# ADVANCED CONTINUOUS MULTI-FACE TRACKING AND MONITORING SYSTEM

*A Project Report Submitted*

by

**UDIT SETHI**

**(2001EE93)**

*In Partial Fulfilment*

*of the Requirements for the award of the degree*

**BACHELOR OF TECHNOLOGY**

**DEPARTMENT OF ELECTRICAL ENGINEERING**

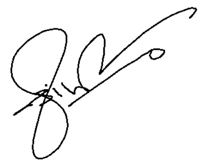
**INDIAN INSTITUTE OF TECHNOLOGY PATNA**

**May 2024**

# THESIS CERTIFICATE

This is to certify that the thesis titled **Advanced Continuous Multi-Face Tracking and Monitoring System**, submitted by **Udit Sethi** , to the Indian Institute of Technology, Patna, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Jawar Singh, PhD
Professor
Department of Electrical Engineering
Indian Institute of Technology Patna



**Prof. Jawar Singh**
Supervisor
Professor
Dept. of Electrical Engineering
IIT-Patna, 801103

Place: Patna

Date: 9th May 2024

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Prof. Jawar Singh, for accepting our project proposal and guiding us throughout the project. His valuable feedback, constructive criticism, and encouragement were instrumental in shaping this project and bringing it to fruition. Also, we are thankful to our friends and well-wishers for their constant support.

# ABSTRACT

Facial recognition emerges as a highly advantageous application, playing a pivotal role in the technological landscape. Recognizing faces holds considerable significance, especially in authentication applications like attendance tracking. Employing facial recognition in attendance systems involves identifying individuals based on their facial characteristics using advanced computing technology and surveillance. The progression of this process aims to modernize traditional manual attendance-taking methods. Current attendance tracking approaches are laborious and inefficient, prone to manipulation through manual record-keeping. Traditional attendance check systems like fingerprint or card scanning also face vulnerabilities to fraudulent proxies. To tackle these challenges, a novel system is proposed, leveraging algorithms such as Histogram of Oriented Gradient (HOG), Convolutional Neural Network (CNN), and Support Vector Machine (SVM). Following face recognition, attendance reports are automatically generated, stored, and managed in Excel format. The system's robustness is tested across diverse conditions including varying lighting, head movements, and different distances between faces and cameras. The proposed system showcases efficiency and reliability in seamlessly marking attendance in classroom settings, requiring minimal time investment and no manual intervention. Moreover, it proves cost-effective due to its minimal installation requirements.

# TABLE OF CONTENTS

# LIST OF FIGURES

# ABBREVIATIONS

| | |
|---|---|
| **HOG** | Histogram of Oriented Gradient |
| **CNN** | Convolutional Neural Network |
| **SVM** | Support Vector Machine |
| **ASD** | Active Student Detection |
| **FLE** | Face Land-mark Estimation |
| **FR** | Face Recognition |
| **KNN** | K-th Nearest Neighbour |

# CHAPTER 1

# INTRODUCTION

## 1.1 Overview

Attendance is crucial for administrative purposes, yet it often poses challenges and is susceptible to errors. Traditional attendance-taking methods encounter difficulties, particularly in large groups, leading to potential inaccuracies in record-keeping [1]. Organizations have explored various strategies, from manual paper-based approaches to digital solutions like fingerprinting and card-based systems. However, these methods present their own challenges, such as long queues and reliance on identity cards [2]. Biometric-based digital attendance systems, specifically those utilizing facial recognition, offer a more efficient alternative. While facial recognition technology has shown promise in improving attendance tracking, it still encounters obstacles like variations in lighting, changes in pose, and potential obstructions [3]. To tackle these challenges, this paper proposes an innovative method that enhances existing systems by strategically deploying multiple cameras within classrooms. These cameras capture video for face detection, feature extraction, and recognition, thereby enhancing the accuracy and efficiency of attendance management [4]. To counteract potential blurring caused by student movement, an advanced image processing step precedes face detection to improve image clarity [5, 6]. The system initially identifies faces in the images, extracts facial features, and then matches these features to recognize students' identities. Face recognition is achieved using Convolutional Neural Networks (CNN) and Support Vector Machine (SVM) algorithms. Following these processes, the system records students' attendance in Excel format, including their names and entry times. Moreover, this system is cost-effective due to its minimal hardware requirements.

## 1.2 Motivation

The realm where face recognition truly shines is in biometrics, particularly for authentication, streamlining various tasks. It's a widely adopted technology with diverse applications, ranging from attendance tracking in schools and colleges to aiding in law enforcement against criminals and terrorists, bolstering public safety and national security.

Governments leverage face recognition for tasks like voter list verification, finding missing persons, conducting population census, and facilitating immigration processes. Additionally, it plays a pivotal role in combating internet scams, safeguarding E-commerce transactions, and finds extensive use in the realms of medicine and healthcare.

Given its multifaceted utility, there's a growing demand for real-time face recognition systems, both among the public and governmental entities. Deploying such robust systems would significantly simplify numerous activities. The processing of images captured by cameras occurs on hardware-powered computers, with researchers delving into various aspects such as..

- Recognition of the visible object
- Face detection and recognition
- Tracking the face
- Recognizing human emotions
- Human gender and age determination
- Sign language detection.

## 1.3 Main Challenges in Automatic Face Recognition Algorithms

The primary obstacles encountered in popular automatic facial recognition algorithms are outlined below:

1. **Facial expression diversity:** Effective facial recognition algorithms should be capable of adapting to changes in facial expressions resulting from the subject's emotional states.

2. **Variations in lighting conditions:** Facial images are often captured in diverse lighting environments, requiring facial recognition algorithms to extract features that are robust to changes in illumination.

3. **Adaptation to pose variations:** Automatic algorithms must be flexible enough to accommodate various head poses assumed by subjects.

4. **Management of scaling discrepancies:** Automatic algorithms must be able to handle differences in size or resolution when comparing a new image with those in the dataset.

5. **Overcoming obstacles:** Physical obstructions, such as other objects in the scene or personal attributes like glasses, beards, or mustaches, may obscure facial features, presenting challenges that facial recognition algorithms must overcome.

Figure 1.1: Challenges

## 1.4 Problem Statement

**Advance Attendance System:**

The primary goal is to investigate an algorithm for use in biometric attendance systems, utilizing appropriate methods and available inputs. In a classroom environment with a substantial student population, such as 100 or more students, efficiently and accurately recording their attendance simultaneously presents a considerable logistical hurdle. Hence, there's a necessity for an automated and efficient solution to mark the attendance of a large number of students simultaneously during live classroom sessions.

## 1.5 Steps of FR Process

In this chapter, a comprehensive multi-level taxonomy focusing on face recognition is introduced. The taxonomy consists of four main levels: face structure, feature extraction approach, feature support, and sub-approach of feature extraction. Although various methods are available for performing these comparisons, the fundamental steps usually remain consistent. The following steps delineate a generic automated face recognition model.

1. **Acquire:** This step involves capturing the face.
2. **Detection:** Face detection is used to isolate the facial area from the background.
3. **Alignment:** If the face is not perfectly aligned vertically in the camera, alignment is necessary.
4. **Extraction:** Facial features that are unique to each individual are used to create face templates and a face print.
5. **Matching:** Scores are generated by matching face prints and face templates which stored in the database.
6. **Reporting:** The generated scores are utilized to determine the final matches.

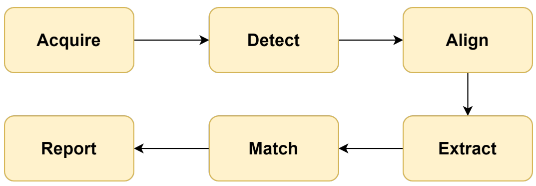



Figure 1.2: Steps of Face Recognition

Traditionally, a Facial Recognition (FR) system consists of four modules, as illustrated in Figure 2.3:

- Image Acquisition: This involves capturing images using visible cameras.
- Face Detection and Alignment: The process of face detection involves identifying the position of a face within an image, while alignment focuses on standardizing the face's orientation. Alignment ensures that the face is centrally positioned and that the line connecting the eye's centers aligns parallel to the horizontal axis. This standardization significantly improves the precision of face recognition.
- Feature Extraction: This component converts the raw pixel information in an image into a more polished and practical format, such as matrices ideal for comparison. It encompasses manually crafted features like Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG), along with deep learning characteristics referred to as embeddings.
- Matching: The process of face recognition includes comparing the extracted features. This task is categorized into two parts:

  Face Identification: Here, the system searches for a specific face within a database containing numerous faces, often termed as 1-to-n matching, with 'n' representing the total faces in the database.

  Face Verification: This step involves comparing one face with another, known as 1-to-1 matching. The accuracy of a face recognition system is determined by match scores, which indicate the level of similarity between a gallery image (an image stored in the database) and a probe image (the input or query image).

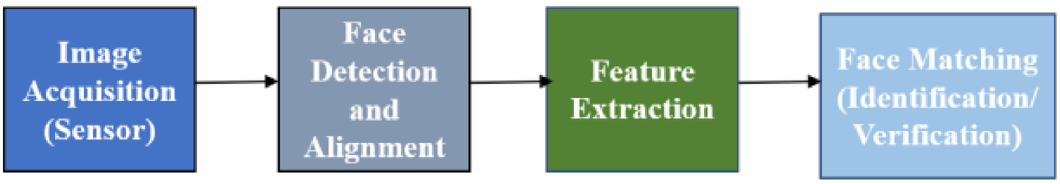



Figure 1.3: Four Modules

# CHAPTER 2

# LITERATURE SURVEY

**Thida Nyein, et al. [1]** proposed a Face Recognition Attendance System utilizing FaceNet and Support Vector Machine (SVM). The system comprised three main stages:

Pre-processing of Raw Data: Raw data underwent face alignment, followed by conversion into a training dataset. Images were then trained using a model and classifier. FaceNet facilitated feature extraction, while Support Vector Machine (SVM) handled classification.

Flow of the Proposed System: The system commenced by taking an image as input, performing face detection via OpenCV. Subsequently, FaceNet extracted features and embedded them. Feature matching was then executed using SVM against the trained dataset. Attendance was recorded based on identified faces, with an Excel file generated for attendance records.

Limitations: Despite achieving an 80 percent accuracy rate, the system had limitations. FaceNet encountered difficulties in accurately detecting faces when individuals wore accessories, thereby affecting the algorithm's performance and reducing accuracy.

While this system laid the groundwork for the Face Recognition Attendance System, enhancements are necessary to address FaceNet's challenges in accurately detecting faces with accessories.

**Ketan Mahajan, et al. [2]** introduced a dual-phase attendance system strategy incorporating Probability-based Face Mask Pre-Filtering (PFMPF) and Pixel-based Hierarchical Feature Ad-boosting (PBHFA) algorithms. These algorithms were tailored to overcome the shortcomings associated with the Harr cascade method. The system architecture comprised two primary phases:

Training Phase: This phase consisted of two steps. Initially, face detection was conducted using the Viola-Jones algorithm, followed by feature extraction using the PCA algorithm.

Testing Phase: The testing phase encompassed two sections - training dataset and testing dataset. However, the system encountered several limitations, including prolonged training times, a high false detection rate, and effectiveness limited to frontal face views. Additionally, it lacked a face centralization method.

While the proposed system introduced innovative approaches to address specific challenges like mask pre-filtering and feature boosting, significant limitations affected its overall performance and reliability.

**Vidya Patil, et al. [3]** proposed an automatic student attendance marking system utilizing the Kth Nearest Neighbor (KNN) algorithm. The system workflow includes image acquisition from the camera, preprocessing via Histogram Equalization, face detection using the HAAR cascade algorithm, and feature extraction with Linear Discriminant Analysis (LDA). Face recognition is conducted using three algorithms: LDA, Support Vector Machine (SVM), and KNN.

However, the system encountered limitations. KNN's performance is affected by large datasets and sensitivity to noisy and missing data. The proposed system aims to mitigate these drawbacks.

**In Ali Elmahmudi, et al. [4]**, a deep face recognition system was proposed for recognizing partial faces. The methodology involved utilizing a pre-trained VGGF model for feature extraction, with cosine similarity employed for feature comparison and SVM for classification.

Interestingly, when dealing with partial faces in images, SVM's accuracy declined, while the Cosine similarity method maintained higher accuracy under such circumstances.

**Yohie Kawaguchi, et al. [6]**, introduced a model focused on face recognition and continuous surveillance. This system incorporates Active Student Detection (ASD) technology, utilizing two cameras installed inside the classroom: one as a sensing camera and the other for face detection. The proposed approach involves estimating seating areas using ASD and capturing students with the capturing camera.

