International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

# Real Time Identification and Detection of Face for Surveillance

*Anusha Bhilawadi[1], Pratik[2], Anusha Teggi[3], Ritu Sansuddi[4], Dr. S. F. Rodd[5]*

[1]Dept. of Computer Science Engineering S. G. Balekundri Institute of Technology. Belagavi, Karnataka, India. bhilawadianusha@gmail.com
[2]Dept. of Computer Science Engineering S. G. Balekundri Institute of Technology. Belagavi, Karnataka, India. pratikspparasagond@gmail.com
[3]Dept. of Computer Science Engineering S. G. Balekundri Institute of Technology. Belagavi, Karnataka, India. teggianusha@gmail.com
[4]Dept. of Computer Science Engineering S. G. Balekundri Institute of Technology. Belagavi, Karnataka, India. sansuddi.ritu@gmail.com
[5]Proferssor, Dept. of Computer Science Engineering S. G. Balekundri Institute of Technology. Belagavi, Karnataka, India. sfroddjforkts@gmail.com

**ABSTRACT—**

Individuals are recognised by their faces. Face detection and face recognition are the two key components of face recognition, which is employed in this research to identify identities. Both detection and recognition are different from one another; detection looks for faces, while recognition identifies those faces. Effective way of the face detection which is part of the computer vision which helps the real-time detection of faces and performance and real-time accuracy In this project face recognition and face detection using the OpenCV library which is used for computer vision task and we are using the algorithm to detect the faces and recognize it with help of OpenCV which provides built-in methods so it does the task easy and improves code complexity. The algorithm we are using for project is an LBPH algorithm which tells us about texture classification and which labels the images with the pixel the result of the pixel is in a binary number

***Keywords—Face Recognition, OpenCV, LBPH, Face detection.***

## 1. Introduction

Face detection and recognition technologies are employed to locate and identify human faces within images or live video feeds. The primary objective of face detection is to detect all faces present in an image or video, regardless of their position, orientation, or lighting conditions. On the other hand, face recognition entails training the system to identify specific individuals by utilizing multiple images of their faces. In applications such as video surveillance systems, real-time facial images are captured by IP or web cameras and transmitted to a computer. The system then employs pre-recorded facial data to identify and match the faces of known individuals. This process involves capturing several frontal images of an individual's face and using them to train the system to recognize that particular person.

In real-world scenarios, deploying face recognition systems in real-time environments can pose challenges as the outcomes may vary compared to controlled testing conditions. The Haar feature-based cascade classifier, a machine learning technique for object detection, particularly in face detection, was co-developed by Paul Viola and Michael Jones. The approach involves training a cascade function using two sets of images: positive images (containing faces) and negative images (without faces). To train the classifier, features are extracted by evaluating the disparity between the pixel sums within a white rectangle and a black rectangle. This process generates an extensive set of features, which are subsequently employed to train the classifier in identifying faces in different images. To achieve accurate results, the algorithm requires a substantial number of positive and negative images for training. Once the classifier is trained, it becomes capable of swiftly and precisely detecting faces in new images, making it a valuable tool for face detection tasks. The Local Binary Patterns Histograms (LBPH) algorithm is a commonly utilized computer vision technique for detecting and recognizing faces. The LBPH algorithm works by extracting texture information from an image by comparing each pixel with its neighboring pixels. In the LBPH algorithm, a face image is partitioned into smaller regions, and a binary code is generated within each region based on pixel intensities. These binary codes are combined to form a histogram representing the face, enabling recognition by comparing the histogram of an unknown face with those stored in a dataset. By assessing the similarity between the histograms, the algorithm identifies the unknown face as the known face with the most closely matching histogram. In simpler terms, LBPH analyzes the texture patterns in a face image by examining the relationship between pixels. It creates a histogram representing the face and compares it to known faces to determine the closest match, allowing for face recognition.

OpenCV, a widely recognized open source computer vision and machine learning software library It is employed for face detection and identification, utilizing pre-trained classifiers to detect faces in images or video streams. When using OpenCV for face detection, the library employs pre-trained classifiers to locate faces within the given image or video. These classifiers are designed to recognize specific facial features such as the eyes, nose, and mouth. They utilize either Haar cascades or LBP (Local Binary Patterns) cascades to identify these features and subsequently locate and identify faces within the image or video stream.

