

DOCFACE:ID Document TO Selfie Matching

Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

P.M.V.VIVEK (38110653)

Y.ROHITH (38110668)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119**

APRIL – 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **P.M.V.VIVEK (38110653)** and **Y.ROHITH (38110668)** who carried out the project entitled "**ID DOCUMENT TO SELFIE MATCHING**" under my supervision from November 2021 to March 2022.

Internal Guide

Dr.Rajalakshmi,M.E,(Ph.D.) CSE

Head of the Department

Dr. S. Vigneshwari M.E., Ph.D., and Dr. L. Lakshmanan M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **P.M.V.VIVEK (Reg No:38110653)** and **Y.ROHITH (Reg No: 38110668)** hereby declare that the Project Report entitled "**ID DOCUMENT TO SELFIE MATCHING**" done by us under the guidance of **Dr.Rajalakshmi,M.E,(Ph.D.) CSE.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in 2018-2022.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E.,Ph.D., Dean**, School of Computing **Dr.S.Vigneshwari M.E., Ph.D.** and **Dr.L.Lakshmanan M.E., Ph.D.** , Heads of the Department of Computer Science and Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and a deep sense of gratitude to my Project Guide **Dr.Rajalakshmi,M.E,(Ph.D.) CSE.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Matching live images (“selfies”) to images from ID documents is a problem that can arise in various applications. A challenging instance of the problem arises when the face image on the ID document is from early adolescence and the live image is from later adolescence. We explore this problem using a private dataset called Chilean Young Adult (CHIYA) dataset, where we match live face images taken at age 18-19 to face images on scanned ID documents created at ages 9 to 18. State-of-the-art deep learning face matchers (e.g., ArcFace) have relatively poor accuracy for document-to-selfie face matching. To achieve higher accuracy, we fine-tune the best available open-source model with triplet loss for a few-shot learning. Experiments show that our approach achieves higher accuracy than the DocFace+ model recently developed for this problem. Our fine-tuned model was able to improve the true acceptance rate for the most difficult (largest age span) subset from 62.92% to 96.67% at a false acceptance rate of 0.01%. Our fine-tuned model is available for use by other researchers.

TABLE OF CONTENTS

Chapter No.	TITL E	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	11
	2.1. RELATED WORK	12
3	METHODOLOGY	13
	3.1. EXISTING SYSTEM	13
	3.2. PROPOSED SYSTEM	13
	3.3. OBJECTIVE	8
	3.4. SOFTWARE AND HARDWARE REQUIREMENTS	14
	3.4.1. SOFTWARE REQUIREMENTS	14
	3.4.2. HARDWARE REQUIREMENTS	14
	3.5. SYSTEM ARCHITECTURE	15
	3.7. ALGORITHMS USED	15
	3.7.1. NAIVE BAYES	16
	3.7.2. DECISION TREE	16

	3.8 MODULES	16
	3.8.1 USER REGISTRATION	17
	3.8.2. USER UPLOAD CERTIFICATES	17
	3.8.3. GET CERTIFICATES	17
	3.8.4 .QR REQUEST	
4	RESULTS AND DISCUSSION	18
	4.1. WORKING	18
5	CONCLUSION	19
	5.1. CONCLUSION	19
	REFERENCES	16
	APPENDICES	17
	A. SOURCE CODE	22
	B. SCREENSHOTS	16
	C. PLAGIARISM REPORT	
		32
	D. JOURNAL PAPER	33

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
KNN	K-Nearest Neighbour
RSA	Rivest-Shamir-Adl eman
SH-256	Secure Hash Algorithms
J2EE	Java 2 platform enterprise edition

CHAPTER 1

INTRODUCTION

An application scenario of the DocFace+ system. The kiosk scans the ID document photos or reads the photo from the embedded chip and the camera takes another photo of the holder's live face (selfie). By comparing the two photos, the system decides whether the holder is indeed the owner of the ID document. As we all know the saying which tells that "Prevention of the disease at an early stage is much better than the cure which we take after we get affected by the disease".

ePassport chips containing their digital photos and then capture their face images using a camera mounted at the SmartGate. After verifying a traveler's identity by face comparison, the gate is automatically opened for the traveler to enter Australia. Similar machines have also been installed in the UK (ePassport gates) [3], USA (US Automated Passport Control) [4] and other countries. In China, such verification systems have been deployed in various locations, including train stations, for matching Chinese ID cards with live faces [5]. In addition to international border control, some businesses [6] [7] are utilizing face recognition solutions to ID document verification for online services. This project helps to get the idea about the disease of an individual based on the symptoms he/she have, and get the treatment easily by contacting the concern doctor.

A practical and common approach to this problem involves comparing an individual's live face to the face image found in his/her ID document. For example, immigration and customs officials look at the passport photo to confirm a traveler's identity. Clerks at supermarkets in the United States look at the customer's face and driver license to check his/her age when the customer is purchasing alcohol. Instances of ID document photo matching can be found in numerous scenarios

1.1

1.2 OVERVIEW

Numerous activities in our daily life require us to verify who we are by showing our ID documents containing face images, such as passports and driver licenses, to human operators. However, this process is slow, labor intensive and unreliable. As such, an automated system for matching ID document photos to live face images (selfies) in real time and with high accuracy is required. In this paper, we propose DocFace+ to meet this objective. We first show that gradient-based optimization methods converge slowly (due to the underfitting of classifier weights) when many classes have very few samples, a characteristic of existing ID-selfie datasets. To overcome this shortcoming, to update the classifier weights, which allows faster convergence and more generalizable representations.

Identity verification plays an important role in our daily lives. For example, access control, physical security and international border crossing require us to verify our access (security) level and our identities. to verify who we are by showing our ID documents containing face images, such as passports and driver licenses, to human operators. However, this process is slow, labor intensive and unreliable. As such, an automated system for matching ID

Data are stored in different nodes, and anyone who wishes to modify a particular internal datum must request that other nodes modify it simultaneously. Thus, the system is highly reliable. We developed a decentralized application and designed a certificate system based onEthereumblockchain.

CHAPTER 2

LITERATURE SURVEY

2.1 RELATED WORK

To the best of our knowledge, the first study on ID-selfie matching is attributed to Starovoitov et al. [9] [12]. Assuming all face images are frontal faces without large expression variations, the authors first localize the eyes with Hough Transform. Based on eye locations, the face region is cropped and gradient maps are computed as feature maps.

scanned document photos, and a high quality live face images. To eliminate the degradation caused by scanning, Bourlai et al. inserted an image restoration phase before comparing the photos using a general face matcher. In particular, they train a classifier to classify the degradation type for a given image, and then apply degradation-specific filters to restore the degraded images!

the document photos in our datasets are read from the chips embedded in the Chinese ID Cards. Additionally, our method is not designed for any specific degradation type and could be applied to any type of ID document photos. Concurrent with our prior work.

Another field related to our work is the low-shot learning problem. In low-shot learning [36] [37] [38], a model is trained in such a way that it is able to generalize to unseen classes, which may have only a few samples. There are two training phases in lowshot learning: the model, or learner, is first trained on a larger classification dataset, and then in testing..

Heterogeneous face recognition (HFR) is an emerging topic that has become popular in the past few years [23]. It usually refers to face recognition between two different modalities, including visible spectrum images (VIS), near infrared images (NIR) [24], thermal infrared images [25], composite sketches [26], etc. IDselfie matching can be considered to be a special case of HFR since the images to be matched come from two different sources and thus need possibly different modeling approaches.

Another field related to our work is the low-shot learning problem. In low-shot learning [36] [37] [38], a model is trained in such a way that it is able to generalize to unseen classes, which may have only a few samples. There are two training phases in lowshot learning: the model, or learner, is first trained on a larger classification dataset, and then in testing, a few labeled samples of new classes are given and the model is required to learn a new classifier given these classes. The adjective “low-shot” refers to a small number of images per class

This problem has been receiving growing interest from the machine learning community because humans are very good at adapting to new types of objects (classes) while conventional deep learning methods require abundant samples for discriminating a specific class from others. This is related to the ID-selfie problem since most of the identities only have a few samples, resulting in a shallow dataset.

