iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.36094

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

Attendance Management System using Google Facenet for Facial Recognition

Rakesh Duggempudi[1], Himanshu Kuncha[2], K N Vaishnavi[3], Kunduru Venkat Reddy[4]
[1, 2, 3]*U.G. Student, ECE, Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad*
[4]*Guide, ECE, Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad*

Abstract: ***Attendance management system is a required tool for attaining attendance in any habitat where attendance is essential. Yet, many of the available techniques consume time, are invasive and it demands manual work from the users. This research is directed at building a less invasive, cost effective and more efficient automated student attendance management system using face recognition that leverages on OpenCV functions for facial recognition. The system provides a GUI for marking attendance. It provides an interface for updating attendance using facial recognition libraries of OpenCV. The system stores attendance in a database which is maintained by the administrator. The administrator can view, update, and change the attendance of the students. The students can view and update their attendance. The system is developed on Open-Source image processing library and the interface is developed using Python Tkinter module. The Tkinter module is an open-source module by which we can develop GUI screens hence, it is not software dependent nor vendor hardware. The OpenCV module used for image processing is interfaced using python.***

Keywords: less invasive, Tkinter, attendance management, facial recognition, image processing

I. INTRODUCTION

Attendance management is very essential in all educational institutions. Every institution has its own system of providing student attendance. Some institutions use manual paper-based approach and others have espouse automated methods such as fingerprint biometric techniques which requires waiting in queues which is time consuming, and it is invasive.

To identify a person, humans often use faces but the progress in the computing capability over time now enable similar identification automatically. The technology involving Facial Recognition is one of the least invasive and fastest growing biometric technology. It works by identification of human beings using the most unique features of their faces.

Facial recognition has features those other biometrics technologies do not have. Facial images can be captured from a distance and any exceptional or particular actions are not required for authentication. Due to such features, the facial recognition technique is used widely, not only for applications like security but also to image retrievals, natural user interfaces and image indexing.

Appearances are incredibly requesting and dynamic articles that are utilized as biometric proof, in character confirmation. Biometrics frameworks, throughout the years have known to be required security devices, in which mass coordinating of enlisted individuals and watch records is performed consistently. Facial acknowledgment innovation has additionally been utilized for both recognizable proof and confirmation of understudies in a study hall. This exploration is pointed toward growing a less obtrusive, savvy and orderly computerized understudy participation the executive's framework utilizing facial acknowledgment. The traditional attendance taking procedure takes up a lot of time and students can also give proxy during this process. This can be avoided by using facial recognition systems which helps in eliminating the false attendance marking chances.

The facial recognition is implemented in a local system and uses a database present in the local system. Hence maintenance of the database is easy and fast. The database provides us better performance than the old storage method using excel sheets. The user should first login before marking the attendance thereby we include protection.

II. LITERATURE SURVEY

In this chapter we explain in detail about the literature survey we have done to attain a base for the project and collect all material, formulas and methods required to implement our solution.

In 2017, from the paper "A Survey on Face Recognition based Students Attendance System", according to Binyam Tesfahun Liyew, Prasun Hazari, the automatic attendance management system using face recognition helps in minimizing the time required to make attendance and increase the overall efficiency of the attendance system. They came to this conclusion by comparing various attendance management systems.[1]

In 2018, from the paper "Automatic Attendance Monitoring System Using Face Recognition Techniques", according to Chetan Chaudhari, Rahul Raj, Swajey Shrinath ,Mrs Tanuja Sali, Viola-Jones algorithm can be used for face detection and Principal Component Analysis algorithm can be used for face detection. These algorithms eliminate manual calling and attendance entry in institutional websites.[2]

In 2019, from the paper "Face Recognition Based Attendance System", according to Nandhini R, Duraimurugan N, S.P.Chokkalingam by capturing the faces of every student must be captured and the images are processed using CNN(Convolutional Neural Networks)and the attendance for the students are marked automatically without human consent.[3]