In simpler terms, OpenCV's face detection functionality employs pre-existing classifiers to detect faces by identifying key facial features. It uses either Haar cascades or LBP cascades to recognize these features and thereby identifies and localizes faces in the provided image or video.

Face identification comes next after face detection.Face detection, face alignment, feature extraction, and classification are some of the processes in the face recognition process using OpenCV.

To accomplish face detection and identification, OpenCV offers a variety of libraries and tools, including CascadeClassifier for detecting faces using Haar Cascades and LBPHFaceRecognizer Face recognition is accomplished through the utilization of the Local Binary Pattern (LBP) algorithm among others. Building precise and effective face detection and identification systems for a variety of applications, including security, surveillance, and access control, is doable with these tools and libraries.

## II. LITERATURE REVIEW

In study [1], It is focuses on exploring face detection using OpenCV, with a particular emphasis on algorithms like Adaboost and Haar cascades. The objective is to develop an application that can track and detect faces in videos and cameras for various purposes. The paper includes a comprehensive analysis, including a tabular comparison of different algorithms, aiming to provide readers with a comprehensive understanding of face detection best practices. Haar cascades are identified as the most efficient algorithm for face detection, taking into account factors such as space and time efficiency. The research also explores the differences between implementing face detection using OpenCV and Matlab, discussing the pros and cons of each approach. Based on the findings, it is suggested that Haar cascades and the Camshift algorithm outperform motion detection in terms of performance and accuracy. However, if time is a critical factor, the Camshift algorithm and motion detection algorithm are more suitable options. Ultimately, the research demonstrates that Haar cascades deliver improved Accuracy in Detecting facial expressions.

In study [2], It focuses on detecting faces in indoor surveillance videos using a proposed method that combines skin color segmentation, Haar feature extraction, and classification. The method involves converting RGB images to the YCbCr color space and analyzing histograms to extract skin pixels. Haar features are then utilized, and a cascaded AdaBoost classifier is employed to classify faces into frontal and side-view faces while filtering out non-face regions. The experimental results demonstrate the effectiveness of the proposed method in achieving its objectives detected an average of 70.96% of frontal faces, but the performance for side-view faces was relatively low, with an average of 32.67%. The study highlights the influence of human-computer interaction on face detection methods and provides insights into the challenges associated with detecting faces in different viewpoints. This paper contributes to face detection by integrating skin color segmentation and Haar features for feature extraction. Skin color segmentation helps remove non-skin pixels, although it may lead to false segmentation in areas with similar colors to human skin. The Haar features stage involves exploring various parameters such as the number of stages and merge threshold values. The face detector successfully detects frontal faces, half-profile faces, and full-profile faces, but occlusion presents a limitation. The cascaded AdaBoost classifier is to classify face regions as frontal or side-view faces, but misclassifications can occur due to a reduction in Positive Samples and similarities in Haar features between the two face types. Future research will focus on incorporating additional features, such as contour analysis, to enhance the detection performance of side-view faces.

In study [3], proposes a comprehensive method for training face models and recognizing human faces using Python, PyQt for user interface design, and the OpenCV library for image processing. The fundamental elements of the system consist of face model training and face recognition modules that utilize algorithms like EigenFace, FisherFace, and Local Binary Pattern (LBP) from the OpenCV library.To enhance recognition accuracy, the method incorporates parameter adjustments such as recognition threshold and feature point optimization. By storing face images and corresponding tags in CSV format, the face model training module utilizes the OpenCV library to process and train the data. The proposed method demonstrates high recognition efficiency, laying a robust foundation for future advancements in computer vision research. This research paper concentrates on developing a face detection and recognition system designed specifically for the Linux platform using Python programming language. The performance of EigenFace, FisherFace, and LBP algorithms is extensively examined through thorough testing and analysis. The study uncovers the strengths, weaknesses, and suitable application scenarios for each algorithm. Acknowledging that effective face recognition necessitates the consideration of both local and global facial information, the research emphasizes the significance of combining these elements. Additionally, the paper investigates the effectiveness of integrating multiple features and classifiers to enhance the performance of facial recognition.

In study [4], Accurately and efficiently detecting faces in both images and real-time scenarios is a complex problem that requires effective solutions. This study aimed to determine the most suitable face detection algorithm, particularly for real-time applications. Three algorithms, namely Haar-like features, Local Binary Pattern (LBP), and Histogram of Oriented Gradient (HOG), were evaluated and compared. The evaluation was conducted using the RECOLA database and implemented using Python with the OpenCV and dlib libraries. The results indicated that the Histogram of Oriented Gradient algorithm achieved the highest detection rate of 62.16%, while the LBP algorithm demonstrated the fastest frame rates at 33.79 frames per second. The Adaboost learning algorithm was utilized to construct strong classifiers capable of determining whether a sub-window contained a face. This involved selecting the most important features from a large set of Haar features to build weak classifiers. During training, these weak classifiers were combined to form strong classifiers that could be implemented as a cascade classifier, enabling rapid rejection of non-face windows. This study aimed to assess and compare various face detection techniques, such as Haar-like cascade, Local Binary Pattern cascade, and Histogram of Oriented Gradients with Support Vector Machine, in terms of their performance for real-time face detection applications. The objective was to identify an approach that combines both high detection rates and fast processing speeds. The authors employed videos from the RECOLA multimodal database capturing spontaneous reactions of subjects for testing the techniques. Detection rates and processing speeds were compared, with the HOG method exhibiting the highest detection rate and the LBP method demonstrating the fastest performance. To further enhance the study, the authors recommended evaluating all three techniques using the same cascade classifier and exploring videos with more diverse real-world conditions encompassing lighting variations, different orientations, poses, expressions, as well as videos involving multiple subjects.

In study [5], The detection and recognition of human faces pose significant challenges in the field of computer vision. This research paper addresses these challenges by exploring and comparing two distinct algorithms: the Local Binary Patterns Histogram (LBPH) algorithm and Convolutional Neural Networks (CNN). The evaluation of these algorithms involved multiple datasets, including ORL, FEI, and a self-created dataset comprising 12 individuals with 100 images each. The results indicated that the CNN algorithm outperformed the LBPH algorithm in terms of accuracy and performance. The findings of this study have valuable implications in various applications such as security systems, attendance monitoring, crime prevention, locating missing persons, and access control to secure areas. By comparing the performance of two face recognition algorithms across different datasets, this research provides valuable insights for future applications. The primary objective of this study is to improve real-time face detection and recognition systems. To accomplish this, the research implements the Local Binary Patterns Histogram algorithm and Convolutional Neural Networks algorithm, utilizing the ORL, FEI, and a self-created dataset. The accuracy rates of these algorithms are extensively analyzed for each dataset. The notable aspect of this work lies in the comparative analysis of two distinct face recognition algorithms. In the future, the system has the potential to be integrated into a student monitoring system, streamlining attendance tracking processes efficiently.

## III. METHODOLOGY

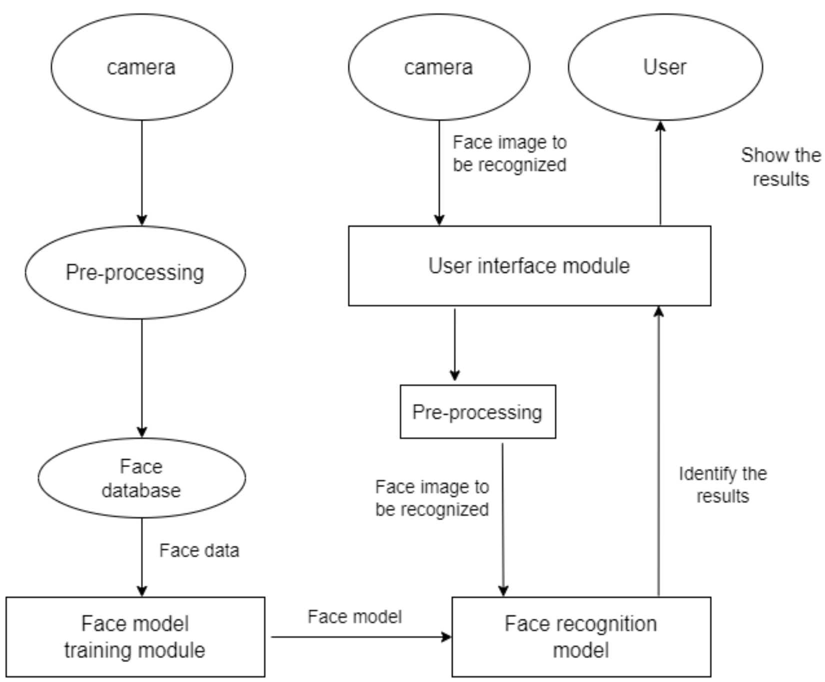


Fig 1. Data Flow diagram

The first step in building a face detection and identification system is to collect data. In this step, we need to gather images of individuals whose faces we want the system to detect and identify. This is typically done by capturing images of the individuals using a camera or sourcing images from an existing dataset.The images need to be captured in a controlled environment with consistent lighting and background to ensure the quality and accuracy of the data. Multiple images of each individual should be captured from different angles and with different facial expressions to capture variability in appearance. Once the images are captured, they need to be pre-processed to improve the quality of images and remove any noise or unwanted artifacts. This includes techniques such as image cropping, resizing, normalization, and contrast enhancement.

After collecting the dataset of faces, the next step in face detection and identification is to utilize the Haar Cascade Classifier Algorithm to identify faces in images and Videos captured by a camera. This algorithm involves using a pre-trained Model that has been specifically trained to detect faces based on certain features. The model has learned to recognize Patterns and shapes that are typically associated with human faces, allowing it to accurately detect and locate faces in new images or videos.

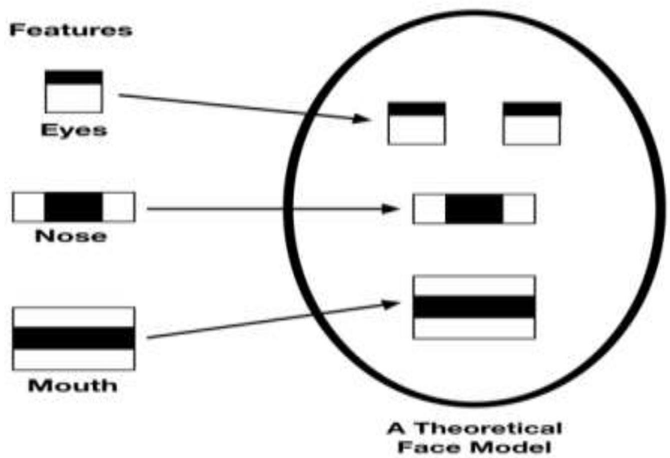


Fig 2. Haar Features For Face Recognition

To identify an edge in an image using Haar features, the algorithm scans the entire image. Each Haar feature calculates a single value by subtracting the sum of pixel intensities in the white rectangle from the sum of pixel intensities in the black rectangle. This computed value indicates the presence or absence of a specific feature in a particular region of the image. The formula for computing the value of a Haar feature is expressed as Value = Sum of pixel intensities in the white area minus the sum of pixel intensities in the black area.

Fig 3. Detection of Faces in different parts

The Haar feature is systematically applied pixel by pixel throughout the image to search for a specific feature. The traversal of Haar features begins at the top left corner of the image and proceeds towards the bottom right corner. Furthermore, the Haar feature is applied at various sizes to ensure that all possible features are considered. This iterative process continues for each Haar feature within the classifier until all features have been examined. Once all Haar-like features have been computed, the model can be saved as a YAML file. This YAML file can then be utilized in real-time applications to detect faces in images or video streams.

After detecting the faces, the LBPH (Local Binary Patterns Histogram) algorithm is employed to extract features from the images and videos using the OpenCV library. To begin, a dataset of facial images for the individuals we want to recognize is required. Each image needs to be associated with a unique ID, such as a number or name, to aid the algorithm in identifying them. Images of the same person should have the same ID. Once the dataset is prepared, we can proceed with the computational steps of the LBPH algorithm to train it..

The primary objective of the algorithm is to improve the facial features present in the original image by generating an intermediate image. This is achieved by employing a sliding window technique, where the algorithm analyzes each pixel within the window. By considering the pixel's neighboring pixels, a binary value is calculated to capture their relationship. These binary values are subsequently utilized to construct the intermediate image, which enhances the facial features.

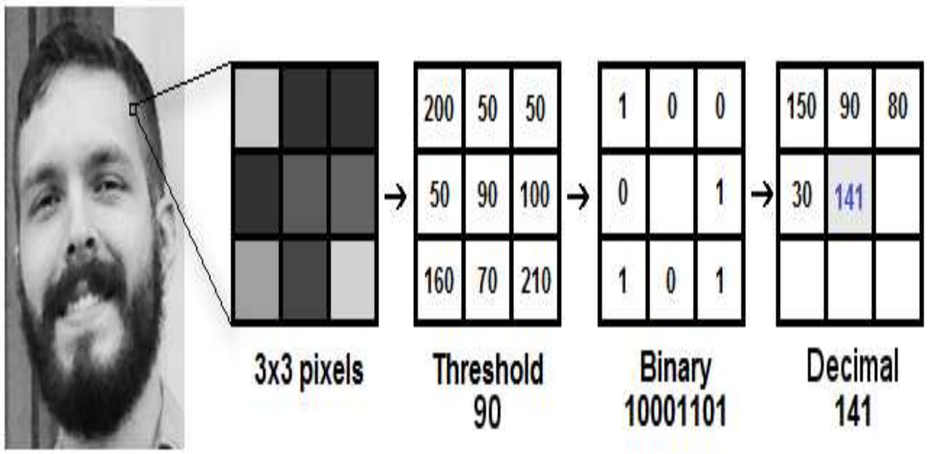


Fig 4 . Binary Value Computation of the Image

The image generated is then divided into multiple grids based on the Grid X and Grid Y parameters. This division is used to extract histograms for each grid separately.

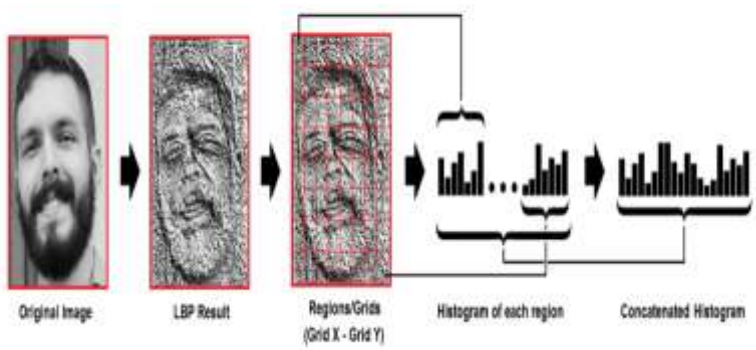


Fig 5. Extracting Histogram from Image

Here We have grayscale image, each histogram from each grid will have 256 Positions (0-255) that represent the occurences of each pixel intensity .For each grid, we calculate a separate histogram of pixel intensities.Each histogram is then concatenated to create a bigger histogram.The final histogram contains 16,384 positions if we have 8x8 grids.The final histogram represents the characteristics of the original image.

Then trained algorithm uses the histograms generated from each image in the Training Dataset to represent them. Then, for a new input images, same steps are performed to generate its histogram and compare it with the histograms of the training images to recognize the face.

After the Face Detection and Identification model is trained, the System can be used to mark attendance in real-time. If a person's face matches the pre-existing faceprint in the system, the system will mark them as present, and if not, they will be marked as absent. This allows for automatic and accurate attendance tracking, without the need for manual input or verification. The system continuously analyzes faces in real-time, ensuring that attendance is updated as individuals enter or leave the area being monitored.

After attendance is marked in real-time by thesystem, the attendance data is stored in dataset. This dataset can be accessed and updated as needed, providing a real-time record of attendance for a particular event or class.

## RESULT AND DISCUSSION

A Face Detection and Recognition System is designed to automatically identify and locate human faces in images or videos, and subsequently match or verify the identity of individuals based on their facial features.

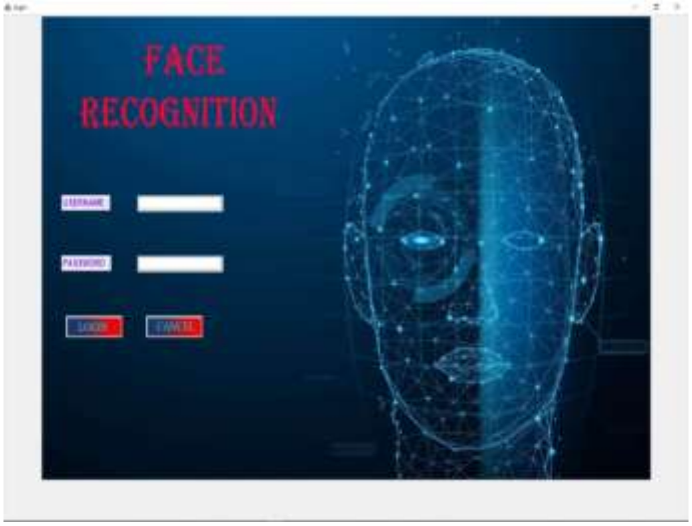


Fig 6. Login Page

To access the system or a specific application, users initiate the login process, usually through a user interface.

