BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI

FACIAL DETECTION FOR ATTENDANCE MONITORING USING A PTZ CAMERA

Submitted by:

Somya Kansal	2020B3A30944P
Akhil Goel	2021A8PS2548P
Vastal Agarwal	2021A3PS1074P
Shreya Kalantri	2021A3PS2049P
Vaibhav Gandotra	2020B3A30812P

Submitted to:

Mr. Pawan K Ajmera

Department of Electrical and Electronics Engineering

BITS Pilani

Acknowledgements

The success of this report would not have been possible without the assistance of our instructor and mentor, Mr Pawan K Ajmera, for allowing us to do this project under him and for guiding us throughout the semester to make the project comprehensive through his insights and knowledge. We would like to express deep appreciation to the students who provided their images, allowing us to create a comprehensive dataset essential for our facial recognition system. Their participation and cooperation have been vital in assuring the diversity of the dataset, which is paramount to the project's success. Lastly, we would also like to thank our family members and close friends for their moral support. Together, they have helped make this project a reality. We are eternally grateful for their constant support.

Abstract

The identification of a person's face is increasingly relying on promising methods such as face recognition, fingerprint recognition, and other biometric authentication approaches. This approach can also be applied to the classroom.

Handling attendance for a large number of students in a big classroom takes a lot of work to do manually. The usual ways of taking attendance have some problems and could be more reliable. This study suggests a new system that uses face detection and recognition to automate attendance management. The goal is to simplify the process, save time, and be more efficient. The system uses a face recognition algorithm on real-time video feed using a PTZ Camera to detect and mark attendance during a class. It then generates a file with the attendance information at the end of the lecture.

This automated system helps reduce the workload for professors. The research involves collecting data, detecting and cropping faces, pre-training, creating embeddings, recognising faces, and marking attendance.

Introduction

In the constantly changing field of educational technology, finding effective attendance management solutions continues to be a significant difficulty. In previous studies, we have accomplished this by using several cameras to record student attendance, which calls for intricate setups and frequently raises operating expenses. This project seeks to improve attendance tracking by using a single Pan-Tilt-Zoom (PTZ) camera and a single image of the student in the dataset, thus reducing hardware requirements, problems with data collection, and manual labor while preserving accuracy. This strategy is significant because it can simplify administrative procedures, lighten the load on resources, and support the development of a more economical and sustainable learning environment.

Our research uses the Insightface library, a potent tool based on Convolutional Neural Network (CNN) technology, for facial recognition model training in order to achieve this goal. We explore the complexities of deep learning with CNN in order to improve the model's accuracy and resilience. The capacity of the facial recognition algorithm to identify people from a single image in the dataset is a significant breakthrough of our study. This feature greatly lessens the system's reliance on large image databases, increasing its adaptability to a variety of classroom situations.

We will give a thorough explanation of the Insightface-based model in the parts that follow, explaining its design, training process, and PTZ camera system integration. We will also carry out a comparative analysis, investigating alternative face recognition models that are presently being used in the industry. The objective of this comparative analysis is to shed light on the advantages and disadvantages of various models in order to advance our understanding of the state of technology as it relates to facial recognition for attendance monitoring.

This research seeks to contribute to the continuing discussion on utilizing technology for effective educational processes as we explore the details of our proposed system, which includes the integration of a single PTZ camera with sophisticated facial recognition models. We hope that this investigation will lead to a future in which attendance control will easily adapt to changing educational requirements, improving sustainability and efficacy at the same time.

Literature review

[5]Y. Su, in his paper, presented a strategy to improve scene analysis performance by using a distributed optimization framework for dynamic camera networks and an Extended Kalman-Consensus filtering algorithm. While the distributed control architecture facilitates cooperative decision-making for ideal camera settings, the application of consensus filtering enhances the accuracy of target tracking in dynamic contexts.

[6] Aditya has performed face detection under a heterogeneous Database Based on Fusion of Catadioptric and PTZ Vision Sensors. The system automatically adjusted the PTZ camera parameters using a joint calibration technique to track and detect the face ROI at a higher resolution and project it in facespace for Eigenface algorithm detection.

In his study [8] Unsang, has developed an entirely novel Coaxial-Concentric camera system that, for face identification, can track and record high-resolution face images at any distance between 6 and 12 meters (with an interpupillary distance of roughly 100 pixels). A wide working distance is offered by the Coaxial-Concentric camera setup, allowing for highly accurate tracking and recognition of moving subjects. For reliable tracking, they have unveiled a pan and tilt motion velocity control technique as well as a linear prediction model.

[4] In ‘An End-to-End Real-Time Face Identification and Attendance System using Convolutional Neural Networks’, the paper talks about the implementation of facial recognition for attendance management in lectures. The MTCNN model recognizes the faces using the FaceNet Module. The paper talks about the various experimental results they got by altering the environmental conditions and testing the effectiveness of the face recognition model. They use images with and without padding and compare accuracy. They train their model with over 18000 images of 18 students and talk about accuracy by adding noise and increasing brightness and hue. The experiment gives positive results with an average accuracy of the model up to 96%. We can also see the major role of augmentation which increases the accuracy of the model by more than double (from 38% to 96%)

[9] In ‘Attendance-Based Systems for Face Recognition Using a Camera Rail Approach’, the research presents a practical implementation of a sliding camera system for efficient face recognition-based attendance, addressing issues related to camera coverage and blind spots in traditional setups. It also talks about automated camera movement using face detection algorithm in tandem and using multiple cameras for increasing field-of-view (FOV)

Related work in face recognition

Introduction: Facial recognition technology has seen significant advancements in recent years, enabling various applications, from identity verification to surveillance systems. FaceNet, and MagFace are some popular facial recognition models that have gained recognition for their accuracy and efficiency.

FaceNet: FaceNet, introduced by Google in 2015, is a deep convolutional neural network (CNN) model designed for face recognition. It learns a mapping from the face image space to a high-dimensional feature space, where the similarity between faces can be measured by Euclidean distance. FaceNet's essential contribution is developing a triplet loss function, which ensures that the distance between an anchor face and a positive face (same person) is smaller than between the anchor face and a negative face (different person).

MagFace: MagFace, proposed in 2020, is a face recognition model that aims to address some of the limitations of existing models like discriminative feature extraction. It incorporates a margin-based softmax loss function that enhances inter-class separability, making it more robust to illumination,pose, and occlusion variations. MagFace also introduces a module called "magnet,"which applies adaptive margin values based on the features of the training data, enabling better generalization to unseen faces.

Methodology

I. Data Collection

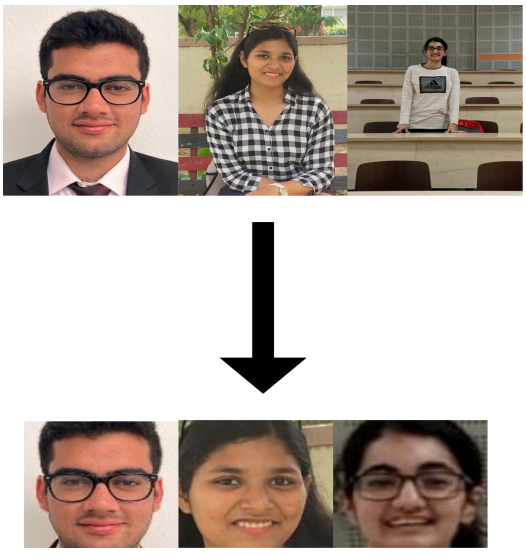
Data collection is the foundation of any successful machine learning venture. The initial and essential step in the process was to create an extensive data set for training our model. This dataset serves as the foundation for developing and expanding our model's intelligence.

To create this dataset, we started photographing students in a particular class. We photographed each student, encompassing different angles and facial accessories like spectacles. Each image was effectively documented and stored in a tagged folder system, with the students' names and identification numbers.

Our extensive image collection, with different characteristics, angles, and expressions, serves as the raw material for our model's training. This is the bedrock of our machine learning project, allowing our model to recognize and learn from the intricate details offered by each student's appearance.

We've built the framework for our model to gain information, recognize patterns, and accurately identify and classify students based on their facial features and characteristics.

II. Data Pre-Processing and Preparation:



All images across every dataset were scaled down to 1 megapixel for vectorization of the image pixel values.

The **prepare_data.py** file in the Keras InsightFace library is a script that performs various data preparation steps for training a face recognition model. First, it imports the necessary libraries and modules, such as TensorFlow, NumPy, and argparse. These libraries provide functionalities for working with deep learning models and handling command-line arguments.

