Disguised Facial Recognition Using Neural Networks

Conference Paper · July 2018

DOI: 10.1109/SIPROCESS.2018.8600440

5 authors, including:

Saumya Kumaar Saksena
University of Twente
15 PUBLICATIONS **214** CITATIONS
SEE PROFILE

Ravi M. Vishwanath
Indian Institute of Science Bangalore
7 PUBLICATIONS **51** CITATIONS
SEE PROFILE

Omkar S N
Indian Institute of Science Bangalore
164 PUBLICATIONS **4,104** CITATIONS
SEE PROFILE

Abrar Majeedi
University of Wisconsin–Madison
8 PUBLICATIONS **17** CITATIONS
SEE PROFILE

2018 IEEE 3rd International Conference on Signal and Image Processing

Disguised Facial Recognition Using Neural Networks

Saumya Kumaar, Ravi M. Vishwanath, S. N. Omkar
Indian Institute of Science
Bengaluru
e-mail: kumaar324@gmail.com;
ravi.vishwanath.m@gmail.com,
omkar@iisc.ac.in

Abrar Majeedi, Abhinandan Dogra,
National Institute of Technology
Srinagar, India
e-mail: abrarcs19@gmail.com;
abhinandan.dogra9086@gmail.com;
m.hanan2938@gmail.com

***Abstract*—In this paper, we present a real-time deep neural network architecture (called DiFRuNNT) for disguised face verification. The proposed model consists of two neural networks, first one being a convolutional neural network (CNN) that predicts 20 facial key-points in the image and the second neural network classifies the subject based on the angles and ratios calculated from the predicted points. The accuracies are 67.4% and 74.8% for prediction and classification respectively and the results have been compared with the state-of-the-art methodologies also.**

***Index Terms*—Convolutional Neural Networks, Facial Key-points, Disguised Face Identification, Angular Orientations.**

I. Introduction

Face verification or what is commonly referred to as face recognition has been one of the most well researched topic till current date. Facial recognition is a classic example of a modern real-world challenge and it is directly implemented in places that are required to be impenetrable to any kind of human intrusion.

Imagine a highly secured facility where human penetration is subjected to numerous tests before permitting access. Machines can do the same task without the interference of any humanly factors which usually result in errors or security flaws. The concept of face verification becomes utmost important in these areas but the problem cannot be tackled effectively if the faces are disguised. Here comes into picture, the role of our algorithm and it's effectiveness and robustness in complex background noise.

However, before attempting to solve this issue a lot of survey was done to identify the key concepts and fundamental problems in human face detection and recognition. An interesting approach to solving problems related to facial expression detection or face recognition itself, is by using facial key-points as feature vectors. For instance, Berretti *et. al.* [1] have used SIFT descriptors of auto-detected key-points for examining facial expressions. They trained a SVM for each facial expression and were able to achieve a classification rate of 78.43% on the BU-3DFE database. Sun *et. al.* [2] attempted to predict and analyse five facial key-points, the left eye center, right eye center, nose tip, left mouth corner and right mouth corner and they cascaded three levels of CNNs to perform a *coarse-to-fine* tuning of the predicted outcomes. Wang *et. al.* [3] have used histogram stretching of the input images and principal component analysis (PCA) on the stretched images to obtain the eigenfaces. In these so obtained eigenfaces, they performed their algorithm of mean patch searching to predict and analyse the left and right eye centres, for any input face image. Very recently, Shi *et. al.* [4] have shown how to use PCA and local binary patterns descriptor to process the data and they reviewed the outputs of various algorithms to that processed data like linear regression, tree based model, CNNs etc. for facial key-point detection. This particular paper provided a lot of insights to the various architectures for the same aspect of key-points.

