Rapport de Projet de Machine learning avancé - M2 ISI 2020/2021

Groupe 3, sujet 2 : Estimating a person's age from the image of their face

XIA Sylvain - ZHANG Yuesheng - ZHANG Longyu - RAPHAEL Ronan

XIA Sylvain SYLVAIN.XIA@ETU.SORBONNE-UNIVERSITE.FR
Master Ingénierie des Systèmes Intelligents
Sorbonne Université
Paris, France

ZHANG Yuesheng YUESHENG.ZHANG@ETU.SORBONNE-UNIVERSITE.FR
Master Ingénierie des Systèmes Intelligents
Sorbonne Université
Paris, France

ZHANG Longyu LONGYU.ZHANG@ETU.SORBONNE-UNIVERSITE.FR
Master Ingénierie des Systèmes Intelligents
Sorbonne Université
Paris, France

RAPHAEL Ronan RONAN.RAPHAEL@ETU.SORBONNE-UNIVERSITE.FR
Master Ingénierie des Systèmes Intelligents
Sorbonne Université
Paris, France

Editor: Machine Learning Avancé (2020-2021)

Abstract

This project aims to estimate a person's age from an image of their face. For this we have at our disposal a large database and labeled item from the following github (1) which is linked to an article. The database provides centered and not centered images. To estimate the age from the not centered images we first needed to extract the face and remove useless information in the image, this task is fulfilled using a cascade of Haar-classifier. This part is then followed by our prediction model trained on (1). We selected two existing architectures and fine tuned them in order to achieve our classification problem, we took inspiration from VGG16 and the work in (3). We successfully extracted the face from the non centered images but we did not obtained good results with our custom model. We will first present some existing works on age estimations that inspired our approach, then we are going to justify the chosen methods and finally show a detailed experimental setup and some results.

Keywords: Age estimation, Haar cascades, VGG16, CNN.

1. Introduction

Age estimation from images is a very common task in machine learning because of it's wide possible applications. However it is not a trivial task because of the high variability of the face thus becoming difficult for networks to extract good age estimation features. Even for humans, the task is not that easy. Furthermore for inferring someone's age we often use other information rather than only face such as the voice, behavioral and also the context. Moreover, some people can also appear "younger" or "older" than their ages.

To accomplish the age estimation task we used architectures based on CNN. However, in order to make our network learn optimally we needed to provide him images that contain as few useless information as possible such as the background and surrounding faces. We needed robust method to process each images and detect the right face if there is many of them and try to have a bounding box around this face. As for the age estimation network we had to deal with a big database which created many computational problems such as memory issues in our computers. We found that the database contains under-represented classes compared to others and this leads to a decrease in prediction performance in our model. One task that could be done is to create a normalized database, i.e. one that contains as much data in each class. We also had to find solutions processing our database to make our network learn without over fitting.

2. Related work

A first idea was to use a single YOLO network as described in (6), with the right anchor box as an output we could build a real time face detection and age estimator. The anchor box could tell us if there is a face, the size of the detected face and the estimated age for the detected face. However the label needed for YOLO were very different from the one we had. We wanted to focus on building the architecture rather than remaking the labels. Thus (6) gave us crucial information about features detection but we decided to take focus on distinct blocs for face detection in image processing and age estimation.

The real time aspect of YOLO was really interesting to us so we decided for the image processing part to find a method which would allow very fast computational process. (7) propose a real time face detection system using adaboost algorithm with Haar-like features. Using integral image, Haar-like features can be computed quickly. We took inspiration of this idea for our project.

(8) propose an architecture which allows multi-task learning. The paper explores four type of existing architecture and they find out that a VGG network which is commonly used for image classification could also bring good results for age estimation. (9) also shows that CNN networks can be used to accomplish age estimation. Over fitting problems could be solved with pre-trained CNN on a large database for face recognition. Thus we decided to take inspiration of a VGG16 model as it has proves of good performances.

(3) propose an architecture which also allows multi-task learning. For this work, the author uses utk database with a rather small number of exemples.The work also shows good results in age estimation which is done with regression. The network used is significantly simpler than VGG16 which is very suited for our computers that were not powerfull.

We choose to try out two architectures and see which one of them could give us better results.

Our approach is inspired by the algorithm described in (3) and (4) for the age estimation part and (2) for the face detection task.

3. Proposition

To extract the face from the non centered images we decided to use Haar cascade classifier because of its computational speed. Haar cascade are based on Haar features as shown below in Figure 1.

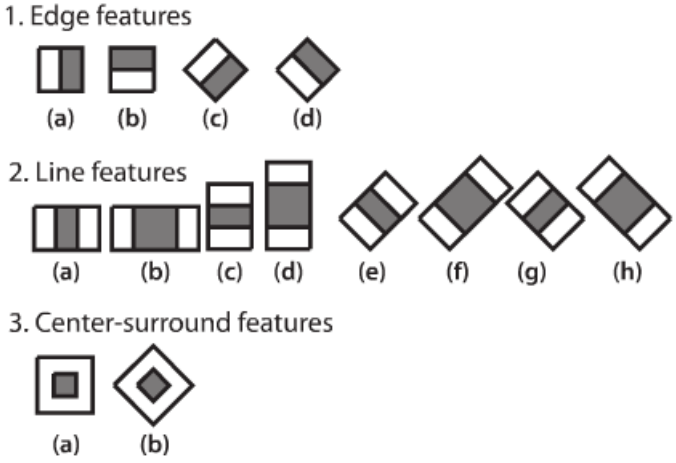

Figure 1: Haar features

For one image, each features will respond a value on a given sub region of the image, for one classifier we will use different numbers of Haar features. We then assemble each classifier to have a cascade, the result of the cascade which tells us if there is a face in the image will be the total response of each classifier as presented in Figure 2. A pre-trained Haar cascade classifier already exist in openCV and can easily detect faces from a given image. This method allows us to choose the size of the sub region we want to analyze. Database have in general a known scale for every image. We process the detection we found with the pre-trained cascade as it can still commits some error, and choose the actual face we are interested in. At the end we have an image centered on the face which facilitate the relevant features extraction for age estimation.

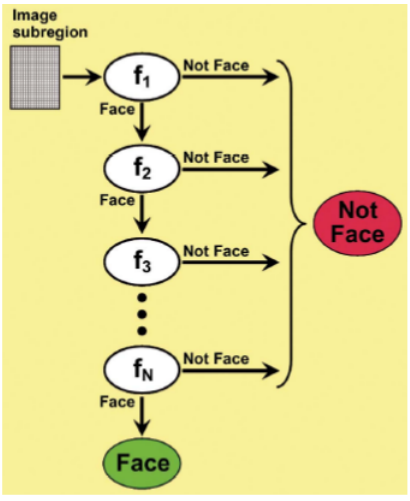

Figure 2: Haar cascade classifier

For the age estimation network we chose two models based on CNN and compare there results. First one is based on VGG16 as described in (8) and (9) as it was shown to bring good estimations

and also provide solution for over fitting. Moreover, the dimension of the used image were identical as ours. The original network is shown in Figure 3.

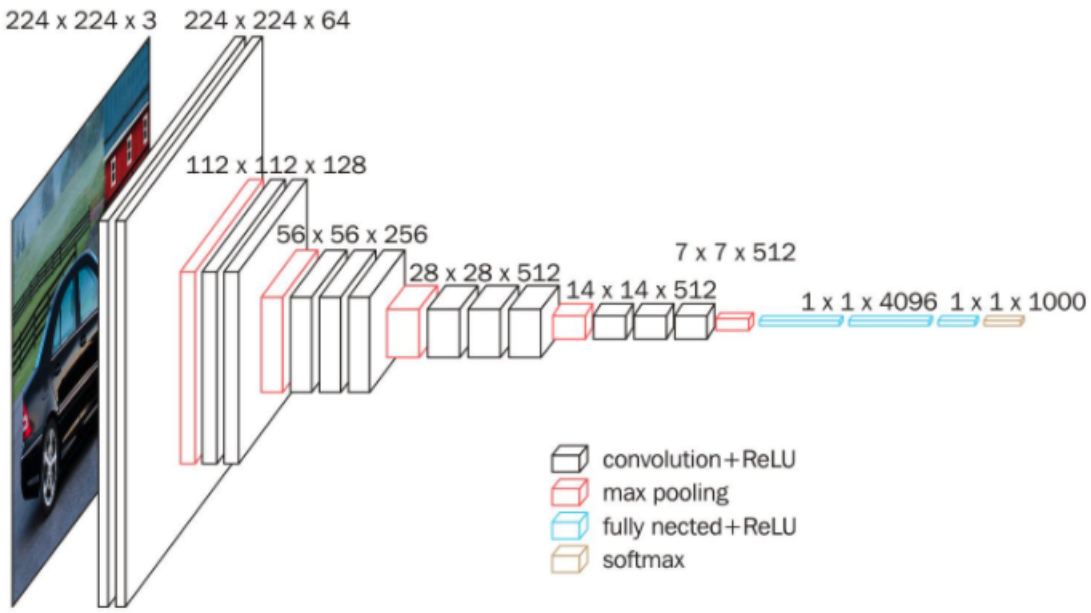

Figure 3: VGG16

In order to achieve the estimation task we changed the head bone of the network and replaced it with a softmax with the number of classes we had.

Second one is based on (3) because we had some memory issues with our computers and this work proposed good prediction with a rather small database and simpler model which makes the training process way faster. The base model is shown in figure 4.

Figure 4: Base model used by (3)

As for this model the only part we are interested in is the age prediction output. In this work, age estimation was done doing regression but our labels were made for classification. As the task is the same, we think that this network can extract relevant features for age estimation and we decided to take inspiration of it and fine tune it.

4. Experiment

4.1 Data

We had two type of data provided, one with centered images and another with non centered images, an exemple si given in Figure 5. Each set were divided in training (86 744 images) and validation data (10 954 images).

Figure 5: 20^{th} image of train set, (left : not centered image, right : centered image)

Figure 6: Exemples of not centered images in train

Figure 7: Exemples of centered images in val

As we can see from figures above, the dataset contains various type of images. The images can contain multiple person, be rotated, blurry in shades of gray, taken from front or from profile.

We had 9 different classes with age (years old) range as follow : 0-2, 3-9, 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, more than 70. We labeled the classes from increasing age range giving us the classes distributions as shown in Figure 8 below.

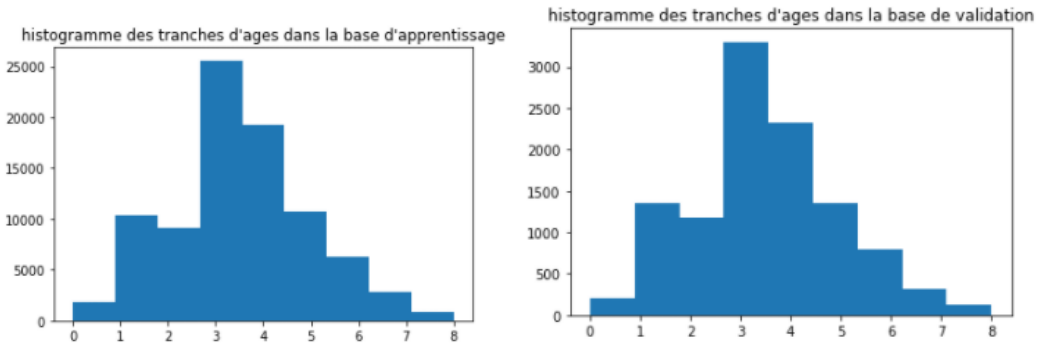

Figure 8: Classes distribution for training set (left) and validation set (right)

The classes distribution is pretty much identical for both training and validation data. However we can notice that we have many examples for classes 3 and 4 which corresponds to age ranges 29-29 and 30-39 but others classes are underrepresented. The classes distribution is uneven.

4.2 Experimental setup

To extract the face we used a pre trained Haar cascade classifier from openCV, an example of how to use it is shown (2). We partially used the code in order to comprehend how to use the functions from openCV only the detection part is being reused in our project. The detector gives a list of bounding box, each bounding box correspond to one detected face. In some cases we had either no detection at all or many faces detected at the same time depending on the parameters that we had to tune ourselves. Another part of the work was to eliminate all the false detection in case many faces were detected. After analyzing our database we thought that in principle the face of interest was the one for which the given bounding box occupies the biggest surface. We then had to crop the non centered image around the detection in order to have the centered image with a specific dimension. In case of non detection at all we chose to crop the original image from its center. To evaluate this part we only watched the results on some images containing blur, profile faces, many people in the image. for future work we could create a metric in order to see the difference between the computed pictures and the provided centered one and its accuracy.

We encountered memory problems while loading our data, thus to learn the model we used the validation data set as training data which only has 10 954 images and used validation split while fitting the model. Its accuracy was then tested on some images from the training set. No examples from the training set were injected to learn the model. During learning we used the accuracy metrics and displayed the loss in order to monitor our training session.

For the VGG network we first used the base network as shown above by only changing the head bone in order to predict 9 classes. We saw that the model wasn't learning at all and deduced that we had a vanishing problem. Thus we added BatchNormalization layers after each conv2D and saw that the model was indeed learning but also over fitting. We thought that using Dropout would help but it just kept over fitting.

On the other hand for the network inspired from (3) we also encountered over fitting issues.

4.3 Results

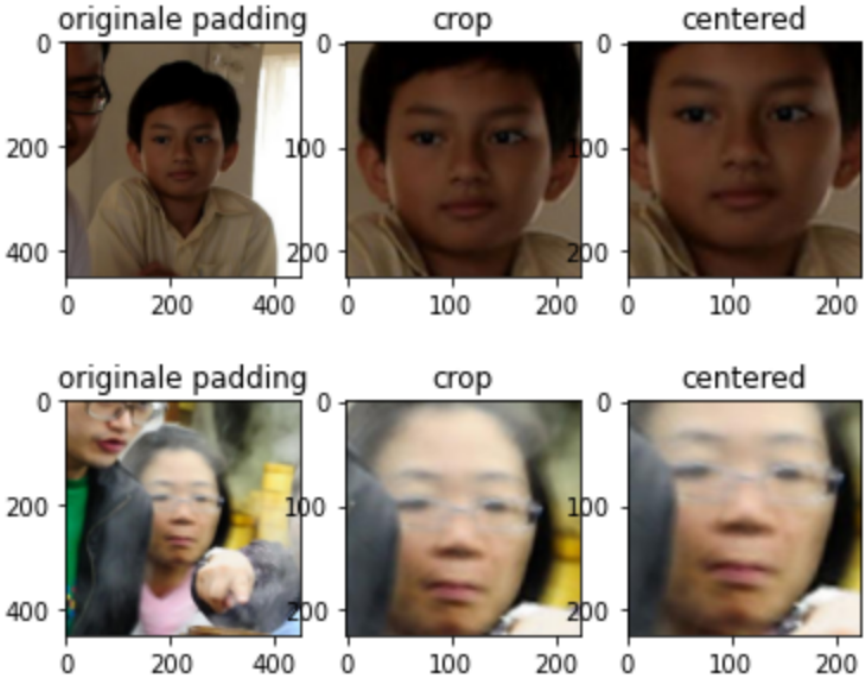

Figure 9: Left : image from original padding data set, Center : centered image we made, Right : centered image from data set

Figure 9 shows 2 examples for our face extraction and centering algorithm, we can see that the results are not exactly the same as the centered images provided in the data set however we do have an image centered on the face of interest. Our algorithm was pretty simple as we could have centered the image from the nose or other information, we decided to make it simpler and only used the whole face which gives us pretty good results even if it could be more precise.

Table 1: summary of our training results

base network	Dropout	BatchNormalization	EarlyStopping	train accuracy (%)	val accuracy (%)
VGG16	NO	NO	NO	30	30
VGG16	NO	YES	YES	94	36
VGG16	YES	YES	YES	64	36
(3) based	YES	YES	YES	67	36

The results were obtained with Adam optimizers with a learning rate of 1e-3 for VGG16 based network and 1e-4 for the other one. We used an EarlyStopping callback on the validation loss in order to stop the learning process if it is over fitting.

As shown in Table 1 we can see the effect of BatchNormalization layer here it significantly allowed our networks to learn. However we still faced over fitting problem, in fact we can see that the training rate does increase but it is significantly higher than the validation accuracy.

From Figure 10 we notice that for the first 10 epochs the network is learning well, however the validation accuracy is not improving after that and the loss is increasing. Both accuracy curves and loss curves are intertwined for the 10 first epochs but it starts over fitting after that.

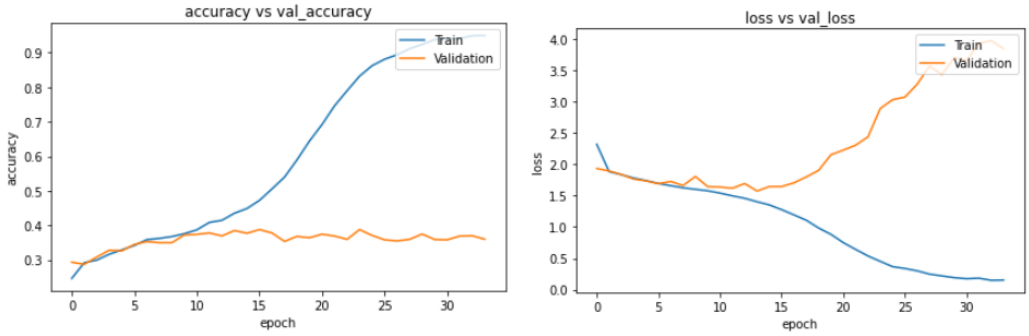

Figure 10: Accuracy and loss for VGG16 based network BatchNormalization added

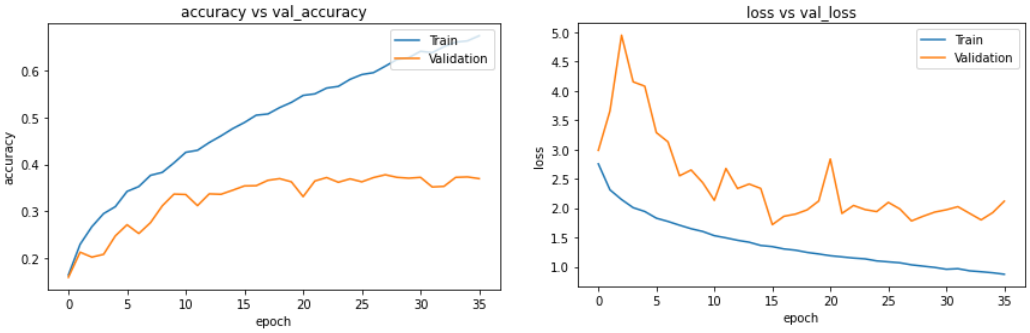

Figure 11: Accuracy and loss for (3) based network

For the other network the results shown Figure 11 also point an over fitting issue. Even if the validation loss is overall decreasing, the model is still over fitting since the validation accuracy is not increasing.

For each training session we have tried, the validation accuracy did not went higher than 40%.

We tried our network on few images of the train set (which did not serve to learn) and it gave us around 20% accuracy.

4.4 Discussion

For the face extraction part we can see that it is over all doing well, however the image isn't centered like the one of (1). For images where the face is from profile we would introduce lots of error since we only center the image from the bounding box given around the face.

We can see that the age estimation networks are over fitting and we didn't achieve good accuracy over all. Since we used the validation data set for training, the number of images was rather small. We tried to use the fit_generator function but it did not worked out. We think that the problem first comes from the small amount of images we used and also from the uneven classes distributions.

5. Conclusion

We did not obtain good prediction accuracy however we do have clues about what caused the over fitting issue and how to improve our network. Thus we think that using a pre trained model as explained in (9) could bring benefit for our case. It is a rather simple task since we used VGG16, we could load pre trained weight from a known data base. Moreover the small amount of images we used was also a major problem, if we were given more time we would come up with a strategy

to learn our model on more images. A simple one could have been to save the model and create another small data set to train on it and so on to train on the entire database.

6. Bibliography

References

[1] Karkkainen, K., Joo, J. (2021). FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1548-1558).
URL: https://github.com/joojs/fairface

[2] https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Object_Detection_Face_Detection_Haar_Cascade_Classifiers.php

[3] https://github.com/rodrigobressan/keras-multi-output-model-utk-face

[4] http://www.soolco.com/post/30827_1_1.html

[5] Wójcik, Waldemar and Gromaszek, Konrad and Junisbekov, Muhtar, 2016, Face Recognition: Issues, Methods and Alternative Applications

[6] Chen, W., Huang, H., Peng, S. et al. YOLO-face: a real-time face detector. Vis Comput (2020).

[7] J. Zhu and Z. Chen, "Real Time Face Detection System Using Adaboost and Haar-like Features," 2015 2nd International Conference on Information Science and Control Engineering, Shanghai, 2015, pp. 404-407, doi: 10.1109/ICISCE.2015.95.

[8] Ito, Koichi and Kawai, Hiroya and Okano, Takehisa and Aoki, Takafumi, 2018 Age and Gender Prediction from Face Images Using Convolutional Neural Network

[9] Qawaqneh, Zakariya and Abumallouh, Arafat and Barkana, Buket, 2017, Deep Convolutional Neural Network for Age Estimation based on VGG-Face Model