Convolutional Neural Networks for Age and Gender recognition

Lorenzo Ferrarin
University of Padova
lorenzo.ferrarin@studenti.unipd.it

Alessandro Benetti
University of Padova
alessandro.benetti.1@studenti.unipd.it

July 17, 2020

Abstract

Automatic age and gender identification has many possible useful applications. However this subject has not received the same amount of attention which has been given to the very closely related one of face recognition. As such it has not benefited as much as it could from the progress that this latest subject has shown in the recent years. In this paper we intend to analyze and evaluate current methods of age and gender identification, and discuss which paths are showing promise for further development and improvement of systems based on it. In particular we'll be focusing on the implementation and use of Convolutional neural networks to solve the recognition problem, as they've proven to be very useful in related fields.

1. Introduction

The problem of age and gender recognition is quite complex, and has not received as much attention as other similar ones such as face recognition. As a result even nowadays there are very few studies which are focused on finding approaches to tackle the subject. The two however should be talked about separately, as their complexity is quite different from one another. While gender recognition has by now managed to reach good results even on challenging datasets, age recognition appears to be much more complex, and even state of the art approaches still struggle to produce results suitable for use in real life applications.
The continuous and exponential growth of the number of images made available through the net is also a stimulus in improving these techniques. With such a huge amount of potential data available both for training and use for future application the refinement of techniques for solving these problems becomes even more appealing.
In this paper we'll analyze in particular methods for solving this problem with the use of Convolutional Neural Networks, which have proven to be very effective in the related subject of face recognition. In practice we'll be evaluating a series of models on the Adience benchmark, a notably challenging dataset which has often been used to examine the performance of such models.
The first model we'll use is the CNN proposed in a paper by Gil Levi and Tal Hassner [8], which was designed specifically to be trained on the Adience dataset. We'll then be using a more general yet simple model presented in the book from Tony Holdroyd [6], which was developed to be used on the CIFAR10 dataset. We also tried developing our own model, which will be presented in this paper. Finally we'll be covering the performance of the ResNet models made available through Keras, and their performance on the dataset while using the weights acquired by being previously trained on ImageNet [3].

2. Related Work

The topic of age and gender recognition has been the subject of several different papers and studies over the years, such as [5], [8], [11], [10]. The performance has been slowly improving for both challenges as the techniques are refined, and shows a marked rise in accuracy with the use of CNNs. For gender recognition it reaches results of up to 91% [1] on the challenging Adience benchmark, while for age recognition it has climbed from 45.1% [5] to 64% [10] on the same dataset. As we can see and have said before the gender recognition problem appears to be more easily solved than the age recognition one, which still proves extremely difficult in real life situations such as those in which the Adience pictures are taken. While the approaches of the various papers differ from each other all the most successful ones appear to be using CNNs. In particular the process of using a model with weights obtained on a much bigger dataset, such as ImageNet, and then refit to work on the Adience benchmark show significant promise.

Gender	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60+	Total
Male	745	928	934	734	2308	1294	392	442	8192
Female	682	1234	1360	919	2589	1056	433	427	9411
Both	1427	2162	2294	1653	4897	2350	825	869	19487

Table 1. **The Adience Faces benchmark.** Breakdown of the Adience Faces benchmark into the different Age and Gender classes.

3. Dataset

For our dataset the choice has fallen on Adience, which made its first appearence in the 2014 paper by Eran Eidinger, Roee Enbar and Tal Hassner [5], and is currently publicly available at The OUI Adience Face Image Project website [4]. We have decided to use this dataset because unlike lots of other collections of face images that are available it was designed specifically to capture faces in the wild, meaning in a more natural environment instead of under controlled conditions. The source for the photos in the dataset are Flickr.com albums, and as such the dataset contains the amounts of variations in appearance, noise, pose and lighting that are to be expected from images taken in average situations, without the careful preparation or subject posing which are often expected in more controlled environments. This means that it's particularly suited to analyze how a specific model will behave with images closer to those that might be used in real life applications. The dataset is also quite contained in size, which made it easier to use with the tools at our disposal.

The original dataset offers all the images both with aligned and unaligned faces in jpeg format, but since our aim was to test age and gender recognition models and not alignment techniques we used the already previously aligned images. The alignment was done with the tool described in the paper by Tal Hassner, Shai Harel, Eran Paz and Roee Enbar (2015) [14], and made public at [13].
The dataset comes already divided in five folds and each of these folds is described in its own text file containing the various information about the images, including the age and gender labels which we are interested in. This permade division allows for easy use of cross validation, while at the same time incentivizing the possibility for comparison, since all users, if they choose to, can have the exact same division of the dataset. The entire original dataset contains 26,580 images belonging to 2,284 individuals. Each individual was assigned with its apparent biological gender and age category. The possible age categories are eight and the assignment for the labels was done manually.

