STM JOURNALS

Journal of
Image Processing & Pattern Recognition Progress

ISSN: 2394-1995
Volume 11, Issue 1, 2024
January—April
DOI (Journal): 10.37591/JoIPPRP

https://journals.stmjournals.com/JoIPPRP

Review JoIPPRP

Facial Recognition System Utilizing Real-time Deep Learning Techniques

V. Suganthi[1,*], M. Yogeshwari[2]

Abstract

This research introduces an openly accessible deep learning-based framework designed for facial recognition. The system encompasses five key stages: face segmentation, detection of facial features, face alignment, embedding, and classification. Deep learning methods are employed for the extraction of fiducial points and embedding within the system. For the classification task, a Support Vector Machine (SVM) is utilized due to its efficiency in both training and inference phases. Notably, the system achieves a facial features detection error rate of 0.12103, demonstrating proximity to state-of-the-art algorithms. Additionally, it achieves a remarkable face recognition error rate of 0.05. By integrating advanced deep learning methods for extracting fiducial points and embedding, the system gains enhanced robustness and effectiveness. Importantly, the system is capable of real-time operation, providing timely and efficient facial recognition. This study marks a noteworthy stride in advancing facial recognition systems, demonstrating commendable accuracy and real-time functionality.

Keywords: Facial recognition, advanced learning, facial attributes, neural computing, pattern identification

INTRODUCTION

The impressive progress in computational capabilities, coupled with the widespread accessibility of extensive data and the proliferation of social media platforms, has played pivotal roles in the swift advancement of facial recognition technology. Originally designed for forensic investigations, facial recognition algorithms have now become a ubiquitous aspect of our daily lives, seamlessly integrated into various applications such as social networks, mobile device unlocking systems, and more. This study introduces a novel facial recognition system that harnesses the power of deep learning, specifically leveraging Convolutional Neural Networks (CNNs), a type of sophisticated neural network tailored for computer vision tasks. By utilizing specialized convolutional layers as feature extractors, CNNs excel in tasks like facial recognition [1].

The system we propose is broken down into five essential components, each playing a vital role in how it operates. These components include face segmentation, the use of deep learning for detecting facial features, face alignment, codification, and subject identification through Support Vector Machine (SVM). For a detailed understanding of each stage, one particularly crucial aspect is the detection of facial key points, also known as facial landmarks or fiducial points. This is not just important for our proposed system but has

***Author for Correspondence**
V. Suganthi
E-mail: suganthi@cttewc.edu.in

[1]Assistant Professor, Department of Computer Science, Chevalier T. Thomas Elizabeth College for Women, Chennai, Tamil Nadu, India
[1]Research Scholar, Department of Information Technology, School of computing sciences, Vels Institute of Science, Technology & Advanced Studies, Chennai, Tamil Nadu, India
[2]Assistant Professor, Department of Information Technology, School of computing sciences, Vels Institute of Science, Technology & Advanced Studies, Chennai, Tamil Nadu, India

Received Date: February 16, 2024
Accepted Date: March 08, 2024
Published Date: April 03, 2024

Citation: V. Suganthi, M. Yogeshwari Facial Recognition System Utilizing Real-time Deep Learning Techniques. Journal of Image Processing & Pattern Recognition Progress. 2024; 11(1): 14–20p.

implications for various other applications as well. Additionally, we pay special attention to the classification step, a pivotal component responsible for subject identification, making it integral to the overall functioning of the system.

RELATED WORK

In the present-day scenario, the commonly used methods for detecting and tracking facial features, along with recognizing individuals, predominantly hinge on machine learning. These methods can be broadly categorized as either shallow or deep, reflecting the diverse approaches available in the field.

Shallow methods employ descriptors like Scale-Invariant Feature Transform (SIFT), Local Binary Patterns (LBP), and Histogram of Oriented Gradient (HOG) [2–4]. These descriptors are combined to create an overall face description using a pooling mechanism, such as the Fisher vector [5]. In simpler terms, these methods use specific features of a face, like its texture or gradient, and pool them together to form a comprehensive representation for recognition.

Convolutional Neural Networks (CNNs) are crucial components in modern deep learning techniques, acting as powerful tools for extracting nonlinear features. One notable instance is the Deep Face system, which utilizes a specific type of CNN called a Siamese network. This innovative architecture assesses the differences in distances between descriptors produced by CNN when pairs of facial images are input. During the training phase, the focus is on reducing the distance between descriptors of matching faces (i.e. the same person) and increasing the distances between descriptors of different faces (i.e. individuals with different identities). Deep Face takes this concept even further by incorporating a committee of CNNs in the preprocessing stage. This committee works to align facial images to a standard pose using a 3D model.

The newest update of the Deep Face [6] database features a remarkable 10 million individuals, each with 50 corresponding images. This showcases the remarkable scope and effectiveness of advanced deep learning techniques when it comes to managing massive amounts of data for facial recognition purposes.

Among the remarkable methods harnessing the power of deep neural networks is FaceNet. This approach stands out by leveraging a vast dataset featuring 200 million identities and 800 million image pairs to train a convolutional neural network (CNN), following the footsteps of previous successful studies. FaceNet's superiority has been evident in its performance on well-known datasets like Labeled Faces in the Wild (LFW) and YouTube Faces (YTF). Currently, two CNN-based systems, DeepFace and FaceNet, stand out in the literature for their exceptional accuracy. It is important to acknowledge that these models are trained on private datasets comprising millions of images sourced from social media platforms, surpassing the scale typically seen in datasets used for research purposes.

In order to overcome this limitation and facilitate the creation of a mobile-friendly facial recognition system that is available for public use, a team of researchers introduced OpenFace in a separate study. This versatile library draws inspiration from DeepFace and FaceNet, but with a specific emphasis on accessibility. In this current study, the authors propose a deep learning-based classification system similar to OpenFace. However, their system is specifically tailored for real-time usage. To ensure efficient performance, the architecture utilizes a minimal number of parameters. The classification process employs a Support Vector Machine (SVM) along with a linear kernel [7, 8].

Unlike OpenFace, which uses deep learning to detect facial features, the proposed system sets itself apart by relying on public datasets for training and validation and being open source under the MIT license. This sets it apart even more from DeepFace and FaceNet [9–11].

PROPOSED SYSTEM

In this section, we provide a detailed overview of the proposed system, breaking down the process into five distinct blocks, as depicted in Figure 1.

Figure 1. Block diagram of the proposed system.

The aim is to improve understanding and simplify the implementation process. Each block integrates a blend of computer vision and machine learning techniques specifically designed to accomplish its assigned task.

Face Detection

To identify an individual in an image, the initial step is to locate their face. Although the traditional Viola-Jones technique from 2004 has seen widespread use, there is a growing preference for more recent and sophisticated methods leveraging deep learning. In our strategy, we depend on an approach that melds Histogram of Oriented Gradient (HOG) with Max-Margin Object Detection (MMOD), implemented through the dlib library [12].

HOG is like a super-smart descriptor that can handle changes in scale, rotation, and lighting [13]. It has been the go-to tool for many image processing and computer vision applications. Now, MMOD, which works kind of like an SVM, is trained in the HOG feature space to be a professional at detecting objects with super high accuracy [12]. This dynamic duo helps us pinpoint and segment the face, getting it ready for the next step in our facial recognition journey.

Now, let us dive into the next step where we enlist the help of a regressor to pick out important facial key points like eyes, eyebrows, nose, lips, and more. There are numerous ways to tackle this task, but we prefer methods rooted in computer vision as they tend to be less costly and invasive. In the current landscape, algorithms powered by deep learning tend to deliver the best results.

Facial Extraction

In our experiment, we put Multi-Layer Perceptron (MLP) and Convolutional Neural Networks (CNN) to the test. These are like the brainy assistants that we are training to pinpoint those crucial facial features accurately. Let us see how they perform:

1. *Multi-Layer Perceptron Neural Network:* Let us unravel the complexities of the Multi-Layer Perceptron (MLP) Neural Network.
   Imagine a network with two layers, adorned with N_1 neurons in the hidden layer and N neurons at the output. For an input 'x' belonging to the real numbers, denoted as $x \in R^{(C*H*W)}$, representing an image with C channels, height H, and width W, the output unfolds through a sequence of operations:

   $$h = W_{xh}{}^{x} + w_h,\ z = \sigma(h),\ y = W_{zl}{}^{z} + w_l, \quad (1)$$

   Here, $\boldsymbol{W}_{xh} \in R^{(C_* H_* W)*N1}$ and $\boldsymbol{w}_h \in R^{N1}$ are the weights at the input layer. The function $\phi(\cdot)$ is a differentiable nonlinear function. Additionally, W_{zl}$\in \mathbb{R}^{N_1 \times N}$ and w_{l}$\in \mathbb{R}^{N}$ are the weights at the output layer.
   Throughout training, the network endeavors to minimize a cost function related to its parameters. Due to the inherently non-linear nature of the problem, it is strongly advised to optimize the function by employing small batches from the training set [1]. This approach significantly enhances the network's performance and facilitates improved learning capabilities [1].
2. *Convolutional Neural Networks:* MLP networks usually overlook spatial information in images to reduce the number of parameters. In contrast, CNNs utilize filters to detect local structures, like edges in images. This makes CNNs particularly effective in a range of image processing applications and areas with high-dimensional data and rich spatial structures. Significantly, CNNs accomplish this task with lower computational costs and a reduced number of parameters compared to MLPs. The fundamental elements of CNNs commonly involve convolutional layers and pooling layers.

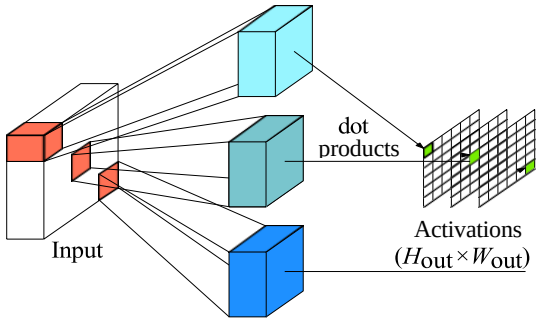

Figure 2. Convolutional layer.

CNNs function with tensors instead of vectors, and they achieve this through convolution, which is the reason for their name, CNN networks. A convolutional layer comprises F filters (f_f) of the same size, characterized by width (W_f), height (H_f), and the same number of input channels. A tensor incorporating all parameters can be represented as $\boldsymbol{F} \in R^{F \times C \times H_f} * W_f$. It is common to set $W_f = H_F = K$ (kernel size) with values like K=3, 5, or 7.

The inner product between each filter f_f and each position of the image generates a point in the features map a_f. These maps are then stacked to form the output a, which is a tensor of dimensions $\boldsymbol{a} \in R^{F \times H_0 \times W_0}$. The width and height of this output depend on the size of the input image, the padding (P), and the stride (S). This entire process is depicted in Figure 2.

Commonly, CNN architectures are designed with blocks that include convolutional layers, a nonlinear activation function (such as ReLU [14]), a regularization layer (like dropout [15] or batch norm [16]), and intermittent pooling layers placed between these blocks. At the conclusion of the architecture, there is an inclusion of an MLP network, also recognized as a Fully Connected (FC) layer, or a global average pooling layer [17], to produce the final output. The determination of hyper parameters, such as the number of layers, kernel size, stride, etc., holds significant importance.

Building a ConvNet does not follow a one-size-fits-all recipe, much like many machine learning algorithms. However, certain architectures have gained popularity, such as LeNet, AlexNet, VGG, ResNets, and DenseNets [1]. It is notable that ResNets, in particular, find extensive use in state-of-the-art computer vision algorithms due to their simplicity and high generalization capability. In this work, we opt for the ResNet CNN architecture because of its mentioned properties.

Face Alignment

The aim of this step is to standardize the input for the classifier, thereby simplifying the classification model. Given the known location of facial features, the process involves aligning faces in the image to ensure that the nose, eyes, and mouth are as centered as possible. This alignment is achieved using an affine transformation, a technique that preserves the relative positions of facial landmarks, ensuring that parallel lines in the original image remain parallel after the transformation.

To implement this, the getAffine Transform function from the OpenCV library is employed. This function calculates the rotation and translation needed to align the original points with the desired ones, typically based on an average mask derived from points in the training set. Subsequently, the warpAffine function, also part of the OpenCV library, is utilized to apply the transformation, which includes scaling the resulting image.

Embedding

Having the aligned images in hand, the next step involves recognizing the person in the current image. However, directly feeding the classifier with the raw pixel values of the image is not an efficient approach. Instead, an embedding process is employed to optimize the recognition task's efficiency.

