Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 132 (2018) 76–83

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2018)

Deep Learning on Binary Patterns for Face Recognition

A Vinay, Abhijay Gupta, Aprameya Bharadwaj, Arvind Srinivasan,
K N Balasubramanya Murthy, S Natarajan

Center for Pattern Recognition and Machine Intelligence
PES University, Bengaluru, India

Abstract

In this paper an efficient and robust method for real-time face recognition is proposed. As a part of pre-processing to remove noise and unwanted features, a filter is applied to the images of standard datasets. Subsequently binary patterns are extracted from these images which are further fed into multilayer perceptron to classify images. The proposed method is tested on four benchmark datasets namely FACES94, FACES95, FACES96 and Grimace which pose challenges in illumination, pose, expression, head scale and rotation. This method delivers accuracies in the neighbourhood of 91% when tested on these datasets. Our methodology was further extended to embedded systems like Raspberry Pi 3 to give it more real-time and practical scenario. This test gave promising results and proved the model to be efficient and useful in day to day scenarios for face recognition.

Keywords: Local Binary Patters; Gaussian Filter; MLP Classifier; Artificial Neural Network; Bag of Words; Raspberry Pi 3

1. INTRODUCTION

Face recognition is a popular field for researchers all over the world. Over the years, there have seen big advancements in its technology; but there is still scope for so much more. The most commonly used method in face recognition is to extract the important facial features from the images. Then, different images, or faces, can simply be identified by these features. Current technologies perform really well in ideal situations. But in practical applications, difficult challenges like illumination changes, head movements, expression changes, and variations in background are faced.

Corresponding author email: arvind.srini.8@gmail.com

10.1016/j.procs.2018.05.164

In the mid-19th century, the English began taking “mug shots” of miscreants to introduce some safety measures. These photos, however, had to be matched with other faces by humans with their naked eyes. In the later 19th and early 20th centuries, a French police official introduced some guidelines to match faces. These rules could be used to compare facial features systematically. The field of Face recognition has grown immensely since it became computerized in the late 20th century. Now, most of the important features in images can be extracted, and persons can be classified accurately.

Its main application has always been security. Government agencies all over the world use it to identify criminals. It has become so effective, that even home security systems use it to identify thieves. But its applications do not end here. Some countries have begun to use it along with other biometrics in their digital identification systems. Some banks have also started to use face recognition to validate online transactions.

In our method, a pre-processing step is used to modify the original images of some standard datasets by highlighting their important regions. Then, features are extracted from these images using a binary feature detector-descriptor. These features are aggregated and then classified using a feed forward neural network. The above method was implemented using a Raspberry Pi. The results were tabulated and studied.

2. RELATED WORK

The method proposed in [1] is the support vector machine classifier which maps data points in space. These data points are to be classified by analyzing their degrees of similarities. When two points have similar attributes, the distance between them in space is small. Classes are assigned to these data points by maximizing the distance between points of different classes. Although this method is inherently a linear classifier, using different functions, the classifier can be made non-linear in nature. When the SVM classifier was used along with the feature extractor that was used in the proposed method, it was observed that the accuracies were inferior to the accuracies obtained when the Multi-Layer Perceptron Classifier was used.

The method proposed in [2] is the convolutional neural network which performs very well on images and videos. It consists the convolutional layers, pooling layers and the fully connected layer. In the convolutional layers, the convolutional function is applied on the input data and the result is transmitted to the next layers. The pooling layers then combine all these outputs together. The fully connected layer connects all the neurons in a layer together. This method is used extensively to perform multiclass classification.

The scale invariant feature transform [3] is one of the most popular methods used in image processing. It is used to detect features in images and then describe them. In particular, it detects local features. Its popularity stems from the fact that it is both rotation and scale invariant. It works in the following way. First, it uses the Difference of Gaussian to identify the key points in the image. Here, different blurred images of the original image are subtracted in order to obtain the important points in the image. Then, some important aspects of the key points like their location and scale are determined. Finally, to minimize the distortion in the image, the gradients around each of the identified key points are measured. Although this feature extractor yields good accuracies, it consumes a lot of space due to the fact that it stores 128-dimensional feature vectors. Hence, using it in applications like embedded systems where space is a constraint becomes impractical.