However, to our knowledge, the methodology of using key-points for identifying disguised faces was not adopted and people still continued to use the existing face descriptors for the same. Yoshine *et. al.* [5] had suggested a method of *morphometrical matching* for identifying disguised faces, where they took 2D right oblique images (with three disguises per subject only) and superimposed them on 100 different subjects. They reported a difference of $2.3-2.8$mm for a match (which approximately comes to an offset of about 11 pixels or spatial units, depending upon the camera parameters they used). Singh *et. al.* [6] suggested an architecture called *2DLPGNN* which resulted in a commendable verification performance. But for the challenging scenarios of multi-disguise variations, the accuracy drops to 65.6% This research however, proved that existing algorithms are not effective enough to handle a substantial amount of disguise variation. Righi *et. al.* [7] conducted three different cognitive visual research-based experiments to observe the effects of disguises on observer's face recognition activity using natural images (in the wild) in which individuals had a mixture of eyeglasses and wigs as the only disguises. Their paper presented some fundamental cognitive techniques that humans use to disambiguate disguised faces which could be implemented using machine vision techniques to deploy more intelligent autonomous systems for disguised face disambiguation. But the method suggested by Dhamecha *et. al.* [8] proved to be state-of-the-art classification technique for disguised faces, when they reported an accuracy of 53%. Facial Key points based disguised face verification was first used in 2017 by Singh *et. al.* [9] where they proposed to use an existing architecture of spatial fusion convolutional network, which was suggested earlier by Pfister *et. al.* [10] for human pose estimation. They were however, able to significantly improve the previous state-of-the-art performance by a margin of 9%.

Deriving inspiration from the above mentioned works, we propose a experimental method for the task of disguised face recognition, using a convolutional neural network (CNN) and a standard neural network architecture for classification. The architecture so presented, has been made user-friendly in terms of deployment to real-world problems like intruder detection etc.

978-1-5386-6396-7/18/$31.00 ©2018 IEEE

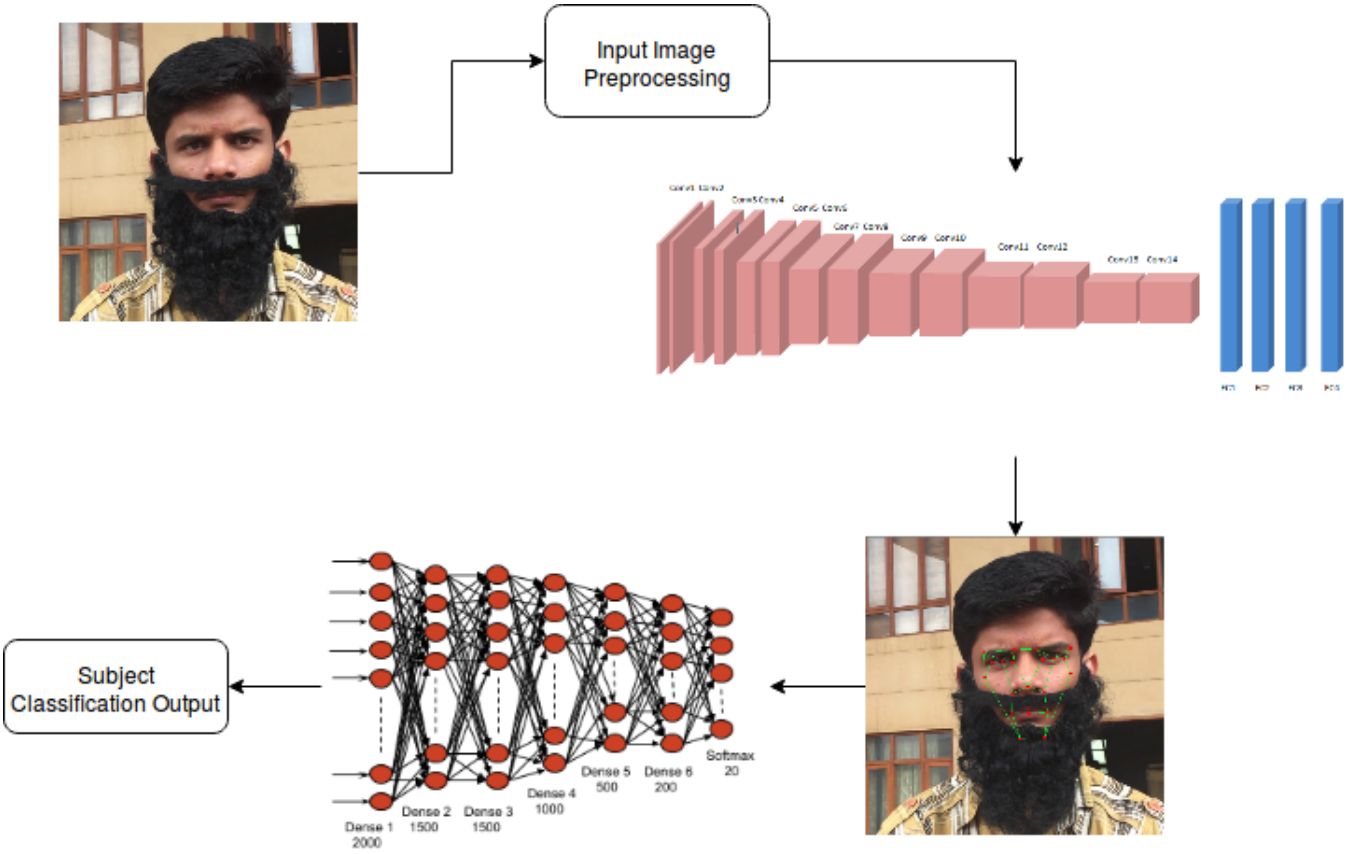

Fig. 1: Overall System Architecture

II. Methodology

The methodology and the overall architecture is explained vividly in the Fig. 1.

A. Dataset

A complex custom dataset (as suggested by Singh *et. al.* [9]) has been used to evaluate the performance of our algorithm. The dataset contains a mixture of simple, semi-complex and complex images, with the complexity pertaining to the background noise. This noise has not been altered to improve the accuracy of the algorithm, instead in order increase the robustness of the system, the complexity has been maintained. This allows our algorithm to learn and perform in non-favourable conditions also which might affect the orientation of faces or other aspects. Furthermore, there are 10 different disguises used in this experimental research, which could be seen in the Fig. 2. Our dataset comprises of a total of 4,000 images of 20 different subjects taken under varying illumination conditions. There are a total of 8 different backgrounds and the faces exhibit 5 different viewpoints ranging from -20° to 20°, as shown in the Fig. 3. The disguises are listed as follows :

1) Beard,
2) Cap,
3) Glasses,
4) Scarf,

Fig. 2: Different disguises used in the research. Note that the backgrounds are complex and semi-complex which make distinguishing more difficult.

5) Cap & Glasses,
6) Beard & Glasses,
7) Beard & Cap,
8) Scarf & Glasses,
9) Cap & Scarf,
10) Cap, Glasses & Beard.

B. Feature Extraction

With the most commonly observed disguises in the wild like glasses, scarf, beards, etc. or any combination of these, there are only a few facial features that remain unaltered for identification. For instance if a face is covered with a scarf, the cheekbone eyes and the eyebrows might still be visible,

Fig. 3: Facial orientations that differ from -20° to 20°, which allows the algorithm to learn more about the relative positions and the presence/absence of certain facial key-points.

Fig. 4: 20 facial key-points as suggested to use. The points are located strategically to help identify the faces better. This facial key-point mask-net structure could also be implemented to regular face disambiguation also.

with probably a faintly protruding nose tip and chin, depending upon the orientation of the scarf. An in-depth research of the concept of biometrics revealed that disguised faces could be identified using features like broadness, length, the relative angular placement of eyes, width and curvature of eyebrows, width of glabella, length of philtrum, length of nose and its thickness and the width of the chin and its distance from the mouth. Given a heavily disguised face (e.g. with scarf, glasses and cap, all at once) it is very difficult even for a human to identify the person in the image, because the human brain is not totally evolved to recognize disguised faces and hence the inherent high Bayes error for this problem.

End to end learning would have significant impact in this case, since we would have to train the model for every human being in every possible disguise to effectively identify them later. So, by training the system to extract 20 facial key points from a human face, disguised or not, makes the model extremely robust, as in this case, all we need a single undisguised image of a person to get the relative distances and angular information of the persons face.

C. Key-Point Prediction Model

This is the first part of the architecture and it comprises of a 14-layered CNN. The CNN outputs 40 values which are the (x,y) coordinates of the 20 facial key-points. The network has been trained on 3500 images and tested on 500 unseen (by the network) images, with input sizes of 227×227. The convolutional network is followed by 4 fully-connected layers and the predicted coordinates are overlaid on the input

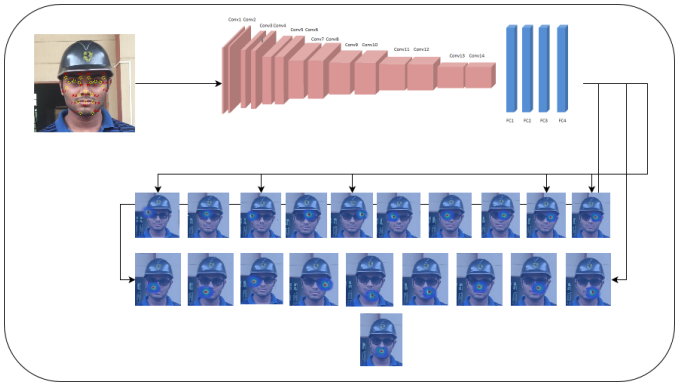

Fig. 5: Our DFR Key-Point Prediction Architecture with the generated heat maps. Although convolutional neural networks have been in extensive use for quite some time, there incredible unexplainable nature still leaves an open-ended research problem. The above network has been experimentally proven to work effectively the custom dataset with the generated heat-maps.

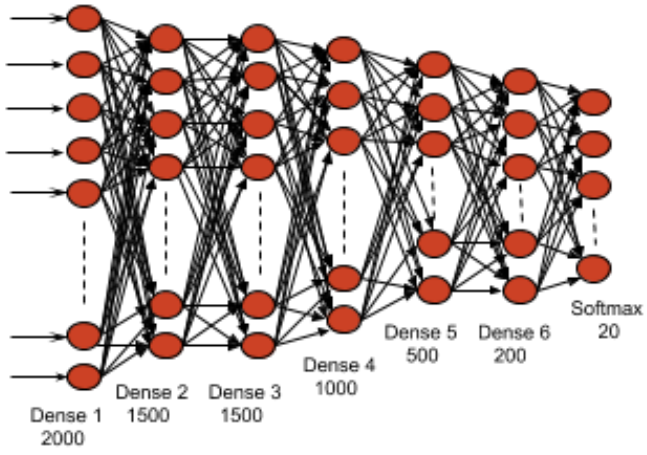

Fig. 6: The neural network architecture for facial classification based on the angles and ratios as calculated using the depictions shown in Fig. 7.

images for a better visualization. The architecture could be seen graphically in Fig. 5.

D. Face Classification Model

Once the previous CNN predicts the facial key-points, the relative angles and ratios between the points are calculated and the calculations are fed to a standard deep neural network with only fully connected layers for further classification. The angles taken into consideration are shown graphically in the Fig. 7 and the neural network employed in shown in Fig. 6.

Although SVMs (Support Vector Machines) have been used extensively [11], [12] and [13] for facial recognition there is a certain intuitive understanding of using neural networks for the same. The following architecture is proven experimentally to outperform SVMs in the concerned domain.

III. Hardware Implementation

As of now, the algorithm runs on a portable system with the configuration mentioned in Table I. With a security system that

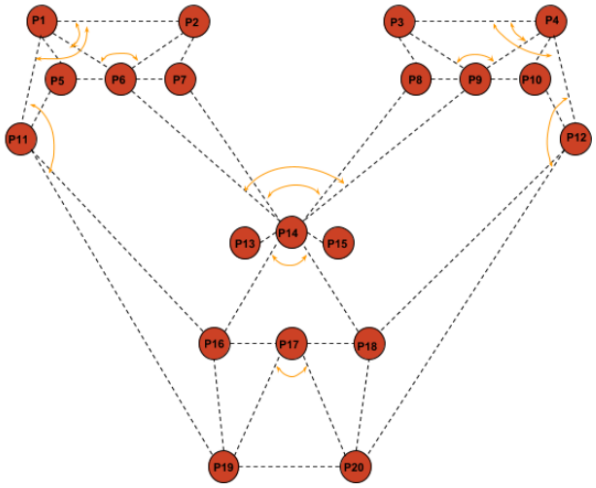

Fig. 7: The relative angles and ratios calculated as shown in the figure. These specific angles are sufficient to classify disguised faces with decent accuracies and maintaining real-time performance.

has a system with similar specifications the architecture would run on it too, detecting key-points and identifying disguised faces in real-time using tensorflow (Abadi *et. al.* [14]).

TABLE I: Training System Specifications

Hardware	Specification
Memory	32 GB
Processor	Intel Core i7-4770 CPU @ 3.4 GHz x 8
GPU	GeForce GTX 1050 Ti/PCIe/SSE2
OS Type	64-bit Ubuntu 16.04 LTS

TABLE II: Real-Time Test Bench Specifications

Hardware	Specification
Memory	8 GB
Processor	Intel Core i5-4770 CPU @ 2.3 GHz x 2
GPU	N/A
OS Type	64-bit Windows 7 Pro

We are looking at development of a robotic platform for the implementation of our architecture. The training of the model was done on a system with specifications mentioned in Table II.

A. Time Complexity

We have observed that our architecture takes 2.598 seconds to process a batch of 50 frames completely (from prediction to classification), which means that it takes around 0.0518 seconds/frame, which eventually results in a decent FPS rate of 19.3 frames/second. This performance is almost real-time for real-life security solutions deployed around the world.

IV. Evaluation and Results

During our research we observed that not many metrics are relevant to the concept of disguised facial identification and Singh *et. al.* [9] and Dhamecha *et. al.* [8] also reported their classification rates on custom datasets. In light of this matter, we have used very basic metrics to evaluate the performance of

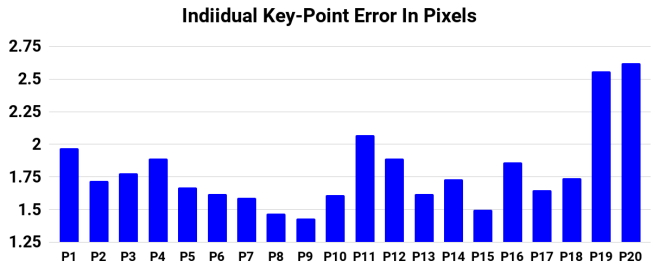

Fig. 8: Depiction of the Average Error of Prediction of every individual key-point (Y-Axis). As it can observed, the maximum variation of the key-point prediction is by 2-3 spatial units (X-Axis) only. The difference (or the spatial distance) between the prediction and the ground truth is calculated using the standard Euclidean Distance Formula. In order to calculate this evaluation metric, the testing (unseen) images were first hand annotated by experts. These images were later fed to model for prediction. The metric suggests how much was a particular predicted key-point is offset from the actual.

our algorithm. We have use the *Mean Absolute Error* (MAE) which is defined as the following :

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\phi_i - \hat{\phi}| \tag{1}$$

ϕ_i shows the position of i^{th} facial key-point in the annotated image and $\hat{\phi}$ represents the predicted position of the same i^{th} facial key-point in the corresponding image. Fig. 8 graphically shows the average error in the detection of key-points as compared against the ground truth. The error metric chosen to calculate the MAE was the standard euclidean distance defined as the error between the predicted key-points x_1, y_1 and annotated key-points x_2, y_2 :

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{2}$$

Furthermore, the prediction accuracy of the classifier has been represented graphically in Fig. 9. The angles marked in the color orange are taken into consideration as during the research it was observed that only certain angles and ratios affect the final classification outputs. So firstly, the slope of the lines is calculated using the formula :

$$m = \tan\theta = \frac{y_2 - y_1}{x_2 - x_1} \tag{3}$$

Then the angles between the lines are calculated using :

$$\phi = \tan^{-1}\frac{m_1 - m_2}{1 + m_1 m_2} \tag{4}$$

As compared with the other state-of-the-art techniques in the field of disguised facial identification, (Singh *et. al.* [9]), we report an improvement in the ideology by a margin of 9.8%. Classification rates have been reported on both simple and complex datasets (Table III), with however minimal attention to the simple dataset, as that directly comes under the scope of regular facial recognition.

TABLE III: Performance Results

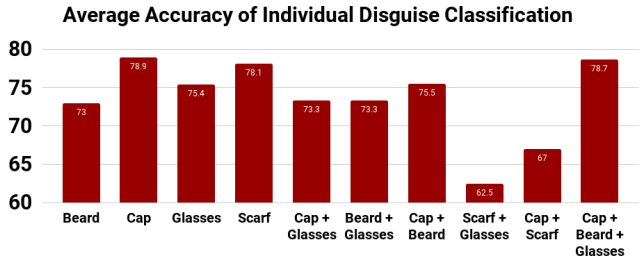

Fig. 9: Figure shows the classification results of our Neural Network (Fig. 5) classifier. The performance of the algorithm is decent as observed. It is however, interesting to note that the accuracy of facial disambiguation drops to 60-68% when the disguise contains a scarf. The occurrence of this could be attributed to the fact that the lips key-points are hidden in these disguises which play a crucial role in disguised face recognition. Rest all disguises where the point is visible the classification rates are higher. The results have been evaluated on 500 unseen images and graphical results are presented as the average of the classification rates.

