Using Neural Networks to Discriminate between Benign and Malignant Moles

Michael Birkholz michael.birkholz@estudiantat.upc.edu
David Dueñas Gaviria david.duenas.gaviria@estudiantat.upc.edu
David Fernández Aldana david.fernandez.aldana@estudiantat.upc.edu
Ronald Rivera Torres ronald.rivera@estudiantat.upc.edu

Abstract

This is the abstract. Write it after most of the rest of the paper is complete.
Should have:

- background/motivation/context
- aim/objective/problem statement
- approach/method
- results
- conclusions/implications

Just adding some extra text to fill this out so it occupies roughly the expected amount of space it will need to fill. I'm estimating roughly 250-300 words, which should come out to about 18-20 lines. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed malesuada ligula dui, sed posuere lorem congue vitae. Nunc quis finibus nulla. Donec semper quam risus, eget consectetur mi sollicitudin lacinia. Mauris lacus tortor, volutpat vel fermentum ac, rutrum eget enim. Nullam fringilla, felis ac malesuada hendrerit, magna eros ultricies ex, et mattis sapien orci et enim. Aliquam laoreet diam nisl, vitae ullamcorper eros iaculis eget. Praesent vestibulum ullamcorper efficitur. Etiam malesuada purus sed metus blandit ornare. Ut ac iaculis lacus, sed convallis quam. Donec ultrices, felis et aliquet elementum, sem lorem ultrices elit, sagittis venenatis orci lectus quis turpis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Phasellus aliquam, nisi quis facilisis fermentum, felis enim venenatis enim, sed pretium erat nunc a urna. Duis vel lorem eu mi dignissim venenatis sit amet ut orci. Phasellus gravida massa sit amet massa sagittis volutpat. Phasellus fringilla porttitor.

Keywords: Neural Networks, Convolutional Neural Networks, CNN, Inception-v4, Data augmentation, Skin cancer, Mole, Benign, Malignant, Deep Learning

1 Problem Statement and Goals

According to the Skin Cancer Foundation[1], skin cancer is the most common cancer worldwide, with 1 in 5 Americans suffering from it by the time they reach age 70. It is also an incredibly deadly disease, killing more than two Americans per hour on average. However, if it is detected early, the 5-year survival rate for melanoma, one of the deadliest skin cancers, is 99 percent. This final statistic is a major motivation for undertaking this project, which studies the feasibility of applying machine learning methods to classify skin lesions as cancerous or benign. The hope is that having an accessible, convenient, reliable, inexpensive method capable of making an initial assessment of skin lesions may significantly improve the prognosis for many skin cancer patients since the cancers could be detected much earlier in many cases.

We have obtained a labeled dataset from a Kaggle project compiled by Claudio Fanconi[1] where the original source data comes from the ISIC-Archive[2]. It contains 1800 pictures of benign moles and 1497 pictures of malignant ones. All pictures are 224x224 pixels. As this is not a very large dataset, we have explored the use of data augmentation to synthesize additional input images from the given data.

The primary Computational Intelligence (CI) technique we have applied is using neural networks as classifiers. Since this is an image processing problem, convolutional neural networks (CNNs) are particularly well-suited to the job. For this reason, we have chosen a number of different architectures to compare, including a custom network built from scratch, Inception-v4 and ResNet50. Additionally, in order to overcome the lack of input data to train a complex network from scratch, we have also explored transfer learning using pre-trained networks and adding additional layers to distinguish between images of malignant or benign lesions.

2 Previous Work

Machine learning approaches have been applied to skin cancer classification problems for quite some time, but in 2016 there was a revolution. Convolutional neural networks (CNNs) came onto the scene and essentially replaced all other machine learning methods in every single entry in the International Symposium on Biomedical Imaging[3], which marked the start of a new era in machine-learning applied to medical diagnostic tasks with a visual component. Convolutional neural networks are particularly well-suited to image analysis because the convolution operation provides a convenient way to extract features in a translation-invariant manner. And as an added benefit, it is actually practical for use due to the computational advantages provided by the sparse connectivity between layers as contrasted with a fully-connected multi-layer perceptron (MLP).

Our approach to using transfer learning, initially trained on a large dataset (ImageNet) containing general non-medical images and then subsequently refined based on images containing examples of malignant or benign marks on the skin, is supported by the literature. Specifically, in Menegola et al's 2017 paper[4], this particular approach is analyzed and they reach the conclusion that it is in fact better to use a network trained on generic images instead of one fine-tuned for even another medical application, although they leave the door open to future research in that regard. Their supposition is that the feature extractors obtained from a generic set of images are superior to the specific ones derived based on retinopathy, a very tangentially-related medical use case, in their study.

