**National University of Sciences and Technology**

**School of Electrical Engineering and Computer Science**



**Digital Attendance System**

**Advanced Topics in Computing**

**Semester Project Report by**

Waleed Ahmed Shahid 283290

Hafiz Haseeb Ahmad Butt 313193

Sana Ullah 294942

**Instructor:**

Mam Momina Moetesm

**Bachelor of Science in Computer Science (2019-2023)**

Department of Computing

School of Electrical Engineering and Computer Science

National University of Sciences & Technology

Contents

[Problem Identification: 2](#_Toc135691575)

[Overview of the problem and potential application area: 2](#_Toc135691576)

[Overview of the proposed solution: 2](#_Toc135691577)

[Methodology: 2](#_Toc135691578)

[Workflow diagram: 2](#_Toc135691579)

[Attendance Module: 2](#_Toc135691580)

[Counting Students Module: 2](#_Toc135691581)

[Brief details about all steps included: 3](#_Toc135691582)

[Dataset and preprocessing: 3](#_Toc135691583)

[Deep learning models: 3](#_Toc135691584)

[Training and Hyperparameter Tuning: 4](#_Toc135691585)

[Training Process: 4](#_Toc135691586)

[Hyperparameter Tuning: 4](#_Toc135691587)

[Model Testing and Evaluation: 4](#_Toc135691588)

[Real-Time Data Testing: 4](#_Toc135691589)

[GUI design: 4](#_Toc135691590)

[Implementation: 4](#_Toc135691591)

[Implementation details: 4](#_Toc135691592)

[Deployment: 5](#_Toc135691593)

[Use-case visualization: 5](#_Toc135691594)

[Computation of inference time: 7](#_Toc135691595)

[Annexure Code: 8](#_Toc135691596)

[Attendance Taken Code 8](#_Toc135691597)

[Counting Students Code: 11](#_Toc135691598)

# Problem Identification:

## Overview of the problem and potential application area:

The problem identified is the manual and time-consuming process of taking attendance in educational institutions. This project aims to automate the attendance process using computer vision techniques. The potential application area is in schools, colleges, and other educational institutions where efficient and accurate attendance management is required.

## Overview of the proposed solution:

The proposed solution is to utilize computer vision techniques to perform face recognition for attendance tracking. By employing deep learning models and a user-friendly graphical user interface (GUI), the system will identify and record the attendance of students in real-time.

# Methodology:

## Workflow diagram:

### Attendance Module:

The workflow diagram of recording attendance module is as follows:

A diagram of a face detection

Description automatically generated with medium confidence

### Counting Students Module:

The workflow diagram of counting students is as follows:

A picture containing text, screenshot, diagram, font

Description automatically generated

As we can see in the above diagram, the attendance taken will automatically get initiated again if the value is less than the criteria value.

# Brief details about all steps included:

## Dataset and preprocessing:

The system employs a dataset consisting of student images for training the face recognition model. The dataset uses preprocessing techniques such as image resizing, normalization, and noise removal. First the rgb frame from the video is being transformed into greyscale, then filters like gaussian and other have been used to remove noise and blurness from the frame and for making the face clear.

## Deep learning models:

The project utilizes the face recognition library, specifically the ResNet model, for face recognition tasks. The ResNet model is chosen for its exceptional performance in extracting facial features and generating face encodings.

#### Justification for using ResNet model:

**High accuracy:** ResNet has demonstrated state-of-the-art performance in face recognition tasks due to its deep architecture and skip connections, which alleviate the degradation problem.

**Robust feature extraction:** ResNet's deep layers enable the extraction of rich and discriminative features from facial images, improving the model's ability to distinguish between different individuals.

**Pretrained weights:** The availability of pretrained ResNet models on large-scale datasets (e.g., ImageNet) allows for transfer learning, where the model can be fine-tuned on a smaller dataset specific to face recognition.

**Efficient computation:** Despite its depth, ResNet has a relatively low computational cost compared to other deep architectures, making it suitable for real-time face recognition applications.

## Training and Hyperparameter Tuning:

In the digital attendance system using computer vision, the face recognition model employs a pretrained ResNet model for encoding facial features. As a pretrained model, it has already undergone extensive training on large-scale datasets like ImageNet. However, fine-tuning the model is necessary to adapt it to the specific task of face recognition in the attendance system.

### Training Process:

To train the face recognition model, the pretrained ResNet model is loaded and then fine-tuned using a labeled dataset of student images. The dataset consists of samples representing each student's face, with corresponding labels indicating their identities.

### Hyperparameter Tuning:

Hyperparameters play a crucial role in optimizing the model's performance. These parameters define the configuration of the model and the training process, such as learning rate, batch size, number of layers, and optimizer. Fine-tuning the pretrained model may require adjusting these hyperparameters to achieve the best results. We have tried to fine tune the model by updating the hyperparameters but results were getting worsed, so we ended up going with the default parameters.

## Model Testing and Evaluation:

The trained face recognition model is subjected to rigorous testing using real-time data to evaluate its accuracy and performance. Testing the model on real-time data is crucial to assess its robustness and reliability in a practical setting.

### Real-Time Data Testing:

To evaluate the model's performance, a separate set of real-time student images is collected, which represents scenarios that the model will encounter during actual attendance tracking. This dataset includes variations in lighting conditions, facial expressions, poses, and occlusions.

During testing, the model is presented with these real-time student images, and it predicts the identities of the individuals. The predictions are then compared with the ground truth labels to measure the accuracy and performance of the model. The predictions are done accurately.

## GUI design:

A user-friendly GUI is designed to facilitate easy interaction with the system. The GUI should allow video real-time inference, and display of attendance records.

# Implementation:

## Implementation details:

The project utilizes the face recognition library, which integrates deep learning frameworks for model training and face recognition tasks. Specify which deep learning frameworks are being employed (e.g., TensorFlow, PyTorch). Additionally, mention the libraries used for GUI implementation and any other relevant libraries employed in the project.

The other libraries used are as follows:

Table : Libraries Used

|  |
| --- |
| import face\_recognition  import cv2  import numpy as np  import os  import xlwt  from xlwt import Workbook  from datetime import date  import xlrd, xlwt  from xlutils.copy import copy as xl\_copy |

# Deployment:

## Use-case visualization:

The whole use-case visualization is shown below:

Starting with the attendance system in which face detection and recognition is being done:

A person with a beard

Description automatically generated with low confidence

As, my face have been detected and recognized, then it will be added to the excel sheet and to the GUI.

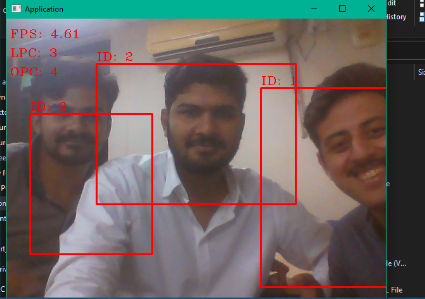
A computer screen shot of a brick wall

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

Now, moving onto the counting students module, this is how faces are detected and counted.



## Computation of inference time:

The time it takes in face detection and recognition is 1 sec and for counting its 1.5 second.

# Annexure Code:

## Attendance Taken Code

A screen shot of a computer program

Description automatically generated with low confidence

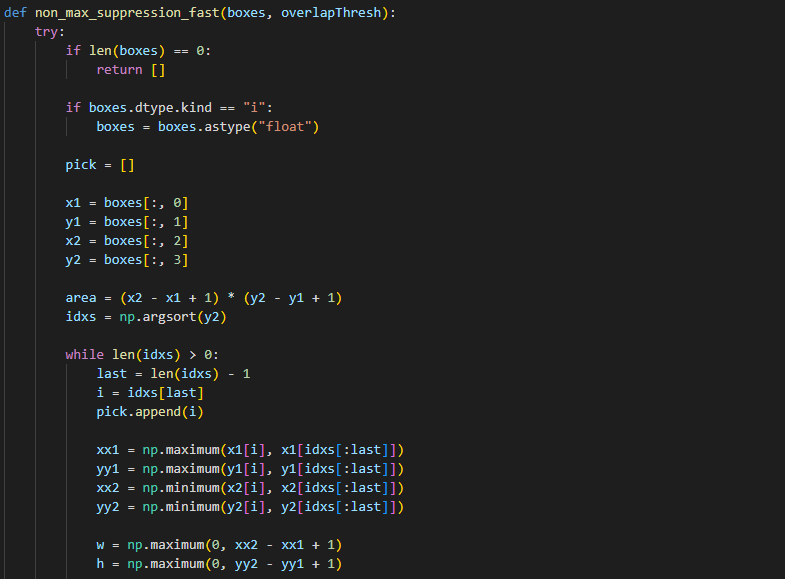
A picture containing text, screenshot, software, display

Description automatically generated

A picture containing text, screenshot, software, font

Description automatically generated

## Counting Students Code:



A picture containing text, screenshot, software

Description automatically generated