Face Recognition using Random Frame Selection

Nirmal G K, Narukurthi Hrishita, Sreeja Kochuvila and Navin Kumar  
Dept. of Electronics and communication Engineering  
Amrita Vishwa VidyapeethamBangalore  
[gknirmal18@gmail.com](mailto:gknirmal18@gmail.com), [hrishita.cv@gmail.com](mailto:hrishita.cv@gmail.com), [k\_sreeja@blr.amrita.edu](mailto:k_sreeja@blr.amrita.edu), [navinkumar@ieee.org](mailto:navinkumar@ieee.org)

***Abstract*—Face recognition has emerged as one of the crucial components in many sensitive applications. Several algorithms use different technologies for Face recognition. Additionally, new algorithms can leverage emerging technology to achieve better accuracy and processing time. In this study, we propose a novel approach that employs the random frame selection technique in conjunction with the Visual Geometric Group Face (VGG-Face) and RetinaFace algorithms for face detection and recognition. The experiment was conducted in a typical classroom environment measuring 32x25 meters, accommodating approximately 80 students, where a standard camera was positioned on the front wall. Real-time face recognition using deep convolutional neural networks was implemented to get an accuracy around 90-96% with 20-40 number of random frames. These findings underscore the system's effectiveness in crowded settings, paving the way for broader applications in high-density environments.**

***Keywords—Face recognition, frame selection, VGG model***

# Introduction

Face recognition (FR) is a biometric technology that aims to automatically identify or verify individuals by analysing their facial features [1]. It has gained significant attention and research interest due to its wide range of applications in various domains, including security, surveillance, access control, human-computer interaction, and personalization. In today’s world, face recognition is being introduced into a number of applications to enhance, security, or improve systems. Face recognition can be used to prevent accidents [2], enhance the security of a place [3], or even improve existing systems like attendance systems [4] in schools and offices. It can also be used to ensure that people are following protocols, like wearing masks [5] while entering certain areas, like hospitals. Such scenarios require extensive processing power, i.e., large resources. Here, we try to showcase a method to do processes in a more efficient manner, allowing systems to cut costs by using fewer resources.

At its core, FR involves capturing and analysing facial images or videos to extract unique characteristics and patterns that distinguish one individual from another. These characteristics can include the shape of the face, the arrangement of facial landmarks (such as eyes, nose, and mouth), and the texture or appearance of specific facial regions. If features and characteristics are extracted properly, the system can identify the person with greater accuracy. For example, many times a person’s face goes undetected or is wrongly detected because of ambient light, a mask or covering of face by some other clothes, people looking away from the camera, and so on. In some of these instances, wrong person might be questioned. Therefore, FR becomes extremely important and can play a very important role in our society. Many methods and techniques are used for FR. They vary in their simplicity, computation, and cost. In the landscape of data-driven tasks like face recognition, the effectiveness of established techniques for handling data imbalances is under scrutiny. The ubiquitously used vanilla triplet loss proves inadequate in addressing class imbalance, hindered by triplet explosion and inconsistencies, as noted in [6]. However, a remedy emerges in Cluster-based Large Margin Local Embedding (CLMLE), showcased to outperform rivals, particularly in imbalanced data scenarios. Similarly, X. Yin et al. in [7] presented a centre-based feature transfer framework for improved face recognition, countering under-represented data and classifier bias. Authors advocate against vanilla triplet loss in imbalanced deep representation learning and propose a novel loss function and clustering-based method, ameliorating efficiency and discrimination in such contexts.

M. S. Ejaz et al. [8] examined the use of Principal Component Analysis (PCA) on masked and unmasked face recognition. The suggested system employs the Viola-Jones method for detecting faces and PCA for efficient gallery picture representation. During the training phase, Eigen faces or ghost faces are obtained from the PCA feature space. In the test scenario, a target facial image (masked or not) is shown to be recognised. They demonstrated a statistically significant difference in accuracy between masked and unmasked face recognition using PCA. However, the proposed system has some limitations. The accuracy of the system depends on the quality of the input images and, the system may not work well with low-quality images. Additionally, the system may not be able to recognize faces that are heavily occluded or masked. CNN based face recognition systems are proposed by several authors. N. Ragesh et al. [9] investigated the masked face identification capabilities of Region-Based Convolutional Neural Networks (R-CNN). For door access control, a facial recognition system using R-CNN in hardware was presented. Kharchevnikova A, Savchenko AV [10] selected high-quality frames to study the challenge of face detection in video sequences. They created a lightweight convolutional neural network (CNN) model for face quality analysis termed "FaceQNet mobile" that was trained utilising knowledge distilled from the FaceQNet ResNet-50 model. When compared to traditional approaches, the methodology improved both the running time and the accuracy of face detection. The research shows that frame selection and quality evaluation are successful in video-based face recognition. Authors of the paper faced several limitations in their centre-based feature transfer framework for face recognition. Firstly, the computational costs associated with real-time video face recognition using deep convolutional neural networks (CNN) are high, requiring powerful servers with graphical processing units (GPUs) which can restrict performance on resource-limited platforms. Secondly, image quality variability due to environmental factors like lighting, resolution, and blur can lead to errors and unstable performance in the system. Selecting an appropriate threshold for key frame determination poses a challenge, as empirical methods may not be optimal. Existing frame selection techniques, including clustering-based algorithms and optical flow methods, can be time-consuming and may not provide accurate results compared to deep learning-based approaches. Hence, determining the best face recognition model hinges on various factors, including application-specific requirements and computational resources. Among the available options, VGG-Face emerges as a compelling choice due to its notable attributes. VGG-Face strikes an optimal balance between accuracy and simplicity, making it an accessible and efficient solution for many face recognition tasks. Its architecture, based on the VGGNet framework, is renowned for its effectiveness in extracting facial features, and it offers dependable performance across a range of scenarios. While other models such as FaceNet, OpenFace, ArcFace, and DeepFace have their strengths [11], VGG-Face's consistent reliability, ease of integration, and robust feature extraction capabilities position it as a highly persuasive choice for most face recognition applications.

In this work, we have used RetinaFace for detection and the VGG-Face model for recognition. Both models are highly efficient but require a lot of processing power to process an entire video or real-time footage. By taking out random frames, we are setting a limited number of processing iterations to get nearly the same level of accuracy. This helps cut costs and reduces processing time to give the same outcome as an entire video. By choosing random frames, we also try to avoid the possibility of blurry faces as people may be moving, and the movement may affect not one but multiple frames as they are continuous. For an average of 20 to 40 random frames we are able to record an accuracy ranging between 90% to 96%.

The rest of the manuscript is arranged as follows. In Section II, we have briefly discussed the overall system architecture. In Section III, the algorithm is discussed and presented while in Section IV, results are discussed. Finally, Section V concludes the paper.

# System Architecture

## Overall Architecture

The overall system is shown in Fig.1. The system consists of an Internet Protocol (IP) camera, an Access Point (AP), FR database and Processor and a Local system. The IP camera is set up in each of the location to capture the live feed frame by frame for a period of 30 minutes. An IP camera enables remote access as well. Frames will be selected at random using a selection algorithm and sent to the cloud for further processing. The IP camera provides a unique Real Time Streaming Protocol (RTSP) when connected to the common access point, in this case Raspberry Pi, which can be used for capturing the live feed. The camera stream is password-protected, which the administrator can configure. The RTSP connection comprises the camera's login, password, IP address, and stream number.

The Raspberry Pi acts as a common access point where the devices will be connected to the same network. Each device is assigned a unique IP address by the AP, often using Dynamic Host Configuration Protocol (DHCP). This IP address allows the camera and the processing unit to communicate with each other and share data without the need for WiFi connectivity.

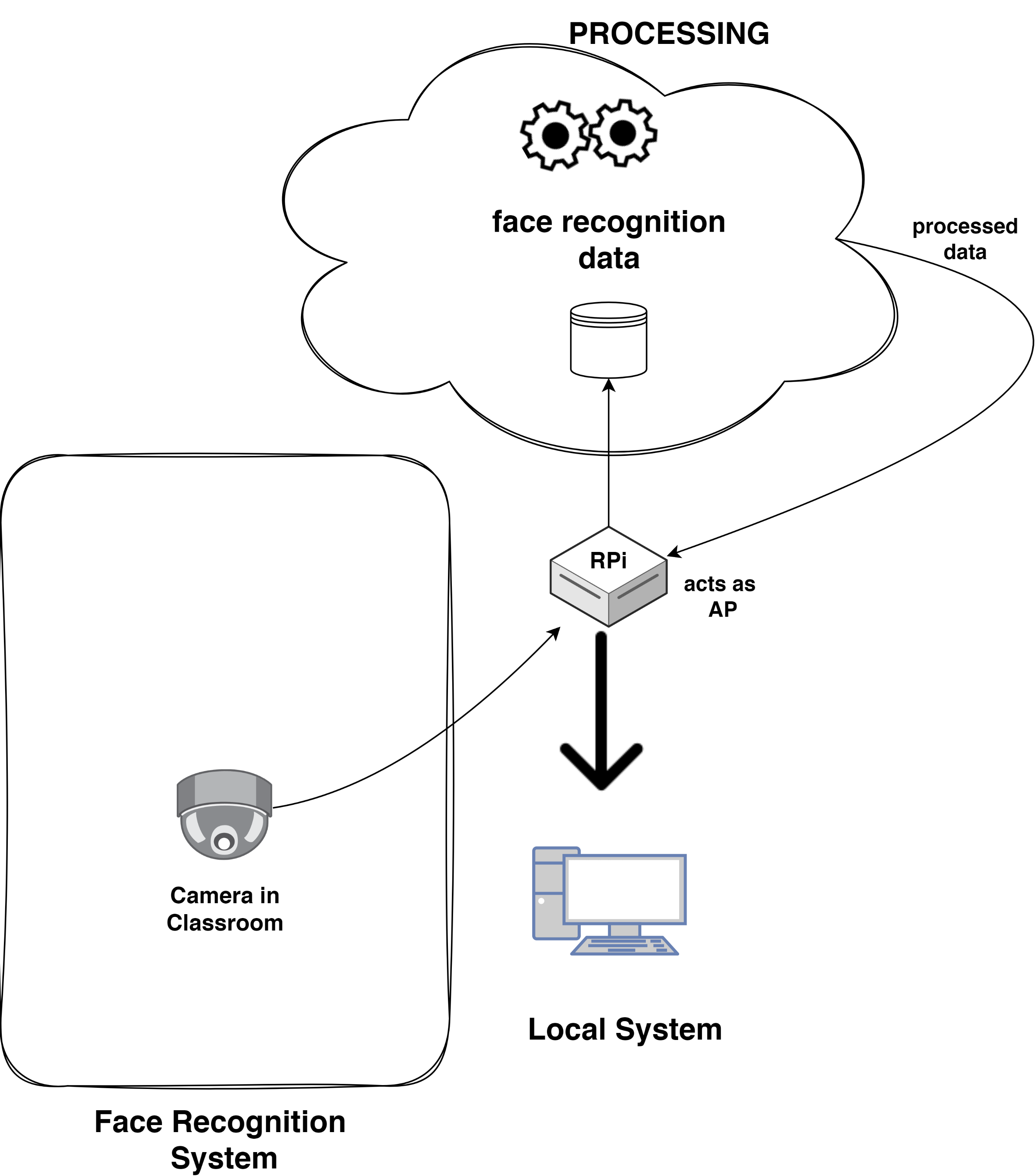


Fig.1: System Architecture

The system captures the video for a set period (ex: 15 seconds). Then it passes these 15 sec for processing to the random frame selection function which in turn provides an Excel sheet that contains the list of the recognised faces. Each frame first goes to face detection which will detect all the faces in the frame, and then those faces will be further processed to obtain the face embeddings. These embeddings will be compared against embeddings obtained from the training dataset to obtain the names of the individual in the frame.

The system uses the VGG-Face model for detection and the RetinaFace model for prediction.

# Design and Development

## Overall Algorithm

The algorithm as shown in Fig.2 uses RetinaFace for the detection of faces and VGG-Face for recognition. The algorithm involves the following steps:

Step1: Initialise the camera, and store a recording of a defined length. In this case, it is 15s.

Step2: Run the function to generate a set of random frames sampled from the video, based on the user’s input on how many frames need to be used.

Step3: Over each of these randomly selected frames, the RetinaFace detection algorithm is run over it. The extracted faces are then passed to the VGG-Face.

Step4: VGG-Face compares the extracted faces to the present faces in the database and returns the possible matches with a confidence value which indicates the similarity.

Step5: The output is stored as a data frame and exported in .xlsx format

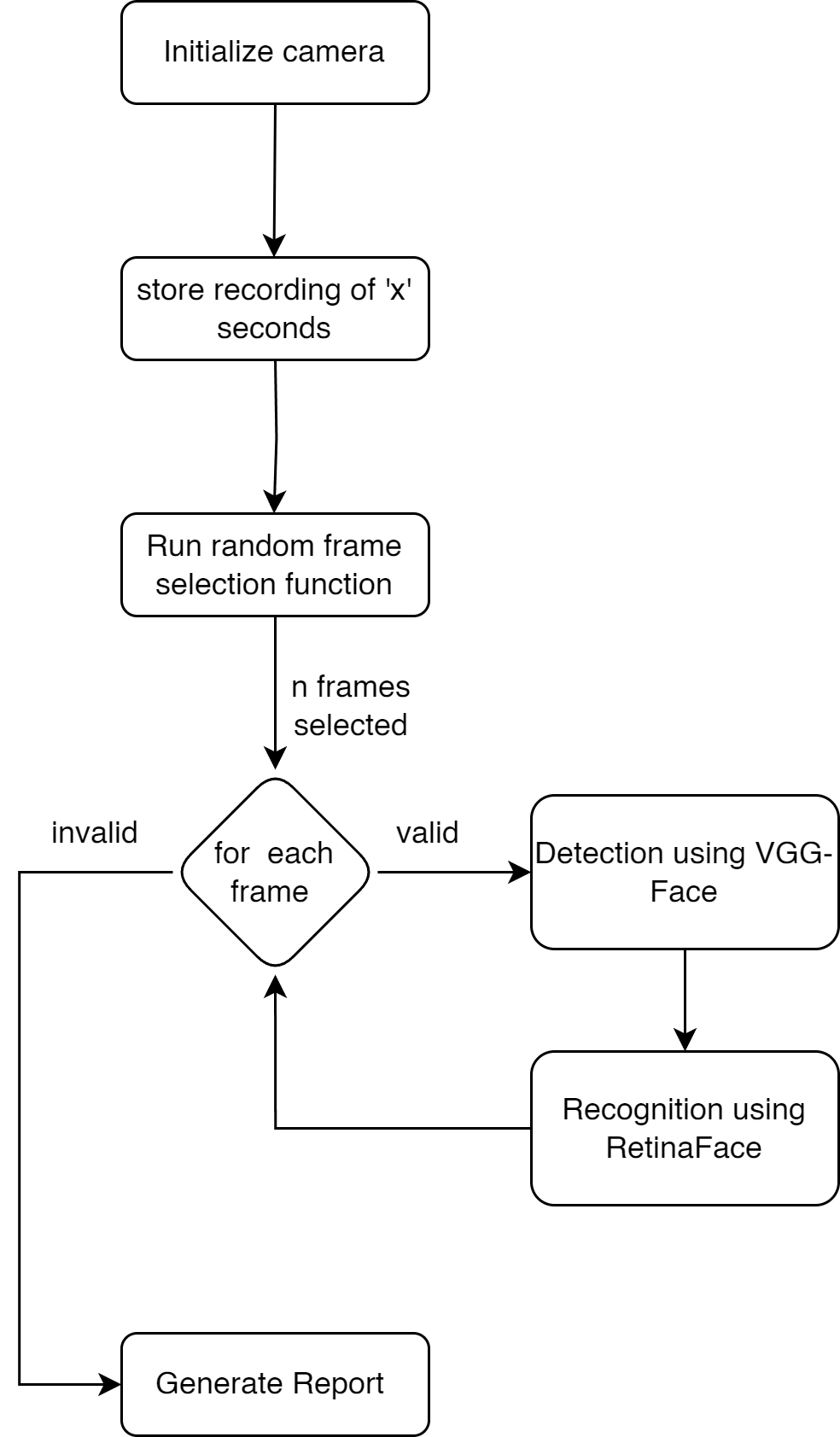


Fig. 2. Overall Algorithm

For the face detection, we use RetinaFace. The algorithm is presented next.

* 1. *RetinaFace Algorithm*

RetinaFace is a deep learning-based face identification system that achieves great accuracy [12]. It employs both extra-supervised and self-supervised multi-task learning. In detection, the following actions are taken:

Step1: Collect a diverse dataset of face images; ideally containing various poses, expressions, and lighting conditions.

Step2: Preprocess the dataset by resizing images, normalizing pixel values, and annotating facial landmarks for training.

Step3: Design or consume a RetinaFace architecture, a deep neural network specialised in detecting faces and localisation of facial landmarks.

Step4: Using the annotated dataset, train the RetinaFace model for effective detection of faces and landmark localisation.

Step5: Fine-tune the model, if necessary, on domain-specific data to improve detection performance.

Step6: Implement the trained RetinaFace model for face detection in new images or video frames.

Step7: Post-process the detected faces to filter out false positives and extract facial regions accurately.

Step8: If needed, perform facial landmark detection using the same RetinaFace model, allowing precise facial feature localization.

Step9: Utilize the detected faces and landmarks for various applications like facial recognition, emotion analysis, or facial expression recognition.

Step10: Continuously monitor and update the RetinaFace model to adapt to changing data distributions and improve detection performance.

Step11: Address privacy and ethical concerns by following best practices in handling facial data and ensuring compliance with applicable regulations.

Step12: Deploy the RetinaFace algorithm in our application ensuring efficient real-time face detection and landmark localization.

* 1. *VGG-Face Algorithm*

VGG-Face is a popular deep face recognition model developed by the Visual Geometry Group at the University of Oxford [11]. It consists of 22 layers and 37 deep units, similar to the VGG model trained with 1.2 million images from 1000 different categories. Unlike other models like Dlib and OpenFace, VGG-Face provides a 2622-dimensional face vector as its output. This model was trained using a dedicated dataset to improve its accuracy in face recognition tasks. A face recognition API based on the VGG-Face model has also been developed [13]. The following steps are followed:

Step1: Collect a labelled dataset of facial images containing individuals to be recognized.

Step2: Resize and normalize the images to a consistent format, e.g., 224x224 pixels, and augment them for diversity.

Step3: Acquire a VGG-Face model that has already been trained on a wide range of facial data.

Step4: Optionally, fine-tune the pre-trained model using your dataset to improve recognition for specific identities.

Step5: Implement a face detection algorithm (e.g., RetinaFace) if needed to identify and extract faces from input images.

Step6: Extract face embeddings by passing detected faces through the VGG-Face model to create unique feature vectors.

Step7: Create a database linking these face embeddings to their respective identities.

Step8: For recognition, extract embeddings from new faces, compare them to database embeddings using a similarity metric (e.g., cosine similarity), and identify the closest match.

Step9: Optionally, set a similarity threshold for recognition decision-making.

Step10: Implement post-recognition actions, like displaying the recognized person's name or performing access control.

Step11: Evaluate algorithm performance using a test dataset, measuring accuracy, precision, and recall.

# Results And Discussion

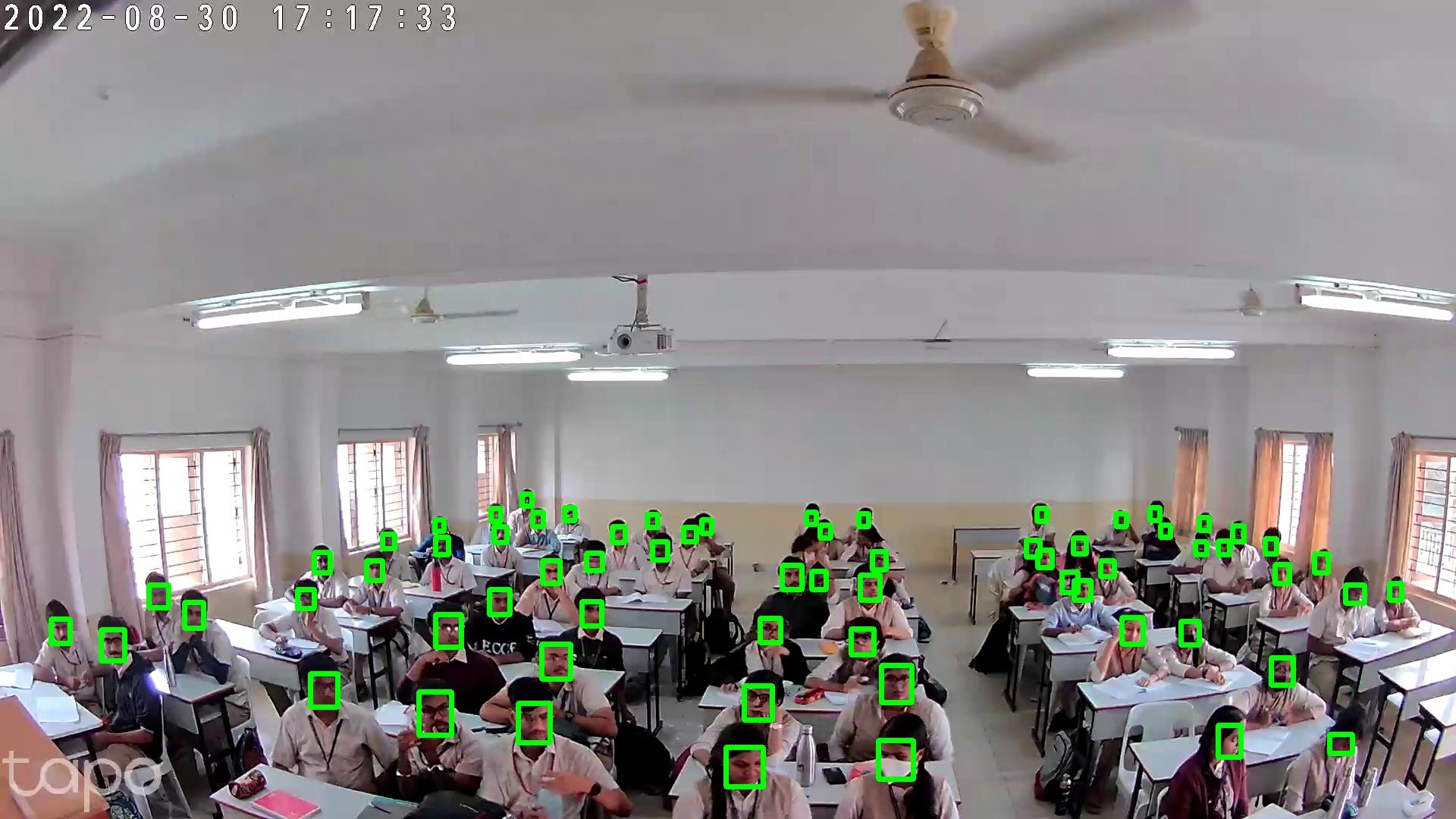


Fig. 5: Output of the Faces detected using Retina Face Model

The experimental setup was situated within a well-illuminated standard classroom accommodating a total of 86 students. The classroom's dimensions measured 30 meters in length and 25 meters in breadth. Seating within the classroom was organized in a grid-like fashion, resembling a 5-column table arrangement with two students seated at each table. Each column featured a total of 8 table chairs. The seating arrangement is illustrated in Fig.3. A camera, positioned at an approximate height of 3 meters on the front wall (not shown), was employed to capture the visual data. It is worth noting that the seating arrangement presented in Fig.3 represents individual frames extracted from a continuous video stream. The displayed output does not encompass the entirety of the data, as certain faces may not have been detected or recognized in a given frame but may have been identified in subsequent frames. Furthermore, it is imperative that all components and equipment utilized within this system operate within the same network infrastructure. The camera utilized in this experiment is the Tapo C110 WiFi Camera, which has a resolution of 3 MP.



Fig.6: Output of Facial Recognition using VGG-Face

Fig.3 shows a single frame from the captured video before being processed. This is one of the frames captured through random selection. The VGG-Face detection algorithm is run over this.

In Fig.4, shows the comparative analysis in terms of accuracy for four different face recognition models. We can see the accuracy of the most used face recognition models for each distance metric. As shown in the figure, the VGG-Face model, while using Euclidean as a distance metric, has the highest accuracy. All analyses performed will use this model along with this distance metric.



Fig.3: A single frame of the recorded video

Fig.5 shows the output after the face detection process. It is seen that almost all of the faces are detected. However, a couple of faces go undetected. These faces will be sent to the VGG-Face model for recognition.

In Fig. 6, we can see the output after the face recognition process. This is the output for a single frame and not for all the frames. The accuracy will increase as the number of frames increases, as shown in Fig.7.

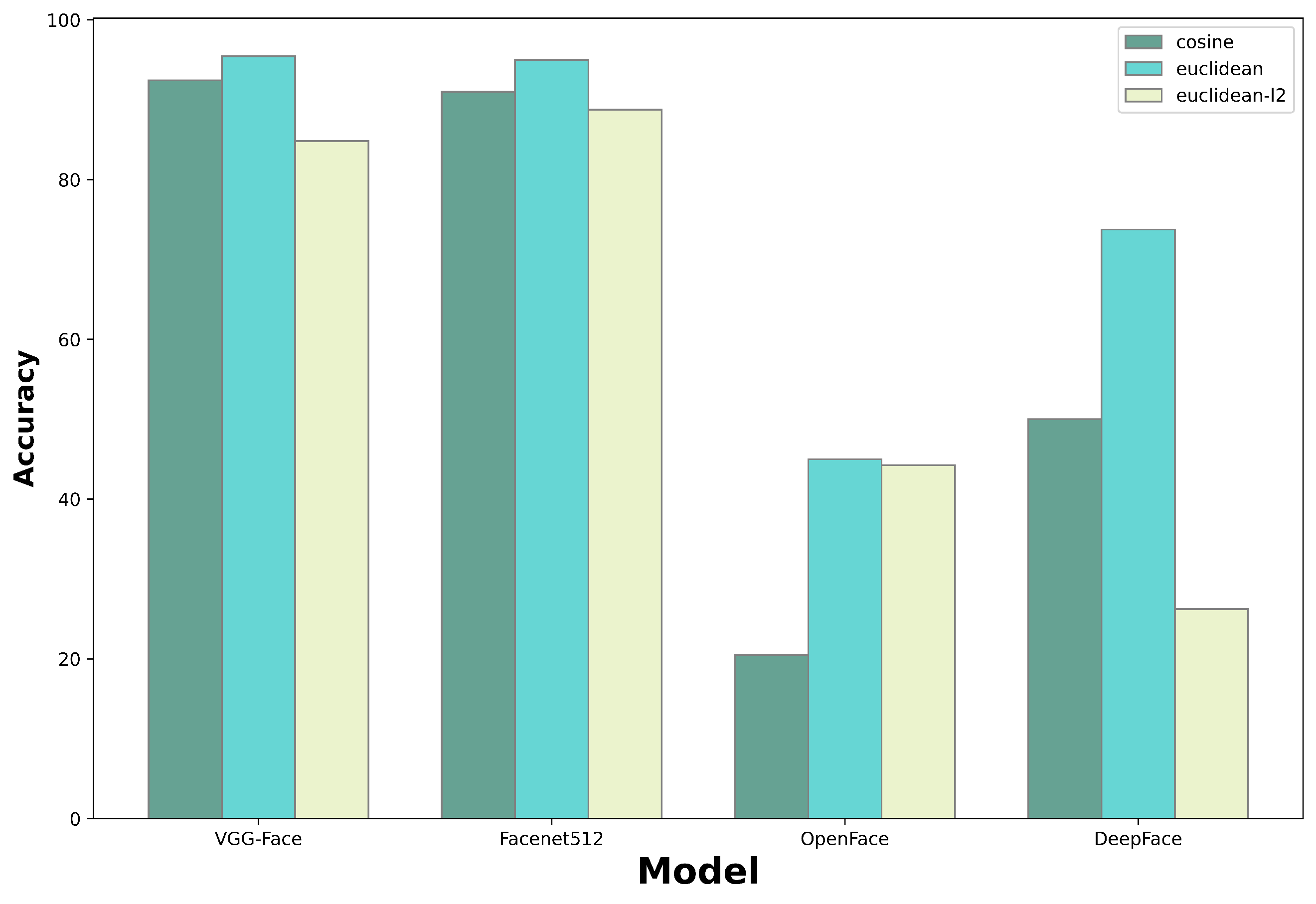
******Fig.4: Comparative Analysis of the models

Fig.7 shows the relationship between accuracy and the number of frames considered for detection and recognition. The graph also illustrates the corresponding processing time. Notably, it becomes evident that accuracy reaches its peak after analyzing 50 frames, with further increases in frame count yielding no significant accuracy improvements. This observation suggests that fewer frames suffice for effective detection and recognition.

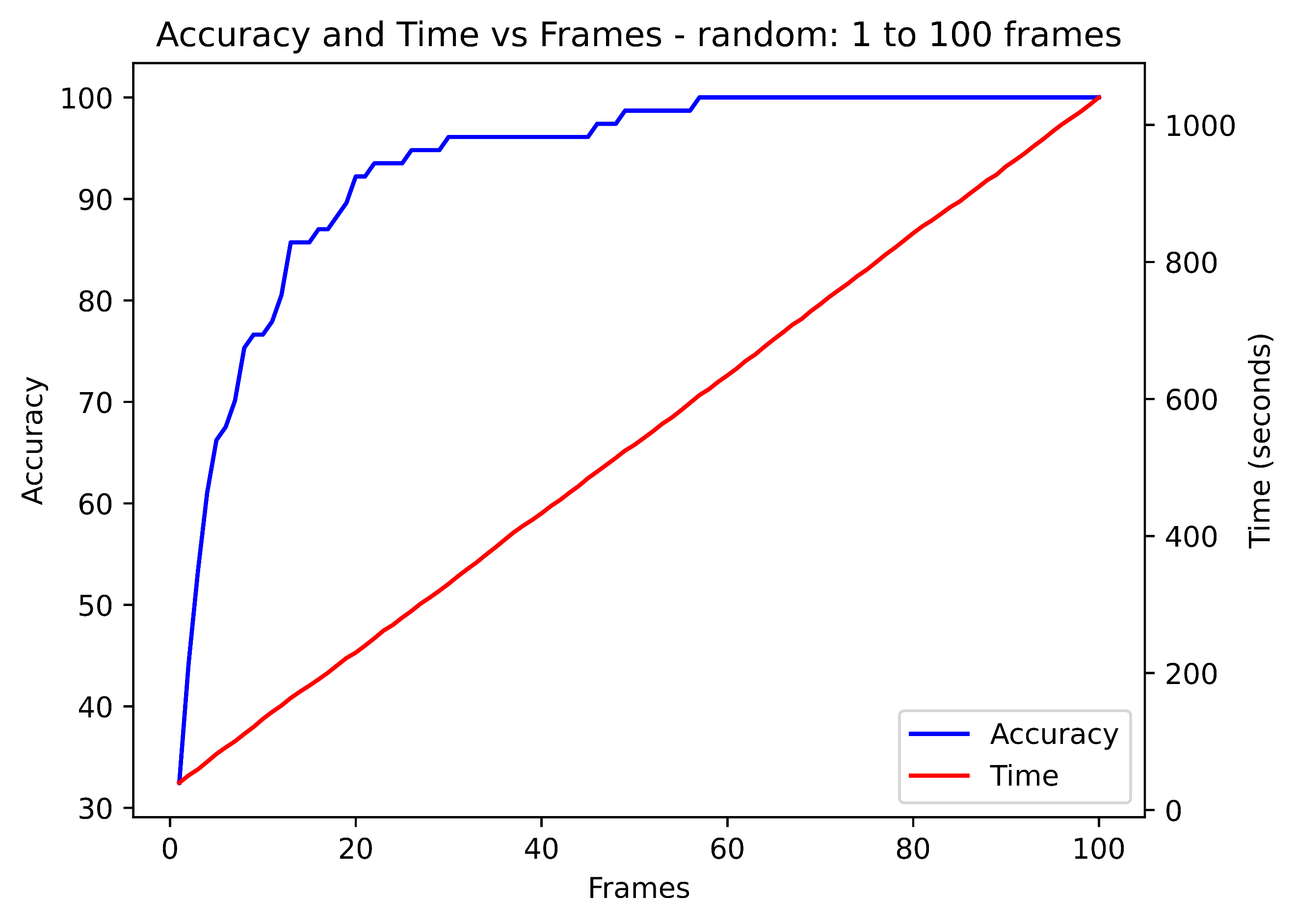


Fig.7: Accuracy and Time vs Frames Graph for random selection of frames

Additionally, the graphical representation depicts the progressive growth in processing time as the frame count increases, acknowledging that the processing time of each frame contributes cumulatively to the overall processing duration.

Fig.8, when compared to Fig.7, shows that processing continuous frames requires more frames to achieve the same level of accuracy. This is likely due to motion blur, which makes faces less clear.

# Conclusion

In this study, we employed a random frame selection-based face recognition system utilizing deep convolutional networks which yielded a detection accuracy ranging from 90% to 96% while demanding minimal processing power. The utilization of a reduced number of randomly selected frames allows us to maintain high accuracy in face detection and recognition while minimizing computational resources and processing time. This efficiency makes it more practical and efficient for deployment in densely populated environments. Furthermore, our observations indicate that the system functioned effectively with a single standard camera, though with limited detection range, and without precise positioning measurements. To enhance detection accuracy, future research may explore the deployment of high-quality cameras and multiple cameras positioned at optimized locations.

##### References

1. L. Li, X. Mu, S. Li and H. Peng, "A Review of Face Recognition Technology," in IEEE Access, vol. 8, pp. 139110-139120, 2020, doi: 10.1109/ACCESS.2020.3011028.
2. A. Manikandan and M. Sujith, "A Novel System for Real Time Drowsiness Warning and Engine Ignition Authorization using Face Recognition," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2021, pp. 1658-1663, doi: 10.1109/ICESC51422.2021.9532599.

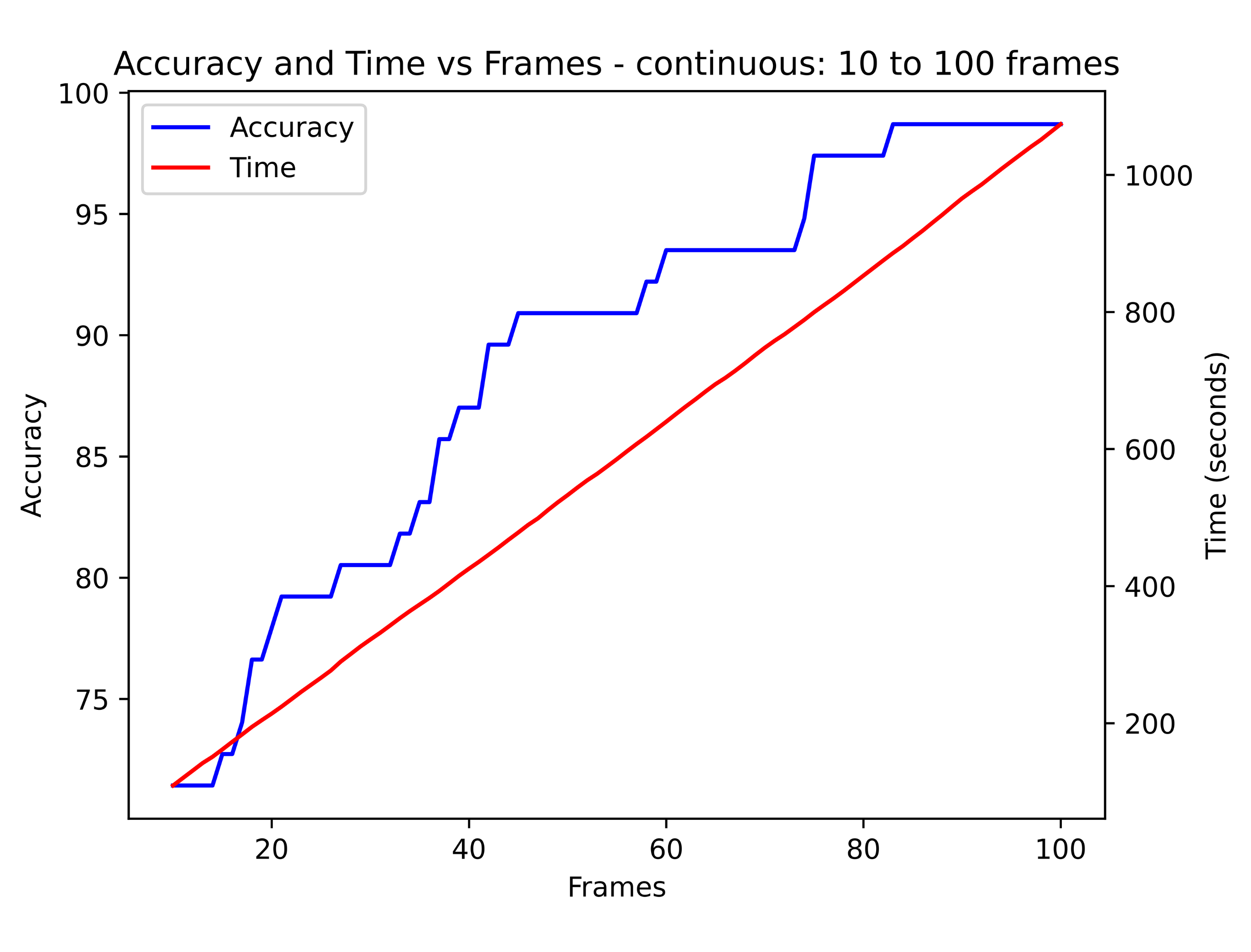


Fig.8: Accuracy and Time vs Frames for continuous selection of frames (normal video)

1. A. Stalin, A. Sha, A. S. Kumar, S. Nandakumar and G. Gopakumar, "Face Recognition at varying angles from distant CCTV Footage using Siamese Architecture," 2022 3rd International Conference for Emerging Technology (INCET), Belgaum, India, 2022, pp. 1-6, doi: 10.1109/INCET54531.2022.9824723.
2. P. N et al., "Fast and Reliable Group Attendance Marking System Using Face Recognition In Classrooms," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India, 2019, pp. 986-990, doi: 10.1109/ICICICT46008.2019.8993323.
3. E. Soumya, T. Singh and S. Santhanalakshmi, "An Intelligent System for Face Mask Recognition using Open Computer Vision and Modified YOLO Algorithm," 2022 IEEE 19th India Council International Conference (INDICON), Kochi, India, 2022, pp. 1-5, doi: 10.1109/INDICON56171.2022.10040177.
4. C. Huang, Y. Li, C. C. Loy and X. Tang, "Deep Imbalanced Learning for Face Recognition and Attribute Prediction," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 11, pp. 2781-2794, 1 Nov. 2020, doi: 10.1109/TPAMI.2019.2914680.
5. X. Yin, X. Yu, K. Sohn, X. Liu and M. Chandraker, "Feature Transfer Learning for Face Recognition With Under-Represented Data," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 5697-5706, doi: 10.1109/CVPR.2019.00585.
6. M. S. Ejaz, M. R. Islam, M. Sifatullah and A. Sarker, "Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 2019, pp. 1-5, doi: 10.1109/ICASERT.2019.8934543.
7. N. Ragesh, R. Ranjith and P. Sivraj, "Fast R-CNN based Masked Face Recognition for Access Control System," 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2022, pp. 1049-1055, doi: 10.1109/ICIRCA54612.2022.9985509.
8. Kharchevnikova A, Savchenko AV. 2021. Efficient video face recognition based on frame selection and quality assessment. PeerJ Computer Science 7:e391 <https://doi.org/10.7717/peerj-cs.391>
9. R. Tolegenov, K. Bostanbekov, D. Nurseitov and K. Slyamkhan, "Qualitative Evaluation of Face Embeddings Extracted From well-known Face Recognition Models," 2021 IEEE International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, 2021, pp. 1-5, doi: 10.1109/SIST50301.2021.9465952.
10. M. E. Wibowo, A. Ashari, A. Subiantoro and W. Wahyono, "Human Face Detection and Tracking Using RetinaFace Network for Surveillance Systems," IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society, Toronto, ON, Canada, 2021, pp. 1-5, doi: 10.1109/IECON48115.2021.9589577.
11. A. Firmansyah, T. F. Kusumasari and E. N. Alam, "Comparison of Face Recognition Accuracy of ArcFace, Facenet and Facenet512 Models on Deepface Framework," 2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE), Jakarta, Indonesia, 2023, pp. 535-539, doi: 10.1109/ICCoSITE57641.2023.10127799.