To determine student presence, the system employs background frame subtraction and inter-frame subtraction. Notably, the authors addressed the linear sum assignment issue by correlating seats with students.

# CHAPTER 3

# PROPOSED SYSTEM

## 3.1 Framework

The system is simple to use and operates seamlessly. It maintains a database containing students' face images along with their names. Multiple cameras are strategically installed on classroom walls, ensuring comprehensive coverage of the entire area. These cameras capture video footage of the lecture sessions.

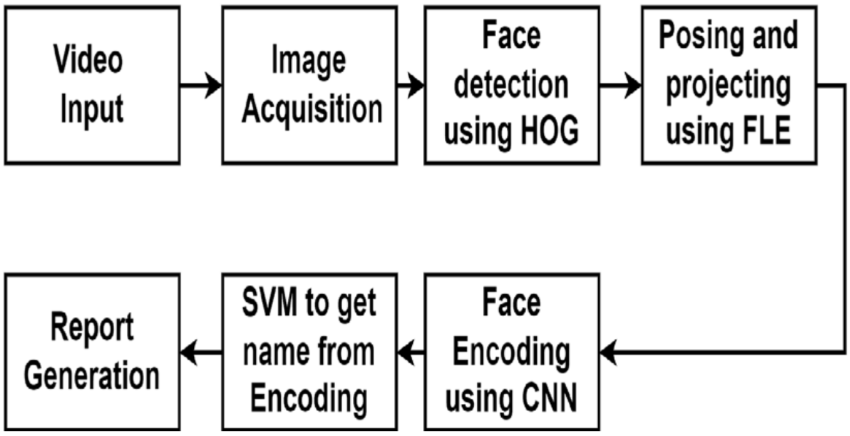



Figure 3.1: System Flowchart

The effectiveness of our system will be enhanced by the presence of multiple cameras, ensuring coverage of all students even if one camera misses certain individuals. Given the various poses students may adopt, the system can adapt to movements that initially evade face detection, capturing those faces in subsequent image acquisitions.

Upon detecting a face, the system proceeds to extract facial features and compares them with entries in the database. If a match is identified, the student's name is updated in the Excel sheet. To prevent duplicate markings, the system ensures each student is marked present only once per lecture, even if their face is detected multiple times.

The suggested system's flowchart, as depicted in Figure 3.1, delineates these steps for clarity.

## 3.2 System Design

Our system will feature ceiling-mounted cameras within the classroom, capturing video input directed towards the system. The system will then execute multiple operations and algorithms to produce an attendance report. The system comprises the following components:

- Capturing Images
- Detection of faces
- Utilizing FLE for posing and projecting
- Encoding of faces Using CNN
- Use of SVM for comparison
- Report Generation

### 3.2.1 Capturing Images

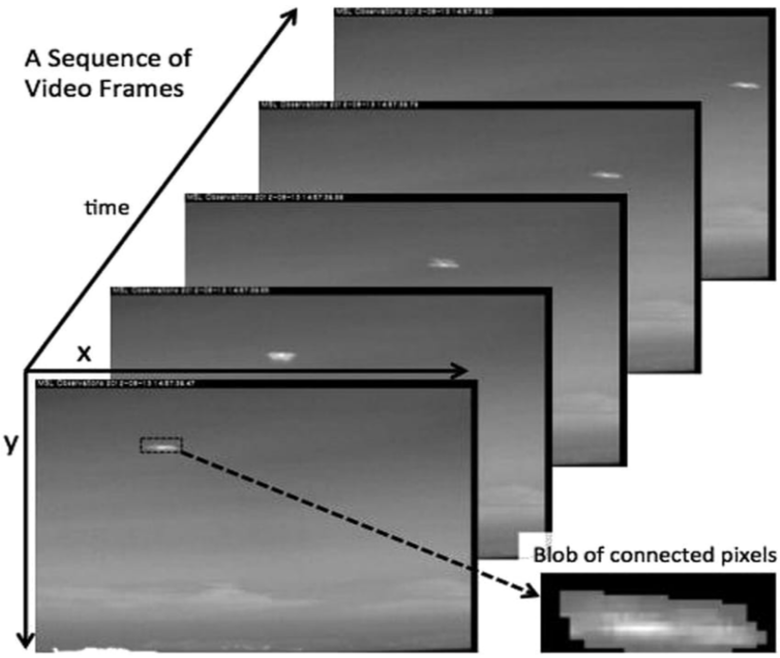



Figure 3.2: Capturing Images

Certainly, a video comprises a series of consecutive images captured swiftly. In our setup, we'll isolate individual frames from the camera's video feed and assess them for facial features. If a frame contains a face, we'll then proceed to align and process it.

Conversely, if a frame lacks facial features, we'll proceed to the subsequent frame in the video sequence.

### 3.2.2 Detection of faces utilizing HOG

Certainly, face detection poses a significant challenge within the machine learning domain. Our system employs the Histogram of Oriented Gradient (HOG) method for this task, extracting essential features from various facial images. These features are pivotal in our recognition framework. HOG, an acronym for Histogram of Oriented Gradient, is a prominent feature descriptor extensively utilized in machine vision for image analysis, notably in object detection. Its adeptness in identifying moving objects is widely acknowledged, rendering it a valuable asset in our face detection methodology.