A paper by Zhang et al, Skin Cancer Diagnosis Based on Optimized Convolutional Neural Network[5], is especially interesting from a computational intelligence perspective, due to the comparison of many techniques taken from nature. The core of the paper is an algorithm used to optimize a neural network based on whale hunting techniques. This technique itself is not novel, but the the team proposed a modification to it by using the Lévy flight mechanism to avoid premature convergence and thereby provide more optimal results in identifying skin cancers. They compare their optimization algorithm against a number of other ones including a genetic algorithm, shark smell optimization, World Cup optimization, grasshopper optimization and particle swarm optimization and achieves significantly better results on many benchmarks. Unfortunately, the paper does not provide an actual number to quantify how much better their final results are at classifying malignant vs. benign than other referenced algorithms, but instead the reader must estimate the improvement from a graph in the results section.

One of the common threads seen throughout most papers we have looked at is that in most cases the researchers have access to much larger datasets. Our dataset is on the order of 3000 images, while the research teams behind many of the papers have access to many times that amount. For example, Esteva's team[6] had access to a database with nearly 130,000 images. On the other extreme, there was a very small, two-layer CNN trained on only 136 images which achieved over 80% on sensitivity, specificity and accuracy[3]. However, these results must be viewed with a healthy dose of skepticism as the test set only comprised 34 images. This last case was the exception to the rule, however, and most research used many more training images than we had access to for this exercise.

Another outcome or goal of many of the papers reviewed was to be able to properly segment the lesion and separate it from the surrounding skin. In some cases the shape of the segmentation was used as an input to the classification algorithm[7]. In other cases, it appeared to simply be a byproduct of using the CNN[5]. In fact, on the Kaggle website where the dataset was obtained, there is an open task to create an unsupervised model that can learn to segment moles from the surrounding skin, but no submissions have been attempted for that task.

3 The CI Methods

CI methods are computational techniques whose inspiration is found in natural phenomena. Of the major CI An assortment of CI methods used in this project are outlined in this section.

3.1 Activation Function

In the cancer classification problem the test cannot be positive and negative at the same time, so the outputs are mutually exclusive. Differently to a problem where the classification of a lung disease could be cancer and pneumonia at the same time. Usually, the Softmax activation is used with multi-class classification problems, because it distributes the probability on each output node. However, in this study using Sigmoid for a binary classification problem is equal to softmax and therefore the better choice for the binary classification is to use one output unit with Sigmoid instead of Softmax with two output units, just for for simplicity besides the fact that it presumably update faster.

3.2 Data Augmentation

The purpose of applying data augmentation is to increase the generalizability of the model. Given that the context of the application of the Network is in a somewhat uncontrolled environment, it is constantly seeing new, slightly modified versions of the input data, so with data augmentation is able to learn more robust features. E.g. skin tone, exposure, noise, blur, rotation changes, etc.

Data augmentation encompasses a wide range of techniques used to generate training samples taken from the original set of images by applying various and random perturbations without changing the class labels of the images.

For this reason, in the first method we will not apply data augmentation and simply evaluate our trained network on the unmodified testing data. But for the Keras Augmentation approach, different testing might be interesting to achieve. To see the changes in the testing accuracy and at the end having a better performance at the expense of a slight dip in training accuracy. Although using Keras build in module for Image augmentation $'ImageDataGenerator'$ might be a convenient approach for many situations were the lack of images would not reach a plenum state but instead underfit, the downside of using this generator is that a full control over the augmentation is not held and if this is not being treated carefully it might lead to an unrealistic generation of images which will only add noise to the model.

The idea of generating augmented data also helps to have an equally balanced dataset, distributed among the classes. So, in short the generation of modified versions of train images is to make sure that the model gets feed with enough data for the generalization, and secondly for robustness in terms of becoming invariant to small changes like rotation or camera angles when applied in a real world scenario.

Table 1: **Keras Augmentation parameters**

	Horizontal Flip	**Vertical Flip**	**Shear**	**Zoom**	**Rotation**	**Brightness**	**Accuracy**	**F1 score**
1	False	False	0.2	0.2	0.2	-	0.7	**0.91**
2	True	False	0.2	0.5	0.2	0.9 1.1	0.7	**0.01**
3	False	False	0.2	0.5	0.1	0.9 1.1	0.7	**0.91**
4	True	True	0.2	0.2	0.2	-	0.7	**0.91**
5	True	True	0.2	0.2	0.1	0.9 1.1	0.7	**0.88**

3.3 Convolutional Neural Networks

Neural networks are clearly patterned after biological structures and were in fact born from the fields of psychology and neurophysiology rather than computer science. While they are much simpler than biological neurons, in large groups and when exposed to enough training data, a behavior emerges which mirrors organic neurons. The CNNs are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. The inspiration for Convolutional Neural Networks was somewhat indirectly found in nature as well. They are based on the neocognitron[8] which was intended to mimic the behavior of cells in the visual cortex of cats and monkeys.