In 2019, from the paper "A Study on Automated Attendance System using Facial Recognition", according to Ms. Munmun Bhagat, Chaitanya Kirkase, Yash Hawaldar, Aishwarya Paigude, Shivani Nimbalkar the automated attendance system using facial recognition can be implemented using SVM on LBP feature. The system integrates the face recognition algorithm with an android application.[4]

Our project is an attendance management system using the FaceNet algorithm by Google. This algorithm can be accurate to more than 90% for facial recognition. FaceNet recognition takes the image as 128D unit hypersphere.It does facial recognition in 3 steps:Clustering ,Similarity detection and Classification. It is also popular as one-shot facial recognition which uses only one shot for recognizing faces.

III. PROPOSED SYSTEM

While there are various other attendance management systems present in the market, we propose to do the same task using an algorithm called FaceNet developed by google. We are using FaceNet specifically due its ability to provide an astonishing 90% accuracy with respect to the created dataset. The project will be executed in python using various libraries such as tkinter, sqlite3, pil, imltis, numpy, dateime, argparse and pickle. Each has its specific significance in the execution of the project. While tkinter is being used for building the GUI screens required for a user-friendly interference, sqlite3 is being employed for the database. Sqlite3 is specifically being used due to the features it provides such as user-friendly environment, easy implementation, can run on smaller and portable devices such as smart phones.

Initially we create tables for the database we store the attendance in. Now, for generating a dataset, we define the location for the dataset and also use classifiers to detect the face and capture the required images limited only to the face and not the external environment. For extracting the embeddings, we construct the argument parser and parse the argument, load our serialized face detector and serialized face embedding model from the disk. From here, grabbing the paths to the input images in our dataset, initializing our list of facial extracted embeddings and performing various other such functions we get our model where we compare the photos, that is converting them into 128-bit vector images helping the algorithm to detect the face efficiently.

Implementing a interactive model for user friendly experience being one of the motive of this project, we are using tkinter, a default library available in python to build the GUI screens. For the admin module specifically, we will have a screen where we will be able to add a new student, modify their attendance, remove a student, check their attendance and other such options which would generally be made available to an admin. Here adding a new student, that is creating a new dataset, previously built tables and dataset generators come into play.

Figure 1: Dataset Generation Block diagram

Once the dataset is generated for a new subject or student, we now move on to the student module where the similar GUI screens are built for the student to login in his/her account where they will be able to give their attendance and also check their present attendance percentage. The database tables created, help in storing the student attendance percentage in the student module as well as in the admin module with different GUI screens.

Figure 2: Implementation Block diagram

Here when the student tries to give his/her attendance we import videostream (library) which basically enables us to operate the webcam. Now, constructing the arguments, we load the actual face recognition model along with the label encoder. Initializing the videostream and allowing the camera sensor to warmup, we now assign the value to resize the frame to have a width, maintain the aspect ratio, and grab the image dimensions. We construct a blob for the face ROI, then pass the blob through our face embedding model to obtain the 128-d quantification of the face.

A. The Method

Deep convolutional web (network) is actively in use by that constitutes two core architectures:

The contemporary Inception type web (networks) and the Zeiler&Fergus style networks. The Zeiler&Fergus model which constitutes local response normalizations, manifold max pooling layers, interleaved layers of convolutions and non-linear activations. In addition to that we add several 1x1xd convolution layers.

The subsequent design is established on the I formation mockup of Szegedy et al. that was as of late utilized as the triumphant methodology for ImageNet 2014. [5]

These organizations utilize blended layers that run pooling layers in equal and a few distinctive convolutional and connect their input. It was determined that these mockups can get the quantity of impediments down by up to numerous times and have the power to get down the quantity of FLOPS needed for same kind of successful run.

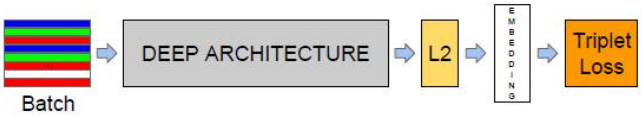


Figure 3: Model Structure

A group input layer and a deep Convolutional Neural Networks followed by L2 normalization, resulting in the face embedding is constituted by the network. During the training, the above process is followed by triplet loss.