The Binary Robust independent elementary features [4] is a feature descriptor that was designed to improve efficiency. Methods like the one proposed by [3] are very slow and space inefficient, for the reason that they use a vector of many large floating-point numbers to describe each key point in the image. The method proposed by [4] tries to bypass this step by describing the features using binary strings. This change causes a huge improvement in the speed of the overall process. This is a result of the fact that comparing two strings using the Hamming distance is much less compute intensive that using the Euclidean distance and working with floating point numbers.

The extreme learning machine [5] is a unique classifier, in the sense that it uses a very different approach to learning. The first hidden layer is randomly connected and the output layer is supervised. So, the parameters of the nodes in the first hidden layer along with the weights connecting these nodes to other layers are randomly assigned. The weights connecting the output layer are learned. Although this classifier isn't a very popular classifier in the deep learning community, it has a high computational efficiency. So, it is much faster than other deep learning architectures, especially ones that follow the backpropagation mechanism. When the ELM classifier is used along with the feature extractor that has been used in the proposed method, the accuracies are in the range of 70-80%. This is considerably poorer than the accuracies that are obtained from the proposed method.

3. PROPOSED METHOD

3.1. Image Pre-processing

The Gaussian Filter [6] is an image pre-processing technique that's used to reduce noise and distortion in the image. Here, the images that are worked on are gray-scaled and these are 2-D matrices. This transformation is given by:

$$g(x,y)=\frac{1}{2\pi\sigma^2}\cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

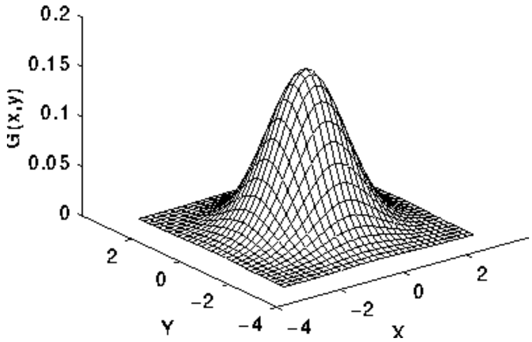

Fig. 1. Gaussian Distribution in 2-Dimension[7]

Here, x and y are the coordinates of the pixel in the image and σ is the standard deviation of the Gaussian distribution created around the pixel.

Using this method as a pre-processing step makes the images illumination invariant, reduces the number of unwanted features in the image and hence decreases computational times. This is how an image looks after applying this filter:

Fig. 2. Image before and after applying Gaussian Filter

Local Binary Patterns [8] is a descriptor used to extract binary features from an image. It is mainly used for texture analysis [9], but hereits applications in the field of face recognition is seen. First, the image is divided into many subsections. For a given pixel in a subsection, it is analyzed by comparing it to its 8 neighbors in a circle around it. If the pixel being analyzed is brighter than one of its neighbors a '0' is appended to an empty bit string, otherwise a '1' is appended. By following this method an 8-bit binary number is obtained. Histograms are then calculated for the subsection. Each histogram is a 256-dimensional feature vector. When all these vectors from all the subsections are concatenated, a feature vector for the entire image is obtained.The histogram is given by:

$$H_i = \sum_{x,y} I\{f_l(x,y) = i\}, i = 0,\ldots,n-1 \tag{2}$$

where $f_l(x,y)$ is the LBP labeled image and n is the number of different labels produced by the LBP operator.When A is true, $I\{A\}$ is 1 and when A is false, $I\{A\}$ is 0.The Local Binary Patterns descriptor has been shown to perform well even when there are significant changes in alignment of images [10].

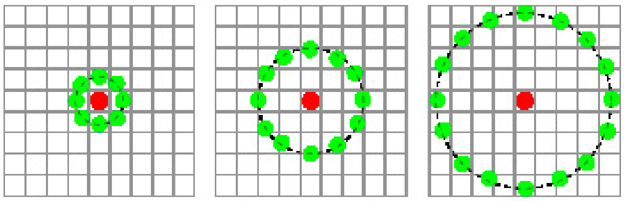

Fig. 3. Detecting Features in LBP [11]