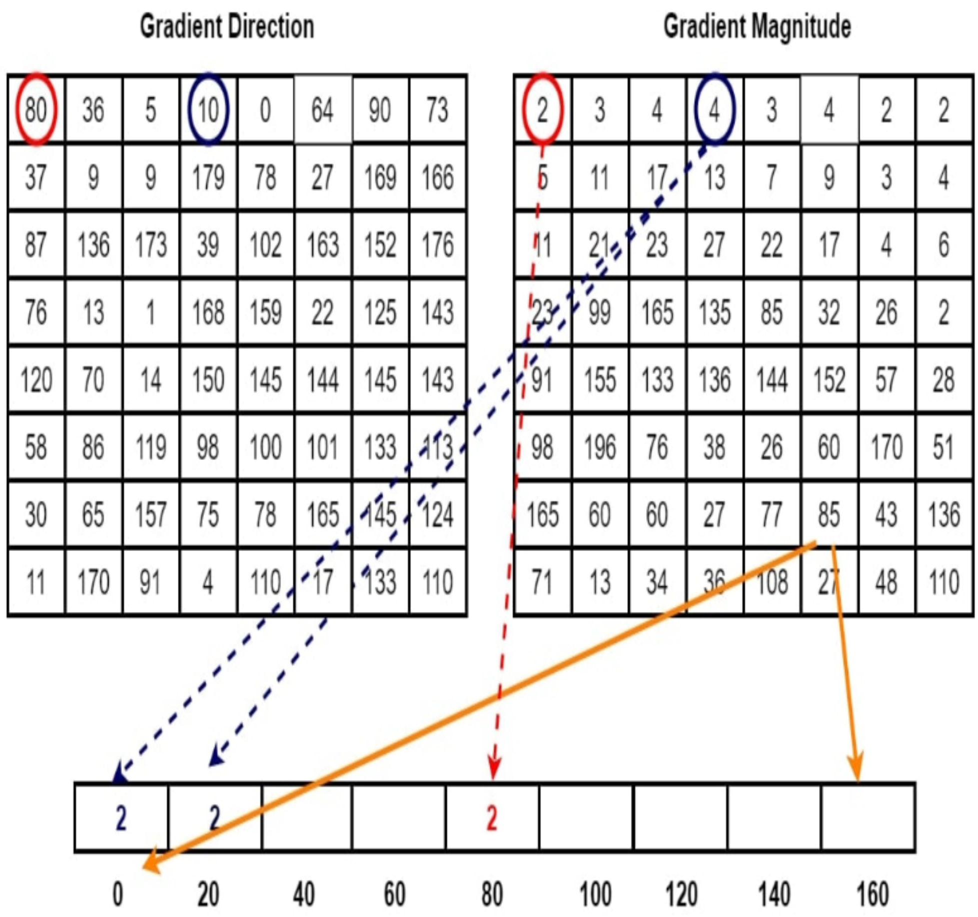



Figure 3.3: Extraction of features using HOG

In Figure 3.3 we can see that HOG is doing feature extraction using two things first one is Gradient direction and other is Gradient magnitude. The formulas for the calculation of Gradient and Magnitude are given in figure 3.4.

$$\textbf{Magnitude :} \quad \|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

$$\textbf{Direction :} \quad \theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right)$$

Figure 3.4: Calculation of HOG

Following the successful computation of Gradient Direction and Gradient Magnitude for every pixel, the extraction of features can be demonstrated as depicted in Figure 3.5.

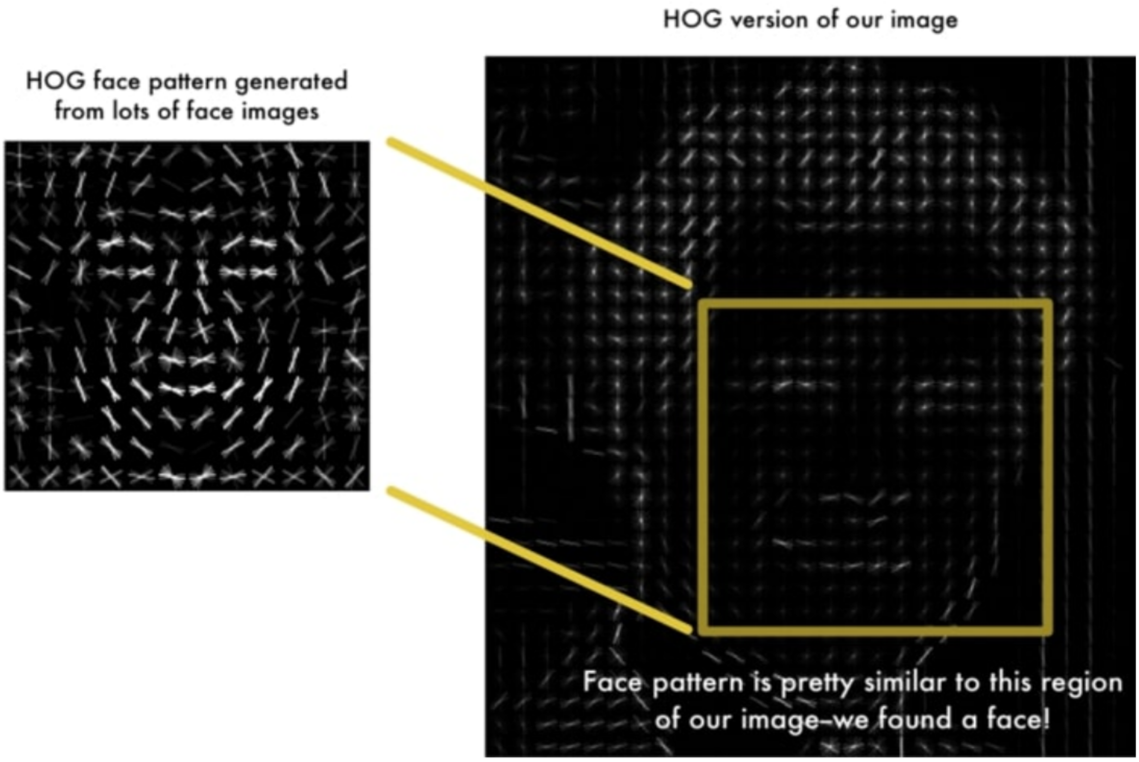



Figure 3.5: Detection of face utilizing HOG

### 3.2.3 Utilizing FLE for posing and projecting

Head movements during lectures are frequent and can influence the precision of our system. For instance, when the same individual looks in various directions, the computer perceives these as unique images despite representing the same person. This variance in head orientation can create complexities for our system, as demonstrated in Figure 3.6.

Figure 3.6: Image Showing Person Gazing in Various Directions

To tackle the challenges posed by head movements and varying face orientations, we will integrate the Face Landmark Estimation Algorithm (FLE) [12] into our system. This algorithm identifies 68 key points on the face, including the edges of both eyes, the chin's top, and the eyebrows' edges. It then measures the distances between these landmarks, such as those between the eyes, nose, and chin. This process is illustrated in Figure 3.7.

By marking these 68 landmark points on the face, we obtain a centralized image, as depicted in Figure 3.8. This centralized image will be fed into a Convolutional Neural Network (CNN) to produce a 128-dimensional encoding, which is highly effective for face recognition tasks. This strategy helps mitigate the impact of head movements and diverse face orientations on our system's accuracy.

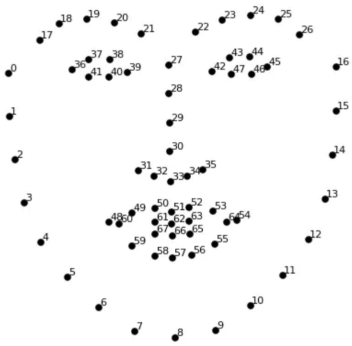


Figure 3.7: Face Landmark Points

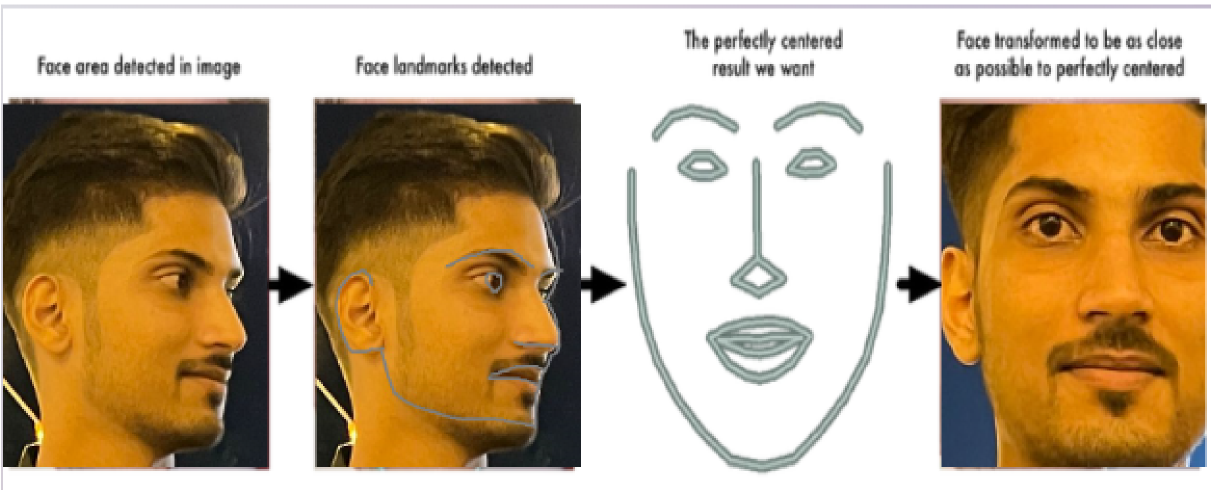



Figure 3.8: Aligning images using the Facial Landmark Estimation (FLE)

### 3.2.4 Encoding of faces Using CNN

The process of training the dataset is illustrated in Figure 3.9. Initially, an image of an individual is captured to produce a 128-dimensional encoding. Test images of other individuals are also captured, and corresponding encodings are generated. These results are then compared, leading to slight adjustments in the neural network based on these comparisons.

The Convolutional Neural Network (CNN) utilized in our system employs a Mask-based approach, a deep learning technique known for its effectiveness in tasks such as face recognition [13]-[15].

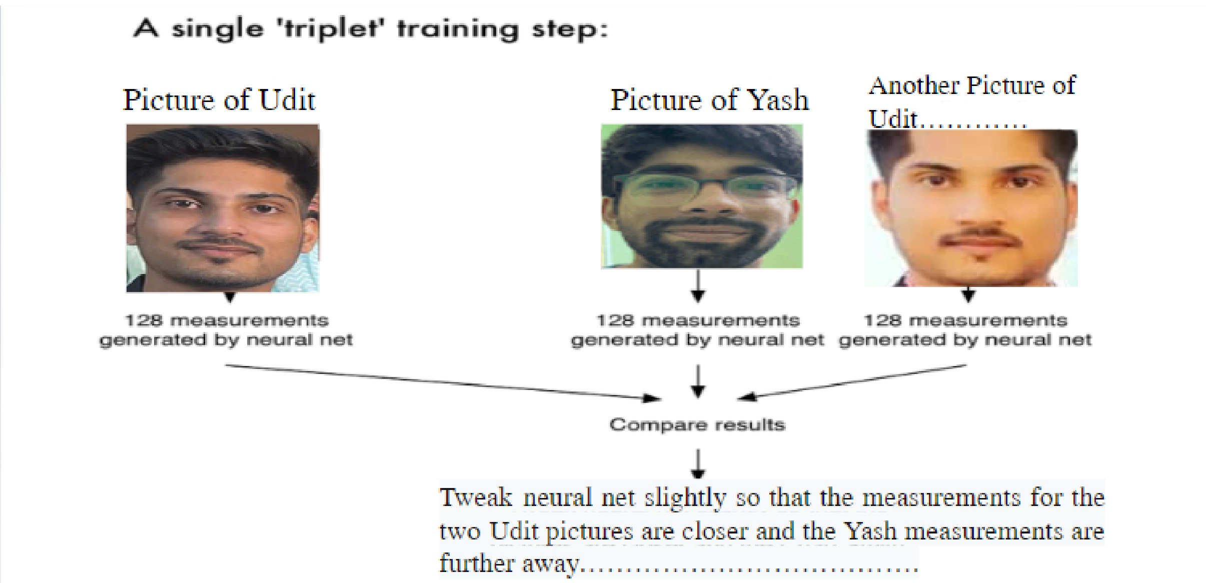



Figure 3.9: Training steps of Convolutional Neural Network

Absolutely! The Convolutional Neural Network (CNN) is structured with multiple layers of neurons. The initial layers specialize in extracting basic features such as horizontal and vertical edges. As we move deeper into the network, it becomes capable of detecting more complex features like corners, objects, and ultimately, faces.

Our system leverages a CNN model to derive essential metrics from each face. Through training, this model generates a 128-dimensional encoding for every face, as depicted in Figure 3.10. These encodings encapsulate critical facial attributes that significantly contribute to precise face recognition and verification.

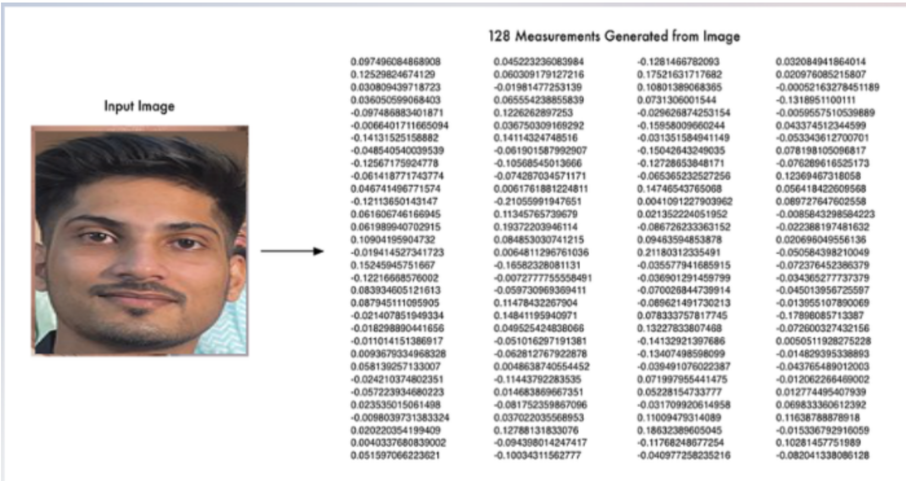



Figure 3.10: 128 Face Encoding generated by CNN

### 3.2.5 Use of SVM for comparison

Indeed, the Support Vector Machine (SVM) serves as a linear classifier in our system. Its primary function involves comparing the measurements of unrecognizable faces with those stored in our database [16]-[18]. This comparison process enables the SVM to pinpoint the closest match based on these measurements, thereby revealing the name of the individual associated with the closest match.

SVM performs its computations swiftly, often completing them within milliseconds. Its operation involves introducing a hypothetical plane between the two sets of measurements and then determining the closest match based on the distance from this plane. This methodology is instrumental in ensuring our system's ability to accurately identify and verify individuals based on their facial measurements.

### 3.2.6 Report Generation

After finishing the face recognition process, the attendance report undergoes an update. The recognized student's name is added to the report, alongside the timestamp indicating when the face recognition occurred. Importantly, the attendance of each student is updated only once, ensuring there's a single occurrence of each student in the report. This approach maintains the accuracy and integrity of the attendance records, preventing duplicate entries for individual students.

# CHAPTER 4

# IMPLEMENTATION DETAILS

## 4.1 Hardware

For optimal performance of this system, you'll require the following components:

High-Quality Camera: A camera capable of capturing clear videos, with a minimum resolution of 1600x1200 or higher.

Operating Device: A device equipped with an operating system compatible with your requirements (Windows, Linux, MacOS). It should have an operating voltage range of 19V, an input voltage between 20-40V, at least 1 USB port, an i7 11th gen or equivalent processor, minimum 8GB RAM, a clock speed of at least 2.4GHz, and storage capacity of at least 100GB.

These specifications ensure that the system can handle the processing requirements, data storage, and necessary connectivity for effective operation.

## 4.2 Software

To run this system, you'll require the following:

Integrated Development Environment (IDE): Choose an IDE like PyCharm, VS-Code, or any other that suits your preference to facilitate convenient development and coding.

Python: Version 3.0 or higher of the Python programming language is necessary for running the system.

Python Modules: Install the following Python modules/packages to enable various functionalities:

OpenCV: A computer vision library for image and video processing.

NumPy: For numerical computations and handling arrays.

CSV: For reading and writing CSV files, which may be used for data storage or processing.

Datetime: To work with date and time data, which can be useful for timestamping events.

Dlib: A library with tools for machine learning, computer vision, and image processing tasks.

OS: For interacting with the operating system, managing files and directories.

Time: To handle time-related operations and delays in the code execution. Having these components and modules in place will ensure that your system runs smoothly and effectively.

## 4.3 Implementing System

To execute the system, it is necessary to connect a camera meeting the recommended specifications to the designated computer. This computer will serve as the storage hub for all videos and conduct all necessary computations. Additionally, the computer must run one of the specified operating systems, have an integrated development environment (IDE) installed, a designated Python interpreter, and all requisite modules pre-installed.

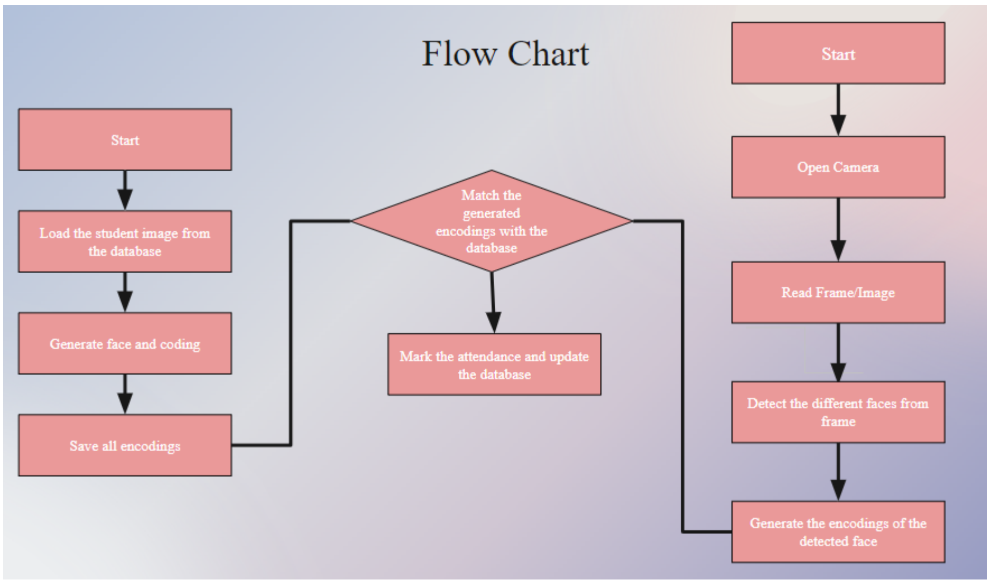



Figure 4.1: Flow Chart

Once all conditions are met, our initial step will involve developing a program to generate an Excel file for each lecture scheduled on a specific day. Each Excel file will bear the name format "subject-datecsv," ensuring the creation of a new file for the same subject daily. These files will be stored within the respective teacher's designated folder.

### 4.3.1 Detect and identifiy all people

Steps for detecting and identifying people are:

Step 1 - Preprocessing: Utilize Dlib and OpenCV libraries to preprocess the image, standardizing the input and extracting human face features.

Step 2 - Convolutional Neural Network (CNN) Construction: Develop a CNN that takes images as input and outputs 128-dimensional embeddings using the OpenFace pre-trained model.

Step 3 - Face Identification: Pass the preprocessed image through the CNN and compare its embedding with a prepared database to identify the person with the smallest embedding difference. Utilities Used:

$facerecog\_utils.py$: For image preprocessing and embedding comparison with the database.

$\boldsymbol{database\_utils.py}$ **:** For data augmentation and database generation.

$\boldsymbol{align\_.py}$ **:** For face detection and alignment using the OpenFace project.

$\boldsymbol{inception\_blocks.py}$ **:** For constructing the CNN architecture, sourced from Andrew Ng's Convolutional Neural Network course.

### 4.3.2 Face detection and alignment

To ensure optimal performance, our input images must align with the characteristics used to train the CNN model. Hence, the initial project step involves extracting all faces and applying suitable transformations to meet specific criteria:

- Image dimensions should be 96 pixels × 96 pixels.

- Each image should contain only one face.
- Faces should be aligned with eyes horizontally oriented and positioned 20 pixels from the top of the image.

Several pretrained libraries can assist in this task. I utilize the AlignDlib utility from the OpenFace project, which detects and aligns faces based on pre-trained landmarks. An alternative is the pre-trained Haar_cascade_frontalface from OpenCV, but it lacks alignment capabilities.

Data abundance enhances performance, yet acquiring it is costly. Therefore, I augment the database by adjusting original images' orientation, brightness, and contrast. This not only expands the dataset but also mitigates the impact of varying photo orientations and lighting conditions, optimizing the system's performance.

### 4.3.3 The OpenFace CNN

The OpenFace CNN, based on the FaceNet paper, operates on a unique principle. Instead of training the neural network to recognize individuals directly, it trains the network to determine whether two images belong to the same person. This is achieved through a training process that involves three input images: an anchor image ($x_a$) of the test subject, a positive image ($x_p$) depicting the same person, and a negative image ($x_n$) representing a different individual are utilized in the network. The network's goal is to minimize the distance between ($x_a$) and ($x_p$), while also maximizing the distance between ($x_a$) and ($x_n$). This objective is achieved by applying a triplet-loss function.

### 4.3.4 Face Recognition from Photo

The code executes the face recognition process through these steps:

- Feed each extracted face into the CNN, receiving their respective embedding values and calculate the embedding distance for each face by comparing it with all values in the database, identifying the person with the smallest distance.
- Assign the label of the recognized person to the corresponding face in the original image.

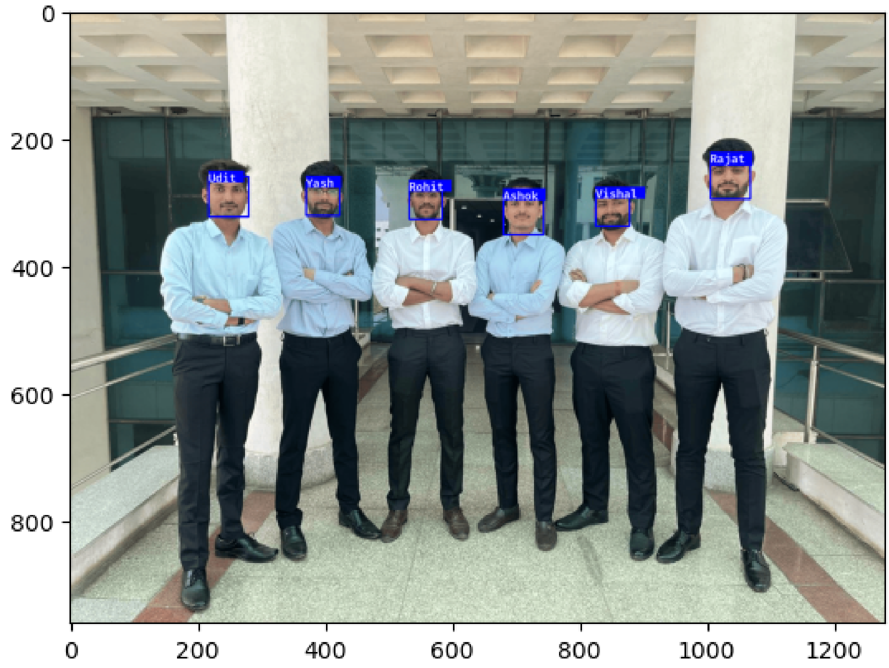



Figure 4.2: Face recognition from Photo

### 4.3.5 Face Recognition from Webcam Video

To transition from processing single images to handling webcam recordings, we incorporate code that captures and preprocesses frames from the video, treating each frame as a single image stream passed to the CNN. Here is an example of the face-recognition results taken from my webcam.

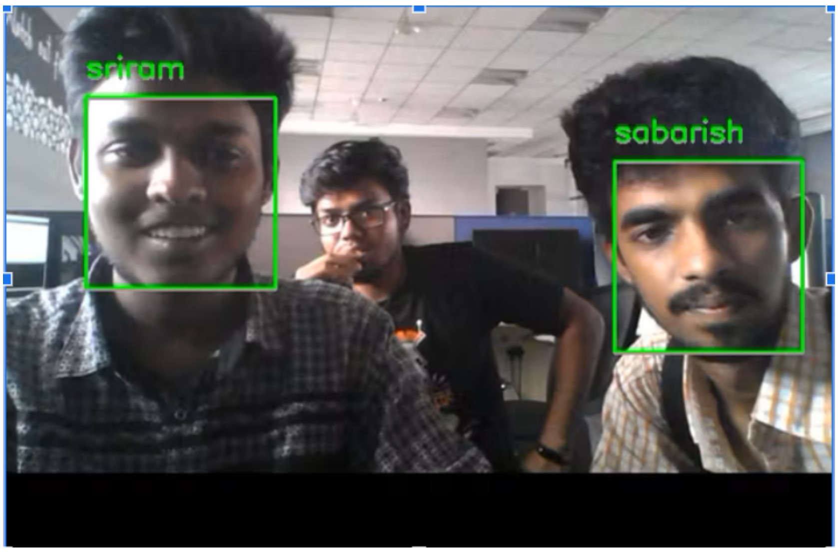



Figure 4.3: Webcam Video

# CHAPTER 5

# RESULT

The implemented solution enables the development of a software capable of handling both live and pre-recorded videos. During runtime, users can interact to label a recognized face in a frame. Subsequently, a feature vector for the labeled face is generated and stored in the trained image folder, accommodating various pose, lighting, and distance variations.

