ISSN 2395-8618

Facial Recognition using Convolutional Neural Networks and Supervised Few-Shot Learning

Rafael Gallardo García, Beatriz Beltrán, Darnes Vilariño, and Rodolfo Martínez

***Abstract*—The paper presents a feature-based face recognition method. The method can be explained in two separated processes: A pretrained CNN-Based face detector looks for faces in images and return the locations and features of the found faces, this face detector will be used to train the models for the classifiers and then will be used to find unknown faces in new images. The used classifiers are: K-Nearest Neighbors, Gaussian Naive Bayes and Support Vector Machines. Each model will be trained with a different quantity of training examples in order to obtain the best version of the method. When the models are ready, each classifier will try to classify the faces with the previously trained models. The accuracy of each classifier in few-shot face recognition tasks will be measured in Recognition Rate and F1 Score, a comparative table of the results is presented. This paper has the goal to show the high accuracy achieved by this method in datasets with several individuals but few examples of training.**

***Index Terms*—Convolutional neural network, facial recognition, artificial vision, few-shot learning.**

Fig. 2 shows the used structure for tests before being analyzed for the facial recognition system, *Fig. 3* shows the output after performing the method over the *Fig. 2.*

I. Introduction

ARTIFICIAL vision is one of the artificial intelligence disciplines which tries to develop, improve and research new methods trough which computers are able to acquire, process, analyze and understand real-world images in order to get numerical or symbolic information that can be more easily processed by computers. The face recognition is one of the most common applications of the Biometric Artificial Intelligence, a facial recognition system is capable of identify or verify persons in digital images or videos. Face recognition technology can be used in wide range of applications such as identity authentication, access control, and surveillance. A face recognition system should be able to deal with various changes in face images [1].

II. Related Work

One of the first and most successful template matching methods is the eigenface method [2], which is based on the Karhunen Loeve transform (KLT) or the principal component analysis (PCA) for the face representation and recognition. Every face image in the database is represented as a vector of weights, which is the projection of the face image to the basis in the eigenface space [1]. Usually the nearest distance criterion is used for face recognition. Guodong *et al.* [1] focused on the face recognition problem and showed that the discrimination functions learned by SVMs can give much higher recognition accuracy than the pupular eigenface approach [2] working on larger datasets.

Convolutional Neural Networks (CNNs) have taken the computer vision community by storm, significantly improving the state of the art in many applications. One of the most important ingredients for the success of such methods is the availability of large quantities of training data [3]. Parkhi *et al.* [3] traversed through the complexities of deep network training and face recognition to present methods and procedures to achieve comparable state of the art results on the standard LFW and YTF face benchmarks.

King Davis, developed a frontal face detector [4] using *Histogram of Oriented Gradients(HOG)* and *linear SVMs*, this HOG+SVM method achieved good results in frontal face detection but it don't detect faces at odd angles. King Davis also included a CNN-based face detector on his *Dlib library* [4] which is slower but better detecting faces in all angles, this face detector cannot perform face recognition by itself.

III. Structure of the Face Recognition Method: A Theoretical Approach

This paper describes a feature-based face recognition method, in which the features are derived from the intensity of the data without assuming any knowledge of the face structure. The feature extraction model is biologically motivated, and the locations of the features often corresponds to face's landmarks, this could be nose, mouth, eyes, eyebrows and chin. In the presented method, a CNN search and extracts face's landmarks and then a classifier will try to find the person who that face's landmarks belongs.

A. Convolutional Neural Networks

CNNs are a class of artificial neural networks where the used neurons are very similar to the biological neurons that corresponds to the receptive field of the *primary visual cortex(V1)* [5], this means that were inspired by biological processes [5], [6]. The CNN are a regularized variation of the *multilayer perceptrons*, this means that a CNN is a fully

Manuscript received on June 24, 2019, accepted for publication on August 29, 2019, published on December 30, 2019.

The authors are with the Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, Mexico (e-mail: rafael.gallardo@alumno.buap.mx, {bbeltran,darnes,beetho}@cs.buap.mx).

ISSN 2395-8618

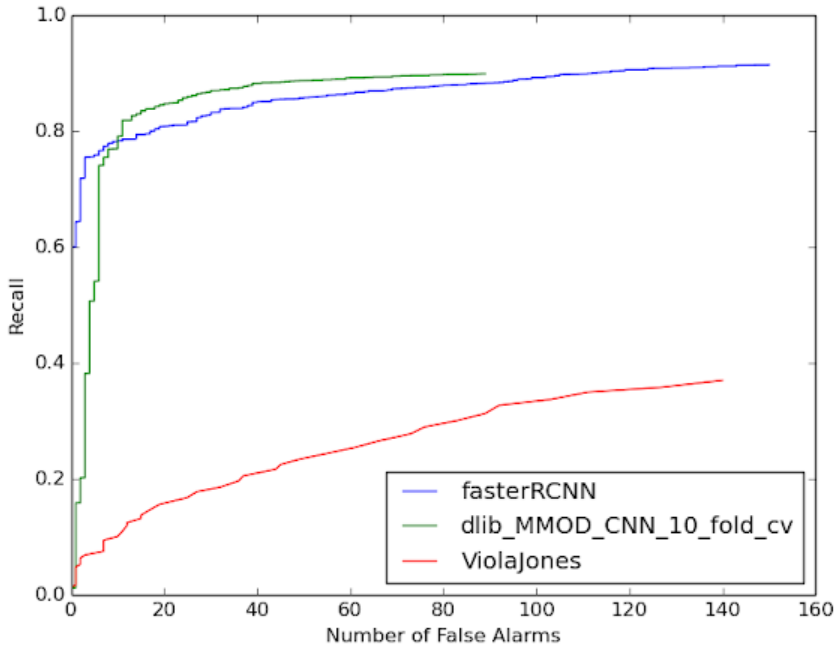

Fig. 1. CNN-MMOD on FDDB challenge.

connected network, that is, each neuron in one layer is connected to all neurons in the next layer [7], a regularized network has some magnitude measurement of weights on the loss function. Convolutional neurons works over bidimensional matrices and the extraction of the features is realized by processors that operate over the image data, the output of each convolutional neuron is:

$$y_j = g(b_j + \sum_i k_{i,j} \otimes y_i), \tag{1}$$

where the output y_i of a neuron j is a matrix that were calculated with the linear combination of the y_i neuron's output on the previous layer, each of this neurons is operated with a convolutional kernel k_{ij}, this quantity is added to a b_j influence. Then, the outputs are activated with an g non-linear activation function. The convolutional operator has the function of filter the given image and transform the input data in such way that the important features become more relevant at the output [6]. Once the feature extraction is complete, the *classification neurons* will try to classify the found features with base on the imposed rules by the previous training. The behavior of this neurons is similar to the *multilayer percentron's* neurons, and is calculated as follow:

$$y_j = g(b_j + \sum_i k_{i,j} \bullet y_i), \tag{2}$$

where the y_j output of a j neuron is calculated with the linear combination of the y_i neuron's output on the previous layer multiplied for a W_{ij} weight, the output of this operations is added to the influence factor b_j, then is activated with a g activation function.

B. K-Nearest Neighbors Classification

This is a non-parametric supervised learning method where the estimations are based on a training dataset and all computation is deferred until classification. k-NN estimate the density function $F(x|C_j)$, where x are the predictors and C_j is each class. The training examples are vectors in a multidimensional characteristic root, each example is described in terms of p attributes considering q classes for classification. A partitioning of the space is performed in order to separate regions and it labels. A point e belongs to C if this class is the most frequent in the k closest training examples, this is:

$$d(x_i, x_j) = \sqrt{\sum_{r-1}^{p} (x_{ri} - x_{rj})^2}, \tag{3}$$

the training phase stores the eigenvectors and the labels of the classes of the training examples, this is: for each $< x, f(x)$, where $x \in X$, this example should be added to the examples structure. In the classification phase, on an example x_q that will be classified, being $x_1, ..., x_n$ the k nearest neighbors to x_q in the training examples, then the output will be:

$$\hat{f} \leftarrow argmax_{v \in V} \sum_{i=1}^{k} \varsigma(v, f(x)), \tag{4}$$

ISSN 2395-8618

TABLE I
SCORES OF THE FACIAL RECOGNITION METHOD.

Training Shots	Algorithm	RecognitionRate	Precision	Recall	F1
1-Shot	k-NN	99.15	0.991	0.991	0.991
	SVM	N/A	N/A	N/A	N/A
	GNB	99.09	0.921	1	0.959
2-Shots	k-NN	99.159	0.991	0.991	0.991
	SVM	100	0.915	1	0.955
	GNB	99.159	0.921	1	0.959
3-Shots	k-NN	99.15	0.991	0.991	0.991
	SVM	100	0.915	1	0.955
	GNB	99.159	0.921	1	0.959
4-Shots	k-NN	100	0.991	1	0.995
	SVM	100	0.915	1	0.955
	GNB	100	0.922	1	0.959
5-Shots	k-NN	100	0.991	1	0.995
	SVM	100	0.915	1	0.955
	GNB	100	0.922	1	0.959
6-Shots	k-NN	100	0.991	1	0.995
	SVM	100	0.915	1	0.955
	GNB	100	0.922	1	0.959
7-Shots	k-NN	100	0.991	1	0.995
	SVM	100	0.915	1	0.955
	GNB	100	0.922	1	0.959
8-Shots	k-NN	100	0.991	1	0.995
	SVM	100	0.922	1	
	GNB	100	0.922	1	0.959
9-Shots	k-NN	100	0.991	1	0.995
	SVM	100	0.922	1	0.959
	GNB	100	0.922	1	0.959
10-Shots	k-NN	100	0.991	1	0.995
	SVM	100	0.929	1	0.963
	GNB	100	0.937	1	0.967

where $\varsigma(a, b) = 1$ if $a = b$ and 0 in other cases. If $k = 1$ the closest neighbor to x_i determines it value.

C. Support Vector Machines Classification

Support Vector Machines (SVM) are supervised learning models, or learning algorithms which builds *non-probabilistic binary linear classifers* that assings new examples to a category. Given training vectors $x_i \in \mathbb{R}^p$, $i = 1,n$, in two classes, and a vector $y \in \{1, -1\}^n$, the algorithm solves the following problem:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i, \tag{5}$$

subject to:

$$y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \zeta_i \geq 0, i = 1, ..., n, \tag{6}$$

and its dual is:

$$min_\alpha \frac{1}{2} \alpha^T Q \alpha - e^T \alpha, \tag{7}$$

subject to:

$$y^T \alpha = 0 \, and \, 0 \leq \alpha_i \leq C, i = 1, ..., n, \tag{8}$$

where e is the vector of all ones, $C > 0$ is the upper bound, Q is an *n by n* positive semidefinite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, where $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is the kernel. The training vectors are implicity mapped into a higher dimensional space by the function ϕ. The decision fuction is:

$$\sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + \rho. \tag{9}$$

D. Gaussian Naive Bayes Classification

Naive Bayes methods are supervised learning algorithms based on the application of the Bayes' theorem and by assuming the conditional independence between every pair of features given the values of the class variable, this is a naive assumption. Naive Bayes classifiers uses the following classification rule:

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y), \tag{10}$$

ISSN 2395-8618

Fig. 2. Example of testing subset.

the previous equation could be write as:

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i \mid y). \tag{11}$$

In *Gaussian Naive Bayes*, the likelihood of the features is assumed to be Gaussian:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right), \tag{12}$$

σ_y and μ_y are estimated using maximum likelihood.

IV. Structure of the Face Recognition Method: A Technical Approach

In general, the facial recognition method consists in a face detector, and a clustering algorithm, for experimental purposes, three different classifiers were tested, technically the *Table 1* presents a comparison between the methods, which have the same CNN-based face detector, but its accuracy will depend on the effectiveness of each classifier when clustering and recognizing faces. Details are specified below.

A. CNN-based Face Detector

This method used a *Maximun-Margin Object Detector(MMOD)* [8] with CNN-based features. The face detector were trained with various datasets like ImageNet, PASCAL VOC, VGG, WIDER and Face Scrub, the training dataset for the face detector contains 7220 images even so, a good CNN-based face detector can be obtained with a training dataset with just 4 examples. The CNN version of MMOD tested with the 10-fold cross-validation version of the FDDB challenge [9], gives the results shown in Fig. 1. The X axis is the number of false alarms produced over a dataset with 2885 images. The Y axis is the fraction of faces found by the detector (recall). The green curve is the CNN-based MMOD trained with 4600 faces, the red curve is the old Viola Jones

ISSN 2395-8618

Fig. 3. Example of results over the testing subset.

detector, and the blue curve is the Faster R-CNN [10] trained with 159,424 faces. The results presented on the chapter 5 were realized by running the CNN face detector over the *NVIDIA CUDA Deep Neural Network (cuDNN)* library, with a NVIDIA 960M GPU with 4GB of VRAM with Maxwell architecture and 5.x compute capability.

B. Structure of the Experiments

In order to test the facial recognition method and the accuracy of the classifiers as few-shot learning, 10 models were trained per each classifier, each model has a different quantity of training examples per subject, from 1 training example to 10 training examples per subject per classifier, 30 models at the end of training. The accuracy of the method is the result of the combination of the achieved accuracy for the CNN face detector when searching for faces in images and extracting it face's landmarks, and the achieved accuracy of the clustering algorithms when finding the class of each unknown face. The experiments were performed over the *MIT-CBCL Face Recognition Database* [11], this database contains face images of 10 subjects in high resolution, including frontal, half-profile and profile views. Test images have a size of 115x115 pixels.

V. Experimental Results

Table I presents the obtained scores with the three different classifiers, when using the CNN-MMOD as face detector and the classifiers as face recognizer, this experiments were performed over a subset of the *MIT-CBCL Face Recognition Database* [11], the results on the *table 1* belongs to an experiment performed over a subset of 119 known faces with 11 unknown faces to be recognized.

VI. Conclusions

The experiments are clear, the presented facial recognition method can be considered as a success, the three used

ISSN 2395-8618

classifiers gave a $> 99\%$ of Recognition Rate and a F1 score > 0.9, mixing the accuracy of a CNN-based face detector with face clustering techniques gives very interesting results.

Focusing on the difference between the classifiers, it's easy to see that the k-Nearest Neighbors classifier has a recognition rate similar to the other classifiers but it's noticeably better in Recall and Precision scores, in consequence it is better in the F1 score. Surprisingly, the quantity of training examples per subject did not greatly affect the accuracy of the facial recognition method. k-Nearest Neighbors was notably better while recognizing known people and was better while labeling faces as "unknown", Support Vector Machines and Gaussian Naive Bayes versions of the facial recognition method gave excellent results while recognizing known faces but gave disappointing results while trying to identify unknown faces, labeling faces as "known" and giving false positives.

Execution times of the CNN-MMOD face detector is very slow when it runs on a Intel Core i7-6700HQ CPU, with a 4.5 FPS average, but this FPS improve if the face detector runs over a CUDA GPU, reaching up to 200 FPS, this FPS makes CNN-MMOD+k-NN a feasible method for real-time facial recognition. k-NN reaches high accuracy scores, both in F1 and in Recognition Rate, with feasible times by using parallel programming.

Future work include optimizing the execution time to obtain a faster facial recognition system, this time optimization consists in the optimization of the classifiers and the CNN. Also, this facial recognitions method will be tested in different type of cameras and places in order to evaluate it performance on the real world.

VII. Acknowledgments

Credit is hereby given to the Massachusetts Institute of Technology and to the Center for Biological and Computational Learning for providing the database of facial images.

Also thanks to the vice-rectory of research and postgraduate studies *(VIEP)* of the Benemérita Universidad Autónoma de Puebla for the project: *Metodología basada en gramáticas para la detección de eventos anómalos por medio de redes complejas en tiempo real.*

References

[1] G. Guodong, Z. L. Stan, and C. Kapluk, "Face recognition by support vector machines," in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 1970.

[2] M. A. Turk and A. P. Pentland, "Eigenfaces for recognition," *Cognitive Neuroscience*, vol. 3, pp. 71–86, 1991.

[3] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference (BMVC)*, X. Xie, M. W. Jones, and G. K. L. Tam, Eds. BMVA Press, September 2015, pp. 41.1–41.12. [Online]. Available: https://dx.doi.org/10.5244/C.29.41

[4] E. D. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, 2009.

[5] D. H. Huvel and T. N. R. f. a. Wiesel, "and functional architecture of monkey striate cortex," *The Journal of Physiology*, vol. 195, pp. 215–243, 1968.

[6] M. Matusugu, M. Katsuhiko, and M. Yusuke, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks*, vol. 16, pp. 555–559, 2003.

[7] L. Hongxin, S. Xiaorong, and R. F.-N. Haibing, "Joint convolutional neural networks for face detection and attribute recognition," in *9th International Symposium on Computational Intelligence and Design*, 2016.

[8] E. D. King, "Max-margin object detection," arXiv [cs.CV] 1502.00046, 2015.

[9] V. Jain and E. Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," Dept. of Computer Science, University of Massachusetts, Amherst, Technical Report UM-CS-2010-009, 2010.

[10] H. Jiang and E. Learned-Millerr, "Face detection with the faster R-CNN," arXiv [cs.CV] 1606.03473, 2016.

[11] B. Weyrauch, J. Huang, B. Heisele, and V. Blanz, "Component-based face recognition with 3D morphable models," in *First IEEE Workshop on Face Processing in Video*, 2004.