3.1. Preprocessing

As previously described the images we used were already aligned previously. However, since the dataset is not ordered and all the information is inside the text files describing the different folds, they did require being matched with their respective labels. As such we had to first read the folds' descriptions and compute the paths leading to the corresponding images, creating a new dataframe with all the desired information.
Each image was then read and resized to the desired dimension, which in our case due to the limitations of the tools at our disposal was 128 by 128 pixels. The image was afterwards turned into a python array which was then normalized for ease of analysis.
As described the dataset was already divided in five, allowing us to easily use four of the five folds for training and one for testing, however we also further split off 10% of the training folds to create a validation set to use during fitting.

3.2. Data augmentation

The original Adience dataset is quite small for today's standards, as such we decided to use data augmentation techniques to aid the training process. In particular we used an image data generator made available through Keras to augment our images in real time through the use of the flow function.
The different types of transformations we've applied to the images are the following:

- Rotation: the image can be randomly rotated in the range of 0 to 10 degrees;
- Horizontal shift: the image can be shifted horizontally up to 10% of its total width;
- Vertical shift: the image can be shifted vertically up to 10% of its total height;
- Horizontal flip: the image can be flipped horizontally;
- Zoom: the image can be zoomed up to 10%.

For a deeper look into data augmentation in the field of deep learning we recommend reading *Deep residual learning for image recognition* [12].

4. Method

The approach we have taken to solving age and gender recognition is to use Convolutional neural networks. They have proven to be very useful for face recognition [9], and have shown great promise in age and gender classification as well [8] [11].
To create and test our models we have decided to use Tensorflow as it's the most suitable library for the job. The recently released version 2 of the software however did require updating and implementing a lot of the older models which had been developed either with Caffe or

previous versions of Tensorflow. To more easily assemble our models we used Keras, which also proved essential for the already trained models it provides.

We have focused our efforts in particular on four different models, analyzing their performance on the Adience benchmark. The first model is the one proposed and described in [8]. It's not a particularly deep CNN, but it has been tailored specifically to be trained and evaluated with the dataset we have chosen. It is composed of only three convolutional layers, followed by two fully-connected ones with 512 neurons each, using dropout to reduce overfitting. The convolutional layers have 96, 256 and 384 filters respectively, and their kernel size is 7 by 7, 5 by 5 and 3 by 3. The architecture uses ReLU as its activation function and a series of max pooling layers to downsample the images. The centering and rescaling of the fatures is done through batch normalization, and the final fully connected layer is fed to a soft-max activation layer which calculates the probabilities for each output class and then proceeds to make the prediction based on the class with the highest probability. The choice to keep the CNN relatively narrow was conscious and the first reason is that the classification problem has a very small number of target classes, two for gender and eight for age. The second reason is that, especially with such a small dataset, with a deeper network the risk of overfitting would have become much higher.

The second model, which can be found in [6], was originally designed to analyze the CIFAR10 dataset [2], which isn't tied directly with age, gender or face recognition. However we decided to include it to analyze how a somewhat more generic model fares in this challenge, as it might provide some interesting insights. This model has three sections with the same structure. They start with two convolution layers of size 3x3 divided by an elu activation layer and batch normalization. The number of filters in the two convolution layers increases in each section, with values of 32, 64 and 128 respectively. Each section is closed by a max pooling layer with a size of 2x2, and then a dropout layer with increasing probability equal to 0.2, 0,3 and 0,4 respectively. The model ends with a fully connected layer using soft-max to map the results to the different output categories the same way the previously described model does.

The third type of model we have decided to use is ResNet [7], in particular ResNet50 and ResNet152. ResNet is short for residual network, meaning a network that takes advantage of residual learning to counterbalance its deeper architecture. The deeper a network goes in fact the more likely it is for it to reach accuracy saturation or degradation. Residual networks aim to remove this problem by using shortcut connections to skip some of the layers. Describing the entire architecture of ResNet lies outside the aim of this paper, but it can be found in the original one [7], for this paper we have chosen to use ResNet50 and ResNet152, where the number refers to the amount of layers in the architecture. An important note about our use of ResNet is that we purposefully initialized the model with the weights from its previous training in ImageNet [3], a particularly large dataset of over 14 million images. We did it because our aim was to check how well the model would be able to transfer the knowledge from the much bigger dataset to our smaller and more specialized benchmark. To do this we simply took the model's output and fed it first to a max pooling layer, followed by a dropout layer with a probability of 0,5 and then finally to the Softmax layer which maps it to the desired output classes.

Finally the last model, shown in figure 1, was developed

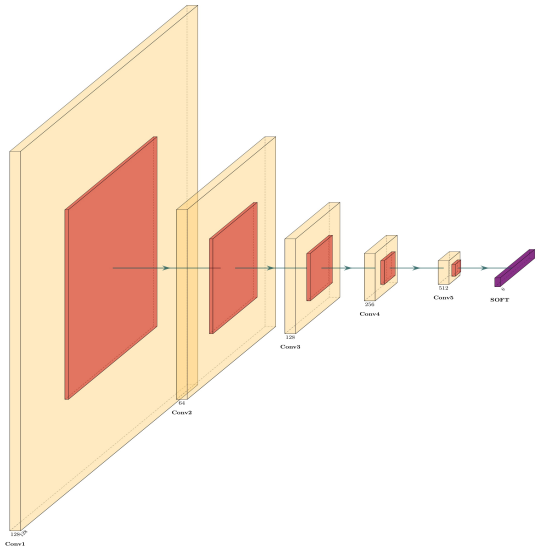

Figure 1. Architecture of our model.

by ourselves. To choose the number of the various layers of the model we continuously trained new models on our dataset, modifying one at a time the number of layers, filters, dropout chances and other variables. As we trained the models we monitored its performance, and chose each time the number that would bring us a better outcome. The final result is composed of five groups of layers with very similar structure. Each group starts with a convolutional layer of kernel size 5. The number of filters of this layer doubles at each group, starting from 32 on the first one up to 512 on the last one. This layer is followed by a max pooling of size 2 for downsampling, which itself is followed by an activation layer with relu, a batch normalization and finally a droupout layer. The chance of dropout increases as we go deeper in the network, starting at 0,2 for the fist three layers and then increasing to 0,3 and 0,4 on the last two. The output from the final group is fed to the Softmax layer

which maps it to the output categories.
For each of these models we have set the loss function to minimize as categorical cross-entropy. The categorical cross entropy loss function is obtained from the combination of a Softmax activation plus a cross-entropy loss. The softmax activation function squashes a vector in the range (0, 1) so that all the resulting elements add up to 1 and is applied to the output vector s from a CNN. The Softmax function cannot be applied independently to each element of the vector s_i, which represent each class, since it depends on all elements of s. For a given class s_i, the Softmax function can be computed as:

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}}$$

Where s_j are the scores inferred by the net for each of the classes in C.
The cross-entropy loss function instead is calculated as following:

$$CE = -\sum_i^C t_i log(s_i)$$

Where C is the number of classes our output can fall into, t_i is the ground truth expected value and s_i is the CNN's score for each class. In the case of Multi-Class classification the labels are one-hot encoded, which means that only the positive class C_p keeps its term in the loss since there is only one element of the Target vector t which is not zero $t_i = t_p$. It follows that by discarding the elements of the sum which are zero due to target labels and combining the two functions we obtain the categorical cross-entropy function:

$$CE = -log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right)$$

Where Sp is the CNN score for the positive class.

5. Experiments

For our experiments we have used a local machine and a free use cloud machine made available by Kaggle. The cloud machine has a limit of 13GB of RAM with a 2-core Intel Xeon as CPU, and offers a Tesla P100 with 16GB of VRAM as GPU. The local machine instead uses a CPU with 8 logical cores (4 physical ones), 16 GB of RAM and a Geforce 1070 (dedicated Nvidia GPU with 2056 CUDA cores). Ideally, and to keep the same standard as previous papers, we intended to use the pictures from the dataset scaled to 256x256 pixels, however due to the limitations imposed by the hardware at our disposal we had to settle for the smaller resolution of 128x128. As such we have to keep this into account as we analyze the results of the various models on the benchmark, and it's another reason why we decided to implement and test again models which had already been previously evaluated.
The development was done on the cloud environment made available by Kaggle, this meant however that during testing the maximum size of the images we could use was 64x64 pixels. In the end though the final results were obtained by running the various models on our local machine, which allowed us to increase the size up to 128x128.
As explained earlier all the experiments were run on six models:

- Levi-Hassner model: Model created based on the specifications given in [8];
- Original model: New original model created for this paper;
- Tony Holdroyd model: Model taken for Holdroyd's book and originally designed for CIFAR10 [6];
- ResNet50 model: Residual network model of depth 50, made available through Keras [7];
- ResNet152 model: Residual network model of depth 152, made available through Keras [7].

To test the accuracy of the different model we have divided our dataset in five folds, as they were described in the text files included in the Adience dataset. We then used four of those folds as a training set, and one as test set. We repeated this process five times, each time selecting a different fold for our test set. We further split a portion equal to 10% off the training set to use as validation. We then calculated the accuracy of the models on each of these five configurations, and at the end computed the average, which is the data we are presenting.
As described earlier the dataset we've used for our tests was Adience. The number of images in the dataset is quite contained compared to others, and to mitigate this problem we used the data augmentation techniques described in the dataset section in real time while training the models. The fitting operation was limited to a maximum of 100 epochs. To speed up the training we used a callback monitoring the model's loss function during the epochs, which stopped the fitting process once the performance stopped improving.

The results for gender prediction can be found in Table 2. As we can see the model with the best performance is ResNet50. This is most likely due to the features it learned during its previous training on ImageNet, which we were able to transfer to the age classification problem. Interesting to note is that ResNet152, a deeper version of the same architecture, performs worse. This could be due to the overfitting caused by training a network too deep on a small dataset. It is interesting to note that the model designed to solve CIFAR10 actually performs better than both our own original model and the one proposed in [8]. This could be

Method	Accuracy
Levi-Hassner	80.3
Tony Holdroyd	83.4
Original model	82.7
ResNet50	**86.2**
ResNet152	83.8

Table 2. **Gender estimation results on the Adience benchmark.** Listed are the accuracy results of each model, with the best result highlighted in bold.

Method	Accuracy	One-off Accuracy
Levi-Hassner	41.0	81.84
Tony Holdroyd	37.7	75
Original model	41.9	79.50
ResNet50	**49.0**	**86.9**
ResNet152	47.9	86.3

Table 3. **Age estimation results on the Adience benchmark.** Listed are the accuracy results of each model, with the best result highlighted in bold.

due to the fact that it was designed to work on very small images, since the ones included in CIFAR10 are 64x64 pixels, and it's then able to capture the features better than the other who were designed for bigger sizes.

For the problem of age classification we also calculated the one-off accuracy, which is a measure where errors of one age group are not considered. This is very useful because it shows how accurate the model was in distinguishing the subjects between more general and forgiving age groups. The prediction results are found in Table 3. Again we can see that the best results are obtained wit the ResNet models, while the others are clearly behind in performance. Again the deeper ResNet152 has worse results than ResNet50, but only slightly this time. It's also very interesting to note that the architecture designed for CIFAR10 is now performing much worse than the others, probably due to it not being able to rely on either a structure optimized for the problem or a huge amount of data. Again we see that the best idea appears to be using a model trained on a bigger dataset, and then transferred to perform age and gender prediction.

As extra information we also created confusion matrices by comparing the various models' predictions against the true values. An example of these matrices is found in Table 4. In a confusion matrix each row represents the instances that were predicted being in a particular class, while each column represents the actual instances inside the class. This can be very useful to discern whether the model is often confusing two classes with one another. While not as insightful for gender prediction this is very useful for age classification, since it makes it very easy to spot which classes the model finds harder to categorize successfully. As we can see for all age groups most matches are within one group distance from the correct one, showing that the one-off accuracy is actually quite high as we calculated before. Interesting is that the model seems to struggle categorizing images in the 15 to 20 age group, preferring instead to categorize them in the 25 to 32 group. This is probably due to the much higher number of images in the latter, as we can see in Table 1, and the relative similarity between people belonging to the two groups.

Age group	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60+
0-2	**0.5**	0.44	0.03	0	0.03	0	0	0
4-6	0.04	**0.54**	0.36	0.04	0.02	0	0	0
8-13	0.02	0.17	**0.54**	0.06	0.17	0.04	0	0
15-20	0	0	0.06	**0.09**	0.73	0.11	0	0.01
25-32	0	0	0.02	0.05	**0.8**	0.1	0.01	0.02
38-43	0	0	0.01	0	0.52	**0.33**	0.04	0.1
48-53	0	0	0.01	0	0.19	0.28	**0.1**	0.42
60+	0	0	0.02	0	0.05	0.17	0.07	**0.69**

Table 4. **Age estimation confusion matrix of the ResNet50 model on the Adience benchmark.** Showed are the percentages of labels for each group. Every row/column pair shows the percentage of those matches between instances predicted to be that label and instances which actually have that label. This means that the diagonal shows the percentage of correct matches for each age group.

6. Conclusion

The aim of the project was to compare different models for age and gender recognition, while evaluating which showed more promise for future development. As expected the results are much better for gender rather than age prediction. However it's also true that the one-off accuracy of most models is actually relatively high, meaning that they mange to recognize a bit more broadly whether the subject in the picture is young or old quite well.

Our results are predictably a few percentages lower than those reported in other papers, such as [8] [10], but this was to be expected since we were forced to use images with a lower resolution (128x128 against 256x256) which must have impacted the performance. We also can't exclude that other variables might be at play, such as the different kind of data augmentation used in our experiments compared to others.

As we've seen in the end the most effective models appear to be the ResNet architectures, thanks to their previous knowledge obtained by training on a much larger dataset which we were able to transfer to our benchmark successfully.

6.1. What we have learned

During the writing of this paper we managed to learn how to create and structure Convolutional neural networks

in Tensorflow. We had to learn how to correctly read and understand various papers and recreate the architectures described in them. We also learned how to translate code from older versions of Tesorflow to the recently released new one, and recreated implementations of old models which are not currently available in Tensorflow 2.
We also learned how to adapt a previously trained architecture, in our case ResNet, to our specific needs, using the information it had already gained on a different subject to solve the problem we gave it.
We learned how to enhance a dataset with data augmentation, as well as how to preprocess an image, decoding and scaling it before normalizing the information for ease of learning.
We also had to learn how to manage the very long training times of the models, trying to find optimizations such as the use of Tensorflow datasets to both speed the data loading, as well as to handle it in more manageable bashes.

6.2. Future work

The most promising path for age and gender prediction seems to lay in transferring to this problem the knowledge gained in other related fields such as that of face recognition. The huge datasets which are today available for the training of models aimed at solving those challenges can still be useful even for the problem presented in this paper, since the features learned by training on them can then be relatively easily transferred. It would be interesting to see the performance of more modern and complex models for visual recognition being transferred on the Adience benchmark, potentially also with a more complex transfer architecture than the basic one we have used for this paper.
These efforts could help bridge the gap which today has formed between these different but related challenges, improving even considerably the performance on this challenging benchmark.
Our results also showed the importance of having a big dataset for training, unlike the relatively small one provided by Adience. Creating a deep and complex architecture and then training it on a very small dataset is not very useful and can be counterproductive, as such it would be very interesting to see some bigger datasets with curate age and gender labels being publicly released, allowing the development of deeper CNNs with the specific aim of solving this problem.

References

[1] G. Shu A. Dehghan, E. G. Ortiz and S. Z. Masood. Deep age, gender and emotion recognition using convolutional neural network. *arXiv:1702.04280*, page 1–14, 2017.

[2] AlexKrizhevsky. Learning multiple layers of features from tiny images, 2009.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. The oui adience face image project. *https://talhassner.github.io/home/projects/Adience/Adience-data.html*.

[4] Roee Enbar Eran Eidinger and Tal Hassner. The oui adience face image project. *https://talhassner.github.io/home/projects/Adience/Adience-data.html*.

[5] Roee Enbar Eran Eidinger and Tal Hassner. Age and gender estimation of unfiltered faces. *Transactions on Information Forensics and Security (IEEE-TIFS), special issue on Facial Biometrics in the Wild*, 9(12):2170 – 2179, 2014.

[6] Tony Holdroyd. Tensorflow 2.0 quick start guide, 2019. *published by Packt*.

[7] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, page 770–778, 2016.

[8] Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston*, 2015.

[9] Weihong Deng Mei Wang. Deep face recognition: A survey. *arXiv:1804.06655v8*, 2019.

[10] R. Timofte R. Rothe and L. Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, page 1–14, 2016.

[11] Klaus-Robert Muller Sebastian Lapuschkin, Alexander Binder and Wojciech Samek. Understanding and comparing deep neural networks for age and gender classification. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1629–1638, 2017.

[12] Khoshgoftaar T.M. Shorten, C. A survey on image data augmentation for deep learning. *Journal of Big Data 6*, 2019.

[13] Eran Paz Tal Hassner, Shai Harel and Roee Enbar. Imagenet: A large-scale hierarchical image database. *http://www.image-net.org/papers/imagenet$_c$vpr09.bib*.

[14] Eran Paz Tal Hassner, Shai Harel and Roee Enbar. Effective face frontalization in unconstrained images. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston*, 2015.