Automatic Attendance Registration System using Convolutional Neural Networks for Facial Recognition

Cristian H Sánchez Saquín
Technological University of Queretaro
Av. Pie de la Cuesta 2501, Nacional, 76148 Santiago de Querétaro, Qro.

Gabriel Rodríguez Feregrino
Technological University of Queretaro
Av. Pie de la Cuesta 2501, Nacional, 76148 Santiago de Querétaro, Qro.

Alejandro Gómez Hernández
Technological University of Queretaro
Av. Pie de la Cuesta 2501, Nacional, 76148 Santiago de Querétaro, Qro.

ABSTRACT
A system has been developed to optimize the attendance taking process in a school using computer vision and artificial intelligence techniques. The traditional method of keeping attendance records using a printed list is time-consuming and prone to errors. To address this issue, a system was implemented using neural networks, specifically YOLOv5 and VGGFace, for face detection and recognition of students in a group at the Technological University of Querétaro. The system operates autonomously throughout the entire class duration, utilizing the Intel RealSense SR300 camera as the video source for capturing images. These images were then labeled using the CVAT software and used to train the neural networks in Google Colab. A Python-based implementation was employed, combining the results of the two neural networks using a voting approach to achieve more accurate identification and recognition of each student. The obtained results, including the recognized students, are sent to the specified email address in the CSV file containing the class schedule, as well as to the email of the responsible teacher. By leveraging artificial intelligence and computer vision, this solution aims to streamline and enhance the attendance taking process in the school. It reduces the time dedicated to this task and minimizes errors associated with the traditional method of using a printed attendance list.

Keywords
face recognition, convolutional neuronal networks, face identification, YOLOv5, VGGFace.

1.INTRODUCTION
The process of taking attendance in schools has been a common task for teachers and administrators. Although it may seem like a routine task, attendance taking holds various importance and benefits in the educational environment. One of them is that it allows teachers to maintain an accurate record of student attendance, which is essential for meeting administrative and legal requirements, as well as for tracking each student's attendance throughout the school period. The attendance record is also useful for identifying patterns of absence and taking preventive measures in case of recurring issues.

Furthermore, the roll call at the beginning of the class establishes a routine and an orderly environment from the start, as students understand that their presence is required and that active participation in educational activities is expected. It is also an important tool for ensuring student safety, as in emergency situations such as evacuations or drills, the roll call allows teachers to quickly verify if all students are present and accounted for.

However, this routine activity can be time-consuming and, at times, prone to errors. Common errors include students not hearing their names and being marked as absent, teachers misplacing or damaging their attendance list, or students leaving the session after roll call. It is important to consider that this process takes between 4 to 8 minutes, resulting in the loss of valuable time that could be used for other activities.

Due to these challenges, various methods of attendance taking have been developed over time, such as electronic lists in Excel, automatic records in online classes, and even autonomous attendance systems. In the era of Artificial Intelligence (AI), there is a growing interest in applying these technologies to improve educational processes. In this regard, the use of neural networks has become an attractive option for automating attendance taking and streamlining this task in classrooms.

This research addresses the development of an automated attendance system in a university, utilizing neural networks as the primary. A combination of techniques based on YOLOv5 [1] and VGGFace [2] neural networks has been implemented to perform face detection and recognition of students in a group of Automation Technologies Engineering (ITA) throughout the entire duration of a class. This innovative approach aims to optimize the educator's time and enhance the accuracy of the attendance taking process.

Firstly, research was conducted on various pre-trained neural networks for image classification, such as ResNet50 [3], InceptionResNetV2 [4], and VGG16 [5]. These neural networks offer unique capabilities for identifying and classifying images based on their content in images and videos. After a thorough evaluation, it was determined that the YOLOv5 network was the most suitable choice for the purpose of this automated system, as it is specifically designed for multi-object detection in images and videos. The YOLOv5 network was trained using a dataset of faces collected from the group of ITA students, enabling it to detect and differentiate the faces of the students as if they were objects. However, although these models are capable of detecting objects within an image, even with some variations, they are unable to detect and recognize a person's face based on facial features.

To complement the process, a facial recognition network based on the VGGFace architecture was incorporated. Facial recognition is a system that detects facial features such as eyes, mouth, nose, distances between these features, and other characteristics. These features are then used to determine a person's identity. Facial recognition systems can recognize individuals even if they wear accessories or have experienced growth or changes over time, unlike the detection of their face alone, which can be affected by the variations. This network was trained using facial images of the students in the specific group, allowing for the validation of the results obtained by YOLOv5 through a voting system. By combining the face detection of YOLOv5 with the facial recognition of VGGFace, more accurate identification and recognition of the students' faces can be achieved.

Once a student's face is detected and recognized, the results are sent to the specified email address in a CSV file, which contains the class schedule and the corresponding teacher's email address. In this way, the educator has access to updated attendance records in an automated manner, saving time and effort in manual attendance taking.

This automated attendance system has the potential to revolutionize the way this task is carried out in schools. By eliminating the need to review a printed list and reducing the time spent on the process, teachers and administrators can focus their attention on other educational activities, thereby improving the overall efficiency of the educational process.

Multiple related works have been conducted in this area. M. Sajid et al. [6] propose a model that consists of two databases: one for storing attendance records and another for storing patterns of students' faces. The camera is positioned at the front of the classroom at an angle that allows it to capture all the faces. The captured image goes through Gabor filters, and 31 fiducial points are calculated for each student, considering the nose, eyes, and lips. This information is compared to the data stored in the database, and the process is repeated three times throughout the class. The processing calculations are performed by a server.

T. A. Kiran et al. [7] propose a work with a similar functioning to the previous one, with the difference that the camera is placed on the door frame, and the student is required to look at it. The captured image is converted to grayscale, and feature extraction is performed using the Fisher Face method. Recognition is carried out by comparing the extracted features with the data in their database.

J. W. S. D'Souza et al. [8], similar to [6] and [7], utilize a database of students' faces. The camera captures 5 images at 5-minute intervals. Each image is segmented using the Haar Cascade algorithm, which leverages edge and corner features. For facial recognition, they employ the Histograms for face identification algorithm, which is claimed to work well under varying lighting conditions. Finally, the extracted facial features are compared with the information stored in the database.

Sagar Chanchal, Tanmay Desai, and Dipti Jadhav [9] mention a method where the teacher records a video of the students in the classroom using an application, and the information is sent to a computer for processing. The system utilizes the Haar Cascade algorithm to detect and extract the face of each person in the video. Facial recognition is then performed using the Local Binary Patterns Histograms (LBPH) algorithm. If a face is not detected, a notification is sent to the teacher for manual review.

A. R. S. Siswanto et al. [10] propose a system where students are required to enter their ID into the system, which activates the camera to detect their face and check if it matches the database. The database is created using images of students' faces captured in a controlled environment with consistent lighting, background, and facial pose. For face detection, they employ the LBP Cascade and Haar-Left Eye and Haar-Right Eye algorithms to determine the positions of the left and right eyes, ensuring that the subject is looking directly at the camera.

Aditya Kapse et al. [11] propose a solution that leverages artificial intelligence in addition to computer vision techniques. In their approach, the camera is placed on the classroom ceiling, capturing a video in which they attempt to detect faces in each frame. If a face is not found in a frame, they move on to the next one. Face detection is performed using the Histogram of Oriented Gradients (HOG) method, primarily for detecting frontal poses. However, if the face is rotated, they employ the Face Landmark Estimation Algorithm (FLE) to consider the distances between the eyes, nose, and chin. The extracted features are then fed into a Convolutional Neural Network (CNN) to detect finer characteristics and obtain a 128-bit feature vector for each face. Finally, a Support Vector Machine (SVM) is used to compare the feature vectors with the database of faces and determine the person's identity.

E. Winarno et al. [12] utilize the Viola-Jones algorithm for face detection, which marks the detected face with a region of interest (ROI) to determine its location in the image. Once the ROI is obtained, the face is cropped and converted to grayscale. They perform histogram equalization as a contrast and brightness adjustment to enhance facial recognition. Subsequently, they apply Principal Component Analysis (PCA) to reduce the dimensionality of the resulting vector. For classification purposes, they use the Mahalanobis distance method to compare the calculated vector with the vectors stored in the database.

R. C. Damale and B. V. Pathak [13] employ a Deep Neural Network (DNN) for face detection, which is based on the Single Shot Detector (SSD) with ResNet as the underlying network. They use a pre-trained network with caffe prototxt files for face detection. Once the faces are detected, they are cropped and scaled to 128 x 128 pixels. They then utilize PCA and LDA for extracting facial features, and the faces are divided into training and testing sets. With the extracted feature vectors, they perform classification using SVM and Multilayer Perceptron (MLP) algorithms to identify the faces.

Nurkhamid et al [14] developed a facial recognition system for remote meetings via platforms like Google Meet. They collected training images from virtual sessions they conducted. For face detection, they utilized the HOG (Histogram of Oriented Gradients) algorithm. Then, they employed Face Landmark Estimation to recognize facial landmarks. With the extracted facial features, they utilized a Deep Convolutional Neural Network (DCNN) for facial encoding, and SVM as the classification algorithm. The system was tested with frontal face positions as well as upward and downward orientations.

Khan, S et al [15] propose a system in which the camera captures images of the group at the beginning and end of a session, recognizing the faces that belong to the group and those that do not. At the end, it sends an attendance report via email to teachers, parents, and students. For face detection, they utilized a pre-trained YOLOV3 model. Facial recognition is performed using Microsoft Azure services. Tkinter is used to create the user interface for the system.

2.METHODOLOGY

An ensemble of two neural networks was created, where one network is responsible for detecting multiple faces in an image, while the other network is used for face recognition. This ensemble was implemented in Python along with an automated system to execute it on a weekly schedule for a semester. At the end of each session, an attendance list is sent to the professor via email.

2.1.Facial detection neural network

To develop the system, a network capable of detecting multiple faces in a single image was required, specifically 16 faces. It was also important to have a good balance between accuracy and speed, meaning the network should be able to detect objects with high accuracy in real-time. Additionally, the chosen network needed to be compatible with embedded systems and easy to use and implement. YOLOv5 fulfilled all these requirements, as it was one of the most advanced models available at the time of development.

To employ YoloV5, it was necessary to create a dataset of the faces of the students in the target group. For this purpose, Intel RealSense SR300 camera was used, which has a resolution of 1920 x 1080 pixels at 30 fps. To create a realistic dataset that captured the actual scene information, a 2-hour session was recorded. During this session, the students were asked to wear commonly used accessories such as caps, face masks, sunglasses, contact lenses, etc. They alternated the use of these accessories and performed natural body and head movements. Additionally, they randomly sat in different positions, and the recording took place at night using the classroom's ambient light to ensure constant lighting conditions. It is important to emphasize that the experiments were conducted with prior authorization from both the coordination and the students; however, the dataset is not public. Following this, individual recordings of each student were made by focusing the camera's field of view on their face, and as mentioned before, the accessories were also alternated during these recordings.

After extracting individual frames from each student's videos and filtering out low-quality images, the faces in each image were annotated using the online software CVAT. A total of 10,672 group images were annotated, where all the students were present in each image. Each face was annotated as a separate class, resulting in a total of 16 classes. On average, approximately 2,500 images were annotated per class. Each face was labeled as a class, resulting in a total of 16 classes, with an average of 2,500 images per class. Finally, the annotated images were exported in the Yolo1.1 format and split into an 80% training set and a 20% validation set.

The training of the YoloV5 model was conducted in Google Colab using the Python programming language. The model was trained for 40 epochs, and the following results were obtained:

- A precise and real-time detection of multiple faces in a single image was achieved in the field of image processing and artificial intelligence.
- The YOLOv5 model demonstrated excellent capability to generalize face detection, even in the presence of accessories and lighting variations.
- The implementation of the model in an embedded environment met the requirements of efficiency and limited resources.
- The ease of use and simple implementation of YOLOv5 facilitated the integration of the system into the target group's environment.

After training YOLOv5, the network is tested with 2676 randomly selected images from the recorded session. It is important to note that these images were not used for training and validation. Subsequently, metrics are calculated to provide an understanding of how well the model generalized the classes during validation. Using the F1 confidence curve, we can assess the overall system performance by combining precision and true positive rate, as shown in Figure 1.

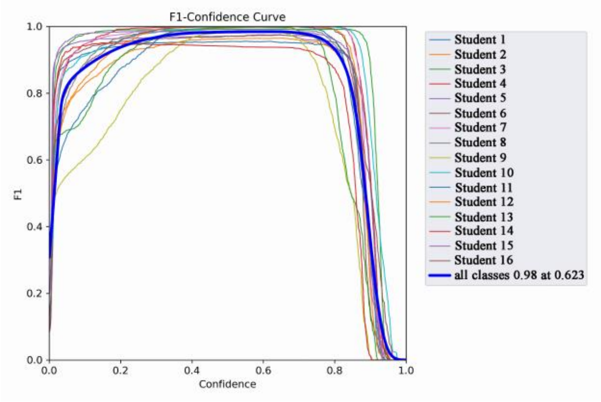

Fig. 1. YOLOv5 F1-Confidence curve with 2676 random images from the recorded session.

On the other hand, the confusion matrix is used for evaluating classification models and allows visualizing the performance of the model in predicting the classes of the dataset by comparing the model's predictions with the true classes.

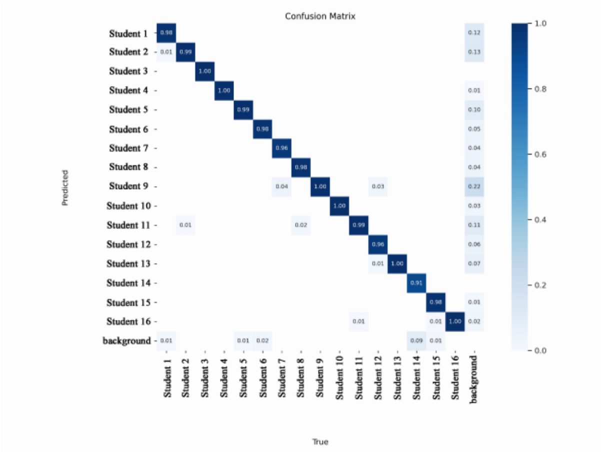

Fig. 2. YoloV5 confusion matrix.

As shown in Figure 2, the training of YOLOv5 resulted in 10 classes with a range of 99-100% accuracy, with the lowest being 91% for student 14 in predictions on the validation dataset.

2.2.Facial recognition neural network

The detection model used allows us to identify the faces of the individuals it has been trained on. However, this model treats faces as individual objects, searching for similarities in the image based on the information it was trained with. This can result in false positives when there is a significant similarity between the faces of two different people, leading to the erroneous detection of a person as a member of the group.

To address this issue, the VGGFace facial recognition model was utilized. This model analyzes unique facial features such as eyebrows, eyes, inter-eye distance, size, among others, to determine the identity of a face. By incorporating the VGGFace model, false positives can be reduced. VGGFace is a facial recognition model developed by the Visual Geometry Group (VGG) research group at the University of Oxford. It is based on the VGGNet architecture, a widely known and utilized deep convolutional neural network (CNN) in the field of computer vision [2].

Like YOLOv5, the VGGFace model also requires training using a dataset containing images of the desired individuals to be recognized. However, unlike YOLOv5, each image in the dataset should only contain the face of the respective person, and these images should be organized in individual folders. To achieve this, the same dataset used in YOLOv5 was utilized, but the faces were cropped, scaled to 224 x 224 pixels, and organized using the bounding boxes and corresponding labels obtained from the predictions made by YOLOv5 once it was trained. Consequently, the original 16 classes were kept, and an additional class was added, representing random faces. This additional class contained images of random faces not belonging to the studied group. The following graphs were obtained during training.

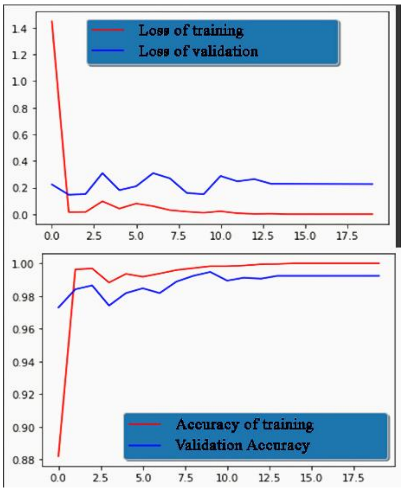

Fig. 3. VGGFace Training graphs with 5100 randomly selected faces from the recorded session.

As shown in Figure 3, a 98% learning accuracy was achieved by the network, and no pattern of overfitting or underfitting is observed.

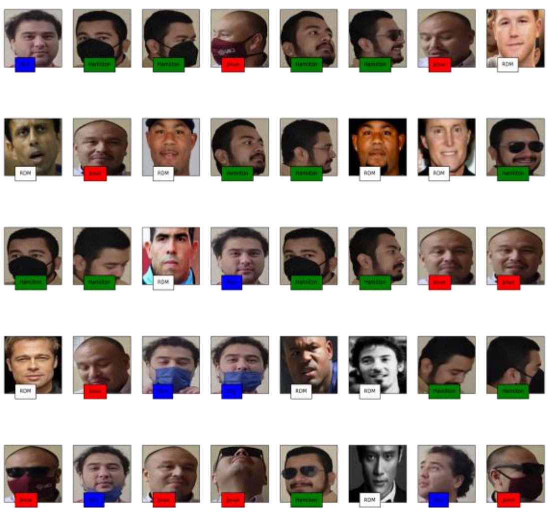

Fig. 4. Application of VGGFace on images with different poses.

In Figure 4, the performance of VGGFace can be observed for face detection using different accessories such as masks, sunglasses, reading glasses, head movements, and combinations thereof.

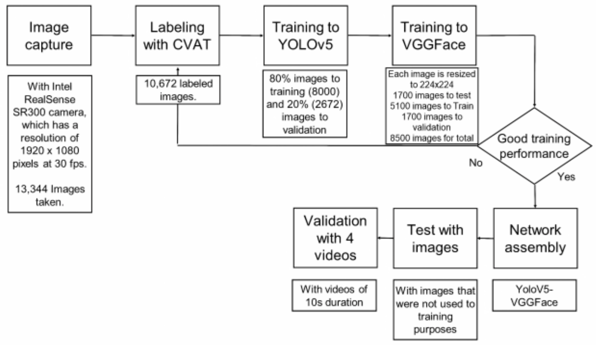

Fig. 5. Flow chart of de system face detection system.

Figure 5 shows the series of steps for the facial recognition system. It begins with on-site image capture and labeling. Once the images are labeled, the training of YOLOv5 and VGGFace takes place, repeating the process until achieving at least an 85% confidence index. Following this, the networks are assembled and tested with some images that were not used for training or testing. To validate the network, 4 random videos from the recorded session were used.

2.3.Automated attendance system

Up until now, the system consisted of two separate neural networks. The first network was responsible for detecting the faces of the students in the video and determining the identity of the detected face among the individuals it was trained on. The second model, on the other hand, was responsible for recognizing the face of a person when given an exclusive image of their face. However, in order to automate the process, it was necessary to assemble these two models together.

YoloV5 serves as the main model because it can detect the location of faces within the entire image and providing predictions of potential individuals. These predictions are accompanied by bounding boxes for each predicted result, allowing the image to be cropped to isolate the face detected by YoloV5. This cropped face is then used as input for the VGGFace model, which makes its own prediction of the face's identity. Finally, the results from both models are compared using the probabilities assigned by each method, and the actual identity of the person is determined.

Once the two models are assembled, it is necessary to make the system function during class hours. For this purpose, an input file in CSV format is required, containing the weekly schedule of the group, as well as the name and institutional email of the responsible professor. The system will run continuously every day throughout the semester, and when the local date and time match the scheduled date and time, it will start analyzing an image every 3 seconds. Each image will go through the ensemble of neural networks, and the results will be stored in a CSV file.

Finally, the results will be filtered to include only relevant data for the teacher, such as the time of the first detection and the time of the last detection for each student. The data will also include the student's name, date, attendance percentage (determined by the number of detections throughout the class), and the attendance status. The attendance status will be classified as follows: less than 80% will be considered as an absence, between 80% and less than 90% will be considered as a delay, and 90% or above will be considered as attendance. Please refer to Table 1 for an overview of the results.

Table 1. Attendance pass table.

Roll call ITA					
Name	Detections	Percentage	Entrance	Exit	State
Student 1	258	94.85%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Present
Student 2	271	99.63%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Present
Student 3	269	98.90%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Present
Student 4	272	100.00%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Present
Student 5	266	97.79%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Present
Student 6	272	100.00%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Present
Student 7	270	99.26%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Present
Student 8	193	70.96%	[' Date: 2023-04-17 15:46:41.162168']	[' Date: 2023-04-17 16:16:31.030075']	Missing
Student 9	267	98.16%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Present
Student 10	268	98.53%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Present
Student 11	152	55.88%	[' Date: 2023-04-17 15:48:38.823377']	[' Date: 2023-04-17 16:16:31.030075']	Missing
Student 12	161	59.19%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Missing
Student 13	23	8.46%	[' Date: 2023-04-17 15:46:16.688848']	[' Date: 2023-04-17 16:15:56.413317']	Missing
Student 14	259	95.22%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Present
Student 15	272	100.00%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Present
Student 16	246	90.44%	[' Date: 2023-04-17 15:44:24.342824']	[' Date: 2023-04-17 16:16:31.030075']	Present

The system's camera should be positioned in a way that provides a complete view of the students' faces. Therefore, it needs to be placed in an elevated position with a specific tilt to prevent taller students from obstructing the view of shorter students.

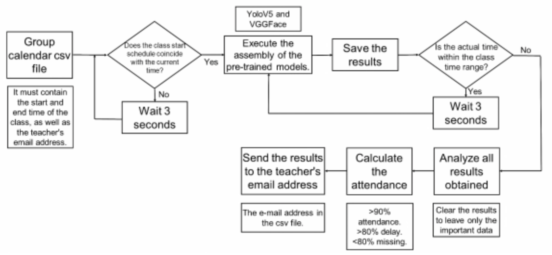

Fig. 6. Flowchart for the attendance-taking system.

Figure 6 illustrates the flowchart for the attendance-taking system explained at the beginning of Section 2.3.

2.4.Face identification

For the voting system, a comparison is made between the results obtained by YOLOv5 and VGGFace. If these results are identical, the outcome is considered correct. For example, if both models detect student 1, then it is indeed student 1. However, if the results differ, an additional verification process is carried out.

First, the prediction of YOLOv5 is examined to determine if it exceeds a confidence threshold of 82%. If it does, the second prediction given by VGGFace is then checked. If this second prediction matches that of YOLOv5, it is considered a correct detection. This conclusion is based on experimental evidence.

However, if the predictions from YOLOv5 and VGGFace still differ after the previous verification, an additional evaluation is conducted. In this stage, the confidence of YOLOv5's prediction is checked to see if it exceeds the established threshold of 85% through an experimental trial-and-error process. If this criterion is met, it is concluded that YOLOv5 correctly detected the face.

If none of the conditions are met, it is considered that the result is incorrect.

3.EXPERIMENTATION

The video referred to in section 2.1 is employed, wherein the camera is positioned in the corner of the classroom at an approximate height of 2 meters and tilted to capture most of the students' faces. It records the natural movements of students during a class, including leaning forward to take notes, changing positions in their seats, raising their arms to participate, turning their heads to converse with classmates beside or behind them, and the random use of accessories, among other actions.

This video was divided into four random sections, each lasting 10 seconds. Each section captures moments where it is possible to observe all the students in the classroom, occasionally showing partial facial views. This division was carried out to assess the accuracy of the proposed implementation. The videos were analyzed using the autonomous recognition system explained in the preceding section, resulting in the following outcomes.

Fig. 7. Overview of the classroom.

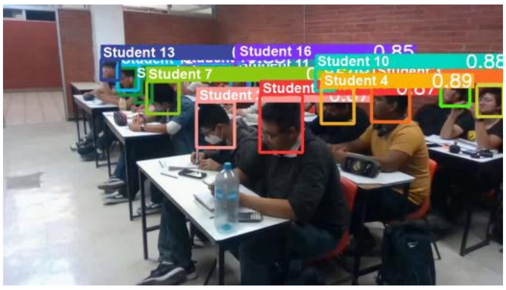

Fig. 8. Frame 510, video 1. Predictions using YOLOv5+VGGFace.

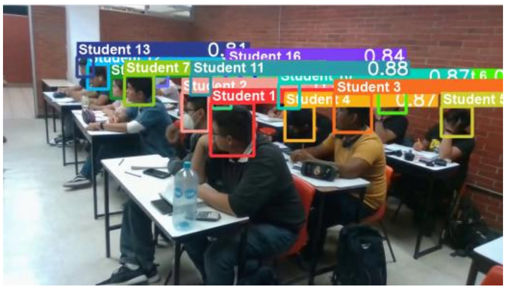

Fig. 9. Frame 310, video 2. Predictions using YOLOv5+VGGFace.

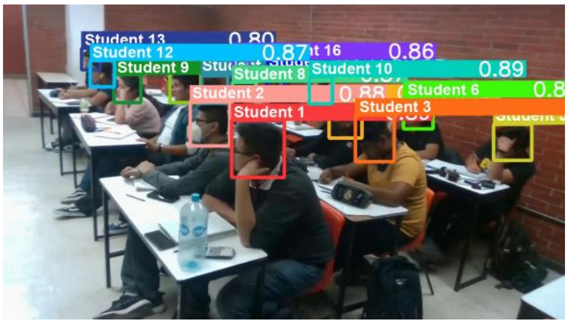

Fig. 10. Frame 345, video 3. Predictions using YOLOv5+VGGFace.

Through an analysis of the obtained voting results, it was observed that the student identified as 'Student 13' exhibited a low detection rate, even being nonexistent, as seen in figures 7-9. This circumstance can be attributed to their positioning in a corner of the classroom. When combined with the movements of their peers and the camera perspective, it resulted in them going undetected in numerous video frames. Specifically, it became evident that their facial features were not fully captured, preventing the VGGFace model from achieving accurate detection. A parallel situation unfolded with the students identified as 'Student 11' and 'Student 12'.

Table 2. Results of the videos applying YOLOv5 and YOLOv5+VGGFace

	Detections							
	Video 1		Video 2		Video 3		Video 4	
Names	Yolo	Yolo+VGGFace	Yolo	Yolo+VGGFace	Yolo	Yolo+VGGFace	Yolo	Yolo+VGGFace
Student 1	84	84	27	27	165	165	272	258
Student 2	84	84	27	26	165	149	272	271
Student 3	84	84	27	27	165	157	272	269
Student 4	84	84	27	27	165	152	272	272
Student 5	84	83	27	27	165	161	272	266
Student 6	84	84	27	27	165	165	272	272
Student 7	84	83	27	27	165	165	272	270
Student 8	82	82	26	23	165	164	262	193
Student 9	73	64	22	18	164	164	272	267
Student 10	84	83	27	26	165	159	272	268
Student 11	50	41	27	12	154	66	233	152
Student 12	81	28	27	17	165	143	272	161
Student 13	82	3	24	0	165	12	237	23
Student 14	84	72	27	27	163	163	272	259
Student 15	84	84	26	26	165	165	272	272
Student 16	83	80	27	27	165	165	272	246

Table 2 presents the results of detections using YOLOv5 and YOLOv5+VGGFace. It is noteworthy that YOLOv5 does not exhibit the same impact as the YOLOv5-VGGFace combination, as observed in video 4 of Table 2. This discrepancy is attributed to the fact that YOLOv5 does not directly recognize the face; instead, it discerns and evaluates whether the facial region resembles any of the classes, achieving a minimum of 82% similarity for the individual in question. This is due to the training dataset, which captured images from the shoulders upward and incorporated diverse head positions, as illustrated in Figure 4. Consequently, YOLOv5 encompasses information about both the lateral and frontal aspects of each head. Therefore, even if the face is not entirely visible, YOLOv5 can still detect it. In contrast, VGGFace lacks sufficient facial information for precise detection.

However, relying solely on YOLOv5 for facial detection and recognition is not advisable. In cases where the face is not fully visible, it can easily result in false positives and confuse one person with another. This could lead to inaccuracies throughout a 2-hour class. Nevertheless, based on the conducted experimentation, it performs well in generalization.

4.ANALYSIS

The system correctly recognizes each of the classes, but it has its limitations. When a student rotates their head and important facial features are not visible to the camera, VGGFace fails to make accurate detections, while YOLOv5 succeeds. This is because YOLOv5 was trained with images of students' heads in various positions and rotations, providing it with the necessary information to differentiate between classes even with challenging angles.

On the other hand, student 13 was situated at positions slightly farther from the camera, and being of shorter stature, they were often obstructed by their fellow classmates in most of the analyzed images. This resulted in their non-detection in most of the images. This issue can be addressed by ensuring a uniform distribution of tables and students within the classroom.

To overcome these limitations, it is recommended to consider a combination of both YOLOv5 and VGGFace, leveraging the strengths of each model for improved detection and recognition accuracy. Additionally, optimizing the setup of the classroom, such as ensuring proper positioning and minimizing obstructions, can enhance the overall performance of the system.

One advantage compared to the authors mentioned in section 1.6 is that a database was not used to perform the comparison of facial features with every image. This process would take much longer than the current 3-second sampling time. Consequently, the use of platforms for storing and accessing databases was also avoided.

By avoiding the reliance on a database, the system can operate more efficiently and with faster response times. It eliminates the need for extensive computational resources and reduces the complexity of the overall implementation.

Performing image analysis every 3 seconds throughout the entire class makes the system more robust compared to approaches [6, 7, and 15], where only the beginning or beginning and end of the class are considered. By analyzing images at regular intervals, the system can capture potential events such as a student leaving the classroom, which may go undetected if only the initial or final moments are considered.

On the other hand, in [9 and 10], input parameters such as student IDs or the requirement for the teacher to capture an image or video before the class starts are needed, which makes the process less automated. Additionally, the training data used in these approaches was collected in controlled environments.

Requiring input parameters or manual intervention from the teacher can introduce additional steps and dependencies that may not be ideal for a fully automated system. The need for controlled environments during the training phase may limit the system's performance when applied in real-world, uncontrolled settings.

In contrast, the proposed approach aims to achieve a more seamless and automated process by analyzing the video data directly without relying on manual inputs or pre-captured images. This allows for a more efficient and practical implementation, especially in scenarios where real-time monitoring or continuous surveillance is required.

Another important point is that the training data used in the proposed approach was custom-made, and a pre-trained network was not implemented. By training both networks using custom data, excellent results were achieved, surpassing 98% accuracy in each network.

5. CONCLUSION

Based on the analysis conducted, it can be inferred that the combination of approaches such as YOLOv5 and VGGFace in facial recognition within educational environments has the potential to improve the accuracy and effectiveness of traditional attendance systems. Additionally, implementing periodic image analysis throughout the entire class proves to be an effective strategy for continuous monitoring and timely detection of relevant events. Moreover, customizing the training data and training from scratch allows for the adaptation of models to the specific characteristics of the educational setting, enhancing the accuracy of the obtained results.

It was observed that YOLOv5 has the potential to perform the task on its own, but further experimentation would be required since it is primarily designed for object detection rather than facial recognition.

The system is functional and can be implemented for other activities, such as tracking the entry and exit of teachers from the parking lot, detecting unauthorized individuals in specific buildings or universities, and monitoring access control. However, it is worth considering that the system is currently being run on a computer, and it would be more practical and cost-effective to utilize an industrial computer or a Raspberry Pi to make it more portable and accessible.

By exploring these potential applications and considering the hardware requirements, the system can be further optimized and tailored to specific use cases, providing enhanced security and automation in various educational settings.

6. ACKNOWLEDGMENTS

Thanks to the Intel laboratory at the Technological University of Queretaro for providing the necessary hardware for carrying out the project. On the other hand, thanks to the students who were part of the project, ensuring that everything complies with the university's terms and conditions.

7. REFERENCES

[1] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, 曾逸夫(Zeng Yifu), Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, … Mrinal Jain. (2022). ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime InstanceSegmentation (v7.0). Zenodo. https://doi.org/10.5281/ zenodo.7347926.

[2] Parkhi, O., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. BMVC 2015 - Proceedings of the British Machine Vision Conference 2015, 1–12.

[3] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[4] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning", 23 February 2016. https://doi.org/10.48550/ arXiv.1602.07261.

[5] Karen Simonyan, Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition", 4 September 2014 https://doi.org/10.48550/arXiv.1409.1556.

[6] M. Sajid, R. Hussain and M. Usman, "A conceptual model for automated attendance marking system using facial recognition," Ninth International Conference on Digital Information Management (ICDIM 2014), Phitsanulok, Thailand, 2014, pp. 7-10, doi: 10.1109/ ICDIM.2014.6991407.

[7] T. A. Kiran, N. D. K. Reddy, A. I. Ninan, P. Krishnan, D. J. Aravindhar and A. Geetha, "PCA based Facial Recognition for Attendance System," 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2020, pp. 248-252, doi: 10.1109/ICOSEC49089.2020.9215326.

[8] J. W. S. D'Souza, S. Jothi and A. Chandrasekar, "Automated Attendance Marking and Management System by Facial Recognition Using Histogram," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 66-69, doi: 10.1109/ ICACCS.2019.8728399.

[9] Class Attendance Management System using Facial Recognition. Clyde Gomes, Sagar Chanchal, Tanmay Desai and Dipti Jadhav. ITM Web Conf., 32 (2020) 02001. DOI: https://doi.org/10.1051/itmconf/20203202001.

[10] A. R. S. Siswanto, A. S. Nugroho and M. Galinium, "Implementation of face recognition algorithm for biometrics based time attendance system," 2014 International Conference on ICT For Smart Society (ICISS), Bandung, Indonesia, 2014, pp. 149-154, doi: 10.1109/ICTSS.2014.7013165.

[11] Face recognition Attendance system using HOG and CNN algorithm. Aditya Kapse, Tejas Kamble, Ashutosh Lohar, Shubham Chaudhari and Digambar Puri. ITM Web Conf., 44 (2022) 03028. DOI: https:// doi.org/10.1051/itmconf/20224403028.

[12] E. Winarno, I. Husni Al Amin, H. Februariyanti, P. W. Adi, W. Hadikurniawati and M. T. Anwar, "Attendance System Based on Face Recognition System Using CNN-PCA Method and Real-time Camera," 2019 International Seminar on Research of Information Technology and

Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 2019, pp. 301-304, doi: 10.1109/ISRITI48646.2019.9034596.

[13] R. C. Damale and B. V. Pathak, "Face Recognition Based Attendance System Using Machine Learning Algorithms," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 414-419, doi: 10.1109/ICCONS.2018.8662938.

[14] Intelligent Attendance System with Face Recognition using the Deep Convolutional Neural Network Method. Nurkhamid, Pradana Setialana, Handaru Jati, Ratna Wardani, Yuniar Indrihapsari and Norita Md Norwawi. Journal of Physics: Conference Series,Volume 1737, 3rd International Conference on Electrical, Electronics, Informatics, and Vocational Education 5 October 2020, City, Indonesia.

[15] Khan, S, Akram, A. & Usman, N. Real Time Automatic Attendance System for Face Recognition Using Face API and OpenCV. Wireless Pers Commun 113, 469–480 (2020). https://doi.org/10.1007/ s11277-020-07224-2.

[16] Q. Cao, L. Shen, W. Xie, O. M. Parkhi and A. Zisserman, "VGGFace2: A Dataset for Recognizing Faces across Pose and Age," 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an, China, 2018, pp. 67-74, doi: 10.1109/FG.2018.00020.

[17] Ijraset. (s. f.). Online Classroom Attendance Marking System Using Face Recognition, Python, Computer Vision, and Digital Image Processing. IJRASET. https://www.ijraset.com/research-paper/onlineclassroom-attendance-marking-system-using-facerecognition.

[18] S. Lukas, A. R. Mitra, R. I. Desanti and D. Krisnadi, "Student attendance system in classroom using face recognition technique," 2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea (South), 2016, pp. 1032-1035, doi: 10.1109/ICTC.2016.7763360.

[19] Hugo, D. R. V. (2016, 9 diciembre). Sistema de reconocimiento facial utilizando filtros de correlación. http://tesis.ipn.mx/handle/123456789/20333.

[20] Padilla Cando, A. L., & Sánchez Pilay, J. Y. (2013-12). Sistema de reconocimiento facial utilizando filtros de correlación. Recuperado a partir de http://repositorio.ug.edu.ec/handle/redug/48932.