iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VI **Month of publication:** June 2020

DOI: http://doi.org/10.22214/ijraset.2020.6275

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 8 Issue VI June 2020- Available at www.ijraset.com

Payment Interface using Face Recognition

Reetik Mathankar[1], Vipul Karande[2], Rahul Hipparkar[3], Prof. Shikha Pachouly[4]
[1, 2, 3]*Departmentt of Computer Engineering, AISSMS COE, Savitri Bai Phule University, Pune*
[4]*Assistant Professor, Department of Computer Engineering, AISSMS COE, SavitriBai Phule Pune University, Pune*

Abstract: ***The 3 modes of payment that people use to carry out any transaction are: cash, cards, e wallets. But there might be some circumstances when the buyer cannot use any of the above mentioned modes of payment. To deal with such case a payment interface can be developed which won't require the customer to have a phone or a card or cash. Only the shopkeeper needs to have a smartphone having a camera. The main motive of this paper is to study and implement face-detection using a mobile phone. The customer is authenticated through face recognition based on CNN in order to perform transaction.***
Keywords: Transaction Processing, Face Recognition, Database Handling, Security, E-Commerce, Payment Gateway, Android

I. INTRODUCTION

Every now and then in our day to day lives, we are always visiting a shop to buy some things according to our needs. In order to buy a product from a shop we need to pay the of that product in from money. This payment can be done either by cash, or card or an e-wallet. Nowadays e-wallets has become a trend, since mobile phones are widely used by the people. E-wallets are intended to replace the existing physical wallet, with its notes, coins, plastic cards etc. E-wallets makes it easier for the customers as well as the shopkeepers it requires no cash physically to carry out any transaction. Also e-wallets provide not only secure transactions but also they keep the records of each and every transactions that has been carried out successfully. There might be some circumstances where the customer can neither pay through cash, card or any e-wallet. This is where our proposed system can be used. The proposed payment interface makes use of the Aadhar Card that is linked with the user's bank account for verification purpose. The main advantage of our system is that there is no need to collect any data from the customer and make use of what is existing already. Facial Recognition is quite famous these days. Many smartphones has the feature of unlocking the phone on basis of face-recognition alongside with fingerprint. The new of Iphones uses face-recognition as their primary lock. As it is secure, it is more trending. Face recognition is the most crucial part of this project. Face recognition is used to identify or verify person from a digital image or a videos frame from a video source. Face recognition has been a crucial human-computer interaction tool due to its importance in security systems, access-control, video surveillance. Face recognition has become main method for identifying a person due to its non-intrusive nature as compared to other biometric techniques. To perform face recognition we can use traditional machine learning approaches But face recognition using traditional machine learning methods facing challenges like pose variation, facial disguises, lighting of the scene, the complexity of the image background, and changes in facial expressions. To overcome problems in traditional machine learning approaches we could use deep learning based methods. As compared to traditional machine learning approaches deep learning based methods have shown better performance in terms of accuracy and speed of processing in image recognition.

II. RELATED WORK

Our research has shown different methods in which face-recognition can be implemented.

TABLE I
LITERATURE SURVEY

Sr. No	Name of the Reference Paper	Publish Year	Author	Description
[1]	When Face Recognition Meets with Deep Learning: An Evaluation of Convolutional Neural Networks for Face Recognition	2015	Guosheng Hu, Yongxin Yang, Dong Yi, Josef Kittler, William Christmas, Stan Z. Li, Timothy Hospedales	CNN based face detection model is built. The model built is trained with public data which consists of faces in wild. Their experiment shows that they have built an efficient CNN based face detection model and the performances of face detection has improved considerably. They have used multiple layers of CNN model to get higher results which in turn made it a robust detection model.