Our team created a convolutional neural network to attempt to solve the binary classification problem of benign vs. malignant when presented with an image of a lesion. We refer to this network being built ”from scratch” since we used only the basic building blocks available in the Keras library instead of relying on an existing deep learning architecture. As this was a custom design, importing existing weights and training via transfer learning was not an option for this network.

The network itself starts with a number of convolution layers interleaved with max pooling layers. The first two convolution layers are set up to learn 32 filters each and use a 3×3 kernel, stride 1. ReLU was chosen as the activation function for all intermediate layers since current research indicates that it provides favorable characteristics to facilitate training over the sigmoid function. The third and last convolution layer was set up to learn 64 filters instead of 32. Each of the 2×2 max pooling layers is intended to reduce sensitivity to local variations and reduce the volume of data that needs to be fed forward in the network. Next, the 3-D tensor is flattened to 1-D. After this, the architecture is flexible enough to incorporate a number of optional layers. For example, in Table 2, there is an optional dropout layer included. Another pattern that works well is including a fully-connected layer before the dropout layer. Finally, the last two layers include a fully-connected layer that aggregates all of the outputs of the previous layers and feeds a sigmoid activation function, chosen since this is a binary classification problem.

Table 2: Summary of CNN from Scratch Model with one Optional Layer

Layer (type)	**Output Shape**	**Param #**
conv2d (Conv2D)	(None, 222, 222, 32)	896
activation (Activation)	(None, 222, 222, 32)	0
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 32)	9248
activation_1 (Activation)	(None, 109, 109, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 32)	0
conv2d_2 (Conv2D)	(None, 52, 52, 64)	18496
activation_2 (Activation)	(None, 52, 52, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 64)	0
flatten (Flatten)	(None, 43264)	0
Optional Layer: dropout (Dropout)	(None, 43264)	0
dense (Dense)	(None, 1)	43265
activation_3 (Activation)	(None, 1)	0
Total params: 71,905 Trainable params: 71,905 Non-trainable params: 0		

3.4 Transfer Learning

In some scenarios, it can be difficult and expensive to obtain training data that fits the feature space and the expected characteristics of the test data. This is how the need to create a high-performance learner for the target domain comes[9]. This learner consists in essence, in taking advantage of a large amount of information related to solving a problem and using it on a different one, but that shares certain characteristics with it. In other words, modify already trained patterns (or Neural Networks) to be able to recognize similar ones. This is the motivation for the Transfer Learning, which is used to improve learners by transferring information from related fields. A few specific implementations of Convolutional Neural Networks for image processing tasks which were adapted in the Transfer Learning field, ResNet and Inception, follow.

3.4.1 ResNet

Residual Networks or commonly know as ResNet, one of the most popular deep convolution neural network developed by Microsoft research team in 2015. To this day, ResNet still outperform most of neural networks on computer vision implementations. The mayor contribution of the ResNet research is that in sufficient deep model adding additional layers could cause a hinder in the model performance. This is due to the degradation problem not being addressed. The degradation problem occurs when applying convolution layers we are constantly downsizing the map of features, and it is expected that at certain point in the model the features map should maintain the same feature size, due to obtain strong features of the evaluated image. For this reason the next block should be a copy of the precious one, this block is known as the identity mapping. The problem happen when the neural networks can not identify the identity map. It is speculated this is cause by the statistical approximation on neural networks. The researchers team approach to address the degradation problem is to add an skip connection to the residual information and then add residual values to the results of the next convolution layers. The goal of this approach is to use the residual information as the identity map and incorporate the new features obtained with the following layers. Is worth mentioning the research team, have a word of caution mentioning this approach is not optimal, but it high performance is good enough for the model.

3.4.2 InceptionV4

Inception v4 is an evolution of GoogLeNet (Inception v1) and Inception v3, developed in 2016. This new version is a more simplified architecture and more inception modules than Inception third version. Before proceeding with the explanation of the fourth version of Inception we first need to explain the earlier iteration of this model since it build on top of them. On the first Inception version it propose to instead of using a deeper neural network it uses a wider network. The idea is is to run multiple filter sizes at the same level and then concatenate the results. The first version comprehend in 9 of these inception modules stacked linearly and 22 layers. In the third version the main focus was to simplify more the inception model. The researcher team notice that the auxiliaries classifiers did not contribute much until the end of the process, when the accuracies are almost saturated. In addition, to improve the computational speed hey introduce factorization of the 7x7 convolution to two 5x5 convolution that them self would also be factorized to two 3x3 convolution. Finality, in the version use on our analysis Inception v4 the initial set of operation before running the inception blocks. Lastly, it was also introduced reduction blocks to change height and width on the grids of the model. As previous improvements, the goal of these changes was to boost even more the model performance.

Our implementation, deemed ”BinaryInceptionV4” in code, makes use of InceptionV4 (based on https://github.com/kentsommer/keras-inceptionV4 updated to work in Tensorflow 2.4) without the top layers, adding a Flatten layer, an optional 128-unit dense layer and dropout layer in the middle (see the particular experiment) and finally a single neuron dense layer with Sigmoid activation to do the binary classification.

4 Results and Discussion

4.1 General methodology

4.1.1 Data preprocessing

The images pixel values are normalized between 0 and 1 as Resnet50 and InceptionV4 perform best with such reescaling. Also the folds are pre-created in disk to allow for better performance and complete determinism of the cross-validation procedure as explained in the next section.

4.1.2 Validation

Stratified 5-fold validation is used to validate each model performance. The number of folds was chosen to be a compromise between the runtime time that more folds would make and the training-validation splits, since:

$$split_{validation} = \frac{1}{|folds|} = \frac{1}{5} = .2$$

and thus:

$$split_{training} = 1 - split_{validation} = 1 - .2 = .8$$

Less folds would result in a less representative result because of less training data (generalization issue) and runs (more variance) while a higher number not only would take a considerable amount of time to train but also because of lower validation split, would end up being less representative as well. This stratified approach was used since there is a small class imbalance problem with about 55% being 'benign' and about 45% being 'malignant', while the folding approach was chosen to mitigate the fact that we don't have many samples to work with.

4.1.3 Performance metrics

With respect to the performance scores used to evaluate and improve the models[10], in this binary classification problem , the metrics considered were:

- **F1 Score**: This is the harmonic mean value of "Precision" and "Recall", which can better measure misclassification cases than the classic "Accuracy" metric.

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- **Precision**: It is a measure of correctly identified positive cases from all predicted positive cases. Therefore, it is useful when the cost of false positive results is high (which is the case).

$$Precision = \frac{True\,Positives}{True\,Positives + False\,Positives}$$

- **Recall**: It is a measure of correctly identified positive cases from all real positive cases. This is important when the cost of false negatives is high.

$$Recall = \frac{True\,Positives}{True\,Positives + False\,Negatives}$$

- **Accuracy**: One of the more obvious indicators is the measurement of all correctly identified cases. Most commonly used when all classes are equally important, while F1-score is used when the False Positives or False Negatives are crucial.

4.1.4 Static parameters

Every experiment in the report (with or without augmentation) has the following configuration:

- **Maximum number of epochs**: 50. This number is chosen as a compromise between the runtime and classification performance.
- **Early stopping**: Monitoring validation loss with a patience of 10 epochs restoring the best epoch weights. This is used both to improve runtime and classification performance.
- **Adaptive Learning Rate**: Monitoring validation loss with a patience of 5 epochs with a factor of .5 and a minimum LR of 10^{-7}. This is used to complement some old algorithms like SGD that do not have this capability.
- **Loss function**: binary cross entropy. This is used since we are doing a binary classification in conjunction with a sigmoid last layer activation.
- **Classification layer**: 1-unit dense layer with sigmoid activation.

4.2 Neural Networks without data augmentation

The results from the three Neural Networks trained without augmentation and five fold cross-validation can be found bellow.

4.2.1 From Scratch with no augmentation

Wisi forensibus mnesarchum in cum. Per id impetus abhorreant, his no magna definiebas, inani rationibus in quo. Ut vidisse dolores est, ut quis nominavi mel. Ad pri quod apeirian concludaturque, id timeam iudicabit rationibus pri. Erant putant luptatum ex sit, error euismod ad qui, meliore voluptatum complectitur an vix. Clita persius sed et, vix vidit consulatu complectitur ex. Per nonummy postulant assentior an, mea audiam fabellas deserunt id.

Table 3: **Results from Scratch Model model with no augmentation**

Dropout	**Dense Layer units**	**Optimizer**	**Epochs**	**Learning Rate**	**Accuracy**	**Recall**	**Precision**	**F1 score**
.7	0	Adam	50	10^{-5}	.7811	.7826	.7476	**.7637**
0	128	Adam	50	10^{-4}	.7936	.7968	.7604	**.7778**
0	0	SGD	50	5×10^{-3}	.7777	.8211	.7251	**.7688**
0	128	RMSprop	50	10^{-4}	.7754	.7550	.7513	**.7520**
.1	128	Adam	50	10^{-5}	.8031	.7985	.7765	**.7849**
.5	128	Adam	50	10^{-4}	.7925	.8068	.7539	**.7789**
.5	64	Adam	50	10^{-4}	.8217	.8687	.7695	**.8159**

Eam ex integre quaeque bonorum, ea assum solet scriptorem pri, et usu nonummy accusata interpretaris. Debitis necessitatibus est no. Eu probo graeco eum, at eius choro sit, possit recusabo corrumpit vim ne. Noster diceret delicata vel id.

4.2.2 ResNet50 with no augmentation

A slightly improvement was achieved with the first Transfer Learning method, presiding from the use of the Dropout Regularization Technique in the case of the second model, with a F1 score of .822, taking into account the same amount of epochs the difference that can be seen in Table 3.

Table 4: **Results from ResNet50 model with no augmentation**

Dropout	**Dense Layer units**	**Optimizer**	**Epochs**	**Learning Rate**	**Accuracy**	**Recall**	**Precision**	**F1 score**
.7	0	Adam	50	10^{-5}	.8058	.7959	.7849	**.7857**
0	128	Adam	50	10^{-4}	.8334	.8453	.8007	**.822**
0	0	SGD	50	5×10^{-3}	.7207	.6863	.6805	**.6811**
0	128	RMSprop	50	10^{-4}	.8247	.8286	.7954	**.8111**
.1	128	Adam	48	10^{-5}	.8354	.8386	.8067	**.822**

The SGD model was the one performing the worst, with an F1 score of .6811, although this is expected, taking into account that it started with a higher LR and still using a dynamic LR in all the models, ADAM perform better for all of them. The RMSprop have had to a fair performance as both it also use adaptive methods for convergence. However, not using a Dropout or using as little as .1 seem to have an improvement in the overall performance of the network which can be notice with the last model, achieving the same F1 score but with slightly less recall with respect to the second one.

4.2.3 InceptionV4 with no augmentation

Surprisingly, the best results, as measured by the F1 score were obtained using the Stochastic gradient descent (SGD) with a high learning rate (LR) and without additional dropout or dense layers between the last flatten layer and final classifier layer, so it is just using the base network and their weights as-is (the inceptionv4 layers weights are frozen). The potential issue of having a high LR could be mitigated by the use of the dynamic LR reducer callback but still, we expect ADAM, a more modern algorithm with built-in LR adjustment and a lower starting LR to perform better, however, but this was not the case having an advantage of over .01 F1 score.

Table 5: **Results from InceptionV4 model with no augmentation**

Dropout	**Dense Layer units**	**Optimizer**	**Epochs**	**Learning Rate**	**Accuracy**	**Recall**	**Precision**	**F1 score**
.7	0	Adam	48	10^{-5}	.7667	.6957	.7682	**.7288**
0	128	Adam	50	10^{-4}	.8095	.7968	.7887	**.7910**
0	0	SGD	50	5×10^{-3}	.8232	.7926	.8143	**.8026**
0	128	RMSprop	50	10^{-4}	.8065	.7651	.7999	**.7818**
.1	128	Adam	50	10^{-5}	.8008	.7667	.7888	**.7769**

As for the worst result, the first one, it behaved very poorly because of 70% dropout, which was too excessive to prevent any posible overfitting. Continueing with another algorithm, RMSprop which could be thought as an in-between between SGD and ADAM, as it also has an adaptive LR, overall performed the worst but not to bar behind ADAM, only around .01 F1-score but further hyper-parameter tuning could make it perform closer to ADAM.

4.3 Neural Networks using data Augmentation

Now using data augmentation the results are these:

4.3.1 From Scratch with augmentation

Wisi forensibus mnesarchum in cum. Per id impetus abhorreant, his no magna definiebas, inani rationibus in quo. Ut vidisse dolores est, ut quis nominavi mel. Ad pri quod apeirian concludaturque, id timeam iudicabit rationibus pri. Erant putant luptatum ex sit, error euismod ad qui, meliore voluptatum complectitur an vix. Clita persius sed et, vix vidit consulatu complectitur ex. Per nonummy postulant assentior an, mea audiam fabellas deserunt id. Eam ex integre quaeque bonorum, ea assum solet scriptorem pri, et usu nonummy accusata interpretaris. Debitis necessitatibus est no. Eu probo graeco eum, at eius choro sit, possit recusabo corrumpit vim ne. Noster diceret delicata vel id. Here will be a table with the results from internal 5 test

Table 6: **Results from Scratch model with augmentation**

Dropout	**Dense Layer units**	**Optimizer**	**Epochs**	**Learning Rate**	**Accuracy**	**Recall**	**Precision**	**F1 score**
.7	0	Adam	50	10ˆ-5	.8058	.7959	.7849	**.7857**
0	128	Adam	50	10ˆ-4	.8334	.8453	.8007	**.822**
0	0	SGD	50	5·10ˆ-4	.7207	.6863	.6805	**.6811**
0	128	RMSprop	50	10ˆ-4	.8247	.8286	.7954	**.8111**
.1	128	Adam	48	10ˆ-5	.8354	.8386	.8067	**.822**

4.3.2 ResNet50 with augmentation

Wisi forensibus mnesarchum in cum. Per id impetus abhorreant, his no magna definiebas, inani rationibus in quo. Ut vidisse dolores est, ut quis nominavi mel. Ad pri quod apeirian concludaturque, id timeam iudicabit rationibus pri. Erant putant luptatum ex sit, error euismod ad qui, meliore voluptatum complectitur an vix. Clita persius sed et, vix vidit consulatu complectitur ex. Per nonummy postulant assentior an, mea audiam fabellas deserunt id. Eam ex integre quaeque bonorum, ea assum solet scriptorem pri, et usu nonummy accusata interpretaris. Debitis necessitatibus est no. Eu probo graeco eum, at eius choro sit, possit recusabo corrumpit vim ne. Noster diceret delicata vel id. Here will be a table with the results from internal 5 test

Table 7: **Results from ResNet50 model with augmentation**

Dropout	**Dense Layer units**	**Optimizer**	**Epochs**	**Learning Rate**	**Accuracy**	**Recall**	**Precision**	**F1 score**
.7	0	Adam	48	10ˆ-5	.7667	.6957	.7682	**.7288**
0	128	Adam	50	10ˆ-4	.8095	.7968	.7887	**.7910**
0	0	SGD	50	5·10ˆ-4	.8232	.7926	.8143	**.8026**
0	128	RMSprop	50	10ˆ-4	.8065	.7651	.7999	**.7818**
.1	128	Adam	50	10ˆ-5	.8008	.7667	.7888	**.7769**

Table 8: **Results from InceptionV4 model with augmentation**

Dropout	**Dense Layer units**	**Optimizer**	**Epochs**	**Learning Rate**	**Accuracy**	**Recall**	**Precision**	**F1 score**
.7	0	Adam	48	10ˆ-5	.7667	.6957	.7682	**.7288**
0	128	Adam	50	10ˆ-4	.8095	.7968	.7887	**.7910**
0	0	SGD	50	5·10ˆ-4	.8232	.7926	.8143	**.8026**
0	128	RMSprop	50	10ˆ-4	.8065	.7651	.7999	**.7818**
.1	128	Adam	50	10ˆ-5	.8008	.7667	.7888	**.7769**

4.3.3 InceptionV4 with augmentation

Eam ex integre quaeque bonorum, ea assum solet scriptorem pri, et usu nonummy accusata interpretaris. Debitis necessitatibus est no. Here will be a table with the results from internal 5 test

4.4 Model comparisons

Table 9: Best Models comparison

Models	**Optimizer**	**Epochs**	**Accuracy**	**Recall**	**Precision (%)**	**F1 score**
Scratch	Adam	14	0.91	88.8	100	**0.91**
ResNet50	SDG	36	0.91	0.91	100	**0.01**
InceptionV4	RMSprop	50	0.91	0.91	74.93	**0.91**

Wisi forensibus mnesarchum in cum. Per id impetus abhorreant, his no magna definiebas, inani rationibus in quo. Ut vidisse dolores est, ut quis nominavi mel. Ad pri quod apeirian concludaturque, id timeam iudicabit rationibus pri. Erant putant luptatum ex sit, error euismod ad qui, meliore voluptatum complectitur an vix. Clita persius sed et, vix vidit consulatu complectitur ex. Per nonummy postulant assentior an, mea audiam fabellas deserunt id. Eam ex integre quaeque bonorum, ea assum solet scriptorem pri, et usu nonummy accusata interpretaris. Debitis necessitatibus est no. Eu probo graeco eum, at eius choro sit, possit recusabo corrumpit vim ne. Noster

Confusion Matrix Results

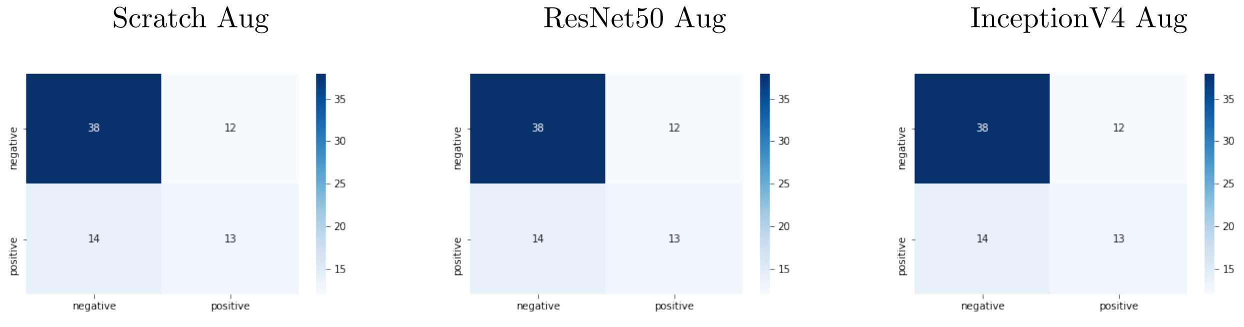

Figure 1: Confusion matrix for the three models.

diceret delicata vel id. Here will be a table with comparison from best results Also a graphic and Confusion matrix Here we compare the three augmentation models

5 Strengths and Weaknesses

Once the theory is understood and applied, with the code and software set, the model ready and the dataset pre-processed, it can be realized that the real work with Neural Networks has not started in the sense that, the better or even acceptable hyperparameters are not yet being set, so the discussion about experimental issues has to gain relevance with respect to the work presented here.

First things first, an attempt of working with our own image augmentations script was performed with the dataset and it was found that the CNNs were very sensitive to the changes performed (various filters, gamma, translations, rotations). Luckily a suited and robust function was already build-in on Keras that performed state-of-the-art augmentation with slight or heavy changes (depending on the parameters used), overall allowed the Network to learn the under-laying patterns in the data.

With respect to the model selection which encompass the Architecture, Activation Function, Learning Rate, Optimizer etc. There is not a better way to do a selection of the best parameters than by trial and error. Therefore, figuring out the performance and its behaviour with respect to only few parameters or a combination of many is something in which this study is carried out exhaustively, taking into account that several trials were performed along hours and days to test as much parameters and combinations in parallel as possible with the limited resources.

A stratified k-fold was performed to avoid, within the possibilities, having biased models to the particular data that is contained in the training and validation dataset.

A particular weakness of our system was not including domain knowledge. As none of the team is a dermatologist nor has any dermatological training, we were limited to general machine-learning techniques. One of the papers that had a very impressive accuracy in identification of benign vs. malignant moles of above 96%[7] adapted some techniques used by dermatologists to decide whether a lesion merits a further look. In particular, they used the ABCD rules (A - asymmetry, B - border, C - color and D - diameter) in order to extract features to then be used as inputs to an artificial neural network. There are certainly many more examples of other such details that could be incorporated and any serious follow-up work that seeks to advance the current state-of-the-art should consult with experts in the field in order to ensure that applicable domain knowledge is applied if appropriate.

...

Be critical of your work...

6 Conclusions

We have explored the application of Neural Networks in the medical field, with the focus on classification of cancerous and non-cancerous skin marks using images. The data used to evaluate this classification consist of 3,297 images, 1,800 for benign and 1,497 for malignant. For these two category it was divided in 80% and 20% for train and test sets, respectably. In addition, we evaluate if the images augmented performed better for the same tested neural network models. The image augmentation was conducted primary by normalizing the images and modifying theirs properties (gamma, scale, filters). The test conducted comprehend of traditional convolution neural with X layers, ResNet network and InceptionV4 model. The results achieved using the data without augmentation we obtained X% of accuracy in standard model, X% for ResNEt and in InceptionV4

X% accuracy. In the other hand for the augmented images we achieved X% in standard model, X% for ResNet and X% accuracy in InceptionV4. As it was expected the better accuracy's was obtained in the augmented images due to the removal of noise in the evaluated images. When evaluating the models the best perform model is the InceptionV4, this results was expected due to be a state of the art model and having a more complex layer architecture. With this evaluation we have probe the easy scalability of the neural network models in medical and similar implementations.

Is worth mentioning that our system have some limitations that would need to be addresses and taken in to consideration for future development and application. On of the because concern of the team at the moment of evaluating the data was the lack of diversity of the training data. For example all the data tested was from people of white skin color, only having this has training data would cause that the model perform poor in different skins color samples. Another team concern is the lack of "complex" scenarios if the sample data. For addressing these concern, we highly recommended that at the moment of selecting the training data it is taken taken form diverse and various sources. Our initial model results, show a high accuracy, but there is still a large gap between the accuracy desired for a medical application of a +99% due to the importance of the data evaluate, cancerous and non-cancerous skin marks. Finally, further scaling of the tested model would likely lead to improved performance.

Considering the fact that dermatologists are rarely able to surpass the 80% threshold in skin cancer diagnosis from visual inspection alone[3], the results we have obtained are very encouraging for the widespread adoption of CNNs as an aid to improve skin cancer detection rates in the future. With the help of a dermatoscopic camera, dermatologists are able to improve their accuracy a bit further and may reach approximately 84%, a number our models have approximated. However, our models still have too high of a false negative rate to truly be comparable to a trained dermatologist. The next section includes some ideas on how that might be able to be turned into a reality.

