Real-Time Gender Recognition From Face Images Using Deep Convolutional Neural Network

Carlos Ismael Orozco[1], **Florencia Iglesias**[2], **María Elena Buemi**[2], **Julio Jacobo Berlles**[2]

[1]Departamento de Informática, FCE, Universidad Nacional de Salta. Argentina.
[2]Departamento de Computación, FCEyN, Universidad de Buenos Aires. Argentina.
iorozco@exa.unsa.edu.ar, iglesiasflorencia@gmail.com, {mebuemi, jacobo}@dc.uba.ar

Keywords: Gender Recognition, Convolutional Neural Network, Ubunsa Architecture, Deep Learning.

Abstract

Gender recognition is a topic of interest in computer vision due to its applications such as surveillance in public places, directed advertising, among others. The good results obtained using deep convolutional neural networks in vision tasks make them an attractive tool to improve the capacities of gender recognition systems. In this work we propose a deep convolutional network architecture to classify as male or female person the candidate regions previously detected using Haar features embedded in an AdaBoost. The data set used for training and testing come from the Labeled Faces in the Wildand Gallagher's dataset. We have evaluated the classification results on the proposed architecture and have obtained an average of $\sim 95.42\%$ and $\sim 91.48\%$ accuracy for the training set and for the test set, respectively, that are competitive with those mentioned in the bibliography. We have also carried out a real-time evaluation of the system using a web camera.

1 Introduction

The importance of gender recognition systems comes from the demand for computer-based applications in areas such as surveillance of public places, collection of demographic information, market research, directed advertising, among others. This constitutes an important challenge because of the variety of scales, positions, lighting conditions and ethnicity in which a person can be found within the image.

Currently there are many papers proposing different strategies to solve this problem. They can be grouped into two large categories: conventional approaches and deep learning approaches. In the conventional approaches, features are extracted, as in [1], where gender recognition is done using Support Vector Machine (SVM) and Radial Basis Function (RBF). In Bekios et. al. [2], gender recognition is based on Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Bayes Classifier. In [3], the authors propose a gender recognizer based on Local Binary Histogram Fourier Features (LBP HF). As examples of a deep learning approach we can cite the work done by Mansanet et.al. [4], based on a Local Deep Neural Network (LDNN), and the paper of Wolfshaar et. al. [5], that uses ConvNet+SVM.

The objective of this work is to implement a real-time gender recognition system. To this end, we have combined the speed of the Haar detector for the detection of the ROIs with the robustness of the Convolutional Neural Networks for their classification. The rest of the paper is organized as follows: in Section 2 describe the general structure of the system; in Section 3, the datasets used and the proposed architecture, are described; Section 4 shows the experiments performed, together with the results obtained and finally, Section 5 presents the Conclusions and future work.

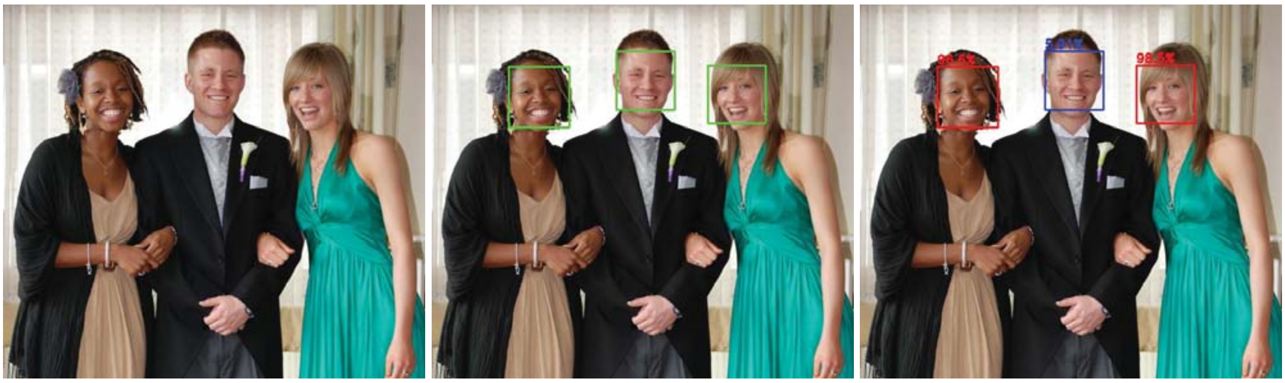
2 Gender Recognition Approach

The system proposed in this paper consists of 2 main stages: (a) Generation of candidate regions (ROIs) delimited by Bounding Boxes (BBs). (b) Classification of the candidate regions into two classes: male person and female person.A general scheme of the proposed approach is shown in Figure 1.

The generation of candidate regions consists in the extraction of portions of the image that can potentially contain the face of a person. The input of this stage is the complete image, while the output is a set of ROIs. A simple method for generating candidate regions is the one proposed by Viola and Jones [6]; this very well known algorithm provides a high hit rate at a low computational cost, and allows the detection of faces in real time. For this stage we used the implementation available at the OpenCV 2.4.13 [1] library.

The goal of the classification stage is to identify which of the ROIs obtained in the previous stage correspond to a male person and which correspond to a female person. For this task, we propose the use of a convolutional neural network (CNN) that takes the candidate regions and classifies them as male or female. CNNs are generally organized in a sequential manner, where each layer takes as input the output of the previous one. The first layers extract characteristics using convolution and subsampling operations, while the last 3 layers make up a fully connected Perceptron that finally clasifies the ROIs.

[1] opencv.org



(a) Input image (b) ROI generation (c) ROI classification

Figure 1. (a) Input image. The gender recognition system consist of 2 main stages. (b) ROI detection, where each face is associated to a Bounding Box. (c) ROI classification, where each candidate region is classified as male person (Blue BB) or female person (Red BB).

3 Convolutional Neural Network

3.1 Data preparation

One of the main aspects of CNNs is the amount of training data needed to achieve good performance. There are currently several datasets [7, 8, 9, 10], each with their strengths and weaknesses, that are used as a basis for gender recognition; for example, FEI [9] and FERET [10] are small databases with faces taken in a controlled environment. The experiments carried out in this work are focused particularly on 2 datasets which have the common characteristic that their images are taken in an unrestricted environment:

Labeled Faces in the Wild (LFW) [7]: which contains $\sim$ 13000 face images of celebrities, that were collected from the web, labeled with the person's name. Some sample images are shown in Figure 2 (Top row).

Gallagher's dataset [8]: which contains $\sim$ 28000 face images collected from Flickr images. Some sample images are shown in Figure 2 (Botton row).

Each image is converted to grayscale and all the pixel values are scaled to the range $[0, 1]$ and resized to 62×62 pixels.

3.2 Architecture

In this work we have used a CNN architecture that was originally used in another paper to classify pedestrians [11]. Here, we have adapted it to the task of gender recognition. We have normalized the candidate ROIs to be used as inputs to a fixed size of 62×62 pixels.The implementation was done in Python using the Keras library [12] for neural networks and the Theano library [13, 14] for Numerical Calculus with GPU support. A schematic of the architecture is shown in Figure 3.

This architecture, we have called Ubunsa [2], has 8 layers. It should be noted that the activation function in all layers is a Rectified Linear Unit (ReLU) where $f(x) = ln(1 + e^x)$, except for the output layer of the multilayer perceptron that uses a softmax activation function.

4 Experiments and Results

- The supervised training of the CNN was done using a Stochastic Gradient Descent (SGD) method, with parameters shown in Table 2.
- The dataset was built with a ratio 1:1 for positive to negative samples.
- The experiments were carried out on a Tesla C2017 GPU.

4.1 Evaluation Metrics

To evaluate the classification performance of our CNN we calculated the accuracy (ACC), that is defined as the ratio between the number of samples that were classified correctly, and the total number of samples, this is

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

with TP: True Positive, TN: True Negative, FP: False Positive and FN: False Negative.