Algorithms	Simple Dataset	Complex Dataset
Dhamecha *et. al.* [8]	65.2%	53.4%
Singh *et. al.* [9]	78.4%	62.6%
DiFRuNNT (*Ours*)	**86.6%**	**72.4%**

V. Supplementary Material

All the codes written during the conducted research (training, testing and data annotation) along with the trained models have been made available at the following GitHub repository : https://github.com/Abhinandan11/Disguised-Facial-Recognition-DiFRuNNT

VI. Discussions and Future Work

The task of identifying disguised human faces could be very challenging for humans themselves and imparting such a intelligence to machines would take a lot of time and training. Although the research presented in this paper is an attempt to classify disguised faces in real-time, there is still a great scope of improvement in domains like dataset enlargement and camera orientation. If the system is aware of larger number of disguises possible and it becomes independent of the orientation of the camera of the subject's face, the robustness would increase manifold.

Moreover, the problem of disguised facial identification has a slightly different approach to than regular face recognition techniques. In face recognition, normally, the faces are detected first using some algorithms and the classification techniques are applied to the detected faces only and not on the whole image. This methodology is time efficient as it avoids the redundant parts (non-face) of the image. However, there is no existing algorithm for detecting disguised faces, as there could be multiple types of disguises possible. Therefore, in order to speed up the architecture, we are researching upon how to design a disguised face detector logic that would isolate the disguised faces from the environment.

Fig. 10: The above images show the prediction accuracy of the CNN. Above are the original images and below them are the same images with the prediction points overlaid. As claimed earlier, the errors in the facial key-points are very minimal (around 2 spatial units only). Moreover the facial orientations are more than what the model was trained for, even then the performance is desirable and does not degrade.

References

[1] Berretti, Stefano, Boulbaba Ben Amor, Mohamed Daoudi, and Alberto Del Bimbo. "3D facial expression recognition using SIFT descriptors of automatically detected keypoints." The Visual Computer 27, no. 11 (2011): 1021.

[2] Sun, Yi, Xiaogang Wang, and Xiaoou Tang. "Deep convolutional network cascade for facial point detection." In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 3476-3483. IEEE, 2013.

[3] Wang, Yue, and Yang Song. "Facial Keypoints Detection." (2014).

[4] Shi, Shenghao. "Facial Keypoints Detection." arXiv preprint arXiv:1710.05279 (2017).

[5] Yoshino, Mineo, Kasumi Noguchi, Masaru Atsuchi, Satoshi Kubota, Kazuhiko Imaizumi, C. David L. Thomas, and John G. Clement. "Individual identification of disguised faces by morphometrical matching." Forensic science international 127, no. 1-2 (2002): 97-103.

[6] Singh, Richa, Mayank Vatsa, and Afzel Noore. "Recognizing face images with disguise variations." In Recent Advances in Face Recognition. InTech, 2008.

[7] Righi, Giulia, Jessie J. Peissig, and Michael J. Tarr. "Recognizing disguised faces." Visual Cognition 20, no. 2 (2012): 143-169.

[8] Dhamecha, Tejas Indulal, Richa Singh, Mayank Vatsa, and Ajay Kumar. "Recognizing disguised faces: Human and machine evaluation." PloS one 9, no. 7 (2014): e99212. 45

[9] Singh, Amarjot, Devendra Patii, G. Meghana Reddy, and S. N. Omkar. "Disguised face identification (DFI) with facial keypoints using spatial fusion convolutional network." In 2017 IEEE International Conference on Computer Vision Workshop (ICCVW), pp. 1648-1655. IEEE, 2017.

[10] Pfister, Tomas, James Charles, and Andrew Zisserman. "Flowing convnets for human pose estimation in videos." In Proceedings of the IEEE International Conference on Computer Vision, pp. 1913-1921. 2015.

[11] Phillips, P. Jonathon. "Support vector machines applied to face recognition." In Advances in Neural Information Processing Systems, pp. 803-809. 1999.

[12] Guo, Guodong, Stan Z. Li, and Kapluk Chan. "Face recognition by support vector machines." In Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, pp. 196-201. IEEE, 2000.

[13] Dniz, Oscar, M. Castrillon, and Mario Hernndez. "Face recognition using independent component analysis and support vector machines." Pattern recognition letters 24, no. 13 (2003): 2153-2157.

[14] Abadi, Martn, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin et al. "TensorFlow: A System for Large-Scale Machine Learning." In OSDI, vol. 16, pp. 265-283. 2016.