ONLINE TRANSACTION SECURITY USING FACE RECOGNITION

Sankalp Patil*1, Prakhar Jain*2

*1Professor. Anurag Punde, Department Of Computer Science And Engineering, Acropolis Institute Of Technology And Research, Indore, M.P, India.

ABSTRACT
Credit/debit cards are widely being used all over the world. There is an assumption, that if one has a credit card and passes verification, the access is granted. Such an approach entails few threats: using of credit card being in possession of unauthorized people (stolen or just borrowed), risk of cloning card. One of the solutions is verification of the biometric linkage between the signed facial image of the credit card holder embedded in credit card and the user’s facial image captured by a webcam during usage of the card. Our goal is to compare the similarity of faces embedded in the credit card and user before the final step of the transaction is finished.

Keywords: Convolutional Neural Networks, Common Objects In Context, Open-Source Computer Vision, JavaScript Object Notation, Count In Frame, Graphical Processing Unit, YOLO- You Only Look Once.

I. INTRODUCTION
Since information of users can leak easily, many authentication approaches are proposed to secure data and systems. Authentication method should be able to “declare anyone that they are who they claim to be”. However, it is not important in authentication process that what identity is authorized for. Moreover, they can be disclosed and stolen by intimates or colleagues. Some systems use an object such as passport or ID card to identify a user. It is practical in use but also is very easy to be stolen or copied. Therefore, biometrics which is unforgettable and more unique becomes a choice for secure process. Biometric technologies can be fingerprint, face, iris, palm print or even hand geometry. Among these technologies, face image seems to be one of the most popular used features due to its usability, collectability and acceptability. For usability and collectability, face technology is non-intrusive technology as test subjects are not required for a direct contact. For acceptability, many studies claim that their correct recognition rates are more than 95%. Other biometrics such as fingerprints yields high matching accuracy rate as well. However, fingerprint can be copied by using plastic sheet and created 3-dimension fake fingerprint. Moreover, fingerprint images that are stored in a database need to be high-quality images for accuracy of recognition process, so this takes a lot of capacity in database in case of applying to a system with a very large number of persons. On the other hand, iris scanning seems to be the most secure and accurate method for biometric technologies. However, iris recognition machines are expensive. Here we are using Local Binary Pattern Histogram (LBPH) as our base face recognition algorithm.

This algorithm is efficient and has been customized for our specific purpose and more efficient face recognition. It is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighbourhood of each pixel and considers the result as a binary number. We implement the LBP using OpenCV. OpenCV was started at Intel in 1999 by Gary Bradsky and the first release came out in 2000. Right now, OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning and it is expanding day-by-day. Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc and is available on different platforms including Windows, Linux, OS X, Android, iOS etc. Also, interfaces based on CUDA and OpenCL are also under active development for high-speed GPU operations. OpenCV-Python is the Python API of OpenCV. It combines the best qualities of OpenCV C++ API and Python language. Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability exceeding the accuracy of manual recognition.

II. LITERATURE REVIEW
After experiencing many ecommerce websites/apps and online transaction apps and other apps or websites which involves money like Flipkart, Amazon, Paytm, Phone pe and Google pay, we have come to the conclusion that, A simple multi-factor authentication setup involves asking a user for their username and password (something they know) as well as verifying their identity through a second factor such as an SMS message to their phone (something they have). That covers two factors of authentication, but adding in image recognition...
as well adds an extra layer of security to the login process without making it frustrating or overly complicated for authorized users. Many banks use image recognition as part of the multi-factor authentication process so their customers can login to their accounts and authorize various financial transactions in a secure environment. On the web, this sort of image recognition authentication is ideal for preventing phishing attacks where another website could mimic the look and feel of your bank. Even if a phishing site looked identical to your bank’s log in page, it would not know what your authentication image looked like, and would not be able to fool users as easily on that page of the process. Also, that sort of phishing attack would render the bank username and password useless. Even with the password, the data thief would not be able to log in to the account without also knowing the authentication image, which they would not be able to steal in a phishing scheme. Facebook also uses a form of image authentication. The site uses this feature primarily during the first time you log in to an account from a previously unrecognized location or computer. Instead of asking you to answer a security question you previously submitted an answer to, Facebook asks you to name a number of your friends by their profile pictures. This sort of image recognition would deny access to your account unless the hacker was also familiar with the identities of your friends. Image-based authentication (IBA) can solve many of the common problems with passwords. IBA as a password substitute works primarily by accepting a valid username and then presenting an image set. The user than must identify the correct image or images from the set to be allowed into the account. Another image-based authentication factor could be something like implementing captcha. That can effectively reduce issues of automated brute force attacks since machines are not able to solve the CAPTCHA images reliably. Identifying the person entering a password as human is a strong first step towards security. Even in the apps/websites which asks for KYC (Know the Customer) it do asks for the facial identity of the customer but it's also a onetime thing, it does not ask to verify the facial identity every time you use your debit/credit card to pay the money or transfer it.

III. METHODOLOGY

Proposed system

1. Dataset Classification
2. Building the Face Recognition By using LBPH
3. Training our Network
4. Testing

1. **DataSet Classification:**
   - **Image pre-processing**
     In these phases, we require better resolution images and with better quality. All these images are resized with specific manner and resolution. These images we remove noise content and rotate the images using a data augmentation process.
1. datasetCreate

```python
import cv2

camera = cv2.VideoCapture(0)
detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
Id = input('enter your id: ')
sampleNum = 0
while (True):
    ret, img = camera.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = detector.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        cv2.rectangle(img, (x,y),(x+w,y+h),(255,0,0),2)
        sampleNum = sampleNum + 1
        cv2.imwrite("dataset/User." + Id + '+' + str(sampleNum) + ".jpg",
                    gray[y:y+h,x:x+w])
        cv2.imshow('frame', img)
    if cv2.waitKey(100) & 0xFF == ord('q'):
        break
    elif sampleNum > 100:
        break
    camera.release()
    cv2.destroyAllWindows()
```

**Figure 1:** Code for Creating Dataset

● **Feature extraction**

In this feature extraction process, extract some of the important features of the defected leaf. It can create colored structure and convert the RGB color to gray scale or vice versa of the provided image. The feature we can use to train our system.

2. Building the Face Recognition using LBPH & Training our network

```python
images = os.listdir(path)
for imagePath in images:
    f,d = os.path.splitext(imagePath)
    img = cv2.imread(os.path.join(path, imagePath), cv2.IMREAD_GRAYSCALE)
    faces.append(f)
```

**Figure 2:** Code of Feature Extraction from the Dataset images
Code For the Face Recognition:

```python
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read("trainer/trainer.yml")
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);

cam = cv2.VideoCapture(0)
while(cam.isOpened()):
    ret, im = cam.read()
    gray = cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray, 1.2,5)
    for (x,y,w,h) in faces:
        Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
        if(conf<100):
            cam.release()
            cv2.destroyWindow("im")
            cherry()
            break;
        else:
            Id="Unknown"
            cam.release() |
            cv2.destroyAllWindows("im")
            cherry()
            break;
    cv2.imshow("im",im)
    if cv2.waitKey(100) & 0xFF==ord('q'):
        break
```

3. Testing

Testing is the process of evaluation of a system to detect differences between given input and expected output and also to assess the feature of the system. Testing assesses the quality of the product. It is a process that is done during the development process.

**Strategy Used**

Tests can be conducted based on two approaches –
- Functionality testing
- Implementation testing

The testing method used here is Black Box Testing. It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

**Test Case and Analysis**

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>TCO01</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test Case Summary</strong></td>
<td>It will check whether the system detects the face with accuracy $\geq 50%$ or not.</td>
</tr>
<tr>
<td><strong>Test Procedure</strong></td>
<td>Run the program and start the camera</td>
</tr>
<tr>
<td><strong>Expected Result</strong></td>
<td>The face must be detected with accuracy greater than $50%$.</td>
</tr>
<tr>
<td><strong>Actual Result</strong></td>
<td>The students with accuracy greater than 50% are detected.</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>Pass</td>
</tr>
</tbody>
</table>
Test Case Output 1:

![FacePay interface](image)

Amount to be paid: **Rs. 1000**

**Payment Details**

- **Card Number**
- **MM**
- **YY**
- **CVV**

**Pay**

Hello 123456789,
Your Current Balance is: **Rs. 77500**

![Face recognition interface](image)

**Figure 3:** Successfully recognized Face
## TEST CASE: 2

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>TC002</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case Summary</td>
<td>It will check whether the transaction is completing successfully or not</td>
</tr>
</tbody>
</table>
| Test Procedure | 1. Run the project  
2. Check the updated balance. |
| Expected Result | result should be updated by deducted amount |
| Actual Result | Result is updated successfully |
| Status        | Pass  |
Project Description

Data flow diagram

Figure 4: Data Flow diagram

Block Diagram of The Proposed System:

<table>
<thead>
<tr>
<th>Getting each pixel image as a block in a matrix</th>
<th>Threshold each matrix pixel with the center pixel of the image</th>
<th>Result in binary format</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Result in Decimal number format</td>
</tr>
</tbody>
</table>

Figure 5: Block Diagram

Tools Used:

OpenCV

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

Hardware

- Processor: Intel Duo Core
- RAM: 4 GB
- GPU: Internal GPU or above
- Camera: System camera or 5 MP or above

Software
Operating System: Ubuntu (14) (64bit), Windows 7 or above.

Programming Languages: Python, JavaScript, CSS, html5, PHP.

Limitations: - If the Camera’s resolution is low or the lighting conditions of the surrounding are not favorable, they it may provide the undesired outcome, resulting in errors.

Activity diagram

IV. RESULT DISCUSSION

This system can convert the less security into more security transaction. This system creates a more reliable communication means user and bank system. Also, it generates face at the end of transaction which is verify by various algorithm.

1. Payment system for online transaction is based on face recognition provides authenticate user data privacy and prevents misuse of data in world.
2. This technique identify theft and prevent customer data which improve security.
3. This system can convert the less security into more security transaction. This system creates a more reliable communication means user and bank system.
4. This is useful to reduce or totally remove fake transaction, i.e to read OPT then it required face of user which is provide for transaction. Then the transaction is done.
5. It will also go a long way in ensuring that innocent people are not wrongly arrested based on previous crimes.

V. CONCLUSION

This system can convert the less security into more security transaction. This system create a more reliable communication means user and bank system. Also it generate face at the end of transaction which is verify by various algorithm.

1. Payment system for online transaction is based on face recognition provides authenticate user data privacy and prevents misuse of data in world.
2. This technique identify theft and prevent customer data which improve security.
3. This system can convert the less security into more security transaction. This system create a more reliable communication means user and bank system.
4. This is useful to reduce or totally remove fake transaction, i.e to read OPT then it required face of user which is provide for transaction. Then the transaction is done.
5. It will also go a long way in ensuring that innocent people are not wrongly arrested based on previous crime.
ACKNOWLEDGEMENT

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely. There are number of people without whom this projects work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor Prof. Anurag Punde, Professor, AITR, Indore for his motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to Dr Kamal Kumar Sethi, HOD IT, AITR Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of IT Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if do not acknowledge the support and guidance received from Dr S C Sharma, Director, AITR, Indore whenever needed. We take opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for project to achieve our objectives.

We are grateful to our parent and family members who have always loved and supported us unconditionally. To all of them, we want to say “Thank you”, for being the best family that one could ever have and without whom none of this would have been possible.

VI. REFERENCES