Then, the script defines several helper functions:

a. parse_arguments(): This function parses the command-line arguments passed to the script, such as the input directory containing face images and the output directory to store the processed data.

b. load_image(): This function loads an image file and returns the image as a NumPy array derived from the BGR (Blue Green Red) pixel vectorisation technique. It uses the OpenCV library to read and process the image.

c. resize_image(): This function resizes an image to a specified size by taking an input image and the target size as arguments and using the OpenCV library to perform the resizing operation.

d. process_data(): This function performs the central data processing steps. It takes the input and output directories as arguments. Within this function, the script iterates through the images in the input directory and performs the following steps:

• Loads each image using the load_image() function.

• Resize the image to a desired size using the resize_image() function.

• Save the resized image to the output directory with the same filename.

• Keep track of the processed images and their corresponding labels (i.e., the person's identity in the picture) in separate lists.

e. save_data(): This function saves the processed image filenames and labels into a CSV file in the output directory. The CSV (Comma-Separated Values) file serves as a mapping between the image filenames and their associated tags.

```python
def resize_dataset(dataset_dir, target_shape=224):
    from tqdm import tqdm
    from glob2 import glob
    import cv2

    target_dataset_dir = dataset_dir.replace("112", str(target_shape))
    aa = glob(os.path.join(dataset_dir, "*/*"))
    for ii in tqdm(aa):
        target_imm = ii.replace(dataset_dir, target_dataset_dir)
        target_dir = os.path.dirname(target_imm)
        if not os.path.exists(target_dir):
            os.makedirs(target_dir)
        imm = cv2.imread(ii)
        cv2.imwrite(target_imm, cv2.resize(imm, (target_shape, target_shape), interpolation=cv2.INTER_CUBIC))
```

The script defines the program's main entry point of the program and checks if it is being run as the main module, and executes the program's main entry point. The prepare_data.py script loads face images from an input directory, resizes them to a specified size, saves the processed images to an output directory, and creates a CSV file that maps the image filenames to their corresponding labels. This processed data can then be used for training a face recognition model.

III. Face Detection (Bounding Boxes)

Now that we have collected and pre-processed the data, we need to make sure that the image itself and the size of the same is a good fit for the model. Essentially make sure that the images we feed to the model consist mainly of the person's face without including any external or background information that may possibly confuse the model.

To enable this, the model that we chose has two options available:

- YoloV5FaceDetector
- SCRFD

SCRFD (Single-Shot Scale-variant Face Detector) is a face detection model. The term "SCRFD Onnx" refers to the SCRFD model being converted to the ONNX (Open Neural Network Exchange) format, which is a standard for representing deep learning models. The ONNX format allows models to be interoperable across different frameworks and tools. The conversion of the SCRFD model to ONNX enables it to be used with platforms and libraries that support the ONNX format, providing flexibility for deployment and inference.

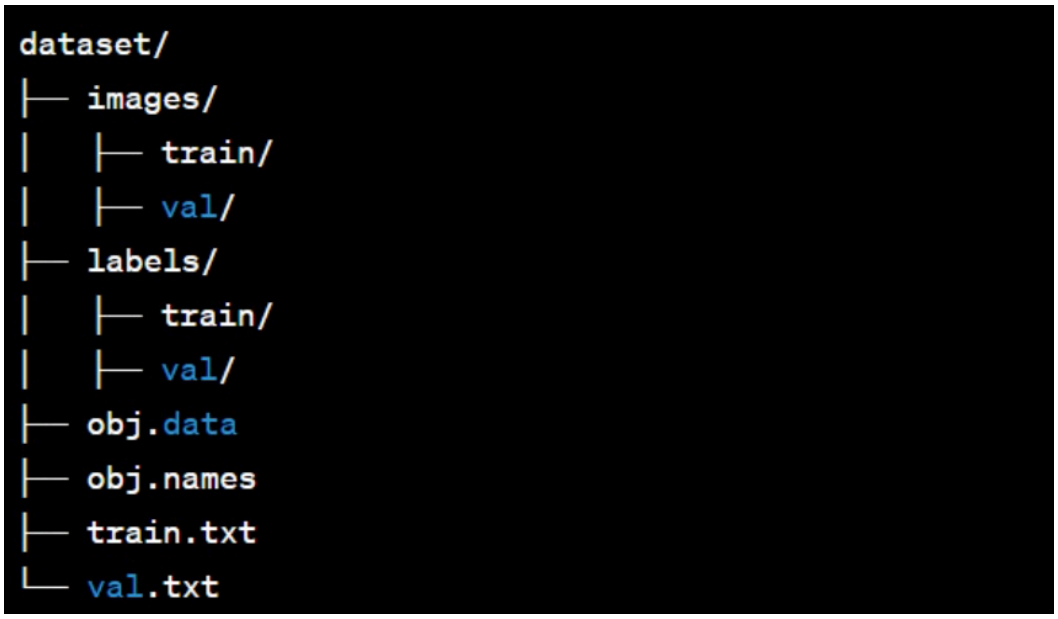
YOLOv5 face detector is a real-time, high accuracy face detection model based on the YOLOv5 object detector. It is designed to detect faces in complex scenes with high accuracy and speed. The model is optimized for mean average precision (mAP) and speed, and it can detect faces of different sizes, including small faces. The model is available on GitHub and can be installed using pip. The YOLOv5 face detector has been used to improve the detection accuracy of small faces in surveillance images. The model has achieved state-of-the-art performance in almost all the Easy, Medium, and Hard subsets of the WiderFace dataset

We used the YoloV5FaceDetector as it had shown better results then SCFRD in tests.
It was also the default used method in the model and required minimal new setup process to integrate.
Although YoloV5 is primarily designed for object detection tasks rather than face detection specifically. However, it can be adapted for face detection by training it on face-specific datasets. The data preparation process for training YOLOv5 involves annotating the dataset for object detection tasks.

1. Annotation: Annotate the dataset by creating bounding box annotations around each face in the images. Various annotation tools, such as LabelIng, RectLabel, or RectLabel Lite, allows you to draw bounding boxes and assign class labels to them.

2. Label Format: YOLOv5 requires the annotation data to be in a specific format. Each annotation should be represented as a text file with the same name as the corresponding image file but with a .txt extension. Each line in the annotation file should contain the class index and the normalized coordinates of the bounding box (center x, center y, width, height), all separated by spaces. The coordinates are relative to the width and height of the image.

3. Directory Structure: Organize the dataset into a specific directory structure required by YOLOv5. The recommended system is as follows:



• The dataset/images/train/ directory contains training images.

• The dataset/images/val/ directory contains validation images.

• The dataset/labels/train/ directory contains annotation files for training images.

• The dataset/labels/val/ directory contains annotation files for validation images.

• The obj.data file contains configuration information, such as the number of classes and paths to training and validation datasets.

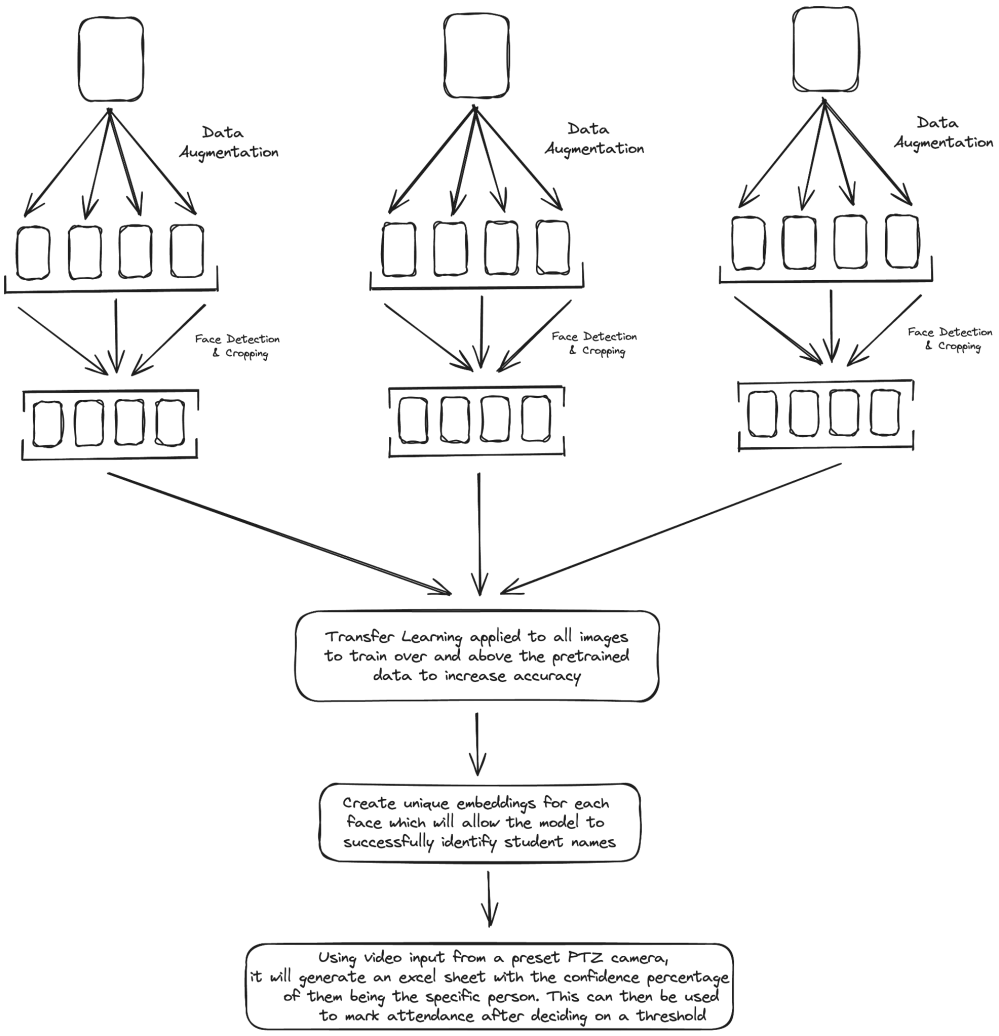
• The obj.names file contains the class names, with each class name on a separate line.

• The train.txt file contains the paths to the training images.

• The val.txt file contains the paths to the validation images.

4. Configuration: Modify the YOLOv5 configuration files (yolov5s.yaml) to suit the specific dataset and training requirements. Parameters such as the number of classes, anchors, image size, and training hyperparameters are adjusted. The script will iterate through the images and annotations, optimizing the model's parameters to minimize the detection error.

IV. Flowchart



V. Siamese Network Embeddings

Siamese networks are a type of artificial neural network that use the same weights while working in tandem on two different input vectors to compute comparable output vectors. They consist of two or more identical subnetworks, each processing a different input sample with the same weights. The outputs from these sub-networks are then compared in the final layer in order to generate a similarity score. Siamese networks are often used for tasks such as image recognition, object tracking, signature verification, and biometric authentication systems. They are particularly helpful for situations with little data as they can learn from very little data and are more robust to class imbalance. Siamese networks can be trained to see if two images are the same, which enables them to classify new classes of data without training the network again.

Siamese networks differ from other neural network architectures in that they are specifically designed for comparing two instances and inferring if they belong to the same object. They are composed of two parallel identical neural networks, whose output is a vector of features. This vector of features is then used to infer the similarity between the two instances by measuring a distance metric. Unlike traditional neural networks that learn to predict multiple classes, siamese networks learn a similarity function, which allows them to classify new classes of data without retraining the network again.

Some common loss functions used in siamese networks include contrastive loss, triplet loss, and angular loss. Contrastive loss aims to minimize the distance between similar instances and maximize the distance between dissimilar instances. Triplet loss extends contrastive loss by using triplets of instances: an anchor, a positive, and a negative, and aims to ensure that the anchor is closer to the positive than to the negative by a margin. Angular loss considers the angle between the vectors of the instances instead of the distance and tries to minimize the angle between similar instances and maximize the angle between dissimilar instances while also ensuring that the vectors have unit length. Multi-task loss can also be used, which includes a combination of different loss functions such as contrastive loss, triplet loss, and cross-entropy loss.

Reason for choosing this model

The method described in the paper integrates face detection and recognition alongside Pan-Tilt-Zoom (PTZ) cameras to automate attendance management. This method was chosen for several reasons:

Efficiency and Accuracy: Traditional attendance methods can be time-consuming and prone to errors. By leveraging facial recognition technology, this system aims to streamline the attendance process, reducing manual effort for professors and enhancing accuracy.
Reduced Hardware Requirements: Unlike previous methods using multiple cameras, this system employs a single PTZ camera and a dataset with single images of students, reducing hardware complexity and cost.
Adaptability: The system's reliance on a single image per student in the dataset enhances its adaptability to various classroom settings and scenarios, offering flexibility and scalability.
Face Recognition Model: The use of Insightface, a powerful CNN-based facial recognition model, indicates a focus on leveraging advanced technology for accurate facial recognition.
Real-time Attendance Logging: The system operates in real-time, detecting faces during lectures and generating attendance information at the lecturer's conclusion, providing instant records.
Transfer Learning: The application of transfer learning, utilizing pre-trained models suggests an attempt to leverage existing knowledge and adapt it to a custom dataset for improved accuracy.
PTZ Camera Integration: The choice of the SONY SRG-300SE PTZ camera allows for automated control via CGI commands, enabling seamless integration with the facial recognition system for attendance monitoring.

By combining advanced facial recognition models, PTZ camera technology, and an understanding of potential challenges, this method presents a comprehensive approach to automate attendance management in classrooms. Its emphasis on accuracy, adaptability, and real-time processing makes it a promising solution in the realm of educational technology.

Transfer Learning

Transfer learning is a technique in machine learning in which parameters from a model are re-used to boost the performance of a similar task. For the purpose of this paper, we are comparing results from multiple other models which are trained on different datasets and different backbone.

Model backbone	Training	lfw	cfp_fp	agedb_30	IJBB	IJBC
Resnet34	CASIA, E40	0.994667	0.949143	0.9495		
Mobilenet emb256	Emore,E110	0.996000	0.951714	0.959333	0.887147	0.911745
Mobilenet distill	MS1MV3,E50	0.997333	0.969	0.975333	0.91889	0.940328
se_mobile_facenet	MS1MV3,E50	0.997333	0.969286	0.973000	0.922103	**0.941913**
Ghostnet,S2,swish	MS1MV3,E50	0.997333	0.966143	0.973667	0.923661	0.941402
Ghostnet,S1,swish	MS1MV3,E67	0.997500	0.981429	0.978167	0.93739	0.953163
EfficientNetV2B0	MS1MV3,E67	0.997833	0.976571	0.977333	**0.940701**	**0.955259**
Botnet50 relu GDC	MS1MV3,E52	0.9985	0.980286	0.979667	0.940019	0.95577
r50 swish	MS1MV3,E50	0.998333	0.989571	0.984333	0.950828	0.964463
se_r50 swish SD	MS1MV3,E67	0.9985	0.989429	0.9840	**0.956378**	**0.968144**
Resnet101V2 swish	MS1MV3,E50	0.9985	0.989143	0.9845	0.952483	0.966406
EfficientNetV2S	MS1MV3,E67	0.9985	0.991143	0.986167	**0.956475**	**0.968605**
EffV2S,AdamW	MS1MV3,E53	0.998500	0.991429	0.985833	**0.957449**	**0.97065**
EffV2S,MagFace	MS1MV3,E53	0.998500	0.991571	0.984667	**0.958325**	**0.971212**
r100,AdaFace	MS1MV3,E53	0.998667	0.992286	0.984333	0.961636	0.972849
r100,AdaFace	Glint360k,E53	0.998500	0.993000	0.986000	0.962415	0.974843

Here is the summary of the multiple models that were considered for transfer learning. In our project we used model backbone as r100,Adaface and training dataset as MSIMV3,EF3 as it showed promising results with SOTA accuracy on multiple testing datasets.

To train our model on our own custom dataset we freeze the parameters for the first 95 layers and train the model for the last 5 layers which showed promising results. The number of layers that are used for training is a variable and can be increased for a bigger dataset to get better results but it also leads to overfitting for small datasets.

PTZ Camera Integration

Our study investigates how Pan-Tilt-Zoom (PTZ) cameras and facial recognition software can be combined to automate and expedite the process of recording attendance in classes.

Traditional attendance-taking techniques, such as roll calls and manual sign-ins, can be inaccurate and waste crucial instructional time. Facial recognition technology combined with PTZ cameras' dynamic capabilities provides a game-changing answer to this enduring problem. Our project aims to create a system that can identify and log student presence in real-time while causing the least amount of disturbance to the teaching and learning process.

As students are settled in the classroom, the PTZ camera scans through the whole classroom by covering every frame of the class. By matching photographed faces with an existing database thanks to the integration with facial recognition algorithms, the system can precisely record attendance without the need for user involvement. This adds an extra degree of security to educational institutions while also improving the effectiveness of attendance tracking.

We have used SONY SRG300SE Camera for this particular study. We were able to work and integrate it by using CGI.

Initial Camera Setup - Camera SONY SRG- 300SE initially is set via SNCToolBox (software) which allows it to use a static IP address for transmission and use of web applications on a PC. The camera allows the use of VISCA-over-IP commands, 3G-SDI, and CGI commands. It is designed professionally for lecture recordings, surveillance, etc. We particularly used CGI commands since they are easy to automate via Python scripting and also open up the possibility of ML use in the future for improvement in its movement. Hardware requirements for the camera are a LAN cable and an AC power source to connect via cable. Software SNCToolBox is required for the initial setup of the camera.

Why SONY-SRG300SE?- The SONY SRG-300SE, a renowned PTZ camera, boasts an extensive feature set that renders it particularly well-suited for such a project. It allows multicast

streaming, and automated control via ‘preset commands’ and ‘preset calls’. It also allows change of other video features such as brightness, exposure, white balance, and frames per second. Its 63.7-degree wide viewing angle and 30x optical zoom allow it to be used in small and big classrooms alike. It has a pan range of +/- 170 degrees and a tilt of +90 to -20 degrees.

Steps involved in setting up the environment

Hardware Components: SONY SRG-300SE, Ethernet Cable, 3G SDI Cable, SDI to HDMI Converter, and Laptop

Software Requirements: SNCToolBox

We primarily used CGI commands via Python scripting to automate the operations.

1. **Python Libraries**

   The libraries for giving out CGI commands are — ‘requests’, ‘threads’ and ‘time’

   The library for video capture - ‘opencv-python’

2. **CGI Commands**

   The syntax of CGI Commands for SONY SRG-300SE is:

   - ‘**http://<camera_address>/command/<cgi>?<parameter>=<value>[&<parameter>=<value>…]**’

   <camera_address> corresponds to the HTTP address;

   <cgi> replaced with ‘camera.cgi’ or ‘inquiry.cgi’ for some action or information respectively

   Say we use the command to get maximum zoom. The command in the terminal will be:

   - **‘wget http://192.168.0.102/command/ptzf.cgi?AbsoluteZoom=4000’**

There are 21 different types of commands for SONY SRG-300SE. We primarily use ‘PresetSet’ and ‘PresetCall’. The camera allows us to set upto 256 preset positions.

We integrate it via Python script and record its response. We predefine preset positions as per the lecture class in order to set up a tour and take attendance. We then integrate both and do attendance marking via video.

How is it better than other cameras?

We chose SONY SRG-300SE camera which comes with minimal setup requirements and full functionality. Unlike other Sony PTZ cameras, we are able to control the movement and functionality of the camera with the help of ethernet cable using cgi commands while others require RS-422 interface (VISCA-over IP command). The camera has capabilities to store upto 256 preset positions, has multicast (IP) streaming option and can show upto 3 video codec at a time.

Configuring Preset Positions

The following are the steps to set preset positions in a camera to call them:

1. Use the default web application of the camera to navigate and adjust the zoom and focus of the camera at a particular position
2. Put the CGI command given above in the terminal. At the start write ‘wget’ followed by the URL
3. It will set up the given preset position. Refresh the browser to look it up in the app
4. Repeat the above steps to add more presets

Challenges

1. As we zoom in for larger distances, the video gets darker. Changing the exposure is one fix or we can use other video formats
2. Slow movement of camera for larger zoom value slows the whole operation. It is also necessary in order to not get video blurred or distorted
3. Since we’ve used SDI video output in the above case, the camera doesn’t allow us to capture audio of classroom, It might hinder in further studies

Results

The report offers an end-to-end face detection and attendance system. It describes the architecture, training procedures, and recognition methods.

1)Training the model: The pre-trained model is trained on the generated dataset till the accuracy exceeds 95%.

2)Recognition: The system detects faces in real time and then updates spreadsheets to record the attendance of recognized kids.

3)Evaluation of Performance: The suggested system attained a real-time accuracy of 95%.

4)System Testing: The system's performance was evaluated under a variety of settings, including variable lighting levels and camera angles. Higher brightness and optimum conditions were found to boost accuracy.

Overall, the research conducted extensive tests into constructing an efficient end-to-end face detection and attendance system.

Limitations

- *Class-Specific Setting:* This study's PTZ camera is set up for a particular class. Making manual adjustments would be necessary to adapt it to different classes.

- *Low-Light Conditions:* In poorly lit areas, the system may have trouble correctly identifying and detecting students. This restriction may affect the attendance monitoring system's overall dependability, particularly in schools with poor lighting.

- *Possibility of Misidentification:* The system might unintentionally record false positives when students try to mark attendance for their peers by displaying a photo on their mobile device. Various parameters, such as the distance at which the image is presented, may affect the model's ability to discern between authentic and fake attendance attempts.

- *Dependency on eye visibility:* This architecture is impotent for faces with their eyes closed which makes this system restricted to faces with eyes open analogous to a camera.

- *Camera limitations:* Even with advancements in face recognition at a distance, the PTZ camera's physical limitations (e.g., mechanical defects) may prevent it from efficiently capturing faces in challenging environments or at long distances.

Conclusion & Scope

In this paper, we present a comprehensive end-to-end system that uses Convolutional Neural Networks (CNN) to monitor attendance with accuracy and precision. Our suggested technology has been rigorously tested in classroom settings, displaying impressive results and providing valuable real-time feedback to instructors on attendance rates.The system's resilience shines through as it overcomes common challenges found in attendance tracking systems, most notably overcoming occlusion, alignment errors, changing orientations, and varying levels of luminescence. These accomplishments highlight the system's reliability in real-world educational settings.

The experimental results show that the suggested system is capable of providing appreciable results in practical settings, and thus it can be utilized in various kinds of locations to carry out the attendance process.

However, it's imperative to acknowledge notable drawbacks within the proposed system. It suffers with lower-resolution photos and faces challenges when encountering distant faces. Furthermore, when presented with closed-eye instances, the CNN architecture displays weaknesses, limiting its effectiveness to faces with open eyes only.

To solve these restrictions and improve the system's performance, future research could concentrate on the development of systems capable of recognising faces regardless of video resolution limits. A potential option for improvement is to incorporate a preprocessing step that uses a super-resolution module, allowing the system to manage variable resolutions and improving its capacity for effective face recognition spanning multiple locations.

This progressive strategy strives to improve the system's adaptability, allowing it to thrive in a wide range of circumstances while eliminating existing constraints, hence increasing the effectiveness of attendance monitoring systems in real-world situations.

References

(1) Keras insightface: https://zenodo.org/badge/latestdoi/229437028

(2) SONY SRG-300SE PAN/TILT/ZOOM CAMERA: https://www.youtube.com/watch?v=OWJWzh1LBmI

(3) SONY PRODUCT SITE:https://pro.sony/en_IN/products/ptz-network-cameras/srg-300se

(4) An End-to-End Real-Time Face Identification and Attendance System using Convolutional Neural Networks https://ieeexplore.ieee.org/abstract/document/9029001

(5) Y. Su, X. Cui, Q. Xi, R. Zhang and J. Shen, "An Industry-Ready Single PTZ-Camera Based Attendance Management System," 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), London, UK, 2020, pp. 1-4, doi: 10.1109/ICMEW46912.2020.9105968.

(6) Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013, 2013, Volume 226 ISBN : 978-3-319-00968-1 Aditya Raj, Redouane Khemmar, Jean Yves Eratud, Xavier Savatier

(7) El-Sayeed Yasser, "Building Facial Recognition System in 30 days" https://medium.com/geekculture/building-a-facial-recognition-attendance-system-in-30-days-d4fceafbfba1

(8) U. Park, H. -C. Choi, A. K. Jain and S. -W. Lee, "Face Tracking and Recognition at a Distance: A Coaxial and Concentric PTZ Camera System," in IEEE Transactions on Information Forensics and Security, vol. 8, no. 10, pp. 1665-1677, Oct. 2013, doi: 10.1109/TIFS.2013.2261061.

(9) Ubong Essien, Godwin Ansa，"Attendance-Based Systems for Face Recognition Using a Camera Rail Approach" in REJOST ,Feb 2023