Università degli Studi di Milano

Data Science and Economics

Deep Learning for automated diagnosis of pigmented skin lesions

Guido Giacomo Mussini
988273

ACADEMIC YEAR 2022-2023

Abstract

The purpose of the report is to analyze images concerning different types of skin lesions, with the dual goal of creating models able to classify them and define whether they are benign or malignant. The preprocessing phase focused on data augmentation and class balancing. The models used were convolutional neural networks, which provided results that were not entirely satisfactory. In particular, the performance on unseen data recorded about 71.3% accuracy for binary classification and 65% for multi-class classification. In order to maximize the results, a simple hyperparameter tuning over the Dropout level has been performed.

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

Contents

1 Research Questions

This work has two main goals.
The first one, more specific, is to train a model capable of recognizing, as accurately as possible, different types of skin lesions.
The second one, more general, is to distinguish between benign and malignant lesions.

For these reasons, the project will develop in two parallel ways, seeking to apply the best techniques to achieve each of the two goals.

2 The Data

2.1 Dataset

The Data is a collection of dermatoscopic images from different populations, shared in a Dataset called *HAM10000* [1]. The original source of these images is the ISIC (International Skin Imaging Collaboration) Archive [2], which serves as a public resource of medical images for different purposes.

The Dataset is composed by 10015 images and a metadata table, containing the lables of the images and other information such as the body position of the skin lesion and the age of the patient.
For the purpose of this analysis, there have been considered the images and the labels only.

2.2 Classes

The images are divided in 7 classes representing different types of skin lesions. Specifically:

- **Actinic keratoses and intraepithelial carcinoma (akiec)**: Actinic keratoses are common, sun-induced lesions that have historically been

regarded as "**premalignant**." Evidence supports their inclusion along a continuum with squamous cell carcinoma. [3]

- **Basal cell carcinoma (bcc)**: the most common form of **skin cancer**, it rarely metastasize to other parts of the body. [4]
  However, if not treated properly and timely, it can be deadly. [5]
- **Benign keratosis-like lesions (bkl)**: it is a common **benign skin growth**, it is not pre-cancerous, but they can resemble other skin growths that are. [6]
- **Dermatofibroma (df)**: Dermatofibromas are referred to as **benign fibrous histiocytomas** of the skin, superficial/cutaneous benign fibrous histiocytomas, or common fibrous histiocytoma. [7]
- **Melanoma (mel)**: Melanoma is a type of **malignant skin cancer** that develops in the skin cells called melanocytes and usually occurs on the parts of the body that have been overexposed to the sun. [8]
- **Melanocytic nevi (nv)**: Melanocytic nevi (pigmented mole, banal nevus, common nevus, and acquired nevus) are **benign melanocytic neoplasms** that commonly present as flat or raised, brown to black pigmented lesions. [9]
- **Vascular lesions: angiomas, angiokeratomas, pyogenic granulomas and hemorrhage (vasc)**: Vascular lesions are relatively common abnormalities of the skin and underlying tissues, more commonly known as birthmarks. They can be considered **benign lesions**. [10]

These lesions can be therefore grouped in

- **Benign Lesions (M)**: bkl, df, nv, vasc
- **Malignant Lesions (B)**: akiec, bcc, mel

The following graphs show the distribution of the classes. The green bars represent the Benign Lesion, while the red ones the Malignant.

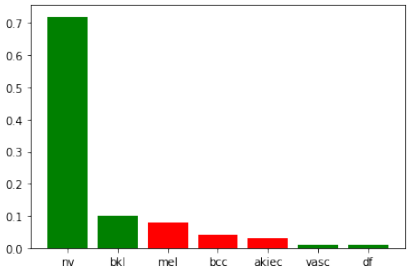

Figure 1: 7 classes distribution

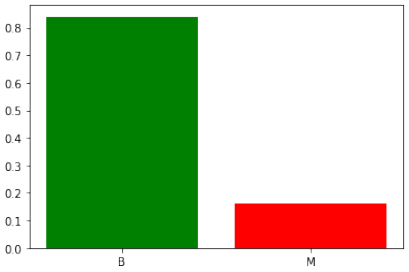

Figure 2: Binary distribution

It can be seen that the distribution is very unbalanced in both cases. Numerically, the situation is:

-	**akiec**	**bcc**	**bkl**	**df**	**mel**	**nv**	**vasc**	**Sum**
Benign	-	-	0.10	0.01	-	0.72	0.01	**0.84**
Malignant	0.03	0.04	-	-	0.08	-	-	**0.16**

Where the table represents the percentages of each Lesion.
The strong majority of the lesions are benign, and the most common one is, by far, the *Melanocytic nevi (nv)*.
This asymmetric situation will complicate the learning process of the models. For example, it is more likely that, statistically, many more benign lesions than malignant lesions will be drawn in each mini-batch, leading the model to be biased toward benign lesions.
Another issue concerning the Multiclass classification problem is that one class is much more frequent than the others. This means that to rebalance the classes, all the others will have to be oversampled, as the scarcity of data does not allow for undersampling.

3 Data Manipulation

3.1 Missing Values and duplicates

Before proceeding with any analysis, it has been checked the presence of missing values and Duplicate rows.
Only 52 Missing values were found in the '*Age*' column, however, since that column was not used in the analysis, they were not imputed.

With regard to the presence of duplicates, it has been performed a check on both the columns '*lesion_id*' and the '*image_id*', in order to avoid different images of the same lesion, or the same lesion represented in more than one row.
Approximately 2,500 duplicates were found and removed. Precisely, the procedure reduced the rows in the dataset to 7470.
This small amount of data could not be sufficient to ensure a adequate learning process, so synthetic data will be generated.

3.2 Shape

The images, originally shaped as (75, 100, 3), were resized to the **shape** of (32, 32, 3), since this framework ensures a good balance between computational speed and information loss.

3.3 Train, Validation and Test set

The dataset has been divided, initially, in training and validation set, where the training set will in turn be divided into train and validation set, after data augmentation.
At this stage the situation, for both the Binary classification and Multiclass Classification, is:

- Training set: 5229 observations
- Test set: 2241 observations

Note that the dataset has been split maintaining the distribution of the classes.

3.4 Data Augmentation

This section will address the scarcity of data in the Dataset.
Deep learning models requires to be trained over a huge amount of data in order to avoid overfitting, and taking into account that the dataset has to be split in Train, Validation and Test Set, 7470 images are not sufficient to train a model with decent generalization capability.

There are several methods for doing data augmentation, however, one must try to figure out which transformations are best suited to improve the model. Speaking of skin lesions, what differentiates them 'visually' are: **shape**, **color**, and **extent**.

Since **extension** is related to the distance and focus with which they were photographed, of which there is no indication in this dataset, one must focus on the first two features.

Color is particularly important and delicate; it can vary depending on the reflections and shadows generated by lesions, which, while to a human eye provide the 'sense of depth' of the image, they can confuse 'the eye' of the machine, that may mistake a red, shaded lesion for a black one, for example. Another key point is the color difference between the skin and the lesion, which again can mislead the model.
Finally, all lesions vary on gradations of especially red and black.
Given these premises, generating data artificially by transforming the color, such as inverting it, transposing it to a grayscale or adding noise, could worsen the model, which would be trained on *impossible* data

Shape can tell a lot about the type of lesion: the regularity of the edges, the number of 'spots' and the contour can be important indicators for distinguishing lesions.
An important characteristic of these features is that they can be considered much less dependent on the way the image was collected.
Therefore, to generate credible images, it is important to apply transformations that preserve distances, to maintain the shape.

The transformations that possess these characteristics are, for example, the flipping along the horizontal [11] and vertical [12] axis.
These transformations have been randomly applied to the whole Train and Validation set, so that the number of observations has doubled, reaching the amount of 10458.
Note that in the test set have been used only original images.

After this procedure, the Training set has been divided in Train e validation Set as follow:

- Train Set: 7320
- Validation Set: 3138
- Test Set: 2241

3.5 SMOTE

The second issue to be solved concerns the balancing of classes, the 2 main approaches used in these cases are **OverSampling** and **UnderSampling**.

Since even after doing data augmentation the dataset is not very large, it would be inconvenient to eliminate observations to balance the classes. For this reason, it was chosen to do OverSampling.
The technique chosen to perform the oversampling is called **SMOTE**. (Synthetic Minority Oversampling Technique) [13]

At this point a distinction must be made between the two classification problems: Binary and Multi-class classification.

Binary

For the Binary classification (figure 16), it is simple to oversample the less frequent class, the Malignant lesions class, since it is unique. The procedure creates a number of new training examples, equal to the difference between the cardinality of the Benign class in the train set minus the cardinality of the Malignant class in the train set. After this procedure, the Train Set cardinality is 12328.

The global situation for the binary classification is now:

- Binary Train: 12328
- Binary Validation: 3138
- Binary Test: 2241

MultiClass

In this framework there are more possibilities, one can decide to generate examples only of the less frequent class, or to generate examples of all the classes excluded the most frequent one.
Given the distribution of the classes in this case (figure 17), the second option seems to be best one, but since the most numerous class is much more numerous than the others, this would mean generating a large amount of synthetic data [1] of lower quality, generating worse predictions.
In particular, models trained after oversampling never provided an accuracy greater than 50% in the test set.
Therefore it has been decided to not augment the data in this scenario.

The global situation for the multi-class classification doesn't change:

- Multi-class Train: 7320
- Multi-class Validation: 3138
- Multi-class Test: 2241

3.6 Normalization and Parameters

The last preprocessing step was normalizing the data into train, validation, and test set.
This step usually improve the learning process.

[1] The train set would be composed of more than 26000 observations

4 CNN - Multiclass Classification

Framework

- 7 classes
- Augmented Data
- Normalized Data
- Unbalanced classes

4.1 Model N

Model N is a model inspired by the one [14] used to solve the same classification problem, but applied to data treated in a different way.

4.1.1 Architecture

The **loss function** used is the **Cross Entropy Loss**, while the **optimizer** is the **Adam optimizer**. Moreover, *reducelronplateau*[16] has been used to dynamically adjust the learning rate based on the improving rate of the train loss. The activation function for the output layer is the **Log-Softmax**.
The **batch size** it has been chosen equal to 64, and the models were trained for 100 **epochs**.

In figure 6 it can be observed the architecture of the model.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 32, 32]	896
ReLU-2	[-1, 32, 32, 32]	0
MaxPool2d-3	[-1, 32, 10, 10]	0
BatchNorm2d-4	[-1, 32, 10, 10]	64
Dropout-5	[-1, 32, 10, 10]	0
Conv2d-6	[-1, 64, 10, 10]	18,496
ReLU-7	[-1, 64, 10, 10]	0
MaxPool2d-8	[-1, 64, 3, 3]	0
BatchNorm2d-9	[-1, 64, 3, 3]	128
Dropout-10	[-1, 64, 3, 3]	0
Flatten-11	[-1, 576]	0
Linear-12	[-1, 1024]	590,848
ReLU-13	[-1, 1024]	0
BatchNorm1d-14	[-1, 1024]	2,048
Dropout-15	[-1, 1024]	0
Linear-16	[-1, 256]	262,400
ReLU-17	[-1, 256]	0
BatchNorm1d-18	[-1, 256]	512
Dropout-19	[-1, 256]	0
Linear-20	[-1, 42]	10,794
ReLU-21	[-1, 42]	0
BatchNorm1d-22	[-1, 42]	84
Dropout-23	[-1, 42]	0
Linear-24	[-1, 7]	301
LogSoftmax-25	[-1, 7]	0

Total params: 886,571
Trainable params: 886,571
Non-trainable params: 0

Figure 3: Model N Summary

4.1.2 Performance on train and validation set

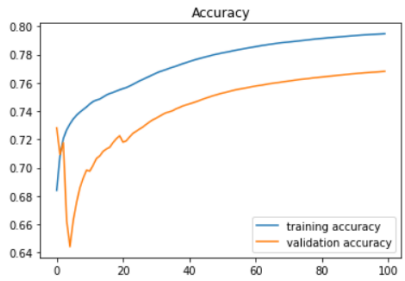

Figure 4: Accuracy

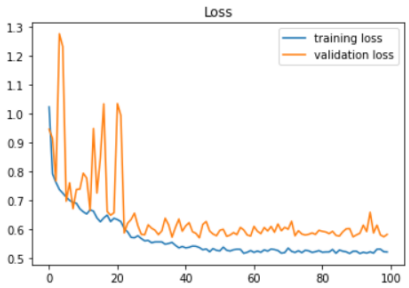

Figure 5: Loss

It can be observed that the model is overfitting, moreover, the validation loss seems to converge to a higher value than the training one.
The presence of the dropouts, the adjusting learning rate, and the fact that the validation loss is evaluated at the end of each epoch while the training one during each batch-iteration could explain some differences or strange behavior of the validation loss, but surely there is some difficulty in generalizing by the model

4.1.3 Performance on the test set

The performance on the test set was not very good. In fact, the model achieved an **accuracy of 57%**, producing the following confusion matrix:

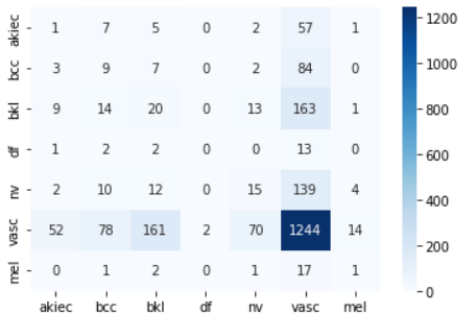

Figure 6: Multiclass Model N confusion matrix: the rows are the true label, while the column the predicted ones

the matrix is not easily interpretable, however it can be seen that the values

are scattered along the matrix, and not centered on the main diagonal, which represents the correct predictions.
The matrix also highlights the problem of unbalanced distribution among classes: a lot of cells result empty due to the lack of observation.
Moreover, it is evident that the most frequent class is predicted incorrectly almost uniformly in all other classes, highlighting a lack of factors usable by the algorithm to discriminate the classes.

4.2 Multiclass Model 1

4.2.1 Architecture

The framework is the **same as Model N**:
Cross Entropy Loss, Adam Optimizer, plateau-based learning rate adjustment, Log-SoftMax in the output layer. Batch size of 64 and model trained for 100 epochs.
Figure 6 shows the architecture of this model:

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 32, 32, 32]             896
              ReLU-2          [-1, 32, 32, 32]               0
         MaxPool2d-3            [-1, 32, 6, 6]               0
       BatchNorm2d-4            [-1, 32, 6, 6]              64
         Dropout-5              [-1, 32, 6, 6]               0
            Conv2d-6            [-1, 64, 6, 6]          18,496
              ReLU-7            [-1, 64, 6, 6]               0
         MaxPool2d-8            [-1, 64, 2, 2]               0
       BatchNorm2d-9            [-1, 64, 2, 2]             128
        Dropout-10              [-1, 64, 2, 2]               0
           Conv2d-11           [-1, 128, 2, 2]          73,856
             ReLU-12           [-1, 128, 2, 2]               0
        MaxPool2d-13           [-1, 128, 1, 1]               0
      BatchNorm2d-14           [-1, 128, 1, 1]             256
        Dropout-15             [-1, 128, 1, 1]               0
          Flatten-16                  [-1, 128]               0
           Linear-17                   [-1, 64]           8,256
             ReLU-18                   [-1, 64]               0
      BatchNorm1d-19                   [-1, 64]             128
        Dropout-20                     [-1, 64]               0
           Linear-21                   [-1, 42]           2,730
             ReLU-22                   [-1, 42]               0
      BatchNorm1d-23                   [-1, 42]              84
        Dropout-24                     [-1, 42]               0
           Linear-25                    [-1, 7]             301
       LogSoftmax-26                    [-1, 7]               0
================================================================
Total params: 105,195
Trainable params: 105,195
Non-trainable params: 0
----------------------------------------------------------------
```

Figure 7: Multiclass Model 1 summary

4.2.2 Performance on train and validation set

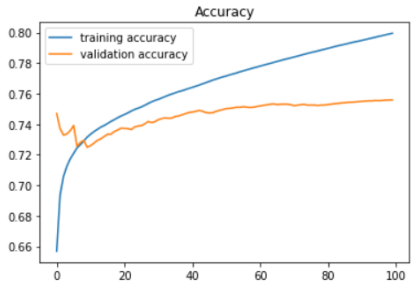

Figure 8: Accuracy

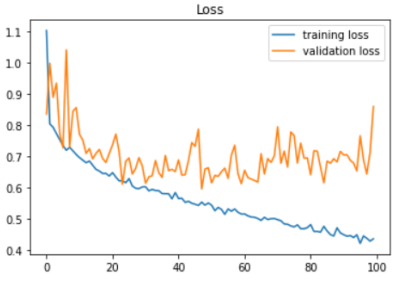

Figure 9: Loss

The situation is similar to the one observed for Model N: Model 1 is clearly overfitting, and the loss curve has an even more irregular pattern.

4.2.3 Performance on the test set

The performance on the test set, however, was better than the Model N one, in fact Model 1 reached an **accuracy of 64.8%**.

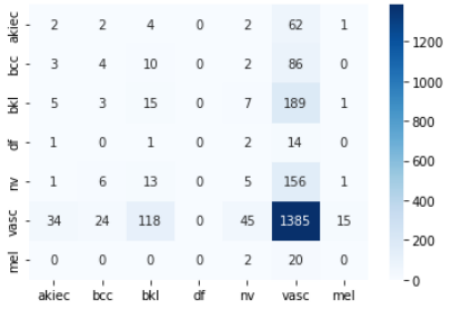

Figure 10: Multiclass Model 1 summary: the rows are the true label, while the column the predicted ones

the situation is slightly improved, but the problems are evident and they are the same ones described for the previous Model.

4.3 Multiclass Summary

The results obtained in this phase are strongly limited by the dataset used to train the models.
The dataset would probably require to be furhter augmented and, mainly, balanced in a proper way.

Recalling the considerations of section 3.4, a possible way to balance the dataset could be to generate synthetic data by randomly rotate the images of the less frequent classes.
Doing so would avoid, with high probability, to generate two times the same image, keeping unchanged the distances and proportions of the images

5 CNN - Binary Classification

This section will present the models and results of binary classification. This generalization was made in order to try to simplify the learning process.

Framework

- 2 classes
- Augmented Data
- Normalized Data
- SMOTE - Balanced Classes

5.1 Binary Model 1

5.1.1 Architecture

The model uses the **stochastic gradient descent** as optimizer, while the loss function is the **binary cross entropy**. The learning rate is adjusted in the same way seen above, using the **plateau method** to adjust it during the training. **Sigmoid** is the activation function of the output layer, which is a single neuron.
The batch size is 64 while the model is trained for 100 epochs.
Below the architecture of the model:

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 16, 32, 32]	448
ReLU-2	[-1, 16, 32, 32]	0
MaxPool2d-3	[-1, 16, 10, 10]	0
BatchNorm2d-4	[-1, 16, 10, 10]	32
Dropout-5	[-1, 16, 10, 10]	0
Conv2d-6	[-1, 32, 10, 10]	4,640
ReLU-7	[-1, 32, 10, 10]	0
MaxPool2d-8	[-1, 32, 3, 3]	0
BatchNorm2d-9	[-1, 32, 3, 3]	64
Dropout-10	[-1, 32, 3, 3]	0
Conv2d-11	[-1, 64, 3, 3]	18,496
ReLU-12	[-1, 64, 3, 3]	0
BatchNorm2d-13	[-1, 64, 3, 3]	128
Dropout-14	[-1, 64, 3, 3]	0
Conv2d-15	[-1, 128, 3, 3]	73,856
ReLU-16	[-1, 128, 3, 3]	0
BatchNorm2d-17	[-1, 128, 3, 3]	256
Dropout-18	[-1, 128, 3, 3]	0
Flatten-19	[-1, 1152]	0
Linear-20	[-1, 64]	73,792
ReLU-21	[-1, 64]	0
BatchNorm1d-22	[-1, 64]	128
Dropout-23	[-1, 64]	0
Linear-24	[-1, 32]	2,080
ReLU-25	[-1, 32]	0
BatchNorm1d-26	[-1, 32]	64
Dropout-27	[-1, 32]	0
Linear-28	[-1, 16]	528
ReLU-29	[-1, 16]	0
BatchNorm1d-30	[-1, 16]	32
Dropout-31	[-1, 16]	0
Linear-32	[-1, 1]	17
Sigmoid-33	[-1, 1]	0

Total params: 174,561
Trainable params: 174,561
Non-trainable params: 0

Figure 11: Binary Model 1 summary

5.1.2 Performance on train and validation set

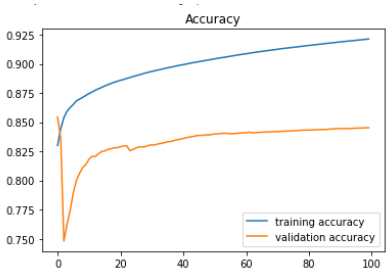

Figure 12: Accuracy

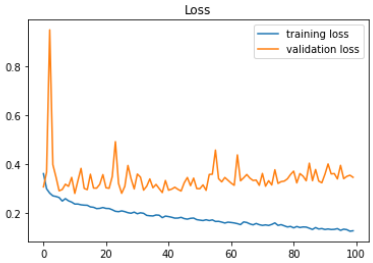

Figure 13: Loss

The model is clearly overfitting, but here we reach higher levels of accuracy than in the multivariate case.
The validation loss is scattered, and it does not converge to the training loss, this behavior could be another indicator of the small number of examples it is trained on, not enough for the model to learn.

5.1.3 Performance on the test set

The performance on the test set are slightly better than the multivariate case, the **accuracy** here is 73.1%.
As it can be observed in the confusion matrix, the model seems to fail in particular to classify malignant lesions.
The most disturbing finding, with a view to using these techniques in real-world settings, is the very high percentage of malignancies predicted as benign: In fact, only 23.6% of malignant lesions are predicted as such. This error is obviously much more serious than the opposite case.

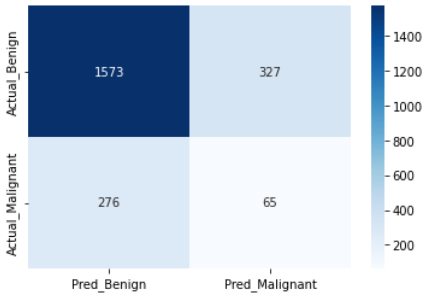

Figure 14: Model 1: Confusion Matrix

5.2 Binary Model 2

5.2.1 Architecture

The loss function is the **binary cross entropy** and the **plateau method** it has been chosen to adjust the learning rate.
Unlike what was done with the first model, here **Adam** was used as optimizer.
As before, the **Sigmoid** activation function has been used in the output layer, represented by a single neuron.
The batch size is 64 and the model has been trained for 100 epochs. Below the architecture.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 128, 32, 32]           3,584
              ReLU-2         [-1, 128, 32, 32]               0
         MaxPool2d-3         [-1, 128, 10, 10]               0
       BatchNorm2d-4         [-1, 128, 10, 10]             256
         Dropout-5           [-1, 128, 10, 10]               0
            Conv2d-6          [-1, 64, 10, 10]          73,792
              ReLU-7          [-1, 64, 10, 10]               0
         MaxPool2d-8            [-1, 64, 5, 5]               0
       BatchNorm2d-9            [-1, 64, 5, 5]             128
         Dropout-10             [-1, 64, 5, 5]               0
         Flatten-11                 [-1, 1600]               0
          Linear-12                   [-1, 64]         102,464
            ReLU-13                   [-1, 64]               0
      BatchNorm1d-14                   [-1, 64]             128
         Dropout-15                   [-1, 64]               0
          Linear-16                   [-1, 32]           2,080
            ReLU-17                   [-1, 32]               0
      BatchNorm1d-18                   [-1, 32]              64
         Dropout-19                   [-1, 32]               0
          Linear-20                    [-1, 1]              33
         Sigmoid-21                    [-1, 1]               0
================================================================
Total params: 182,529
Trainable params: 182,529
Non-trainable params: 0
```

Figure 15: Binary Model 2 summary

5.2.2 Performance on train and validation set

The performances seems to be similar to the previous one, the general level of accuracy is lower, but validation loss seems to be more smooth.

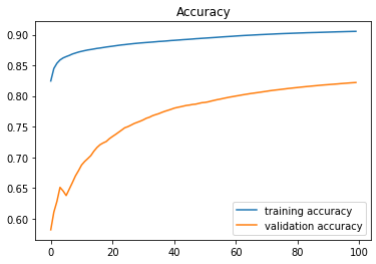

Figure 16: Accuracy

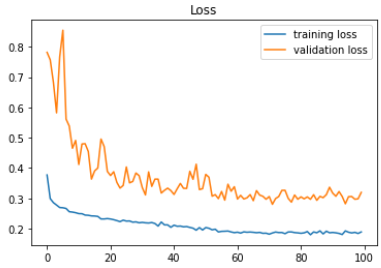

Figure 17: Loss

5.2.3 Performance on the test set

The **accuracy** obtained is 67.3%, which is lower than Model 1, but here the situation regarding the false negative[2] is slightly better, although it remains negative: only the 32,7% of the actual malignant lesions is predicted as such.

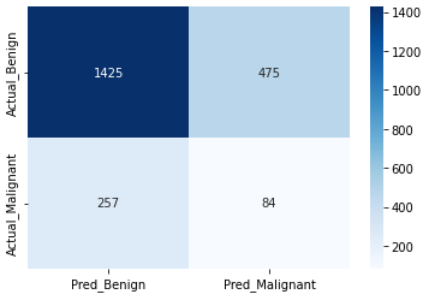

Figure 18: Model 2: Confusion Matrix

5.3 Binary Summary

The overall performance in this context is better than in the 7-class problem. However, overfitting problems persist, resulting in poor generalization ability. Another problem in this framework could be the use of SMOTE: the dataset is in fact trained on a balanced train set, when in reality the classes are not.

Taking accuracy as a reference metric, the best model turned out to be

[2]actual malignant lesions predicted as benign

Model 1. Therefore, as a last analysis, hyperparameter tuning on the Model 1 architecture will be done, trying different dropout values, which should provide a meaningful impact on overfitting.

5.4 Hyperparameter tuning

A simple tuning of the dropout level has been performed. The effects of all dropout values between 0 and 0.5, with steps of 0.05, were evaluated.

5.4.1 Description of method and metric

Overfitting relates to the difficulty of the model to perform well with unseen data; it can be visualized as the distance between the curves of training accuracy ($\mathbf{t}$) and validation accuracy ($\mathbf{v}$). Therefore, a metric based on this concept was defined.
First, the distance between the two arrays was calculated[3] and then multiplied by 100:

$$\mathbf{diff} = (\mathbf{t} - \mathbf{v}) \cdot 100$$

The multiplication aims to make almost all the values greater than 1, which is useful mostly for visualization purposes. The next step is to take the square of **diff**:

$$\mathbf{diff\ square} = \mathbf{diff}^2.$$

This exploit the property of the square function of stretch the bigger distance, so 'punishing' them in a minimization context. After this, the metric has been obtained by taking the median of **diff square**:

$$\text{Overfitting Metric} = Median(\mathbf{diff\ square})$$

The median has been preferred to the mean since it is less sensitive to outliers, that in this case can occur in the first epochs.

The idea is that the model which has the lowest 'level of overfitting' is the

[3] the difference is computed point by point.

model with the dropout which minimize this metric.

5.4.2 Results

This metric has been evaluate using a model with the same architecture of the Model described in section 5.1, Model 1[4]. Figure 19 shows the results of the computation of this metric.

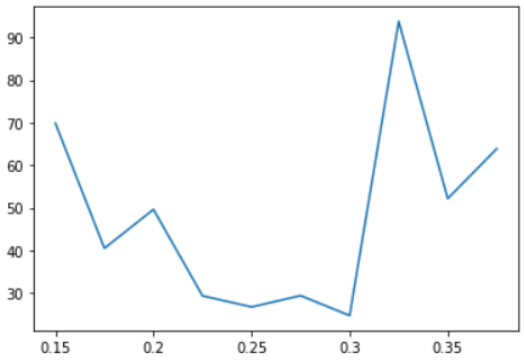

Figure 19: Overfitting Metric

It can be seen that the minimum is reached by a dropout level of 0.3.

The model was then evaluated on the test set, achieving an accuracy of 67.3%.
The resulting confusion matrix is similar to the one obtained by Model 2.

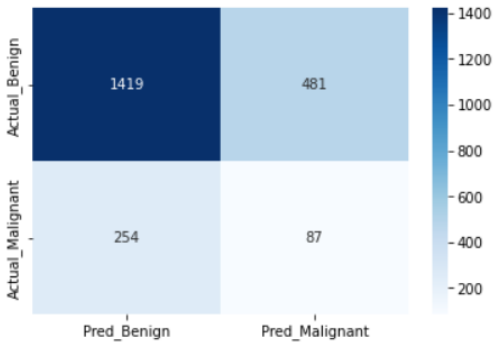

Figure 20: Model with tuned Dropout: Confusion matrix

[4]trained for 50 epochs.

It seems that the results have not improved. This may mean that the main cause of overfitting is to be found in the construction of the train set, or in the complexity of the models even if, for example, 'Model N' performs well in a similar context, in which the only big difference is the dataset composition. To confirm this hypothesis, a grid search could be performed on several parameters, such as optimizer, loss functions, pooling size, e.g.

5.4.3 Limitations of this approach

Note that this method may itself suffer from overfitting, since it evaluates a metric on a fixed training set and validation set.
Theoretically it finds the dropout value that minimizes the metric on that specific training set. To obtain more accurate results, one should perform a cross validation [15], which, however, is computationally time-consuming.

Moreover, note that dropouts are not fixed, but turn off different neurons, randomly, at each iteration. This can lead to different results in subsequent trials.

6 Conclusions and future works

As expected, generalizing the system to a binary classification problem increased the performance of the algorithms. The biggest problem is the high level of overfitting, probably due to preprocessing.
More sophisticated techniques, such as GAN[17], could be used for both data augmentation and class rebalancing to obtain more consistent data.
In addition, image 'cleaning' techniques, for example techniques that can isolate and remove hairs from images[18], could be adopted to improve the performance of the algorithms.
In parallel, techniques such as active contour[19], capable of separating the part of the image concerning the lesion from the healthy body could be applied.
In addition, prediction could be enriched by implementing the use of metadata along with images.

References

[1] HAM10000
https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T

[2] ISIC Archive
https://www.isic-archive.com/#!/topWithHeader/wideContentTop/main

[3] Actinic keratoses and intraepithelial carcinoma
https://pubmed.ncbi.nlm.nih.gov/16004019/

[4] Basal cell carcinoma(1)
https://www.cancer.gov/publications/dictionaries/cancer-terms/def/basal-cell-carcinoma

[5] Basal cell carcinoma(2)
https://www.skincancer.org/international/il-carcinoma-basocellulare/

[6] Benign keratosis-like lesions:
https://my.clevelandclinic.org/health/diseases/21721-seborrheic-keratosis

[7] Dermatofibroma:
https://www.ncbi.nlm.nih.gov/books/NBK470538/

[8] Melanoma:
https://www.cancer.org.au/cancer-information/types-of-cancer/melanoma

[9] Melanocytic nevi:
sciencedirect.com/topics/medicine-and-dentistry/melanocytic-nevus

[10] Vascular lesions:
https://dermoscopedia.org/Benign_lesions_-_angioma_and_other_vascular_lesions

[11] Horizontal Flip:
https://pytorch.org/vision/main/generated/torchvision.transforms.RandomHorizontalFlip.html

[12] Vertical Flip:
https://pytorch.org/vision/stable/generated/torchvision.transforms.RandomVerticalFlip.html#torchvision.transforms.RandomVerticalFlip

[13] SMOTE:
https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.htm

[14] Paper on which is presented the Model N:
https://www.nature.com/articles/s41598-022-22644-9#Sec3

[15] Cross Validation:
https://cesa-bianchi.di.unimi.it/MSA/Notes/crossVal.pdf

[16] plateau method documentation:
https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html#torch.optim.lr_scheduler.ReduceLROnPlateau

[17] GAN:
https://www.sciencedirect.com/science/article/abs/pii/S0169260720302418

[18] Hair detection and removal:
https://pubmed.ncbi.nlm.nih.gov/36385676/

[19] Active contour application:
https://link.springer.com/chapter/10.1007/978-981-16-4538-9_13