Deep Expectation of apparent age from a single image
Cyan team challenge project

Nawel Medjkoune Ahmed Mazari NGO HO Anh Khoa Mihaela Sorostinean
Laurent Cetinsoy Herilalaina Rakotoarison Matthieu Ré

January 29, 2017

Bundle URL: https://drive.google.com/open?id=0Bwu37Uq8bDW5WEpvRUloZ0Q2Rjg
Sample submission URL: https://drive.google.com/drive/u/1/folders/0B1SN1X3DFpshQm1sRndiX2ZMWHc
Starting Kit URL: https://drive.google.com/file/d/0B82JJcrh8r0bbm9RZkk4M1BrRlU
Challenge URL: https://competitions.codalab.org/competitions/15790?secret_key=cf627d89-2ea0-435a-af88-99dc7f371cbd
Challenge URL (LRI): https://codalab.lri.fr/competitions/11
Github URL: https://github.com/lcetinsoy/ComputerVisionChallenge/
Video URL: https://www.youtube.com/watch?v=HblziKzeRng&feature=youtu.be
Preprocessed data URL: https://drive.google.com/drive/u/1/folders/0B1SN1X3DFpshdzl2OTI3a190UWc
Raw data URL: https://drive.google.com/drive/u/1/folders/0B1SN1X3DFpshYkVJTzBCcl9BRVk

Abstract

The purpose of our project was to organize a challenge which uses features extracted from images with human faces, in order to classify the person's age group in the picture as adult or minor. The WIKI database was used, as well as the CNN provided along with it, in order to extract the features. Participants of our challenge are provided with train and validation datasets and they are required to find the best approach in order to classify an image in one of the two categories. The adult/minor classification has important applications in various domains. In this report, we describe in more details the set up of the challenge and present some approaches for tackling the problem.

1 Introduction and Background

This challenge addresses the issue of estimating the apparent age in face images. Age estimation is defined by using a machine learning algorithm to predict the specific age of a person, given his/her face image. The age interval is usually one year, but sometimes, this estimation is reduced to just knowing the group age of a person: baby, young, adult, senior. As humans, we are able to tell a person's age group from his/her face recent picture, and this estimation can be most often correct. This task is not that simple using computers, for that reason it has been investigated in computer vision for decades [1]. What makes it that challenging is the fact that different people age differently, as the process of aging is not only determined by genes but also with external factors as health, environment, living style...etc [2].

Age estimation of a person from facial images is a really important task as it has many applications in real life. One of the main applications that we can think of is control of resource access (website, drug purchase, violent movie, etc.) based on the age of a person. Indeed a lot of violent content is accessible on the internet and 45% of children under 12 are not monitored by parental control. Other applications for automatic age estimating may include human-machine interaction for example in the purpose of advertising content based on the user's age; automatic scanning of videos looking for persons in a certain age group for security or law enforcement purposes and automatic detection of children in unattended places.

As said before, the task of automatic age estimation is very important and difficult from the machine learning and computer vision point of view. In the shorter term, improvements in this task lead to an improvement of our understanding of how human vision works to get precise estimation of a person's age. Gaining this precision may have numerous benefits especially with the improvements of robots development. In the long ride, finding a way of analyzing efficiently aging characteristics as features such as skin wrinkles will assist in extending their use in facial expression recognition.

This challenge was inspired by the work of Rothe *et al.* [6], in which a method for age detection from images is proposed. The method consists in locating the face in the image and then use a CNN model to estimate the age of the person. A demonstration of the application is available at http://howhot.io/. The authors of this work report better results than the current state-of-the-art. However, there are other notable papers concerning the problem of age estimation form the perspective of computer vision. In 1999, Kwon *et al.* [1] reports the first work involving age classification from images. In their work, they use cranio-facial development theory and skin wrinkle analysis to classify human faces into three categories: baby, young adult and senior adult. They reported 100% classification accuracy, but they used a really small dataset, only 47 images. In a more recent work, Guo *et al.* [3] developed an age estimation framework based on manifold analysis of face pictures. A different approach, proposed by Geng *et al.* [4] is based on associating a label distribution for each face image, while an algorithm for learning this label

distributions is developed. The algorithm uses an iterative optimization process based on the maxi-mum entropy model.

Although previous challenges have been organized with the purpose of age estimation in images, the novelty of our challenge is that we don't propose a regression problem which aims to predict the actual age of a person. Instead, in this challenge we propose to solve a binary classification problem. The goal is, as it was mentioned before, the classification of a person into minor/major based on their picture. Therefore, we expect better accuracy.

In this challenge, we provide participants with a large dataset of facial images of people from two age groups: young and adults. The goal is to predict the age group of a given new person based on its picture. The participants need to exploit the pre-processed features we made available in order to make predictions as accurate as possible. In the following section, we describe in more details the set up of this challenge, starting from describing the dataset, the preprocessing and feature extraction made on the raw data, the evaluation metric we use in the scoring board and finally we present, analyze and compare different approaches to solve the problem.

2 Materials and methods

2.1 Dataset

The dataset which was used for this challenge is the one built and employed in the work of Rothe *et al.* [6]. We used the WIKI database [11], which contains 62.328 images of celebrities extracted from Wikipedia, with gender and age labels. An example of images from the dataset can is shown in figure 1. Figure 2 shows the age distribution for this dataset. However, several images contain artifacts and some others do not contain any face, so these images had to be removed. Moreover, we also noticed that the number of children (under 10) is really low. Nevertheless we can set the age limit between classes to 30. Figure 3 shows the number of samples per class for two different age threshold, 18 and 30, on the dataset. We can observe that classes are more balanced with an age limit of 30. In the case of a classification task where the age threshold A is 18 the classes are skewed favorably towards adult people.

Figure 1: Examples of images from the Wiki dataset

2.2 Preprocessing and feature extraction

One of the main tasks before predictive analysis is to prepare data and extract features. For images, these tasks are very complex and time / CPU-resource consuming. In order to ease undergraduate, the dataset was preprocessed by extracting features from raw images with a convolutionnal neural networks (CNN).

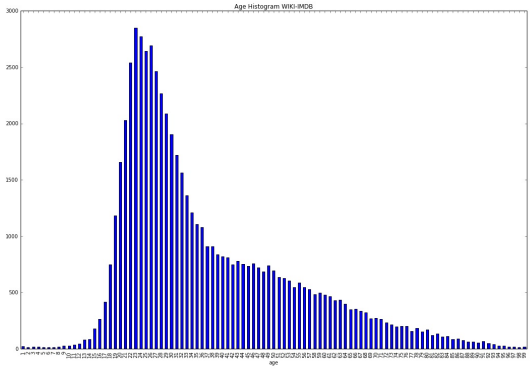

Figure 2: Age distribution in the WIKI dataset samples

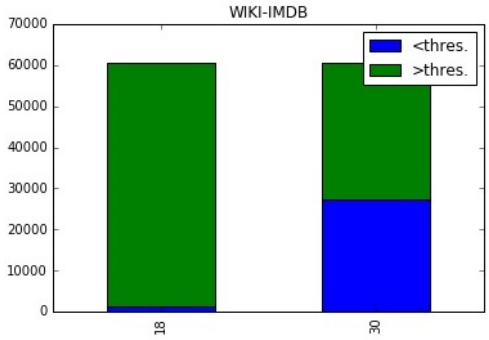

Figure 3: Comparison of classes balancing for WIKI dataset for a threshold of 18 and 30 year-old

Rothe et al. summarize the preprocessing and feature extraction task using the following pipeline.

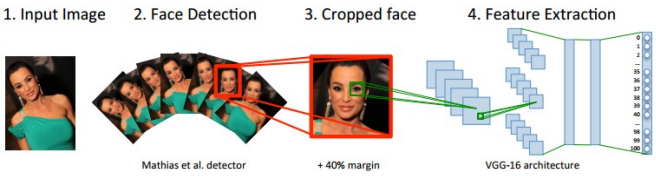

Figure 4: Preprocessing and feature extraction pipeline

Faces are first detected, using the off-the-shell face detector of Mathias et al [12] , on the original image and other rotated versions which are:

- between -60° and 60° in 5° step
- -90°, 90° and 180°

Detection score, which indicates the presence of face in image, is then used to remove images without faces from the dataset. As the proposed method only detects face without bells and whistles, we extent face sizes to 40% of its width to the left and right and 40% of its height above and below. Rothe et al. argue that this extension helps the prediction accuracy. The resulting images are finally scaled to 224 x 224 pixels.

Then, images features were extracted with a CNN model which was pretrained on large visual database called ImageNet [20]. For the task of image classification, CNN

are known as the state-of-the art technique. Here is a figure of the network used, the VGG16 architecture [8, 9]. It is a very deep network with a lot of convolution layer followed by max-pooling. The image descriptor are taken from the The model 15^{th} layer containing 4096 neurones. Each image fed to the network will be transformed in a 4096 dimension vector. The code for data preprocessing and feature extraction is available in the github repository: https://github.com/lcetinsoy/ComputerVisionChallenge/

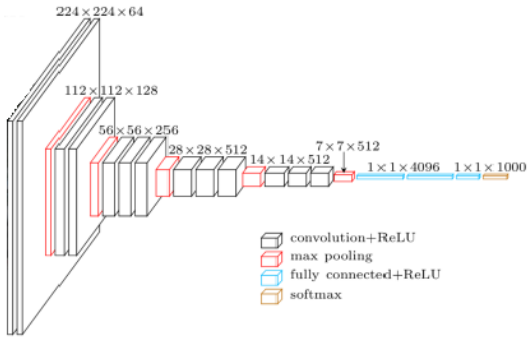

Figure 5: VGG-16 architecture

In order to point out to the most important features resulting from the CNN feature extraction, we ran an Extra Trees Classifier [21] on the train set. In Figure 6 we plotted the projection of data using pair plots of the 5 most important features returned with the extra trees classifier.

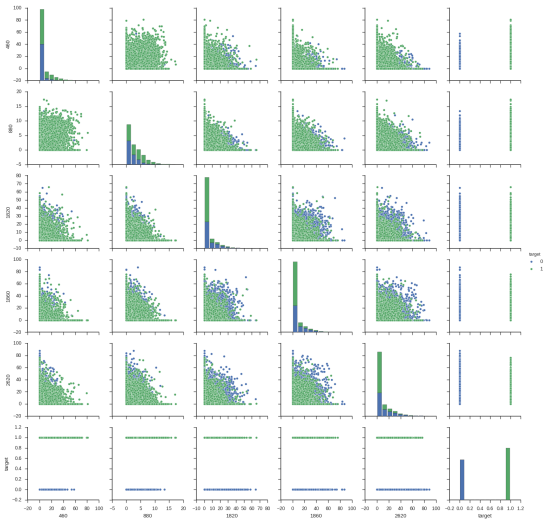

Figure 6: Most important features

In order to explore further our features, PCA was ran on the training data-set over various number of components. Figure 7 represents the ratio of explained variance over the number of components selected with PCA. We observe that the curve goes up exponentially for the first 200 components (and reaches 70% variance) and then it tends to follow an asymptotic behavior until 1000 components with only 20% improvement. This can suggest that the first 200 components explains our data to a large extent and the rest can be only added noise.

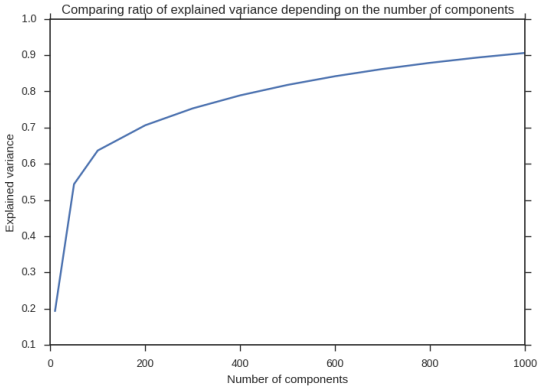

Figure 7: Explained variance over number of components

3 Evaluation metrics

The main objective of the challenge is to make good predictions of the target variable. To do so, A classifier that can distinguish between different classes should be trained. For a given dataset, various decision boundaries can be learned as shown in figure 9.
To evaluate them, predictions made by the classifier on the classifier are compared to the true labels. These results may be presented summarized by a confusion matrix [18]. It reports the number of right predictions (true positive and true negative ones) and wrong predictions (false negative and false positive ones).

		Prediction	
		Class +1	Class -1
Truth	Class +1	tp	fn
	Class -1	fp	tn

Figure 8: Confusion matrix [18]

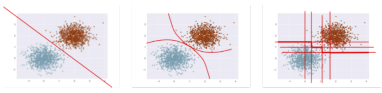

Figure 9: A decision boundary from various classifiers

Two more indicators,sensitivity and specificity, are also common and defined as:
$Sensitivity = tp/pos$
$Specificity = tn/neg$
with tp being the true positives, tn, the true negatives, $pos = tp + fn$ is the total number of positive examples and $neg = tn + fp$ the total number of negative examples.

When classes are imbalanced (i.e. when one label frequency is way higher than the other one like in figure 10).the classical error metric is not the appropriate one. For instance if a class was 99% and 1%, a model predicting 1 all the time would make 1% error only. In order to circumvent the issue of unbalanced (binary) classes the BAC binary metric [19] was introduced. BAC metric is normalized to cope with unbalanced classes such that the expected value of the score for a "trivial guess" based on class prior probabilities is 0 and the optimal

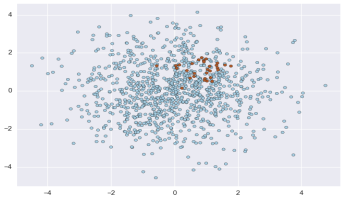

Figure 10: In this example, the red class is way less frequent than the blue one.The points show examples distribution in two dimensions. The axes (X,Y) represent the coordinates of the examples and the blue, red colors represents the two classes.

score is 1.It is defined as follow :

$$BAC = (TP/P + TN/N)/2 \quad (1)$$

such that :
TP: Number of True Positive examples.
TN: Number of True Negative examples.
P: Number of Positive class.
N: Number of Negative class.
The behavior of BAC metric is illustrated in 11.

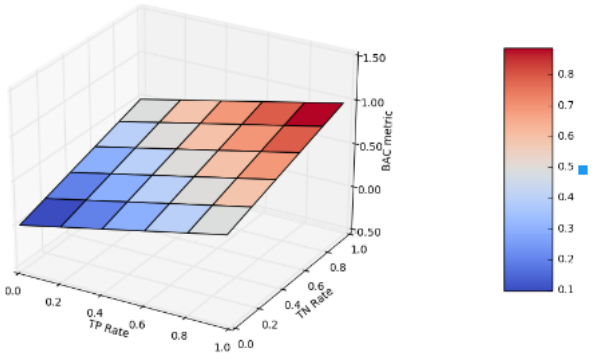

Figure 11: BAC metric. It's expressed in term of true positive TP and true negative TN. When the true positive (resp negative) rate approaches 1.0, it means that most of positive examples (resp negative) are affected to the appropriate classes (resp positive negative). The performance of random classifier represents the center of the diagonal where only 50% of positive (resp negative) examples are well affected.

Other techniques can be considered in order to balance data-set classes. One can think about over-sampling the minor class or under-sampling the major class as it's depicted in the figure 12.

4 Results and discussion

Although this challenge was inspired by the work of Rothe *et al.* [6], their results cannot be used as a baseline because the type of their problem is different. While in the previously mentioned work, a regression problem was proposed in order to estimate the apparent age of a person, we simplified the problem and transformed it into a binary classification.

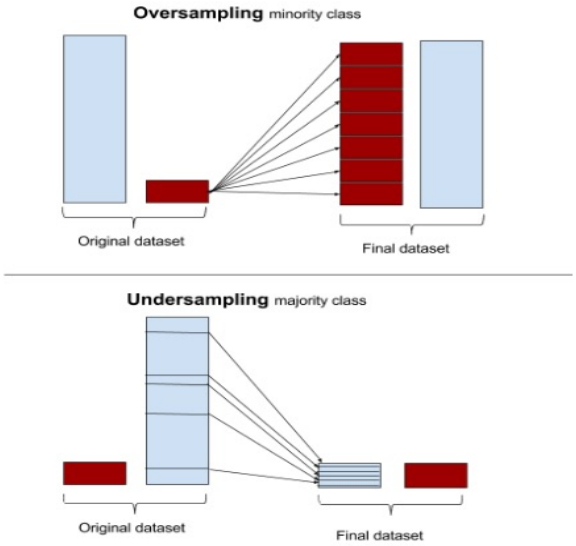

Figure 12: over-sampling and under-sampling classes.

This section discusses the results obtained while training various classifiers. It also discuss aspects of the preprocessing and feature extraction phase.

4.1 Training on raw data

To evaluate the importance of feature extraction, we trained and tested several classifiers using the raw data. Figure 13 shows the ROC curves obtained for: Random Trees Embedding(RT) with Logistic Regression(LR), Random Forest(RF), Random Forest with Logistic Regression(RF + LR), Gradient Boosting Trees (GBT) and Gradient Boosting Trees with Logistic Regression (GBT + LR). As it can be seen in the plot, the ROC curves for all classifiers are relatively flat and almost approaching the random classification. Therefore, using CNN features rather than raw data has an important effect on improving classification accuracy.

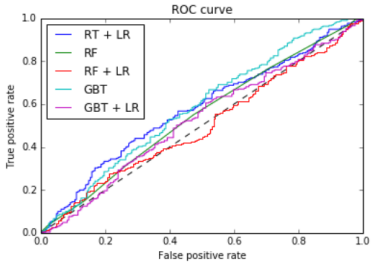

Figure 13: ROC curve for raw data models

4.2 PCA components vs CNN features

In order to see what the PCA decomposition brings to our accuracy rates, we compare results of training our classifier using the train set with the CNN features and the train set transformed after PCA.

Figure 14 shows clearly that the 200 components of PCA taken to transform the data are not enough to explain the

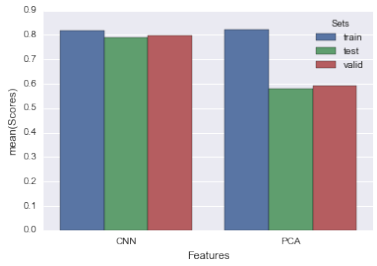

Figure 14: Comparing classification performance using PCA transformed data and CNN features

dataset. There is a clear overfitting in the training step when using PCA components for training. For the following experiments, we will use the original CNN extracted features instead.

4.3 Comparing classifiers

We provide in the following three baseline approaches to solve this challenge. Our first idea is to train our data with the three classifiers and choose the best for parameter tuning. Figure 15 shows the scores obtained for the three classification methods which we used: Random Forest, Decision Tree and Gradient Boosting. As it can be observed in the previously mentioned graph, the accuracy of training set is always higher than the accuracies of validation and testing set. The scores of Gradient Boosting Classifier is slightly higher than Random Forest and Decision Tree, which means that Gradient Boosting is a promising algorithm. Besides, there is a small overfitting when using the Decision Trees and Random Forest compared to Gradient Boosting. After fine tuning on the two first, the overfitting was still clearly present. For this reason, we have chosen the Gradient Boosting as the best classifier.

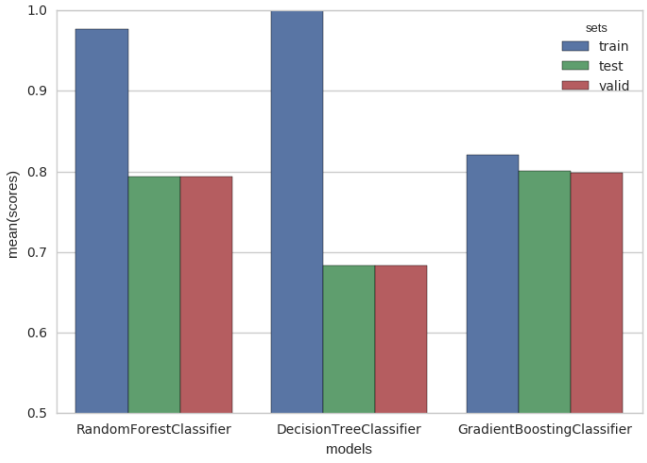

Figure 15: Models proposed in training process

4.4 Feature selection

To recall, the number of features extracted using the VGG-16 is 4096. A lot of features have negligible values and may be removed. For example, figure 16 shows the large differences between these features. As a result, we select the features which separate better the two classes by using SelectKBest function in sklearn [22] using the chi-squared stats. This chi score can be used to select the features with the highest values for the chi-squared statistical test [23] and shows the relation between each non-negative feature and class. From figure 17, one can observe that the more number of features are used, the less the prediction is. The two patterns of test and validation sets have decline trends from about 0.740 to about 0.727 and 0.715 respectively. The training set results were not plotted as they gave 100% accuracy. So consequently, a lower number of features can be used while preserving the classification accuracy. In this case, the best number of features found was 50.

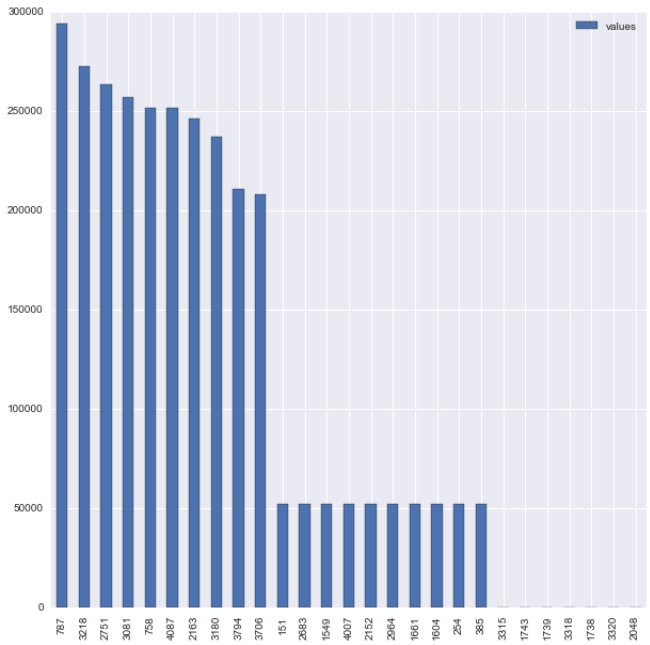

Figure 16: Sum of values taken by features

Figure 17: Accuracy Over Number of Features

4.5 Influence of sample size

Another aspect, shown in figure 18, is the influence of the number of samples over the accuracy of the model. More precisely, the accuracy increases from about 0.67 (500 samples) to about 0.72 (all samples).

Figure 18: Accuracy Over number of Samples

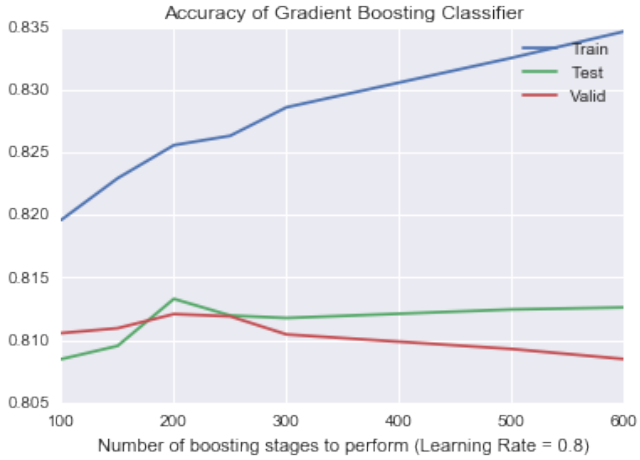

Figure 20: Accuracy Over Number Of Boosting Stage

4.6 Parameter tuning for Gradient Boosting

After choosing the best classifier and reducing the number of features, the hyper-parameters of the Gradient Boosting were tuned.

The plots in figure 19 and 20 show the evolution of accuracy depending on the modified parameters for achieving better scores. Before achieving this, we first scale our data in the interval [0,1].

Firstly, the learning rate was tuned by selecting values in the [0.01, 1]. Figure 19 shows the results. The curve follows an exponential shape and becomes flatter after a learning rate of 0.5. The best accuracy reached for the train and validation sets is about 0.819 and 0.809 respectively with learning rate 0.8.

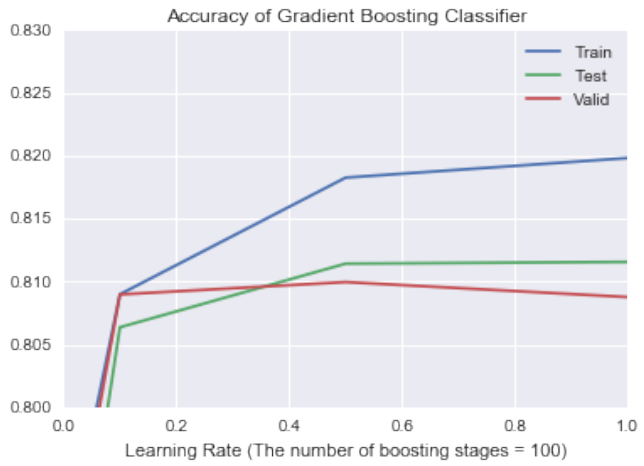

Figure 19: Accuracy Over Learning Rate

The second that was tuned was the number of boosting stages. Figure 20 shows results for the accuracy on train, validation and test sets over multiple values of boosting iterations. This parameter is also important which improves the accuracy of validating set to about 0.812 at the value 200.

Error bar Finally, to ensure the quality of our decision function we have computed the standard deviation of our model applied on different combination of the testing set and according to the BAC metric. This value ensures that our best Gradient Boosting classifier has a 0.81 ± 0.0038 balanced accuracy.

5 Conclusion

The purpose of our project was to propose, analyze, create and test a data challenge. We chose a problem of binary classification, which aims to split people into two categories: minor or major, based on features extracted from the image of their face. We used a pre-trained CNN to extract the features from the WIKI database. Thereafter, these features are split into train, validation and test sets and their are made available for the participants of our challenge. We used several models to solve the classification problem and discussed the results obtained for each one of them. Among all, the Gradient Boosting Classifier obtained the best results.

References

[1] Kwon, Young H., Niels da Vitoria Lobo: Age classification from facial images, Computer Vision and Image Understanding 74.1 (1999): 1-21

[2] Guo, Guodong, et al. "Image-based human age estimation by manifold learning and locally adjusted robust regression." IEEE Transactions on Image Processing 17.7 (2008): 1178-1188.

[3] Guo, Guodong, et al.: Image-based human age estimation by manifold learning and locally adjusted robust regression, IEEE Transactions on Image Processing 17.7 (2008): 1178-1188.

[4] Geng, Xin, Chao Yin, and Zhi-Hua Zhou: Facial age estimation by learning from label distributions, IEEE transactions on pattern analysis and machine intelligence 35.10 (2013): 2401-2412.

[5] G.Mahalingam,C.Kambhamettu.Face Verification with Aging Using AdaBoost and Local Binary Patterns.

[6] R. Rothe, R Timofte, L. Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks

[7] M. G. Rhodes, 'Age Estimation of Faces: A Review'(2009), Appl. Cognitive Psychology. 23, pp 1–12.

[8] O. M. Parkhi, A. Vedaldi, A. Zisserman: Deep Face Recognition, British Machine Vision Conference, 2015

[9] VGG Face Descriptor, http://www.robots.ox.ac.uk/7Evgg/software/vgg_face/

[10] 2015 Looking at People CVPR Challenge http://gesture.chalearn.org/2015-looking-at-people-iccv-challenge

[11] IMDB-Wiki – 500k+ face images with age and gender labels https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/

[12] M. Mathias, R. Benenson, M. Pedersoli, L. V. Gool: Face detection with bulls and whistles.

[13] A. Rizzi, C. Gatta, D. Marini: A new algorithm for unsupervised global and local color correction

[14] Box Drawings for Learning with Imbalanced Data. Siong Thye Goh and Cynthia Rudin. KDD-2014, August 24–27, 2014, New York, NY, USA.

[15] Class Imbalance, Redux. Wallace, Small, Brodley and Trikalinos. IEEE Conf on Data Mining. 2011.

[16] Machine learning with imbalanced data sets. https://www.youtube.com/watch?v=X9MZtvvQDR4

[17] Learning from unbalanced classes. http://www.kdnuggets.com/2016/08/learning-from-imbalanced-classes.html/2

[18] Causality challenge. http://www.causality.inf.ethz.ch/challenge.php?page=evaluation

[19] Brodersen, Kay Henning, et al. "The balanced accuracy and its posterior distribution." Pattern recognition (ICPR), 2010 20th international conference on. IEEE, 2010.

[20] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

[21] Geurts, Pierre, Damien Ernst, and Louis Wehenkel. "Extremely randomized trees." Machine learning 63.1 (2006): 3-42.

[22] Sklearn SelectKBest function API http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

[23] Chi-squared test https://en.wikipedia.org/wiki/Chi-squared_test