By default, the training data resides in the Trained Faces folder within the application path. It comprises a single XML file containing tags for each person's name and corresponding training image file name. However, evaluating detection accuracy against existing benchmarks proves challenging due to the absence of standardized benchmark videos.


Figure 5.1: Designed Software

# CHAPTER 6

# CONCLUSION

The proposed system has demonstrated remarkable precision and accuracy in recognizing faces and marking attendance, all while maintaining low computational complexity. Additionally, the system's economic and physical footprint is minimal. Through the integration of advanced algorithms like HOG and CNN, we achieved exceptional accuracy levels. Our study concludes that the Attendance system's success is attributed to the combination of various algorithms, showcasing commendable accuracy and efficiency.

During our exploration, we evaluated several algorithms such as Haar cascade, which could potentially replace HOG. However, HOG proved to be more efficient, especially in scenarios involving a limited number of faces, a common scenario in biometric and attendance systems. We also acknowledge the potential for enhancing algorithm performance by leveraging CUDA GPU cores and robust CPUs, typically available in server systems compared to personal computers. Despite variations in lighting conditions, our system consistently achieved high accuracy rates.

Comparing our system to others utilizing algorithms like KNN and SVM, we found our system to be superior in terms of efficiency and accuracy. Moving forward, our system can be further enhanced for various applications. For instance, incorporating multiple cameras interconnected to track individuals could be beneficial. This enhanced system could find utility in scenarios such as ATM surveillance for fraud prevention or in elections for voter identification through facial recognition technology.

# CHAPTER 7

# FUTURE WORK

Future research will focus on several aspects to enhance the current work:

- Integrating a video processing pipeline that combines various image processing modules to streamline the process.
- Developing a solution to handle a large number of images in the trained folder efficiently, ensuring scalability and robustness.
- Integrating human body detection and tracking into the system to leverage both face and body information for improved recognition and tracking capabilities.
- Implementing an enhanced multi-face tracking system capable of detecting and tracking a large number of faces simultaneously, with consideration for effective hardware requirements to support such tasks.

# REFERENCES


[1] Nyein, Thida and Aung Nway Oo. “University Classroom Attendance System Using FaceNet and Support Vector Machine.” 2019 International Conference on Advanced Information Technologies (ICAIT) (2019): 171-176.

[2] Ketan N. Mahajan and Nagaraj V. Dharwadkar, “Classroom Attendance System Using Surveillance Camera”,Int. j. innov. eng. res. technol., pp. 1–4, Mar. 2021.

[3] V. Patil, A. Narayan, V. Ausekar and A. Dinesh, "Automatic Students Attendance Marking System Using Image Processing And Machine Learning," 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 542-546, doi: 10.1109/ICOSEC49089.2020.9215305.

[4] A. Elmahmudi and H. Ugail, "Experiments on Deep Face Recognition Using Partial Faces,"2018 International Conference on Cyberworlds (CW), 2018, pp. 357-362, doi: 10.1109/CW.2018.00071.

[5] S. Dev and T. Patnaik, "Student Attendance System using Face Recognition," 2020 International Conference on Smart Electronics and Communication (ICOSEC), 2020, pp. 90-96, doi: 10.1109/ICOSEC49089.2020.9215441.

[6] Yohei Kawaguchi, Tetsuo Shoji, “Face Recognitionbased Lecture Attendance System”, 2005.

[7] Hubballi, Kirti Bhat, Vandana, "Attendance System for Face Recognition using GSM module", International Journal of Computer Sciences and Engineering, (2018), 6. 1180-1184. 10.26438/ijcse/v6i7.11801184.

[8] S. Bhattacharya, G. S. Nainala, P. Das and A. Routray, "Smart Attendance Monitoring System (SAMS): A Face Recognition Based Attendance System for Classroom Environment," 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT), 2018, pp. 358-360, doi: 10.1109/ICALT.2018.00090.

[9] E. Varadharajan, R. Dharani, S. Jeevitha, B. Kavinmathi and S. Hemalatha, "Automatic attendance management system using face detection," 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 2016, pp. 1-3, doi: 10.1109/GET.2016.7916753.

[10] J. -M. Guo, C. -C. Lin, M. -F. Wu, C. -H. Chang and H. Lee, "Complexity Reduced Face Detection Using Probability-Based Face Mask Prefiltering and PixelBased Hierarchical-Feature Adaboosting," in IEEE Signal Processing Letters, vol. 18, no. 8, pp. 447-450, Aug. 2011, doi: 10.1109/LSP.2011.2146772.

[11] S. Z. Li, R. Chu, S. Liao and L. Zhang, "Illumination Invariant Face Recognition Using Near-Infrared Images," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 4, pp. 627-639, April 2007, doi: 10.1109/TPAMI.2007.1014.

[12] Samal, Ashok and Prasana A. Iyengar. “Automatic recognition and analysis of human faces and facial expressions: a survey.” Pattern Recognition. 25 (1992): 65-77.

[13] R. Brunelli and T. Poggio, "Face recognition: features versus templates," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 10, pp. 1042-1052, Oct. 1993, doi: 10.1109/34.254061.

[14] Guodong Guo, S. Z. Li and Kapluk Chan, "Face recognition by support vector machines," Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), 2000, pp. 196-201, doi: 10.1109/AFGR.2000.840634.

[15] Puri, D., Ingle, R., Kachare, P., Awale, R., “Wavelet Packet Sub-band Based Classification of Alcoholic and Controlled State EEG Signals”, International Conference on Communication and Signal Processing (ICCASP), Atlantis Press, 2016, pp. 562-567, DOI:10.2991/iccasp- 16.2017.82.

[16] Kehri V., Puri D., Awale R.N., “Entropy-Based Facial Movements Recognition Using CPVM”, Applied Computer Vision and Image Processing. Advances in Intelligent Systems and Computing, vol 1155, 2020, Springer, Singapore. https://doi.org/10.1007/97898115402952.

[17] Puri D., Nalbalwar S., Nandgaonkar A., “EEG Based Diagnosis of Alzheimer’s Disease Using Kolmogorov Complexity”, Applied Information Processing Systems. Advances in Intelligent Systems and Computing, vol 1354. Springer, Singapore. https://doi.org/10.1007/978-981-16-2008-9-15.

[18] M. Swapna, Y. K. Sharma, and B. Prasad, “A survey on face recognition using convolutional neural network” in Data Engineering and Communication Technology. Springer, 2020, pp. 649–661.