Pattern Recognition Letters
journal homepage: www.elsevier.com

Minimalistic CNN-based ensemble model for gender prediction from face images

Grigory Antipov[a,b,**], Sid-Ahmed Berrani[a], Jean-Luc Dugelay[b]

[a] *Orange Labs – France Telecom, 4 rue Clos Courtel, 35512 Cesson-Sévigné, France*
[b] *Eurecom, 450 route des Chappes, 06410 Biot, France*

ABSTRACT

Despite being extensively studied in the literature, the problem of gender recognition from face images remains difficult when dealing with unconstrained images in a cross-dataset protocol. In this work, we propose a convolutional neural network ensemble model to improve the state-of-the-art accuracy of gender recognition from face images on one of the most challenging face image datasets today, LFW (Labeled Faces in the Wild). We find that convolutional neural networks need significantly less training data to obtain the state-of-the-art performance than previously proposed methods. Furthermore, our ensemble model is deliberately designed in a way that both its memory requirements and running time are minimized. This allows us to envision a potential usage of the constructed model in embedded devices or in a cloud platform for an intensive use on massive image databases.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

The human's gender plays a fundamental role in social interactions. Automatic gender classification has many important applications like intelligent user interface, visual surveillance, collecting demographic statistics for marketing, etc. Therefore, automatic gender recognition from face images has been extensively studied in computer vision. However, the difficulty of this problem largely depends on the application context and on the experimental protocol: a recognition model can be trained and tested on faces from the same dataset or from different datasets (i.e. cross-dataset experiment), images of input faces can be taken under controlled or uncontrolled conditions and finally faces can be aligned before gender prediction or not. The state-of-the-art performance in the most stringent conditions (i.e. cross-dataset, in uncontrolled environment and with no image preprocessing) reaches 96.86% of accuracy and was very recently obtained by Jia and Cristianini (2015) using a huge private training dataset of 4,000,000 images.

Deep Convolutional Neural Networks (CNNs) (LeCun and Bengio (1995)) have recently become the golden standard for object recognition (Krizhevsky et al. (2012); Simonyan and Zisserman (2014)). Today, CNNs are the primary choice for the large variety of computer vision tasks (Goodfellow et al. (2013); Taigman et al. (2014); Vinyals et al. (2014)). However, there are 2 problems which make the practical usage of CNNs difficult in some cases. The first problem is related to the big size of the training data which is often required to train them. Collecting large datasets of faces can be costly and can raise a number of privacy protection issues. That is why, successful face-related applications of CNNs are often trained on huge private datasets containing several millions of images (like in Taigman et al. (2014)) making the obtained results non-reproducible for the scientific community. The second problem lies in the domain of the computational and memory requirements of CNNs (He and Sun (2014); Gong et al. (2014)). This problem often hinders importing CNNs onto embedded platforms like smartphones and tablets or their usage in cloud computations. For example, 16-layers CNN described in Simonyan and Zisserman (2014) has a weights file bigger than 500MB and requires about $3.1 \cdot 10^{10}$ floating point operations per image. Specifically, 90% of its weights is taken up by the fully-connected layers and more than 90% of its running time is taken by the convolutional layers (He and Sun (2014)). It means that if we want to minimize both the running time and the required memory we have to minimize both fully-connected and convolutional layers.

In this work, we address the problem of gender recognition from face images taking into account the memory and the running time issues and by using a relatively small training

**Corresponding author: Tel.: +33 2 99 12 41 88; fax: +33 2 99 12 40 98;
e-mail: grigory.antipov@orange.com (Grigory Antipov)

dataset. In particular, we design a CNN-based ensemble model obtaining the state-of-the-art performance on gender recognition from face images in the most stringent conditions. We use a publicly available dataset of face images to train our CNN-model obtaining the highest recognition accuracy with about 10 times less training data than the state-of-the-art authors ((Jia and Cristianini, 2015)). Our model is also minimized both in terms of the running time and the memory requirements making its usage possible even on devices with a limited memory and without dedicated graphical processors for computations.

The rest of the paper is organized as follows: the overview of the related literature is done in Section 2; the datasets used for training and test are presented in Section 3; the Starting CNN and the methodology to progressively minimize it are proposed in Section 4; the procedure of minimization of the Starting CNN is described in Section 5; the classification results are analysed in Section 6; and the conclusions are summarized in Section 7.

2. Related work

In this section, we make on overview of existing works on gender recognition from face images.

Early works on gender recognition from face images focused on the case of frontal faces in a controlled laboratory environment. In the beginning of the 90's, many authors tried neural networks to deal with this problem. For example, Golomb et al. (1990) trained a 2-layers fully-connected neural network and achieved 91.90% accuracy on a tiny test set of 90 images. The benchmark dataset of frontal faces in a controlled environment is FERET (Phillips et al. (1998)). With the emergence of SVM, Moghaddam and Yang (2002) used this classifier with an RBF kernel on raw pixels and obtained 96.62% accuracy on FERET (though having the same persons presented both in training and test sets). Rather than using SVM, Baluja and Rowley (2007) used AdaBoost on raw pixels and obtained 96.40% on FERET without mixing people in training and test sets. Li et al. (2012) combined facial information with clothing and hair components obtaining 95.10% accuracy on the FERET dataset. Ullah et al. (2012) used the Webers Local texture Descriptor to reach almost perfect performance of 99.08% on FERET. This result suggests that the FERET benchmark is saturated and not enough challenging for modern methods.

As a result, the majority of contemporary works deals with the problem of gender recognition from face images in an uncontrolled environment. The Labeled Faces in the Wild (LFW) dataset (Huang et al. (2007)) is the most frequently used one in this case. Different works on gender recognition in an uncontrolled environment are compared in Table 1. Shan (2012) employed Local Binary Patterns (LBP) features with an AdaBoost classifier to obtain 94.81% on LFW. Shih (2013) used the Active Appearance Model (AAM) in order to align face images and to model them using small patches around the detected landmarks. The Bayesian framework was employed as a classifier. The resulting model obtained 86.50% classification accuracy on the combination of the color FERET and LFW datasets. Tapia and Perez (2013) fused LBP features with different radii and spatial scales and used an SVM classifier above. The authors performed 2 experiments: in the first one, they trained and tested their models on different subsets of LFW, while in the second one, the training was done on a separate dataset. Results of these 2 experiments (95.60% and 98.01%) differ quite significantly from each other proving that the cross-database protocol is more challenging. Bekios-Calfa et al. (2014) showed that it may be advantageous to predict the person's gender simultaneously with the person's age and pose in the photo. They got 79.11% gender recognition accuracy training their model on the GROUPS dataset and testing on the LFW dataset. The most recent attempt to employ CNNs for gender recognition from face images was done by Levi and Hassner (2015). Authors trained a CNN on the newly created Adience dataset. They obtained a relatively modest accuaracy of 86.80% mainly because of the low quality of images in Adience. Finally, the most recent result on the LFW dataset under the cross-database protocol was obtained by Jia and Cristianini (2015). The authors used a huge private dataset of 4,000,000 images to train a C-Pegasos classifier (a variation of SVM) using LBP fetaures. They obtained a state-of-the-art accuracy of 96.86% on LFW referring their success mainly to the size of the training dataset.

In this work, we use the result of Jia and Cristianini as a baseline for comparison with our models.

Table 1. Gender recognition results in an uncontrolled environment.

Authors	Test dataset	Method	Cross-Dataset	Accuracy
Shan (2012)	LFW	LPB + AdaBoost	No	94.81%
Shih (2013)	color FERET + LFW	AAM + Bayesian	No	86.50%
Tapia and Perez (2013)	LFW	multiscale LBP + SVM	No	98.01%
			Yes	95.60%
Bekios-Calfa et al. (2014)	LFW	appearance-based + LDA	Yes	79.11%
Levi and Hassner (2015)	Adience	CNN	No	86.80%
Jia and Cristianini (2015)	LFW	multiscale LBP + C-Pegasos	Yes	96.86%

It should be mentioned that face images are by far not the only possible modality to predict a person's gender. There are works on gender predictions from gait (Lu et al. (2014); Flora et al. (2015)), speech (Metze et al. (2007)), images of silhouettes (Antipov et al. (2015)) and even web forum messages (Zhang et al. (2011)). However, in this work, we focus only on the gender prediction from face images and therefore do not consider other modalities.

3. Datasets

In this section, we present face datasets which have been used in our experiments.

We have used 2 publicly available face datasets: CASIA WebFace and Labeled Faces in the Wild (LFW). The first one is

used for training and validation whereas the second one is used only for testing. While collecting the CASIA WebFace dataset, its authors made sure that there are no subject intersections between CASIA WebFace and LFW (Yi et al. (2014)).

3.1. CASIA WebFace dataset

CASIA WebFace dataset was collected for the face recognition purposes by Yi et al. (2014). The dataset contains photos of actors and actresses born between 1940 and 2014 from the IMDb website.[1] Images of the CASIA WebFace dataset include random variations of poses, illuminations, facial expressions and image resolutions. In total, there are 494,414 face images of 10,575 subjects. To the best of our knowledge, CASIA WebFace is the biggest publicly available face dataset today, and that is why we have used it to train CNNs in this work.

Authors of CASIA WebFace provide names of 10,575 subjects but not their genders. We have annotated genders using the metadata provided by IMDb and also by manual annotation.

3.2. LFW dataset

Being collected by Huang et al. (2007), the LFW dataset has become a benchmark for face gender recognition in an unconstrained environment. It consists of 13,233 face images of 5,749 celebrities. Contrary to CASIA WebFace, LFW does not only contain photos of actors and actresses but it also contains photos of politicians, sportsmen and sportswomen.[2]

3.3. Data preprocessing

Images of both CASIA WebFace and LFW are face-centred and have an initial resolution of 250x250 pixels. The two datasets have been processed in the same way: the faces are firstly extracted with the Viola-Jones face detector (Viola and Jones (2001)), and then they are rescaled to a certain square size (the particular size depends on the input dimensions of a CNN). This process is illustrated in Figure 1. In case if several faces are found in an image, only the biggest one is taken; if no faces are found in an image, the image is ignored. After face extraction, we have obtained 452,042 face images from the CASIA WebFace dataset. These images have been split into training and validation sets in the proportion of 95% and 5% respectively. We have ensured that there are no subject intersections between training and validation sets. In order to be able to fairly compare our results with the current state-of-the-art in gender recognition on LFW, we have used exactly the same test set of 10,147 face images as the authors of the current best result on LFW (Jia and Cristianini (2015)). Following their work, we have not performed any sort of alignment to the test images prior to gender classification. More details on the data split into training, validation and test sets are given in Table 2.

[1] `http://www.imdb.com/` Intenet Movie Database (IMDb) is an online database of information related to films, television programs and video games.

[2] Gender annotations for the LFW dataset are available at `http://face.cs.kit.edu/431.php`

Fig. 1. Example of an extracted face which is used as an input to a CNN.

Table 2. Data split into training, validation and test sets.

	Dataset	Men faces	Women faces
Training	CASIA WebFace	229,330	197,129
Validation	CASIA WebFace	12,440	13,143
Test	LFW	7,804	2,343

4. Starting CNN and the way to minimize it

In order to address the problem of gender recognition from face images, we design a powerful and complex CNN performing as good as the current state-of-the-art by Jia and Cristianini (2015) (96.86%) on the LFW dataset. This CNN is referred as "Starting CNN" below.

Starting CNN is a simplification of the CNN proposed by Simonyan and Zisserman (2014) for the Imagenet classification (Russakovsky et al. (2014)) (in particular, we simplify the CNN B from their article). Following the work of Simonyan and Zisserman, in the Starting CNN, filters of all convolutional layers have a spatial dimension of 3x3 pixels and in all layers, rectified linear units (ReLU) are used as activation functions.

However, the Starting CNN has several differences from its initial prototype by Simonyan and Zisserman (2014). In the Starting CNN, the input image resolution is 128x128 pixels instead of 224x224 pixels. We use a lower resolution because initial resolutions of face images in CASIA WebFace and LFW vary approximately from 60x60 to 120x120 pixels, and it does not make sense to significantly upsample input faces. Taking into account the smaller inputs, Starting CNN contains 8 instead of 10 convolutional layers. Finally, due to the fact that our problem is less complex than Imagenet classification (2 target classes instead of 1000 classes), we have reduced the number of filters in the convolutional layers and we have used only one fully-connected layer. The architecture of Starting CNN is described in details in the first column of Table 3.

Having constructed the Starting CNN, we have focused on its optimization: the objective is to drastically reduce the running time and the memory requirements while preserving the classification performance.

We propose to perform this optimization by:

1. Minimizing the input image size and the associated number of convolutional layers. We associate the input image size with the number of convolutional layers in order to keep the number of connections between the last convolutional layer and the fully connected layer fixed. It allows us to optimize the running time without varying the total number of connections too much.

2. Minimizing the number of filters in convolutional layers.
3. Minimizing the size of the fully-connected layer.

In the steps 1 and 2, we minimize the size and the number of convolutional layers while in the last step, we minimize the fully-connected layer. Thus, in the steps 1 and 2, we minimize the running time of the CNN while in the last step, we minimize its memory requirements.

The optimization proceeds as follows: in every step, we progressively reduce each parameter by a factor of 2 until the CNN performance on the validation set starts to deteriorate significantly. Every step of the proposed methodology is described in details in the next section.

5. Minimization of the Starting CNN

We design an optimal CNN for gender recognition from faces by progressively minimizing the Starting CNN described in Section 4. Every optimization step, a certain CNN architecture is selected to be further optimized in the following step.

5.1. Progressive optimization

5.1.1. Step 1. Input size and number of convolutional layers

In the first step, we minimize the size of input images and the number of convolutional layers. We progressively reduce the size of input images from 128x128 pixels in the Starting CNN by a factor of 2. Thus, we obtain 3 CNNs: *A*, *B* and *C*. They are detailed in Table 3. The size of the outputs of the last convolutional layer is kept constant (64 feature maps of 8x8 pixels) among all networks (the Starting CNN, *A*, *B* and *C*) by varying the number of the convolutional and the pooling layers.

The performances of different CNNs are assessed based on the classification accuracies that are observed on the validation set. The results are presented in Figure 2. In this figure, the bars are ordered in the same way as columns in Table 3. For each considered CNN architecture, 3 CNN instances have been trained from scratch (each time an initialization of weights is random) and in Figure 2, the bars represent the mean accuracies. Corresponding standard deviations are given by error segments. We have fixed a selection threshold accuracy of 97.5% on the validation set (shown by the dash-dotted horizontal line in Figure 2), which corresponds to the accuracy of the Starting CNN on the validation set. This threshold is used to select CNN architectures in all 3 steps of the optimization procedure.[4] In order to illustrate how the validation accuracy on the CASIA WebFace dataset is related to the test accuracy on the LFW dataset, we also present the test accuracies of all compared CNNs as well as the state-of-the-art baseline by Jia and Cristianini (2015) in Figure 2. However, we highlight that the results on the test set are not used in the model selection.

[3]"Conv: N@MxM" denotes a convolutional layer with N filters of size MxM. "MaxPool: MxM" means that input maps are downsampled by a factor of M using Max-Pooling. "FC: N" denotes a fully-connected layer with N neurons.

[4]We consider that a certain CNN architecture satisfies the selection threshold if the threshold (i.e. 97.5%) is inside the $[-\sigma, \sigma]$ segment for this architecture (where σ is the standard deviation).

There is no significant difference between the accuracies of the Starting CNN, the CNN *A* and the CNN *B*. All of them show the validation results which are very close to the threshold accuracy (with respect to standard deviations). The accuracy of the CNN *C* is significantly lower than accuracies of the first 3 networks: about 1.5% decrease of accuracy is observed on the validation set. Therefore, as the objective is to reduce the complexity of the Starting CNN while preserving its performance, the CNN *B* is selected after this first optimization step.

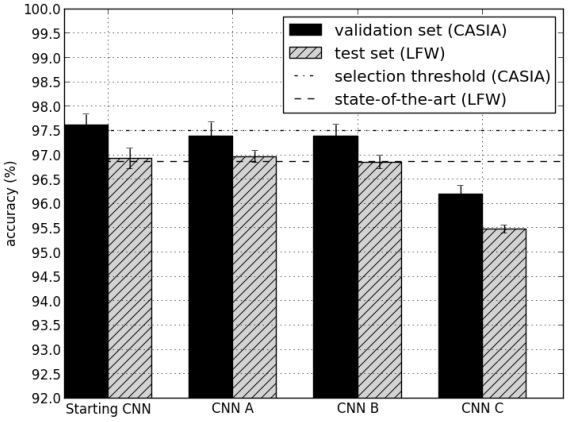

Fig. 2. Step 1: performances of the Starting CNN and the CNNs *A* and *B*.

5.1.2. Step 2. Number of convolutional filters

In this step, the goal is to minimize the width of convolutional layers (here, the width refers to a number of convolutional filters at each convolutional layer).

As detailed in Table 3, the number of convolutional filters of the CNN *D* is divided by 2 comparing to the CNN B (which has been selected during the first optimization step). The results obtained using CNNs *B* and *D* are summarized in Figure 3.

The CNN *D* is clearly below the selection threshold on the validation set (with respect to standard deviations). Therefore, the CNN *B* is selected again after the second optimization step.

5.1.3. Step 3. Size of the fully-connected layer

This step focuses on the minimization of the fully-connected layer size. Starting from 512 neurons in the CNN *B*, we have reduced the number of neurons by a factor of 2 until only 2 neurons are left in the CNN *L* (see Table 3). We also evaluate the performance of the CNN *M* where there is no fully-connected layer at all (in this case, the outputs of the last convolutional layer are directly connected with 2 neurones of the Softmax layer). The obtained results are presented in Figure 4.

This time, the difference between CNNs is less significant than in the 2 first optimization steps. Apparently, the size of the fully-connected layer is less influential on the final accuracy than the number and the width of the convolutional layers.

However, we can observe that CNNs *B*, *E* — *I* reach the selection threshold of 97.5% classification accuracy with respect to standard deviations, while the performances of the CNNs

Table 3. Optimization of the Starting CNN.[3]

Starting CNN	**Optimization: candidates at step** 1			**Optimization: candidate at step** 2	**Optimization: candidate at step** 3								
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
Input: 128x128	Input: 64x64	Input: 32x32	Input: 16x16	Input: 32x32	Input: 32x32								
Conv: 32@3x3 Conv: 32@3x3	Conv: 32@3x3 Conv: 32@3x3	Conv: 32@3x3 Conv: 32@3x3	Conv: 64@3x3 Conv: 64@3x3	Conv: 16@3x3 Conv: 16@3x3	Conv: 32@3x3 Conv: 32@3x3								
MaxPool: 2x2	MaxPool: 2x2	MaxPool: 2x2	MaxPool: 2x2	MaxPool: 2x2	MaxPool: 2x2								
Conv: 32@3x3 Conv: 32@3x3	Conv: 32@3x3 Conv: 32@3x3	Conv: 64@3x3 Conv: 64@3x3		Conv: 32@3x3 Conv: 32@3x3	Conv: 64@3x3 Conv: 64@3x3								
MaxPool: 2x2	MaxPool: 2x2	MaxPool: 2x2		MaxPool: 2x2	MaxPool: 2x2								
Conv: 64@3x3 Conv: 64@3x3	Conv: 64@3x3 Conv: 64@3x3												
MaxPool: 2x2	MaxPool: 2x2												
Conv: 64@3x3 Conv: 64@3x3													
MaxPool: 2x2													
FC: 512					FC: 256	FC: 128	FC: 64	FC: 32	FC: 16	FC: 8	FC: 4	FC: 2	
Softmax: 2													

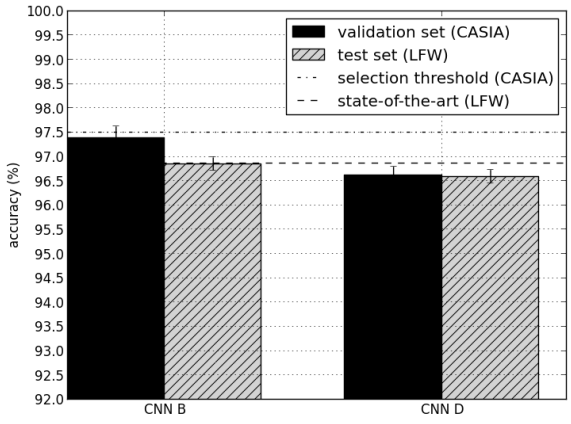

Fig. 3. Step 2: performances of the CNN *B* and the CNN *D*.

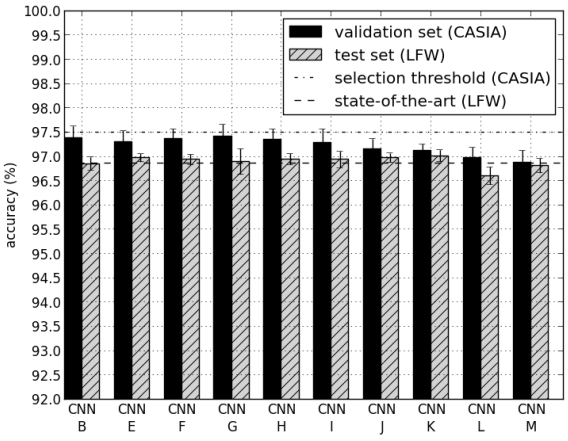

Fig. 4. Step 3: performances of the CNN *B* and the CNNs *E* — *M*.

J — *M* are below the selection threshold on the validation set. Hence, the CNN *I* is selected after the last optimization step.

5.2. The optimization gain

In order to better understand the computational and memory gains of the optimization, we have compared the Starting CNN, the CNN *B* and the CNN *I*. CNNs *B* and *I* have been selected in different steps of the optimization procedure. The comparison is performed both in terms of the required memory and the running time. Results are presented in Table 4.

The number of weights and the size of the weights file are clearly proportional. The exact size of the weights file depends on the particular implementation of the weights storage, therefore corresponding values in Table 4 are indicative.

[5]The timings have been calculated based on 5 trials. Used CPU: Intel E5-1620v2, 3.70GHz, 8-cores. Used GPU: Tesla K20c.

Table 4. Computational and memory gains from the optimization.[5]

CNN	Number of weights	Size of the weights file (KB)	Time to classify 100 images (s)	
			GPU	CPU
Starting CNN	2,256,610	8,851	0.2395 ± 0.0003	27.9624± 0.0278
CNN *B*	2,164,258	8,477	0.0184 ± 0.0001	2.5013 ± 0.0065
CNN *I*	131,154	534	0.0180 ± 0.0001	2.4657 ± 0.0091
Ensemble of 3 **CNNs *I***	131, 154∗3 = 393, 462	534∗3 = 1, 602	0.0501 ± 0.0005	7.3951 ± 0.0312

The CNN *B* has fewer convolutional layers than the Starting CNN. The fully-connected layer is exactly the same in both networks. Hence, the two CNNs have similar numbers of weights but the CNN *B* is about 11 to 13 times faster (depending on CPU/GPU processing). The absolute running time values show that the time gain becomes really essential when the processing is done by a CPU, which is often the case in big computational clusters or when images are processed separately rather than in batches (in the latter case, an impact from GPUs is negligible).

CNNs *B* and *I* share the same number of convolutional layers. However, the fully-connected layer of the CNN *I* is 32 times smaller than the fully-connected layer of the CNN *B*. As a result, the difference in running times between the CNNs *B* and *I* is negligible, but the weights file of the network *I* is about 16 times smaller than the weights file of the network *B*.

Thus, by performing the progressive minimization of the Starting CNN, we have obtained an equally accurate CNN *I* which is about $11 - 13$ times faster and about 16 times more memory efficient than the initial network.

As a result, the proposed 3-step methodology has proved to be very efficient in minimizing the Starting CNN for the gender prediction from face images. However, the problem of choosing an optimal CNN architecture for a specific problem remains an open subject. In this paper, we do not pretend to answer it in a general case, as the main goal of our study is designing an optimized and efficient CNN model for gender recognition from face images. Nevertheless, the proposed empirical CNN optimization methodology can be easily adapted to any problem of interest. For that, we suggest to start from an established and well-known CNN architecture (as it is done with the Starting CNN in this work) and progressively minimize it using the proposed 3-step methodology. Moreover, our approach can be combined with other optimization strategies (e.g. He and Sun (2014); Gong et al. (2014)) to minimize running time and required memory of a CNN model.

5.3. Training details

The training of all CNNs in this work has been carried out by optimizing the cross-entropy objective function using the mini-batch Nesterov's accelerated gradient descent (Nesterov (1983)). Backpropagation of the gradient has been performed with an initial learning rate of 0.01 and the momentum of 0.9. Contrary to some recent works where CNNs are used just as feature extractors and followed by other classification methods (like SVM (Razavian et al. (2014)) or ELM (Zeng et al. (2015))), in our experiments, CNNs have been used both as feature extractors and as classifiers. Input RGB-images have been normalized before CNN processing. Every epoch, faces are randomly substituted by their mirrored copy with the probability 0.5 (i.e. either face or its mirrored copy participates in every epoch). The size of a mini-batch has been set to 128. In order to prevent the CNNs from overfitting, we have employed the "dropout" regularization (Srivastava et al. (2014)) on the activations of convolutional layers and the fully-connected layer. We have made the ratio of the "dropout" to be dependent on the particular size of the convolutional or the fully-connected layer varying it from 0 (i.e. no "dropout") to 0.5. The training has been stopped once the validation accuracy stops improving. It corresponds to the moment when the training accuracy is between 98.0 and 98.1% (depending on the particular CNN architecture). Training has taken about 30 epochs with slight variations depending on the particular CNN, which corresponds to about 27 hours of training for the Starting CNN and 2.5 hours of training for the CNN *I* on the contemporary GPU. All experiments in this work have been performed using Theano deep learning library (Bastien et al. (2012)).

6. Analysis of classification results

In this section, we compare classification results of the selected CNN *I* performing alone and of several instances of the CNN *I* combined in a single model (which is referred as an "ensemble" model below). We also measure the impact of the size of the training data on the performance of a single CNN *I*.

6.1. Ensemble of models

In order to evaluate a gain from combining several CNNs *I* together, we have trained 3 instances of the CNN *I* (each instance is trained from scratch with a random initialization of weights). These instances have been combined in a single ensemble model by averaging the outputs of the corresponding softmax layers. Performances of a single CNN *I* and the ensemble model are compared in Table 5. For the single CNN *I*, we provide the resulting mean accuracy alongside with the corresponding standard deviation.

Table 5. Overall performance.

	Classification accuracy (LFW)
Jia and Cristianini (2015)	96.86%
CNN *I*	96.94 ± 0.18%
Ensemble of 3 **CNNs *I***	97.31%

The ensemble of 3 CNNs *I* performs better than the single CNN *I*. The gain is about 0.4% comparing to the mean accuracy of the single CNN *I*. The ensemble of CNNs *I* improves the current state-of-the-art performance by about 0.5%.

In the last row of Table 4, we provide the memory requirements and the running time for the ensemble of 3 CNNs *I*. Though the running time has been calculated by executing 3 CNNs one after another (executing them in parallel should be faster), the ensemble model is about 4 times faster than the Starting CNN. The total size of 3 weights files used in the ensemble CNN remains 5 times smaller than the weights file of the Starting CNN.

We have not observed better results by combining more than 3 CNNs in one ensemble. Therefore, *we have chosen the ensemble of* 3 *CNNs I as the final model in this work.*

6.2. Size of the training data

In order to assess the impact of the size of the training set on the resulting performance, we have trained several instances of the CNN *I* varying the number of images in the training set. The corresponding subsets of images have been selected randomly from the initial training set. For each considered training set size, we have trained 3 instances of the CNN *I*. Results of the comparison are summarized in Figure 5.

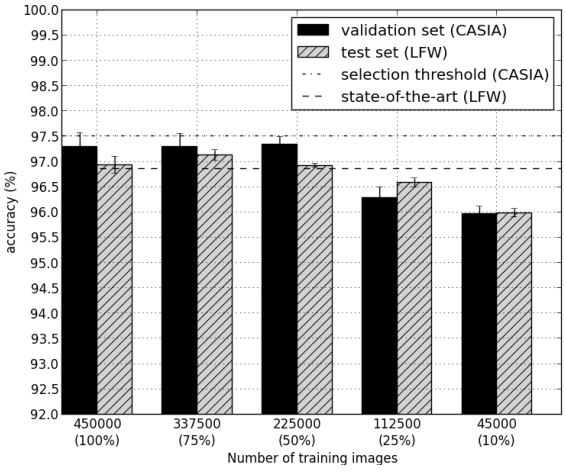

Fig. 5. Impact of the training set size on the performance of the CNN *I*.

Surprisingly, the accuracy of the CNN *I* trained only on a half of available images does not distinguish significantly from the accuracy of the CNN *I* trained on all data (with respect to standard deviations). In other words, the CNN *I* (performing alone) obtains the same performance as the baseline model by Jia and Cristianini (2015) with the training set of only 225, 000 images, which is almost 20 times smaller than the training set used by Jia and Cristianini (2015).

Further reducing of the training set down to 25% and 10% of its initial size leads to a loss of performance. However, this loss is relatively small. Thus, the CNN *I* which is trained on only 10% of the initial data (i.e. on 45, 000 images) performs reasonably well: 95.98% of correct gender predictions on LFW (it is better than, for example, the model by Shan (2012) which was the state-of-the-art on LFW in 2012).

The obtained results do not however show that the gender recognition performance of the CNN *I* is saturated. To illustrate this, we have trained the CNN *I* using the training images corresponding to only one half of the available persons in CASIA WebFace (contrary to the experiment in Figure 5, in this case we have reduced the number of different persons and not just the number of images). The resulting classification accuracy of the CNN *I* trained on one half of the persons has been about 0.3% lower than that of the CNN *I* trained on one half of the images. This result suggests that the CASIA WebFace dataset is a little bit redundant in the sense that the number of images is excessive with respect to the number of subjects. Therefore, the classification accuracy of the CNN *I* could have been improved even further with more diverse training dataset.

7. Conclusion

In this work, we have designed a CNN-based ensemble model for gender recognition from face images. The following results have been achieved:

1. The record performance of 97.31% on the LFW dataset has been set using the ensemble of 3 CNNs.
2. The record-breaking ensemble model has been trained with almost 10 times less images than the previous state-of-the-art model (Jia and Cristianini (2015)). Moreover, a single CNN *I* performs as good as the the model by Jia and Cristianini (2015) with almost 20 times less training images. This result is of a particular importance, given the cost and complexity of collecting large image datasets.
3. The CNNs that are used in the final ensemble model have been optimized in terms of their memory requirements and running times. As a result, the ensemble model requires about 1.5MB of the memory storage and is able to process 10 face images in less than 1 second on a contemporary CPU. It allows our ensemble model to be used in the context of limited computational and memory resources or in the context of processing of massive image datasets.
4. The proposed CNN optimization methodology is simple but efficient. It can be employed to minimize CNNs in other problems and combined with other CNN optimization approaches. This is a part of our future work.

The designed gender recognition ensemble model can be freely tested online via a simple demo website.[6] The results of this work are fully reproducible, since the final model and all dataset annotations have been made public.[7]

Acknowledgement

We thank Jia and Cristianini (2015) for providing us the exact list of LFW images used in their work and corresponding gender annotations. This allowed us to fairly compare our work with theirs. We also thank the anonymous reviewers for their constructive and relevant comments which helped us in improving the quality of the paper.

[6] The demo website: `https://cactus.orange-labs.fr/genderreco/`

[7] The model and the annotations of used datasets can be consulted at `https://cactus.orange-labs.fr/genderreco/cnnmodels/` and at `https://cactus.orange-labs.fr/genderreco/datasets/` respectively.

References

Antipov, G., Berrani, S.A., Ruchaud, N., Dugelay, J.L., 2015. Learned vs. hand-crafted features for pedestrian gender recognition, in: Proceedings of the conference on Multimedia, ACM, Brisbane, Australia.

Baluja, S., Rowley, H.A., 2007. Boosting sex identification performance. International Journal of Computer Vision 71, 111–119.

Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., Bengio, Y., 2012. Theano: new features and speed improvements. CoRR abs/1211.5590.

Bekios-Calfa, J., Buenaposada, J.M., Baumela, L., 2014. Robust gender recognition by exploiting facial attributes dependencies. Pattern Recognition Letters 36, 228–234.

Flora, J., Lochtefeld, D., Bruening, D., Iftekharuddin, K., 2015. Improved gender classification using nonpathological gait kinematics in full-motion video. IEEE Transactions on Human-Machine Systems 45, 304–314.

Golomb, B.A., Lawrence, D.T., Sejnowski, T.J., 1990. Sexnet: A neural network identifies sex from human faces., in: Proceedings of the conference on Advances in Neural Information Processing Systems, Denver, USA.

Gong, Y., Liu, L., Yang, M., Bourdev, L., 2014. Compressing deep convolutional networks using vector quantization. CoRR abs/1412.6115.

Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V., 2013. Multi-digit number recognition from street view imagery using deep convolutional neural networks. CoRR abs/1312.6082.

He, K., Sun, J., 2014. Convolutional neural networks at constrained time cost. CoRR abs/1412.1710.

Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E., 2007. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report. University of Massachusetts, Amherst.

Jia, S., Cristianini, N., 2015. Learning to classify gender from four million images. Pattern Recognition Letters 58, 35–41.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Proceedings of the conference on Advances in Neural Information Processing Systems, Lake Tahoe, USA.

LeCun, Y., Bengio, Y., 1995. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks 3361.

Levi, G., Hassner, T., 2015. Age and gender classification using convolutional neural networks, in: Proceedings of the Workshop on Analysis and Modeling of Faces and Gestures at CVPR, IEEE, Boston, USA.

Li, B., Lian, X.C., Lu, B.L., 2012. Gender classification by combining clothing, hair and facial component classifiers. Neurocomputing 76, 18–27.

Lu, J., Wang, G., Moulin, P., 2014. Human Identity and Gender Recognition From Gait Sequences With Arbitrary Walking Directions. IEEE Transactions on Information Forensics and Security 9, 51–61.

Metze, F., Ajmera, J., Englert, R., Bub, U., Burkhardt, F., Stegmann, J., Muller, C., Huber, R., Andrassy, B., Bauer, J.G., et al., 2007. Comparison of four approaches to age and gender recognition for telephone applications, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, IEEE, Las-Vegas, USA.

Moghaddam, B., Yang, M.H., 2002. Learning gender with support faces. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 707–711.

Nesterov, Y., 1983. A method of solving a convex programming problem with convergence rate o (1/k2). Soviet Mathematics Doklady 27, 372–376.

Phillips, P.J., Wechsler, H., Huang, J., Rauss, P.J., 1998. The feret database and evaluation procedure for face-recognition algorithms. Journal of Image and Vision Computing 16, 295–306.

Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S., 2014. Cnn features off-the-shelf: an astounding baseline for recognition, in: Proceedings of Computer Vision and Pattern Recognition Workshops, IEEE. pp. 512–519.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., 2014. Imagenet large scale visual recognition challenge. International Journal of Computer Vision , 1–42.

Shan, C., 2012. Learning local binary patterns for gender classification on real-world face images. Pattern Recognition Letters 33, 431–437.

Shih, H.C., 2013. Robust gender classification using a precise patch histogram. Pattern Recognition 46, 519–528.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958.

Taigman, Y., Yang, M., Ranzato, M., Wolf, L., 2014. Deepface: Closing the gap to human-level performance in face verification, in: Proceedings of the conference on Computer Vision and Pattern Recognition, IEEE, Columbus, USA.

Tapia, J.E., Perez, C.A., 2013. Gender classification based on fusion of different spatial scale features selected by mutual information from histogram of lbp, intensity, and shape. IEEE Transactions on Information Forensics and Security 8, 488–499.

Ullah, I., Hussain, M., Muhammad, G., Aboalsamh, H., Bebis, G., Mirza, A.M., 2012. Gender recognition from face images with local wld descriptor, in: Proceedings of the International Conference on Systems, Signals and Image Processing, IEEE, Vienna, Austria.

Vinyals, O., Toshev, A., Bengio, S., Erhan, D., 2014. Show and tell: A neural image caption generator. CoRR abs/1411.4555.

Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features, in: Proceedings of the conference on Computer Vision and Pattern Recognition, IEEE, Kauai, USA.

Yi, D., Lei, Z., Liao, S., Li, S.Z., 2014. Learning face representation from scratch. CoRR abs/1411.7923.

Zeng, Y., Xu, X., Fang, Y., Zhao, K., 2015. Traffic sign recognition using extreme learning classifier with deep convolutional features, in: Proceedings of the international conference on intelligence science and big data engineering (IScIDE 2015), Suzhou, China.

Zhang, Y., Dang, Y., Chen, H., 2011. Gender classification for web forums. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 41, 668–677.