Veton Këpuska, et. al. International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 12, Issue 10, October 2022, pp. 188-199

RESEARCH ARTICLE OPEN ACCESS

A Voice and Facial Recognition System to Protect Students from Being Forgotten Inside School Buses in the Kingdom of Saudi Arabia.

Veton Këpuska[1], Alferaih Ibrahim[2]
[1,2](Electrical& Computer Engineering Department, Florida Institute of Technology,Melbourne,FL,USA

ABSTRACT
In this research, we discuss the situation of students in the three to eight year old age group. Every day in the Kingdom of Saudi Arabia during school days, more than 1,000,000 students are transported throughout the country by more than 15,000 school buses. Everything is dependent on the human element. During school transportation around the world, many children are forgotten inside the school bus and with many of them eventually going missing or even dying. The bus driver is usually busy and tired due to the focus on the road, which may make him lose focus on those on the bus. Therefore, we believe that modern technological development must intervene in order to protect children from this problem. In our research, we find a solution to the problem of children being forgotten inside the school bus, mostly due to falling asleep, and being exposed to problems that may cause them harm or death. Our solution to this problem is to verify school children get on and off the school bus safely, through the use of facial recognition and speaker recognition technologies.

Date of Submission: 13-10-2022 Date of Acceptance: 28-10-2022

I. INTRODUCTION

In this proposal, we address a global problem that occurs within the school bus. Over the past several years, there have been many incidents of students dying inside the school bus due to being forgotten for several hours.[1] In other incidents, students have been left locked inside the us for several hours and found severely dehydrated [2]. Other students either did not make it home or did not make it to school[3]. Several scenarios may occur in the school bus:

- The driver may be tired and not check the bus after students have left.
- There might be some bad bus drivers who forgot the student on the bus and hide the student and tell the police that the student did not board the bus.
- The student may be kidnapped by the bus driver. and the driver informed the police that he had not seen the student.

Our solution to this problem is verify students getting on and off the school bus through the use of facial recognition as well as voice recognition technologies.

In our daily lives, both facial recognition and voice recognition are frequently used.

For example, we find that facial recognition is one of the features of phones and tables to unlock the entry lock of the device. We also find that many governments are using facial recognition when registering passports, matching drivers to car plates, or entering some secure jobs sites. We also find many services like banks, hospitals, and some companies use voice recognition by identifying the gender of the speaker. Both organizations and individuals find this technology interesting because it has immense potential from a commercial perspective. Today, voice recognition has applications in areas like workplace management, banking, marketing and healthcare.

The core concept of voice recognition is to distinguish words and phrases from incoming audio from the user. In the workplace, this technology can be used for creating tables and graphs via voice commands, and schedule and start video conferencing via relevant apps. Similarly, in banking, users can use voice recognition systems to fetch records and perform bank transactions. In the marketing sector, data analysis based on speech can lead to better marketing opportunities. In the healthcare sector, voice recognition systems can be used to analyze queries related to diseases and thus it can improve the overall workload of healthcare providers. Therefore, in this research, we will define and explain how the face and voice are recognized.In our system, we will solve the

problem of those who get on or off the bus by recognizing the voice and face of a student. Because the Features of a student's voice cannot match with any other person's voice, and also the face cannot match the Features of a student's face with any other person's face.

II. RELATED WORKS.

Proposed by Anwar al-Lawati, et.al. A system that works on bus entry and check-in via RFID [4]

This proposed system works on RFID tags and readers to authenticate the student

This system may read any card that passes by it, whether students assigned to the bus or students of other buses, which means that information interferes, which weakens the quality of the system.Proposed by Y. Mori, H. Kojima, E. Kohno, S. Inoue, T. Ohta, Y. Kakuda, et al., "A Self-Configurable New Generation Children Tracking System Based on Mobile Ad Hoc Networks Consisting of Android Mobile Terminals"[5]

This system is expensive and requires specific hardware and components, such as KidTrack. It serves to represent the biometrics as a unique pattern in a student's vein. While getting on or off the bus, the student wipes the palm of his hand with a handheld reader installed on the bus, then sends the student's arrival and departure times to a cloud-based server. On this server, the administration can find the bus information about where and when the child was tracked and where the bus was at that time. Data is also saved locally to the drive if the drive cannot access the cloud. One of the disadvantages of this system is the inability of the child to place the palm of the hand as needed. And the costs are high.

Proposed by Amruta M Sanam. S. D. Sawant

[6]facial recognition system monitors for student ride and drop off times. Then it sends information for each child to the school's database, through which it sends a message to the child's parents. Among the disadvantages of this system, the system may not recognize the students if they wears a face mask, as is the case with (Covid-19).

Face Recognition

There are a number of face recognition systems available including open source and state of the art commercial systems such as Facebook'sDeepFace and Google's FaceNet. However, there is a significant difference in the accuracy of both systems due to several factors including the large number of images in the private datasets for commercial systems. Moreover, they have trained their networks offline; therefore there is space for use of a large number of parameters in the network to be computationally viable with available hardware resources.

III. Open Face Model

OpenFace[7] is an open-source face recognition library using Python and Torch based on deep neural networks. The main focus of the method is to make it computationally fast to be useable for mobile applications in real time face recognition applications. It is based on a similar pipeline used by Google's FaceNet architecture including four steps i.e. face detection, preprocessing to normalize and fix the size of face image, use of a deep neural network as a feature extractor to learn low dimensional representations of face images, and a classifier at the end to recognize the face. Moreover, use of PyTorch in its implementation makes it suitable to run on a central processing unit (CPU) or graphics processing unit (GPU) using CUDA. It can also be utilized for face recognition in Python using Keras and Tensorflow. Moreover, all these libraries allow the use of face recognition, tracking, and clustering applications in video .

Features

OpenFace is a lightweight face recognition library in Python and Torch with a number of salient features making it suitable for use in a number of applications. Here is a list of salient features of OpenFace library making it easier to choose for face recognition applications:

- OpenFace uses a modified smaller neural network suitable for smaller datasets. Therefore, it can be used for face recognition applications, even if the number of available images in the order of thousands as compared to millions required for DeepFace and FaceNet methods.
- It is lightweight and uses a minimalist approach for face recognition. Therefore, it can be used if there are no high-end hardware resources to train a larger neural network.
- On labelled faces in the wild (LFW) benchmark dataset, it has an accuracy of 92.92%. That is very good compared to other methods like Eigenfaces, Eisherfaces and LBPH[7].
- OpenFace library has pre-trained weights available for different neural networks architectures. Therefore, no training is needed for the face recognition application and it will save training considerable time. One simply needs to input test image and it will present its representation on a 128 dimensional hypersphere after passing through deep neural network.

- It uses Euclidean distance between the 128-dimensional hypersphere representations of two images as similarity index. Higher distance means the two images are of different people and a smaller distance means the images are of the same person. One can compute the threshold for the application that gives highest accuracy. Therefore, it can be used for any similarity detection, clustering and classification tasks in the field of face recognition.
- It gives freedom to use any classifier, or machine learning (ML) algorithm for the task after converting the input image into 128-dimensional feature space through pre-trained deep neural network.
- It has an average prediction time of 58.9 ms on CPU and 13.72ms on a Tesla K40 GPU. That is considerably faster than its competitors, when used in mobile applications involving face recognition or classification tasks [9].
- It has 3.7 million tunable parameters as compared to VGG-Face's 145 million parameters and 22.7 million parameters in FaceNet. Therefore, its weights have a size of 14MB as compared to the 566MB and 90MB for VGG-Face and FaceNet respectively [10].
- It has Keras and TensorFlow implementations available. Therefore, it can be used for the task using these Python libraries for deep learning [11].

Application

Due to lightweight and speed, OpenFace can be used in mobile applications and other security and surveillance solutions. A home surveillance system with facial recognition using OpenFace is available and has classification accuracy of 78.39% on a limited dataset [12]. Pyannotate-video [13] is a toolkit in Python for face detection, tracking and clustering in videos that use OpenFace. It can also be used in office attendance management system. These are just few applications but OpenFace can be used for any face recognition techniques if speed is the primary concern over accuracy.

IV. Arc Face Model

ArcFace[14] is a face recognition algorithm used by InsightFace[15] face analysis project that uses an additive angular margin loss function instead of regular softmax loss. The face recognition pipeline includes face detection, alignment, low dimensional representation or embedding and verification. ArcFace is an implementation of the representation step. Originally, it was implemented in MXNet and Python. However, its Keras implementation is also available. Its modified version, known as Sub-center ArcFace, is also available. That is more robust, to perform face recognition tasks on noisy image datasets without accurate annotations. It has achieved comparable performance with available state-of-the-art methods and not computationally expensive despite tested on a dataset of millions of images.

Features:

- Original ArcFace model implemented using MXNet and Python has a verification accuracy of 99.83% on LFW dataset, while its Keras re-implementation has an accuracy of 99.40%. This means it is highly accurate and also easily reproducible with good results [17].
- Its pre-trained convolutional neural network (CNN) weights are of a size of 133MB [17] and they can be used for face recognition tasks without need to re-train the network. Simply feed the input image to get its 512 D feature representation or embedding and apply any of distance metric to verify the image.
- ArcFace model is available in Python's Deepface library and it can be built and run with a few lines of code [18].
- It is also part of the open-source face analysis tool InsightFace implemented in MXNet and Pytorch and is free available.
- It takes 8.9 ms/face for one forward pass of ResNet50 CNN and 15.4 ms/face for ResNet100 to extract the 512-dimensional feature vector while implemented on a NVIDIA Tesla P40 (24GB) GPU [14].
- It also has features like face image generation from feature vectors and can be used in applications like identity preserved face image generation.
- Sub-Center ArcFace can be used to clean a noisy dataset by clustering the data into multiple subclasses, with one dominant class containing clean images and multiple non dominant subclasses ,with noisy data that can be easily removed.

Applications:

ArcFace is a face recognition algorithm with good accuracy results. Therefore, it can be used for face recognition tasks including verification, classification and similarity detection. Moreover, it can be used for face recognition in videos. It can also be usedto get low dimensional representation or embedding of an input face image that can be used as feature vector for any ML algorithm for face recognition task.

Comparison between OpenFace and ArcFace

Feature	OpenFace	ArcFace
Verification Accuracy	92.92% on LFW dataset	99.83% on LFW dataset
Prediction Time	13.72ms (GPU) (Tesla K40) 58.9ms(CPU)	8.9ms for ResNet50 and 15.4ms for ResNet100 on GPU (NVIDIA P40)
Embedding Size	128 dimensional	512 dimensional
Weights' Size	14MB	133-250 MB [14][17]
Implementation	Python, Torch, Keras, TensorFLow	Python, MTXNet, Keras
Availability	OpenFace, Deepface Python library	InsightFace face analysis tool, Deepface python library
Applications	Face recognition tasks including Clustering, Classification and Similarity Detection for images and videos Suitable for mobile devices with real time face recognition To get embedding for face images	Face recognition tasks for images and videos requiring high accuracy Identity preserved image generation tasks Cleaning of noisy image datasets To get feature representation of face images

Both Open Face and ArcFace are face recognition tools with good accuracies. OpenFace is a lightweight solution with speed more suitable for mobile applications or where real time face recognition is required. On the other hand, ArcFace is more accurate with increased dimensionality of embedding or feature representation. Moreover, it is more robust for use on noisy datasets. We can choose either one of them according to intended applications.

Voice Recognition for X- Vectors and Ecapa.

1. X-Vectors

A speaker recognition system consists of two steps including speaker identification to find the identity of speaker and speaker verification matching the identified speaker with its voice samples. The next step in the process, built on top of these steps, is speaker diarization, which splits the parts of an audio belonging to different speakers or 'who spoke when'.

X-Vector [19][20] is basically a deep neural network (DNN) based method, used to achieve better speaker representations or embedding from audio input signal. Therefore, it has applications in the field of speaker recognition and diarization with improved performance. It is a part of an open-source speech recognition toolkit based on PyTorch that is known as SpeechBrain[21]. It is also part of the second version of the Kaldi speech recognition toolkit [22], and is used to replace i-Vectors when extracting embeddings [23].

Features and Applications:

X-Vectors is a successful TDNN based algorithm that is useful to extract embeddings from any audio signal and also has applications in speaker recognition and diarization applications. Here is a brief overview of its features making it desirable for speech recognition systems:

- It is part of PyTorch based speech recognition toolkit SpeechBrain[21] and it can be directly used for speaker recognition and diarization tasks.
- It is available in Kaldi toolkit to extract embeddings and has replaced i-Vector in its second version. Therefore, it can be used from there for speech recognition tasks [23].
- It is also part of end-to-end speech processing toolkit (ESPnet) toolkit based on PyTorch as a deep learning engine [28].
- X-Vectors are more robust to language mismatch during training and evaluation mismatch and it can be used for the speaker recognition tasks irrespective of language of the speaker.
- Pre-trained model of X-Vectors is available with 4.2 million parameters and size of 31 MB that is much lesser than i-Vectors based model having size of 516MB. This pre-trained model can be used to get 512-dimensional embeddings for a dataset to implement the speaker recognition and diarization tasks [29].
- It is a next generation speaker recognition system requiring only the speaker labels for training the TDNN as compared to i-Vectors requiring a lot of transcribed data.
- It has won the NIST speaker recognition challenge and DIHARD challenge depicting its superior performance over earlier methods.
- A variable length audio input can be passed to extract feature representations or X-Vectors of fixed length using this method that is desirable feature for diarization tasks.
- ESPnet-TTS is an end-to-end text to speech recognition toolkit that uses pre-trained X-Vector as speaker embeddings [30].

2. ECAPA

Emphasized Channel Attention, Propagation and Aggregation in a Time Delay Neural Network based Speaker Recognition System (ECAPA-TDNN) [31] is a popular speaker recognition and diarization model build using deep learning (DL). It uses extended TDNN X-Vector architecture as a baseline system and has added modifications to the neural network architecture and statistical pooling layer to improve its performance. It has two major improvements in the baseline system. The first one is by adding skip connections for propagation and aggregation of the channels throughout the system. The second one is incorporation of channel attention in global context in frame layers and statistical pooling layer. Therefore, it has improved performance over the baseline X-Vector system.

It is available as a part of SpeechBrain toolkit [21] having several speech processing algorithms combined in a single library. It is a recently popular technique used to get robust speaker embeddings. Other researchers have proposed modifications and enhancements in the original system to achieve better results.

Features & Applications:

ECAPA-TDNN is an improved version of TDNN architecture used for X-Vector embeddings. It is the recent state of the art system and has applications in speaker recognition tasks including identification and verification and also speaker diarization tasks. Here is a list of salient features of ECAPA-TDNN system and its brief description:

- ECAPA-TDNN is available as a part of an open-source speech processing toolkit SpeechBrain[21]based on PyTorch. It can be directly used for speaker recognition and diarization tasks without need for re-implementation.
- It has outperformed available speaker recognition and diarization models with lower percentage of EER and DER. It has shown best performance in the text independent task of the short duration speaker verification challenge 2020.
- It remained top scoring in IDLAB VoxCeleb Speaker Recognition Challenge 2020 for supervised and unsupervised speaker verification tasks, with an EER of 2.16% for the closed track on validation set [34].
- It can be used to extract the 192-dimensional embeddings from input audio signal. By using any similarity metric including inner product or cosine distance, one can employ any ML method to develop a speaker recognition system.
- A spoken language recognition model trained using ECAPA-TDNN model is available with more fully connected layers and cross entropy loss function [35]. Moreover, it is trained on VoxLingua107 dataset using SpeechBrain library. It can classify a speech utterance according to the spoken language for 107 different languages. It can be used as spoken language recognition system or get embeddings to use for any language identification model.
- Sheikh et al. [36] used the speech embeddings extracted by pre-trained ECAPA-TDNN model for stuttering detection and it had shown overall improvement of 16.74% in accuracy over baseline system using SEP-28k dataset.
- Xue et al. [37] proposed a method for multi-speaker text to speech synthesis and it used ECAPA-TDNN as a speaker encoder to get high quality speech with better similarity and naturalness for both seen and unseen speakers. Their proposed method outperformed X-Vector and other DL based encoders.
- Chen et al. [38] proposed a method for automatic speaker verification by pre-training a deep neural network (DNN) on unlabeled data and feeding representations from all its hidden layers as input to the ECAPA-TDNN. It is known as Self Supervised System (SSL). Their best system outperformed the winner of VoxCeleb Speaker Recognition Challenge 2021.

Comparison between X-Vector and ECAPA-TDNN

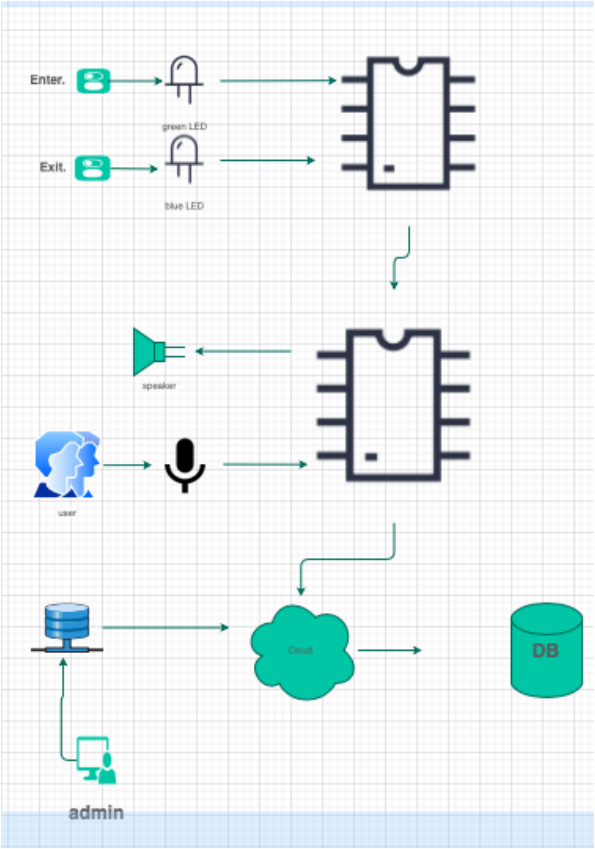
Feature	**X-Vector**	**ECAPA-TDNN**
Equal Error Rate (%) (EER)	1.26 for Extended-TDNN (large) for VoxCeleb1 test set [31]	0.87 for ECAPA-TDNN (large) on VoxCeleb1 test set [31]
Minimum Normalized Detection Cost (minDCF) with $P=10^{-2}$	0.1399 for Extended-TDNN (large) for VoxCeleb1 test set [31]	0.1066 for ECAPA-TDNN (large) on VoxCeleb1 test set [31]
Input Features	24-dimensional MFCCs [20]	80-dimensional MFCCs [31]
Embedding Size	512-dimensional [20]	192-dimensional [31]
Number of Parameters	20.4M for Extended-TDNN (large) [31]	14.7M for ECAPA-TDNN (large) [31]
Implementation	Python, PyTorch, C++	Python, PyTorch
Availability	SpeechBrain, Kaldi's speech processing toolkit, ESPnet, ESPnet-TTS toolkit	SpeechBrain

Applications	Speech recognition tasks including speaker identification, verification and diarization Text to speech system, To get embedding for audio inputs	Speech recognition tasks including speaker identification, verification and diarization Text to speech system, To get robust feature representation of utterances/audio inputs Spoken language recognition applications

Both X-Vector and ECAPA-TDNN are deep learning-based state of the art systems suitable for speech recognition and diarization tasks. Moreover, both have vast range of applications for their use to extract embeddings from audio input signals. X-Vector is a baseline system having availability in multiple open source speech processing toolkits while ECAPA-TDNN is an enhanced and improved version of the TDNN architecture having better performance in almost every aspect of speech recognition and diarization system. Any one of them can be used according to one's dataset and intended application.

V. Methodology:

The system will be operated by the school bus administration. It provides an information screen depicting the inside of the bus. When the student gets on the bus, he must press the button of the system installed in the passage between the seats. When the student steps on the bus, that event will be detected and indicated by a green LED light. The student must identify themself verbally with their name and ID number. This is captured by a microphone installed at the entrance. The system recognizes the voice and the number, and logs in the event with a timestamp. The system immediately confirms the student by matching the face already registered in the database with the name of the student. When getting off the bus, the student must press the switch to turn on the system installed in the passage exit. Then the blue LED light turns on, and the student must provide his student ID number to the installed microphone. The system identifies the student's voice and number, then records the student's exit. The student's exit is recorded by logging the time and date of his/her exit with the face registered. This way there is a record of students entering and exiting their bus to be used as necessary by School Transport Administration.

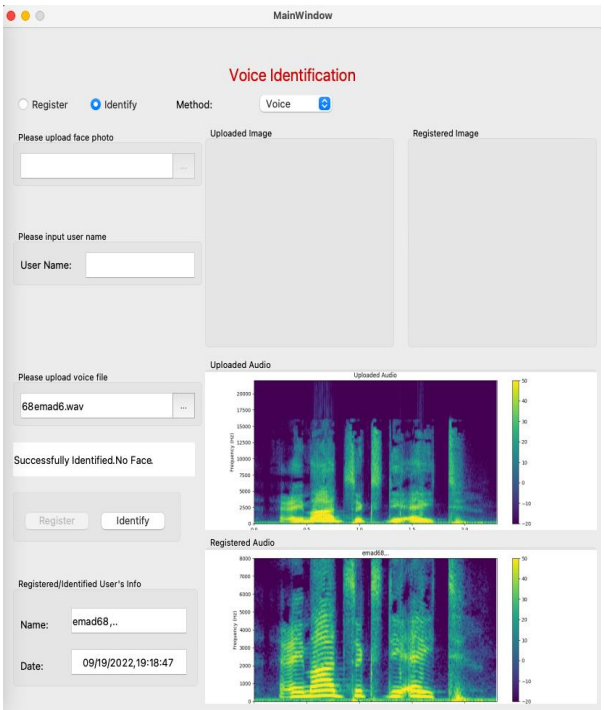


In this paper we will look at how voice is recognized and analyzed in Python programming language.Python programming is a powerful language used in artificial intelligence and machine learning and there are a number of libraries available for facial and speaker recognition. There are various libraries that are used for speaker recognition, but in this paper we will use SpeechBrain because this package in Python is appropriate in terms of the data we have, which is related to children. This library supports voice analysis tools. DeepFace is a Python package that supports face analysis tools. It is open source and the flexibility and efficiency of the face recognition package makes it an excellent choice for our work on student face recognition.

In this research we built a system that works on voice recognition and face recognition combined. We identified the voice by converting the voice files to wav files, then storing them in the system in the form of a database, and then we uploaded a new voice file and checked that the voice file matches the voice files in the database. If it is identical with any of the samples, it will give a successful verification result. But if it does not find a match, it produces the result that this voice is not registered in the system.As for facial recognition, we have converted the images as (jpeg) and stored

them in the database. When a new image is uploaded it is compared with the images in the data base. If a match is verified, then it gives the name of the student and the matching image. If it does not match, the result will be not registered in the system.In the beginning, we made the main window of the user interface system using pyQt5 by Python. Thenwe added an OpenCV package for Python to open source and a shortcut to access the database, and cv2 to read the image and its color. We used the pickle package to store the data in a digital form and to report if there is a picture of a face. We used the Retinaface to check whether there is no face in the photo, or if there is more than one face, as this would make the facial identification ambiguous. The criterion we used for the voice matching threshold is that the highest match must be above 70%. And the face recognition threshold criterion is that the highest the difference between the faces can be is 20%. We also normalized the audio to make the audio files all have the same amplitude and not dependent on volume. We set a limit on the length of the audio clip so that it must be greater than a second and less than twenty seconds. We used facial landmarks to determine the face points in order to identify the least difference or similarity between the sample and the data.We used open-source libraries to apply our experiments to the database that we prepared.

Voice recognition by pronouncing the name and number.



VI. Data

In our database we communicated with the school and explained to them the importance of this project: the safety of the students who ride on the bus. The time from getting on the bus to getting off to school may take more than 45 minutes in some cases.

We collected the sample from 52 students, consisting of 21 girls and 31 boys in grades from kindergarten to the fourth grade.

We took pictures of the students in more than one position, such as from afar, from close to one side, at an angle of approximately 60 degrees, or directly at an angle of 90 degrees. The degree of illumination varied from one image to another. Students were also asked to smile for the camera or stand without expression.

We also took audio samples by recording the voice of each student. We assigned a number to each student from 20 to 71. We asked each student to pronounce the number several times. Then we asked each student to pronounce his first name with the number assigned to him a number of times.

The recording was made inside the school environment. The doors were often opened and closed, the students walked, talked to each other or laughed. On the other hand, they might be too shy to talk to us, so we had to encourage them to speak in a clear voice. However, we made sure that the recording was clear and clean.

VII. Results

The sample used numbered 52 students. The results of the test are shown below:

1. Face and voice recognition systems used: (ArcFace) and (ECAPA).

The system accuracy was 96% for voice recognition using ECAPA.
Accuracy was 96% for face recognition using ArcFace.

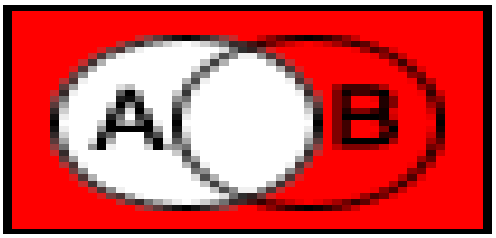
Where is A = face recognition and B = voice recognition,

1- Probability of A NOT occurring: **P(A')**

P(A') = 1 - P(A)

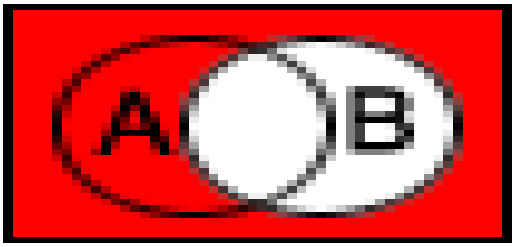
= 1 - 0.96

= 0.04



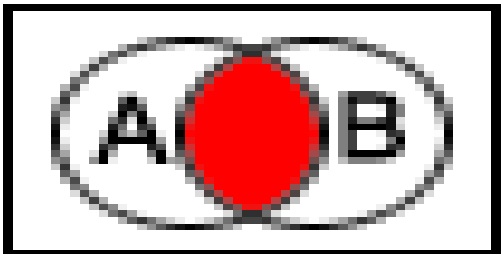
2- Probability of B NOT occurring: **P(B')**

P(B') = 1 - P(B)
= 1 - .96
= 0.04



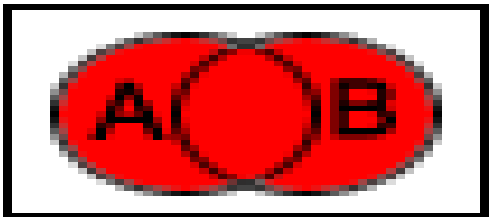
3- Probability of A and B both occurring: **P(A∩B)**

P(A∩B) = P(A) × P(B)
= 0.96 × 0.96
= 0.9216



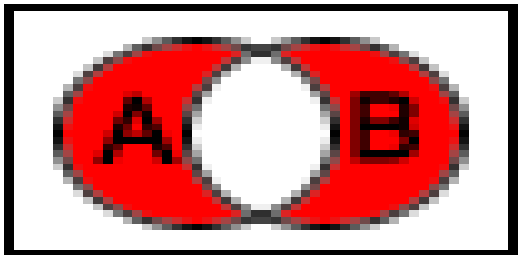
4- Probability that A or B or both occur: **P(A∪B)**

P(A∪B) = P(A) + P(B) - P(A∩B)
= 0.96 + 0.96 - 0.9216
= 0.9984



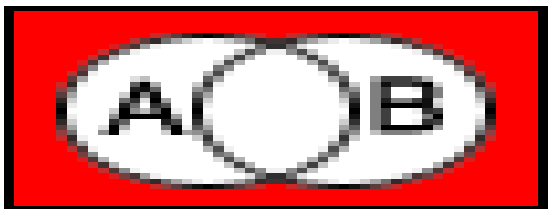
5- Probability that A or B occurs but NOT both: **P(AΔB)**

P(AΔB) = P(A) + P(B) - 2P(A∩B)
= 0.96 + 0.96 - 2×0.9216
= 0.0768



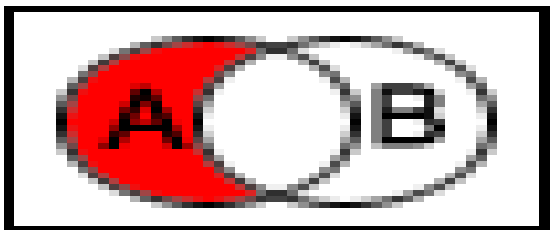
6- Probability of neither A nor B occurring: **P((A∪B)')**

P((A∪B)') = 1 - P(A∪B)
= 1 - 0.9984
= 0.0016



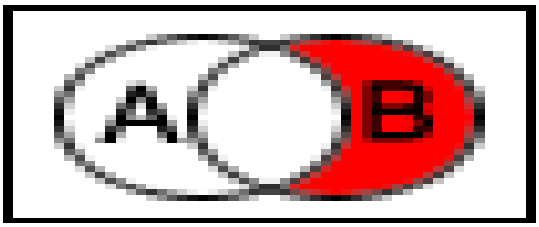
7- Probability of A occurring but NOT B

P(A occur but NOT B) = P(A) × (1- P(B))
= 0.96 × (1 - 0.96)
= 0.0384



8- Probability of B occurring but NOT A

P(B occur but NOT A) = (1 - P(A)) × P(B)
= (1 - 0.96) × 0.96
= 0.0384



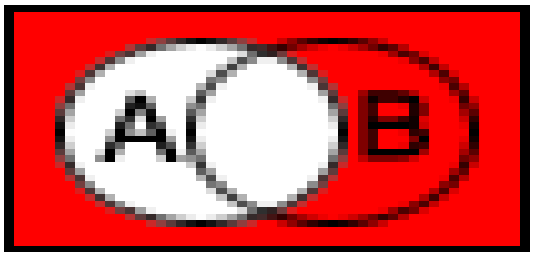
1. Face and Voice Recognition: (OpenFace) and (Xvectors).

The system had an accuracy of 94% for voice recognition while using X-vectors.on our data.
It also had an accuracy of 88% for face recognition while usingOpenFace on our data.

Where is A = face is recognition and B = voice recognition,

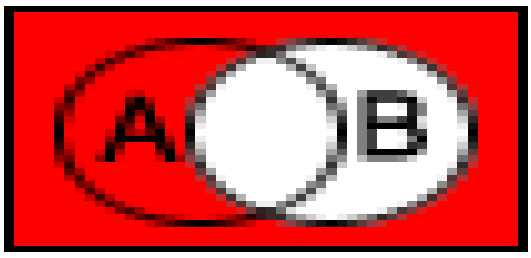
1- Probability of A NOT occurring: **P(A')**

P(A') = 1 - P(A)
= 1 - 0.88
= 0.12



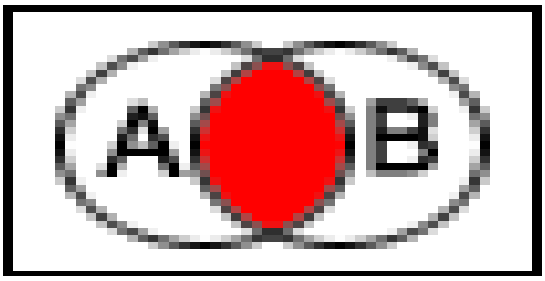
2- Probability of B NOT occurring: **P(B')**

P(B') = 1 - P(B)
= 1 - 0.94
= 0.06



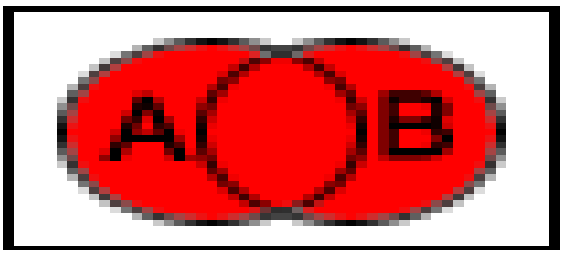
3- Probability of A and B both occurring: **P(A∩B)**

P(A∩B) = P(A) × P(B)
= 0.88 × 0.94
= 0.8272



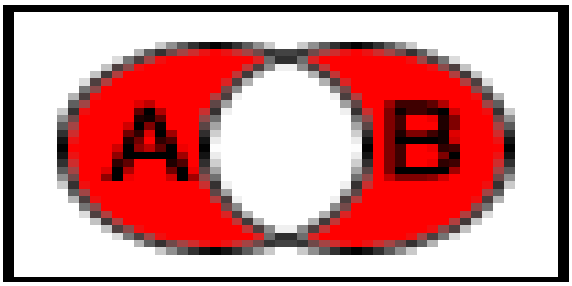
4- Probability that A or B or both occur: **P(A∪B)**

P(A∪B) = P(A) + P(B) - P(A∩B)
= 0.88 + 0.94 - 0.8272
= 0.9928



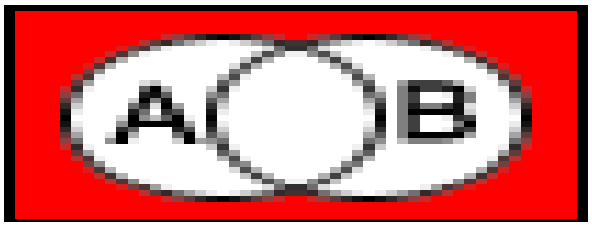
5- Probability that A or B occurs but NOT both: **P(AΔB)**

P(AΔB) = P(A) + P(B) - 2P(A∩B)
= 0.88 + 0.94 - 2×0.8272
= 0.1656



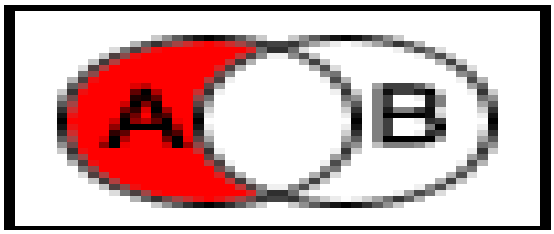
6- Probability of neither A nor B occurring: **P((A∪B)')**

P((A∪B)') = 1 - P(A∪B)
= 1 - 0.9928
= 0.0072



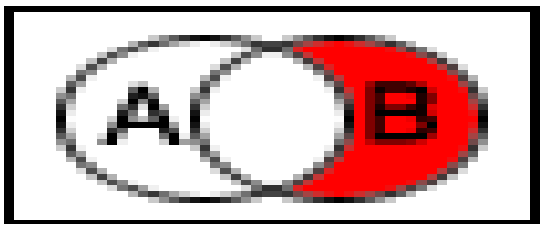
7- Probability of A occurring but NOT B

P(A occur but NOT B) = P(A) × (1- P(B))
= 0.88 × (1 - 0.94)
= 0.0528



8- Probability of B occurring but NOT A

P(B occur but NOT A) = (1 - P(A)) × P(B)
= (1 - 0.88) × 0.94
= 0.1128



VIII. Conclusion

Through the results that we obtained through our database, we find that the facial recognition system gave better results when we used the Arcface model, where it was 96%. With regard to voice recognition, the results of the ECAPA model were better, as they were 96% as well. It is worth mentioning that the idea of this system is to recognize both the voice and the face, or at least one of them. Our system gives better results when combining both models, as the system gave 100% recognition of the student when attempting to recognize either the face or his voice. This is of importance. For example, the environment surrounding a student or a student's illness, like a cold, may affect their voice, which hinders their speaker recognition, but the presence of a facial recognition system will identify the student. Likewise, if there is a problem in the student's facial pose, illumination, or expression the facial recognition model may have difficulty in recognizing the student's face, but the speaker recognition system could still function optimally.

With this system, we will solve a troubling problem for students' parents, school transport and school administration .The goal of school transportation is to deliver students from home to school and backsafely.

We have studied the facial and voice recognition on the basis of our data, with a population consisting of 52 children. In the future, we will add the feature of gender recognition through voice and face to determine the gender of the student to increase the success of the system tracking students oon and off the school bus safely.

References

[1]. 4-Year-Old Boy Dies After Being Left Inside Hot School Bus Outside Kindergarten, 2019.
[2]. 4-year-old faints after being 'forgotten' on UAE school bus, 2018.
[3]. Driver indicted exactly 1 year after 3-year-old boy died from being left on scorching hot bus: DA, 2019.
[4]. S. A.-J. A. A.-B. D. A.-A. M. A. a. D. A.-A. A. Al-Lawati, "RFID-based system for school children transportation safety enhancement," in IEEE 8th GCC Conference & Exhibition, Muscat, Oman, 2015.
[5]. Y. Mori and H. Kojima, "A Self-Configurable New Generation Children Tracking System Based on Mobile Ad Hoc Networks Consisting of Android Mobile Terminals," ieee, p. 18, 2011.
[6]. A. M. Sanam and S. D. Sawant, "Safety system for school children transportation," in International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2016.
[7]. Baltrušaitis, Tadas, Peter Robinson, and Louis-Philippe Morency. "Openface: an open source facial behavior analysis toolkit." *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016.
[8]. Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
[9]. http://cmusatyalab.github.io/openface/models-and-accuracies/
[10]. https://sefiks.com/2019/07/21/face-recognition-with-openface-in-keras/

[11]. https://cmusatyalab.github.io/openface/
[12]. https://github.com/BrandonJoffe/home_surveillance
[13]. https://github.com/pyannote/pyannote-video
[14]. Deng, Jiankang, et al. "Arcface: Additive angular margin loss for deep face recognition." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.
[15]. https://github.com/deepinsight/insightface
[16]. Deng, Jiankang, et al. "Sub-center arcface: Boosting face recognition by large-scale noisy web faces." *European Conference on Computer Vision*. Springer, Cham, 2020.
[17]. https://sefiks.com/2020/12/14/deep-face-recognition-with-arcface-in-keras-and-python/
[18]. https://github.com/serengil/deepface
[19]. Snyder, D., Garcia-Romero, D., Povey, D. and Khudanpur, S., 2017, August. Deep neural network embeddings for text-independent speaker verification. In Interspeech (Vol. 2017, pp. 999-1003). Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
[20]. Snyder, D., Garcia-Romero, D., Sell, G., Povey, D. and Khudanpur, S., 2018, April. X-vectors: Robust dnn embeddings for speaker recognition. In 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 5329-5333). IEEE.
[21]. Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J. and Chou, J.C., 2021. SpeechBrain: A general-purpose speech toolkit. arXiv preprint arXiv:2106.04624.
[22]. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P. and Silovsky, J., 2011. The Kaldi speech recognition toolkit. In IEEE 2011 workshop on automatic speech recognition and understanding (No. CONF). IEEE Signal Processing Society.
[23]. https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2
[24]. Snyder, D., Garcia-Romero, D., Sell, G., McCree, A., Povey, D. and Khudanpur, S., 2019, May. Speaker recognition for multi-speaker conversations using x-vectors. In ICASSP 2019-2019 IEEE International conference on acoustics, speech and signal processing (ICASSP) (pp. 5796-5800). IEEE.
[25]. Landini, F., Profant, J., Diez, M. and Burget, L., 2022. Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: theory, implementation and analysis on standard tasks. Computer Speech & Language, 71, p.101254.
[26]. Tripathi, M., Singh, D. and Susan, S., 2020, October. Speaker recognition using SincNet and X-vector fusion. In International Conference on Artificial Intelligence and Soft Computing (pp. 252-260). Springer, Cham.
[27]. Yang, S.W., Chi, P.H., Chuang, Y.S., Lai, C.I.J., Lakhotia, K., Lin, Y.Y., Liu, A.T., Shi, J., Chang, X., Lin, G.T. and Huang, T.H., 2021. Superb: Speech processing universal performance benchmark. arXiv preprint arXiv:2105.01051.
[28]. https://github.com/espnet/espnet
[29]. http://kaldi-asr.org/models/m7
[30]. Hayashi, T., Yamamoto, R., Inoue, K., Yoshimura, T., Watanabe, S., Toda, T., Takeda, K., Zhang, Y. and Tan, X., 2020, May. ESPnet-TTS: Unified, reproducible, and integratableopen source end-to-end text-to-speech toolkit. In ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 7654-7658). IEEE.
[31]. Desplanques, B., Thienpondt, J. and Demuynck, K., 2020. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. arXiv preprint arXiv:2005.07143.
[32]. Dawalatabad, N., Ravanelli, M., Grondin, F., Thienpondt, J., Desplanques, B. and Na, H., 2021. ECAPA-TDNN embeddings for speaker diarization. arXiv preprint arXiv:2104.01466.
[33]. Pal, M., Kumar, M., Peri, R., Park, T.J., Kim, S.H., Lord, C., Bishop, S. and Narayanan, S., 2021. Meta-learning with latent space clustering in generative adversarial network for speaker diarization. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, pp.1204-1219.
[34]. Thienpondt, J., Desplanques, B. and Demuynck, K., 2020. The idlabvoxceleb speaker recognition challenge 2020 system description. arXiv preprint arXiv:2010.12468.

[35]. https://huggingface.co/speechbrain/lang-id-voxlingua107-ecapa
[36]. Sheikh, S.A., Sahidullah, M., Hirsch, F. and Ouni, S., 2022. Introducing ECAPA-TDNN and Wav2Vec2. 0 embeddings to stuttering detection. arXiv preprint arXiv:2204.01564.
[37]. Xue, J., Deng, Y., Li, Y., Sun, J. and Liang, J., 2022. ECAPA-TDNN for Multi-speaker Text-to-speech Synthesis. arXiv preprint arXiv:2203.10473.
[38]. Chen, Z., Chen, S., Wu, Y., Qian, Y., Wang, C., Liu, S., Qian, Y. and Zeng, M., 2022, May. Large-scale self-supervised speech representation learning for automatic speaker verification. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6147-6151). IEEE.