[2]	Local Binary Pattern Network: A Deep Learning Approach for Face recognition	2016	Meng Xi, Liang Chen, Desanka Polajnar, Weinyang Tong	Local Binary Pattern for Face detection I studied and implemented. It is based on unsupervised deep learning mechanism which uses multiple layers of data which helps to extract and compares multilevel hierarchy. After extracting features it allows to check similarities between two images. For this the model uses LBPnet and deep neural network to extract and classification.
[3]	Frankenstein: Learning Deep Face Representations using Small Data	2017	Guosheng Hu, Xiaojiang Peng, Yongxin Yang, Timothy M. Hospedales, Jakob Verbeek	The author has trained his model with thousands and thousands of images which consisted of animated faces and real life person images. In this the model is trained with 10000 images which is trained using CNN. In this method the author has proposed a 2D data synthesis technique to get details from the smaller data on larger basis. This technique will swap the details of face to get or generate a new face.
[4]	Towards a deep learning framework for unconstrained face detection	2016	Yutong Zheng, Chenchen Zhu, Khoa Luu, Chandrasekhar Bhagavatula, T. Hoang Ngan Le, Marios Savvides	The author has trained his model with thousands and thousands of images which consisted of animated faces and real life person images. In this the model is trained with 10000 images which is trained using CNN. In this method the author has proposed a 2D data synthesis technique to get details from the smaller data on larger basis. This technique will swap the details of face to get or generate a new face.
[5]	Facial Landmark Detection Via Pose-induced Auto-encoder Networks	2015	Yu Chen, Wei Luo, Jian Yang	In this paper the author suggests a Pose Induced Auto encoder Networks which detects the landmarks on people in different positions. First all the landmarks with different poses are detected and then an estimation is prepared which learns various things from it. After that they compose and tune each and every landmark.
[6]	Applying a Deep Learning Convolutional Neural Network (CNN) Approach for Building a Face Recognition System: A Review	2017	Binyam Tesfahun Liyew	In this paper the author has studied various paper from different journal and explained and compared all the models. He has discussed about CNN models and how they work in different cases. The author has has also said the CNN face detection has tremendous benefits in real life applications such as Payment, Criminal Identification, etc.
[7]	Face Recognition based on Convolutional Network	2018	Musab Coskun, Aysegul Ucar, Yakup Demir	n this paper author has stated the importance of face detection in surveillance and security systems. The author proposed a Convolutional (CNN) based architecture to improve the current facial detection methods. He used CNN architecture along with Softmax classifier to extract and classify faces.
[8]	A Convolutional Neural Network based on TensorFlow for Face Recognition	2017	Liping Yuan , Zhiyi Qu , Yufeng Zhao, Hongshuai Zhang, Qing Nian	Author said that face recognition has become very important now a days. Author also stated that traditional methods does not take into account different actions like pose, facial expression, occulision, etc. Author has developed a CNN architecture using TensorFlow which is a open source for deep learning framework. Their system showed great results and had a higher accuracy.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VI June 2020- Available at www.ijraset.com

III. FACE RECOGNITION USING CNN

The approach we are going to used for face recognition fairly straight forward. We are using deep Convolutional neural network to extract features from input images .We are using deep convolutional neural network to extract a bunch of features or pixel values that describe face encodings.

We have developed the architecture of convolutional neural network that need to be trained to automatically identify features of faces and calculate pixel values based on that.

The output of the neural network can be thought of as an identifier for a particular person's face — if you pass in different images of the same person, the output of the neural network will be very similar/close, whereas if you pass in images of a different person, the output will be very different.

To speed-up processing CNN make use of Graphical Processing Units (GPU). We have developed a CNN architecture for face recognition which starts with pre-processing stage: colour space conversion and resize of images, continues with extraction of facial features and afterwards extracted feature set is classified. In our system, Support Vector Classifier(SVC) is used for classification on the basis of facial features extracted from CNN. To calculate similarity between faces we use Euclidean distance. In next section we will discuss CNN architecture and proposed algorithm.

IV. CNN ARCHITECTURE

CNNs are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. CNNs are a type of feed-forward neural networks made up of many layers. In deep learning, a convolutional neural networks is a class of deep neural networks which are most commonly used for image recognition. CNNs consist of neurons that have learnable weights or parameters and biases. Each neuron takes some inputs, performs convolution and optionally follows it with a non-linearity A typical CNN Architecture is shown in Fig. The structure of CNN contains Convolutional, pooling, Rectified Linear Unit (ReLU) and Fully Connected Layers.

