanted to demonstrate the possibility of using different types of neural networks to classify skin cancer lesions. The primary challenge on this dataset is to accurately classify all the seven types of skin lesions, which becomes more difficult due to the high unbalance of samples for each class. Our CNN and CNN HAM10000+ models used less than ten layers and by tuning it was possible to achieve higher Accuracy, Sensitivity and Precision than other ISIC 2018 competitors. Analysing our results and the ISIC leaderboard [6], our CNN model can be compared to the 15th place in accuracy (balanced accuracy). This shows that experimenting with different pooling, architectures and optimisers can also make a big difference on the model behaviour. The usage of Dropout and dynamically reducing the learning rate has also shown to be beneficial for this dataset. Our experiment of adding extra images of the ISIC has proven to be extremely beneficial. By adding a small number of new images (185) of specific classes the sensitivity and precision raised 10%. Comparing the results for Dermatofibroma from CNN and CNN HAM10000+ models, the improvement is 1000%, and it was achieved by adding only 20 samples of this class to the dataset. Due to the time consumed to find unused images and add them to the original dataset, we could only add this limited number of new samples. With more time and more knowledge of the ISIC database for image retrieval, further improvements can be achieved.

The results obtained here are not as impressive if compared to the accuracy obtained by the top participants of ISIC 2018 [7] [8] [9]. This is due to a few reasons. Almost all participants used an ensemble method of several pre-trained networks. We have not applied ensemble methods on this work because the focus was on building and having a better understanding of NN's. Also, the top submissions used more complex pre-trained models that requires considerable computing power. Using the Kaggle web interface, we were able to run our models without consuming much time. This enabled more time to try different combinations and achieve good results in a reasonable time.

One of the drawbacks of this method for skin cancer detection is the limited number of samples of some of the classes. Even though this HAM10000 dataset is the most recent and complete dataset of this kind, it is challenging for the algorithms to learn in this scenario. A more consistent strategy for image data augmentation is required. From our experience, assigning different class weights for the loss function is not a reliable strategy, and online data augmentation has not improved the accuracy in such a representative manner. The best results were achieved by actually adding new samples to the dataset, proving how data is one of the main foundations of a good model. Within this topic, it is interesting to highlight the lack of diversity on this dataset. The limited number of samples for dark skin patients is a concern that has been raised by other researchers. If the model is not trained with this data, it is less likely to diagnose the correct disease according to the patient skin colour. The lack of diversity is mainly caused because African descendants have a lower tendency to develop skin cancer. This could mean that they would be left behind if this technology is used as the ground-truth for cancer diagnosis.

As a future work, it would be interesting to use more advanced data augmentation techniques to attack the unbalanced classes issue. We believe that improving the data would be an exciting route for investigation. The usage of Generative Adversarial Networks for data resampling has shown positive results for [18]. Also, the work from [19] proposes the usage of a different loss function. On their study, the usage of focal loss has improved the model performance since it allows the model to focus on misclassified examples. From the results of the challenge, we also concluded that a study and application of ensembling methods for higher accuracy is mandatory.

V. APPENDIX

The python codes are available in the following link https://bit.ly/2uDOUXi .

VI. BIBLIOGRAPHY

[1] NHS, "Overview: Skin Cancer," 2017. [Online]. Available: https://www.nhs.uk/conditions/non-melanoma-skin-cancer/. [Accessed 28 02 2019].

[2] C. R. UK, “Melanoma Skin Cancer Statistics,” 2016. [Online]. Available: https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/melanoma-skin-cancer#heading-One. [Accessed 28 3 2019].

[3] A.Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau and S. Thrun, “Dermatologist-level Classification of Skin Cancer with Deep Neural Networks,” *Nature,* vol. 542, pp. 115-118, 2 February 2017.

[4] B. Harangi, “Skin Lesion Classification with Ensembles of Deep Convolutional Neural Networks,” *Journal of Biomedical Informatics,* vol. 86, pp. 25-32, 2018.

[5] S. S. Han, M. S. Kim, W. Lim, G. H. Park, I. Park and S. E. Chang, “Classification of the Clinical Images for Benign and Malignant Cutaneous Tumors Using a Deep Learning Algorithm,” *Journal of Investigative Dermatology,* vol. 138, no. 7, pp. 1529-1538, 2018.

[6] ISIC, “ISIC 2018 Leaderboard,” 2018. [Online]. Available: https://challenge2018.isic-archive.com/leaderboards/. [Accessed 21 03 2019].

[7] T. D. Team, “MetaOptima Team Awarded The ISIC 2018 Challenge Disease Classification Prize,” 2019. [Online]. Available: https://www.dermengine.com/blog/metaoptima-awarded-isic-2018-challenge. [Accessed 26 03 2019].

[8] N. Gessert, T. Sentkerac, F. Madestaac, R. Schmitz, H. Kniep, I. Baltruschat, R. Werner and A. Schlaefer, “Skin Lesion Diagnosis using Ensembles, Unscaled Multi-Crop Evaluation and Loss Weighting,” *arXiv:1808.01694v1 [cs.CV]* , Aug. 2018.

[9] A. Bissoto, F. Perez, V. Ribeiro, M. Fornaciali, S. Avila and E. Valle, “Deep-Learning Ensembles for Skin-Lesion Segmentation, Analysis, Classification: RECOD Titans at ISIC Challenge 2018,” *arXiv:1808.08480 [cs.CV],* Aug. 2018.

[10] Y. LeCun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard and W. Hubbard, “Handwritten digit recognition: Applications of neural net chips and automatic learning,” *IEEE Communication,* pp. 41-46, 1989.

[11] P. Tschandl, C. Rosendahl and H. Kittler, “The HAM10000 Dataset, a Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions,” *Scientific data,* vol. 5, p. 180161, 2018.

[12] F. Perez, C. Vasconcelos, S. Avila and E. Valle, “Data Augmentation for Skin Lesion Analysis,” *arXiv:1809.01442v1 [cs.CV],* 2018.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research,* vol. 15, pp. 1929-1958, 2015.

[14] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *ICML'15 - 32nd International Conference on International Conference on Machine Learning*, Lille, France, 2015.

[15] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, Cambridge: MIT Press, 2016.

[16] V. Suárez-Paniagua and I. Segura-Bedmar, “Evaluation of Pooling Operations in Convolutional Architectures for Drug-Drug Interaction Extraction,” *BMC Bioinformatics,* vol. 19, no. 8, pp. 40-47, 2018.

[17] K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition,” *arXiv:1512.03385v1,* 2015.

[18] N. Esmaeilishahmirzadi and H. Mortezapour, “A novel method for enhancing the classification of pulmonary data sets using generative adversarial networks,” *Biomedical Research* , vol. 29, no. 14, pp. 3022-3027, 2018.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, “Focal Loss for Dense Object Detection,” *2017 IEEE International Conference on Computer Vision (ICCV),* pp. 2999-3007, 2017.