AI-enabled Face Live Tracking Attendance System

DADI RAVI VARMA, Dr. SALINA ADINARAYANA

1. **Student, Dept of CSE, Raghu Institute of Technology, Visakhapatnam, A.P, India.**
2. **Head of the Department, Dept of CSE, Raghu Institute of Technology, Visakhapatnam, A.P, India.**

Email: dadiravivarma@gmail.com , adinarayana_cse@raghuinstech.com

ABSTRACT

Face recognition is among the most productive image-processing applications and has a pivotal role in the technical field. We are living in a world where everything is automated and linked online. The internet of things, image processing, and machine learning are evolving day by day. Many systems have been completely changed due to this evolution to achieve more accurate results. The development of this system is aimed to accomplish digitization of the traditional system of taking attendance by calling names and maintaining pen-paper records. This paper proposes a method of developing a comprehensive embedded attendance system using facial recognition. The system is based on Raspberry Pi that runs Raspbian (Linux) Operating System installed on a micro-SD card. The Raspberry Pi Camera, as well as a 4-inch screen, are connected to the Raspberry Pi. By facing the camera, the camera will capture the image and then pass it to the Raspberry Pi which is programmed to handle face recognition by implementing the Local Binary Patterns (LBP) algorithm. If the input image matches with the trained dataset image, then the attendance results will be stored in a CSV sheet. The database is connected to a simple mail transfer protocol (SMTP, which makes attendance reachable to any mail. The system has 98% accuracy with the dataset of 1000-person images.

KEYWORDS: *Deep Learning in medicine, Feature extraction, and classification, TensorFlow, Keras, Data pre-processing, confusion matrix, OpenCV*

INTRODUCTION

Face Recognition system is a common Biometric tool that is leaving digital signatures globally while reducing national threats by exceptionally enhancing the soundness of security applications. Many optimization techniques have interceded over a few decades as the performance has been spread from normal face recognition systems to intensified security applications in many scenarios such as in ATM centers, Airports, shopping malls, and public attendance systems, etc., Several case studies have been proposed and verified in real time scenarios yet failing in some aspects depending upon the environmental conditions and camera accountabilities and so on. Hence, we further gone with many types of research in order to deal with the real-time problems and designed a Custom Universal face detector and implemented our Hybrid Feature Extraction Technique before facial embedding extraction, and compared the faces in real-time with the help of machine/Deep learning classifier in order to perform face recognition in real-time.

A general Biometric platform for common attendance systems to the advanced securities lies in the recognition of digital signatures through a normal platform called the Face Recognition system. However, face recognition system has a variety of applications in real-time fields such as in ATM centers, Airports, shopping malls, Schools and Colleges, etc., yet sometimes fail in real-time scenarios due to environmental conditions, camera accountability, height, and angle considerations. With the advancement of knowledge in the research fields, many researchers have proposed various pre-processing and feature extraction techniques (with the help of mathematical expressions) after face detection and alignment resulted in promising results in real-time

conditions yet there are some conditions where the chance of false acceptance rate has a likely tendency to occur.

Feature Extraction is the fundamental step for image preprocessing technique; where the characteristics of an image are described in terms of descriptors or feature vectors. It is a qualitative approach to preserving the principal characteristics such as color, texture, shape, spectrum, threshold, and wavelet transformations of an image or a sub-image.

Key face detectors such as from basic HAAR cascades to DLIB face detection networks have been implemented in many case studies in order to detect faces in real-time scenarios. But due to their limitations in detection because of varying focal lengths of the cameras, illumination, and angle, architecture speed, and wrong detection problems; diverse face detection frameworks have been proposed with the help of deep learning frameworks such as SSD, Faster-RCNN and Yolo which will be later explained in Section II of our paper (Case Studies).

Various Face encoding/embedding methods have also been done earlier after detecting and aligning the face. One such procedure is the dlib (5,64,128) shape predictors/landmarks. But due to some advancement in deep learning frameworks, several pre-trained models such as VGG_Face, VGG16, VGG19, MT-CNN, Face-net, etc., is used to extract the face embeddings, which can be later trained with different machine learning and deep learning networks.

Hence in this paper, we have undergone different case studies and perceived diverse real-time problems such as false detection, motion and Gaussian blurriness in faces, illumination variation, etc., and designed a real-time framework in which we are extracting the face with our custom universal face detector and solved most of the above problems using a hybrid preprocessing network and extracted the face embeddings using our own Deep learning frameworks and classified those embeddings using different machine learning and deep learning frameworks in which KNN and SVM give best solution in classification for real-time problems. In this paper, the below section represents previous case studies of different researchers which are followed by our proposed architecture with mathematical equations and results in the following Sections which are followed by references.

Hardware Platform

The hardware part mainly consists of a digital computer, a Raspberry Pi Kit, 4-inch LCD displays, a Raspberry camera, and voice recognition which is being discussed along with their specific functions.

Raspberry Pi Kit: The Raspberry Pi is a series of small single-board computers developed by the United Kingdom by the Raspberry Foundation to promote the teaching of computer science in schools and developing countries. Peripherals (including keyboard, mice, and cases) are not included with the Raspberry Pi. It is bundled with onboard Wi-Fi, Bluetooth, and USB boot capabilities. It has a Broadcom System-On-Chip, which includes an ARM-compatible central processing unit (CPU) and an on-chip graphics processing unit (GPU, a video core). CPU speed ranges from 700 to 1.2GHz for Pi 3 and onboard memory ranges from 256MB to 1GB. Secure Digital (SD) cards are used to store the operating system and program memory either in SDHC or microSD sizes. Lower-level output is produced by a number of GPIO pins that supports common protocols. It supports Raspbian, a Debian-based Linux distribution for download as well as the third-party Ubuntu,Windows 10 IT Core, RISC OS, and centralized media center distributions. It promotes Python and Scratch as the main programming languages with support for many other languages. The Raspberry Pi 3 supports 1GB RAM.

Liquid Crystal Display (LCD)

A Liquid Crystal Display screen [8-9] is an electronic display module and find a wide range of applications. A 4-inch LCD display is a very basic module and is very commonly used in various devices and circuits. LCD stands for liquid crystal display. They come in many sizes 8x1, 8x2, 10x2, 16x1, 16x2, 16x4, 20x2 ,20x4 ,24x2, 30x2, 32x2, 40x2 etc. These modules are preferred over seven segments and other multi-segment LEDs. The reasons are: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations, and so on. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in a 5x7 pixel matrix. This

LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling the display, etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

Raspberry Pi camera

The Raspberry Pi camera module can be used to take high-definition video, as well as still photographs. It's easy to use for beginners but has plenty to offer advanced users if you're looking to expand your knowledge. There are lots of examples online of people using it for time-lapse, slow-motion, and other video cleverness. You can also use the libraries we bundle with the camera to create effects. The has a feature a 5MP sensor, Wider image, capable of 2592x1944 stills, 1080p30 video,1080p video supported, CSI, Size: 25 x 20 x 9 mm.

Voice Recognition

Recent models generally use Bluetooth 4.0 or even Bluetooth 5, and wireless speakers generally have a range of 10 meters. Bluetooth devices use a radio communication frequency such that the devices do not have to be in a visual line of sight witheach other.

Wireless speakers use rechargeable batteries to power them. The operating time of the speaker before it has to be recharged is usually 6 hours. Models with more powerful batteries can last up to 10 hours or more. Almost all wireless speakers operate on rechargeable batteries that are not replaceable so the lifespan of these speakers is that of their batteries.

Some speaker models with a large battery capacity can also act as a power bank to charge another device to full capacities, such as a mobile phone.

They are generally recharged with either a B-type plug or a more universal USB connector, mainly through either mini or micro-USB or USB connectors. The complete charging cycle of a speaker generally varies from 3 to 6 hours.

METHODOLOGY

In this model, our aim is to propose a secured model for providing high accuracy using face recognition for smart attendanceauthentication.

Our Face Recognition system comprises two types of Phases, i.e., the Training Phase and Testing Phase as shown in the block diagrams figures 1 & 2 Below. In the Training Phase, we take the person's face from the picture by using our custom detector and extract encodings from the images and store them in a NumPy file. Whereas in Testing Phase, we again calculate the encodings of the face after the face is detected and compare them with the trained encodings with the help of aggregation of distance metrics.

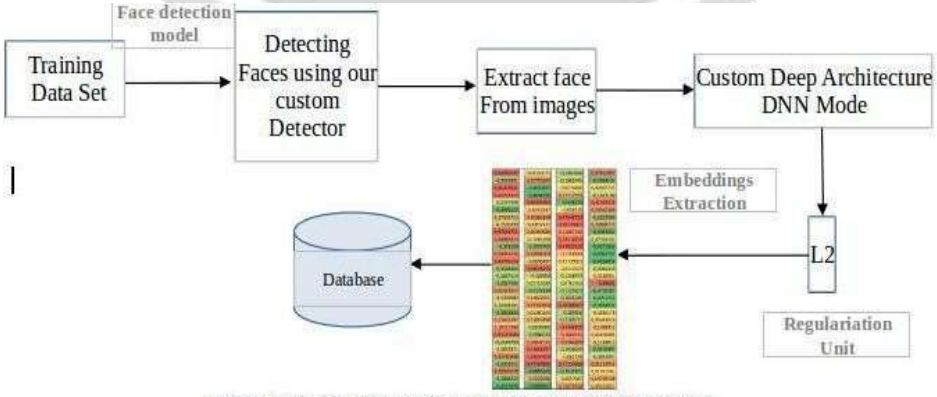

Fig 1. Training Phase of our Face Recognition system

fig.2 Testing phase of our Face

Block diagram of Face Recognition system:

Recognition system

Step1- Data collection and Pre-processing

Data collection can be stated as the process of *gathering* and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate the outcomes of the project. Hence in our project we have collected nearly 20,000 images from real-time IP cameras with different angles and labeled them with the ground truth boxes with (x,y,w,h) coordinates. In order to make a universal face detector, we collected our data from different IP cameras at different angles.

In our process, we have done pre-processing in two ways, one was to clean the data set before labeling in order to make an ideal data set and the other process will be held after face detection. Since our face detection model aligns the face itself after cropping makes a color conversion i.e., from BGR to RGB image since we get BGR images while loading in python and we need to apply filters that remove motion blur in images.
The filter we used here in order to detect and remove the motion blur in images is the local binary patterns (LBP) which is avisual descriptor in computer vision.

Step2 – Face Detection Algorithm:

Face Detection system plays a prominent role not only in identifying faces in an image but also helps in extracting features of a particular face after detection, cropping, and aligning which helps in identifying whether the person belongs to the office or he is just a visitor and so on from the generic attendance system to the security regions by spreading its wings in order to find the criminal in a crowded area.

The Block Diagram of our Face Detection architecture is shown below in figure 1.

Algorithm1: Face Detection in an image

Basic Steps in Identifying Faces in an Image

Input: x image with a sliding window approach for Feature Extraction
1: W:= In all rectangles if r ϵ R (Region of proposal) for f(x,r)
2: Sort W such that $W1 \geq W2 \geq W3 ..... \geq Wn$
3: Output Y* = {}
4: **for** i in range (1 to W):
5: **if** Wi doesn't overlap with any rectangles, then
6: Y = Y U Wi
7: **end** if
8: end
9: Detected Rectangles of the Faces in an image

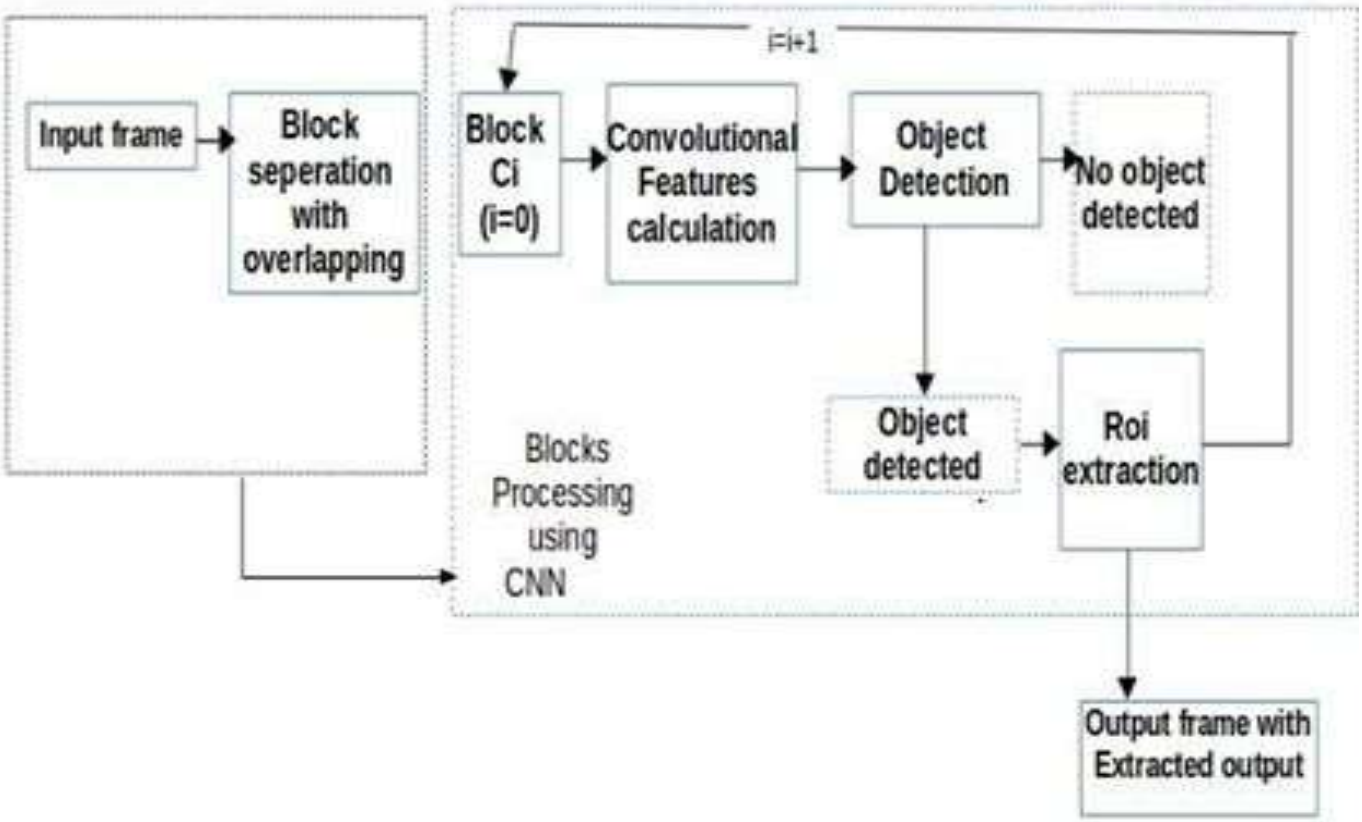

Fig3. Block Diagram of our Face Detection Architecture using Deep Learning Framework

In the basic face detection algorithm, a sliding window approach is taken on an image where a 3*3,6*6, and so on can be taken as a stride and move across the image in both horizontal and vertical directions and obtain the features of the region using either HAAR like features or HoG features in order to obtain the positive and negative windows. A binary classifier is used at the end in order to differentiate whether it is a positive window or a negative window i.e., the window contains the face-like features or not. And at the end, we use a Non-Maximum Suppression (NMS) technique in order to make two boxes not overlap with each order if the ratio of their intersection area to the total area covered is less than 0.5.

With the presence of standard deep learning classification frameworks which are used instead of binary classifiers increases the percentage of accuracy in real-time scenarios. The percentage of accuracy in detection and False positive rate of different classifiers can be explained with the help of the standard mathematical expressions as shown

$?_{i=1} M^k (fi)$ Equation 2

where F is the false positive rate where f_i is the maximum acceptance of the false positive rate and

$?_{i=1} M^k \ di$ Equation 3

where D is the Detection rate and d_i is the maximum detection rate of faces in an image.

In our custom face detector, we use a fully connected network (FCN) which is made up of only convolution layers and up-sampling. It uses three different strides in three different areas helping in securing the low-level features obtained after pooling.

In our algorithm, we split the image into some S*S grid cells and the object is predicted if the center of the object falls into that cell. Each grid cell predicts 5 components that are the coordinates of the bounding boxes (which have to be normalized to [0,1]) and the probability of the class object (here we have only one class that

is facing so there will be one set of probability that falls in between [0 to 1]).

So the formula for detecting an object in a feature map with S * S grid cells with 5 components as said above can be shownas

[s*s*(B*(5+C))] Equation 4

Where C is the number of classes (for our case C = 1) and the five components can be explained in terms of mathematicalexpressions as shown below, $b_x=\sigma(t_x)+C_x$ Equatwion 5

$b_y=\sigma(t_y)+C_y$ Equation 6 $b_w=p_w\exp(t_w)$ Equation7

$b_h=p_h\exp(t_h)$ Equation 8

$(object)=\sigma(t_o)$ Equation 9

where t_x,t_y,t_w,t_h are the coordinates of the bounding boxes whose values should lie in between [0 to 1] and Pr(object) is theprobability of the detected object depending upon the center grid cell.

So, if we use a 1*1 kernel or a grid cell in a feature map in order to find the object in an image the whole equation willfall as

[1*1*(B*(5+C))] Equation 10

The probability of the individual class (Confidence) can be defined as the multiplication of the probability of the object and the Intersection over the union of the predicted and the ground truth boxes, i.e., Pr(Object) * IoU(predicted, Ground Truth values)

If we assume B = 3, then we have a kernel size of 1*1*18.

In real-time, the detection of an object through the camera depends on the camera focal length and distance, i.e., whether the object is far from the camera or near to the camera resulting in 90 percent occupancy of an image or 60 percent occupancy or30 percent occupancy in an image and so on. Hence, we have considered 3 strides in three different regions i.e., at 13*13, 26*26, and 52*52 layers for detecting smaller, medium, and larger objects. Due to the consideration of 3 strides at three different regions, we will get 9 anchor boxes at them which are clustered using K-means Clustering. And the number of bounding boxes in an image depends upon the occupancy of the likely class in the image. i.e., if it is a small image then the bounding box will be 13*13*18 and if it is medium then the number of bounding boxes per image will be 26*26*18 and 52*52*18 for large objects respectively.

Understanding our Loss Function:

In this section, the loss function of our deep learning face detection architecture is best explained in terms of three basicloss parameters based on the objects, their coordinates, and classes with their probabilities.

Step 3 Feature Extraction Technique (Face verification and Face Identification):

Feature extracting is a very important step in the face recognition system. The recognition rate of the system depends on the meaningful data extracted from the face image. If the features belong to different classes and the distance between theseclasses is bigger than these features are important for the recognition of the images. In general, they are two types of techniques used for Feature Extraction. They are the Traditional approach and the Deep Learning approach. In the traditional approach feature extraction from the face is done in 4 kinds of approaches. They are geometry features, holistic, feature-based, and hybrid methods.

Geometric-based models use specialized edge and contour detectors to find the location of a set of facial landmarks and to measure relative positions and distances between them. Whereas, **Holistic approaches** such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Local Preserving Positions (LPP), and Independent Component Analysis (ICA) are used in order to project face images onto a low-dimensional space that discards superfluous details and variations not needed for the recognition task. Unlike geometry-based methods, **feature-based methods** focus on extracting discriminative features rather than computing their geometries. They are SIFT, SURF, Brisk, ORB, and so on. Later they use chi-squared distance to compare these facial features collected in the database. A **Hybrid model** combines two or three models such as Holistic and Feature-based or Holistic and Geometric based and soon in order to extract these features and later compare them using classifiers and other distance metrics. As these 4 traditional approaches are very fast in extracting and comparing the encodings between the faces, they are not very robust in real-time. Hence with the advancement of technology neural networks lead the role in extracting facial features and comparing them in real-time with the help of some classifiers and distance metrics giving a breakthrough in the Face Recognition system.

With the improvements in deep learning architectures, there are four-mile stone systems on deep learning for face recognition that drove these innovations came through. They are Deep Face, the Deep ID series of systems, VGGFace, and Face-net. All these milestone architectures are trained with millions and millions of face recognition databases which requires huge servers, and months and months of training periods which is unrealistic for most face recognition system designers. Hence, we designed a new Feature extraction technique in which we extract 128 embeddings of a face for each person as shown in Fig 4. below and store those embeddings in our database and verify those embeddings with the help of a voting distance metric which takes the mean of the three-distance metrics (Chebyshev, Minkowski, and Cosine).

Our Deep Feature Extraction Model:

In our feature extraction technique, we squeezed layers of two deep learning architectures into one model and extracted 128 embeddings for each image with the help of a sigmoid activation function in the Dense Layers. The L2 normalization technique or Euclidean normalization technique is used above the sigmoid layer in order to calculate the distance of the vector coordinate from the origin of the vector space which is a positive distance value.

In our model, Convolution is the first layer to extract features from an input image. It preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs suchas an image matrix and a filter or kernel that generates ―Feature Maps‖ as outputs. In our process, we have taken the cropped face image from our custom detector and resized it into the shape of (64,64,3) and take a stride or kernel of (5*5) which strides across five pixels on the image matrix (both in h & w direction) and generates the feature maps according to the formula

Feature Maps = $(h - f_h + 1)(w - f_w + 1) * 1$

Equation 16 where, h,w are the height and width of the image and f_h, f_w is the height and width of the kernel.

Limitations of the Face Recognition System

Poor Image Quality Limits Facial Recognition's Effectiveness:

Image quality affects how well facial recognition algorithms work. The image quality of scanning video is quite low compared with that of a digital camera. Even high-definition video is, at best, 1080p (progressive scan); usually, it is 720p. These values are equivalent to about 2MP and 0.9MP, respectively, while an inexpensive digital camera attains 15MP. The difference is quite noticeable.

Small Image Sizes Make Facial Recognition More Difficult:

When a face-detection algorithm finds a face in an image or in a still from a video capture, the relative size of that face compared with the enrolled image size affects how well the face will be recognized. An already small image size, coupled with a target distance from the camera, means that the detected face is only 100 to 200 pixels on a side. Further, having to scan an image for varying face sizes is a processor-intensive activity. Most algorithms allow the specification of a face-size range to help eliminate false positives on detection and speed up image processing.

Different Face Angles Can Throw Off Facial Recognition's Reliability:

The relative angle of the target's face influences the recognition score profoundly. When a face is enrolled in the recognition software, usually multiple angles are used (profile, frontal, and 45-degree are common). Anything less than a frontal view affects the algorithm's capability to generate a template for the face. The more direct the image (both enrolled and probe image) and the higher its resolution, the higher the score of any resulting matches.

Data Processing and Storage Can Limit Facial Recognition Tech:

Even though the high-definition video is quite low in resolution when compared with digital camera images, it still occupies significant amounts of disk space. Processing every frame of video is an enormous undertaking, so usually, only a fraction (10 percent to 25 percent) is actually run through a recognition system. To minimize total processing time, agencies can useclusters of computers. However, adding computers involves considerable

data transfer over a network, which can be bound by input-output restrictions, further limiting processing speed. Ironically, humans are vastly superior to technology when it comes to facial recognition. But humans can only look for a few individuals at a time when watching a source video. A computer can compare many individuals against a database of thousands.

Experiment Evaluation

The Results of the algorithm can be calculated on the confusion matrix which is a table that is often used to describe the performance of a classification model (or ―classifier‖) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. In this scenario, we have calculated the accuracy of our algorithm on Bench Marking Dataset and achieved an overall accuracy of nearly 99 percent accuracy as shown below.