A. Convolutional Layer

Convolutional layer performs the core building block of a Convolutional Network that does most of the computational heavy lifting. The purpose of Convolution layer is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input image. The input image is convoluted by employing a set of learnable neurons. This produces a feature map in the output image and after that the feature maps are given as input data to the next convolutional layer

B. Pooling Layer

Pooling layer reduces the dimensionality of each feature map but continues to have the most important information. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Pooling layer operates in each feature map independently. The most common approach used in pooling is max pooling . In max pooling each region is down sampled by a non-linear operation such as average or maximum. This layer is used to achieve better generalization and faster convergence. It is also used to reduce translation and distortion. It is usually placed between convolution layers.

C. ReLU Layer

ReLU stands for rectified linear unit and it is a non-linear activation function. ReLU is a piecewise linear function which is applied per pixel. It reconstitutes all negative values in the feature map by zero. ReLU is an activation function which is linear for all positive values and zero for all negative values. In order to understand how the ReLU operates , we assumes that there is a neuron input given as x and from that the rectifier is defined as $f(x) = \max(0,x)$.

D. Fully Connected Layer

Fully Connected Layer (FCL) is simply, feed forward neural networks. Fully Connected Layers form the last few layers in the network. Fully Connected Layer is formed by connecting every filter in previous layer to every filter in the next layer. The output from the final pooling or convolutional layer which is flattened or converted into the vector is given as input to the Fully Connected layer.

Image below shows the architecture of CNN.

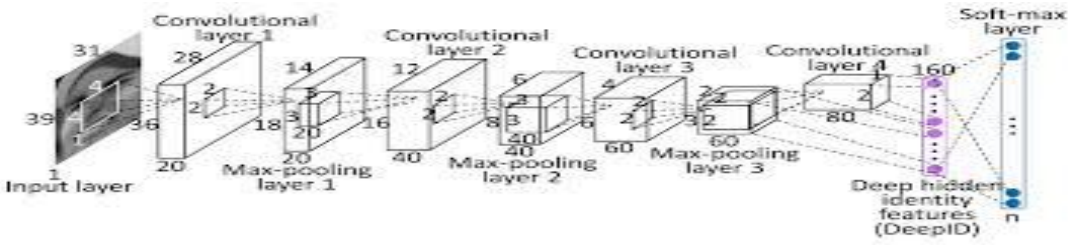


Fig. 1 CNN Architecture

V. ALGORITHM

There are many methods of face-recognition, like 3D-based facial recognition, face-descriptor based methods, artificial neural network method.. We have used CNN for this project. The following diagram shows the flow of this method:

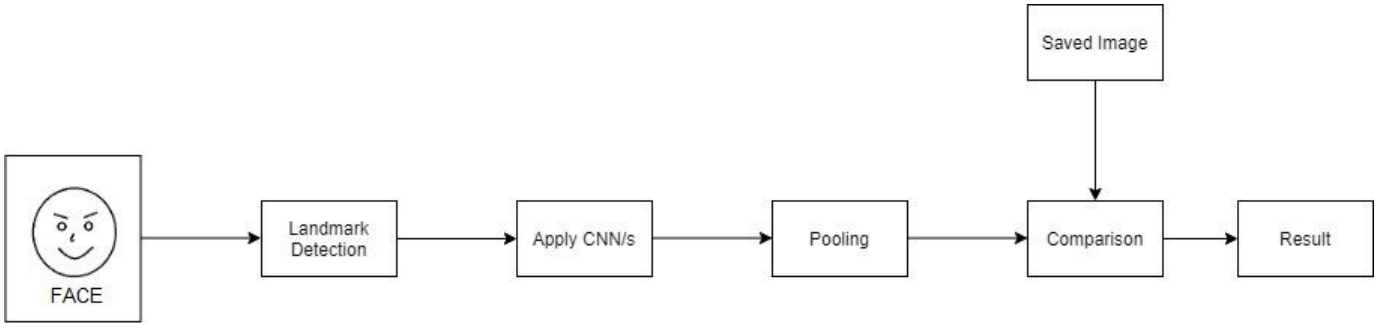


Fig. 2 Flow of Algorithm

A. In the first step an image of the face is provided. By applying Landmark Detection to the image, it actually figures out if the face is present in the image or not. Landmark Detection tries to find out the the points present on a human face. Points such as the eyebrows, eyes, nose, mouth. Total of 67 points are found out in this process. After the points are found out, the corner points i.e. the points at the end in all directions are figured out. Using these corner points a bounding box is found out and cropped from the original image. This box contains the cropped face and is called as "Region of Interest"(ROI). ROI represents the only portion of image that is needed for further process.

B. In the next step CNN classifiers are applied to ROI. CNN tries to extract the features from the face. Features like horizontal lines of the eyebrows, eyes, end points of mouth, etc.

C. The features are then pooled and converted into a vector of 1x128. Another image that is saved is also processed in the same way and converted into a vector of 1x128.

D. These features are are then compared. L1 norm is applied on both the vectors and if the distance between is less than 0.6 then it's a match. This shows the comparison is successful.

VI. IMPLEMENTATION

As mentioned earlier, the main aim of this project is to create a payment interface, which implements facial-recognition as main security feature. The android application must be installed only on the shopkeeper's mobile phone, to carry out any transaction the shopkeeper must enter the amount in the UI, then the camera module starts. The customer verification is done when the server at the backend makes a match. The transaction will either be successful or a failure depending upon response from the server.
There are 3 main components for the implementation:

A. Server

The server made is python and deployed over flask. It has 3 main purposes: i) Respond the User Application. ii) Carry out the process of face recognition. iii) Interact with the database.

Some important Packages imported :-
os - This provides an interface between the OS on which the python is running.
dlib - It is a toolkit which contains all the methods of machine learning and data science applications. It also has a CNN based face detector.
face_recognition - It helps to detect a face in an image.
json - It is used to import json module. json can store booleans, numbers, tuples, etc.
base64 - It is used to encode binary data to base64 format.

There are few APIs that we have built for proper working of the server, below figure shows the basic functionality of the APIs:

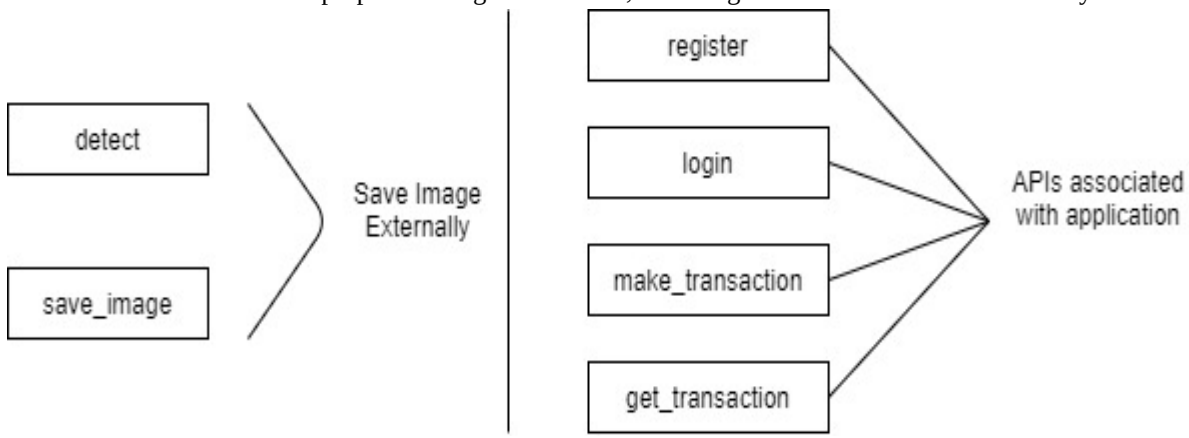


Fig. 2 APIs in Server

1) *Detect Image:* This API helps us to detect the face in image. A standard face image which is stored in the database in json format is encoded to base64 and stored in a array.Then this array is called in the encoding to check with the image captured with the help of camera and is matched with the encoding of the stored image in database. If the variables match then the model has successfully detected a face on its screen. *encodings = face_recognition.face_encodings(im, face_locations)*
2) *Save Image:* Since this is a prototype, we had to create our own database. To insert image along with its data in database, the application named “Postman” is used. This api checks if there is a face in the image. If face is found then it stores it into the database.
3) *Register Shopkeeper:* To register the shopkeeper needs to enter his name, shop-name, and phone number,a long with his image.
4) *Login:* Once the user(in this case shopkeeper) registers itself in the database then he can login anytime in the system. The main authentication factor during login is shopkeeper’s face which is matched with image stored in the database. The detect API part is called again and all the value of the detected image is stored in a variable.

em_need_to_compare = np.load(os.path.abspath(result[0]['embedding_path']))
encoding = face_recognition.face_encodings(im, face_locations[:1])[0]
valid = match_embeddings(em_need_to_compare, encoding)
These line of code checks the embedded values of the images. If the difference in values is less than 0.6 then two faces are similar and shopkeeper gets the access to system.

5) Make Transaction: This is the most important part of the system. In this API. When customer has to make the transaction first the phone no. is taken from the user. Using this we will find out the name and image of the customer. Then all the encoded values of the stored images and the values of the image captured by camera are compared using Hamming Distance and if the difference is less than 0.6, then both the faces are same. After confirming that the person and the money from the account from which he access money are same then the transaction is completed. Before commiting the transaction it also checks where there’s sufficient amount in the account. If not then it rolls backs to the last saved state. This prevents redundancy in the database and also follows atomicity.
6) *Update Database:* In this step all the transactions are updated in the database.

B. Android Application

The application is the main GUI which is used for interaction by the shopkeeper. The application is created using Java.