Many methods have been proposed for low-shot learning problem. Koch et al. [36] proposed a simple yet effective approach by learning metrics via siamese network [39] for one-shot recognition. Vinyals et al. [37] proposed the Matching Net where they simulate the testing scenario in the training phase by learning low-shot recognition in mini-batches.

[42], based on the idea of the prototypes, or proxies [43], proposed to imprint the weights of a new classifier with extracted features. We note that their work differs from ours as they utilized the imprinted weights simply as initialization while we use weight imprinting as an optimization method throughout training by dynamically imprinting.

CHAPTER 3

METHODOLOGY

3.1 EXISTING SYSTEM

In the existing system, Identity verification plays an important role in our daily lives. For example, access control, physical security and international border crossing require us to verify our access (security) level and our identities. to verify who we are by showing our ID documents containing face images, such as passports and driver licenses, to human operators. However, this process is slow, labor intensive and unreliable.

document photos to live face images (selfies) in real time and with high accuracy is required. After verifying a traveler's identity by face comparison, the gate is automatically opened for the traveler to enter. For ID selfie matching, they are comparing a scanned or digital document photo Also.

3.2 PROPOSED SYSTEM

We are proposing a certificate system based on blockchain to overcome the problem. Data are stored in different nodes, and anyone who wishes to modify a particular internal datum must request that other nodes modify it simultaneously. Thus, the system is highly reliable. We developed a decentralized application and designed a certificate system based on Ethereum blockchain.

The system saves on paper, cuts management costs, prevents document forgery, and provides accurate and reliable information on digital certificates and compare user live face with verified document face.

3.3 OBJECTIVE OF PROJECT

The main aim of this project is to solve the problem of counterfeiting certificates we are proposing an digital certificate system based on block chain technology and to verify the traveler's identity using live camera, which allows faster convergence and more generalizable representations.

3.4 SOFTWARE AND HARDWARE REQUIREMENTS

3.4.1 *Software Requirements:*

- ✓ Windows 7 and above
- ✓ JDK 1.7
- ✓ J2EE
- ✓ Tomcat 7.0
- ✓ My Sql

3.4.2 *Hardware Requirements:*

- ✓ Processor: Intel Core i5
- ✓ RAM: 8GB
- ✓ OS: Windows

3.4.3 Algorithm used :

KNN

RSA

SHA-256

3.4.4 Technology Used:

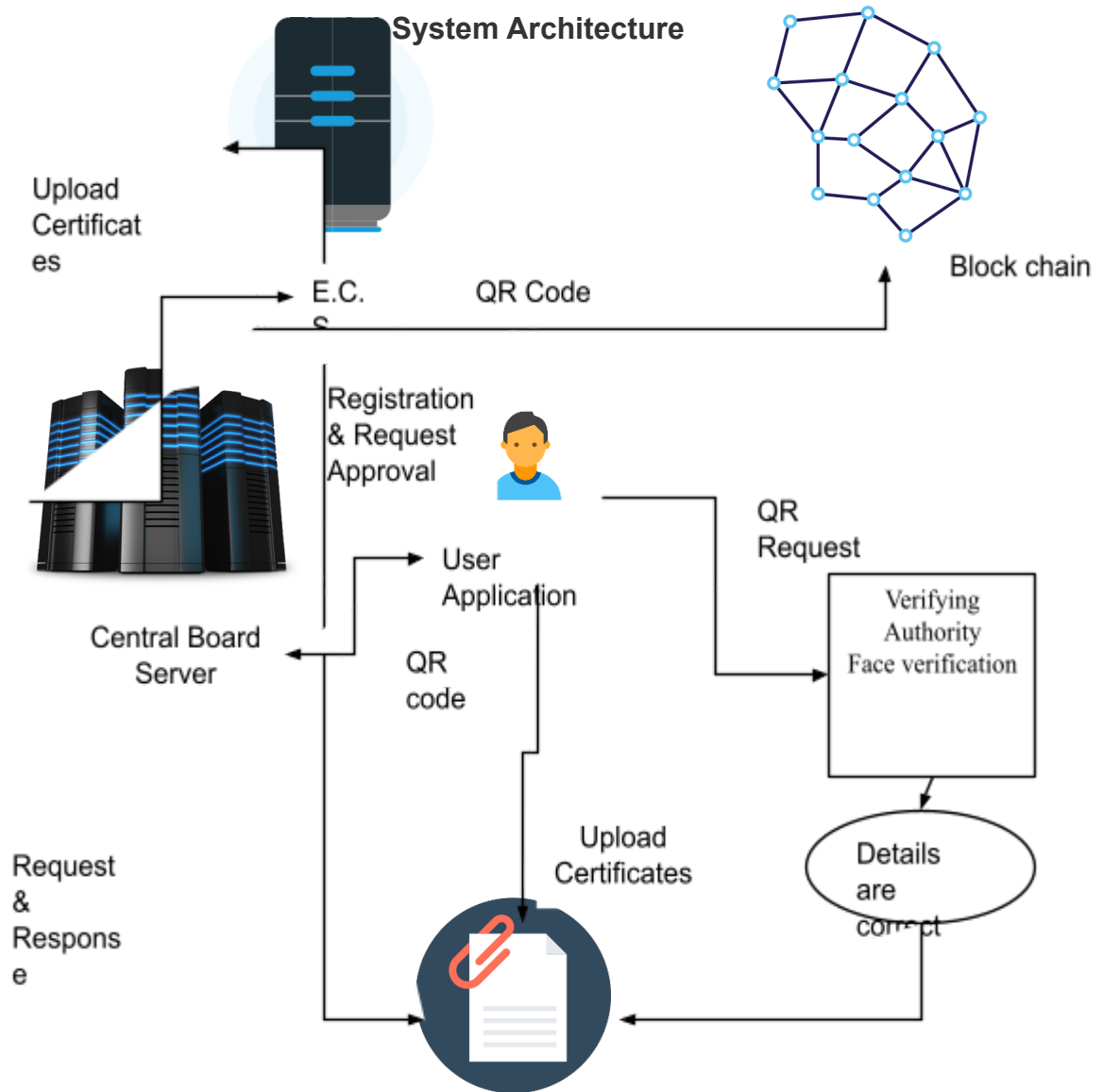
J2EE (JSP, Servlet), JavaScript, HTML, CSS, AJAX.

Hibernate Framework

MVC Pattern

Design Pattern

3.5. SYSTEM ARCHITECTURE



3.7.1 DECISION TREE

Decision Tree is an algorithm whose input and output are known. The information is divided repeatedly using the particular parameter.

Decision tree consists of two main parts namely decision node and leaf nodes. The decision nodes specify the decision at which the parameter should be split.

The leaf nodes are the output brought by the decisions. A decision tree asks for either true or false to divide the data.

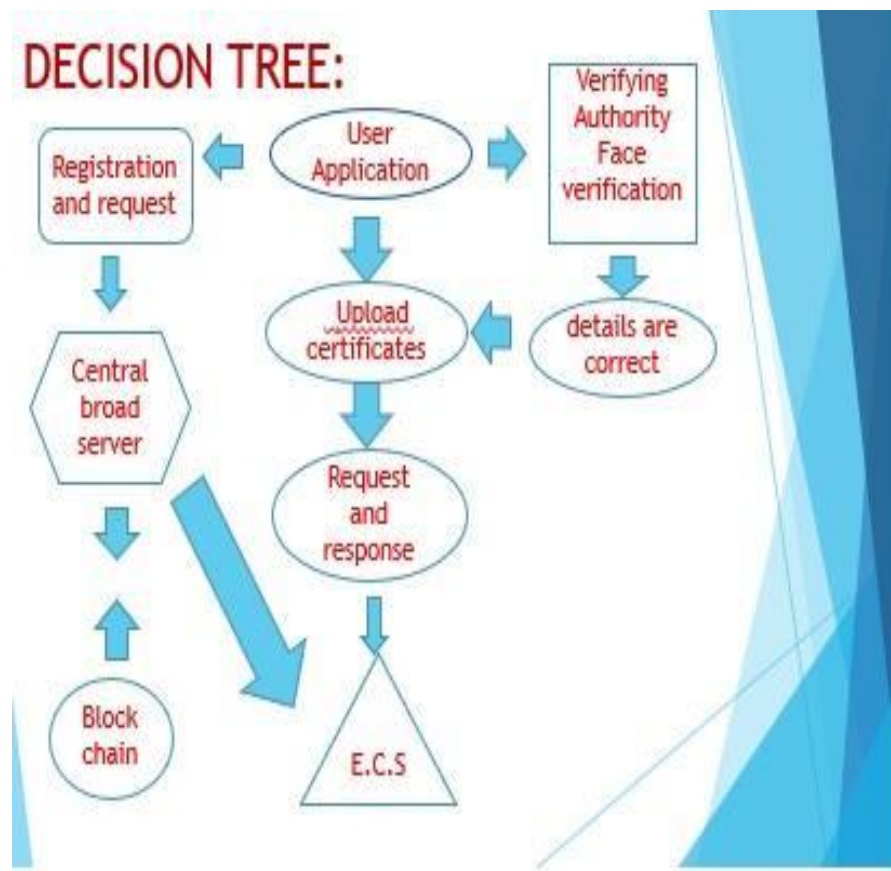


Fig 3.2. Decision Tree

3.8. MODULES

Project contains four parts:

3.8.1 USER REGISTRATION AND AUTHENTICATION.

3.8.2 USER UPLOAD CERTIFICATE.

3.8.3 GET CERTIFICATE

3.8.4 QR REQUEST AND RESPONSE FROM VERIFICATION AUTHORITY.

User Registration And Authentication:

In this module user needs to registers into his application and a request will be sent to central board server for authentication. Unless the central board server approves the request user cannot login into his account. When central board server approves the request a key will be generated and user can login into his account.

User Upload Certificate:

After user login into his account he needs to upload certificates namely pan card, aadhar card, voter id, ssc certificates to central board server. Central board server will review the certificates and accepts or decline the certificates. If central board server accepts the certificate those details will be stored in E.C.S and Blockchain. If central board server declines the certificate it won't be stored in E.C.S. or Block Chain.

Get Certificate:

If user needs a certificate he will send request to central board server. If central board server found the user details to be genuine he accepts the request and forward a request to E.C.S where all the certificates will be there. E.C.S. responds for the request and certificates will be provided to the user.

QR Request and Face verification:

If user wants to apply for any certificates he will send request to central board server and central board server will check the details and forward the request to E.C.S. E.C.S will generate the QR Code and forwarded to user via central board server. User forwards the QR code to the verifying authority and if all details are correct and face matches with live face Verifying authority will issue the document.

CHAPTER 4

RESULTS AND DISCUSSION

- The key concept behind this project is to build a platform that will play a significant role in verifying the authenticity of important files, contracts, certificates, land/property/asset documents more accurately and quickly.
- We have used very well-known and secured methods to build this system. We have used Ethereum blockchain to develop this system. SHA-256 is used along with blockchain. It is a one-way and collision-resistant encryption algorithm to encrypt the data
- The classification error obtained using variants of each architecture. A dash indicates a failure to converge given a large range of hyper-parameters. The tolerant architectures combined with the weighted ℓ_1 distance are able to achieve significantly better results than other methods. The best architecture reduced the error from 37.3% for a Siamese network with shared parameters to 28.6% for a single-stream learned.
- While the Euclidean distance gave the best results for Siamese networks with shared parameters, better results were achieved by learners (and Siamese networks with unshared parameters) using a weighted ℓ_1 distance. In fact, none of the alternative architectures are able to achieve lower error under the Euclidean distance than the shared Siamese net. The dot product was, in general, less effective than the other two metrics.

4.1. WORKING

propose a new face recognition system, named DocFace+, for matching ID document photos to selfies. The transfer learning technique is used, where a base model for unconstrained face recognition is fine-tuned on a private IDselfie dataset. A pair of sibling networks with shared high-level modules are used to model domain-specific parameters. Based on our observation of the weight-shift problem of classificationbased embedding learning loss functions on shallow

datasets, we propose an alternative optimization method, called dynamic weight imprinting (DWI) and a variant of AM-Softmax, DIAMSoftmax

Algorithm	Accuracy
Decision Tree	97%
Naive Bayes	100%

Table 4.1 Accuracies

CHAPTER

5

CONCLUSI

ON

5.1. CONCLUSION

- ▶ In this work, we have shown that it is possible to obtain the parameters of a deep neural network using a single, feed-forward prediction from a second network. This approach is desirable when iterative methods are too slow, and when large sets of annotated training samples are not available.
- ▶ We have demonstrated the feasibility of feed-forward parameter prediction in two demanding one-shot learning tasks in OCR and visual tracking. Our results hint at a promising avenue of research in "learning to learn" by solving millions of small discriminative problems in an offline phase. Possible extensions include domain adaptation and sharing a single learned between different pupil networks.

REFERENCES

- [1] D. White, R. I. Kemp, R. Jenkins, M. Matheson, and A. M. Burton, "Passport officers errors in face matching," PloS one, 2014.
- [2] Wikipedia, "Australia smartgate," [https://en.wikipedia.org/wiki/ SmartGate](https://en.wikipedia.org/wiki/SmartGate), 2018
- [3] Wikipedia, "epassport gates," [https://en.wikipedia.org/wiki/EPassport gates](https://en.wikipedia.org/wiki/EPassport_gates), 2018.
- [4] U.S. Customs and Border Protection, "Automated passport control (apc)," <https://www.cbp.gov/travel/us-citizens/apc>, 2018.
- [5] Xinjiang Heng An Perimeter Security Equipment Co., "What is id-person matching?" [http://www.xjhazj.com/xjhazj/vip doc/8380983.html](http://www.xjhazj.com/xjhazj/vip_doc/8380983.html), 2018.
- [6] Jumio, "Netverify id verification," [https://www.jumio.com/ trusted-identity/netverify](https://www.jumio.com/trusted-identity/netverify), 2018.
- [7] Mitek, "Mitek id verification," [https://www.miteksystems.com/ mobile-verify](https://www.miteksystems.com/mobile-verify), 2018.
- [8] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.
- [9] V. Starovoitov, D. Samal, and B. Sankur, "Matching of faces in camera images and document photographs," in ICASSP, 2000.
- [10] T. Bourlai, A. Ross, and A. Jain, "On matching digital face images against scanned passport photos," in IEEE Int. Conf. Biometrics, Identity and Security (BIDS), 2009.
- [11] T. Bourlai, A. Ross, and A. K. Jain, "Restoring degraded face images: A case study in matching faxed, printed, and scanned photos," IEEE Trans

APPENDICES

A) SOURCE CODE

CertificatesPojo - Notepad

File Edit Format View Help

```
package model;

import java.io.Serializable;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "certificates")
public class CertificatesPojo implements Serializable {
    @Id
    @GeneratedValue
    private long id;
    String email, panCard, panStatus, aadharCard, aadharStatus, voterId,
        voterIDStatus, sslc, sslcStatus, status;

    public CertificatesPojo(String email, String panCard, String panStatus,
        String aadharCard, String aadharStatus, String voterId,
        String voterIDStatus, String sslc, String sslcStatus, String status) {
        super();
        this.email = email;
        this.panCard = panCard;
        this.panStatus = panStatus;
        this.aadharCard = aadharCard;
        this.aadharStatus = aadharStatus;
        this.voterId = voterId;
        this.voterIDStatus = voterIDStatus;
        this.sslc = sslc;
        this.sslcStatus = sslcStatus;
        this.status = status;
    }

    public CertificatesPojo() {
        // TODO Auto-generated constructor stub
    }

    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id = id;
    }
}
```

```
public String getEmail() {  
    return email;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
  
public String getPanCard() {  
    return panCard;  
}  
  
public void setPanCard(String panCard) {  
    this.panCard = panCard;  
}  
  
public String getPanStatus() {  
    return panStatus;  
}  
  
public void setPanStatus(String panStatus) {  
    this.panStatus = panStatus;  
}  
  
public String getAadharCard() {  
    return aadharCard;  
}  
  
public void setAadharCard(String aadharCard) {  
    this.aadharCard = aadharCard;  
}  
  
public String getAadharStatus() {  
    return aadharStatus;  
}  
  
public void setAadharStatus(String aadharStatus) {  
    this.aadharStatus = aadharStatus;  
}  
  
public String getVoterId() {  
    return voterId;  
}  
  
public void setVoterId(String voterId) {
```

```
public void setAadharStatus(String aadharStatus) {  
    this.aadharStatus = aadharStatus;  
}  
  
public String getVoterId() {  
    return voterId;  
}  
  
public void setVoterId(String voterId) {  
    this.voterId = voterId;  
}  
  
public String getVoterIDStatus() {  
    return voterIDStatus;  
}  
  
public void setVoterIDStatus(String voterIDStatus) {  
    this.voterIDStatus = voterIDStatus;  
}  
  
public String getSslc() {  
    return sslc;  
}  
  
public void setSslc(String sslc) {  
    this.sslc = sslc;  
}  
  
public String getSslcStatus() {  
    return sslcStatus;  
}  
  
public void setSslcStatus(String sslcStatus) {  
    this.sslcStatus = sslcStatus;  
}  
  
public String getStatus() {  
    return status;  
}  
  
public void setStatus(String status) {  
    this.status = status;  
}  
}
```

B) SCREENSHOTS

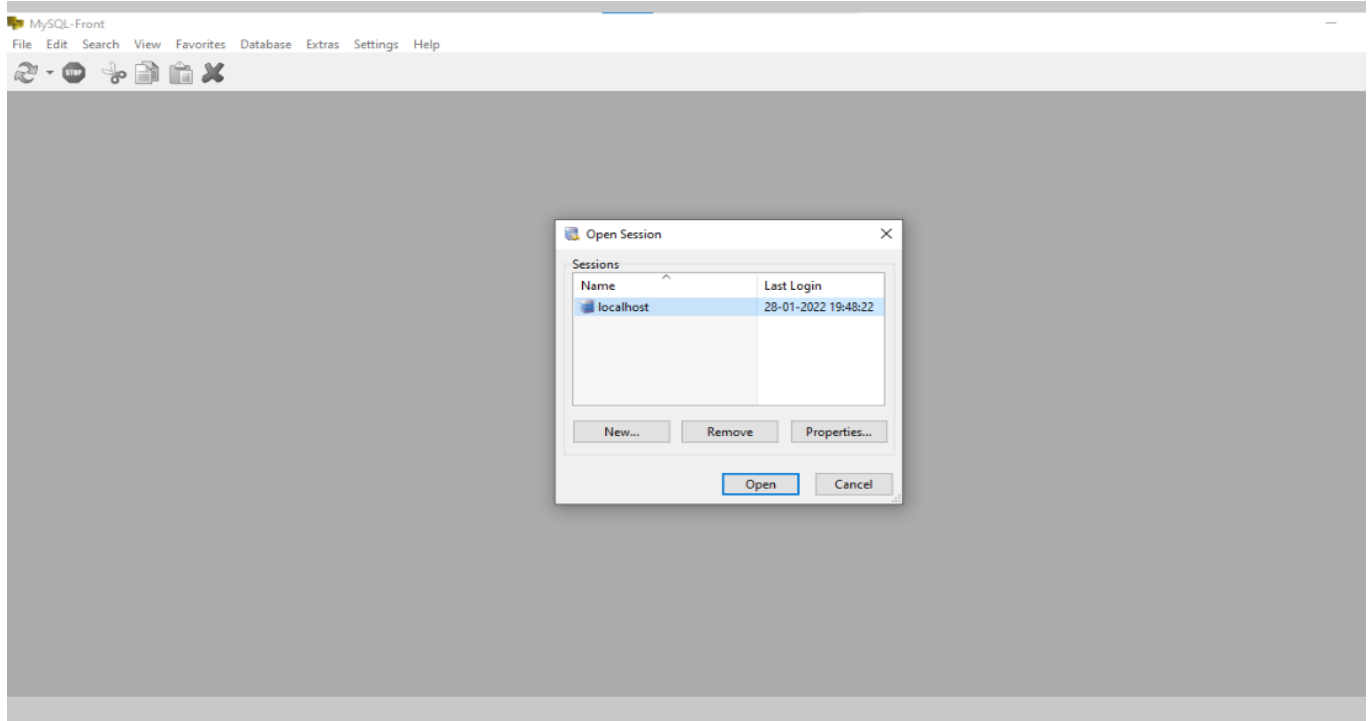


Fig B.1 Dataset

```
system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\
1\1.0\;C:\Windows\System32\OpenSSH\;C:\Program Files (x86)\Java\jdk1.7.0_01\bin
;C:\Users\dell\AppData\Local\Microsoft\WindowsApps;;
Jan 28, 2022 7:48:48 PM org.apache.coyote.http11.Http11Protocol init
INFO: Initializing Coyote HTTP/1.1 on http-9999
Jan 28, 2022 7:48:48 PM org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 432 ms
Jan 28, 2022 7:48:48 PM org.apache.catalina.core.StandardService start
INFO: Starting service Catalina
Jan 28, 2022 7:48:48 PM org.apache.catalina.core.StandardEngine start
INFO: Starting Servlet Engine: Apache Tomcat/6.0.20
Jan 28, 2022 7:48:49 PM com.sun.jersey.api.core.ClasspathResourceConfig init
INFO: Scanning for root resource and provider classes in the paths:
D:\Apache\Tomcat6.0\webapps\ECS\WEB-INF\lib
D:\Apache\Tomcat6.0\webapps\ECS\WEB-INF\classes
Jan 28, 2022 7:48:49 PM com.sun.jersey.api.core.ClasspathResourceConfig init
INFO: Root resource classes found:
class com.rest.server.CertificateRequests
Jan 28, 2022 7:48:49 PM com.sun.jersey.api.core.ClasspathResourceConfig init
INFO: Provider classes found:
Jan 28, 2022 7:48:51 PM org.apache.coyote.http11.Http11Protocol start
INFO: Starting Coyote HTTP/1.1 on http-9999
Jan 28, 2022 7:48:51 PM org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
Jan 28, 2022 7:48:51 PM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/29 config=null
Jan 28, 2022 7:48:51 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 3283 ms

Jan 28, 2022 7:02:30 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory ECSUser
Jan 28, 2022 7:02:32 PM com.sun.jersey.api.core.ClasspathResourceConfig init
INFO: Scanning for root resource and provider classes in the paths:
D:\Apache\Tomcat7.0\webapps\ECSUser\WEB-INF\lib
D:\Apache\Tomcat7.0\webapps\ECSUser\WEB-INF\classes
Jan 28, 2022 7:02:33 PM com.sun.jersey.server.impl.application.WebApplicationImpl pro
cessRootResources
SEVERE: The ResourceConfig instance does not contain any root resource classes.
Jan 28, 2022 7:02:33 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory ECSVerificationAuthority
Jan 28, 2022 7:02:35 PM com.sun.jersey.api.core.ClasspathResourceConfig init
INFO: Scanning for root resource and provider classes in the paths:
D:\Apache\Tomcat7.0\webapps\ECSVerificationAuthority\WEB-INF\lib
D:\Apache\Tomcat7.0\webapps\ECSVerificationAuthority\WEB-INF\classes
Jan 28, 2022 7:02:35 PM com.sun.jersey.api.core.ClasspathResourceConfig init
INFO: Root resource classes found:
class com.rest.service.ProofRequests
Jan 28, 2022 7:02:35 PM com.sun.jersey.api.core.ClasspathResourceConfig init
INFO: Provider classes found:
Jan 28, 2022 7:02:36 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory examples
Jan 28, 2022 7:02:36 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory host-manager
Jan 28, 2022 7:02:37 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory manager
Jan 28, 2022 7:02:37 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory ROOT
Jan 28, 2022 7:02:37 PM org.apache.coyote.AbstractProtocolHandler start
INFO: Starting ProtocolHandler ["http-bio-8888"]
Jan 28, 2022 7:02:37 PM org.apache.coyote.AbstractProtocolHandler start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
Jan 28, 2022 7:02:37 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 9132 ms
2
Started
log4j:WARN No appenders could be found for logger (org.hibernate.type.BasicTypeRegistr
ry).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Hibernate: select this_.id as id0_0_, this_.confirmPassword as confirmP2_0_0_, this_.
email as email0_0_, this_.name as name0_0_, this_.password as password0_0_, this_.uid
as uid0_0_ from user this_ where this_.uid=?
```

Fig B.2 CODE RUNNING

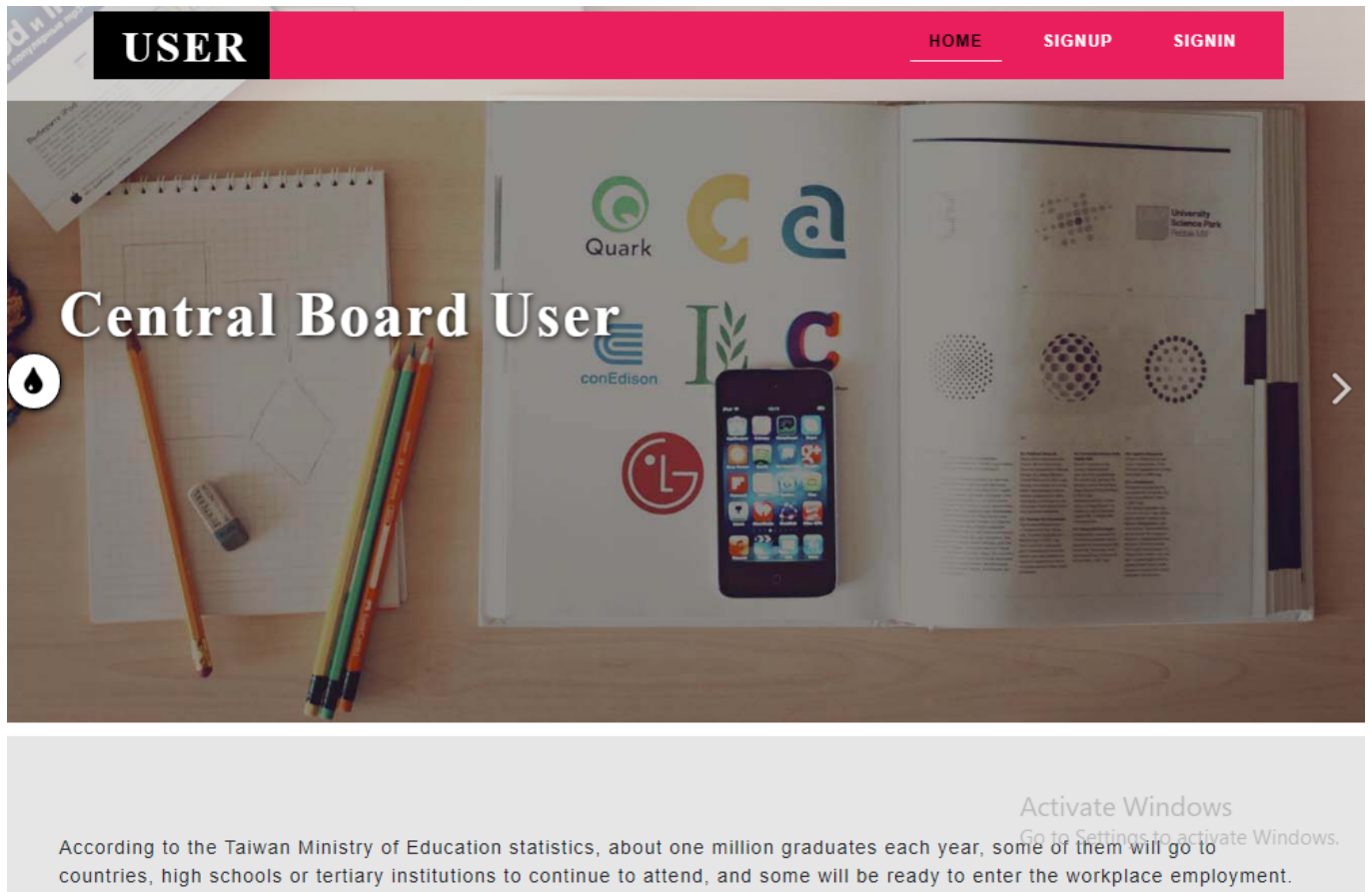


Fig B.3 MAIN PAGE

The image shows a mobile application interface for account creation. The background is a collage of educational materials, including a notebook with a pencil, a book titled 'University Science Park', and a document with the text 'Central Board'. The interface features a dark grey central panel with white text and input fields. At the top left, the word 'USER' is displayed in a white serif font. At the top right, a dark red navigation bar contains the links 'HOME', 'SIGNUP', and 'SIGNIN' in white. The form fields are as follows:

- Name:** A white input field containing the text 'Y ROHITH Y'.
- Email:** A white input field containing the text 'dr1928@gmail.com', highlighted with a blue border.
- Password:** A white input field filled with dots.
- Confirm Password:** A white input field, currently empty.
- Date of Birth:** A white input field containing the placeholder 'dd-mm-yyyy' and a calendar icon.
- Gender:** A label with a corresponding input field below it.

Fig B.4 ACCOUNT CREATING

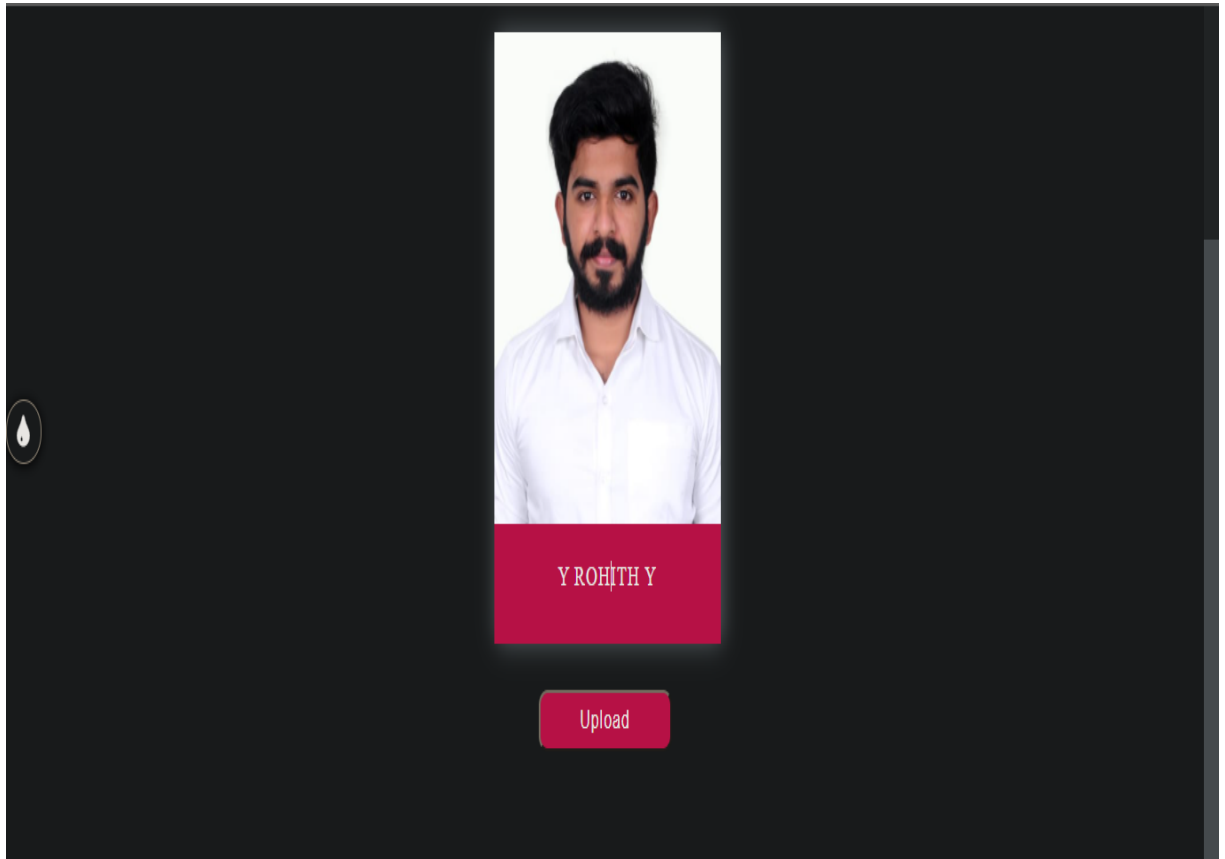


Fig B.5 UPLOADING DOCUMENTS

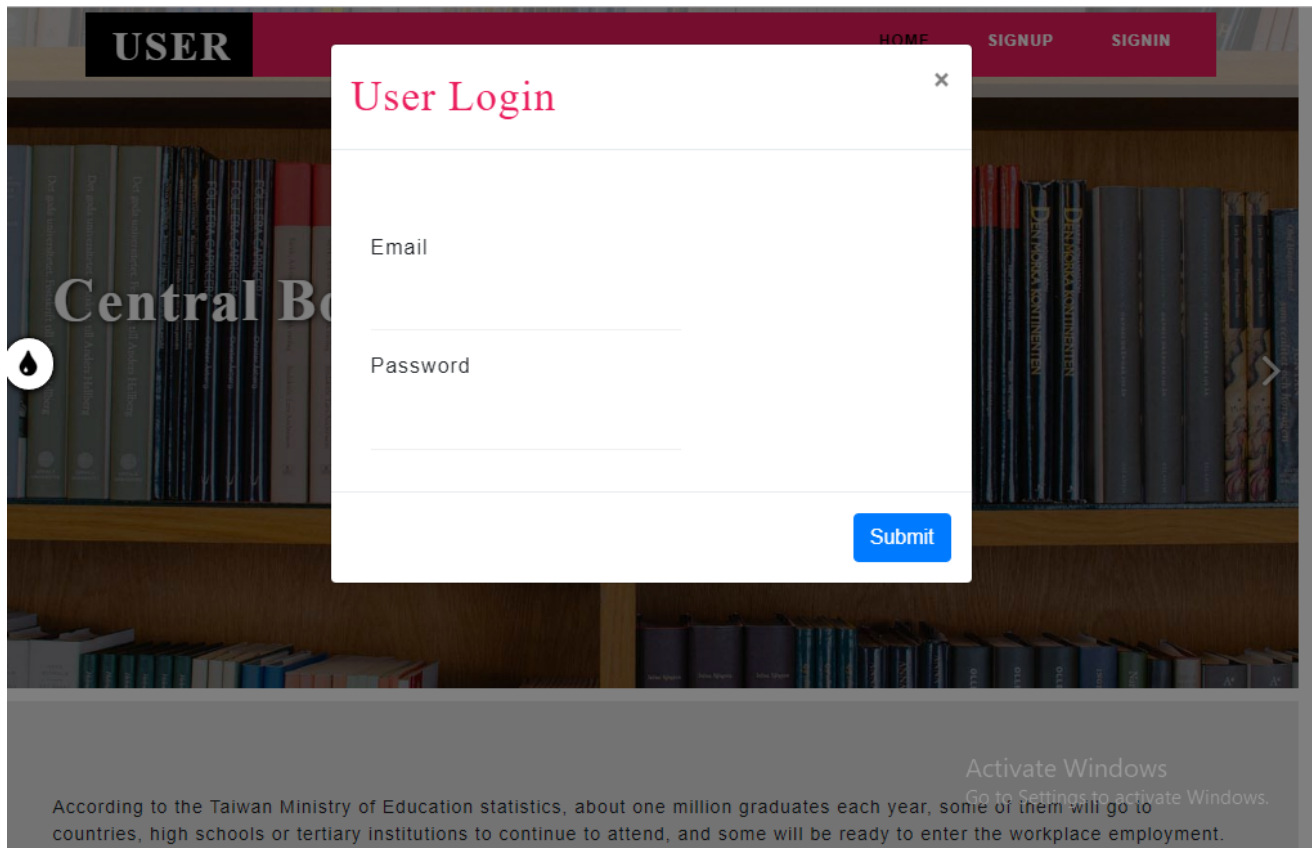


Fig B.6 LOGIN PAGE

Central Board

[HOME](#)  [LOGOUT](#)

UserRequests

SL No.	User Name	Email	UID	Actions
1	Nitishkumar Reddy	nitish@gmail.com	UID7790	Approved
2	a	a@gmail.com	UID3480	Approved
3	c	c@gmail.com	UID8813	Approved
4	d	d@gmail.com	UID8915	Approved
5	k	k@gmail.com	UID6548	Approved
6	Y ROHITH Y	rohithyiseti@gmail.com	UID9548	Approved
7	rohith	sairohithy8@gmail.com	pending	ApproveDecline
8	Y ROHITH Y	dr1928	pending	ApproveDecline

Fig B.7 CENTARL BOARD SERVER

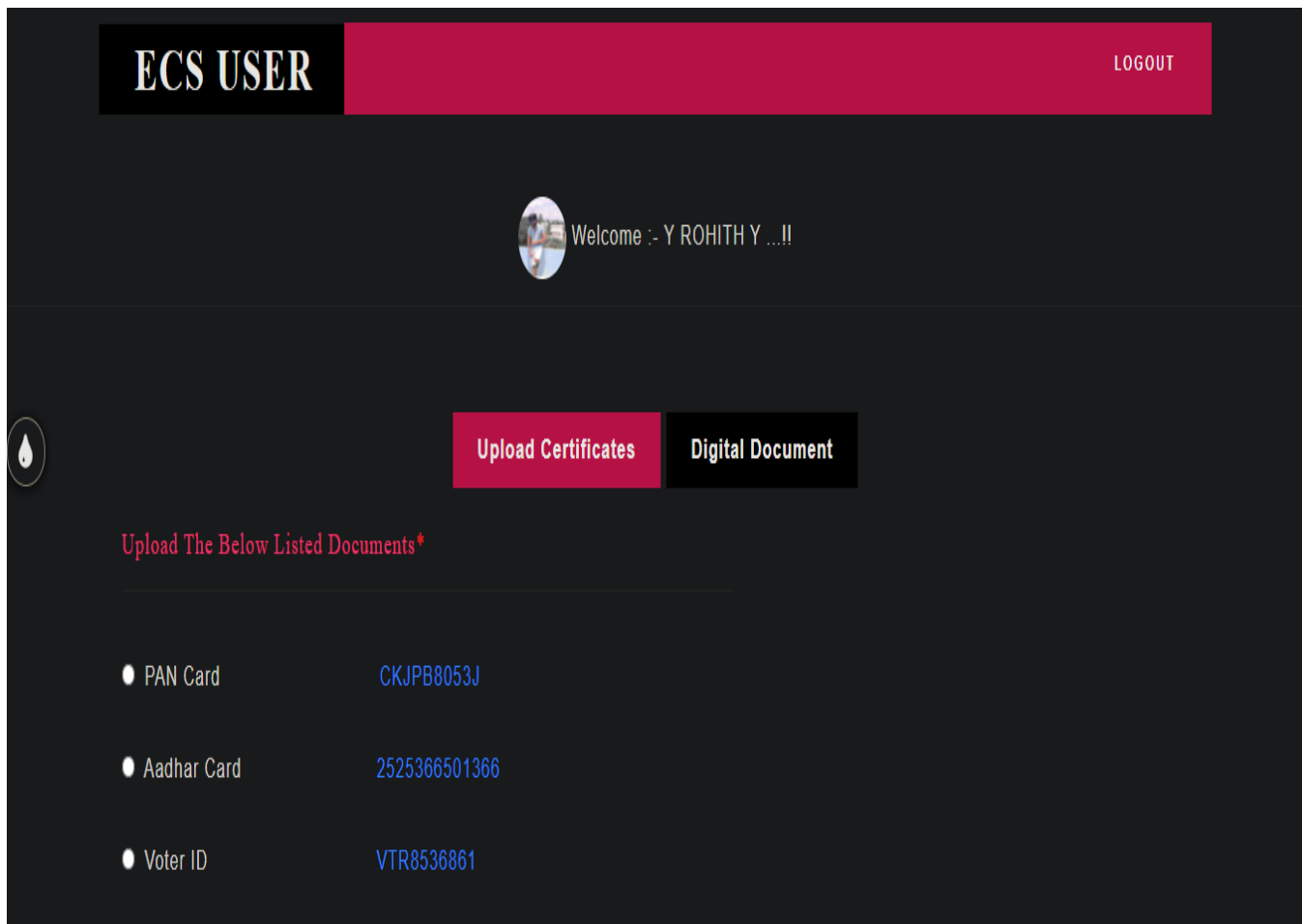


Fig B.8 ECS USER



Central Board

HOME

 LOGOUT

Documents

Verification

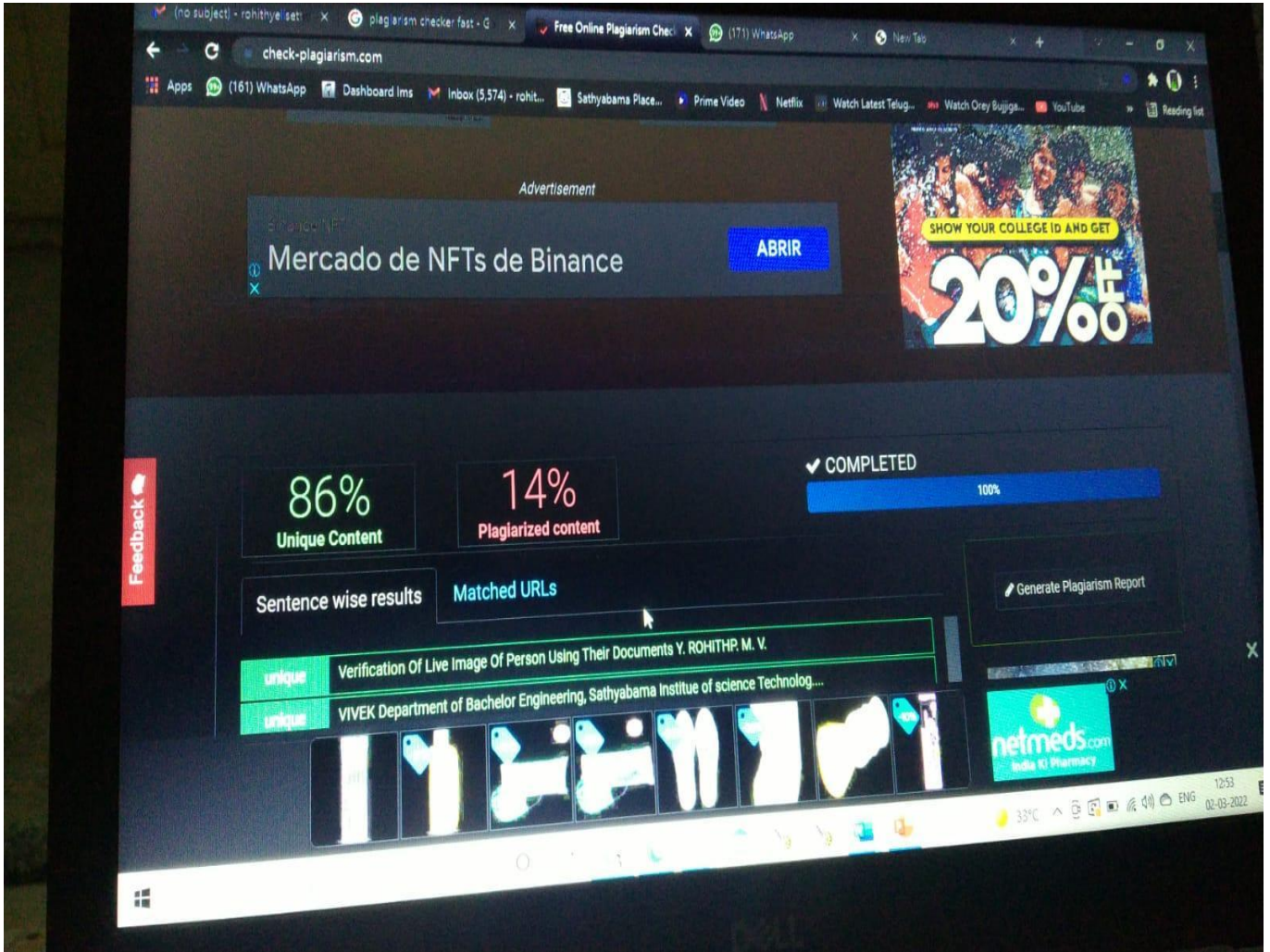
Document Details

PAN CARD	Email:
CKJPB9217J	dr1928
AADHAR CARD	
55427663280639	
VOTER ID	
VTR7460348	
SSLC CERTIFICATE	
379410	

Accept

Decline

C) PLAGIARISM REPORT



Verification Of Live Image Of Person Using Their Documents

Y.ROHITH

P.M.V.VIVEK

Department of Bachelor Engineering, Sathyabama Institute of science and Technology

Department of Computer science , Sathyabama deemed to be university
Sathyabama institute of scienc and technology

Vivek.pavankalyan@gmail.com Rohithyelisetti@gmail.com

Guide Name:**Dr. Rajalakshmi.R**

Abstract

The key properties is to develop the number of documents verification to verify sensations for reducing intra-personal variations while enlarging inter-personal differences. In this paper we can solve the issue for the verification of systems, we show that they can so long after it can be well solved with deep learning and further improved and using both and face recognition to identification for people to solve and verification of peoples faces is important signals as model. The Deep Identification-verification features (Deep) are learned with carefully designed deep convolutional networks. the main key reason to solving the main rules for the face recognition identification to task increases the inter-personal variations by drawing Deep learning features extracted from different identities apart, while the face verification task reduces the intra-personal and normal variations by pulling Deep learning features extracted from the same identity together, both of which are essential to face recognition. The learned Deep and fast ID2 features can be well generalized to new identities unseen in the training data. On the

challenging LFW data, 98 face recognition verification recognition accuracy is achieved. Compared with the best previous deep learning result of an on LFW, the error rate

Introduction

the same identity of faces could look much different when presented in different poses, illuminations, expressions, ages, and occlusions. the same identity of such variants can help could overwhelm the variations due to identifying differences and make face recognition challenging, especially in unconstrained conditions. Now, reducing the personal-intra variations while seeing the difference enlarging the personal -intra so many differences is a normal topic central in face verification recognition. so many normal specifications into subspace face recognition methods such as normal to specify, Bayesian face [16], and unified subspace. there we can expect an example, L,D,A approximates intra- and inter-personal and normal face recognition variations by using any two or three scatter for the normal for this kind of matrices and finds the projection directions to maximize the ratio between them. More recent studies have also targeted the same goal, either explicitly or implicitly. there we can take such ex metric face learning auto maps faces to some normal feature to provide representation such that the faces of the same face identity can be close to each and other while these of different identities stay apart.now we can see the all these fro many models are so much limited and unlimted of some time by their linear sourcer nature or and other shallow the structures, while intra-and inter-personal variations are high complex, highly nonlinear, and can be observed in very high-low dimensional image normal area.