Users may be prompted to provide their username or any other required identification information.


Fig 7. Dashboard page


Fig 8. Students Enrollment

During the enrollment phase, users are typically required to provide their personals Informations such as name, ID, and possibly other identifying details.

Users' faces are captured through a camera or uploaded Images, and the system extracts facial features or creates a unique template representing their face.

The extracted facial features or templates are stored securely in a dataset, associated with the user's account or identifier.

## IV. CONCLUSION

While face detection and identification technologies have demonstrated their effectiveness in networked video surveillance systems, they are not without limitations in terms of accuracy. One such limitation is that as the number of samples or faces to be recognized increases, the accuracy of the algorithms tends to decrease. This indicates that there is a need for improvement in the technology's capability to handle larger datasets and maintain high levels of accuracy.

To enhance the accuracy of face detection and identification algorithms, one possible approach is to utilize advanced hardware resources like high-configuration computers and high-resolution cameras. By employing such hardware, it becomes possible to provide the algorithms with adequate processing power and improved image quality. This, in turn, enables the algorithms to effectively and accurately detect and identify faces in real-time scenarios. Despite these challenges, the use of face detection and identification in networked video surveillance systems has shown promise in enhancing security measures and situational awareness. As with any technology, it is important to balance the benefits with potential ethical and privacy concerns to ensure that its implementation is responsible and ethical.

Face detection algorithms are utilized to identify and locate faces within a video stream, whereas face identification algorithms are employed to match the detected faces with known individuals. These advanced technologies find applications in diverse sectors, including law enforcement, retail, and transportation industries.

## V. FUTURE SCOPE

Integrating facial recognition systems with IoT devices can greatly improve the overall performance and effectiveness of the system. This integration allows for the creation of a cohesive system that seamlessly combines facial recognition capabilities with various IoT devices. As a result, organizations can benefit from enhanced security measures that are efficient and automated, providing a robust solution for their security needs.This integration allows for remote monitoring and management of security systems, leading to enhanced security measures. Cloud-based facial recognition systems offer additional advantages, including scalability, flexibility, and accessibility. Cloud-based systems offer organizations the opportunity to leverage the advantages of facial recognition technology without requiring significant investments in hardware infrastructure. By utilizing cloud-based solutions, organizations can access facial recognition capabilities remotely, taking advantage of scalability, flexibility, and accessibility. Additionally, cloud-based systems provide the benefit of receiving real-time updates and improvements, ensuring that the system remains current and effective. Moreover, advanced Deep Learning models like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have shown promising potential in enhancing the accuracy and efficiency of face detection and identification systems. These models have the ability to learn intricate facial features and patterns, enabling highly accurate identification of individuals even in challenging environments.


### *Acknowledgment*

We thank Dr. S.F Rodd for their expertise and assistance in arranging and writing the Manuscript , we also thank Mr. . Sushant M for guiding us.

We express our gratitude to Dr.B. S.Halakarnimath

HOD CSE sdept. SGBIT and Dr. B. R. Patagundi, principal, SGBIT, Belagavi for funding our paper.


### References


Kruti Goyal, “Face Detection and Tracking Using OpenCV”, International Conference on Electronics Communication and Aerospace Technology ICECA, 2017.

Sharmeena Naido Kuala Lumpur, Rosalyn R. Porle, “Face Detection Using Colour and Haar Features for Indoor Surveillance”, 2020

Limei Fu,Xinxin Shao, Dalian Neusoft, “Reseach and Implementation of Face Detection, Tracking and Recognition Based on Video”, International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 2020.

Amal Adouani, Wiem Mimoun Ben Henia, Zied Lachiri, “Comparative study of face detection methods in spontaneous videos”, 2019.

Agnihotram Venkata Sripriya, Mungi Geethika, Vaddi Radhesyam. “Real Time Detection and Recognition of Human Faces”, 2020.