Methodology

**Team No : **08

Project Title : Automated Face Recognition Attendance System using Deep Learning

Problem Illustration:

In the evolving landscape of education and workforce management, the demand for precise attendance tracking systems is critical. This project introduces an innovative solution—an Automated Face Recognition Attendance System powered by deep learning. Traditional attendance methods often fall short due to inaccuracies, manual efforts, and time-consuming processes. To overcome these challenges, our project leverages advanced deep learning techniques to establish a robust, real-time, and automated attendance tracking system. So, to solve this issue, we have used the technology which enables precise identification through facial recognition, allowing attendance to be effortlessly marked with a simple glance. The system excels in accurately recognizing individuals, even under varying lighting conditions and angles. Emphasizing user experience, our project incorporates an intuitive and user-friendly interface through a GUI application is developed. This ensures seamless integration with educational institutions and corporate environments. Users can easily initiate attendance marking, monitor real-time data, and access historical attendance records. The advantages of our system extend to diverse domains such as event management, security, and access control, where accurate attendance tracking is paramount. As we look ahead, the Automated Face Recognition Attendance System signifies a technological leap, eliminating the complexities associated with traditional attendance management methods. With its precision, efficiency, and adaptability, this system is poised to revolutionize attendance tracking across various industries.

Proposed Method:

The Automated Facial Attendance System using Deep Learning is done using the Python, OpenCV, CNN algorithm which is a deep learning algorithm and KNN algorithm. The Code is written in the VS Code/ PyCharm IDE. The GUI is developed using Python programming Language.

Proposed method Illustration:

The project is developed using the Python Programming language and the deep learning algorithm called CNN algorithm. We also use OpenCV which is used for capturing of the images using webcam. Mainly we use two algorithms for the development of the project those are CNN algorithm of Deep learning and KNN algorithm. Using the CNN algorithm, we are going to extract the image features of a person and by using the KNN algorithm we are going to check the image of a person with the image which is present in the backend or the image which is given at the time of registration of the student. The project is initially designed based on the two users those are admin and faculty. The admin is going to add the student and view the student. He can also view the attendance report of the students. The faculty user has an option of registration of the faculty. He is going to register initially and then he is going to login into the application. After login of the faculty, we have provided two options those are, Take attendance and reports. By clicking on the take attendance, the camera is going to open and take the snapshot of the students. By clicking on the view reports we can see the list of students who are present and absent in the form of table. The faculty can also download the attendance in the form of excel for the easy access.

Parameter Formulas:

KNN algorithm:

KNN stands for K-nearest neighbour, it's one of the Supervised learning algorithm mostly used for classification of data on the basis how it's neighbour are classified. KNN stores all available cases and classifies new cases based on a similarity measure. K in KNN is a parameter that refers to the number of the nearest neighbours to include in the majority voting process.

$d=\sqrt{((x2-x1)^2 + (y2-y1)^2)}$ to find the distance between any two points.

CNN Algorithm:

Convolutional Neural Networks (CNNs) are a class of deep learning algorithms designed for image recognition. They use convolutional layers to extract hierarchical features from input images, pooling layers to reduce spatial dimensions, and fully connected layers for classification. CNNs leverage filters to detect patterns, enabling them to learn and recognize complex visual representations in a hierarchical manner.

Size of Feature Map= (size of input map – size of kernel) + 1

Size of Feature Map if Stride is present = ((size of input map – size of kernel)/stride) + 1

Size of Feature Map if Stride and Padding is present =

((size of input Map – size of kernel + 2*padding)/stride) + 1

We will be also using the max pooling and average pooling (if required) to reduce the dimension of the input by keeping the same information present in output.

Signature Supervisor