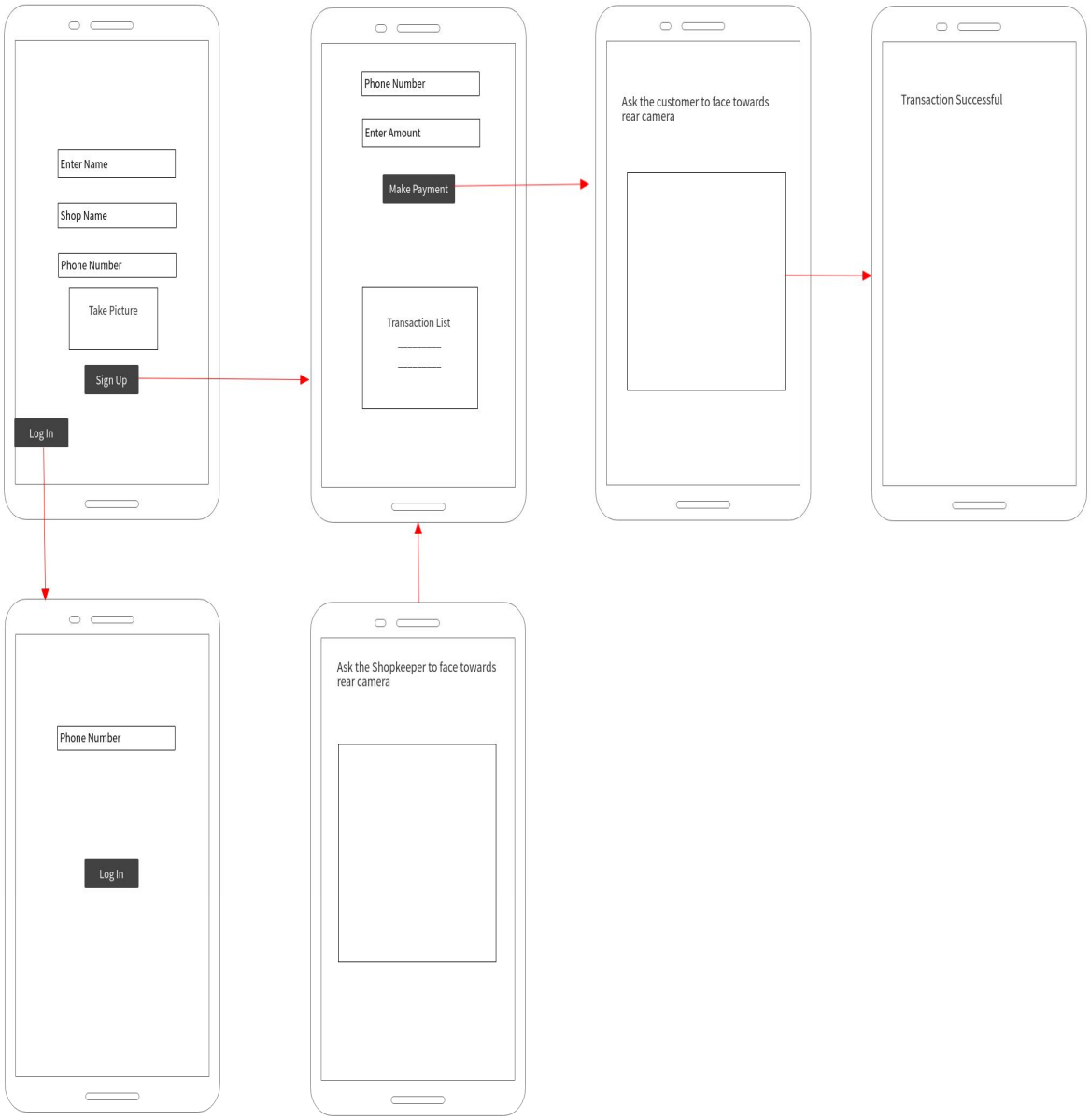


Fig. 2 Wireframe of Application

The application is connected to the server using Volley. For using Volley we had added the following dependency in the gradle:
implementation 'com.android.volley:volley:1.1.1'

C. Database

We have created the prototype of database in using Mysql. There are 3 tables :
>Owner_table which contains the name and phone number of the shopkeeper along with face_encoding used to login.
>Bank_table which contains name, aadhar card number, phone number, balance and face_encoding of a person.
>Transaction_table which contains all the transactions.
NOTE: An owner can only SignUp using his phone number if his entry is existing in Bank_table.

VII. TESTING

After Implementation we have tested the server and the application. Many test cases were used on both server and the application. The following table shows the results of our test cases.

TABLE III
TESTING TABLE

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Pass/ Fail
TU01	Check if server is running.	1. Type the public ip in URL space.	http://192.168.0.107	A page with "hello" message should be returned	"hello" message displayed	Pass
TU02	Check if server is a with instance being stopped	1. Type the public ip in URL space.	http://192.168.0.107	"hello" message should not be displayed	Unable to connect	Pass
TU03	Different protocols used to access server.	1. Using https instead of http.	https://192.168.0.107	Should not be accessed	Secure Connection failed	Pass
TU04	Using inappropriate data while storing data in database using Postman	1. Write names instead of name json in postman	{"names":"abc" }	"something went wrong" message must appear in terminal	"something went wrong" appeared in terminal	Pass
TU05	Sending an image having no face to store in database	1. Upload an image having no face	Image of lemon is given	"No face present in image" message must appear	"No face present in image" message is shown in terminal	Pass
TU06	Storing same data twice in database	1. Store appropriate data once. 2. Store the same data again.	Same data twice	"Phone number already registered" message must appear.	"Phone number already registered" shown in terminal	Pass
TU07	Handling of multiple sign up requests by server through application	1. Sign up on application using two or more phones simultaneously	Fill data to sign up at same time from multiple phones.	Server should update the database properly.	Server added new users successfully in the database table.	Pass
TU08	SignUp using the same phone number	1. Sign up using a unique number in application. 2. Use the same data to sign up again.	Same phone number should be used to sign up two times.	"Phone number already registered" message must appear in the server terminal.	Message displayed on the terminal.	Pass
TU09	While SignUp add an image having two faces.	1. Add an image with two faces	Photo with two faces must be used.	"Something went wrong" message must appear.	Something went wrong appears on the terminal.	Pass
TU10	If image with no face is added while sign up	1.Add an image having no face	Photo without a face must be added.	"No face present in image" message must appear.	No face present in image appears on terminal	Pass
TU11	Phone number that is not registered is used to login	1.Go directly to login and enter a number which is not registered	Unregistered phone number is inserted.	"Phone number does not exist" message should be shown.	Phone number does not exist is displayed in terminal.	Pass
TU12	Face is not matched during payment.	1. Face of some other person is kept infront of camera	Face of an unauthorized person is kept infront of camera	Server should identify correct user	Face does not match message appears.	Pass
TU13	If connection breaks in middle of transaction	1. While transaction disconnect from the network.	NA	Server should rollback the actions it took on database	Database is not updated.	Pass

VIII. RESULTS

Followig are the results of testing the application by users. The success rate found from 10 trials of each test scenario.

TABLE IIIII
RESULT TABLE

	Test Scenario	Result
1.	Add image to database using Postman	10/10 times.
2.	Sign Up with correct credentials	9/10 times.
3.	Login with correct credentials	7/10 times.
4.	Pay after successful facial-recognition	8/10 times.
5.	Transaction roll-back when app is disconnected from the network	10/10 times.

IX. FUTURE WORK

We are willing to add another module for liveness detection in this project, since the current system is not completely fool-proof. It can be fooled by using a photo copy infront of camera. Thus liveness detection is needed.

X. CONCLUSION

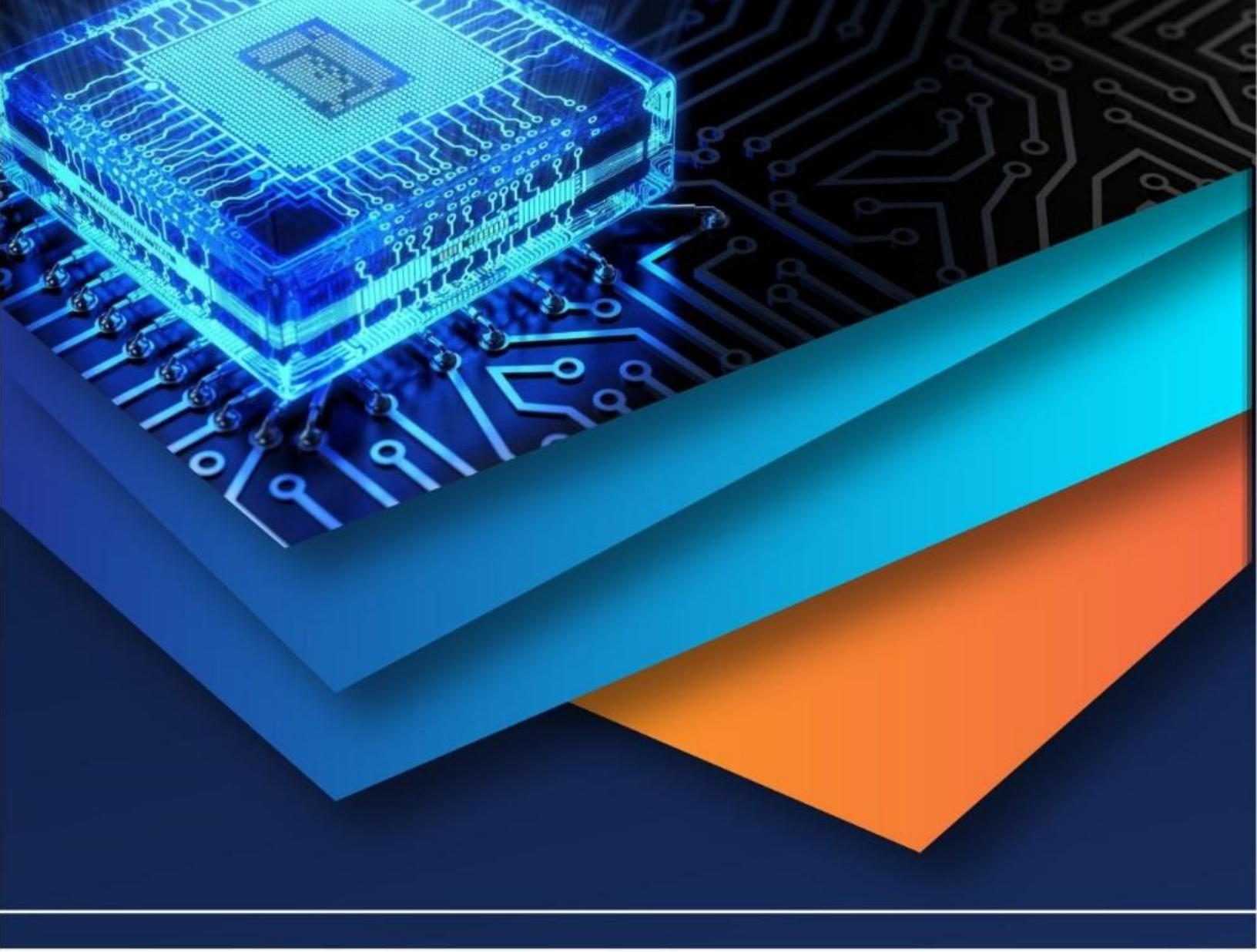
In future people will tend to go hands free, and this proposed system has the such potential. The UID based on face-recognition will soon become a trend when a robust system will be made. Since data is already collected, the system is easy to implement as it has no need to collect the data again. This would revolutionize the whole payment system.

XI. ACKNOWLEDGEMENT

We would like to thank Prof. Shikha Pachouly Ma'am, under whose guidance we were able to come up with this idea of creating a payment interface. We would also like to thank our family members and friends who helped us throughout thi s project.

REFERENCES

[1] Guosheng Hu, Yongxin Yang, Dong Yi, Josef Kittler, William Christmas, Stan Z. Li, Timothy Hospedales, "When Face Recognition Meets with Deep Learning: An Evaluation of Convolutional Neural Networks for Face Recognition", 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), 7-13 Dec. 2015, Santiago, Chile, Pages: 384 - 392.
[2] "Local Binary Pattern Network: A Deep Learning Approach for Face recognition", 2016 IEEE International Conference on Image Processing (ICIP), 25-28 Sept. 2016, Phoenix, AZ, USA, Pages: 3224 - 3228
[3] Guosheng Hu, Xiaojiang Peng, Yongxin Yang, Timothy M. Hospedales, Jakob Verbeek, "Frankenstein: Learning Deep Face Representations using Small Data", IEEE Transactions on Image Processing, 26 September 2017, Pages: 293 - 303, Volume: 27, Issue: 1.
[4] Yutong Zheng, Chenchen Zhu, Khoa Luu, Chandrasekhar Bhagavatula, T. Hoang Ngan Le, Marios Savvides, "Towards a deep learning framework for unconstrained face detection", 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS), 6-9 Sept. 2016, Niagara Falls, NY, USA, Pages: 1 - 8.
[5] Yu Chen, Wei Luo, Jian Yang, "Facial landmark detection via pose-induced auto-encoder networks", 2015 IEEE International Conference on Image Processing (ICIP), 27-30 Sept. 2015, Quebec City, QC, Canada, Pages: 2115 - 2119.
[6] Binyam Tesfahun Liyew "Applying a Deep Learning Convolutional Neural Network (CNN) Approach for Building a Face Recognition System: A Review", JETIR (ISSN-2349-5162), December 2017, Volume 4, Issue 12.
[7] Musab Coskun, Aysegul Ucar, Yakup Demir, "Face Recognition Based on Convolutional Neural Network ", IEEE Papers on Image Processing, November 2018.
[8] Liping Yuan , Zhiyi Qu , Yufeng Zhao, Hongshuai Zhang, Qing Nian, "A Convolutional Neural Network based on TensorFlow for Face Recognition ".



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY