A Review and Construction of a Real-time Facial Recognition System

Eduardo H. Teixeira[1], Samuel B. Mafra[1], Joel J. P. C. Rodrigues[2,3], Werner A. A. N. da Silveira[1], Ousmane Diallo[4]

[1]Instituto Nacional de Telecomunicações (Inatel)
510, João de Camargo Avenue – 37540-000 – Santa Rita do Sapucaí – MG – Brazil

[2]Federal University of Piauí (UFPI), Teresina – PI, Brazil

[3]Instituto de Telecomunicações, Portugal

[4]Department of Informatics, University of Assane Seck, B. P. 523 Ziguinchor, Senegal

eduardot@gea.inatel.br, samuelbmafra@inatel.br, joeljr@ieee.org,
waugustoan@gee.inatel.br, odiallo@univ-zig.sn

Abstract. *The evolution of surveillance technologies allows a reduction in human interaction with the process, since most of the monitoring functions performed by an individual can be replaced by detection and recognition techniques in real-time. This paper proposes the development of a surveillance system, which uses these techniques to identify individuals present within the field of view of camera. A combination of the Histogram of Oriented Gradient and Support Vector Machine techniques is applied for face detection, while a Residual Network is used during the stage of recognizing individuals. This shows the possibility of implementing this set of techniques, even in hardware with processing limitations.*

1. Introduction

A facial recognition system allows to detect and recognize in real-time a person by their facial characteristics. This technology is very important and is used in many applications for various purposes. Thus, a facial recognition system can be installed to monitor and identify people in public or restricted areas for providing security control, photos matching, user verification, user access control, etc. For example, facial recognition can be used to validate attendance during classes in an university [Lin and Li 2019]. Another applicability of this system is to locate suspects, in real-time, using tracking and identification techniques [Chuo, Sheu and Chen 2019].

All the facial recognition technologies can be included in the field of computer vision. A science, with a set of tools, which allows a device to process and analyze real-world images. A process similar to what is done by the human eye in conjunction with the brain. Currently, several computer vision techniques are incorporated into surveillance systems. One of its advantages is to avoid continuous monitoring of images from being performed by a human. Since it is possible to program the algorithm to display only the relevant information.

A surveillance structure is usually composed of cameras with good resolution, connected to processing and storage equipment. The proposal to be presented in this paper is to use this structure to provide an automatic analysis of the images coming from the camera with the application of detection techniques and facial recognition in real-time. In addition, the proposed system includes a comparative analysis of processing time in some operations.

The remainder of this paper is organized as follows: Section 2 discusses the review of the literature of the techniques used for facial detection and recognition as well as their utilization in some real-time applications. Section 3 describes the architecture of the system proposed and provides a greater detail of the model used in that work. Section 4 describes the software flow and the steps required to build the system's operating logic, while in section 5 the experimental results are presented for validation. And finally, Section 6 concludes the paper and pinpoints further research works.

2. Related Work

Several approaches can be used for the detection and recognition of human faces. Among the most well-known techniques, it can be mentioned the Viola-Jones algorithm, widely used for its efficiency in computing time, which allowed its application in real-time systems [Viola and Jones 2004].

There are also appearance-based techniques, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA), and Support Vector Machine (SVM), which use machine learning, to learn the characteristics of the model [Brahmbhatt, Prajapati and Dabhi 2017]. Although the appearance-based approach has good results, it is relatively complex to implement, due to the need of many samples in the training stage.

Feature descriptors are also applied as methods for face recognition. For instance, [Xiang, Tan and Ye 2018] makes a comparison between three famous facial feature descriptors, Histograms of Oriented Gradients (HOG), Gabor and Local Binary Pattern (LBP), where it presents the advantages and disadvantages of using each method.

Considering embedded implementations, it can be also noted different techniques applied during the recognition phase. As the systems proposed in [Patil and Shukla 2014], [Gupta 2016], [Sajjad *et al.* 2017], [Wazwaz *et al.* 2018] implemented using a Raspberry Pi, where the authors use a hybrid of PCA and LDA, only PCA, SVM and LBP, respectively.

In the same way as the work [Patil and Shukla 2014], some authors also propose the use of hybrid techniques to increase the accuracy of facial recognition. The authors in [Jain *et al.* 2018], uses the SVM technique combined with the HOG and Gabor filters to detect emotions in the human face, this demonstrates an improvement in the classification accuracy. The conclusion is that the solution of the combination of HOG and SVM presents the best average of the results. In [Dadi and M Pillutla 2016], also tests the combination of HOG and SVM compared to the PCA for face recognition. This work shows that there is a higher precision of the first method in relation to the second.

Other authors, such as [Li *et al.* 2015], suggest the use of more advanced features, such as the use of Convolutional Neural Network (CNN) for facial detection. The advantage of applying this type of net is the ease of adaptation when the face is in position or angle variations. However, the authors claim that CNN has a high computational cost and presents better performance when running on Graphics Processing Units (GPU's). In section 5, a comparative analysis of the processing time for face detection is presented, between the hybrid method HOG + SVM, when compared to the use of CNN from Dlib [King 2009].

3. System Architecture

In this paper, a real-time facial recognition system was developed, which runs on the hardware of the Raspberry Pi 3 Model B and uses Raspbian as the operating system. Basically, images from the surveillance camera are sent to the Raspberry via the 802.11n wireless standard. During transmission, these images are received and processed, creating a record with information about the detection and identification of all individuals who were within the range of the camera.

Due to the ease of connection to the hardware, many works in the literature use the standard Raspberry camera. In this paper, a standard surveillance camera model is used, which guarantees the prototype shown in Figure 1, greater fidelity in relation to real surveillance systems. Moreover, the use of this wireless standard for data exchange, allows the hardware and the camera to be installed in different locations.

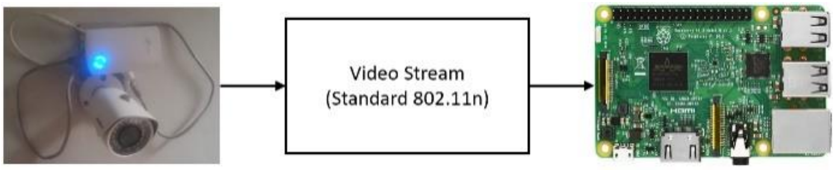

Figure 1. System prototype.

The designed system has two relevant characteristics: The first is the operation in real-time and the second is the use of a model that has a high precision of facial recognition. According to [Valeriani and Poli 2019], the recognition technique used has an accuracy of 99.38% proven in Labeled Faces in the Wild (LFW), one of the most widely used databases for reference in face recognition systems, as mentioned in [Liao *et al.* 2014].

To obtain this accuracy in the recognition process, a Residual Network (ResNet) is used. The construction process of this network is similar to that of a conventional CNN, but it incorporates residual layers together with convolutions, which act as non-linear functions of convolutive layers. The intention of adding this new layer is to prevent the accuracy from degrading in cases where there is the adoption of deeper networks. A residual block of ResNet has two convolutive layers in their infrastructure and two Rectified Linear Unit (ReLU) activation functions, as shown in Figure 2.

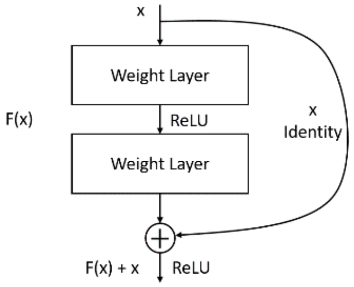

Figure 2. Residual learning block.

The residual layer differs from the other convolutive layers, in the sense that its output is the sum of the output of the second convolutive layer with the input of the residual layer. This structure allows deeper layers to directly receive data from the more superficial layers. The shortcut identity connections in this block do not add extra parameter or computational complexity.

In a non-residual CNN, the network is trained to adapt its parameters to all the content *F(x) + x*. In the residual architecture the value of x is added directly to the output by means of identity operations, so the network has to adjust only to the *F(x)* content. As a result, the residual network becomes simpler to optimize and achieves high accuracy due to its depth [He *et al.* 2016].

For the faces to be introduced in ResNet, they must first be detected, in this step the HOG method is used, learned via Max-Margin Object Detection (MMOD) [King 2015]. The HOG descriptor helps to manipulate the image in order to generate intensity gradient vectors, with directions that are dependent on the variation between pixels. This process is done in order to detect the edges of the image. This allows to identify faces within the frames in which this technique is applied.

The optimization by MMOD is used to improve the classification of the algorithm, its objective function is presented in equation (1), whose intention is to minimize the detection error.

$$\min_{w,\xi} \frac{1}{2}\left\|w\right\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi i \qquad (1)$$

In this equation, *C* is the variable that allows defining the compromise relationship between minimizing errors in the training set in relation to the complexity of the system. This variable is usually configured in the application of the SVM technique, since MMOD and SVM operate in very similar ways. *W* is defined as the parameter of a vector that leads to the least number of detection errors and ξi is an upper limit of the loss incurred by the training example.

The facial model used was set with parameters $C = 50$ and $\xi = 0.01C$. In practical terms, all images received by the Raspberry go through the application of the HOG and, whenever the gradient vectors in some part of the image are similar to the face model, a new face is detected and sent to ResNet for recognition.

4. Used Technologies

Face recognition can be used in two ways, the first is for verification, where the current image is compared to a specific face previously requested. The second is identification, in which the current image is compared with several images within a database, to determine some level of similarity. The application developed in this work uses the second case, in which the image is tested alongside all the users registered in the system.

4.1. Network Training Stage

The proposed facial recognition is based on a ResNet architecture, described in [Li *et al.* 2018]. ResNet has the ability to perform face recognition through deep metric learning. This network architecture was trained with the machine learning toolkit, Dlib [King 2009]. The network used is based on ResNet-34 by [He *et al.* 2016], but with a reduced number of filters and layers.

The ResNet used was trained with an input set of about 3 million faces. The training process works with the analysis of three face images at a time, two from the same person and one from a different person. Then, the algorithm analyzes the measurements generated for each of these three images. Weights are adjusted so that the neural network always generates vectors of 128 similar dimensions for images of the same person and distant values when working with the image of a different person.

After repeating this step several times, for a large amount of images from different individuals, the neural network learns to generate 128 measurements reliably, for each person. As such, images of the same person should always provide, approximately, the same measurements. The advantage is that the training is done prior to its application in a real system. Therefore, even if the training takes a long time, the network can still deliver quick results when it operates in the stage of executing the recognition algorithm.

4.2. Database Registration Stage

In this step, a folder containing images of people with their names is indicated. With that, the algorithm goes through the folder and analyzes the faces present inside each one. Figure 3 shows the complete flow, which is repeated until all users are registered.

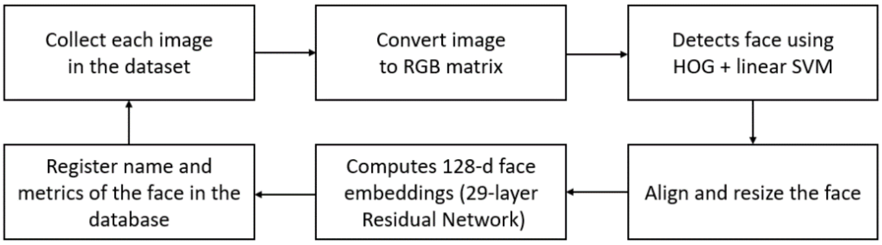
Collect each image in the dataset
Convert image to RGB matrix
Detects face using HOG + linear SVM
Align and resize the face
Computes 128-d face embeddings (29-layer Residual Network)
Register name and metrics of the face in the database

Figure 3. New images registration sequence.

This analysis is performed when converting an image into a matrix and applying a classifier to find the face within the image. The classifier used is a combination of HOG and SVM, due to the good results obtained in previous works already mentioned. The detection performed by this technique can be seen in Figure 4.

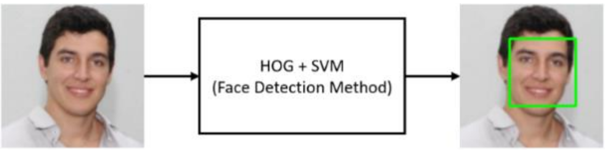

Figure 4. Results of the face detection method.

The next step is to ensure the alignment and resizing of the image, as shown in Figure 5. The collected images must be normalized before going to the metric network, because only in this way the ResNet can generate the metrics reliably.

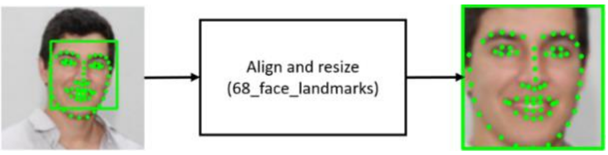

Figure 5. Normalization results of landmarks.

When the facial images are sent to the network, it generates a metric vector of 128 dimensions of the faces of each person. This architecture is a simplified version of ResNet-34, however it has 29 convolution layers and a smaller number of filters per layer.

4.3. Detection and Recognition Stage

While the camera captures the video, these images are accessed by Raspberry via the camera's IP address. This makes it possible to switch between more than one data source, with access to the video stream from different cameras. To make this possible, a standard 802.11n communication module was connected to the camera to send the video stream wirelessly. With that, the video is processed in frames that are loaded and converted into an RGB matrix. The HOG algorithm transforms this matrix into a vector-oriented image, where it is possible to establish the region around all the faces present in the image. This process of demarcating faces could also be done through a face detector based on CNN.

To perform the alignment, a 68 point landmarks pose predictor is used, based on regression trees [Kazemi and Sullivan 2014]. The predictor is used in the image to estimate the positions of the main landmarks on the face, such as eyes, nose, mouth and face contour, as shown in Figure 5. These geometric representations provide metrics and positions that can be used to aid in detection techniques and facial recognition [Wu and Ji 2018]. After identifying where the face is and what is its position, the face is normalized and aligned with a simple 2D transformation. As a result, the images sent to the network have eyes and nose in similar places in all the photos. Projects like OpenFace [Amos, Ludwiczuk and Satyanarayanan 2016], also work with this type of normalization.

The next step is to compare the faces found with the list of people registered in the system. This search is made through correspondences between the vectors of 128 dimensions that have the shortest Euclidean distance, that is, the person in the database who presents the measurements closest to each of found faces. If no image in the database has measurements close, the result returned will be “unknown”. A k-Nearest Neighbors (k-NN) algorithm was used, which implements the Euclidean distance for decision. The expression that describes the implemented rule is given by equation (2), where the closest vectors are recognized via distance metric.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \qquad (2)$$

The complete and detailed flow of the entire system proposed during the operation stage is shown in Figure 6. This flow covers from the moment of capturing the images and applying the sequence of techniques, until the display of the system's output results.

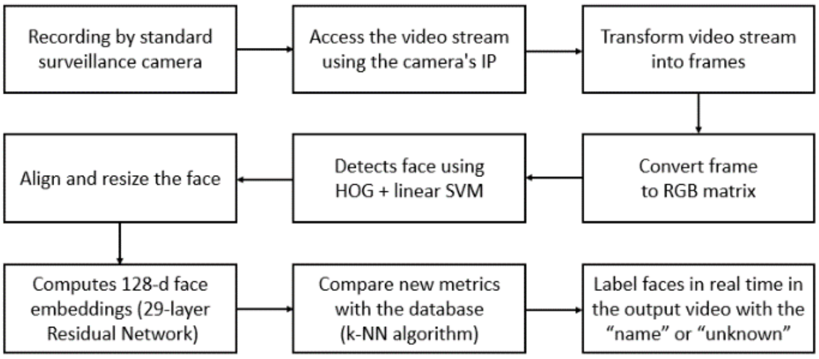
Recording by standard surveillance camera
Access the video stream using the camera's IP
Transform video stream into frames
Convert frame to RGB matrix
Detects face using HOG + linear SVM
Align and resize the face
Computes 128-d face embeddings (29-layer Residual Network)
Compare new metrics with the database (k-NN algorithm)
Label faces in real time in the output video with the “name” or “unknown”

Figure 6. Full flow of system operating mode.

5. System Evaluation and Validation

In this section, experimental results are presented in order to evaluate and validate the proposed technique. The choice of the HOG + SVM detection technique over CNN, can be justified with the analysis of Figure 7, which demonstrates a notable difference in processing time between these two approaches. This time was measured during the face detection stage, using the same sample for both techniques. The measurements were performed seven times, with the collection of data from the execution time. This Proves what was commented by the authors in [Li *et al*. 2015], about the suggestion that CNN method needs greater computational power to be used in real-time systems. Another characteristic that can be analyzed is the smaller variation in the processing time of the HOG + SVM technique compared to CNN, when applied several times on the same image.

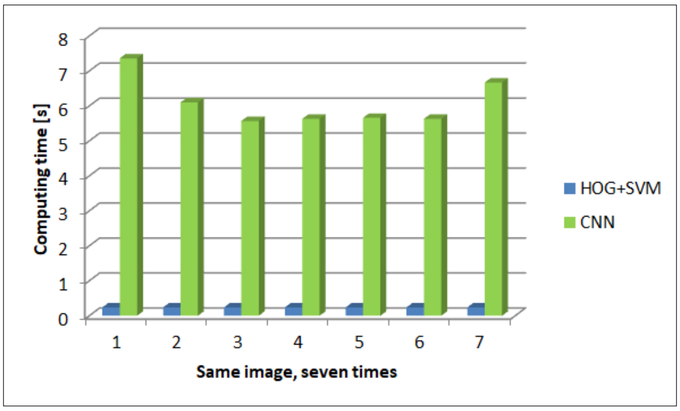

Figure 7. Comparison between the processing time of the HOG + SVM and CNN techniques, for facial detection.

Still during the detection phase, other tests were carried out with the CNN. The objective is to prove the performance of the runtime when this same technique is applied to devices with different computational power. Such tests are done from two devices, the first is the Raspberry, used in all implementations throughout this research work. The second hardware is a notebook with greater computing power. Table 1 presents the comparison between the characteristics of the two machines.

Table 1. Comparison of the processing variables of the two devices.

Model	Raspberry Pi 3 Model B	Notebook Dell Vostro 3560
Processor	Broadcom BCM2837 SoC @ 1.2GHz	Intel Core i7-3632QM CPU @ 2.2GHz
RAM	1GB LPDDR2	8GB DDR3
GPU	VideoCore IV 400 MHz	Radeon HD 7670M 900 MHz

Figure 8 shows that when applying the CNN technique to a second device with greater processing power, there is a significant decrease in the detection time. Thereby, it is possible to admit the possibility of an implementation of face detection in real-time, with the use of CNN, which can be made possible by increasing the computing capacity of the devices or by the evolution of the technique itself. In addition, the data collected reinforce the idea that this method has a greater variation in processing time, even running on devices with different characteristics.

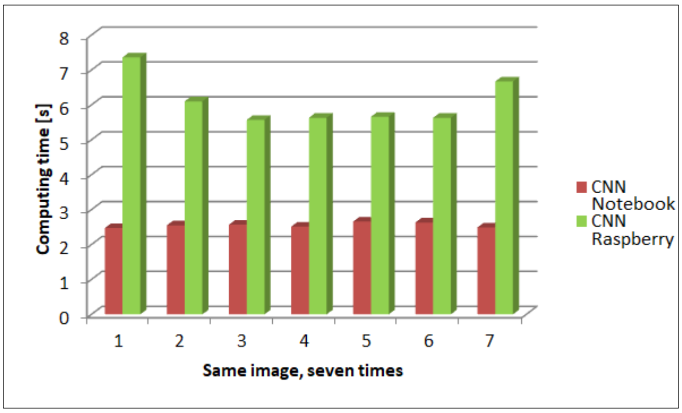

Figure 8. Comparison of the processing time of CNN technique on two hardware (Raspberry Pi 3 Model B and Notebook Dell Vostro 3560).

When passing through the recognition step, all detected faces receive a classification. This process returns the results of the match, with the name of the database user or the code "unknown". Figure 9 shows two of these cases. The first, in which the detected face is registered in the system and the second, in which the user is not yet registered. If a user's face is detected as unknown in the video stream, there is a command capable to add that face to the existing data set. This allows registration of any user even with the system running.

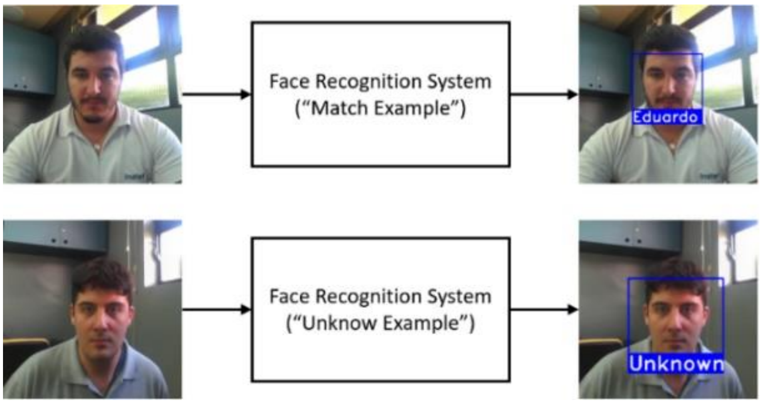

Figure 9. System output examples.

6. Conclusion

This paper proposes a real-time surveillance system that provides an automatic analysis of video that includes detection techniques and facial recognition. The system is highly accurate and is able to recognize any individual registered in the database. In addition, the recognition in the video stream is done in real-time, something really crucial for some applications.

The use of detection through HOG + SVM allows the system to respond quickly and accurately. Since at this stage CNN under current conditions, can cause the real-time feature of the system to be compromised. The contribution of the comparison between the two methods in an empirical way, allows to obtain numerical values of processing time, which can serve as a basis for other works.

Some factors such as lighting, distance between the individual and the camera, resolution of the camera itself, angulation and position of the face can affect the detection and recognition process. Since there is an entire process of standardization of the face before it passes through the recognition network, these problems tend to be minimized by the set of techniques chosen. Even with the processing limitations of the Raspberry, the system is capable of performing all detection and recognition processing in real-time, fulfilling the purpose of this work.

As future work, this system can be integrated with an IoT Middleware, so that the data collected from the recognition system can have new applications, as a surveillance system for smart cities. In addition to analyzing the performance of the solution with variations in brightness and distortion in the images. It is also intended to evaluate the use of CNNs that have a faster response, so they can be used in the detection stage.

7. Acknowledgements

This work was partially supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) and by RNP, with resources from MCTIC, Grant No. 01250.075413/2018-04, under the Radiocommunication Reference Center (Centro de Referência em Radiocomunicações - CRR) project of the National Institute of Telecommunications, Brazil; by FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/EEA/50008/2020; and by the Brazilian National Council for Research and Development (CNPq) via Grants No. 431726/2018-3 and 309335/2017-5. The authors thanks Adam Geitgey and Davis King, for the libraries that were used as a starting point in this study.

References

Amos, B., Ludwiczuk, B. and Satyanarayanan, M. (2016) ‘OpenFace: A general-purpose face recognition library with mobile applications’, *CMU School of Computer Science, Tech. Rep.*, pp. 1–18.

Brahmbhatt, N. R., Prajapati, H. B. and Dabhi, V. K. (2017) ‘Survey and analysis of extraction of human face features’, *Innovations in Power and Advanced Computing Technologies, i-PACT*, pp. 1–8.

Chuo, Y. H., Sheu, R. K. and Chen, L. C. (2019) ‘Design and Implementation of a Cross-Camera Suspect Tracking System’, *International Automatic Control Conference (CACS)*, pp. 1–6.

Dadi, H. S. and M Pillutla, G. K. (2016) ‘Improved Face Recognition Rate Using HOG Features and SVM Classifier’, *IOSR Journal of Electronics and Communication Engineering*, Vol. 11(04), pp. 34–44.

Gupta (2016) ‘Face Detection and Recognition using Local’, *IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, pp. 7923–7929.

He, K. *et al.* (2016) ‘Deep residual learning for image recognition’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

Jain, C. *et al.* (2018) ‘Emotion Detection and Characterization using Facial Features’, *3rd International Conference and Workshops on Recent Advances and Innovations in Engineering, ICRAIE*, pp. 1–6.

Kazemi, V. and Sullivan, J. (2014) ‘One millisecond face alignment with an ensemble of regression trees’, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1867–1874.

King, D. E. (2009) ‘Dlib-ml: A machine learning toolkit’, *Journal of Machine Learning Research*, Vol. 10, pp. 1755–1758.

King, D. E. (2015) ‘Max-Margin Object Detection (MMOD). Available: http://arxiv.org/abs/1502.00046. Accessed on: Mar., 25, 2020.’, *arXiv*, pp. 1–8.

Li, H. *et al.* (2015) ‘A convolutional neural network cascade for face detection’, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 5325–5334.

Li, T. *et al.* (2018) ‘A Research of Character Recognition Based on Residual Neural Network’, *IEEE International Conference of Safety Produce Informatization*, pp. 804–807.

Liao, S. *et al.* (2014) ‘A benchmark study of large-scale unconstrained face recognition’, *IJCB International Joint Conference on Biometrics*, pp. 1–8.

Lin, Z. H. and Li, Y. Z. (2019) ‘Design and Implementation of Classroom Attendance System Based on Video Face Recognition’, *IEEE International Conference on Intelligent Transportation, Big Data and Smart City, ICITBS*, pp. 385–388.

Patil, A. and Shukla, M. (2014) ‘Implementation of Classroom Attendance System Based on Face Recognition in Class’, *International Journal of Advances in Engineering & Technology*, Vol. 7(3), pp. 974–979.

Sajjad, M. *et al.* (2017) ‘Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities’, *Future Generation Computer Systems*, pp. 1–32.

Valeriani, D. and Poli, R. (2019) ‘Cyborg groups enhance face recognition in crowded environments’, *PLOS ONE*, Vol. 14(3), pp. 1–17.

Viola, P. and Jones, M. J. (2004) ‘Robust Real-Time Face Detection’, *International Journal of Computer Vision*, Vol. 57(2), pp. 137–154.

Wazwaz, A. A. *et al.* (2018) ‘Raspberry Pi and computers-based face detection and recognition system’, *IEEE 4th International Conference on Computer and Technology Applications, ICCTA*, pp. 171–174.

Wu, Y. and Ji, Q. (2018) ‘Facial Landmark Detection: A Literature Survey’, *International Journal of Computer Vision*, Vol. 127(2), pp. 115–142.

Xiang, Z., Tan, H. and Ye, W. (2018) ‘The Excellent Properties of a Dense Grid-Based HOG Feature on Face Recognition Compared to Gabor and LBP’, *IEEE Access*, Vol. 6, pp. 29306–29318.