4.2 Experimental CNN Classification

We used the evaluation protocol for gender recognition proposed by Facial Image Processing and Analysis (FIPA); following this protocol, we run a k-fold cross validation with $k = 5$,

[2] Available at: https://github.com/ciorozco/CNN_Ubunsa

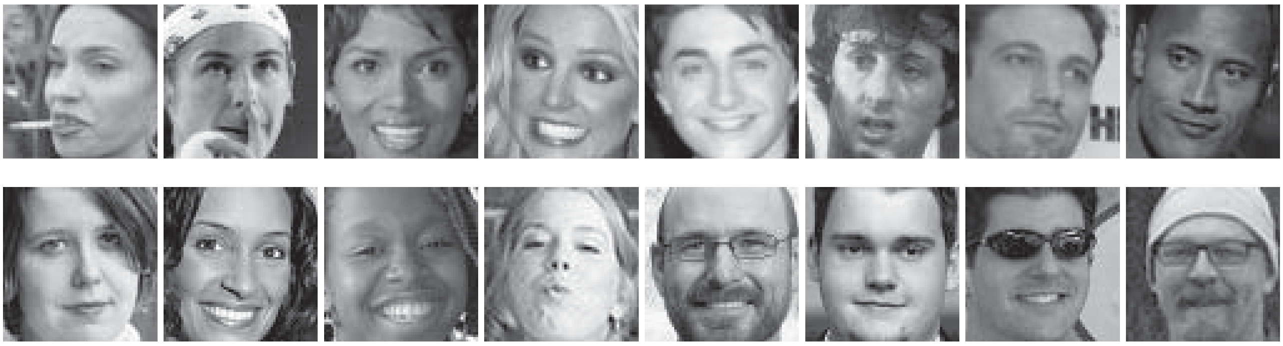


Figure 2. Examples of faces of LFW dataset (top row) and Gallagher's dataset (bottom row). Region cut with VJ [6], converted to grayscale and resized to fixed shape of 62×62 pixels. The datasets have a great variety of people of both genders, poses, ethnic origin and extra accessories (such as: glasses, caps, etc.).

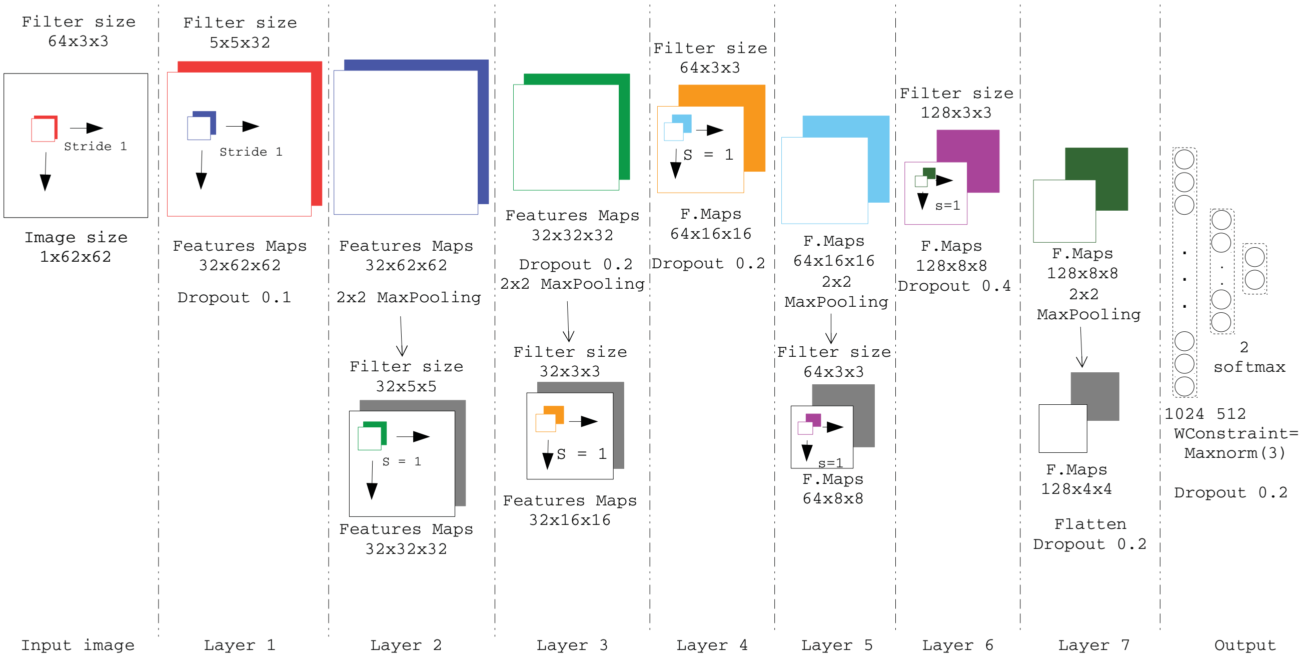


Figure 3. Architecture with 8 layers and $\sim 2 \times 10^6$ parameters to train. The first 7 layers are used for features extraction. The output layer is a Multilayer Perceptron with 1 hidden layer for classification.

Table 1: Summarizing the information on the architecture. The first column indicates the Layer number. The Type column indicates the sublayers type (ie convolution, dropout, maxpooling, flatten, and dense). Output shape indicates the size of the output feature maps. #Param indicates the number of parameters to be trained. The last column shows the shape of the filter (for convolution) or the shape of the mask (for maxpooling).

Layer	Type	Output shape	#Param	Filter (conv.) Mask (pool.)
1	conv.	(32, 62, 62)	1600	(7, 7, 3)
	drop.	(32, 62, 62)	0	-
2	conv.	(32, 62, 62)	25632	(5, 5, 32)
	pool.	(32, 31, 31)	0	(2, 2)
3	conv.	(32, 31, 31)	9248	(3, 3, 32)
	drop.	(32, 31, 31)	0	-
	pool.	(32, 15, 15)	0	(2, 2)
4	conv.	(64, 15, 15)	36928	(5, 5, 32)
	drop.	(64, 15, 15)	0	-
5	conv.	(64, 15, 15)	18496	(3, 3, 32)
	pool.	(64, 7, 7)	0	(2, 2)
6	conv.	(128, 7, 7)	73856	(3, 3, 64)
	drop.	(128, 7, 7)	0	-
7	conv.	(128, 7, 7)	147584	(3, 3, 128)
	pool.	(128, 3, 3)	0	(2, 2)
8	flatt.	(1152)	0	-
	drop.	(1152)	0	-
	dense	(1024)	4195328	-
	drop.	(1024)	0	-
	dense	(512)	524800	-
	drop.	(512)	0	-
	dense	(2)	1026	-

Table 2: CNN training parameters.

Parameters	Values
Epochs	50
Learning_rate	0.01
Decay	$\frac{learning\_rate}{epochs} = 0.0002$
Momentum	0.9
Batch_size	100

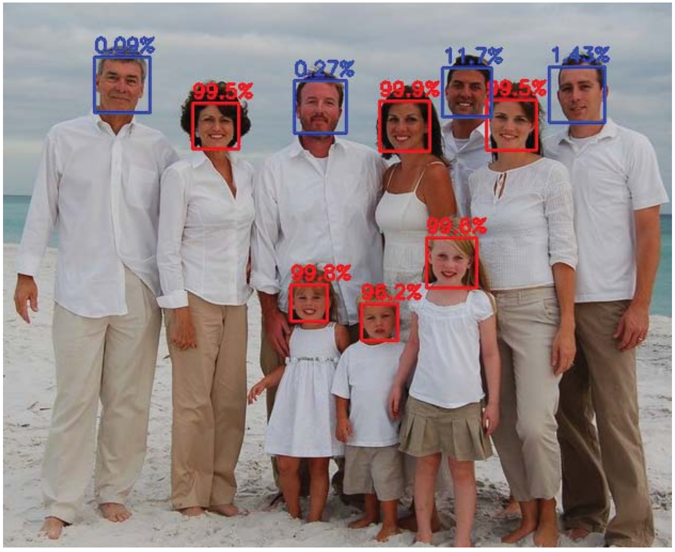
each iteration used 4 folds for training and 1 fold for testing. Table 3 summarizes the average accuracy of 10 runs (for our proposal) alongside the results obtained by other algorithms mentioned in the bibliography. In both evaluation conditions, our proposed CNN obtains a better result using LFW and a competitive result using Gallagher's dataset.

Table 3: Columns 2 and 3 show the accuracy for LFW [7] and Gallagher's dataset [8] respectively. This shows that the CNN's final accuracy is $\sim 95.42\%$ for the test set (LFW), and $\sim 91.48\%$ for the test set (Gallagher's dataset).

Autor	LFW $ACC(\%)$	Gallagher's $ACC(\%)$
Dago-Casas [8]	89.77	86.61
Fazl-Ersi [15]	91.59	**91.59**
Mansanet [4]	94.48	90.58
Ubunsa (Our)	**95.42**	91.48

4.3 Real-Time Recognition

To evaluate the real-time performance of the system, the images captured by a web camera with a frequency of 6 frames per second, were used. On the average, the system classified each ROI in $20msec$, which is adequate for real-time processing. A practical demonstration is shown in `https://youtu.be/Tfi-p2K9NBg`.



drive

Figure 4. Example of the output of the gender recognition system. The red BBs are those classified as female by the CNN, while the blue BBs are classified as male. In all cases the score is displayed.

5 Conclusions

In this work we implemented a gender recognition system, proposing a CNN architecture which we called Ubunsa. This architecture was implemented in Python using the Keras [12] and the Theano [13, 14] libraries. It was trained and tested using Labeled Faces in the Wild (LFW) [7] and Gallagher's dataset [8].

We evaluated the classification performance of the CNN through a k-fold cross validation with $k = 5$. The overall average accuracy was $\sim 95.42\%$ and 91.48% for the training and the testing datasets, respectively.

A distintive characteristic of the CNN in this system is that it separates male person from female person images without the aid of a pre-classification stage, and without the need of special tuning steps or initial conditions, making it more straightforward than other CNN-based solutions.

References

[1] Baback Moghaddam and Ming-Hsuan Yang. Learning gender with support faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):707–711, May 2002.

[2] Juan Bekios-Calfa, Jose M. Buenaposada, and Luis Baumela. Revisiting linear discriminant techniques in gender recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(4):858–864, April 2011.

[3] J. G. Wang, Hee Lin Wang, Myint Ye, and W. Y. Yau. Real-time gender recognition with unaligned face images. In *2010 5th IEEE Conference on Industrial Electronics and Applications*, pages 376–380, June 2010.

[4] Jordi Mansanet, Alberto Albiol, and Roberto Paredes. Local deep neural networks for gender recognition. *Pattern Recogn. Lett.*, 70(C):80–86, January 2016.

[5] J. v. d. Wolfshaar, M. F. Karaaba, and M. A. Wiering. Deep convolutional neural networks and support vector machines for gender recognition. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 188–195, Dec 2015.

[6] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004.

[7] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[8] A. Gallagher and T. Chen. Understanding images of groups of people. In *Proc. CVPR*, 2009.

[9] Carlos Eduardo Thomaz and Gilson Antonio Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image Vision Comput.*, 28(6):902–913, June 2010.

[10] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1090–1104, October 2000.

[11] C. I. Orozco, M. E. Buemi, and J. J. Berlles. New convolutional neural network architecture for pedestrian detection. In *International Conference on Pattern Recognition Systems (ICPRS-17)*, July 2017.

[12] François Chollet. Keras. https://github.com/fchollet/keras, 2015.

[13] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[14] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[15] E. Fazl-Ersi, M. E. Mousa-Pasandi, R. Laganire, and M. Awad. Age and gender recognition using informative features of various types. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5891–5895, Oct 2014.