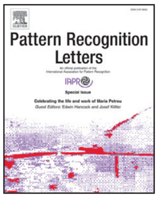
Pattern Recognition Letters 70 (2016) 80–86

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec



Pattern Recognition Letters

Local Deep Neural Networks for gender recognition[☆]

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

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

ARTICLE INFO

Article history:
Received 7 July 2015
Available online 30 November 2015

Keywords:
Gender recognition
Face analysis
Deep learning
Local Deep Neural Network,

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 as 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 [25]: 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 [18]. 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 [2]. 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 [29].

These strategies have shown an excellent performance in challenging problems on the computer vision domain [6,16,36]. 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 [15]. This problem has 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 [15]. A similar idea is used in Deep Convolutional Neural Networks (DCNNs), 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

[☆] This paper has been recommended for acceptance by G. Moser.
[*] Corresponding author: Tel.: +34 963879738; fax: +34 963877309.
E-mail address: alalbiol@iteam.upv.es (A. Albiol).

http://dx.doi.org/10.1016/j.patrec.2015.11.015
0167-8655/© 2015 Elsevier B.V. All rights reserved.

robustness to small translations, occlusions and local distortions, see [39]. 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 DCNNs 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) [14] and the so-called Gallagher's database [10], 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 [24]. 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 [23] or pixel differences [1]. Sometimes, simple features are pooled together as in [17], where image intensities in RGB and HSV color spaces, edge magnitudes, and gradient directions were combined. More elaborated features include Haar-like wavelets [30], Local Binary Patterns (LBPs) [32] or Gabor wavelets [19]. 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 [22] 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 [38] 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 [3,11]. 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 [1,5,32]. In this spirit, an excellent comparison of gender recognition techniques using different methods can be found in [4].

Regarding deep learning techniques, unsupervised models, such as Restricted Boltzmann Machines (RBMs) [34], have been demonstrated to be useful as a way to pre-train these deep architectures [12]. These models are able to automatically extract good features from unlabeled data that are useful in supervised tasks like the gender recognition problem [21]. On the other hand, DCNN models have shown great performance in computer vision tasks by learning from small regions in the visual field [16,33] and have been successfully used for face recognition [28,35,36]. 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 [20].

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 [39], to model the local feature-based classification. This framework is general, but in this work, we particularize it for the problem at one 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 $c = 1, \ldots, C$ 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 $\mathbf{x}^{[i]}$, $i = 1, \ldots, F$, 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, $c_i \in \{1, \ldots, C\}$.

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,

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

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 $\mathbf{x}$),

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

In order to develop a practical model for $p(c|\mathbf{x})$, 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,

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

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,

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

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:

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