6.1 Future Work

Naturally, being our goal of combining the use of a CNN with ubiquitous technology in order to provide medical advice to patients from the comfort of their own home, an approximation was made through the training of a lightweight Convolutional Neural Network, namely the MobileNetV2 designed and optimized for classification purposes in mobile devices, was trained with augmentation and the best parameters found above and was compiled in an simple android .apk. This network performance was not the best given the limitations of the mobile app, but a fair prediction result was achieved when images are shown from the test dataset. Ideally, this technology would be integrated into an online diagnostic tool or a mobile app that would provide usable, fast and accurate guidance about whether the patient should seek professional advice regarding a suspicious lesion. The demo installer only for android devices along with additional documentation can be found in oursite.com.

The idea of using additional inspection techniques and even other information leads to some interesting ideas for follow-on work. If a dataset could be created with the images, or possibly even videos, from dermatoscopic cameras, it may provide significantly more information to train neural networks and improve their results even further past the state-of-the-art today. There may also be valuable information about skin lesions outside the visible spectrum. The inspiration for this idea comes from a study about insect vision[11] where pictures of flowers taken without a UV filter are displayed and oftentimes have strong differentiating features in the UV spectrum that are invisible to the naked eye. Gathering a dataset of images cancerous vs. non-cancerous lesions' responses to many wavelengths outside of the range of visible light may lead to the discovery of features unobservable by the human eye. Another missing component in the kaggle dataset, and in fact many of the datasets used in other studies, is metadata providing more context for each image, such as the age, race, location on body, medical history of patient/relatives, etc., which are all pieces of information a dermatologist has access to when making a decision. It is unclear how exactly this extra metadata could be injected directly into a convolutional neural network, but perhaps there could be a hybrid architecture that treats part of the input as an image-processing task and then combines the output of the image processing with the metadata as inputs to a classification system.

A Proofs

(if applicable)

B Implementation Details

(if applicable)

References

[1] Skin Cancer Foundation. *Skin Cancer Facts & Statistics: What You Need to Know.* URL: `https://www.skincancer.org/skin-cancer-information/skin-cancer-facts/`. (accessed: 30.12.2020).

[2] International Skin Imaging Collaboration. *International Skin Imaging Collaboration.* URL: `https://www.isic-archive.com/`. (accessed: 30.12.2020).

[3] Titus Josef Brinker et al. "Skin Cancer Classification Using Convolutional Neural Networks: Systematic Review". In: *J Med Internet Res* 20.10 (Oct. 2018), e11936. ISSN: 1438-8871. DOI: `10.2196/11936`. URL: `http://www.ncbi.nlm.nih.gov/pubmed/30333097`.

[4] Afonso Menegola et al. "Knowledge Transfer for Melanoma Screening with Deep Learning". In: *CoRR* abs/1703.07479 (2017). arXiv: `1703.07479`. URL: `http://arxiv.org/abs/1703.07479`.

[5] Ni Zhang et al. "Skin Cancer Diagnosis Based on Optimized Convolutional Neural Network". In: *Artificial Intelligence in Medicine* 102 (Nov. 2019), p. 101756. DOI: `10.1016/j.artmed.2019.101756`.

[6] Andre Esteva et al. "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542 (Jan. 2017). DOI: `10.1038/nature21056`.

[7] T. Kanimozhi and Dr. A. Murthi. "COMPUTER AIDED MELANOMA SKIN CANCER DETECTION USING ARTIFICIAL NEURAL NETWORK CLASSIFIER". In: 2016.

[8] K. Fukushima. "Neocognitron". In: *Scholarpedia* 2.1 (2007). revision #91558, p. 1717. DOI: `10.4249/scholarpedia.1717`.

[9] Taghi M. Khoshgoftaar Karl Weiss and DingDing Wang. "A survey of transfer learning". In: *Journal of Big Data* 3.1 (2016), p. 1717. DOI: `10.1186/s40537-016-0043-6`.

[10] Taghi M. Khoshgoftaar Karl Weiss and DingDing Wang. "Beyond Accuracy, F-score and ROC: a Family of Discriminant Measures for Performance Evaluation". In: *Advances in Artificial Intelligence, Lecture Notes in Computer Science* 4304 (2006). DOI: `10.1007/11941439_114`.

[11] James Lincoln and Andrew Davidhazy. "Ultraviolet photography and insect vision". In: *The Physics Teacher* 57 (Mar. 2019), pp. 204–205. DOI: `10.1119/1.5092494`.