Local Deep Neural Networks for gender recognition

Jordi Mansanet[a], Alberto Albiol[a], Roberto Paredes[b]

[a]ITEAM, Universitat Politècnica de València
[b]PRHLT Research Centre, Universitat Politècnica de València

Abstract

Deep Learning methods are able to automatically discover better representations of the data to improve the performance of the classifiers. However, in computer vision tasks, such us the gender recognition problem, sometimes it is difficult to directly learn from the entire image. In this work we propose a new model called Local Deep Neural Network (Local-DNN), which is based on two key concepts: local features and deep architectures. The model learns from small overlapping regions in the visual field using discriminative feed-forward networks with several layers. We evaluate our approach on two well-known gender benchmarks, showing that our Local-DNN outperforms other deep learning methods also evaluated and obtains state-of-the-art results in both benchmarks.

1. Introduction

Gender recognition of face images is an important task in computer vision as many applications depend on the correct gender assessment. Examples of these applications include visual surveillance, marketing, intelligent user interfaces, demographic studies, etc. The gender recognition problem is usually divided into several steps, similarly to other classification problems (Ng et al. (2012b)): object detection, preprocessing, feature extraction and classification. In the detection phase, the face region is detected and cropped from the image. Then, a preprocessing technique is used to reduce variations in scale and illumination. After this normalization, the feature extraction step aims at obtaining representative and discriminative descriptors of the face region. Finally, a binary classifier that learns the differences between male and female representations is trained.

Perhaps, feature extraction is the most critical step in order to achieve good performance. Traditionally, features have come up as a result of the knowledge and expertise of many feature practitioners. However, instead of relying on this human-based process to define the best representation of the data in a specific problem, it would be much more interesting to let the algorithm to discover that representation automatically by itself. For this reason, representation learning has emerged as a promising research field (LeCun et al. (2015)). The main goal of representation learning is to automatically convert data into a form that makes it easier to extract useful information when building classifiers (Bengio et al. (2013)). Deep learning approaches are a particular kind of representation learning procedures that discover multiple levels of representations using neural networks, with higher-level features representing more abstract concepts of the data. These more abstract representations are closer to the semantic content of the data, so they are more useful than the raw data by itself to build classifiers. Also, it has been demonstrated that our brain works in the same way dealing with complex tasks like vision and language. The brain cortex extracts multiple levels of representation from the sensory input, doing progressively more complex processing tasks (Serre et al. (2007)).

These strategies have shown an excellent performance in challenging problems on the computer vision domain (Krizhevsky et al. (2012); Farabet et al. (2013); Taigman et al. (2014)). However, sometimes it is very difficult to directly learn from the entire image using standard Deep Neural Networks (DNN), specially with complex data like natural images (Krizhevsky (2009)). This problem have been tackled using the idea of getting information from sub-regions of the input image. For instance, unsupervised learning can extract useful features looking only at small zones of the images, called patches (Krizhevsky (2009)). A similar idea is used in Deep Convolutional Neural Networks (DCNN), where individual neurons are tiled in such a way that they respond only to overlapping regions in the input field.

Here, we propose a new model called Local Deep Neural Network (Local-DNN). Our model extracts several local features from the input images, and these features feed a discriminative deep neural network. The network learns to classify each local feature according to the label of the image to which it belongs. The final decision for the whole input image is taken based on a simple voting scheme that takes into account all the local contributions. We have found that for some specific applications, where some registration has been applied to the images, e.g. a face detector, the Local-DNN has demonstrated to be superior to other techniques due to a greater robustness to small translations, occlusions and local distortions, see (Villegas et al. (2008)). In this paper, we apply the Local-DNN model to the gender recognition problem using face images. Nevertheless, it is important to note that the Local-DNN is devoted to deal with images with this kind of registration and where some prior knowledge can be applied in order to select the most informative parts of the images, a kind of saliency map, while DCNN are devoted to general problems without any kind of constraint, registration or prior knowledge of the saliency map.

In order to be able to draw relevant conclusions, several experiments have been carried out using two challenging and realistic face image databases called Labeled Faces in the Wild (LFW) (Huang et al. (2007)) and the so-called Gallagher's database (Gallagher and Chen (2009)), where the images were taken in unconstrained conditions. Our Local-DNN framework outperforms other deep learning methods evaluated in this work, such as standard DNNs and DCNNs, and also obtains state-of-the-art results on these databases.

The remainder of the paper is organized as follows. Section 2 describes the related work on gender recognition. Section 3 describes our Local-DNN framework and Section 4 describes the datasets used and the set of experiments carried out. The final section draws some conclusions about the work in this article.

2. Related work

Extracting a good representation of the data is perhaps the most critical step in most of the pattern recognition problems. Initial approaches for gender recognition used the geometric relations between facial landmarks as feature representation (Ng et al. (2012a)). However, these methods required a very accurate landmark detection and it was shown that quite relevant information was thrown away. For this reason, all recent approaches use appearance-based methods, which perform some kind of operation or transformation on the image pixels. Appearance methods can be holistic, when the whole face is used to extract features, or local, when information is extracted from local regions of the face.

The handcrafted features found in the literature for gender recognition can be as simple as the raw pixels (Moghaddam and Yang (2002)) or pixel differences (Baluja and Rowley (2007)). Sometimes, simple features are pooled together as in (Kumar et al. (2009)), where image intensities in RGB and HSV color spaces, edge magnitudes, and gradient directions were combined. More elaborated features include Haar-like wavelets (Shakhnarovich et al. (2002)), Local Binary Patterns (LBPs) (Shan (2012)) or Gabor wavelets (Leng and Wang (2008)). These features work well and are robust to small illumination and geometric transformations. However, they are based on the expertise of the researcher to find the best option for a given problem. For instance, in (Moeini and Moeini (2015)) this expertise is used to compensate pose changes using a 3D model of the face.

Feature representations of the face are usually high-dimensional, and it is common to apply dimensionality reduction techniques. In Villegas and Paredes (2011) the authors show a good comparison of different methods on a gender recognition problem among others. These techniques have been widely used because of their simplicity and effectiveness (Buchala et al. (2004); Graf and Wichmann (2002)). However, they might not capture relevant information to represent a face in the gender recognition problem.

After all these steps, the face representation obtained is fed into a classifier that learns a discriminative model using the labels of the samples. For instance, the AdaBoost and the SVM algorithms have been widely used in the literature (Baluja and Rowley (2007); Shan (2012);

Eidinger et al. (2014)). In this spirit, an excellent comparison of gender recognition techniques using different methods can be found in Dago-Casas et al. (2011).

Regarding deep learning techniques, unsupervised models, such as Restricted Boltzmann Machines (RBMs) (Smolensky (1986)), have been demonstrated to be useful as a way to pre-train these deep architectures (Hinton and Salakhutdinov (2006)). These models are able to automatically extract good features from unlabeled data that are useful in supervised tasks like the gender recognition problem (Mansanet et al. (2014)). On the other hand, DCNNs models has shown great performance in computer vision tasks by learning from small regions in the visual field, (Krizhevsky et al. (2012); Simonyan and Zisserman (2014)) and have been successfully used for face recognition (Taigman et al. (2014); Sun et al. (2014); Schroff et al. (2015)). Focusing on the gender recognition problem, a recently published work used a DCNN to estimate the gender and age attributes using real-world face images (Levi and Hassner (2015)).

3. Local Deep Neural Networks

3.1. Introduction

In this section, we aim to describe the details related to our Local-DNN model. On the one hand, we have used a formal probabilistic framework, introduced in Villegas et al. (2008), to model the local feature-based classification. This framework is general, but in this work, we particularize it for the problem at a hand, where the local features became simple windows extracted from the face image at different locations, called patches. Therefore, from here onwards, the terms *patch* and *local feature* will be used interchangeably. On the other hand, we introduce the idea of using deep networks that are able to learn how to classify each local feature according to its appearance. During testing, all the contributions are fused using a voting scheme.

3.2. Formal framework for local-based classification

We denote the class variable by $c = 1, \dots, C$ and the input pattern (image) by $\mathbf{x}$. Local features (patches) are extracted from the input pattern using some selection criterion. Let F denote the number of local features drawn from the input pattern $\mathbf{x}$. It is assumed that each local feature $\mathbf{x}^{[i]}$, $i = 1, \dots, F$, contains incomplete yet relevant information about the true class label of $\mathbf{x}$, and thus it makes sense to define a local class variable for it, $c_i \in \{1, \dots, C\}$.

In accordance with the above idea, the posterior probability for $\mathbf{x}$ to belong to class c is computed from a complete model including all the local features labels,

$$p(c \mid \mathbf{x}) = \sum_{c_1=1}^{C} \cdots \sum_{c_F=1}^{C} p(c, c_1, \dots, c_F \mid \mathbf{x}) \tag{1}$$

which is broken into two sub-models, the first one to predict local class posteriors (from $\mathbf{x}$ only) and then another to compute the global class posterior from them (and $\mathbf{x}$),

$$p(c, c_1, \dots, c_F \mid \mathbf{x}) = p(c_1, \dots, c_F \mid \mathbf{x})\, p(c \mid \mathbf{x}, c_1, \dots, c_F) \tag{2}$$

In order to develop a practical model for $p(c \mid \mathbf{x})$, the first submodel is simplified by assuming independence of local labels conditional to $\mathbf{x}$; that is, by application of a *naive Bayes* decomposition to it,

$$p(c_1, \dots, c_F \mid \mathbf{x}) := \prod_{i=1}^{F} p(c_i \mid \mathbf{x}^{[i]}) \tag{3}$$

where $\mathbf{x}^{[i]}$ denotes the part of $\mathbf{x}$ relevant to predict c_i; i.e. the ith image patch. This simplification is based on the strong assumption of local features independence. On the other hand, it yields a very simplified model. Similarly, the second submodel is simplified by assuming that the global label only depends on local labels,

$$p(c \mid \mathbf{x}, c_1, \dots, c_F) := p(c \mid c_1, \dots, c_F) \tag{4}$$

The above simplifications are clearly unrealistic, though they may be reasonable if each local feature can be reliably classified independently of each other. In such a case, we may further simplify the second submodel by letting each local feature i vote for c_i in accordance with a predefined *(feature) reliability weight* α:

$$p(c \mid c_1, \ldots, c_F) := \sum_{i=1}^{F} \alpha_i \, \delta(c_i, c) \tag{5}$$

where $\delta(\cdot, \cdot)$ is the *Kronecker delta* function; $\delta(c_i, c) = 1$ if $c_i = c$; zero otherwise. $0 \leq \alpha_i \leq 1$, $i = 1, \ldots, F$, and $\sum_i \alpha_i = 1$.

$$p(c \mid \mathbf{x}) := \sum_{i=1}^{F} \alpha_i \, p(c \mid \mathbf{x}^{[i]}) = \sum_{i=1}^{F} \alpha_i \, p_c^{[i]} \tag{6}$$

where $p_c^{[i]}$ is the probability of the feature $\mathbf{x}^{[i]}$ to predict the global class c associated to the image $\mathbf{x}$. This expression is a simple weighted average over all local class c posteriors, where each feature contributes to the final decision in accordance with a predefined weight α_i ($0 \leq \alpha_i \leq 1$, $i = 1, \ldots, F$, and $\sum_i \alpha_i = 1$). In the simplest case, we may consider all the local features equally important:

$$\alpha_1 := \alpha_2 := \cdots \alpha_F := \frac{1}{F} \tag{7}$$

but in general, α_i should encode the reliability or the discriminative power of each local feature, or at least some surrogate measure.

Finally, we use a Bayes decision rule to perform the final classification of the input image $\mathbf{x}$ by choosing a class with maximum weighted sum of local posteriors,

$$\mathbf{x} \rightarrow c\,(\mathbf{x}) = \underset{c}{argmax}\; p\,(c \mid \mathbf{x}) = \underset{c}{argmax} \sum_{i=1}^{F} \alpha_i \, p_c^{[i]} \tag{8}$$

Therefore, during testing, the global classification is defined by summing all the weighted local posteriors obtained from each local feature contribution. In this paper, we also studied a small modification to perform the final classification in which each local feature chooses a class according to its maximum local posterior. After that, the most voted class among all local features belonging to the same sample is selected as the final decision,

$$\mathbf{x} \rightarrow c\,(\mathbf{x}) = \underset{c}{argmax} \sum_{i=1}^{F} \delta(c, \underset{c'}{argmax}\; p_{c'}^{[i]}) \tag{9}$$

In this voting method α_i is not considered.

3.3. A local class-posterior estimator using DNN

The main problem of using local patches is that the decision boundaries of the classification problem are highly non-linear and multi-modal. In this particular problem the local class posteriors $p_c^{[i]}$ has to be estimated from parts of images that contain parts of faces. These parts lead to a highly multi-modal distribution where the modes are the different parts of the faces extracted from different placements. To deal with this kind of probability distribution we need non-linear and multi-modal estimators like the k-nearest neighbor used in the past. Therefore, a first simple option was to use a k-nearest neighbor estimator (Villegas et al. (2008)), which is very simple yet effective. However, the main problem of this estimator is that it scales poorly to large data sets where the memory requirements grow rapidly. Note that the k-nearest neighbor estimator is based on storing all the local features extracted from the training images, so the model is just the patches. Moreover, in test phase, a nearest neighbor search must be processed, and this is time consuming even using approximate strategies like the $(1 + \epsilon)$-nearest neighbor over kd-trees. Neural networks can deal with multimodal distributions (non-linear problems) while the size of the model is not strictly dependent on the size of the training corpus. Moreover in test phase the class-posterior estimations are very fast, just perform a forward operation through the network. This forward procedure can be done in batch taking profit of the numerical optimizations for matrix operations. For all these reasons we propose to use a DNN as an estimator of the local class posteriors $p_c^{[i]}$, which leads to the *Local-DNN* name for the here proposed approach. In our opinion, using a deep architecture might facilitate the learning of the complex mapping from the appearance of patches to classes.

A graphical representation of this Local-DNN model is shown in Figure 1. As it can be seen, several patches are extracted from the input image and then are fed into a DNN. The network is formed by an input layer and several fully connected hidden layers. An output layer with C *softmax* units, that represents the posterior probability of each local patch. Finally, local posteriors are fused at the end. During training, the network works at a patch level by learning to classify each patch with the label of the image that it belongs. During testing all the contributions from the patches extracted of the image are combined in order to classify this image with a final label.

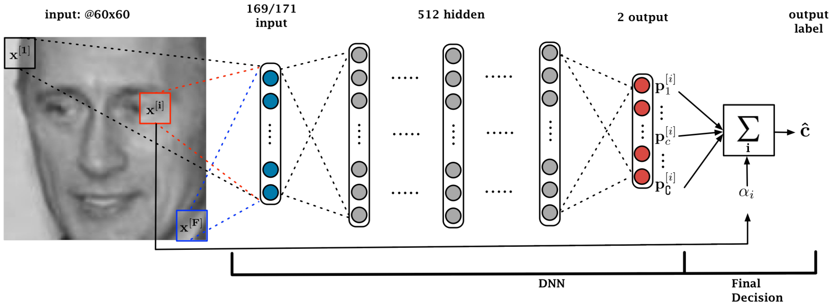

Figure 1: Graphical representation of our Local-DNN model. The parameters above denote the configuration used in our experiments.

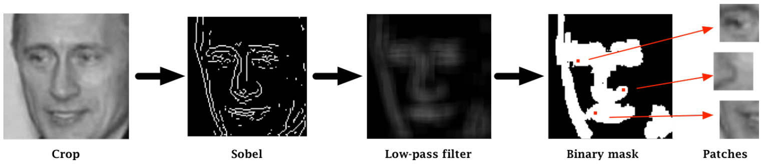

Figure 2: Graphical representation of the process to extract local patches.

3.4. Feature selection and extraction

The definition of a local feature working with images may include a huge variability of options, for instance just varying the size and the shape of the feature. However, we have only considered squared windows of size $w \times w$. Regarding the locations where to extract these patches, the simplest case is to use a fixed sampling grid for all the images. However, this selection leads to a computationally demanding learning due to the huge number of patches obtained. Therefore, we propose to select the patches with high information content, discarding those which associated $p(accuracy \mid \mathbf{x}^{[i]})$ is likely to be very low (for instance, uniform patches).

According to this procedure, given an image, we have to obtain a binary mask in which each active pixel denotes the center position of a patch to be extracted. The process to create this binary mask for each image is as follows. First of all, we create another image with a Sobel filter which emphasizes edges and translations. After that, we apply a low-pass filter over this image and the values are binarized using a threshold. Finally, we extract the patches centered in each active pixel in the binary mask, removing those that lie outside the image. Each patch is normalized to have zero-mean and unit-variance. The entire process is represented in the Figure 2. A similar method was proposed in Paredes et al. (2001) to extract only informative patches.

3.5. Location information and reliability weight

At this point we have defined the most important details of the Local-DNN model. Besides this base version of the model, we have evaluated two optional variations that might enhance its performance. These improvements are described bellow, and both of them have been evaluated in the experiments section.

On the one hand, the simplifications performed in the probabilistic framework to obtain the expression (6) are based on the strong assumption that the placement of each patch is not relevant for the final decision. This topological information might be useful to enhance the contribution of each local feature, as it happens in convolutional neural networks. To evaluate this phenomenon we propose a slight modification in which the key idea is that a local feature $\mathbf{x}^{[i]}$ should contain not only the content of the image patch itself, but also the location where it was extracted. In the context of DNNs, this information can be introduced in the input layer of the network by adding two extra units that encode the horizontal and vertical image coordinates of the center of the patch. This modification allows the network to use the location information together with the appearance of the patch to learn a more accurate discriminative function over the patches. It is important to stress that this improvement only affects the process to obtain the local posteriors, so the fusion step during testing remains unchanged.

An additional enhancement is related to the use of the reliability weights α_i during testing. In the basic version of the model, all the features selected by the binary mask are equally reliable when the final label is defined, with $\alpha_i = 1/F$, being F the number of local features considered in each image. The slight modification proposed is to compute α_i as $\hat{p}(accuracy \mid \mathbf{x}^{[i]})$, being $\hat{p}(accuracy \mid \mathbf{x}^{[i]})$ an empirical estimation of the accuracy of a particular patch $\mathbf{x}^{[i]}$. This estimation is computed by considering only the placement of the patch but discarding its content. Accordingly, the weight associated to a specific location will only depend on the mean classification accuracy of all the patches belonging to that location. These weights are estimated using the training patches once the DNN model has been trained. It is important to make it clear that this modification is independent from the previous one explained, despite both of them use the *location* of the image patch.

4. Experiments

4.1. Datasets

Two different datasets have been considered during the experiments:

Labeled Faces in the Wild

The Labeled Faces in the Wild (LFW) (Huang et al. (2007)) is composed by 13233 face images (10256 male and 2977 female) from 5749 celebrities collected from the web. The Facial Image Processing and Analysis group (FIPA, `http://fipa.cs.kit.edu`) proposed a benchmark to evaluate the gender recognition problem using these images. Following this protocol, we have employed the 5 folds described in FIPA (2011b), using 4 folds for training and 1 for testing, and the results were averaged. Similar articles in the bibliography have used a reduced version of this dataset, taking out a huge number of challenging images (Tapia and Perez (2013); Ren and Li (2014)). However, it is important to note that our experiments are performed using the entire dataset, without discarding any image.

Gallagher's DB

The Gallagher's database (Gallagher and Chen (2009)) contains 28231 face images taken from Flickr, and they were manually labeled with its gender and age group information. There is not an standard protocol for gender classification in Gallagher's DB, so we have used the same protocol proposed by Dago-Casas et al. (2011). On that article, a new version of the dataset was created by removing several low resolution face images. Also, they removed some of the male faces to obtain a final dataset of 14760 images evenly distributed. This image collection was divided in 5 folds, using 4 folds for training and 1 for testing, and the results were averaged. All the information of this protocol is available in FIPA (2011a).

4.2. Image normalization and patch extraction

Due to the unconstrained nature of the images, it is necessary a preprocessing step. First of all, we have used an aligned version of the LFW database created by Huang et al. (2012). In the case of the Gallagher's DB, we have used the location information of the eyes to transform each face to a canonical pose with the eyes located in the same position.

Once the face is aligned, we crop a face region of the image of 105×105 pixels and the cropped image is resized to 60×60 pixels. Each image is converted to grayscale and all the pixel values are scaled to the range [0,1]. After that, we create a binary masks for each image using the method described in Section 3.4, which indicates the locations where the patches are extracted from. This process is the same for both databases and allows to discard 32% and 41% of the available patches in the LFW and Gallagher's datasets respectively. It should be mentioned that due to the fact that there is a clear imbalance between genders in the LFW dataset, we have randomly discarded as many male patches as required to have an equally distributed training set for each fold. Obviously, this process is only done for the training and validation data. The Gallagher's dataset does not need this step because it is already equally distributed. Finally, the patches are normalized to be zero mean and unit variance as explained in Section 3.4.

4.3. Results

This section summarizes the results obtained from the experiments carried out. These experiments have evaluated both the basic version of the Local-DNN model and two extra enhancements, location and reliability weight, explained in Section 3.5. Additionally, cross-database results, in which one database is used for training and the other for testing, are included to show the validity of our approach and generalization capabilities. Finally, our local DNN model is compared against other state-of-the-art approaches.

Our Local-DNN has several parameters to choose. First, we have set the patch size to 13 × 13 pixels. This value was inspired from our experience in other previous works that also use a local feature framework applied to face images, extracting patches similar to the size of an eye in the image (Paredes et al. (2001)). Second, the DNN itself has also several parameters. In this work, we have used hidden layers with 512 ReLU units and we have changed the number of hidden layers to compare the classification performance. A representation of this network can be seen in the Figure 1. Note that the input layer has 169 units because the patch size is 13 × 13. Note that the dimension of the input layer could also be 171 if the patch location information is used. Finally, it should be mentioned that we have used five-fold cross-validation in both databases. The network is trained until the average cross-entropy error on the training data falls bellow a prespecified threshold. To figure out this threshold, we train another network with the same architecture but using only 3 folds from the training data and using the remaining fold as a validation set. Then, the cross-entropy threshold value is fixed with the smallest classification error obtained on the validation set. At the end, the test results on the 5 combinations are averaged.

Table 1 presents the accuracy at the patch level of several networks varying the number of hidden layers. With these results, we can get an idea of how well the network is able to classify each patch as a male or as a female. Note that the results also show the advantage of including the location information of each patch, as a modification of the base model.

According to these results, we can see that there is a big difference between the network with one hidden layer and the networks with two or more hidden layers. This issue occurs in both databases and denotes the poorer representation power in the case of using just one hidden layer. It is also clear that using the location information at the patch level improves the performance and allows the network to better estimate the label of each patch.

Table 1: Accuracy at patch level on the test set for the Local-DNN model varying the depth.

LFW Database			
Model	**Depth**	**Patch Acc. (%)**	
		w/o loc.	**with loc.**
Local DNN	1 layer	68.48	71.18
	2 layer	74.30	77.17
	3 layer	74.50	**77.87**
	4 layer	74.34	77.26

Gallagher's Database			
Model	**Depth**	**Patch Acc. (%)**	
		w/o loc.	**with loc.**
Local DNN	1 layer	64.87	67.14
	2 layer	70.70	**72.83**
	3 layer	70.51	72.65
	4 layer	70.52	72.37

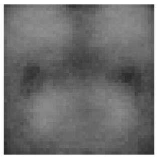

Figure 3: Estimation of the probability of accuracy at patch level

Delving into the topic of the patch-based results, it is also interesting to analyze the distribution of this accuracy depending on the position of the patch in the image. In fact, this could be considered an empirical approximation of the $p(accuracy \mid x^{[i]})$ mentioned in Section 3. To this end, the Figure 3 qualitatively shows the probability of accuracy depending on the position where the patch was extracted. In other words, a pixel in that image indicates how likely is that the patch extracted in that position predicts the correct class. In the image, the light color denotes higher probability and the dark color denotes lower probability. From this figure, we notice that the "best" patches are those centered around the eyes and the mouth. To some extent, this makes sense because those areas of the face are very representatives for distinguishing the gender of a person.

Table 2 summarizes the results obtained with the final decision rule to classify the whole face image. These results are the accuracy obtained for both databases on the test set. As proposed in Section 3, the probabilistic framework allows to make this decision using two methods during testing: the first one is summing all the posteriors given by each patch of the image (Eq. 8), and the second one is based on a voting scheme where each patch chooses a class according to its maximum local posterior (Eq. 9). We present the results for both methods and varying the depth on the network as well. Note that the results obtained by summing posteriors use the same weight for all the contributions, so $\alpha_i = 1/F$.

Table 2: Accuracy on the test set for our Local-DNN model varying the depth.

LFW Database					
Model	**Depth**	**Acc. (%)**			
		Voting		**$\sum$ Posteriors**	
		w/o loc.	**with loc.**	**w/o loc.**	**with loc.**
Local DNN	1 layer	91.63	92.38	91.66	92.64
	2 layer	95.19	95.86	95.35	95.98
	3 layer	95.62	95.74	95.81	96.04
	4 layer	95.71	96.20	95.79	**96.25**

Gallagher's Database					
Model	**Depth**	**Acc. (%)**			
		Voting		**$\sum$ Posteriors**	
		w/o loc.	**with loc.**	**w/o loc.**	**with loc.**
Local DNN	1 layer	82.62	83.76	83.25	84.73
	2 layer	89.14	90.02	89.48	89.96
	3 layer	89.58	90.49	89.74	**90.58**
	4 layer	89.64	89.94	89.85	90.29

According to the results it is clear that summing posteriors yields slightly better results for all cases. Again, the results show a big gap between using a one hidden layer network or using more hidden layers. On the other hand, the improvement obtained at a patch level (previous table) by including the location information produces also an improvement at image level, obtaining the best results in both databases for this case.

Besides these results, we have also evaluated the modification proposed in Section 3.5 regarding the use of different weights when summing posteriors during testing. The α_i values are estimated using the probability of accuracy of each patch placement, as explained before. These values were normalized to sum up 1.0 for all the patches extracted. This means that the higher α_i values will correspond with features with high probability of accuracy, and vice-versa. The experiments were performed only for the best case in both databases. We obtained 96.23% of accuracy in the LFW database and 90.23% in the Gallagher's database. Both results are quite similar to those obtained using equally reliable features in both databases. Probably, this is because the feature selection described in Figure 2 is adequate and it has already selected the most important features for each image.

Table 3: Cross-database accuracy at image level using the Local-DNN model by summing the local class posteriors.

Train using LFW and test using Gallagher's			
Model	**Depth**	**Accuracy (%)**	
		w/o loc.	**with loc.**
Local DNN	1 layer	73.33	78.47
	2 layer	80.50	**83.03**
	3 layer	82.04	82.91
	4 layer	82.15	81.21

Train using Gallagher's and test using LFW			
Model	**Depth**	**Accuracy (%)**	
		w/o loc.	**with loc.**
Local DNN	1 layer	89.56	90.41
	2 layer	93.53	93.93
	3 layer	93.98	94.39
	4 layer	93.76	**94.48**

To further prove the validity of our approach, Table 3 presents cross-database results in which one database is used for training and the other for testing. The cross-database results show that the local-DNN models can generalize well to a different database at the expense of a small performance penalty. It should be emphasized that our cross-database results are better than the only previously published cross-database results presented in Dago-Casas et al. (2011) where a 81.02% and 89.77% are obtained when testing with the Gallagher's and LFW respectively.

Finally, the last Table 4 compares the best results obtained with our Local-DNN model along with other state-of-the art approaches that follow the same evaluation protocol in both databases. In this table we have also included other results obtained using two well-known deep learning networks. On the one hand, we have tested a standard DNN with three hidden layers and 512 hidden ReLU units in each layer. The weights of this network were pretrained using RBMs, and the dropout technique has been applied to the last hidden layer. This configuration was obtained by performing an extensive set of experiments to fix the best configuration for the problem at a hand. Note that the original images were cropped and resized to a smaller size (40 × 32 pixels) to reduce the number of connections. On the other hand, we have also evaluated a Deep Convolutional Neural Network (DCNN). The architecture that we have used is inspired by the excellent results obtained recently in Taigman et al. (2014) with the LFW database in the face recognition problem. More details about the architecture of the network can be found in that reference. All the hidden units are ReLU, and the dropout technique has been also applied to the last hidden layer.

Table 4: Best accuracy on the test set for DNN, DCNN and Local-DNN models, and other published results.

LFW Database	
Model	**Acc.(%)**
DNN	92.60
DCNN	94.09
Best Local-DNN	**96.25**
Gabor+PCA+SVM Dago-Casas et al. (2011)	94.01
Boosted LBP+SVM Shan (2010)	94.44

Gallagher's Database	
Model	**Acc.(%)**
DNN	84.28
DCNN	86.04
Best Local-DNN	**90.58**
Gabor+PCA+SVM Dago-Casas et al. (2011)	86.61
FPLBP+Drop SVM Eidinger et al. (2014)	88.60
LBP+CH+SIFT SVM Fazl-Ersi et al. (2014)	**91.59**

According to these results, our Local-DNN model outperforms other deep learning methods, such as DNN and DCNN, and also obtains the best published results on the LFW dataset. The results obtained with DCNN are worse mainly because the number of images is low in order to get good results with these deep networks. Note that our Local-DNN is trained with patches (millions) while the DCNN are trained with whole images (thousands). In this sense, it is important to note that dropout did not improve the results with the LDNN due to the large number of training patches and the fact that these training patches represent local parts of the face with different translations. Thus, these properties of the training patches act as a kind of regularization. On the other hand, some prior knowledge about the content is used to extract the patches from edge areas, a kind of saliency detection, while the DCNN method has to learn not only the discriminative part of the problem but also the saliency detection with a relative small number of training samples.

Other results using this database presented by Tapia and Perez (2013); Ren and Li (2014), not included in this table, were obtained removing many images, using only 7443 and 6840 samples out of 13233, respectively. It is important to underline that our results are obtained with the entire dataset, without removing any image. For this reason, both results cannot be compared on equal terms. On the other hand, very recently, Fazl-Ersi et al. (2014) obtained a slightly better result in the Gallagher database. However, this result is obtained by using a complex ensemble composed by several handcrafted features such as LBPs, Color Histograms and SIFT. In contrast, our method is quite simple and generic to apply, so it may also work well in other computer vision tasks.

5. Conclusions

This paper presents a new discriminative model called Local Deep Neural Network (Local-DNN), which is based on two key concepts: local features and deep architectures. This model learns to classify small patches extracted from images using a standard DNN. The final classification of each image is performed using a simple voting scheme that takes into account the contributions from all the patches of that image. The experiments carried out have evaluated the model on the gender recognition problem using unconstrained face images, by following two benchmarks proposed for the LFW and the Gallagher datasets.

The results obtained in the experiments confirm the advantage of learning independently from small regions in the visual field when using DNNs in the problem at a hand. In particular, our Local-DNN model works well with networks with at least two hidden layers to be able to learn from small patches. After that, the final decision rule based on summing posteriors yields slightly better results than the simple voting scheme. It is also worth mentioning the improvement obtained by keeping the topological information of each patch, including in the network the location were it was extracted. However, the use of different weights in the final decision, obtained as an estimation of the probability of accuracy of each patch, did not improve the results. Using this configuration of parameters, our Local-DNN model outperforms other Deep Learning models also evaluated in this work, such as pre-trained DNNs and Deep Convolutional Neural Networks (DCNNs). There is also an improvement over other state-of-the-art results in the LFW dataset, which are obtained using traditional handcrafted features and a Support Vector Machine (SVM) classifier. Actually, we obtain the best result published using this protocol without discarding any image from the original database. The result obtained in the Gallagher's dataset is also competitive, considering the simplicity and the generalization capability of the model proposed. Finally, the cross-database results obtained using one database for training and the other one for testing demonstrate that our approach can generalize well, and obtains better results than the only previously published cross-database result presented using the same databases.

Acknowledgments

This work was financially supported by the Ministerio de Ciencia e Innovacin (Spain), Plan Nacional de I+D+i, TEC2009-09146, and the FPI grant BES-2010-032945.

References

Baluja, S., Rowley, H.A., 2007. Boosting sex identification performance. Int. J. Comput. Vision 71, 111–119.

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. Pattern Analysis and Machine Intelligence, IEEE Transactions on 35, 1798–1828.

Buchala, S., Davey, N., Frank, R., Gale, T., 2004. Dimensionality reduction of face images for gender classification. Intelligent Systems, 2004. Proceedings. 2nd International IEEE Conference 1, 88–93 Vol.1.

Dago-Casas, P., Gonzlez-Jimnez, D., Yu, L.L., Alba-Castro, J.L., 2011. Single- and cross- database benchmarks for gender classification under unconstrained settings., in: ICCV Workshops, pp. 2152–2159.

Eidinger, E., Enbar, R., Hassner, T., 2014. Age and gender estimation of unfiltered faces. IEEE Transactions on Information Forensics and Security 9, 2170 – 2179.

Farabet, C., Couprie, C., Najman, L., LeCun, Y., 2013. Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1915–1929.

Fazl-Ersi, E., Mousa-Pasandi, M., Laganiere, R., Awad, M., 2014. Age and gender recognition using informative features of various types, in: IEEE ICIP 2014.

FIPA, 2011a. Folds for gender evaluation on gallagher's database. `http://fipa.cs.kit.edu/download/Gallagher_gender_5folds.txt`.

FIPA, 2011b. Folds for gender evaluation on lfw. `http://fipa.cs.kit.edu/download/LFW-gender-folds.dat`.

Gallagher, A., Chen, T., 2009. Understanding images of groups of people, in: Proc. CVPR.

Graf, A.B.A., Wichmann, F.A., 2002. Gender classification of human faces, in: Proceedings of the 2nd International Workshop on Biologically Motivated Computer Vision (BMCV), London, UK. pp. 491–500.

Hinton, G.E., Salakhutdinov, R., 2006. Reducing the dimensionality of data with neural networks. Science 313, 504–507.

Huang, G., Mattar, M., Lee, H., Learned-Miller, E., 2012. Learning to align from scratch, in: Advances in Neural Information Processing Systems 25, pp. 773–781.

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 07-49. University of Massachusetts, Amherst.

Krizhevsky, A., 2009. Learning multiple layers of features from tiny images. Technical Report.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems.

Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K., 2009. Attribute and Simile Classifiers for Face Verification, in: IEEE International Conference on Computer Vision (ICCV).

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

Leng, X., Wang, Y., 2008. Improving generalization for gender classification., in: ICIP, pp. 1656–1659.

Levi, G., Hassner, T., 2015. Age and gender classification using convolutional neural networks, in: IEEE Conf. on CVPR workshops.

Mansanet, J., Albiol, A., Paredes, R., Villegas, M., Albiol, A., 2014. Restricted boltzmann machines for gender classification, in: 11th International Conference, (ICIAR), pp. 274–281.

Moeini, A., Moeini, H., 2015. Pose-invariant gender classification based on 3d face reconstruction and synthesis from single 2d image. Electronics Letters 51, 760–762. doi:10.1049/el.2015.0520.

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

Ng, C., Tay, Y., Goi, B.M., 2012a. Recognizing human gender in computer vision: A survey, in: PRICAI 2012: Trends in Artificial Intelligence. volume 7458 of *Lecture Notes in Computer Science*, pp. 335–346.

Ng, C.B., Tay, Y.H., Goi, B.M., 2012b. Vision-based human gender recognition: A survey. CoRR .

Paredes, R., Prez, J.C., Juan, A., Vidal, E., 2001. Local representations and a direct voting scheme for face recognition, in: In Workshop on Pattern Recognition in Information Systems, pp. 71–79.

Ren, H., Li, Z., 2014. Gender recognition using complexity-aware local features, in: 22nd ICPR 2014, pp. 2389–2394.

Schroff, F., Kalenichenko, D., Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering. CoRR abs/1503.03832.

Serre, T., Kreiman, G., Kouh, M., Cadieu, C., Knoblich, U., Poggio, T., 2007. A quantitative theory of immediate visual recognition. PROG BRAIN RES , 33–56.

Shakhnarovich, G., Viola, P.A., Moghaddam, B., 2002. A unified learning framework for real time face detection and classification, in: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 16–.

Shan, C., 2010. Gender classification on real-life faces, in: ACIVS (2), pp. 323–331.

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

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

Smolensky, P., 1986. Parallel distributed processing: Explorations in the microstructure of cognition, volume 1. chapter Information Processing in Dynamical Systems: Foundations of Harmony Theory, pp. 194–281.

Sun, Y., Wang, X., Tang, X., 2014. Deep learning face representation from predicting 10,000 classes, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pp. 1891–1898. doi:10.1109/CVPR.2014.244.

Taigman, Y., Yang, M., Ranzato, M., Wolf, L., 2014. Deepface: Closing the gap to human-level performance in face verification. Conference on CVPR, 2014 .

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.

Villegas, M., Paredes, R., 2011. Dimensionality reduction by minimizing nearest-neighbor classification error. Pattern Recognition Letters 32, 633 – 639.

Villegas, M., Paredes, R., Juan, A., Vidal, E., 2008. Face Verification on Color Images Using Local Features, in: CVPR Workshops., pp. 1–6.