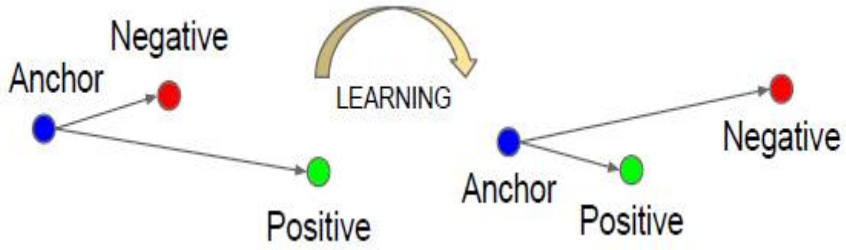


Figure 4: Triple Loss

The Triplet Loss restricts the distance between a positive and an anchor, the two of which have a comparative character, and grows the distance between an antagonistic of a substitute character and the anchor.

The paper uses the triplets misfortune that clearly reflects what we need to achieve in face check, affirmation and bundling. To be obvious, we make a pass at an starting f(x), from a picture x into a component space Rd, with the final objective that the squared distance between all faces, self-sufficient of imaging conditions, of a comparative character is close to nothing, however the squared distance two or three face images from other personalities is gigantic.

The implanting is addressed by f(x) 2 Rd. It's anything but an image x into a d-dimensional Euclidean space. Moreover, we oblige this implanting to gauge on the d-dimensional hypersphere, i.e., kf(x)k2 = 1. This misfortune is roused inside the setting of closest neighbor grouping. Here we might want to make sure that the image xai (anchor) of

a particular individual is nearer to all or some other pictures xpi (positive) of a comparable individual than it's to any picture xni (pessimistic) of the other individual.

$$\|x_i^a - x_i^p\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2, \forall (x_i^a, x_i^p, x_i^n) \in \mathcal{T} .$$

Where is an edge that is executed among positive and negative sets. T is the arrangement of all achievable triplets in the preparation set and has cardinality N. The misfortune that is being decreased is then L =

$$\sum_{i}^{N}\left[\|f(x_i^a)-f(x_i^p)\|_2^2-\|f(x_i^a)-f(x_i^n)\|_2^2+\alpha\right]_+$$

Leading to all reasonable triplets would bring about numerous triplets that are effectively fought. .

To guarantee quick assembly, it is critical to choose triplets that disregard the trio imperative. This implies that, given xai, we need to choose a xpi (hard sure) to such an extent that and comparatively, xni (hard regrettable) to such an extent that

$$\text{argmax}_{x_i^p}\ \|f(x_i^a)-f(x_i^p)\|_2^2$$

.

$$\text{argmin}_{x_i^n}\ \|f(x_i^a)-f(x_i^n)\|_2^2$$

it is infeasible to quantify the argmin and argmax across the entire preparing set. Moreover, it may affect vulnerable arranging, as mislabeled and inadequately imaged countenances would regulate the hard positives and negatives. It is infeasible to figure the argmin and argmax across the entire preparing set. Additionally, it may affect defenseless preparing, as mislabelled and insufficiently imaged appearances would regulate the hard positives and negatives. There are two accumulate decisions that keep up from this issue:

- Generate triplets separated every n steps, making use of the most recent association assigned spot and enlisting the argmax and argmin on a data subset.

-Generate triplets online. this process will be carried out by selecting the hard positive/negative exemplars from the internal database of the selected components of a mini batch.

Here, we base on the online age and make use of huge and more modest than normal groups in the solicitation several thousand models and simply figure the argmax and argmin inside a downsized pack. To have a huge depiction of the anchorpositive distances, it ought to be ensured that an inconsequential number of models of any one character is accessible in every last group. In our examinations we test the readiness data so much that around 40 appearances are picked for every character per minibatch. Besides, indiscriminately analyzed negative appearances are added to each downsized bunch. Maybe than picking the hardest positive, we use all anchorpositive sets in a little cluster while at this point picking the hard negatives. We do not have a close to one another assessment of hard anchor-positive sets versus all anchor-positive sets inside a limited scale cluster, yet we found before long that all anchorpositive technique was all the more consistent and joined imperceptibly speedier around the beginning of getting ready.

We additionally investigated the disconnected age of triplets related to the online age and it might permit the utilization of more modest group sizes, yet the analyses were uncertain. Choosing the hardest negatives can by and by prompt awful neighborhood minima from the beginning in preparing, explicitly it can bring about an imploded model (for example f(x) = 0). To alleviate this, it assists with choosing xni to such an extent that these negative models semi-hard.

$$\|f(x_i^a)-f(x_i^p)\|_2^2<\|f(x_i^a)-f(x_i^n)\|_2^2$$

As they are further away from the anchor than the positive model, yet hard in light of the fact that the squared distance is near the anchorpositive distance. Those negatives lie inside the edge α. As referenced previously, right trio determination is urgent for quick union. From one viewpoint we might want to utilize little smaller than normal clumps as these will in general improve assembly during Stochastic Gradient Descent (SGD). Then again, execution subtleties make clusters of tens to many models more proficient. The primary imperative concerning the cluster size, be that as it may, is the manner in which we select hard pertinent triplets from inside the small groups. In many examinations we utilize a group size of around 1,800 models.

Train the CNN utilizing Stochastic Gradient Descent (SGD) with standard backprop. In various assessments we start with a learning pace of 0.05 which we lower to settle the model. The models are introduced from emotional, as and organized on a CPU bundle for 1,000 to 2,000 hours.

The lessening in the affliction (and expansion in precision) drops down definitely after 500h of preparing, at any rate additional arranging can notwithstanding basically improve execution. The edge α is set to 0.2. We utilized two sorts of models and inspect their compromises in more detail in the exploratory area. Their possible contrasts lie in the capability of cutoff points and FLOPS. The best model might be diverse relying on the application. Considering a example, the mockup that is being processed in a datacenter can possess various cutoff points and have to check out the countless FLOPS, while a mockup running on a remote necessities to have moderately couple of cutoff points, so it's anything but's a way into memory. The entirety of our models utilizes changed straight units as the non-direct incitation work.

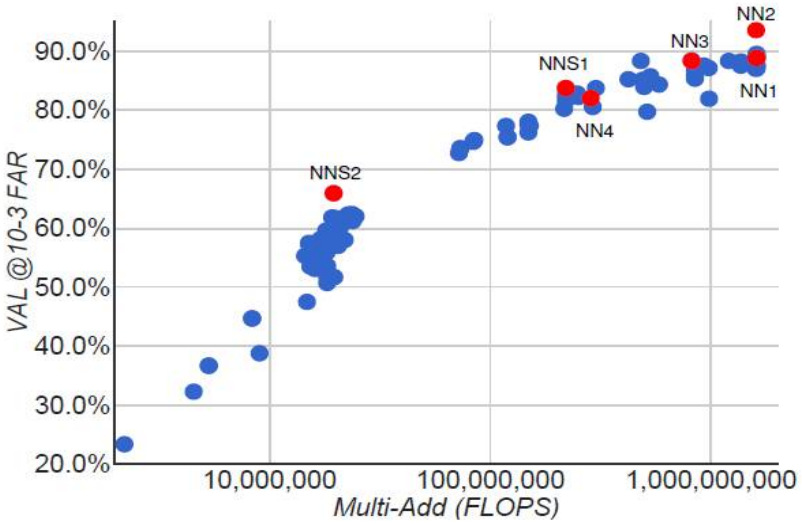


Figure 5: Distribution of data - flops

It is the trade-off between FLOPS and validity for a huge range of architectures and different model sizes. Highlighted are the four mock-ups that we pivot on in our experiments.

