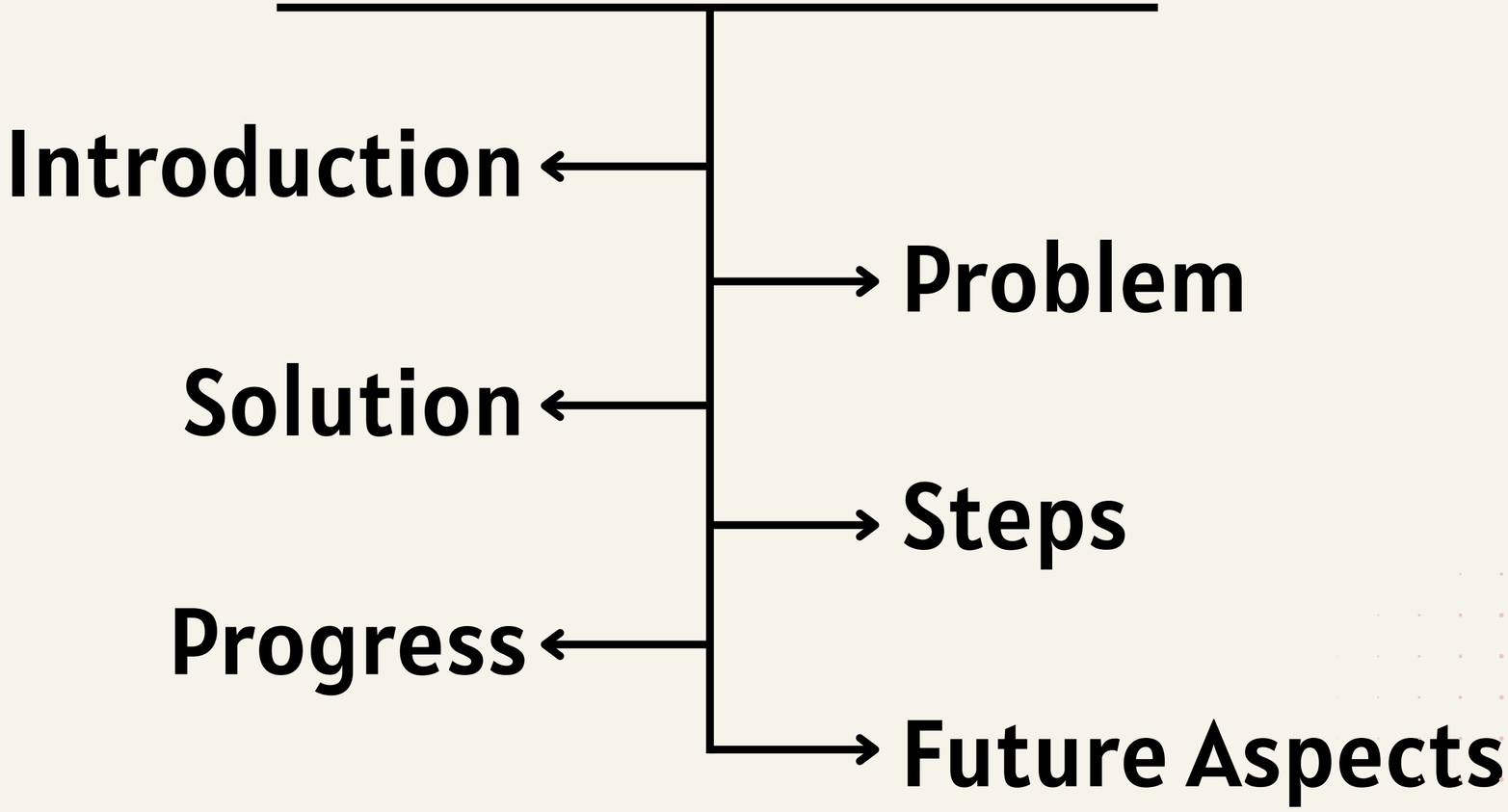
Innovation Laboratory
AE3920I

ATTENUE

Smart Attendence System

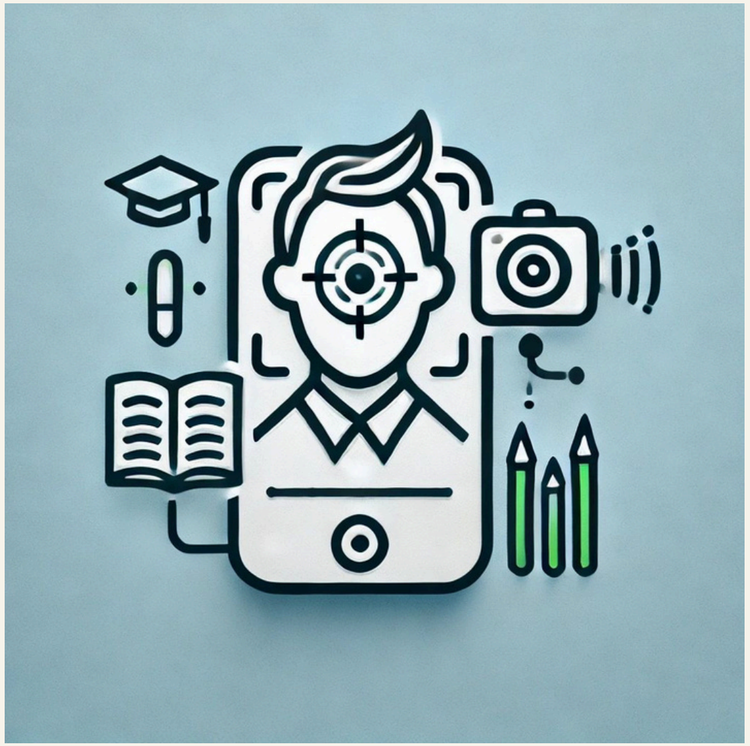
Intermidiate Evaluation

OVERVIEW


Introduction
Problem
Solution
Steps
Progress
Future Aspects

INTRODUCTION

The Smart Attendance System using Computer Vision is an automated solution designed to streamline the process of marking attendance in classrooms. By leveraging face recognition technology, the system enables professors to capture a photo of the class, which is then processed to identify students and automatically record their attendance. This eliminates the need for manual roll calls, saves time, and enhances accuracy, providing a modern, tech-driven approach to classroom management.



PROBLEM

First Problem

Manual Attendance is Time-Consuming and Prone to Errors:

- In large classrooms, taking attendance manually consumes valuable lecture time and often results in errors, such as missed entries or incorrect marking.

Second Problem

Lack of a Reliable System to Prevent Proxy Attendance:

- Traditional attendance methods allow for the possibility of proxy attendance, where students can mark attendance for absent peers, leading to inaccurate records.

SOLUTIONS

1. Automated Attendance through Face Recognition: By using computer vision to automatically identify and record the attendance of students from a classroom photo, the system reduces the time spent on manual attendance.

2. Real-Time Processing and Instant Record Generation: The system processes the class photo in real-time, generating attendance records almost instantly, preventing delays and minimizing disruption during lectures.

3. Elimination of Proxy Attendance: With facial recognition technology, each student is uniquely identified based on their facial features, ensuring that only present students are marked, thereby preventing proxy attendance.

STEPS

- MAKING OF COMPUTER VISION MODEL
- MAKING OF APPLICATION
- INTEGRATE THE MODEL WITH THE APPLICATION
- IoT

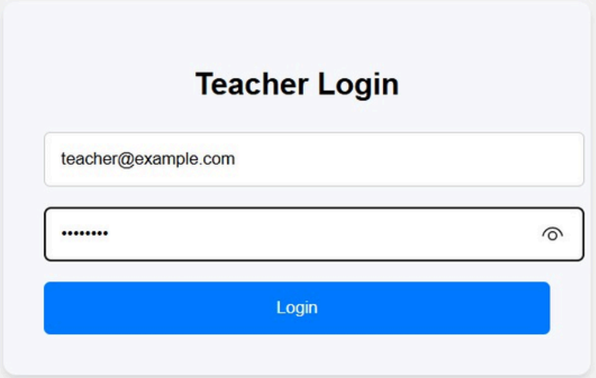
PROGRESS - MODEL

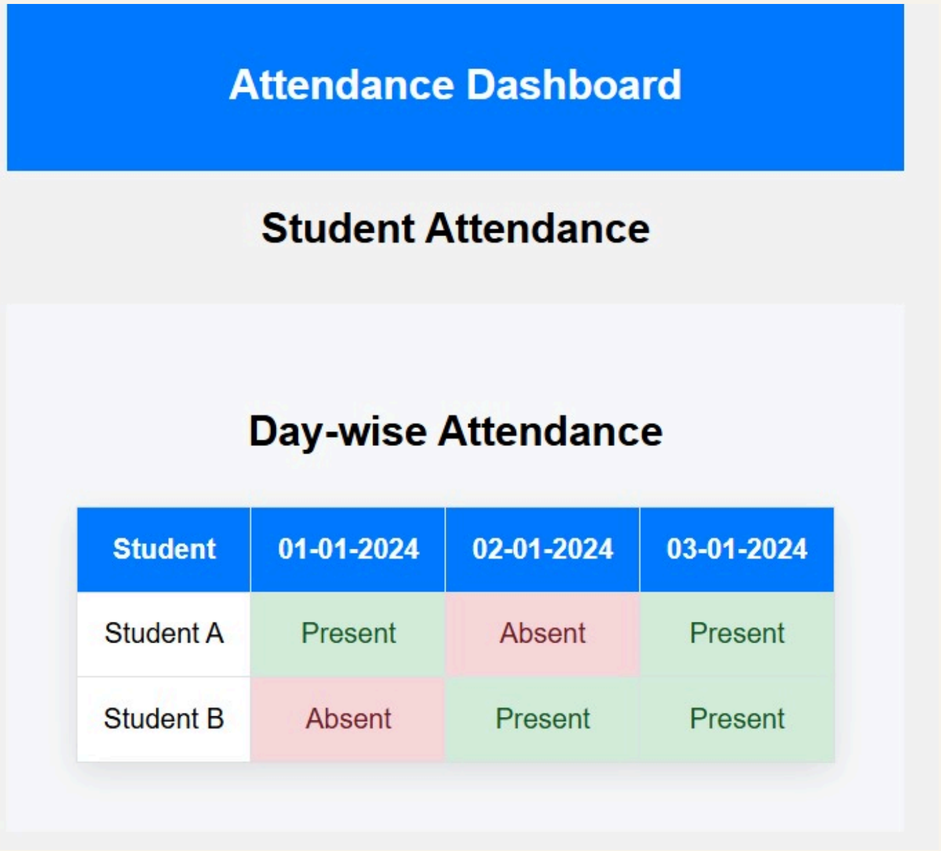
We have used computer vision to make our model. Our model takes images of the students and recognizes them in the image provided by the professor.

```python
import cv2
import numpy as np
import face_recognition
import os

path = '/Users/ishankanodia/Desktop/SEM 5/Innovation Lab/fold'    # Pa
images = []
classNames = []
myList = os.listdir(path)
print(myList)

# Load images and class names
for cl in myList:
    curImg = cv2.imread(f'{path}/{cl}')
    if curImg is None:
        continue
    images.append(curImg)
    classNames.append(os.path.splitext(cl)[0])
print(classNames)

# Function to find encodings of faces in the images
def findEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)
        if encode:  # Ensure there is at least one encoding
            encodeList.append(encode[0])
    return encodeList
```

```python
# Load the image to process (replace the webcam feed with the PNG image)
img_path = '/Users/ishankanodia/Downloads/img.png'  # Use the uploaded image path
img = cv2.imread(img_path)

if img is None:
    print("Error loading image.")
else:
    # Convert color for face recognition processing
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    # Find face locations and encodings in the current image
    facesCurFrame = face_recognition.face_locations(imgRGB)
    encodesCurFrame = face_recognition.face_encodings(imgRGB, facesCurFrame)

    # Compare detected faces with known faces
    for encodeFace, faceLoc in zip(encodesCurFrame, facesCurFrame):
        matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
        faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
        matchIndex = np.argmin(faceDis)

        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
        else:
            name = "UNKNOWN"

        # Add recognized name to the list
        recognized_names.append(name)

        # Draw rectangle around the face
        y1, x2, y2, x1 = faceLoc
        cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
        cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
        cv2.putText(img, name, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
```

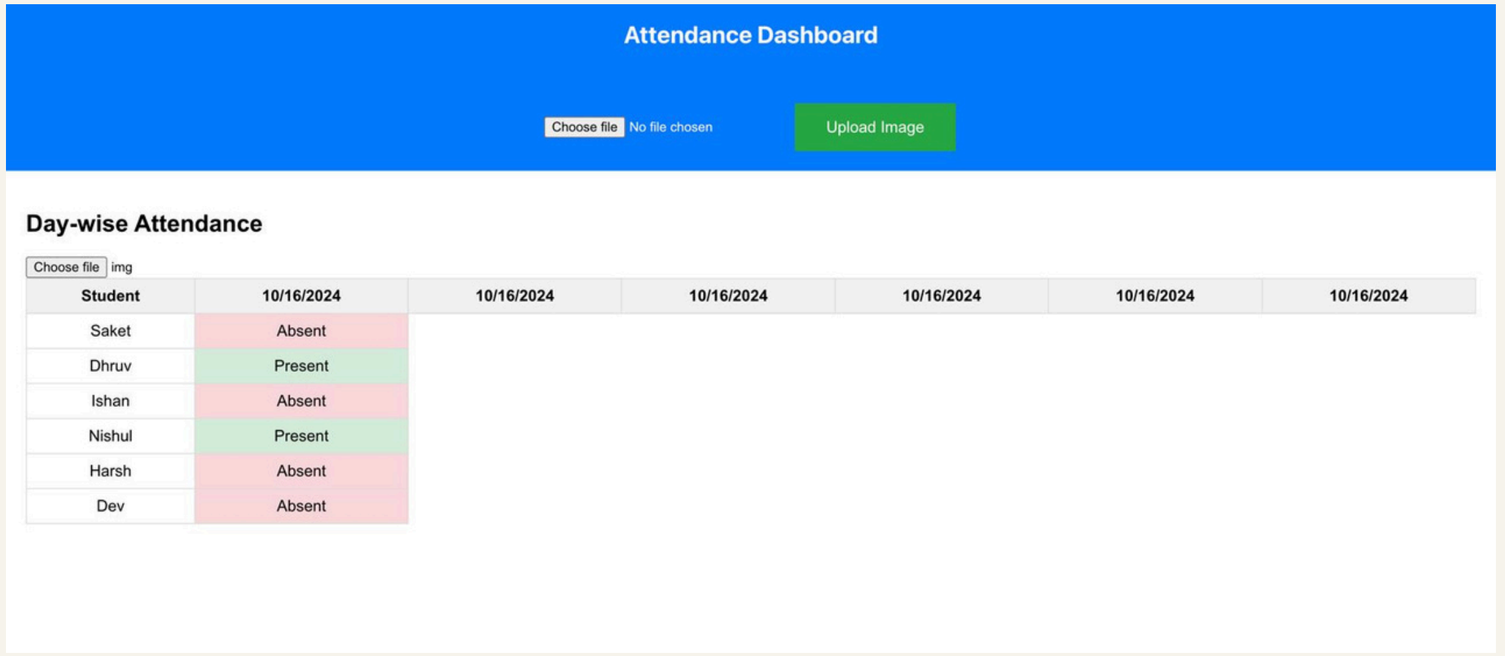
PROGRESS – APPLICATION





This is the application we will be using for the attendance

PROGRESS - INTEGRATION



Student	10/16/2024	10/16/2024	10/16/2024	10/16/2024	10/16/2024	10/16/2024
Saket	Absent					
Dhruv	Present					
Ishan	Absent					
Nishul	Present					
Harsh	Absent					
Dev	Absent					

We have integrated our model which takes input image and marks the attendance.

FUTURE ASPECTS(IOT)

In the future, the system can be enhanced by integrating hardware components like an Arduino camera and Bluetooth module to further automate and streamline the process. With an Arduino camera mounted in the classroom, attendance photos can be captured automatically at specific intervals, removing the need for manual picture-taking. Additionally, integrating a Bluetooth module can allow students' devices to communicate with the system, confirming their presence in the classroom. This combination of hardware and software automation would lead to a fully hands-free, efficient, and tech-driven attendance system, ensuring real-time tracking with minimal intervention.

THANK YOU

Attenue | IIT Kharagpur