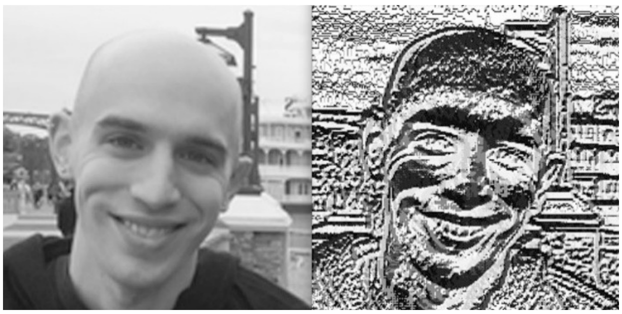

Fig. 4. Image before and after extracting features using LBP [12]

3.3. Feature Aggregation

Once the features are from an image, a feature aggregator is used. This makes it easier to classify the images. The 'Bag of Words' [13] is one such feature aggregator. It has a visual vocabulary that it uses to represent the features in the image. Each feature that has been extracted is mapped to its closest word in the vocabulary. Now, the images are represented in terms of words. Once all the different words from the image are obtained, a histogram of the frequency of these obtained words is built. This model is then split into two parts – the training set and the test set. The training set contains 75% of the data and the test set contains the remaining 25% of the data. After this, any machine learning classifier can be used for classification. The classifier is first trained using the training set and then its accuracy is tested using the test set.

The multi-layer perceptron[14] is a type of feedforward artificial neural network [15] that uses the gradient descent algorithm [16]. This is used to determine the errors in each neuron of the architecture. The method also uses back-propagation [17]. This algorithmuses these errors to minimize a loss function to keep improving the model. In this method, the error function is given by:

$$E = \frac{1}{2n} \sum_{x} \| (y(x) - y'(x)) \|^2 \tag{3}$$

Where n is the number of training samples, y'(x) is the target output on input x and y(x) is the computed output on input x. The partial derivate of the error function with respect to the target output is given by:

$$\frac{\partial E}{\partial y'} = y' - y \tag{4}$$

To identify the ideal weight to be assigned between two neurons, it is continuously changed by:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \tag{5}$$

Here, η is the learning rate and w_{ij} is the weight between two neurons.The activation function used here is the 'relu' function. This is given by:

$$A(x) = max(0, x) \tag{6}$$

Here, x is the input to the neuron. With this, data points that are not linearly separable can be easily distinguished. This classifier is also quite good for practical applications that require real time processing.

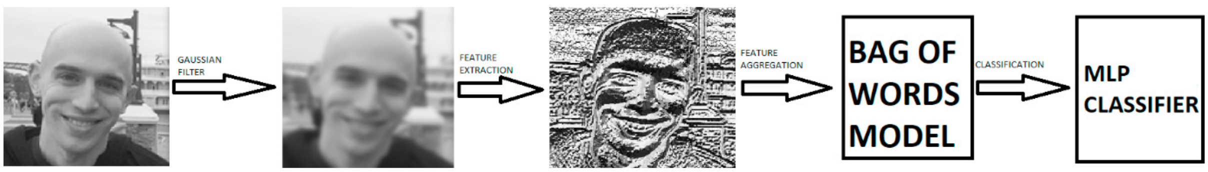

Fig. 5. Pipeline for our proposed method

4. DATASETS

The datasets [18] used here are:

1) Faces94 Dataset - This dataset contains 20 images for 153 individuals each hence the total number of images being 3060. The dimensions of each image is180x200. There is no background variation as all photos are taken against a green background. However, there is a large expression variation between the images.

Fig. 6. Images of Faces94 Dataset [18]

2) Faces95 Dataset - This dataset contains 20 images for 72 individuals each hence the total number of imagesbeing 1440. The dimensions of each image is 180x200. There is no background variation as all photos are taken against a red background. However, there is a large variation in illumination and head scale between the images.

Fig. 7.Images of Faces95 Dataset [18]

3) Faces96 Dataset - This dataset contains 20 images for 147 individuals each hence the total number of imagesbeing 2940. The dimensions of each image is196x196. There is a large variation in background, illumination and head scale between the images.

Fig. 8. Images of Faces96 Dataset [18]

4) Grimace Dataset - This dataset contains of 18 different people with 20 pictures each. The size of each image is 180x200. The background is plain. There is little head scale variation and change in lighting between images. There is a large variation of expression and motion of face in the images.

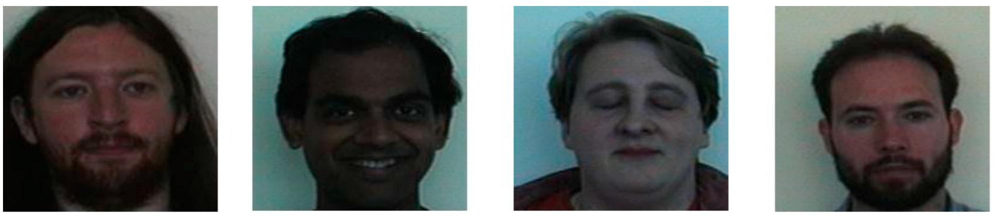

Fig. 9 Images of Grimace Dataset [18]

5. RESULTS

All our programs were executed on an Intel i7 machine that uses Windows 10. The language that was used to execute our programs was Python 3.5. Along with some inbuilt Python packages like os, random and time, Opencv3 (3.2.0), scikit-learn (0.18.1), numpy (1.11.0) were used.

Table 1. Results for proposed method on the standard datasets

DATASET	PRECISION	RECALL	F1-SCORE	ACCURACY
Faces94	0.95	0.94	0.94	0.936
Faces95	0.92	0.91	0.92	0.906
Faces96	0.93	0.92	0.92	0.916
Grimace	0.97	0.97	0.97	0.966

From the above table it is seen that when this method is tested on the Faces94 dataset it gives an accuracy of 93%. This shows that even though the images of the dataset have a large variation in expressions, the method still delivers high accuracy. When tested on the Faces95 dataset, an accuracy of 90% is obtained. Faces95 dataset proposes the challenge of variation in illumination and head scale. Obtaining this accuracy level for this dataset shows that the proposed method is also illumination and rotation invariant. Next, when tested on the Faces96 dataset, an accuracy of 91% is obtained. The images of this dataset are taken in complex backgrounds and also the problem of head scale and illumination variation. The proposed method still delivers a high accuracy here and shows that complex backgrounds for faces don't affect the accuracy. This is very important for real time applications as a studio environment can't be present for images in practical applications. Finally, this method was tested on the grimace dataset, which is a smaller dataset when compared to the others. This dataset proposes the challenge of expression variation and motion in faces. The proposed method still provided high accuracies. Hence, the proposed method overcame all the hurdles imposed by these datasets. Thismethod was then extendedto embedded systems, namely Raspberry Pi 3 and observed that the accuracy has minimal variation and also the computational time is very less. Hence, the proposed method has real time applications for face recognition.The Faces95 dataset was used to test this. This also shows that expensive hardware is not needed for obtaining high accuracies using this method.

Table 2. Tests on Raspberry Pi

Method	Time taken for the Test set	Time taken for a single image
Feature Extraction	32 seconds	0.23 seconds
MLP classification	2.8 seconds	0.03 seconds

The proposed method was also compared with some classic methods for face recognition from [19]. However, a direct comparison can't be made as the hardware used to run it may be different.

Table 3. Comparison of proposed method with some standard methods

Method	Faces94 dataset	Faces95 dataset	Faces96 dataset	Grimace dataset
PCA [20]	72.10%	69.87%	70.95%	74.79%
LDA [21]	79.39%	76.61%	78.34%	81.93%
LBP [22]	85.93%	80.47%	84.145	86.45%
Proposed Method	93.6%	90.6%	91.6%	96.6%

6. CONCLUSIONS

The datasets used to test our proposed method pose many challenges such as pose, illumination, translation, scale and rotation. The proposed method delivers a high accuracy rate even when these challenges are posed. Upon testing this method on an embedded system, it seen that it is also time efficient and can be used in real time scenarios. Even though the size of each dataset varies, the accuracy does not very much.

All these factors put together make the proposed method a robust and efficient method for face recognition. The Local Binary Patterns algorithm to extract features from images is shown to perform poorly when the images are subjected to severe changes in illumination [23]. It is also quite sensitive to the noise in images. To overcome this,the images need to be processed before the features are extracted. The Gaussian Filter solves this problem to someextent, but, it underperforms when the quality of the images is poor.

This method can be further developed to be used in environments which are less sensitive to illumination changes such as home security systems. In the future the proposed approach can be enhanced further to solve the problem of occlusion in photos.

REFERENCES

[1] Suykens, Johan AK, and JoosVandewalle. "Least squares support vector machine classifiers." *Neural processing letters*9, no. 3 (1999): 293-300.

[2] Lawrence, Steve, C. Lee Giles, Ah Chung Tsoi, and Andrew D. Back. "Face recognition: A convolutional neural-network approach." *IEEE transactions on neural networks* 8, no. 1 (1997): 98-113.

[3] Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision*60, no. 2 (2004): 91-110.

[4] Calonder, Michael, Vincent Lepetit, Christoph Strecha, and Pascal Fua. "Brief: Binary robust independent elementary features." In *European conference on computer vision*, pp. 778-792. Springer, Berlin, Heidelberg, 2010.

[5] Huang, Guang-Bin, Qin-Yu Zhu, and Chee-Kheong Siew. "Extreme learning machine: theory and applications." *Neurocomputing* 70, no. 1-3 (2006): 489-501.

[6] Adini, Yael, Yael Moses, and Shimon Ullman. "Face recognition: The problem of compensating for changes in illumination direction." *IEEE Transactions on pattern analysis and machine intelligence* 19, no. 7 (1997): 721-732.

[7] www.homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm

[8] Ahonen, Timo, Abdenour Hadid, and Matti Pietikainen. "Face description with local binary patterns: Application to face recognition." *IEEE transactions on pattern analysis and machine intelligence* 28, no. 12 (2006): 2037-2041.

[9] Ojala, Timo, Matti Pietikainen, and TopiMaenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." *IEEE Transactions on pattern analysis and machine intelligence* 24, no. 7 (2002): 971-987.

[10] Ahonen, Timo, Abdenour Hadid, and Matti Pietikäinen. "Face recognition with local binary patterns." In *European conference on computer vision*, pp. 469-481. Springer, Berlin, Heidelberg, 2004.

[11]www.en.wikipedia.org/wiki/Local_binary_patterns

[12] www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv

[13]Nowak, Eric, Frédéric Jurie, and Bill Triggs. "Sampling strategies for bag-of-features image classification." In *European conference on computer vision*, pp. 490-503. Springer, Berlin, Heidelberg, 2006.

[14] Ruck, Dennis W., Steven K. Rogers, Matthew Kabrisky, Mark E. Oxley, and Bruce W. Suter. "The multilayer perceptron as an approximation to a Bayes optimal discriminant function." *IEEE Transactions on Neural Networks* 1, no. 4 (1990): 296-298.

[15] Hopfield, John J. "Artificial neural networks." *IEEE Circuits and Devices Magazine* 4, no. 5 (1988): 3-10.

[16] LeCun, Yann, D. Touresky, G. Hinton, and T. Sejnowski. "A theoretical framework for back-propagation." In *Proceedings of the 1988 connectionist models summer school*, pp. 21-28. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988.

[17] Sutskever, Ilya, James Martens, George Dahl, and Geoffrey Hinton. "On the importance of initialization and momentum in deep learning." In *International conference on machine learning*, pp. 1139-1147. 2013.

[18]Ebner, Natalie C., Michaela Riediger, and Ulman Lindenberger. "FACES—A database of facial expressions in young, middle-aged, and older women and men: Development and validation." *Behavior research methods* 42, no. 1 (2010): 351-362.

[19] Ahmad, Faizan, Aaima Najam, and Zeeshan Ahmed. "Image-based Face Detection and Recognition: based Face Detection and Recognition: based Face Detection and Recognition." *International Journal of Computer Science* 9, no. 6 (2012).

[20] Yang, Jian, David Zhang, Alejandro F. Frangi, and Jing-yu Yang. "Two-dimensional PCA: a new approach to appearance-based face representation and recognition." *IEEE transactions on pattern analysis and machine intelligence* 26, no. 1 (2004): 131-137.

[21]Chen, Li-Fen, Hong-Yuan Mark Liao, Ming-Tat Ko, Ja-Chen Lin, and Gwo-Jong Yu. "A new LDA-based face recognition system which can solve the small sample size problem." *Pattern recognition* 33, no. 10 (2000): 1713-1726.

[22] Guo, Zhenhua, Lei Zhang, and David Zhang. "A completed modeling of local binary pattern operator for texture classification." *IEEE Transactions on Image Processing* 19, no. 6 (2010): 1657-1663.

[23] Hadid, Abdenour. "The local binary pattern approach and its applications to face analysis." In *Image Processing Theory, Tools and Applications, 2008. IPTA 2008. First Workshops on*, pp. 1-9. IEEE, 2008.