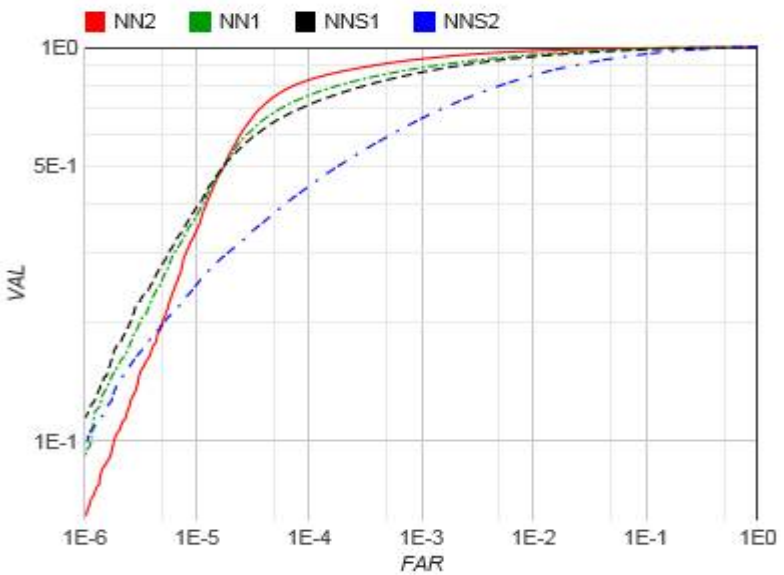


Figure 6: Network Architectures

The above plot comprises of the whole ROC for the four special models on our own photos test set from region 4.2. The sharp drop at 10E-4 FAR can be explained by upheaval in the groundtruth marks. The models masterminded by execution are: NN2: 224×224 data Inception based model; NN1: Zeiler&Fergus based association with 1×1 convolutions; NNS1: little Inception style model with simply 220M FLOPS;NNS2: little Inception model with simply 20M FLOPS.

The examination of various introducing dimensionalities and picked 128 for all tests other than the relationship uncovered. One would expect the greater embeddings toper plan least practically identical to the more unassuming ones, regardless, it is feasible that they need more getting ready to attain a comparable accuracy. It must be seen, that while setting up a 128-dimensional float vector is used, yet it will in general be quantized to 128-bytes without loss of accuracy. In this manner each of the faces is moderately tended to by a 128-dimensional byte vector, that is perfect for gigantic extension gathering and affirmation. More unassuming embeddings are feasible at a minor loss of exactness and can be used on mobile phones.
This addresses the entirety of the photos that were wrongly gathered on LFW. The similar cutting back in blunder is 60%. Utilizing other significant degree more pictures (many millions) give a little lift, however the improvement eases off.
The assessment of model on LFW utilizing the quality show for unlimited, marked external information. Nine preparing parts are wont to choose the L2-distance limit. Classification (same or extraordinary) is then performed on the 10th test split. The picked ideal edge is 1.242 for all test parts aside from split eighth (1.256). Our model is assessed in two modes:

1) Fixed focus harvest of the LFW gave thumbnail.
2) A restrictive face identifier (like Picasa [3]) is run on the gave LFW thumbnails. In the event that it neglects to adjust the face (this occurs for 2 pictures), the LFW arrangement is utilized . Figure 5 gives an outline of all disappointment cases. It shows bogus acknowledges on the top just as bogus oddballs at the base. We accomplish a classification exactness of 98.87%±0.15 when utilizing the fixed focus crop portrayed in and the record breaking 99.63%±0.09 standard blunder of the mean when utilizing the additional face arrangement. This diminishes the blunder detailed for DeepFace in by in excess of a factor of 7 and the past cutting edge revealed for DeepId2+ by 30%. This is the near representation of mockup NN1, in any case, even a lot more modest NN3 completes execution that isn't measurably majorly spectacular.

The normal similitude of all sets of the first 100 edges that our face indicator recognizes in every video. This gives us a classification precision of 95.12%±0.39. Utilizing the first 1,000 casings results in 95.18%. Contrasted with 91.4% who likewise assess 100 edges for every video we diminish the mistake rate by practically half. DeepId2+ accomplished 93.2% and our strategy decreases this mistake by 30%, tantamount to our enhancement for LFW.
The minimized implanting fits be utilized to bunch a clients individual photographs into gatherings of individuals with a similar personality. The imperatives in task forced by grouping faces, contrasted with the unadulterated verification task, lead to really astounding outcomes. It's anything but a reasonable grandstand of the staggering invariance to impediment, lighting, present and even age.

IV. SIMULATION RESULTS

Testing is done to evaluate the overall performance of the system. It is done to check whether the functionalities of the system meet the requirements. Testing is done to check if there any bugs present in the system. By testing, we can know if there are any errors present in the system which will make them inconsistent. Testing is an important phase of the project which is done before deployment.
There are 2 types of testing namely black box testing and white box testing. Black box testing is done to check only the functionalities of the software without checking the implementation part. White box testing is type of testing where both the functionalities and the internal code is tested and checked for any errors. The white box testing is usually done by the developer whereas the black box testing is done by the testers and beta users.

A. Testcases

After successful implementation of the modules, it was tested with various testcases. The various functionalities which were tested are as follows:

1) Creating an admin for the first time.
2) Successful login of admin into the GUI.
3) Registering a new student with a user-id and password.
4) Creating face identification for the new user with the registered user-id.
5) Creation and maintenance of student table, attendance table in the database.
6) Successful login of the user into the GUI which enables them to access the attendance.
7) Student can successfully update the attendance and reflect the same into the database.
8) Students can view their previous attendance.
9) Admin can edit the attendance of a student present in the database.

The above testing was done in real time with the students being notified as their attendance gets updated.

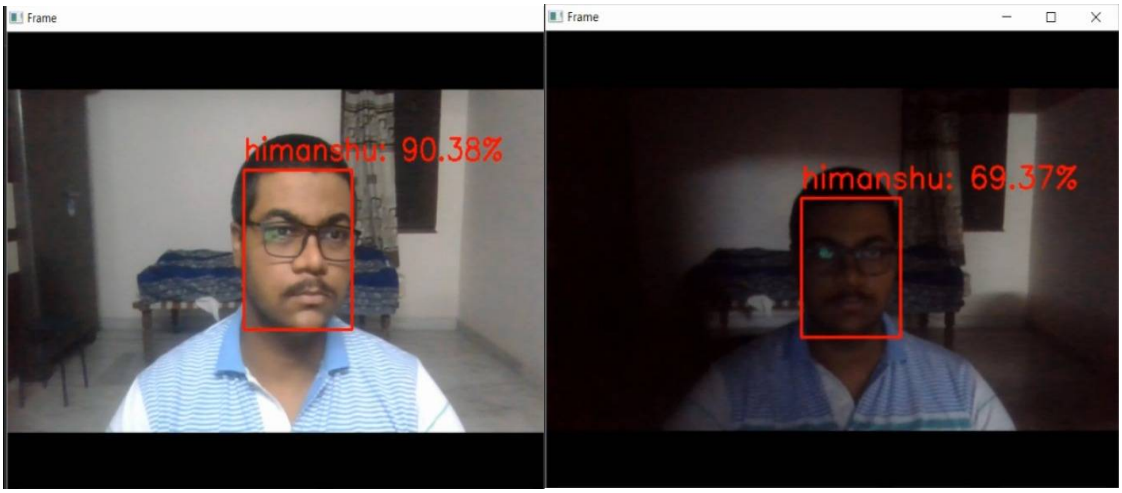


Fig 5. The figure on the left is a successful case and the figure on the right is a failure case.

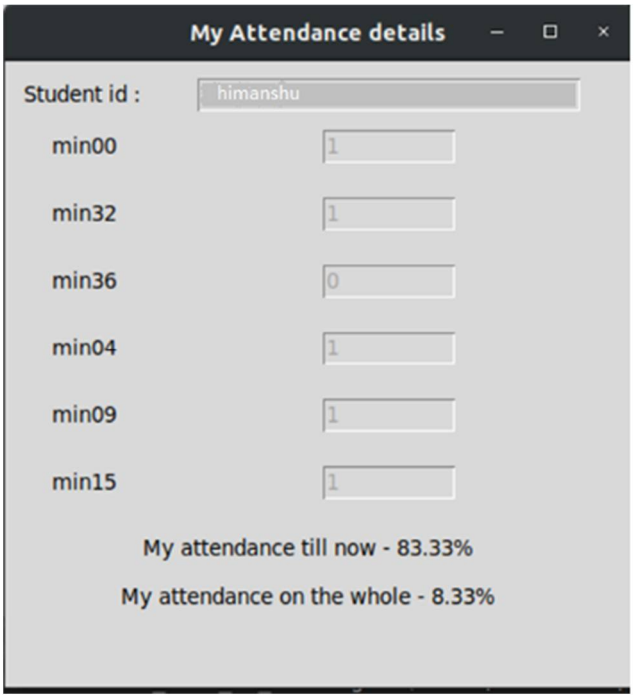


Fig 6. The figure shows the updated attendance of the student.

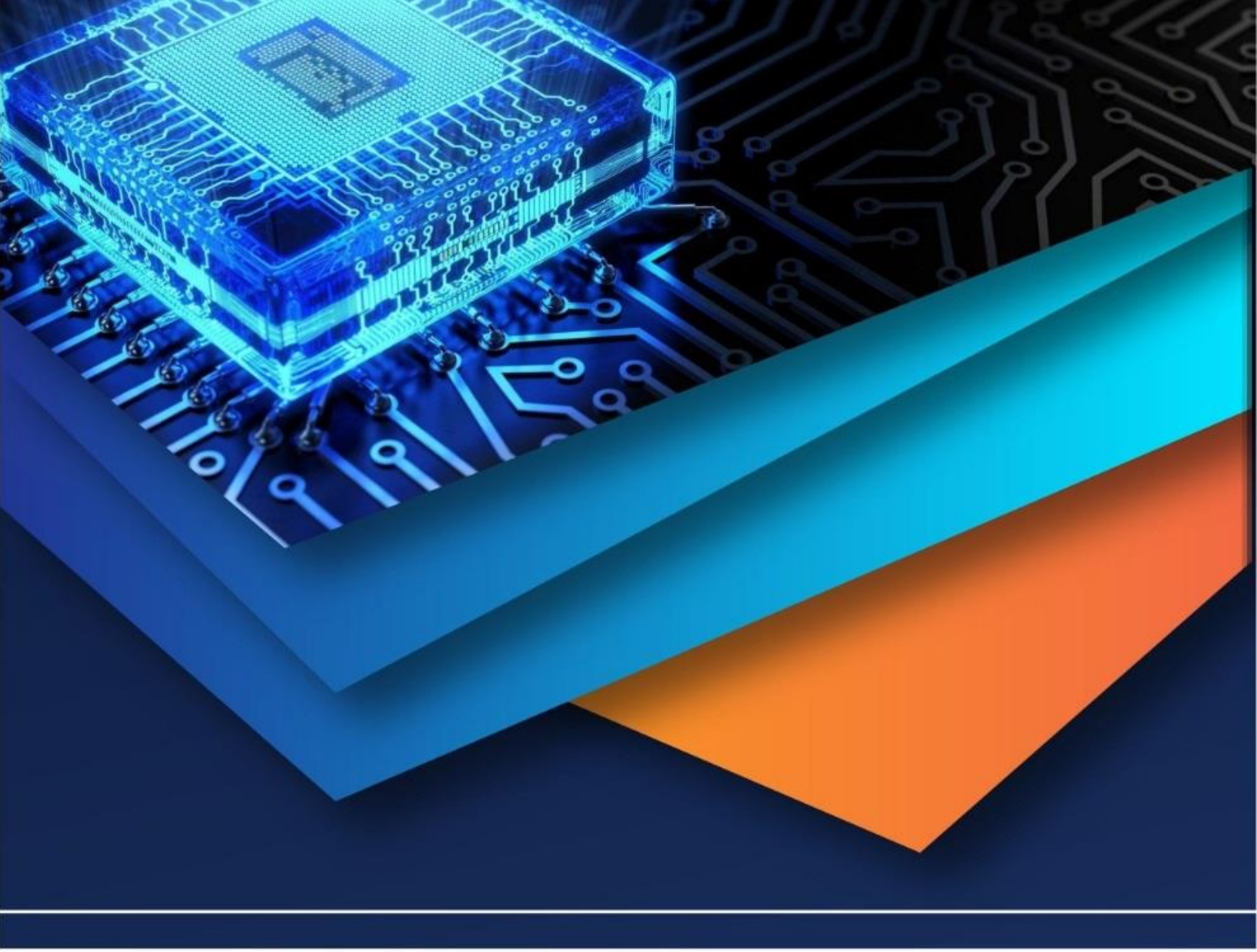
V. CONCLUSION AND FUTURE WORK

The attendance management system using facial recognition is implemented with all the required functionalities and has been tested successfully. Admin can register new students and can update their face id of those students. The attendance of students present in the database can be viewed successfully by the admin which can be updated if required. The database is consistent, and all the changes are committed to it whenever any changes are made. The facial recognition is done with an accuracy of more than 80%. Registered students can update their attendance after recognition of their face. The students can view their previous attendance and check their attendance percentage. The view of attendance to the students is view only and hence security is provided to the database from tampering the data. On successful updating the attendance into the database, student will be notified with the same. Updating the password and face id of students is successfully implemented. The students cannot view the attendance of other students, thereby providing more security. Rows will be added into the database dynamically based on time thereby no maintenance of the database is required for manually addition of the rows.

Student attendance is successfully updated and maintained in the database thereby atomizing the attendance managing tasks. Facial recognition is also implemented successfully, and the overall system is running successfully with all the desired functionalities. The developed system is implemented on a local machine. It can further be deployed on a cloud-based machine where all the computation like facial recognition is carried out on the cloud. This will increase the accuracy of the facial recognition. The GUI can be implemented using web pages and can be hosted on any computer using servers. This will enable to access the system from any system within a network.

REFERENCES

[1] Binyam Tesfahun Liyew, Prasun Hazari(2017) "A Survey on Face Recognition based Students Attendance System ",Parul Institute of Engineering and Technology, Vadodara, India, September 2017, ISSN(online) 2278-1021, ISSN(Print) 2319 5940
[2] Chetan Chaudhari, Rahul Raj, Swajey Shrinath ,Mrs Tanuja Sali(2018) "Automatic Attendance Monitoring System Using Face Recognition Techniques", PCCOE, Akurdi, Pune, April 2018, ISSN 2319-1058
[3] Nandhini R, Duraimurugan N, S.P.Chokkalingam (2019) "Face Recognition Based Attendance System" February 2019, ISSN 2249-8958
[4] Ms. Munmun Bhagat, Chaitanya Kirkase, Yash Hawaldar, Aishwarya Paigude, Shivani Nimbalkar (2019) "A Study on Automated Attendance System using Facial Recognition" ,December 2019, e-ISSN:2395-0056,p-ISSN:2395-0072
[5] FaceNet: A Unified Embedding for Face Recognition and Clustering by Florian Schroff, Dmitry Kalenichenko, James Philbin.
https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/
https://www.pyimagesearch.com/2018/02/26/face-detection-with-opencv-and-deep-learning/
https://docs.python.org/3/tutorials
https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html
https://docs.python.org/3/tutorial/datastructures.html



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY