2015 IEEE Symposium Series on Computational Intelligence

Deep Convolutional Neural Networks and Support Vector Machines for Gender Recognition

Jos van de Wolfshaar, Mahir F. Karaaba and Marco A. Wiering (*IEEE Member*)
Institute of Artificial Intelligence and Cognitive Engineering
Faculty of Mathematics and Natural Sciences
University of Groningen, The Netherlands

***Abstract*—Social behavior and many cultural etiquettes are influenced by gender. There are numerous potential applications of automatic face gender recognition such as human-computer interaction systems, content based image search, video surveillance and more. The immense increase of images that are uploaded online has fostered the construction of large labeled datasets. Recently, impressive progress has been demonstrated in the closely related task of face verification using deep convolutional neural networks. In this paper we explore the applicability of deep convolutional neural networks on gender classification by fine-tuning a pretrained neural network. In addition, we explore the performance of dropout support vector machines by training them on the deep features of the pretrained network as well as on the deep features of the fine-tuned network. We evaluate our methods on the color FERET data collection and the recently constructed Adience data collection. We report cross-validated performance rates on each dataset. We further explore generalization capabilities of our approach by conducting cross-dataset tests. It is demonstrated that our fine-tuning method exhibits state-of-the-art performance on both datasets.**

I. Introduction

The distinction of gender has always played an important role in many aspects of social interactions or communication with machines. Depending on whether a person is a man or a woman an interactive system might have to act accordingly. Gender classification from face images has been a well-studied topic in computer vision. Applications of face gender recognition range from human-computer interaction systems, content based image search, video surveillance and others. Face gender classification is a challenging problem since face images may vary in pose, lighting, expression and other factors.

The common approach to face gender estimation is by using supervised machine learning algorithms. Given the high-dimensional nature of real-world image data, these algorithms typically require large datasets to perform adequately. In the closely related field of face recognition several large datasets are currently available such as the Labeled Faces in the Wild (LFW) [1] and YouTube Faces [2] datasets. Several tremendous steps have been taken towards verification performances that surpass human accuracy rates. Recently, the Adience collection was made openly available particularly for age and gender estimation [3]. In [3], Eidinger et al. pointed out that gender classification on the Adience dataset is considerably more difficult than other datasets for gender classification such as the Gallagher images of groups collection [4].

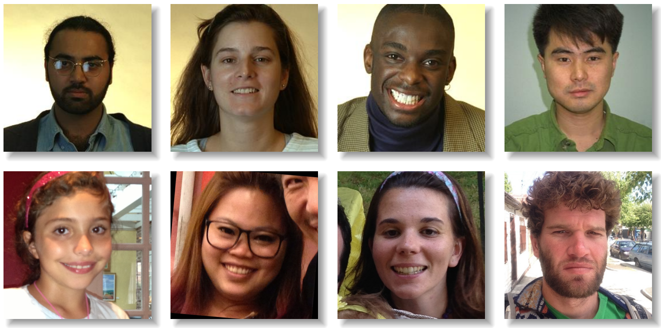

Fig. 1. An impression of the challenging task of face gender classification. Top row: face images from the color FERET dataset [5], [6]. Bottom row: face images from the Adience dataset [3]

The contributions of this paper. In order to investigate the applicability of current machine learning algorithms to face gender classification we propose a hybrid machine learning system. The proposed approach is based on combining a pretrained convolutional neural network (CNN) [7] with a linear support vector machine (SVM) [8]. A major motivation for using CNNs for gender recognition is their impressive success that was extensively demonstrated for face recognition and verification [9]–[13]. Usually, SVM classifiers are used to perform gender estimation on images by feeding them with various image descriptors (e.g. [3]). Recently it was shown that deep features of CNN models carry abstract representations of image contents [14]. Following the approach by [14], the SVM was trained on the deep features of a CNN that is based on the approach formulated by Krizhevsky et al. [15]. In addition, we adopt the dropout-SVM as proposed in [3] to avoid over-fitting. In an attempt to improve the classification rates of the SVM classifiers the CNNs were fine-tuned. Fine-tuning a pretrained CNN has been shown to exhibit high performance with a relatively short training time [14], [16].

Novelty of this paper. The proposed implementation is tested on the color FERET [5], [6] and the Adience [3] data collections (see Fig. I for an impression). Furthermore, we adopt a new partitioning of train and test data of the color FERET dataset. In our partitioning we have included faces from all angles. To the best of our knowledge, we are the first to partition the images in this particular way and test on all possible angles. In order to explore generalization beyond a single dataset we also provide cross-dataset classification rates, which have not been reported for these two datasets

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.37

before. Both datasets are relatively small when compared to popular computer vision benchmarks such as the LFW [1] and the YouTube Faces datasets [2]. This enabled us to feasibly optimize the SVM classifiers. We report the result of hyperparameter determination. In the proposed system C-SVM is considered for which the regularization parameter C is determined using cross-validation. We extend the common hyperparameter search to a combined search for both C and the dropout rate p_{drop}. Finally, we compare the performance of fine-tuned networks against using their deep features together with an SVM.

Outline. Section II discusses other work that is related to our approach. Section III provides a detailed description of our gender classification system. Section IV addresses the experiments that were conducted and their results. Finally, section V provides a conclusion and a discussion of our findings.

II. Previous work

This section addresses recent developments in various related areas of computer vision. Lately, state-of-the-art performance is often achieved using CNN architectures [13], [17]. Hence, this section provides a brief outline of previous work that is related to this approach. This section is concluded with a brief overview of other efforts on face gender classification.

A. *Convolutional neural networks*

Since LeCun et al. introduced CNN models in [7], significant steps have been made towards robust optimization. Nowadays large image datasets are used thanks to millions of digital images available online. Fast optimization can be achieved using modern day hardware and GPU accelerated programs. Recent studies on image classification demonstrate impressive results on highly challenging benchmarks. Some machine learning applications approach human performance, particularly using deep CNNs [9]–[13]. Recent efforts on deep CNNs for face verification tasks have even surpassed human performance [10], [13]. The face verification problem is to determine whether two queried faces are of the same individual. The majority of recent face recognition systems are benchmarked using the LFW dataset [1].

Deep CNN models have been shown to perform well despite strong variance in pose, lighting and expressions. The robust properties of CNNs and their major success in the closely related task of face verification make them a plausible candidate for a face gender classification system.

B. *CNN features as generic image descriptors*

In [14] the features from the pretrained `OverFeat` network [18] were extracted from a hidden layer at the end of the network containing 4096 units. In [14] Razavian et al. demonstrated that the deep hidden units can be used as generic image descriptors. After manipulating the extracted features by conducting PCA and component-wise transformations, linear SVM classifiers were trained on these features to tackle various image recognition problems including object classification, scene classification, object detection and others. Razavian et al. showed that the highest performance is attained when using the deepest feature layers.

In [14] no further model fine-tuning was performed. In our research the effect of fine-tuning on the resulting classification performance is also considered. The idea of fine-tuning a deep CNN will be discussed next.

C. *Fine-tuning*

In [15] a CNN was trained for the Imagenet Large Scale Visual Recognition Challenge (ILSVRC) competition [19]. However, training a CNN is a lengthy process and requires a large amount of data. Even when using multiple high-end GPUs training a network for the ILSVRC competition can take about 2-3 weeks depending on the network size [20]. Model fine-tuning avoids the need for large amounts of data, yet it is reported to achieve state-of-the-art performance [16], [21], [22]. In the fine-tuning procedure a pretrained network is used for a more domain-specific task. The pretrained weights are a useful initialization state for the network. In [21] and [22] a multistage training procedure was adopted where the network was first configured using unsupervised learning followed by domain-specific fine-tuning using supervised learning.

In [15] Krizhevsky et al. constructed a CNN to classify 1.3 million images in the Imagenet Large Scale Visual Recognition Challenge (ILSVRC). The network consists of multiple convolutional layers, max pooling layers and uses Rectified Linear Units (ReLUs). Although the task for ILSVRC is considerably different, the low-level kernels that were trained presumably serve as a good initialization point for a pretrained network devoted to gender classification.

In our approach a pretrained network based on [15] was adopted for implementing the gender classification systems. The CNN architecture used here is based on the submission of [15] for the ILSVRC competition. There are some subtle differences which will be described further on. After initialization of the pretrained weights the model was fine-tuned for gender estimation. We compare the performance of linear SVM classifiers on the pretrained model and the fine-tuned model. We also report the performance of the fine-tuned network alone.

D. *Preventing overfitting*

In general, training large models naively can quickly lead to overfitting which makes models only useful for the particular dataset they were trained for. This section briefly discusses two common methods to reduce the risk of overfitting.

Krizhevsky et al. and many others have demonstrated that data-augmentation can significantly improve classification performance by avoiding overfitting [15]. Data augmentation is a way of increasing the size of a dataset. Moreover, it increases the inner-class variety. Data augmentation during the training phase can be achieved by using label preserving transformations, such as mirroring, random cropping and altering RGB intensities [15], [23]. These transformations do not alter the gender label that belongs to the image.

A second method for avoiding overfitting is the relatively new dropout procedure [24], [25]. Dropout is commonly used for training deep CNNs where it is applied to fully connected layers that are often located at the end of the network. The activation of the neurons within the network are randomly set to zero when processing a training instance with a probability of $p_{\text{drop}} = 0.5$. This prevents hidden units from complex co-adaptations that might ultimately lead to overfitting. Another perspective explains the success of dropout training by considering every iteration as training a different model that consists of only the active units. At test time all units are active simultaneously. By readjusting the weights of the model the prediction mimics the averaged prediction of all possible models together.

In [3] Eidinger et al. proposed a method for training SVM classifiers with dropout. Input units were dropped out by setting the features of the descriptors randomly to zero. Using multiple datasets they demonstrated that dropout effectively improves classification rates. Moreover, they showed that by choosing a dropout rate of 0.8 and presenting each training instance twice, the system outperforms the implementation with using a dropout rate of 0.5. In this paper, the potential of model optimization using dropout is further explored by performing a grid search on the dropout rate p_{drop} and regularization parameter C for the linear SVM.

E. Gender classification

In [26] a CNN was presented using shunting inhibitory neurons which are further discussed in [27]. This particular CNN has only 3 layers and is not comparable with recent CNN architectures. The network was used to perform gender classification on solely frontal face images from the color FERET dataset [5], [6]. Tivive et al. achieved a classification rate of 97.1% on these images after selecting only the frontal face images [26]. In [28] the related gray FERET dataset was addressed where near-perfect classification rates are achieved using Weber's Local Descriptors (WLDs) [29]. Again, in [28] only frontal face images were considered. Moreover, the train and test sets were not configured to be subject-exclusive (i.e. subjects were mixed as they occurred in both the train and test sets). This could have led to gender estimation that is based on a person's identity instead of the gender-dependent attributes. In [30] a subject-exclusive protocol was compared to a mixed protocol. Their results confirmed that the mixed partitioning makes classification considerably more challenging. We adopt our own subject-exclusive partitioning on the color FERET dataset of train and test data which is used to cross-validate algorithm performances.

More recently, Eidinger et al. offered the publicly available Adience dataset particularly intended for age and gender estimation [3]. It is notably more unconstrained than the color FERET dataset and the difficulty is comparable to the LFW or YouTube Faces datasets. This dataset was presented along with their approach to dropout-SVM. In [3] the SVMs were trained with local binary patterns (LBP) and Four Path LBP codes (FPLBP) with which an accuracy of 76.1% was obtained. Compared to CNN features these descriptors are considerably more efficient in terms of representation and time. In [31] the current state-of-the-art for gender classification was achieved on the Adience dataset using a CNN. The CNN architecture provided by [31] is shallower than our architecture. Levi et al. used smaller fully connected layers and less convolutional layers compared to [15], thereby reducing the number of parameters and possibly avoiding the risk of overfitting. As opposed to our approach, their CNN was not pretrained. They reported an average accuracy of 86.1%. We compare our test results with their test results on the Adience dataset.

TABLE I. Detailed architecture that was used for feature extraction and fine-tuning. The architecture is included in the Caffe toolbox [32] and was based on [15]. The original fc8 has been replaced by a two node layer with hinge loss.

Name	Type	Output dimensions	Description
data	Input	$3 \times 227 \times 227$	Mirroring, random crops
conv1	Convolution	$96 \times 55 \times 55$	96 kernels of size 11, stride 4, ReLU
pool1	Max pooling	$96 \times 27 \times 27$	96 kernels of size 3, stride 2, LRN
conv2	Convolution	$256 \times 27 \times 27$	256 kernels of size 5, pad 5, group 2, ReLU
pool2	Max pooling	$256 \times 13 \times 13$	Pool size 3, stride 2, LRN
conv3	Convolution	$384 \times 13 \times 13$	384 kernels of size 3, stride 1, pad 1, group 2, ReLU
conv4	Convolution	$384 \times 13 \times 13$	384 kernels of size 3, stride 1, pad 1, group 2, ReLU
conv5	Convolution	$256 \times 13 \times 13$	256 kernels of size 3, stride 1, pad 1, group 2, ReLU
pool5	Max pooling	$256 \times 6 \times 6$	Pool size 3, stride 2
fc6	Fully connected	$4096 \times 1 \times 1$	ReLU, Dropout
fc7	Fully connected	$4096 \times 1 \times 1$	ReLU, Dropout
fc8	Fully connected	$2 \times 1 \times 1$	Classification with hinge loss

III. Gender classification system

This section elaborates on the details of our approach to gender classification. First we will describe the deep CNN architecture from which features were extracted. Then we briefly explain the use of dropout-SVM and we report the training and test configurations.

A. The CNN architecture

The pretrained network was deployed using the C++ `Caffe` toolbox [32]. This toolbox can be used as a framework for convolutional feature embedding and offers a wide range of layer configurations, some trivial data augmentation techniques and supports NVIDIA GPU acceleration. The pretrained weights are hosted online[1]. The model is referred to as the `BVLC CaffeNet` which was made available for unrestricted use. The model's architecture is based on the architecture proposed by Krizhevsky et al. [15]. The network differs slightly from [15]: During training no relighting data-augmentation was used and the order of pooling and normalization layers is switched.

For our fine-tuning network the final 1000 unit softmax layer with a cross-entropy loss was replaced by a 2 unit layer with a hinge loss function using an $L2$ norm. The reason to favor a hinge loss over a softmax loss is arbitrary, since

[1] http://dl.caffe.berkeleyvision.org/bvlc_reference_caffenet.caffemodel

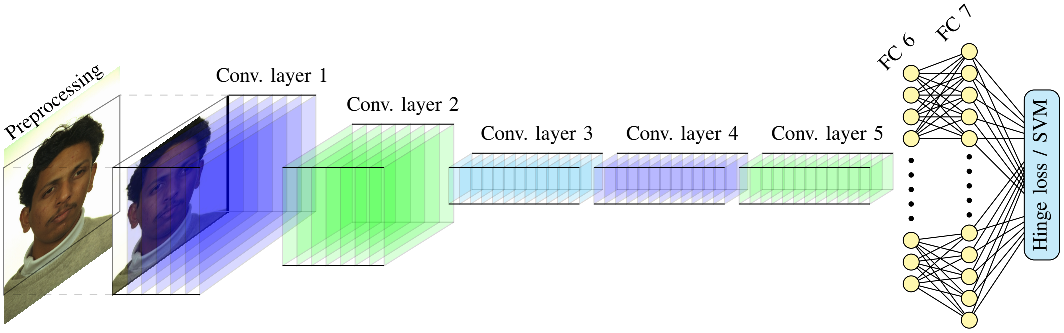

Fig. 2. An illustration of the general model architecture. Preprocessing consists of detecting and cropping the face. The data obtained after detection and cropping is augmented using label preserving transformations. The augmented data is processed by a CNN. The deep features of the CNN are then used for gender classification using an SVM. A more detailed description of the deep CNN structure is given in Table I.

both loss functions tend to perform comparably in practice [33]. However, in [34] it is reported that the squared hinge loss outperforms the cross-entropy loss. Using Caffe we regularize the weights by defining the loss function as follows:

$$\min_{\mathbf{w}} \lambda \mathbf{w}^T \mathbf{w} + \frac{1}{N} \sum_{n=1}^{N} \max(1 - \mathbf{w}^T \mathbf{x}_n t_n, 0)^2 \tag{1}$$

Where

$$t_n = \begin{cases} 1 & \text{if class } n \text{ is the ground truth} \quad \text{(2a)} \\ -1 & \text{otherwise} \quad \text{(2b)} \end{cases}$$

And where N is the number of examples and λ is the weight decay term.

Table I provides a detailed overview of the CNN architecture that we adopted for fine-tuning and feature extraction. At every convolutional layer and fc6 and fc7 a Rectified Linear Unit (ReLU) was used for activation. The ReLU nonlinearity is given by the activation function $f(x) = \max(0, x)$ [35]. After performing max pooling at pool1 and pool2 the activations are locally normalized using Local Response Normalization (LRN) as proposed by Krizhevsky et al [15]. This network also uses parameter grouped convolution. A grouped convolution with two groups implies that the first half of the respective preceding layer is only connected to the first half of the current layer, while the second half of the former is only connected to the second part of the latter.

B. SVM classifiers

In our approach we used the activations at layer fc7 (see Table I) as generic image descriptors. After the features were extracted we used linear SVM classifiers to predict the person's gender. The SVM with linear kernel was trained using the LIBLINEAR library [36]. The SVM classifiers were also trained to minimize the squared hinge loss with a regularization term. In LIBLINEAR the loss function is defined by the following:

$$\min_{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^{N} \max(1 - \mathbf{w}^T \mathbf{x}_n t_n, 0)^2 \tag{3}$$

where $\mathbf{x}_n$ is now the n-th feature vector obtained from the CNN. This loss function also involves a hyperparameter C which determines the penalty for a misclassification. Note that this equation is equivalent to (1). The primal solver from LIBLINEAR was used for optimization.

The features are taken from the last fully connected layer similar to the approach in [14]. Every feature was scaled between 0 and 1 to avoid numerical difficulties and to balance the importance of every feature [37]. The hyperparameter C from equation 3 was determined using cross-validation. The full model architecture is illustrated in Fig. 2.

It is important to stress that the training of these SVM classifiers is, apart from the optimization algorithm, essentially the same as training the fc8 layer of the CNN. However, no feedback was back-propagated through the network as opposed to the fine-tuning stage.

C. Data augmentation and dropout training

In order to reduce overfitting, the training data were augmented using label-preserving transformations [23]. Similar to [15], the data were augmented by taking random crops from the square face images. The resulting 227×227 sub-image was horizontally mirrored or not with a 50% chance. By using both augmentations the theoretical size of the training set increases by a factor of 1682. However, the majority of the augmented instances will be too similar to improve the classification rate significantly. Therefore, every image was augmented only eight times for training the SVMs. Data augmentation has been shown to be effective for both training a CNN [15] and for training SVMs on their deep feature layers [14].

Another measure to prevent overfitting is to use the computationally inexpensive dropout technique [15]. During the fine tuning phase the dropout rate within the CNN was set to $p_{\text{drop}} = 0.5$. Dropout training has been shown to boost the classification accuracy of at least linear SVMs significantly [3], [38]. Here we adopt the approach proposed by [3]. First we drop a random subset of the features. A dropout rate of $p_{\text{drop}} = 0.5$ means that half of all features are set to zero. The learned weights are divided by $(1 - p_{\text{drop}})$ to compensate for this effect. We considered the effect of the dropout rate on the classification rate since in [3] it is demonstrated that this can affect performance substantially.

While testing, there is no dropout and so the network uses all of its connections simultaneously. No data augmentation was used for the test set.

D. Train and test configurations

For fine-tuning the network the weights W to the final classification units are initialized using a Gaussian $W \sim \mathcal{N}(0, 0.0001)$. The bias terms are filled with zeros. All other weights are initialized from the pretrained model. The network was fine-tuned using stochastic gradient descent including a momentum term [39]. Given a loss function $L(W)$ to minimize, the weight updates V are determined by:

$$V_{t+1} = \mu V_t - \alpha \Delta L(W_t) \tag{4}$$

$$W_{t+1} = W_t + V_{t+1} \tag{5}$$

where $\Delta L(W_t)$ is the gradient of the loss function, μ is the momentum coefficient and α is the local learning rate. Here μ was set to 0.9. The learning rate is initialized at 0.0001 and it is then lowered by a factor of 0.1 after a certain amount of iterations, depending on the dataset that was used. For the newly added layer (fc8) the global learning rate was multiplied by 10 and by 20 for the weights and biases respectively. The learning rate is locally higher since this layer has not been trained for any task before. All other layers use the global learning rate as their local learning rate for training the weights and twice the global rate as the rate for the biases. Optimization was conducted using 8 images per batch. The main reason for using this batch size is that of hardware constraints.

At test time the center 227×227 crops of the original 256×256 images were extracted to predict the gender. In an attempt to improve the prediction performance the test data were augmented by conducting an over-sampling procedure. This was accomplished by extracting five crops (one at each corner and one at the center) and their mirrored versions and feeding them to the network. The class scores were then averaged to obtain the final prediction.

IV. Experiments

In order to test our proposed models on gender classification we measured classification performances on the color FERET [5], [6] and the Adience benchmarks [3]. To explore generalization beyond a particular dataset we also tested cross-dataset classifications. The results are reported in section IV-D.

A. The color FERET benchmark

The color FERET benchmark [5], [6] contains face images of 591 men and 403 women. The images were collected by photographing the subjects at 13 different angles. The face often covers only a small part of the image. Therefore, the images were preprocessed to extract the face in a square sub image. The face detection algorithm from `dlib` [40] was adopted to detect and crop the faces from the images. The face detection is implemented using a 68 face landmark predictor as proposed by Kazemi et al. [41]. The bounding box of the face was enlarged by 150% to also capture the hair and possibly other relevant content. If the algorithm could not detect a face, the image was omitted from the dataset. This procedure resulted in discarding the majority of side-views of faces (90 degree angle). After detection and cropping, the dataset consisted of 8364 face images stored in 256×256 jpeg files. No further alignment was applied, since the faces in this dataset are only rotated about the yaw axis.

The data was first partitioned into a training set and a test set using a random permutation of the subjects. The training set conisists of 6073 images, which are 80% of the data. The test set consists of 2291 images, which are the remaining 20%. We have adopted our own partitioning by including non-frontal angles of face images as opposed to many others [26], [28], [29]. It is important to note that in our partitioning a single individual is not both in the train set and the test set even though the photographs were taken form different angles (unmixed partitioning). This separation is to prevent gender classification that is based on a person's identity. The training set was cross-validated tenfold to find C and p_{drop} and to compare classification performances among the models. This means the validation sets within the train data contained either 607 or 608 images each. After cross-validation the final models are trained using the selected C and p_{drop}, after which they are tested on the 20% of the data that were isolated from the parameter search.

B. The Adience benchmark

The Adience benchmark was originally constructed for gender and age classification [3]. Images were collected from Flickr uploads mainly from smart-phone devices. The conditions under which these images were taken more closely resemble real-world challenges like the LFW [1] or Youtube faces [2] collections. Faces from this dataset are often highly occluded by e.g. heavy make-up or by being covered partly by hands. Another challenging factor is that there are many photos of babies where gender dependent attributes are not clearly visible yet.

The Adience dataset contains about 26K images of 2284 subjects. For testing our models we have adopted the partitioning protocol as proposed by Eidinger et al. [3]. We report results on gender classification using their subject-exclusive partitioning for five-fold cross validation. In this protocol, the average result of the cross-validation procedure will be the resulting measure of performance. Hence, we performed 5 tests for each model configuration. To compare our classification method against [3] and [31] we used the aligned and cropped versions of the face images that were also included in the dataset. During the alignment and cropping some images were discarded because of alignment failure, which is why there are 17492 images left. The five test sets contain 3995, 3609, 3137, 3306 and 3445 images, respectively. Again, we determined C and p_{drop} using cross-validation.

C. Implementation details

For training the deep CNN we used the popular `Caffe` toolbox [32]. Training was conducted on a NVIDIA GeForce GT 540M GPU with 96 CUDA cores and 1 GB of DDR3 memory. Training a single deep CNN on a train and validation pair required about 1 to 4 hours depending on when the classification rates saturated for different datasets.

D. Results

While conducting a grid search for C and p_{drop} we observed that the classification rates were slightly higher if we set the dropout ratio of fc6 (see Table I) to zero while extracting the features. This was observed for both datasets. We will not

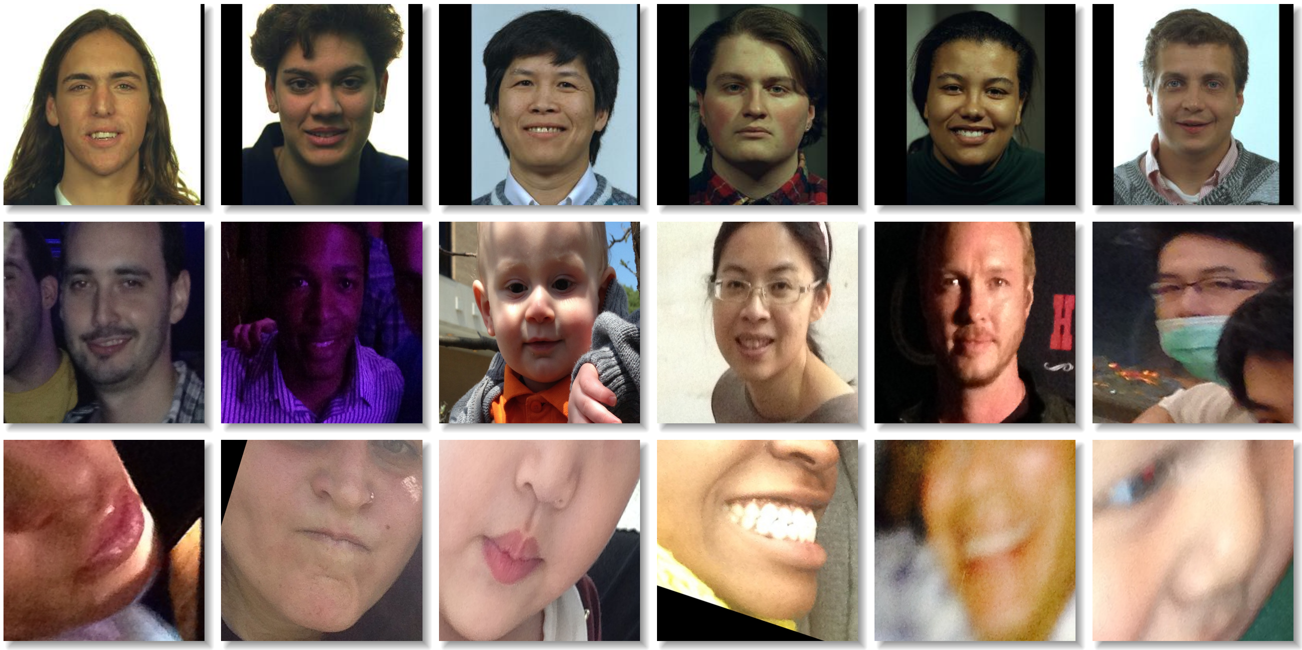

Fig. 3. Misclassifications of the color FERET benchmark (top row) and the Adience benchmark (middle row and bottom row). These images were misclassified while testing the fine tuned networks with using oversampling. The bottom row shows misclassified images where the face is aligned improperly.

discuss the comparison between this dropout rate and other configurations for briefness. The results of the SVM systems that follow were obtained with the dropout ratio of fc6 set to zero. Only the dropout ratio at fc7 was optimized for of the systems with SVMs.

1) The color FERET benchmark: Table II shows classification results on the color FERET dataset. Fine tuning was stopped after 3500 iterations, since the classification rates appeared to saturate at that stage.

For all SVM classifiers C and p_{drop} have been determined using cross-validation. For the SVM that was trained on the fine-tuned network $C = 128$ and $p_{\text{drop}} = 0.5$ were obtained. The SVM that was trained on the pretrained network performed best with $C = 0.011049$ and $p_{\text{drop}} = 0$. From Table II it is evident that fine-tuning the network substantially improves classification performance. In addition, it can be observed that performance benefits from using the oversampling procedure. No further improvement is obtained when using the SVM after fine-tuning the network. This was to be expected, since both algorithms have an equivalent cost function.

TABLE II. CLASSIFICATION SCORES ON THE COLOR FERET DATASET USING OUR UNMIXED PARTITIONING. MEAN ACCURACY (± STANDARD ERRORS).

Method	Accuracy (%)
CNN + dropout-SVM	94.8 ± 0.6
CNN + dropout-SVM + oversampling	95.8 ± 0.6
CNN + Fine tuning	96.9 ± 0.5
CNN + Fine tuning + oversampling	$\mathbf{97.3 \pm 0.5}$
CNN + Fine tuning + dropout-SVM	96.2 ± 0.8
CNN + Fine tuning + dropout-SVM + oversampling	96.6 ± 0.8

TABLE III. CLASSIFICATION SCORES ON THE ADIENCE DATASET USING AN UNMIXED PARTITIONING AS PROPOSED IN [3]. MEAN ACCURACY (± STANDARD ERRORS).

Method	Accuracy (%)
Best from [3]	76.1 ± 0.9
Best from [42]	79.3 ± 0.8
Best from [31]	86.8 ± 1.4
CNN + dropout-SVM	80.2 ± 1.2
CNN + dropout-SVM + oversampling	81.4 ± 1.3
CNN + Fine tuning	86.2 ± 0.7
CNN + Fine tuning + oversampling	$\mathbf{87.2 \pm 0.7}$
CNN + Fine tuning + dropout-SVM	86.2 ± 0.8
CNN + Fine tuning + oversampling + dropout-SVM	87.1 ± 0.7

2) The Adience benchmark: Table III depicts the performance of our methods against two methods using dropout-SVM classifiers [3], [42] or a deep CNN [31]. It can be observed that our best results are somewhat better than what was reported by [31]. In this case the classification rates saturated after 20.000 iterations of fine-tuning.

For the fine-tuned network $C = 0.003906$ and $p_{\text{drop}} = 0.4$ were obtained. The pretrained network performed best when $C = 0.001953$ and $p_{\text{drop}} = 0$. Again, our results confirm that oversampling effectively improves classification rates. Reasonable results were already obtained when using the pretrained network without any fine-tuning. However, the best results were obtained by using the fine-tuned network. For the same reasons as before, there are no performance gains observed when using the SVM after fine-tuning the network.

Fig. 3 shows misclassifications by the fine-tuned networks that were trained on each of the datasets. The faces that are misclassified typically include features of the opposite sex. This is especially true for the misclassifications among the color FERET collection. Men with long hair are often classified as women. The middle row of Fig. 3 suggests that the classifiers still have trouble with faces where gender dependent attributes are not clearly visible, such as the baby. However, some misclassified images still appear trivial to the human observer.

The bottom row of Fig. 3 depicts cases of misalignment. Since alignment and cropping were conducted automatically, these images remained in the dataset since it was released by [3].

3) Cross-dataset results: Cross-dataset experiments were conducted to explore the generalization capabilities of the models. Here the CNNs were fine-tuned once more on each dataset while including all data within the set simultaneously. We trained one model on the color FERET dataset and tested it on the Adience dataset and vice-versa.

The results are depicted in Table IV. It can be observed that there is a considerable drop in accuracy. This might be caused by the fact that only the faces in the Adience dataset were aligned, while the faces in the color FERET dataset were only cropped. Consequently, the average appearance of faces from a side-angle differed considerably between both datasets. Second, the model that was trained on the color FERET dataset might have been subject to overfitting, since the pictures were taken under constrained conditions, whereas the images from the Adience dataset are taken in unconstrained conditions.

TABLE IV. CROSS-DATASET RESULTS. THE CROSS-DATASET RESULTS WERE OBTAINED FROM TRAINING AND TESTING ON ALL AVAILABLE DATA.

		Train set	
		Adience	color FERET
Test set	Adience	87.2	67.1
	color FERET	83.7	97.3

V. Conclusions

In this paper we explored the applicability of deep convolutional neural networks on face gender recognition. We showed that despite the challenging nature of the problem, state-of-the-art classification rates can be achieved using relatively short training times. On both datasets, the best results were obtained when using the fine-tuned networks. Oversampling by averaging class scores of the final classifiers was shown to improve classification rates in all cases.

For future work, results can possibly be improved even further by excluding the extreme cases of misalignment (bottom row Fig. 3) from the train and test phases of our experiments. Furthermore, classification could benefit from frontalization where faces from a non-frontal angle are frontalized with the help of 3D modeling of the face [12], [42]. Further performance gain could be achieved by using other network architectures. Since the ILSVRC 2012 submission of Krizhevsky et al. [15] others have submitted models that exhibit superior results [11], [17]. In addition, ensembles of pretrained and fine-tuned CNNs could be explored as well. Instead of merely averaging the class scores when performing oversampling, the system might be enhanced by training a classifier that linearly combines the class scores for each crop and mirroring.

The results obtained here suggest that there is much left to explore within applications of convolutional neural networks. The large availability of image data nowadays and the major successes with deep CNN systems can push machine learning systems further towards human-level recognition and beyond.

References

[1] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[2] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 529–534.

[3] E. Eidinger, R. Enbar, and T. Hassner, "Age and gender estimation of unfiltered faces," *Information Forensics and Security, IEEE Transactions on*, vol. 9, no. 12, pp. 2170–2179, Dec 2014.

[4] A. Gallagher and T. Chen, "Understanding images of groups of people," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 256–263.

[5] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and vision computing*, vol. 16, no. 5, pp. 295–306, 1998.

[6] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 10, pp. 1090–1104, 2000.

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018

[9] C. Lu and X. Tang, "Surpassing human-level face verification performance on LFW with GaussianFace," *CoRR*, vol. abs/1404.3840, 2014. [Online]. Available: http://arxiv.org/abs/1404.3840

[10] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 1891–1898.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[12] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 1701–1708.

[13] E. Zhou, Z. Cao, and Q. Yin, "Naive-deep face recognition: Touching the limit of LFW benchmark or not?" *CoRR*, vol. abs/1501.04690, 2015. [Online]. Available: http://arxiv.org/abs/1501.04690

[14] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, June 2014, pp. 512–519.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[16] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," *CoRR*, vol. abs/1411.7766, 2014. [Online]. Available: http://arxiv.org/abs/1411.7766

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," *CoRR*, vol. abs/1502.01852, 2015. [Online]. Available: http://arxiv.org/abs/1502.01852

[18] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” *CoRR*, vol. abs/1312.6229, 2013. [Online]. Available: http://arxiv.org/abs/1312.6229

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, 2015.

[20] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 580–587.

[22] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, “Pedestrian detection with unsupervised multi-stage feature learning,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 3626–3633.

[23] P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” in *2013 12th International Conference on Document Analysis and Recognition*, vol. 2. IEEE Computer Society, 2003, pp. 958–958.

[24] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *CoRR*, vol. abs/1207.0580, 2012. [Online]. Available: http://arxiv.org/abs/1207.0580

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2670313

[26] F. H. C. Tivive and A. Bouzerdoum, “A shunting inhibitory convolutional neural network for gender classification,” in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 4, 2006, pp. 421–424.

[27] F. Tivive and A. Bouzerdoum, “Efficient training algorithms for a class of shunting inhibitory convolutional neural networks,” *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 541–556, May 2005.

[28] I. Ullah, M. Hussain, G. Muhammad, H. Aboalsamh, G. Bebis, and A. Mirza, “Gender recognition from face images with local WLD descriptor,” in *Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on*, April 2012, pp. 417–420.

[29] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao, “WLD: A robust local image descriptor,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1705–1720, Sept 2010.

[30] S. Baluja and H. Rowley, “Boosting sex identification performance,” *International Journal of Computer Vision*, vol. 71, no. 1, pp. 111–119, 2007. [Online]. Available: http://dx.doi.org/10.1007/s11263-006-8910-9

[31] G. Levi and T. Hassner, “Age and gender classification using convolutional neural networks,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) workshops*, June 2015. [Online]. Available: http://www.openu.ac.il/home/hassner/projects/cnn_agegender

[32] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *arXiv preprint arXiv:1408.5093*, 2014.

[33] L. Rosasco, E. Vito, A. Caponnetto, M. Piana, and A. Verri, “Are loss functions all the same?” *Neural Computation*, vol. 16, no. 5, pp. 1063–1076, 2004.

[34] Y. Tang, “Deep learning using support vector machines,” *CoRR*, vol. abs/1306.0239, 2013. [Online]. Available: http://arxiv.org/abs/1306.0239

[35] V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[36] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “LIBLINEAR: A library for large linear classification,” *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[37] A. B. A. Graf and S. Borer, “Normalization in support vector machines,” in *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*. London, UK, UK: Springer-Verlag, 2001, pp. 277–282. [Online]. Available: http://dl.acm.org/citation.cfm?id=648286.756268

[38] N. Chen, J. Zhu, J. Chen, and B. Zhang, “Dropout training for support vector machines,” *CoRR*, vol. abs/1404.4171, 2014. [Online]. Available: http://arxiv.org/abs/1404.4171

[39] L. Bottou, “Stochastic gradient descent tricks,” in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, G. Montavon, G. Orr, and K.-R. Müller, Eds. Springer Berlin Heidelberg, 2012, vol. 7700, pp. 421–436. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-35289-8_25

[40] D. E. King, “Dlib-ml: A machine learning toolkit,” *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[41] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 1867–1874.

[42] T. Hassner, S. Harel, E. Paz, and R. Enbar, “Effective face frontalization in unconstrained images,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2015. [Online]. Available: http://www.openu.ac.il/home/hassner/projects/frontalize