At this point, another deep learning technique comes into play to extract this array. Unlike previous networks, the training objective here is to minimize the triplet loss [11]. In each iteration, the network is presented with three images: two distinct images of the same individual and an image featuring a different person.

$$\sum_i^N \left[\left\| f(x_i^a) - f(x_i^p) \right\|_2^2 - \| f(x_i^a) - f(x_i^n) \|_2^2 + \alpha \right] +, \quad (2)$$

EXPERIMENTAL PROCEDURES AND RESULTS

Dataset

The dataset utilized for the detection of fiducial points originated from the 300 Faces In-The-Wild (300-W) challenges [18]. This dataset comprises photographs of individuals captured in diverse poses, lighting conditions, and facial expressions. Each image in the dataset is annotated with 68 fiducial points, as illustrated in Figure 3. Previously established datasets like LFPW [19], AFW [20, 21], and Helen [22] were subsequently re-annotated to adhere to the annotation pattern of the 300-W challenge, forming a comprehensive dataset. The training set encompasses 811 images from the LFPW dataset, 337 images from the AFW dataset, and 2000 images from the Helen dataset, resulting in a total of 3148 labeled images.

Evaluation Method

The predominant evaluation method for fiducial points detection involves calculating the square root of the mean squared error between the computed points and the ground truth, normalized by the intraocular distance [23]. To denote this method more formally, respectively, the points are as:

$\boldsymbol{l}^f=[x^f,y^f,\ldots,x^f,y^f]^T$ and $\boldsymbol{l}^g=[x^g,y^g,\ldots,x^g,y^g]^T$,

Figure 3. Dataset used fiducial marks detection.

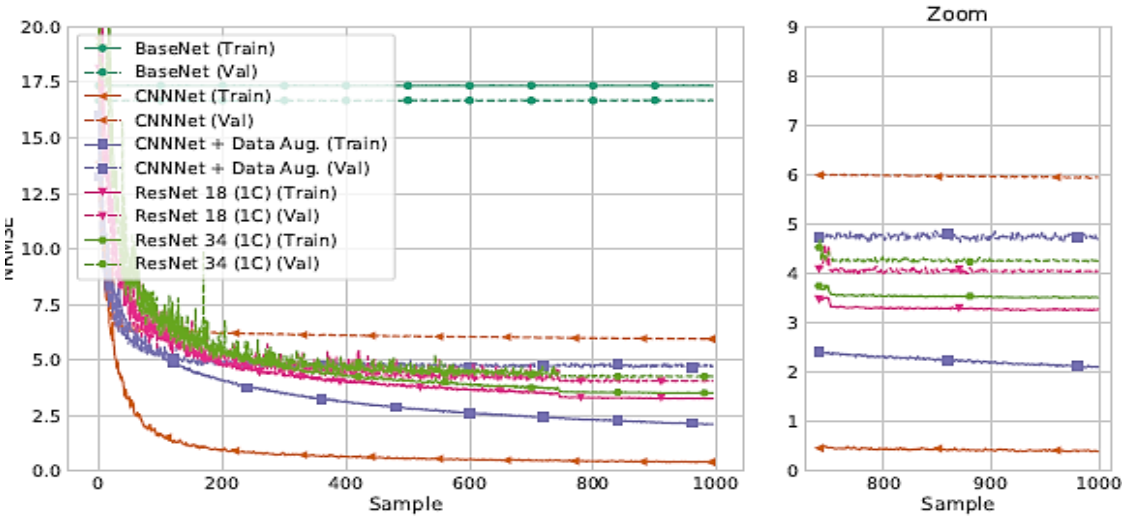

Figure 4. Results of NRMSE over iteration time for networks used in fiducial points extraction task.

The error is as:

$$\text{NRMSE} = \frac{\sum_{i=1}^{N} \sqrt{\left(x_i^f - x_i^g\right)^2 + \left(y_i^f - y_i^g\right)^2}}{d_{10} N} \quad (3)$$

Two Layer Neural Network Model

Initially, an MLP is constructed, referred to as Base Net, comprising a hidden layer with 300 neurons and ReLU activation [14]. The inputs consist of grayscale images stacked into an array ($\boldsymbol{x} \in \mathrm{R}^{256*256*1}$), amounting to a total of 19 million parameters. The training process involves utilizing the stochastic gradient descent (SGD) algorithm with a learning rate of 0.01 and a batch size of 32.

A MLP neural network tends to have many parameters due to high-dimensional inputs. As illustrated in Figure 4, although the network appears to learn something, it struggles to generalize because it fails to extract sufficient information from the image pixels. Consequently, the network predicts a pattern mask to minimize the mean error.

CONCLUSION

In this research, a modular system tailored for person recognition takes center stage, unraveling its functionality through five series of steps: face segmentation, facial features detection, face alignment, embedding, and classification. The incorporation of deep learning techniques plays a pivotal role in pinpointing facial fiducial points and optimizing the embedding process. The study attains optimal outcomes by introducing a tailored architecture, a residual network boasting 18 layers, resulting in a substantial reduction of parameters from 19 million to 700,000.

Amidst the employed preprocessing steps, encompassing face detection, fiducial points detection, face alignment, and embedding, the classification model emerges as a testament to simplicity and efficiency. The system not only showcases competitive accuracy, with a score of 0.95±0.13, but also operates seamlessly in real-time, clocking at 23.23±0.60 frames per second. Moreover, the system is made fully accessible under the MIT license, ensuring widespread usability and collaboration.

REFERENCES

1. Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016 Nov 10.
2. Sivic J, Everingham M, Zisserman A. “Who are you?”-Learning person specific classifiers from video. In 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009 Jun 20; 1145–1152.

3. Wolf L, Hassner T, Maoz I. Face recognition in unconstrained videos with matched background similarity. In IEEE CVPR 2011. 2011 Jun 20; 529–534.
4. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13. 2014; 740–755. Springer International Publishing.
5. Simonyan K, Parkhi OM, Vedaldi A, Zisserman A. Fisher vector faces in the wild. In BMVC. 2013 Sep 9; 2(3): 4.
6. Taigman Y, Yang M, Ranzato MA, Wolf L. Web-scale training for face identification. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2015; 2746–2754.
7. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2015; 1–9.
8. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229. 2013 Dec 21.
9. Amos B, Ludwiczuk B, Satyanarayanan M. Openface: A general-purpose face recognition library with mobile applications. CMU School of Computer Science. 2016 Jun; 6(2): 20.
10. Taigman Y, Yang M, Ranzato MA, Wolf L. Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2014; 1701–1708.
11. Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2015; 815–823.
12. King DE. Max-margin object detection. arXiv preprint arXiv:1502.00046. 2015 Jan 31.
13. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). 2005 Jun 20; 1: 886–893.
14. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM. 2017 May 24; 60(6): 84–90.
15. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014 Jan 1; 15(1): 1929–58.
16. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning. 2015 Jun 1; 448–456.
17. Lin M, Chen Q, Yan S. Network in Network. In Proc of the Int Conf on Learning Representation (ICLR), Banff, Canada. 2014 Apr; 1–10.
18. Sagonas C, Tzimiropoulos G, Zafeiriou S, Pantic M. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In Proceedings of the IEEE international conference on computer vision workshops. 2013; 397–403.
19. Belhumeur PN, Jacobs DW, Kriegman DJ, Kumar N. Localizing parts of faces using a consensus of exemplars. IEEE Trans Pattern Anal Mach Intell. 2013 Jan 15; 35(12): 2930–40.
20. Viola P, Jones MJ. Robust real-time face detection. Int J Comput Vis. 2004 May; 57: 137–54.
21. Zhu X, Ramanan D. Face detection, pose estimation, and landmark localization in the wild. In 2012 IEEE conference on computer vision and pattern recognition. 2012 Jun 16; 2879–2886.
22. Le V, Brandt J, Lin Z, Bourdev L, Huang TS. Interactive facial feature localization. In Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part III 12. 2012; 679–692. Springer Berlin Heidelberg.
23. Jesorsky O, Kirchberg KJ, Frischholz RW. Robust face detection using the hausdorff distance. In Audio-and Video-Based Biometric Person Authentication: Third International Conference, AVBPA 2001 Halmstad, Sweden, June 6–8, 2001 Proceedings 3. 2001; 90–95. Springer Berlin Heidelberg.
24. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998 Nov; 86(11): 2278–324.