The overall accuracy of the algorithm based on benchmarking dataset is 1687/1716=0.983100233. This is the Face Recognition for attendance device

Predicted /Actual	Negative	Positive
False	133	3
True	24	1687

By facing the camera, the camera will capture the image and then pass it to the Raspberry Pi which is programmed to handle face recognition by implementing the Local Binary Patterns (LBP) algorithm. If the input image matches with the trained dataset image, then the attendance results will be stored in a CSV sheet. The database is connected to a simple mail transfer protocol (SMTP), which makes attendance reachable to any mail. The system has 98% accuracy with the dataset of 1000-person images.

REFERENCES

[1] X. Zhao, W. Zhang, G. Evangelopoulos, D. Huang, S. Shah, Y. Wang, I. Kakadiaris and L. Chen, ―Benchmarking Asymmetric 3D-2D Face Recognition Systems.‖ Pattern Analysis and Machine Intelligence, pp. 218–233, 2013.

[2] SE. Rekha and P. Ramaprasad, ―An efficient automated attendance management system based on Eigen Face recognition.‖ 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, pp. 605- 608, 2017.

[3] K. H. Pun, Y. S. Moon, C. C. Tsang, C. T. Chow, S. M. Chan, ―A face recognition embedded system.‖ Biometric Technology for Human Identication, vol. 5779, no. 2, p. 390, 2005.

[4] J. Joseph and K. P. Zacharia, ―Automatic Attendance Management System Using Face Recognition.‖ International Journal of Science and Research (IJSR), ISSN (Online), pp. 2319-7064, 2013.

[5] S. Lukas, A. Mitra, R. Desanti and D. Krisnadi, ―Student attendance system in classroom using face recognition techniques

[6] V. Mohanraj, V. Vaidehi, S. Vasuhi and R. Kumar, ―A Novel Approach for Face Recognition under Varying Illumination

[7] Conditions.‖ International Journal of Intelligent Information Technologies, pp. 218–233, 2018.

[8] J. Khorsheed and K. Yurtkan, ―Analysis of Local Binary Patterns for face recognition under varying facial expressions.‖ 24th Signal Processing and Communication Application Conference (SIU), pp. 2085-2088, 2016.

[9] H. Ebrahimpour and A. Kouzani, ―Face Recognition Using Bagging KNN.‖ 2018

[10] S. Eleftheriadis, O. Rudovic and M. Pantic, "Discriminative shared Gaussian processes for multiview and view- invariant facial expression recognition", *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 189-204, Jan. 2015.

[11] Show in Context View Article Full Text: PDF (2865KB) Google Scholar

[12] T. Hassner, S. Harel, E. Paz and R. Enbar, "Effective face frontalization in unconstrained images", *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4295-4304, Jun. 2015.
[13] Show in Context View Article Full Text: PDF (1131KB) Google Scholar
[14] X. Yin and X. Liu, "Multi-task convolutional neural network for pose-invariant face recognition", *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 964-975, Feb. 2018.
[15] Show in Context View Article Full Text: PDF (4423KB) Google Scholar
[16] K. Mahantesh and H. J. Jambukesh, "A transform domain approach to solve PIE problem in face recognition", *Proc. Int. Conf. Recent Adv. Electron. Commun. Technol. (ICRAECT)*, pp. 270-274, Mar. 2017.
[17] Show in Context View Article Full Text: PDF (252KB) Google Scholar
[18] D. Zhang and S. Zhu, "Face recognition based on collaborative representation discriminant projections", *Proc. Int. Conf. Intell. Transp. Big Data Smart City (ICITBS)*, pp. 264-266, Jan. 2019.
[19] Show in Context View Article Full Text: PDF (1208KB) Google Scholar

[20] H. Tu, K. Li and Q. Zhao, "Robust face recognition with assistance of pose and expression normalized albedoimages", *Proc. 5th Int. Conf. Comput. Artif. Intell. (ICCAI)*, pp. 93-99, 2019.
Show in Context Google Scholar
[21] M. Pietikãinen, "Local binary patterns", *Scholarpedia*, vol. 5, no. 3, pp. 9775, 2010.
Show in Context CrossRef Google Scholar
[22] T. Ahonen, A. Hadid and M. Pietikäinen, "Face recognition with local binary patterns" in Computer Vision—ECCV,Springer, pp. 469-481, 2004. Show in Context Google Scholar