REPORT - IMA 205

Estienne GOIGOUX
(EstienneGGX on kaggle)

Télécom Paris

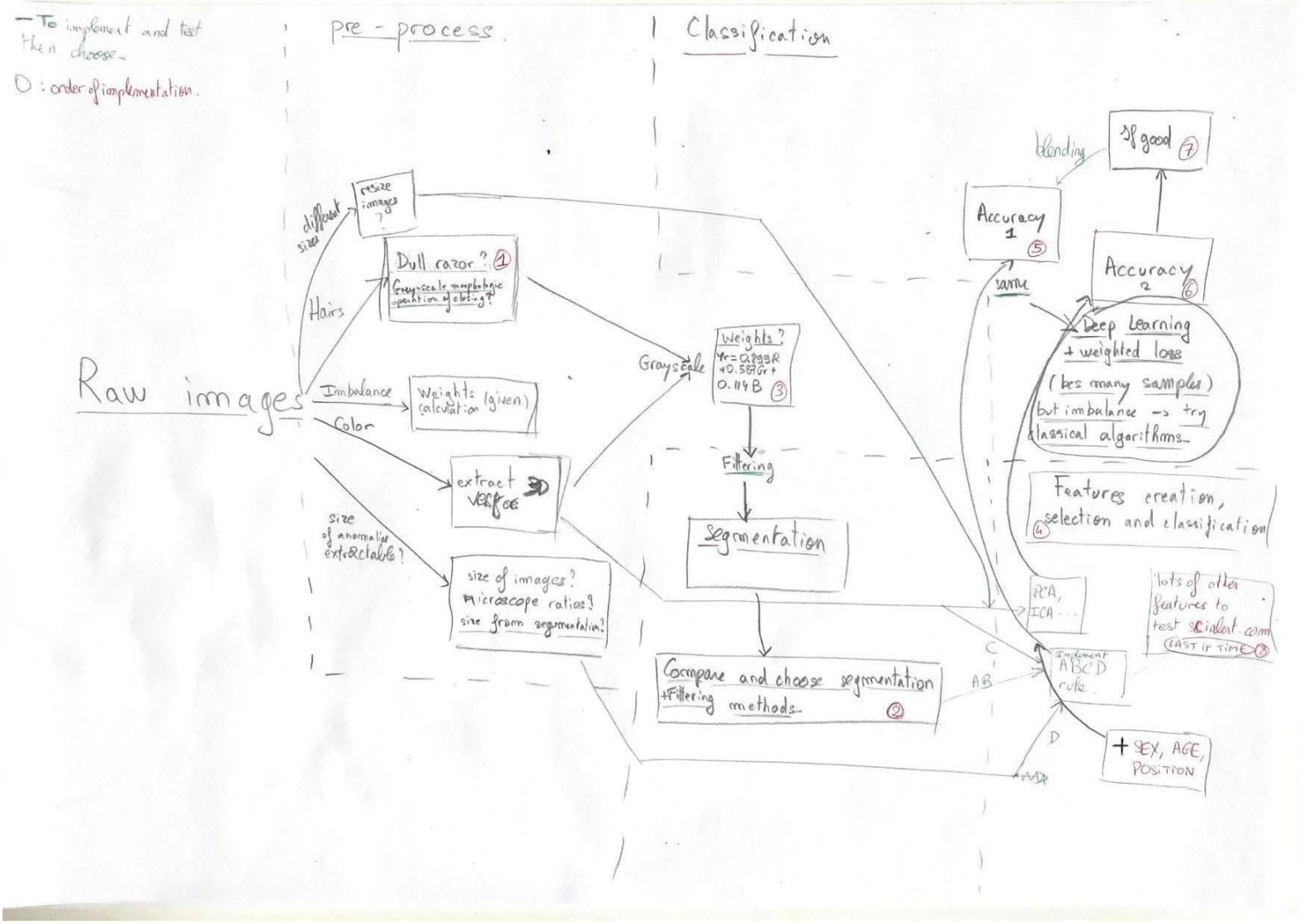
This document is a synthesis of my work for the challenge 'IMA 205 Challenge 2022, Classify dermoscopic images among 8 different diagnostic classes'. It includes an introduction, a chapter for every notebook produced which matches with one step of the logical reasoning and a conclusion. The notebooks outputs are visible when clicking on the link in case the local prediction notebooks gives an error.

SUMMARY

1 **INTRODUCTION**

The problem is a multiclass categorization problem. We have a training and a test set of images of skin anomalies, to their associated metadata and to their clinically determined segmentation for some of them. Our task is to have the best weighted categorical accuracy on the classification of the test set. We are asked to segment anomalies, to extract features from these segmentations and from the images and to use machine learning algorithm to classify them.

Here is the draft of the working schedule (not necessary for the reader to decrypt it) after the data analysis :



2 SCIENTIFIC APPROACH AND DATA ANALYSIS

(Notebook 1 : https://www.kaggle.com/code/estienneggx/1-dataset-analysis)

Scientific approach. The first step was to have a look at the data to define the global strategy, here are the main observations and consequences withdrawn from this notebook :

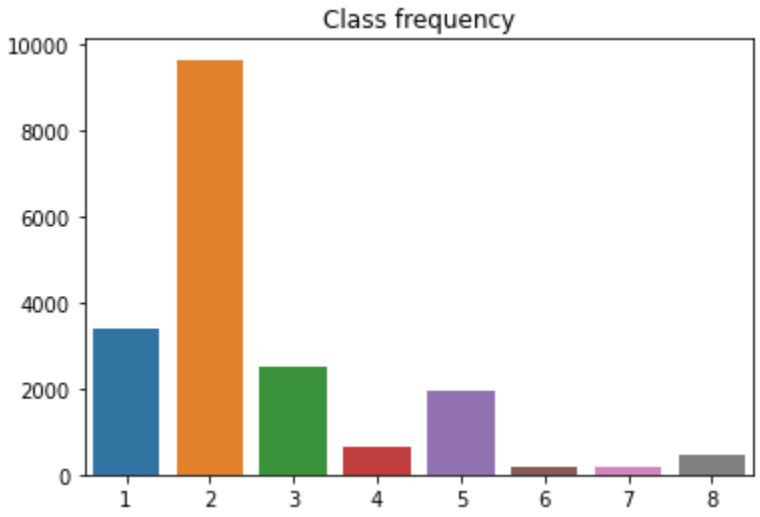


Figure 1: class frequency distribution

1. We have really few training points for class 4,6,7,8. Since the evaluation metric is weighted categorical accuracy, we will have to deal with this issue by performing data augmentation, by resampling and/or by giving weights to each data point depending on its class. We don't have that many samples so neural networks might not be the most efficient algorithms.

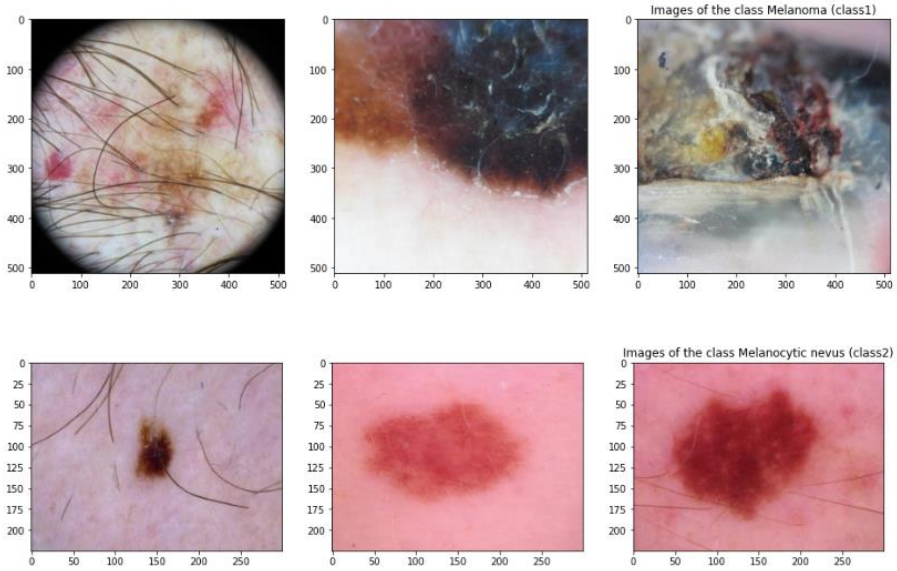


Figure 2: samples from class 1 and 2

2. Remarks on the samples:

We can see that some of the pictures were taken through a **microscope**, it could be accurate to **crop them** to have better segmentation.

There is **some hair** on some pictures, we will also have **to erase them** before performing segmentation.

The **picture's size might change** even within the same kind of anomaly, we also have to resize images before treating them if we use a linear model.

Some of the anomalies (like Vascular lesion [7]) seem to have the same look but some seem to differ a lot within the same anomaly (like Melanoma [1]).

3 DATA PRE-PROCESSING AND FEATURES EXTRACTION

(Notebook 2 : https://www.kaggle.com/code/estienneggx/2-pre-processing-and-features-extraction)

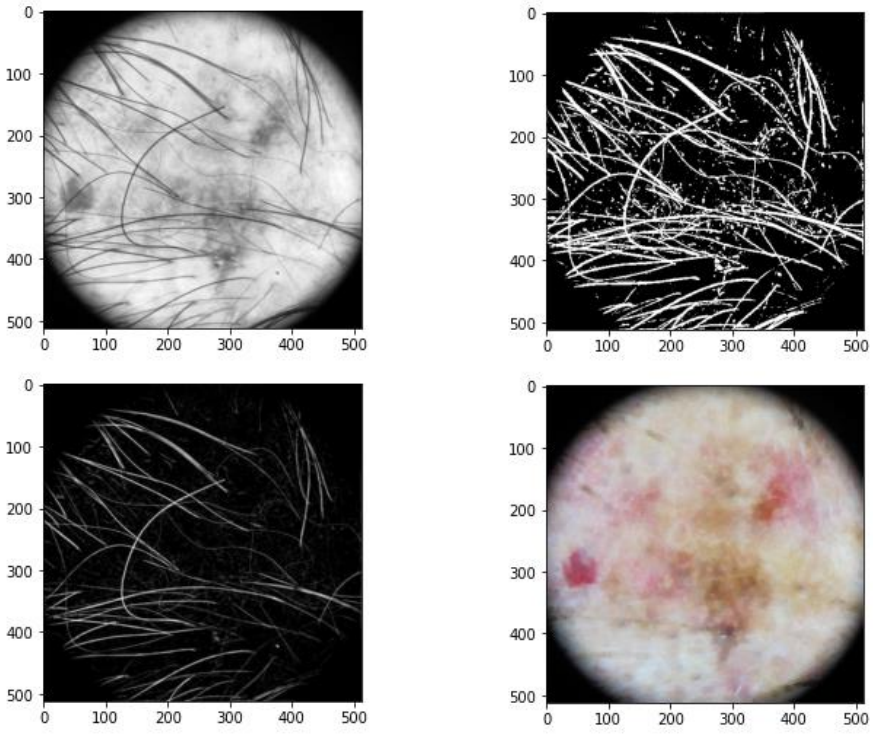
1. Hair removal :

I chose to use the already implemented dull razor technique, since it was cited in a lot of papers on melanomas as the most effective way to deal with that issue. The steps of the algorithm are :

1) Identify the dark hair locations by a generalized grayscale morphological closing operation.
2) Verify the shape of the hair pixels as thin and long structure, and replace the verified pixels by a bilinear interpolation.
3) Smooth the replaced hair pixels with an adaptive median filter.

The code used comes from this notebook :
https://www.kaggle.com/code/antocad/siim-isic-image-preprocessing-dull-razor/notebook

Steps of dull razor :



2. Filtering and segmentation methods

I implemented the adaptive thresholding for skin lesion segmentation using statistical parameters described in the following paper : https://ieeexplore.ieee.org/document/7929752. The steps of my implementation are:

1) Transfer from RGB to YIQ space and extract luminance Y.

$$Y = 0.299^{*}R + 0.587^{*}G + 0.114^{*}B$$
$$I = 0.596^{*}R - 0.274^{*}G - 0.322^{*}B$$
$$Q = 0.211^{*}R - 0.523^{*}G + 0.312^{*}B$$

Figure 3: Equations RGB to YIQ space

2) Invert the pixels of the luminance and withdraw the mean to it.
3) Denoise the image and apply an averaging filter.
4) Compute a binary image by using the standard deviation of the image divided by 2 as a threshold and extract the largest area obtained.
5) Fill the holes if there are some

The result was convincing except when pictures were taken through a microscope so I detected this case by looking at if the border were black on the first and last 15 lines and columns. I implemented an heuristic segmentation method for these cases : The picture is first cropped then converted to black and white. The distance from the mean of the new image is computed and its edges are calculated. An averaging filter is then applied and the image is inverted before thresholding.

3. Features extraction

I used the library skimage to withdraw geometrical properties from the images and withdrawn the mean, median and standard deviation of RGB channels.

4 RANDOM FOREST ON EXTRACTED GEOMETRICAL FEATURES

(Notebook 3a : https://www.kaggle.com/code/estienneggx/3a-random-forest-on-tabular-data)

1. Data cleaning

Some data were missing for age and position so I filled them with mean and most frequent. I added the metadata and I one hot encoded the categorical labels and removed the 'SEX_male' column since it was complementary with 'SEX_female'. I also split the training set into a training and a validation set.

2. Model used

I used boosted trees from xgboost to fit the data. I first had really bad weighted categorical accuracy (WCA) because I had not put weights to each sample. I first resampled the data by drawing data points from each class with the same probability. The WCA improved but was not really good either (~0.3 on validation set). I changed the loss to a weighted loss and removed the resampling and got better results (~0.4 on validation set). Model parameters : since they were really few samples of some classes, one of the challenges was to not overfit, so I took an eta a bit superior to the default one, changed the parameter 'subsample' so that every tree is trained on a subset of the training set and let the learning rate and maximum depth vary. I used grid search to find the best parameters possible and ended up with that set of parameters : eta = 0.5, subsample = 0.9, learning_rate = 10e-2, max_depth = 5 and n_estimators = 300. To get the final model, I trained a new model with those parameters on all the training set.

5 DENSE NET ON EXTRACTED GEOMETRICAL FEATURES

(Notebook : https://www.kaggle.com/code/estienneggx/dense-net-on-tabular-data)

I tried implementing a dense net on the data described in 4 (standardized) but did not know how to compute a weighted loss soon enough to come back to this notebook before the end of the challenge, so the results of this notebook were not incorporated into the final submission. I completed it afterwards and got an ok WCA on the testing set (~0.4).

6 CNN ON THE IMAGES

(Notebook : https://www.kaggle.com/code/estienneggx/dense-net-on-tabular-data)

I used the vgg16 architecture to build a convolutional neural network to classify the data from the colored images.

1. Data pre-processing

The images were converted from RGB to BGR, then each color channel is zero-centered with respect to the ImageNet dataset, without scaling. Then they were resized to (224,224) and grouped in batches of 64 images.

2. Model used

Dense layers, dropout layers, and a normalization layer were added to the vgg16 model. The last layer returned 8 probabilities using the sigmoid activation function to classify the data.

The first model was doing bad on WCA because I had not implemented the weighted loss yet. I taught that I would have poor results because we had really few data points for the minority classes but it performed really well after adding weights to the loss and after 20 epochs (approximately 0.6 WCA on validation set). The model overfit a bit just before the deadline and I had not implemented a history yet so I was not able to correct this error and the WCA validation was ~0.5 in the end.

7 BLENDIND THE MODELS

(Notebook 4 : https://www.kaggle.com/code/estienneggx/4-blending-models-and-final-predictions)

Finally, I blended the models obtained by the boosted tree model and the CNN. I had to normalize the values outputted by the CNN line by line because they weren't probabilities. Blending the two models gave a model with a better generalization capacity. Since the CNN had really better results, I gave it a weight of 0.8 and 0.2 for the tree model, which improved my overall public score to 0.52448 on the public leaderboard, ranking 12 over 40 students.

8 CONCLUSION AND WHAT I WANTED TO TRY

Preprocessing :
-The dullRazor algorithm was really efficient and I don't think I should have done anything different with it.
-The adaptive thresholding for skin lesion segmentation using statistical parameters was pretty efficient but had some faults and the other segmentation method that I implemented was pretty random. I should've computed a metric to evaluate the different segmentation methods and I would have liked to try to implement other segmentation methods, and maybe combine them.
Feature extraction : I think that was one of the biggest flaws in my tree model. Indeed, the feature importance of the geometrical features created were always pretty low compared to the ones regarding the metadata already given. I would have liked to try and implement features giving more precise information on irregularities, shapes and borders.
The training time was pretty high and I would have liked to try and optimize my algorithms.
I would have also liked to extract information from images by doing PCA, ICA or NNMF but did not have time.
Post processing : I did really few post processing, I would have liked to try to study the links between features, to do features selection and to create new ones.
Random Forest : The model was pretty classic. The implementation of weights in the loss was a big step for the algorithm. I would have liked to try to optimize the hyperparameters with methods like the Bayesian one rather than using grid search.
Neural networks and blending : The neural networks were also classic, and blending results did improve the model but was not ideal. Using transfer learning was a good way to obtain good results. What I would really wanted to implement is a fusion of the networks. I wanted to concatenate the trained networks and to retrain from the new network to combine the effects of the two predictors. Indeed, blending added the point of view of two different features set (geometrical and metadata vs images) but it did not combine the training on all those parts at once like a concatenation and retraining would have done.

To conclude, not being in the IMA program, this challenge allowed me to understand and implement many image processing algorithms and to see the main difficulties linked to the image format in data science. It also taught me how to manage imbalanced datasets and how to use transfer learning in a real case scenario.