Multi skin lesions classification using fine-tuning and data-augmentation applying NASNet

Elia Cano [1], **José Mendoza-Avilés** [1], **Mariana Areiza** [1], **Noemi Guerra** [1], **José Longino Mendoza-Valdés** [1], **Carlos A. Rovetto** [Corresp. 1]

[1] Computer Science, Universidad Tecnológica de Panamá, Panama, Not Applicable, Panama

Corresponding Author: Carlos A. Rovetto
Email address: carlos.rovetto@utp.ac.pa

Skin lesions are one of the typical symptoms of many diseases in humans and indicative of many types of cancer worldwide. Increased risks caused by the effects of climate change and a high cost of treatment, highlight the importance of skin cancer prevention efforts like this. The methods used to detect these diseases vary from a visual inspection performed by dermatologists to computational methods, and the latter has widely used automatic image classification applying Convolutional Neural Networks (CNNs) in medical image analysis in the last few years.

This paper presents an approach that uses CNNs with a NASNet architecture to recognize in more accuracy way, without segmentation, eight skin diseases. The model was trained end-to-end on Keras with augmented skin diseases images from the International Skin Imaging Collaboration (ISIC). The CNN architectures were initialized with weight from ImageNet, fine-tuned in order to discriminate well among the different types of skin lesions, and then 10-fold cross-validation was applied.

Finally, some evaluation metrics are calculated as accuracy, sensitivity, and specificity and compare with other CNN trained architectures. This comparation shows that the proposed system offers higher accuracy results, with a significantly reduction on the training paraments. To the best of our knowledge and based in the state-of-art recompiling in this work, the application of the NASNet architecture training with skin image lesion from ISIC archive for multi-class classification and evaluated by cross-validation, represents a novel skin disease classification system.

Type of the Paper (Article, Review, Communication, etc.)

Elia E. Cano, José Mendoza-Avilés A., Mariana Areiza, Noemi Guerra, José Mendoza-Valdés and Carlos A. Rovetto
Computer Systems Engineering Department, Technological University of Panama, Panama City, Panama.

Corresponding Author:
Carlos A. Rovetto
Panama, Panama, Panama
Email address: carlos.rovetto@utp.ac.pa; Tel: +507-68370596

Multi skin lesions using fine-tuning and data-augmentation applying NASNet

Elia E. Cano, José Mendoza-Aviles A., Mariana Areiza, Noemi Guerra, José Mendoza-Valdes and Carlos A. Rovetto[1]

Computer Systems Engineering Department, Technological University of Panama, Panama City, Panama.
[1]carlos.rovetto@utp.ac.pa

Abstract

Skin lesions are typical symptoms of many diseases in humans, they also are indicative of many types of cancer worldwide. Increased risks caused by the effects of climate change and a high cost of treatment, highlight the importance of skin cancer prevention efforts like this. The methods used to detect these diseases vary from a visual inspection performed by dermatologists to computational methods, this last method has widely used automatic images classification applying Convolutional Neural Networks (CNNs) in medical image analysis in the last few years.

This paper presents an approach that uses CNNs with a NASNet architecture to recognize eight types of skin diseases in a more accuracy way. The model was trained end-to-end on Keras with augmented skin diseases images from the International Skin Imaging Collaboration (ISIC). The CNN architectures were initialized with weight from ImageNet, fine-tuned in order to discriminate well among the different types of skin lesions, and then 10-fold cross-validation was applied.

Finally, some evaluation metrics are calculated as accuracy, sensitivity, and specificity and compare with other CNN trained architectures. This comparation shows that the proposed system

offers higher accuracy results, with a significantly reduction on the training paraments. To the best of our knowledge and based in the state-of-art recompiling in this work, the application of the NASNet architecture training with skin image lesion from ISIC archive for multi-class classification and evaluated by cross-validation, represents a novel skin disease classification system.

Keywords: Skin cancer, Skin diseases, Convolutional Neural Network, Image analysis, NASNet.

1. Introduction

Skin cancer is frequent in the USA, Australia, and Europe [1] with 20 percent of Americans developing this kind of disease by the age of 70, 4 percent of all cancers in Asians, 5 percent in Hispanics and an annual cost of treating in the U.S. estimated at $8.1 billion, signifying skin cancer as a severe public health problems [2].

Mutations in the DNA of epidermal cells lead to out-of-control, abnormal growth and cause skin cells to multiply rapidly forming a malignant tumor, the main causes of that mutation are harmful ultraviolet (UV) rays and the use of UV tanning machines. The chief types of skin cancer are squamous cell carcinoma (SCC), basal cell carcinoma (BCC), Merkel cell carcinoma (MCC) and melanoma (MEL). Despite how deadly skin cancer is, with a 5-year survival rate, it can be up to 99% if diagnosed and treated early enough; however, a delayed diagnosis causes a decrease to 23% in the survival rate.

The diagnosis of skin cancer starts with a visual inspection of a dermatologist. Due to the nature of some lesion's types, a correct diagnosis is important [3], although the accuracy of the diagnosis is correlated to the professional experience. Using dermatoscopic images and visual inspection, dermatologists can achieve an accuracy of 75%-84% [4], [5]. Finally, biopsies can detect the malignancy of a skin growth, but they are also the most invasive techniques. This process takes approximately fifteen minutes. Despite being a short time, technological advances open the possibility of improving accuracy while reducing time and costs through image analysis. This work gives an approach to increase the accuracy and reduce the time response in non-invasive way.

Automatic image classification using Convolutional Neural Networks (CNN) has been widely used in the analysis of medical images [6], [7]; however, until 2012 with the work of [8] using the AlexNet architecture that increases the work used by these models for classification of images through CNNs. Other works have used convolutional architectures with important results [9]–[15].

Deep learning technique was applied in the search for automated lesion classification for unifying the dermatologist's professional experiences and supporting them in the diagnosis, convolutional neural network (CNN) training for the detection and classification of skin diseases is carried out using a set of data. This uses highly standardized dermatoscopic images that are acquired through a specialized instrument or histological images acquired through invasive biopsy and microscopy. To train the network, some authors operate with datasets from open-access dermatology repositories, others with repositories belonging to hospitals or clinics where samples are taken [3], [16], or a combination of the previous two. On the other hand, the images used for diagnosis can be taken using digital cameras [17]–[24] or the camera of a smartphone.

The authors in [3] trained a Inception v4 architecture to perform a binary classification between of melanoma and benign skin lesions. The results were compared with the opinion of 58 dermatologists with different levels of experience. The paper outcome is especially valuable because offers proof of the importance of computer aid, independent of the physician experience level.

One of the most significant advances in the field of skin disease classification and detection comes with the creation of ISIC Challenge in 2016 [1]. ISIC publishes the largest skin disease data set, divided by 14 classes. Some of the classes were merged or omitted as a result of the small number of images on it.

Since 2016 CNN was used in skin lesions classifications with several approaches based on the number of classes to classify (binary, multi-class), the way the CNN is used (Feature Extractor, end-to-end training and learning from scratch) [25] and some use the segmentation with a U-net before feeding the training model.

A CNN is used as a feature extractor when is pre-trained with a large dataset (ImageNet) and the fully connected layer is removed—usually, the data is augmented and normalized. In most of

the papers found with this approach, AlexNet architecture is used as feature extractor with a K-neighbor or support vector machine as classifier.

The images used to classify could come from their own source as in [26] with 399 photos taken with a standard camera—achieving a sensitivity of 92.1%, specificity of 95.18% and an accuracy of 93.64%. In contrast, in [27] and [28] use a public libraries as DermoFit [29] and ISIC dataset respectively.

The authors in [27] present a multi-classification performed by logistic regression with a final accuracy of 81.8% — the data is splinted in validation, training and test. In other hand, two-fold cross-validation is used in [28] for two task of binary classification— melanomas vs non-melanoma (accuracy of 93.1%) and melanoma vs atypical nevi (73.9% accuracy).

A widely used approach is transfer learning, were an architecture is initialized with the weight of another data and fine-tuned to fit the new dataset. As in [30] were the authors train a Inception v3 with 129,450 images from a private source and 3374 images obtained from dermatoscopic devices. Two problems of binary classification were tested, keratinocyte carcinomas versus benign seborrheic keratosis and malignant melanomas versus benign nevi. The outcome was present with the Area Under the Receiver Operating Characteristics metric— 0.96 for carcinomas and 0.94 for melanomas.

The authors in [31] proposed train a ResNets for a multi-classification of 12 skin lesions tested with the ASAN dataset (Average AUC 0.91 ± 0.01,sensitivity 86.4 ± 3.5 and specificity 85.5 ±3.2) and DermoFit [29](Average AUC 0.89 ± 0.01, sensitivity 85.1±2.2 and specificity 81.3 ±2.9) . Residual neural network is also used in [32] to evaluate three approaches multi-class classification (Melanoma, Seborrheic keratosis and Nevus), binary classification(Melanoma Vs Seborrheic keratosis and Nevus), and the ensemble approach. The latter approach got the best AUC results with Melanoma 85.40, Seborrheic keratosis 97.60 and average of 91.50 with the ISIC 2017 Challenge dataset.

Other architecture implemented with transfer learning is the VGGNet. One of the first examples of this is show in [33] with DermQuest archive classifying among 198 classes (accuracy 50.27 %) .After, the authors [34] modified VGGNet train with the ISIC 2016 Challenge dataset to

discriminate between malignant and benign. The best configuration achieved the accuracy of 81.33%, sensitivity 0.78 and precision 0.79.

In this paper, NASNet architecture is implemented to recognize 8 skin diseases more than of the majority previous cases. We can identify three types of cancer: Squamous Cell Carcinoma, Melanoma and Basal Cell Carcinoma. Also, our model can discriminate nevus, the most common kind of mole, of the Melanomas.

The rest of the paper is organized as follows: materials and methods are explained in section 2. In section 3 the results and discussion are presented. Finally, the conclusions and future works are given in section 4.

2. Materials and Methods

A formal statement of the problem, from an example-based learning problem or supervised learning problem, is the following one:

In this work a Softmax function is used, therefore, the equation

$$a^l = \sigma(z^l) \tag{1}$$

could be rewritten as

$$\sigma(z^l) = \frac{e^{z^l}}{\sum_{j=1}^{K} e^{z^j}} \tag{2}$$

being Σ^{z^l} a probability distribution that will center around the positions of the values, applying it to the largest entries [35].

Let X and Y be two metric spaces: X (skin image), Y (corresponding class label) and a (target) function y: X $\rightarrow$Y, specified only in the finite aggregate of points: y(X^1) ,…, y(X^8) , i.e. the labels of objects X^1,…, X^8 are known[36]. Where X is split to classes according to the skin disease

['AK','BCC','BKL','DF','MEL','NV','SCC','VASC']; after one-hot encoding was applied the skin disease classes could be noted as H_1,…, H_8, where

$$H_i = \{x \in X | \, y(x) = i\} \text{ at } i \in \{1,\ldots,8\}: X = \bigcup_{i=1}^{8} H_i \quad (3)$$

The target function y: X →Y, that discriminates well among the different class labels, is describes as the working process of a neural network in

$$a_j^l = \sigma\left(\sum_{k=1} w_{jk}^l a_k^{l-1} + b_j^l\right) \quad (4)$$

with these notations; the vector activation´s components a_j^l are represented as the sum over all neurons k in the $(l-1)$ layer in a matrix form, where the weight from each layer l defines as w^l with j and k are the representation of row and columns, respectively. The components of the bias vector are just the values b_j^l. Equation also can be rewritten in a compact vectorized

$$a^l = \sigma\left(\sum_m w^l a^{l-1} + b^l\right) \quad (5)$$

also written in terms of the weighted input, as [37]

$$a^l = \sigma(z^l) \quad (6)$$

In order to quantifies the error between predicted values $\hat{y}$ and expected values y a cost function $J(\hat{y},y)$ is applied.

$$J(\hat{y},y) = -\sum_{j=0}^{M}\sum_{i=0}^{N}\left(y_{ij} * \log\left(\hat{y}_{ij}\right)\right) \quad (7)$$

where the output of the function is given by

$$J(\hat{y},y) = \{1, \hat{y} \neq y \text{ or } 0, \hat{y} = y\} \quad (8)$$

The way to decrease an objective function $J(\theta)$ parameterized by the parameters of a

$$\{\theta = (\hat{y}, y) | \theta \in \mathbb{R}^d\} \quad (9)$$

through the gradient descent model by updating the parameters in the opposite direction of the gradient of the objective function $\nabla_\theta J(\theta)$ with respect to the parameters. The learning rate η determines the size of the steps we take to reach a (local) minimum. Depending on the amount of data, we trade off the accuracy of the parameter update and the time it takes to perform an update. Thus, updated is perform stochastic gradient descent (SGD) for every example $x^{(i)}$ and label $y^{(i)}$ for every mini batch of n training examples

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)}) \; for \; n = 30 \; [38] \quad (10)$$

In other words, the parameters updated was made by feeding the model with mini batches of 45 images—this amount of batches prevent an overload on the GPU memory. It take for the computer 975 steps to update all the parameters, this process was repeated until the error stop to decrease. The mini-batches method is chosen because reduce the variance of parameter updates, which can lead to more stable convergence; it can make use of highly optimized matrix optimizations common to the state-of-the-art deep learning libraries that make computing the gradient with respect to a very efficient mini-batch; and reduce the stored examples on the computer´s RAM.

Adaptive Moment Estimation (ADAM) was applied to calculate the learning rate η and store an exponentially decreasing average of the past gradients

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (11)$$

and the past square gradients

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (12)$$

[39] those are estimates of the first momentum (the mean) and the second moment (the uncentered variance), respectively. The m_t is where the past normalized gradient is recorded, called the first moment, and v_t refers to the adaptive gradient decrease displayed in the RMSprop [39], which in turn is called the second moment. As the authors of Adam explain, β_1 and β_2 refer to the decomposition rate, which are small due to the initial time steps, this causes them to be

biased towards zero[40]. Where g_t denotes the gradient at time step t. $g_{t,i}$ is then the partial derivative of the objective function w.r.t. to the parameter θ_i at time step

$$t: g_{t,i} = \theta_{\nabla} J(\theta_{t,i}) \quad (13)$$

ADAM performance akin to other optimizers as RMSprop, Adadelta in similar circumstances. In [40] shows Adam to slightly outperform RMSprop due to bias correction when optimization is ending and as gradients become more scattered. In the measure, Adam might be the best overall choice.

2.1. Workflow of the proposed systems.

To complete this project is used the methodology summarized in the following Figure 1:

Figure 1. Flow chart of the proposed systems.

2.2. Dataset

The dataset used for this project comes from the ISIC Training Challenge 2019 [41]–[43]. This dataset consists in 25,331 JPEG images of skin lesions, divided in Actinic Keratosis (AK), Squamous cell carcinoma (BCC), Benign keratosis (solar lentigo / seborrheic keratosis / lichen planus-like keratosis) (BKL), Dermatofibroma (DF), Melanoma (MEL), Melanocytic nevus (NV), Squamous cell carcinoma (SCC) and Vascular lesion (VASC).

2.3. Hair Removal

As part of the process of image pre-processing is imperative to remove the hair that appears in skin lesions images. The algorithms for hair removal on skin images have been widely studied, the most simple and efficient one is then carried out by [44] called DullRazor. This algorithm identifies the dark hair location through generalized grayscale morphological closing operation, after the hair

pixels shape is checked, they are replaced using bilinear interpolation and smooth by an adaptive median filter.

The complete process is depicting in the Figure 2 on the hair removal section.

Figure2. Hair removal process applied in one of the images from ISIC dataset.

2.4. Data Augmentation

Due to the class imbalance Data Augmentation is applied. This technique is used to increase the amount of data available in the classes. The proper use of this technique increases the generalization of the model. It also prevents overfitting, since increasing the number of variations in the data brings it closer to reality. With this additional data the model can learn, during training, properties such as contrast invariance, location invariance, rotational invariance, and the like.

Figure 3. Data-Augmentation applied in one of the images from ISIC dataset.

Data Augmentation settings applied to the dataset are described in Table 1 and the effects would be seen in the Figure 3. These methods are applied through the keras library with the ImageDataGenerator function. The process is also used to resize images to 224 x 224.

Some of the parameters that we pass through are:

rotation_range: range of degrees for random rotations

width_shift_range: the fraction of the total width that the image can be shifted by

height_shift_range: the fraction of the total height that the image can be shifted by

zoom_range=0.1: represents the fraction of the image that can be zoomed in or out

horizontal_flip=True: randomly flips the input horizontally

vertical_flip=True: randomly flips the input vertically

fill_mode='nearest': the specification to fill points outside the input limits.

Table 1. Augmentation details

2.5. Neural Network Architectures

The NASNet architecture is a convolutional neural network developed by an IA created by the Google Team in 2018. Their authors said, "Our model is 1.2% better in top-1 accuracy than the best human-invented architectures while having 9 billion fewer FLOPS – a reduction of 28% in computational demand from the previous state-of-the-art model". Due to the accuracy increase registered in the state-of-the-art and reduction of the computational demand, this architecture is applied in this project as a feature extractor..[24].

This architecture is composed of convolutional cells. The two main functions are Normal Cell and Reduction Cell, shown in Figure 4. The Reduction Cell returns a feature map height and width reduced by a factor of two. On the other hand, the Normal Cell returns a feature map with normal cells with the same input dimensions. The model used for this purpose was NASNet-A (4 @ 1056), where the number 4 represents the number of cell repeats, and 1056 corresponds to the number of filters in the penultimate layer of the network.

Figure 4. The two main functions of NASNet architecture, Normal Cell (A) and Reduction Cell (B). [24]

At the end of the last Normal_A_cell, global_max_pooling2d, global_avg_pooling2d, and flatten layers were fed. The dimensions of the tensors were reduced to Nx1056 in the first two layers, where the number of images the system is trained is noted as N. When we applied the filters, we found the largest number in the global_max_pooling2d layer and obtained the arithmetic average in global_avg_pooling2d. The output of the flatten layers was Nx51774. The outputs of these three layers were concatenated in the concatenate layer. This layer fed the dropout layer, which turned off some neurons to prevent overfitting of the network. Finally, it fed the dense layer, which offered the inference of the model through a $\sigma(z^l)$ function. These steps are depicted in Figure 5.

252

253 **Figure 5.** Representation of the steps of the extraction feature map and classification.

254 2.6. Evaluation Metrics

255 Through the confusion matrix we can obtain the productivity of the model during the training
256 or the development of a classification problem. This shows us in detail how many times the model
257 is wrong when making predictions. The number of correct and incorrect predictions is obtained by
258 counting values and separated them from each class. It gives us an idea not only of the mistakes
259 that the model makes but of the types of mistakes it makes. This matrix allows us to measure
260 Recall, Precision, Accuracy, and the AUC-ROC curve. This matrix describes the complete
261 performance of the model. Table 2 shows the data distribution for multiclass classification.

262 Table2. Data distribution for the multiclass classification confusion matrix

263 This matrix is composed of:

264 True Positive (TP): the observation is positive and was predicted to be positive.

265 True Negative (TN): the observation is negative and was predicted to be negative.

266 False Positive (FP): the observation is negative, but it was forecast as positive.

267 False Negative (FN): the observation is positive, but it was predicted as negative.

268

269 From the confusion matrix we compute the accuracy. It is the ratio of the number of correct
270 predictions to the total number of input simples. $Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$. It works well only
271 if there are equal numbers of samples belonging to each class.

272 Recall gives us the number of correct positive results divided by the number of all relevant
273 samples, where all samples should have been identified as positive. $Recall = \frac{TP}{TP + FN}$

274 Precision is a measure of exactness. It defines the probabilities of the number of correct positive
275 results to the number of positive results predicted by the classifier. $Precision = \frac{TP}{TP + FP}$

We use an F1 Score to know how precise and robust our classifier is. F1 Score is a balance between precision and recall. The range for the F1 Score is [0, 1]. The greater the F1 Score the better the performance of our model.

$$F1\ Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

To show the performance of the classification model in all the classification thresholds, we use the curve's ROC. This curve is plotted with the True Positive Rate $TPR = \frac{TP}{TP + FN}$ against the False Positive Rate $FPR = \frac{FP}{FP + TN}$, where TPR is on the y-axis and FPR on the x- axis.

The area under one of the ROC curves can be used as a measure of accuracy in many applications and is called the precision of surface-based measurement. Also, the ROC graph contains all the information contained in the matrix of errors [37].

2.7. Computer characteristics used in the classification of images.

We use a computer with high computational power for the analysis of large quantities of images. CUDA cores are used for to obtain better performance with the TensorFlow and Python library. These cores are owned by NVIDIA brand video cards. Due to this data, an inspection of the models and specifications of the most current cards was carried out.

The selected card is the NVIDIA RTX 2080TI with 11GB of video memory which contains 4352 CUDA cores, incorporates the Turing architecture and brings the Deep Learning Super Sampling (DLSS) technology which includes the Core Tensor. This is an artificial intelligence engine of 114 TFLOP power, which makes the card the one indicated for work on the project. In addition, it has the connection between NVLINK graphics cards that increases the capacity and speed of analyzing data between 5 to 10 times faster than other graphics cards and has a transfer power of 100 GB / s. An Intel Core i9-7900X processor with 10 cores at 3.3 GHz, 64 GB of RAM at 3600 MHz, a 2 TB SSD and a 4 TB HDD. These characteristics are describing below in Table 3.

Table 3. Computer characteristics used in the classification of images.

3. Results

In summary, the proposed algorithm was training, validated, and tested in 25,331 images of skin lesions divided into eight classes, taken from the ISIC 2019 File. All images were resized to a size of 224x244 using bi-linear interpolation, normalized and data augmented to manage the unbalance between classes as in [45].

Multi-class skin lesion classification comes with the problem of severe class imbalance and the small size of those themselves currently available, which represents a challenge for training purposes, therefore, data-augmentation is applied to avoid any bias and overfitting.

The best results with the proposed method were achieved with the following tuning. Firstly, a weighted Cross-entropy is used as loss function to estimate the parameters of all deep models. In addition, the Adam optimizer is initializing with a learning rate of 0.0001 and then it is reduced by 20% if the validation lose function does not decrease by 0.0001 every 45 iterations. Finally, early stopping stops the learning process when the F1-score is not increased by 0.001 through each 45 iterations used to avoid the overfitting that may occur before the convergence of deep models as well as speed up the learning procedures. Thus, the overfitting is prevented, and the bias is reduced. The implementation is carrying out with TensorFlow and Keras libraries. The tuning previous mentioned is also applied to train other CNN architectures.

The dataset is splitting based on 10-fold cross-validation. Usually deep learning workstation use the library sklearn to split in folds the original dataset and run until the end of the model training without any human interference. Nevertheless, due to computational limitation of the equipment used in this project we safe the metrics and restart the running for each fold, making sure that the data was shuffle for every of them.

The Figure 6 depicts the confusion matrix performed by different CNN architectures after been tested with 10-fold cross-validation with 44669 images augmented skin images. The predicted classes are represented as columns while the actual classes are represented as rows. In the diagonal

principal the number of hits can be seen, and the intensity of colour represents how many hits matched that box.

Figure 6. Confusion Matrix for every CNN model, NasnetMobile (A), inception v3 (B), InceptionResNetV2 (C), DenseNet201 (D) and Xception (E)

From the data obtained from the confusion matrix and the formulas explained in section 2.6 implemented in the sklearn.metrics library the Table 4 is created. This shows the performance of every trained CNN model.

Table 1. Classification report for every CNN model

The Figure 7 shows the best classification system through the receiver operating characteristics curve (ROC). By obtaining an area under the curve (AUC), the quality of the classifier is evaluated. The closer this area is to the value of one, the better the classifier.

Figure 7. Curve AUC ROC for every CNN model, NasnetMobile (A), inception v3 (B), InceptionResNetV2 (C), DenseNet201 (D) and Xception (E)

4. Discussion

In order to appraise the performance of a modified version of the Nasnet model for discriminating among eight different skin lessons, it is compared against 5 state-of-the-art models, InceptionV3, InceptionResNetV2, DenseNet201, Xception. The models were applied to the ISIC 2019 dataset. Those models are training using 10-fold cross-validation— a novelty in skin lesion classification for the ISIC dataset, based on the reviews recapitulated in the introduction. From the trained models were obtained the confusion matrix (Figure 6) and AUCROC (Figure 7). All these models differ in terms of computational speed (i.e., run time).

On the confusion matrix (Figure 6), in this case, the proposed model presented a pronounced diagonal principal since most of the predictions were correct. Most of the classes presented few false positives and false negatives; however, the class Melanoma and Nevus presented more errors due of the visual similarity that these classes share, which is congruent with current, state-of-the-art.

The precision metric points to low performance on the most common type of cancer (melanoma). The confusion matrix makes it evident that the models often confound melanoma and Nevus. This confound is also reported by the dermatologist on their diagnostic of skin disease —

leading to the common problem of unnecessary biopsy of a Nevus [46] — which comes from the visual similarity between nevus and melanoma. In future works the results might be improved by adding handcrafted [47] features to the training process.

Studying the precision in this work brings up an absence of this metric similar reviewed papers, therefore should be encouraged to report this metric to enhance this forgot the aspect of skin lesion classification in further works.The classification report for every trained model is depicted in Table 4. Notably, Nasnet has a similar performance than the best score Xception, with a reduction in the number of trainable parameters of at least 75 %.For a future work would be necessary re-training the models on a deep learning workstation and do the comparison after training all the 10-fold at once in order to get a more robust result.

5. Conclusions

This work presents a complete report on the comparation of five different state-of -the-art CNN architectures on the classification task of eight skin lesions (NASNET, InceptionV3, InceptionResNetV2, DenseNet201, Xception). This comparation is establish from the F1-score, Precision, Recall, accuracy and trainable parameters—commonly unreported by similar studies.

The architectures are trained with ISIC 2019 dataset , using 10-fold cross-validation

The obtained results could further be increased using segmentation.

All the models get a significant training accuracy based on a dermatologist one who reaches 75% to 84%. The best performance based on the training accuracy is obtained from Xception with 85%. Nevertheless, it was the heaviest to train with at least five days of continuing running on the computer. In contrast, the NASNET got the second-best performance with 83% but with a considerably reduction of 75% on the training parameters, also decreasing the running on the computer to just 12 hours. The other trained models shows a lower perform than NASNET and Xception.

With the future expansion of ISIC Dataset and another open dataset with dermoscopic pattern annotations become available, future work may consider improve system performance including the use of additional situational contexts, such as patient metadata, history, comparison with other lesions on the patient and evolution through time. In addition, other approaches such as meta-learning, ResNets for semantic segmentation, and complex shape descriptors for classifying diseases might provide additional performance gains. With superior equipment these results may have been more robust and is therefore an area of promising future research.

Author Contributions:

Conceptualization, E.E.C. and M.A.; methodology: M.A, E.E.C., and C.R.; software, J. M.-V. and M.A.; validation, M.A., J. M.-V. and J. M.-A.; data collection and processing, M.A., J. M.-V., and J. M.-A.; writing—original draft preparation, E.E.C., J. M.-A., and M.A.; writing—review and editing, C. R. and N. G..; visualization, supervision, C.R. and N.G.; project administration, E.E.C. and C.R.; funding acquisition, E.E.C. and C.R.

Acknowledgements:

The authors are grateful for the support provided by the National Secretariat of Science and Technology of Panama (SENACYT) and the Technological University of Panama (UTP). Also, we would like to thank the larger community of the International Skin Imaging Collaboration (ISIC) for their effort in organizing the datasets used in this work, as well as engaging and insightful discussions in Dermoscopy and dermatology

Conflicts of Interest:

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article

References

[1] N. Codella, Q.-B. Nguyen, S. Pankanti, D. Gutman, B. Helba, A. Halpern, and J. R. Smith,

"Deep Learning Ensembles for Melanoma Recognition in Dermoscopy Images," *IBM J. Res. Dev.*, vol. 61, no. 4, Oct. 2016, Accessed: Oct. 14, 2020. [Online]. Available: http://arxiv.org/abs/1610.04662.

[2] "Skin Cancer Information - The Skin Cancer Foundation." https://www.skincancer.org/skin-cancer-information/ (accessed Oct. 24, 2019).

[3] H. A. Haenssle, C. Fink, R. Schneiderbauer, F. Toberer, T. Buhl, A. Blum, A. Kalloo, A. Ben Hadj Hassen, L. Thomas, A. Enk, and L. Uhlmann, "Man against Machine: Diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists," *Ann. Oncol.*, vol. 29, no. 8, pp. 1836–1842, 2018, doi: 10.1093/annonc/mdy166.

[4] G. Fabbrocini, V. De Vita, F. Pastore, V. D'Arco, C. Mazzella, M. C. Annunziata, S. Cacciapuoti, M. C. Mauriello, and A. Monfrecola, "Teledermatology: from prevention to diagnosis of nonmelanoma and melanoma skin cancer.," *Int. J. Telemed. Appl.*, vol. 2011, p. 125762, 2011, doi: 10.1155/2011/125762.

[5] A.-R. A. Ali and T. M. Deserno, "A systematic review of automated melanoma detection in dermatoscopic images and its ground truth data," in *Medical Imaging 2012: Image Perception, Observer Performance, and Technology Assessment*, Feb. 2012, vol. 8318, p. 83181I, doi: 10.1117/12.912389.

[6] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis.," *Med. Image Anal.*, vol. 42, no. 1995, pp. 60–88, 2017, doi: 10.1016/j.media.2017.07.005.

[7] S. C. B. Lo, S. L. A. Lou, M. V. Chien, and S. K. Mun, "Artificial Convolution Neural Network Techniques and Applications for Lung Nodule Detection," *IEEE Trans. Med. Imaging*, vol. 14, no. 4, pp. 711–718, 1995, doi: 10.1109/42.476112.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NIPS 2012)*, 2012, p. 4, Accessed: Jun. 03, 2018. [Online]. Available: https://www.nvidia.cn/content/tesla/pdf/machine-learning/imagenet-classification-with-deep-convolutional-nn.pdf.

[9] K. Fukushima, "Neocognition: a self," *Biol. Cybern.*, vol. 202, pp. 193–202, 1980, doi: 10.1007/BF00344251.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[12] G. A. Hembury, V. V. Borovkov, J. M. Lintuluoto, and Y. Inoue, "Deep Residual Learning for Image Recognition Kaiming," *Chem. Lett.*, vol. 32, no. 5, pp. 428–429, 2003, doi: 10.1246/cl.2003.428.

[13] K. Simonyan and A. Zisserman, "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION," 2015. Accessed: Jun. 21, 2019. [Online]. Available: http://www.robots.ox.ac.uk/.

[14] A. R. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir

Anguelov, Dumitru Erhan, Vincent Vanhoucke, "Going deeper with convolutions," *arxiv*, p. 12, 2014.

[15] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[16] Z. Wei, Y. Zhang, H. Günaydın, H. Wang, A. Ozcan, D. Tseng, H. Ceylan Koydemir, Z. Göröcs, K. Liang, Z. Ren, and Y. Rivenson, "Deep Learning Enhanced Mobile-Phone Microscopy," *ACS Photonics*, vol. 5, no. 6, pp. 2354–2364, 2018, doi: 10.1021/acsphotonics.8b00146.

[17] D. Połap, A. Winnicka, K. Serwata, K. Kęsik, and M. Woźniak, "An intelligent system for monitoring skin diseases," *Sensors (Switzerland)*, vol. 18, no. 8, pp. 1–19, 2018, doi: 10.3390/s18082552.

[18] E. Nasr-Esfahani, S. Samavi, N. Karimi, S. M. R. Soroushmehr, M. H. Jafari, K. Ward, and K. Najarian, "Melanoma detection by analysis of clinical images using convolutional neural network," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 1373–1376.

[19] R. Amelard, J. Glaister, A. Wong, and D. A. Clausi, "High-level intuitive features (HLIFs) for intuitive skin lesion description.," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 3, pp. 820–831, 2015.

[20] J. F. Alcón, C. Ciuhu, W. Ten Kate, A. Heinrich, N. Uzunbajakava, G. Krekels, D. Siem, and G. De Haan, "Automatic imaging system with decision support for inspection of pigmented skin lesions and melanoma diagnosis," *IEEE J. Sel. Top. Signal Process.*, vol. 3, no. 1, pp. 14–25, 2009.

[21] P. G. Cavalcanti, J. Scharcanski, and G. V. G. Baranoski, "A two-stage approach for discriminating melanocytic skin lesions using standard cameras," *Expert Syst. Appl.*, vol. 40, no. 10, pp. 4054–4064, 2013.

[22] P. G. Cavalcanti and J. Scharcanski, "Automated prescreening of pigmented skin lesions using standard cameras," *Comput. Med. Imaging Graph.*, vol. 35, no. 6, pp. 481–491, 2011.

[23] E. Chao, C. K. Meenan, and L. K. Ferris, "Smartphone-based applications for skin monitoring and melanoma detection," *Dermatol. Clin.*, vol. 35, no. 4, pp. 551–557, 2017.

[24] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8697–8710, 2018, doi: 10.1109/CVPR.2018.00907.

[25] T. J. Brinker, A. Hekler, J. S. Utikal, N. Grabe, D. Schadendorf, J. Klode, C. Berking, T. Steeb, A. H. Enk, and C. Von Kalle, "Skin cancer classification using convolutional neural networks: Systematic review," *Journal of Medical Internet Research*, vol. 20, no. 10. Journal of Medical Internet Research, p. e11936, Oct. 17, 2018, doi: 10.2196/11936.

[26] V. Pomponiu, H. Nejati, and N. M. Cheung, "Deepmole: Deep neural networks for skin mole lesion classification," in *Proceedings - International Conference on Image Processing, ICIP*, Aug. 2016, vol. 2016-August, pp. 2623–2627, doi: 10.1109/ICIP.2016.7532834.

[27] J. Kawahara, A. Bentaieb, and G. Hamarneh, "Deep features to classify skin lesions," in *Proceedings - International Symposium on Biomedical Imaging*, Jun. 2016, vol. 2016-June, pp. 1397–1400, doi: 10.1109/ISBI.2016.7493528.

[28] N. Codella, J. Cai, M. Abedini, R. Garnavi, A. Halpern, and J. R. Smith, "Deep learning, sparse coding, and SVM for melanoma recognition in dermoscopy images," in *Lecture*

Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2015, vol. 9352, pp. 118–126, doi: 10.1007/978-3-319-24888-2_15.

[29] "Dermofit Image Library - Edinburgh Innovations." https://licensing.edinburgh-innovations.ed.ac.uk/i/software/dermofit-image-library.html (accessed Oct. 08, 2020).

[30] 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, no. 7639, pp. 115–118, Feb. 2017, doi: 10.1038/nature21056.

[31] 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," *J. Invest. Dermatol.*, vol. 138, no. 7, pp. 1529–1538, Jul. 2018, doi: 10.1016/j.jid.2018.01.028.

[32] L. Bi, J. Kim, E. Ahn, and D. Feng, "Automatic Skin Lesion Analysis using Large-scale Der-moscopy Images and Deep Residual Networks." Accessed: Jun. 05, 2020. [Online]. Available: https://challenge.kitware.com/#phase/584b08eecad3a51cc66c8e1f.

[33] X. Sun, J. Yang, M. Sun, and K. Wang, "A benchmark for automatic visual classification of clinical skin disease images," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9910 LNCS, pp. 206–222, doi: 10.1007/978-3-319-46466-4_13.

[34] A. Romero Lopez, X. Giro-I-Nieto, J. Burdick, and O. Marques, "Skin lesion classification from dermoscopic images using deep learning techniques," in *Proceedings of the 13th IASTED International Conference on Biomedical Engineering, BioMed 2017*, Apr. 2017, pp. 49–54, doi: 10.2316/P.2017.852-053.

[35] C. Enyinna Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning."

[36] R. Muhamedyev, "Machine learning methods: An overview," 2015. https://www.researchgate.net/publication/320550516_Machine_learning_methods_An_overview (accessed May 05, 2020).

[37] M. A. Nielsen, "Neural Networks and Deep Learning." Determination Press, 2015.

[38] S. Ruder, "An overview of gradient descent optimization algorithms ∗."

[39] S. Ruder, "An overview of gradient descent optimization algorithms *." Accessed: Jun. 05, 2020. [Online]. Available: http://caffe.berkeleyvision.org/tutorial/solver.html.

[40] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," Dec. 2015.

[41] P. Tschandl, C. Rosendahl, and H. Kittler, "Data descriptor: The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Sci. Data*, vol. 5, pp. 1–9, 2018, doi: 10.1038/sdata.2018.161.

[42] D. Gutman, N. C. F Codella, E. Celebi, B. Helba, M. Marchetti, N. Mishra, and A. Halpern, "Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on Biomedical Imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC)." Accessed: Jun. 05, 2020. [Online]. Available: https://isic-archive.com/.

[43] M. Combalia, N. C. F. Codella, V. Rotemberg, B. Helba, V. Vilaplana, O. Reiter, C. Carrera, A. Barreiro, A. C. Halpern, S. Puig, and J. Malvehy, "BCN20000: Dermoscopic Lesions in the Wild," Aug. 2019, Accessed: Sep. 28, 2020. [Online]. Available: http://arxiv.org/abs/1908.02288.

[44] T. Lee, V. Ng, R. Gallagher, A. Coldman, and D. McLean, "Dullrazor®: A software

approach to hair removal from images," *Comput. Biol. Med.*, vol. 27, no. 6, pp. 533–543, Nov. 1997, doi: 10.1016/S0010-4825(97)00020-6.

[45] J. Jianbo Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000, doi: 10.1109/34.868688.

[46] C. Carrera and A. A. Marghoob, "Discriminating Nevi from Melanomas: Clues and Pitfalls," *Dermatologic Clinics*, vol. 34, no. 4. W.B. Saunders, pp. 395–409, Oct. 01, 2016, doi: 10.1016/j.det.2016.05.003.

[47] L. Nanni, S. Ghidoni, and S. Brahnam, "Handcrafted vs. non-handcrafted features for computer vision classification," *Pattern Recognit.*, vol. 71, pp. 158–172, Nov. 2017, doi: 10.1016/j.patcog.2017.05.025.

© 2020 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

Table 1(on next page)

Table 1. Augmentation details

Data Augmentation settings applied to the dataset are described in Table 1 and the effects would be seen in the Figure 3. These methods are applied through the keras library with the ImageDataGenerator function. The process is also used to resize images to 224 x 224.

1 ***Table 1. Augmentation details***

Augmentation	**percentage or ratio range**
rotation_range	180
width_shift_range	0.1
height_shift_range	0.1
zoom_range	0.1
horizontal_flip	true
vertical_flip	true
fill_mode	nearest

2

Table 2(on next page)

Table 2. Data distribution for the multiclass classification confusion matrix.

Through the confusion matrix we can obtain the productivity of the model during the training or the development of a classification problem. This shows us in detail how many times the model is wrong when making predictions. The number of correct and incorrect predictions is obtained by counting values and separated them from each class. It gives us an idea not only of the mistakes that the model makes but of the types of mistakes it makes. This matrix allows us to measure Recall, Precision, Accuracy, and the AUC-ROC curve. This matrix describes the complete performance of the model. Table 2 shows the data distribution for multiclass classification.

Table2. Data distribution for the multiclass classification confusion matrix

Known Class	**0**	**1**	**2**	**…**	**j**
0	**TP**	FN	FN	FN	FN
1	FP	TN	FN	FN	FN
2	FP	FN	TN	FN	FN
…	FP	FN	FN	TN	FN
j	FP	FN	FN	FN	TN

Table 3(on next page)

Computer characteristics used in the classification of images.

An Intel Core i9-7900X processor with 10 cores at 3.3 GHz, 64 GB of RAM at 3600 MHz, a 2 TB SSD and a 4 TB HDD. These characteristics are describing below in Table 3.

1 Table 3. Computer characteristics used in the classification of images.

Component	Description
Power Supply	Cooler Master Watt Maker 1500 – 1500 W
Mother board	Asus ROG STRIX X299-E GAMING – LGA2066
Chip	Intel Core i9-7900X a 3.3Ghz (Skylake-X)
RAM	64 GB 3600 Mhz
GPU	GeForce RTX 2080 Ti GDDR6 (x2) NVLINK 11 GB RAM
SSD	2 TB
HDD	4 TB

2

Table 4(on next page)

Table 4. Computer characteristics used in the classification of images.

Through the years several studies have been conducted in this field. A continuation (Table 4) comparison between previous studies and our proposed method is presented.

1

2 *Table 4. Classification report for every CNN model*

3

	NasnetMobile			inceptionv3			InceptionResNetV2			DenseNet201		
Diseases	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
AK	0.87	0.82	0.85	0.86	0.82	0.84	0.74	0.71	0.72	0.74	0.70	0.72
BCC	0.71	0.80	0.75	0.72	0.72	0.72	0.61	0.63	0.62	0.61	0.61	0.61
BKL	0.72	0.75	0.74	0.63	0.70	0.67	0.57	0.60	0.59	0.59	0.57	0.58
DF	0.99	0.94	0.97	0.95	0.97	0.96	0.91	0.88	0.89	0.90	0.89	0.89
MEL	0.68	0.73	0.70	0.63	0.61	0.62	0.58	0.60	0.59	0.52	0.63	0.57
NV	0.76	0.78	0.77	0.74	0.70	0.72	0.69	0.69	0.69	0.65	0.67	0.66
SCC	0.94	0.84	0.89	0.85	0.89	0.87	0.75	0.75	0.75	0.78	0.74	0.76
VASC	0.99	0.98	0.99	0.99	0.96	0.97	0.95	0.94	0.95	0.94	0.91	0.92
Accuracy			0.83			0.79			0.72			0.71
Macro Avg	0.83	0.83	0.83	0.80	0.80	0.80	0.73	0.72	0.72	0.72	0.71	0.71
Weighted Avg	0.84	0.83	0.83	0.79	0.79	0.79	0.72	0.72	0.72	0.72	0.71	0.71
Total params	4,700,572.00			22,245,160.00			54,668,520.00			19,105,352.00		
Trainable params	4,663,834.00			22,210,728.00			54,607,976.00			18,876,296.00		
Non-trainable params	36,738.00			34,432.00			60,544.00			229,056.00		

4

Figure 1

Figure 1: Methodology

To complete this project is used the methodology summarized in the figure 1.

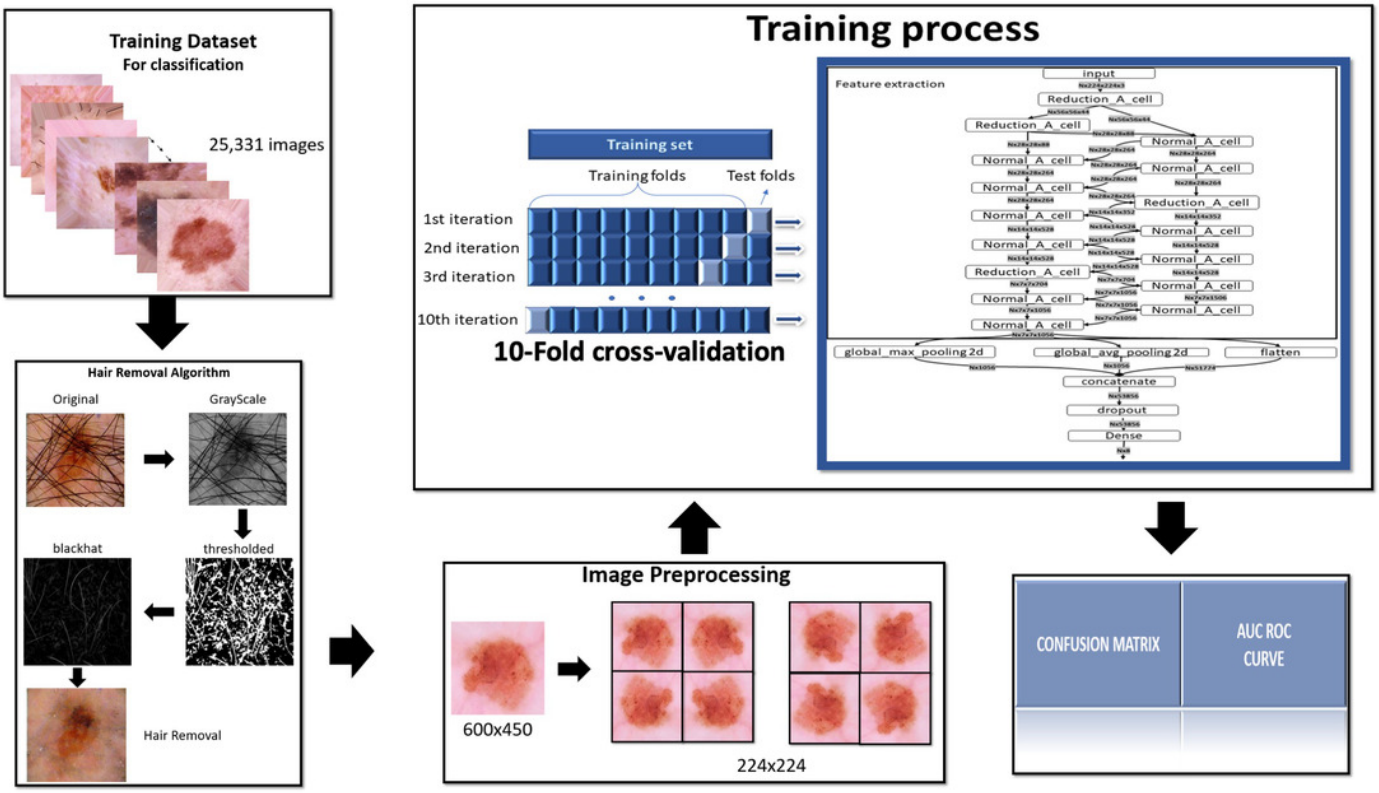


Figure 2

Figure2. Hair removal process

Figure2. Hair removal process applied in one of the images from ISIC dataset.

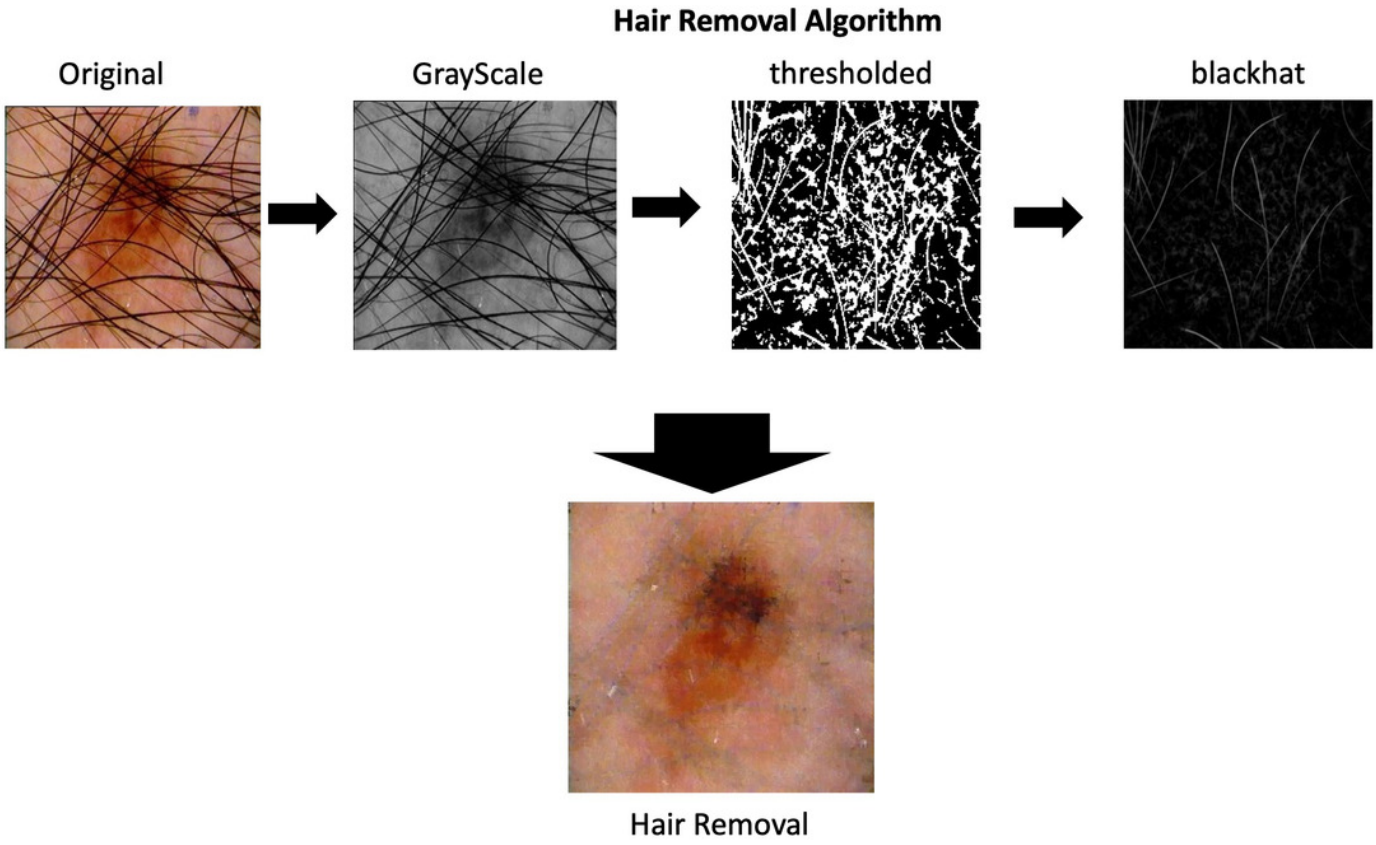


Figure 3

Figure 3. Data-Augmentation

Figure 3. Data-Augmentation applied in one of the images from ISIC dataset.

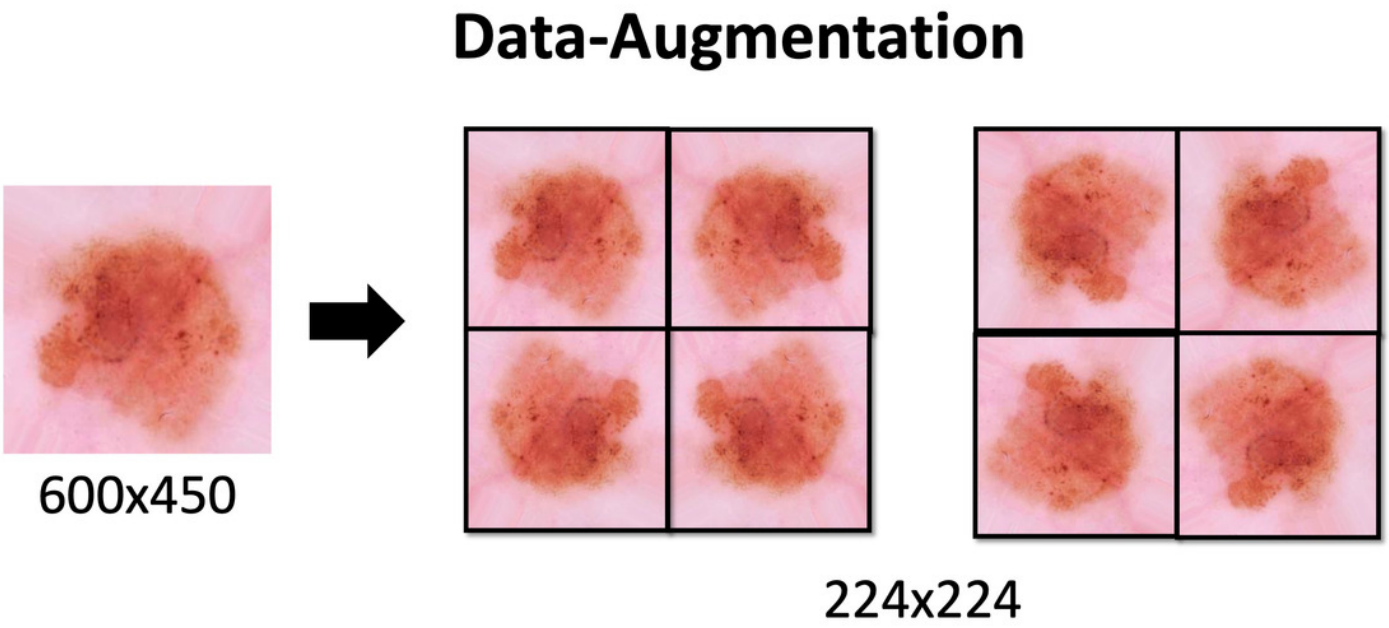


Figure 4

Figure 4. The two main functions of NASNet architecture

This architecture is composed of convolutional cells. The two main functions are Normal Cell and Reduction Cell, shown in Figure 4. The Reduction Cell returns a feature map height and width reduced by a factor of two. On the other hand, the Normal Cell returns a feature map with normal cells with the same input dimensions. The model used for this purpose was NASNet-A (4 @ 1056), where the number 4 represents the number of cell repeats, and 1056 corresponds to the number of filters in the penultimate layer of the network.

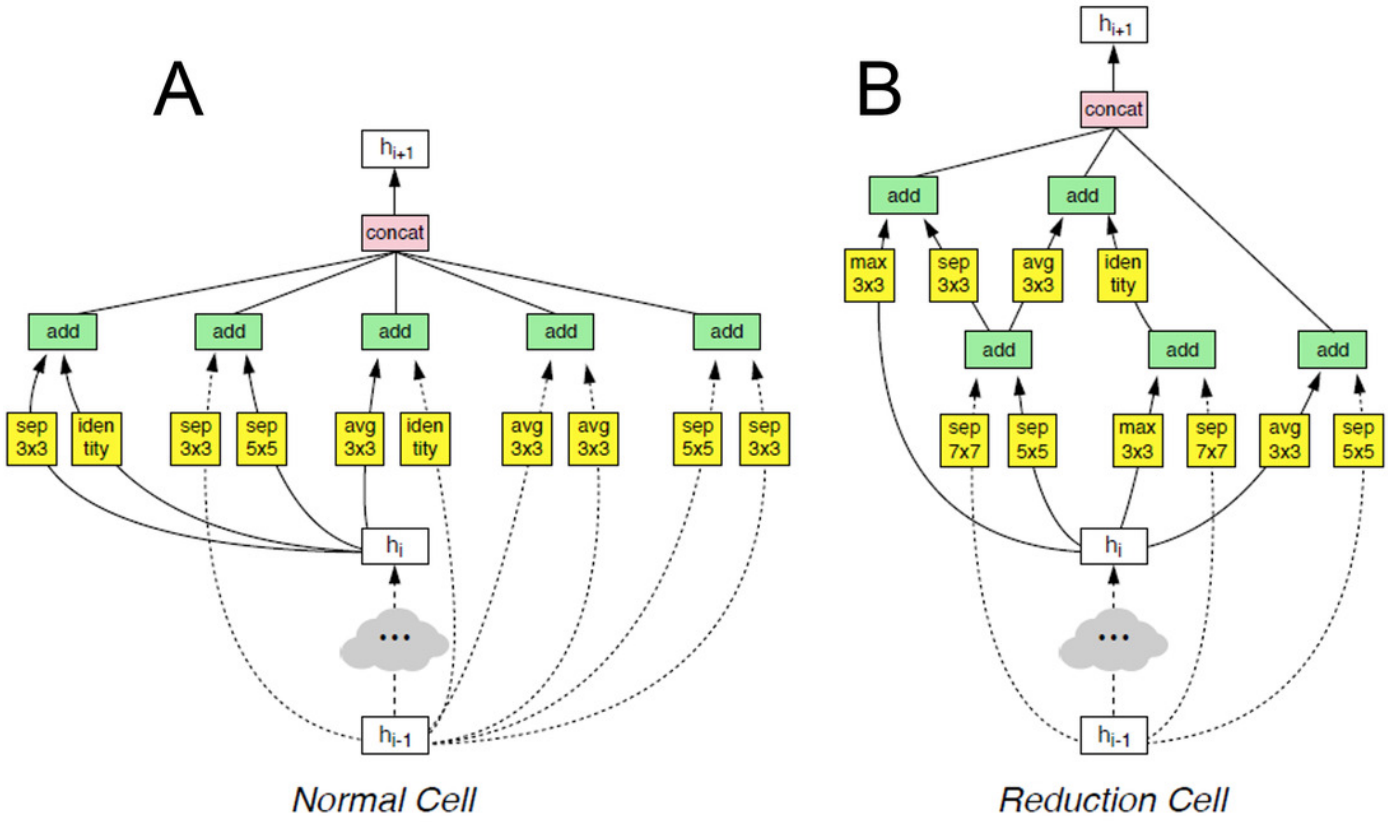


Figure 5

Figure 5. Representation of the steps of the extraction feature map and classification.

This layer fed the dropout layer, which turned off some neurons to prevent overfitting of the network. Finally, it fed the dense layer, which offered the inference of the model through a σ(z^l) function. These steps are depicted in Figure 5.

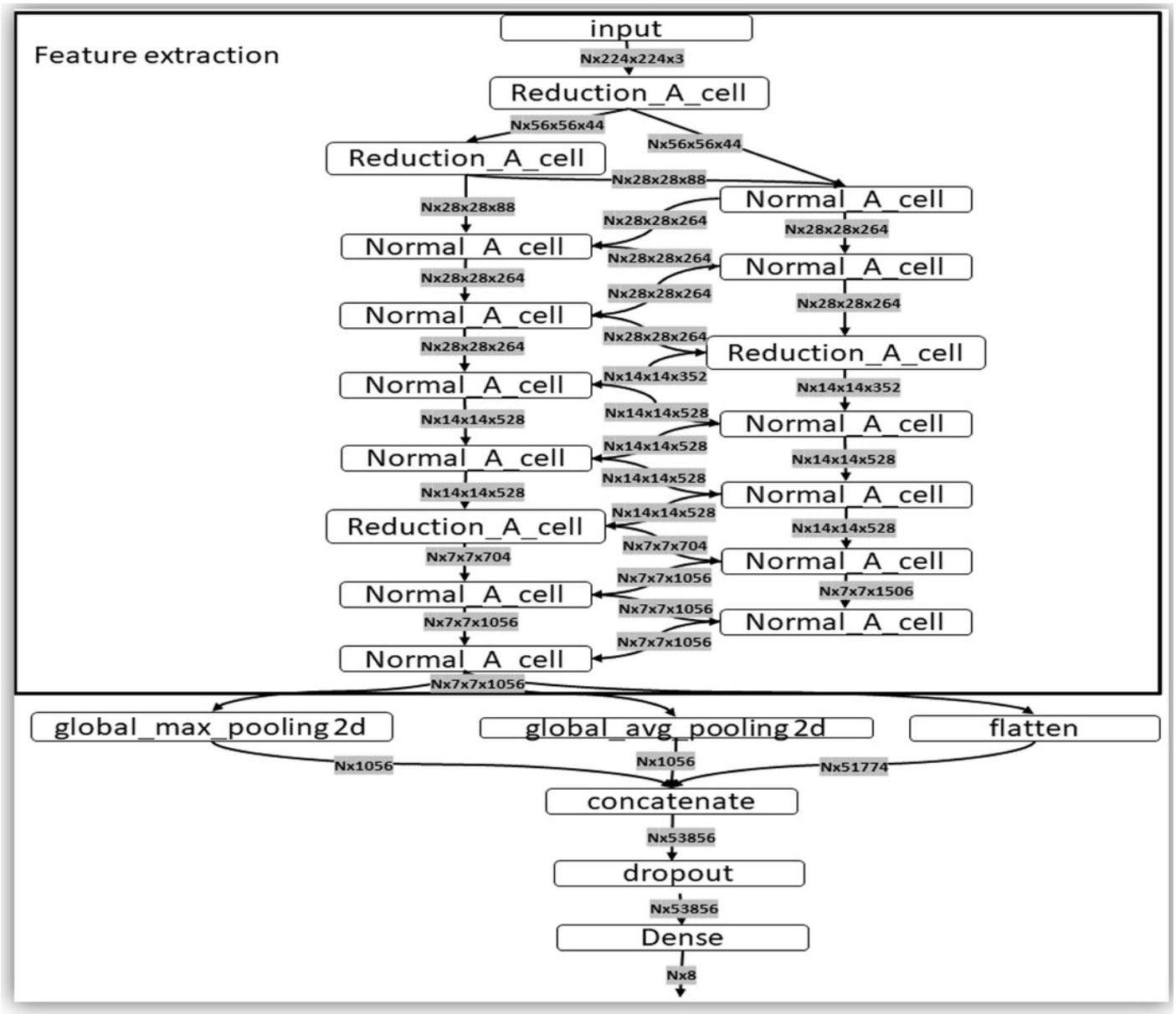


Figure 6

Figure 6. Confusion matrix yielded by the proposed CNN method.

Figure 6. Confusion matrix yielded by the proposed CNN method. Where Actinic keratosis (AK), Basal cell carcinoma (BCC), Dermatofibroma (DF), Melanoma (ML), Nevus (NV), Pigmented benign keratosis (PBK), Seborrheic keratosis (SK), Squamous cell carcinoma (SL), Vascular Lesion (VL).

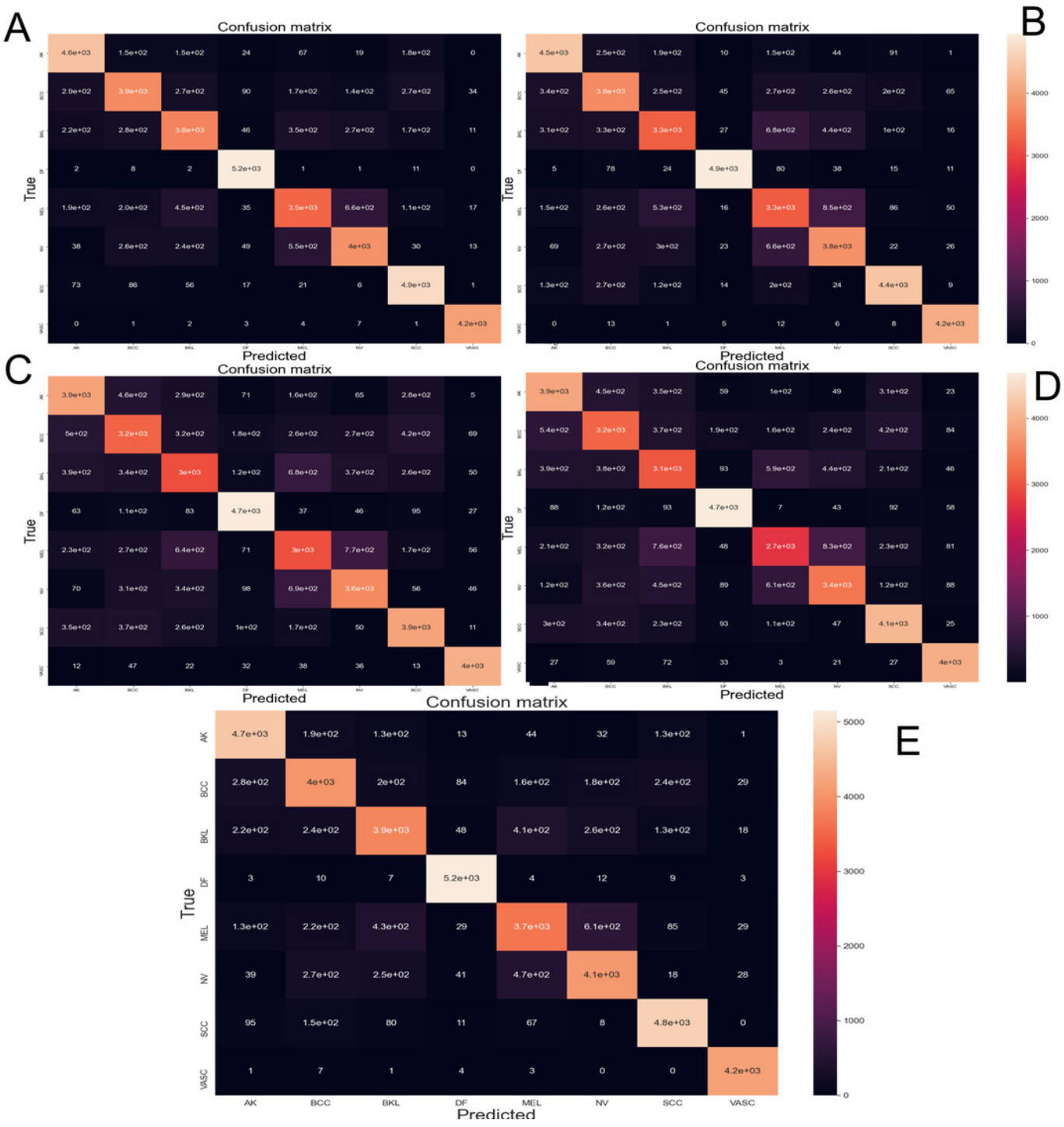
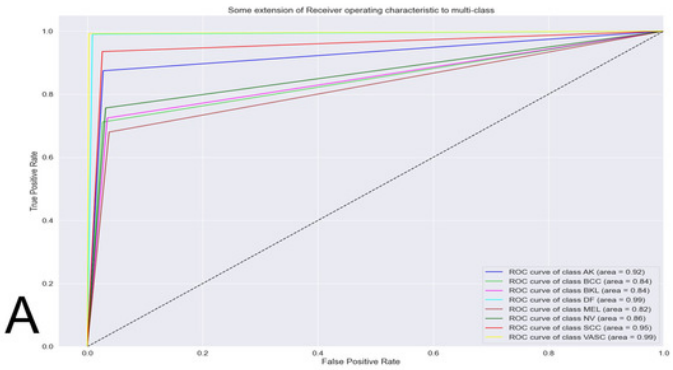
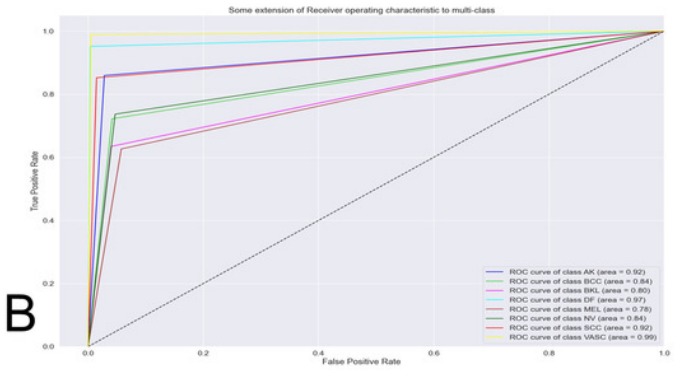

A
Confusion matrix
True
Predicted
B
Confusion matrix
True
Predicted
C
Confusion matrix
True
Predicted
D
Confusion matrix
True
Predicted
E
Confusion matrix
True
AK
BCC
BKL
DF
MEL
NV
SCC
VASC
Predicted

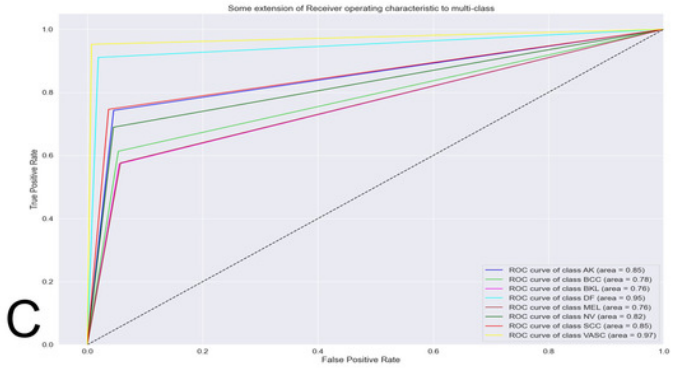
Figure 7

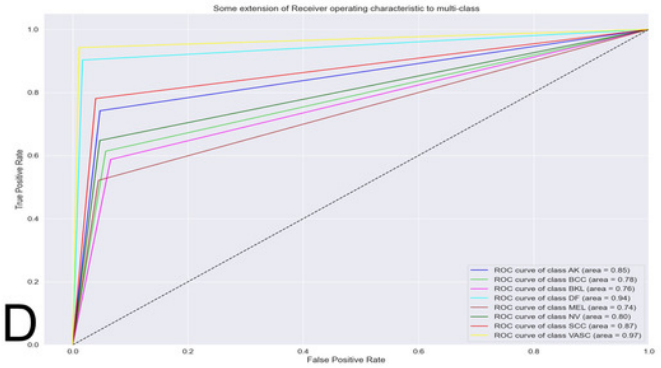
Figure 7. Curve AUC ROC.

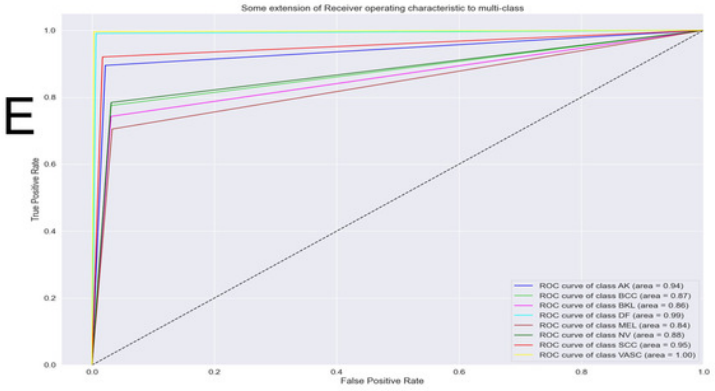
The Figure 7 shows the best classification system through the receiver operating characteristics curve (ROC). By obtaining an area under the curve (AUC), the quality of the classifier is evaluated. The closer this area is to the value of one, the better the classifier.


A
Some extension of Receiver operating characteristic to multi-class
True Positive Rate
False Positive Rate
ROC curve of class AK (area = 0.92)
ROC curve of class BCC (area = 0.84)
ROC curve of class BKL (area = 0.84)
ROC curve of class DF (area = 0.99)
ROC curve of class MEL (area = 0.82)
ROC curve of class NV (area = 0.86)
ROC curve of class SCC (area = 0.95)
ROC curve of class VASC (area = 0.99)


B
Some extension of Receiver operating characteristic to multi-class
True Positive Rate
False Positive Rate
ROC curve of class AK (area = 0.92)
ROC curve of class BCC (area = 0.84)
ROC curve of class BKL (area = 0.80)
ROC curve of class DF (area = 0.97)
ROC curve of class MEL (area = 0.78)
ROC curve of class NV (area = 0.84)
ROC curve of class SCC (area = 0.92)
ROC curve of class VASC (area = 0.99)


C
Some extension of Receiver operating characteristic to multi-class
True Positive Rate
False Positive Rate
ROC curve of class AK (area = 0.85)
ROC curve of class BCC (area = 0.78)
ROC curve of class BKL (area = 0.76)
ROC curve of class DF (area = 0.95)
ROC curve of class MEL (area = 0.76)
ROC curve of class NV (area = 0.82)
ROC curve of class SCC (area = 0.85)
ROC curve of class VASC (area = 0.97)


D
Some extension of Receiver operating characteristic to multi-class
True Positive Rate
False Positive Rate
ROC curve of class AK (area = 0.85)
ROC curve of class BCC (area = 0.78)
ROC curve of class BKL (area = 0.76)
ROC curve of class DF (area = 0.94)
ROC curve of class MEL (area = 0.74)
ROC curve of class NV (area = 0.80)
ROC curve of class SCC (area = 0.87)
ROC curve of class VASC (area = 0.97)


E
Some extension of Receiver operating characteristic to multi-class
True Positive Rate
False Positive Rate
ROC curve of class AK (area = 0.94)
ROC curve of class BCC (area = 0.87)
ROC curve of class BKL (area = 0.86)
ROC curve of class DF (area = 0.99)
ROC curve of class MEL (area = 0.84)
ROC curve of class NV (area = 0.88)
ROC curve of class SCC (area = 0.95)
ROC curve of class VASC (area = 1.00)