In this work, we can again show that deep face learning can provide much and more powerful tools fro to handle

and the three or many more types of variations. this method can be used to is easy for faec to verify for its deep face verification

Some users quickly for it and face verify identification can be learned by hierarchy complex nonlinear mappings. We can understand Mant's reasoning that it is simple to learn such common features by simultaneously using two and supervisory signals, namely the face identification and verification signals, and the learnt features were referred to as Deep face Identifying features). Many know information an output, input image into a huge no osame identity or not (i.e. binary classification). Given an input face image containing an identification signal, DeepID2 features can be extracted in the learned hierarch nonlinear feature representation's top hidden layer in the training stage, and then through another function g , mapped to one of a huge number of identities (DeepID2). The learnt DeepID2 features can be applied to other tasks (such as face verification) and new identities not seen in the training data during the testing step. Because distinct identities must be sorted into different classes, the identification supervisory signal tends to separate the DeepID2 properties of different identities. As a result, the learnt characteristics would exhibit a wide range of identity-related or inter-personal variances. However, because distinct DeepID2 characteristics could be mapped to the same identity using function $g()$, the identification signal has a relatively weak constraint on DeepID2 features retrieved from the same identity. When DeepID2 features are generalised to new tasks and identities in tests where g is not present, this causes issues. no longer available We tackle this by adding a face verification signal that needs all DeepID2 feature vectors derived from the same identity to be close to each other, while those obtained from different identities to be kept away. DeepID2 features with a high per-element restriction can substantially eliminate intra-personal variations. However, utilising the verification signal alone (i.e. simply distinguishing a pair of DeepID2 extracted features at a time) is not as effective as using the identification signal in extracting identity-related features (i.e. distinguishing thousands of identities at a time). As a result, the two supervisory signals should be used together because they stress different aspects of feature learning.

Complement DeepID2 features can be extracted from diverse face regions and resolutions and concatenated to generate the final visual features after PCA feature reduction to characterise faces from differing viewpoints. Because the learnt DeepID2 traits differ between identities while remaining consistent within the same identification, face recognition becomes easier. On the challenging and thoroughly investigated LFW dataset [11], we achieved the greatest 99.15 percent face verification accuracy using the learnt feature representation and a recently suggested face validation model [3]. This is the first time a machine with only the facial image achieves an accuracy of 99.20 percent in comparison to a human who is presented with the whole LFW face image, such as the face region and huge background area. Complimentary DeepID2 features are extracted from diverse faces and resolutions and concatenated to generate the final visual features after PCA dimension reduction to characterize faces from differing viewpoints. Because the learnt DeepID2 traits differ between identities while remaining constant within the same identification, face recognition becomes easier. On the challenging and extensively studied LFW dataset [11], we achieved the greatest 99.15 percent face verification accuracy using the learnt representations and a recently suggested face verification model [3]. This is the first time a machine with only the facial image achieves an accuracy of 99.20 percent when compared to a human who is provided with the whole LFW face image, including the face region and large background area.

1 Deep learning for verification process

Fully convolutional neural networks (deep Convent's) [12] are being used to learn features. Deep Convents' convolution and pooling operations are specifically designed to extract visual features in a hierarchical fashion, from local low-level features through large number of high features. Our deep Convent's use topologies similar to those in [20]. It has four convolutional layers, the third and fourth of that which feature local weight sharing [10]. At the last layer (DeepID2 layer) of the feature extraction cascade, the Convolution extracts a 160-dimensional DeepID2 feature vector. Both the 3rd and 4th convolution layer are fully connected to the DeepID2 layer to be learned. In the convolution layer and the DeepID2 layer, we use rectified linear units (ReLU) [17]. An example of the DeepID2 extract Convent structure. Using an RGB source of size 55 47, the elements are shown in Fig. 1. When the input region's size changes, the map sizes in the following layers change as well. $f = \text{Conv}(x, c)$ symbolizes the DeepID2 feature extraction process, where $\text{Conv}()$ is the Convent feature extraction function, x is the input face patch, f is the obtained DeepID2 vector, and c denotes the Convent variables to be learned.



Two supervisor messages are used to learn DeepID2 features. The first is the face identification signal, which assigns each facial images to one of n (for instance, $n = 8192$) distinct identities. Following the DeepID2 layer with an n -way soft - max, which outputs a confidence interval over the n classes, identification is accomplished. The network can be trained to minimise the identifying loss, which is the cross-entropy loss. It's referred to as **Ident**

$$(f, t, \theta_{id}) = - \sum_{i=1}^n p_i \log \hat{p}_i = - \log \hat{p}_t, \quad (1)$$

The DeepID2 feature map is f , the target class is t , and the softmax layer variables are id . The goal probability distribution, p_i , is 0 for all i except $p_t = 1$ for the target class t , and $p_i = 0$ for all i except $p_t = 1$ for the target class t . The projected probability distribution is denoted by $\hat{p}_i$. The DeepID2 layer must create discriminative identity-related features in order to correctly classify all of the classes at the same time (i.e. features with large inter-personal variations). The second signal is the face validation signal, which promotes DeepID2 features extracted from the same identity's faces to seem alike. DeepID2 features are directly regularised by the validation signal, which effectively reduces intra-personal variances. The L1/L2 norm and cosine similarity are two common constraints. The accompanying loss function depends on the L2 norm, which was proposed originally. **Verif**

$$(f_i, f_j, y_{ij}, \theta_{ve}) = \frac{1}{2} (y_{ij} - \sigma(wd + b))^2, \quad (3)$$

where $d = \frac{f_i \cdot f_j}{\|f_i\|_2 \|f_j\|_2}$ is the cosine similarity between DeepID2 feature vectors, $\theta_{ve} = \{w, b\}$ are learnable scaling and shifting parameters, σ is the sigmoid function, and y_{ij} is the binary target of whether the two compared face images belong to the same identity. All the three loss functions are evaluated and compared in our experiments.

Our goal is to learn the parameters θ_c in the feature extraction function $\text{Conv}(\cdot)$, while θ_{id} and θ_{ve} are only parameters introduced to propagate the identification and verification signals during training. In the testing stage, only θ_c is used for feature extraction. The parameters are updated by stochastic gradient descent. The identification and verification gradients are weighted by a hyperparameter λ . Our learning algorithm is summarized in Tab. 1. The margin m in Eq. (2) is a special case, which cannot be updated by gradient descent since this will collapse it to zero. Instead, m is fixed and updated every N training pairs ($N \approx 200,000$ in our experiments) such that it is the threshold of Table 1: The DeepID2 feature learning algorithm.

input: training set $\chi = \{(x_p, l_i)\}$, initialized parameters θ_c , θ_{id} , and θ_{ve} , hyperparameter λ , learning rate $\eta(t)$, $t \leftarrow 0$

while not converge do

$t \leftarrow t + 1$ sample two training samples (x_p, l_i) and (x_p, l_j) from χ $f_i = \text{Conv}(x_p, \theta_c)$ and $f_j = \text{Conv}(x_p, \theta_c)$

$\partial \text{Verif}_{id}$

id

$$\nabla \theta_{id} = \frac{\partial \text{Ident}(f_i, l_i, \theta_w)}{\partial \theta} + \frac{\partial \text{Ident}(f_j, l_j, \theta_w)}{\partial \theta}, \text{ where } y_{ij} = 1 \text{ if } l_i = l_j, \text{ and } y_{ij} = -1 \text{ otherwise.}$$

$$\nabla_{f_i} \theta_{ve} = \frac{\partial \text{Ident}(f_i, l_i, \theta_w)}{\partial f_i} + \lambda \cdot \frac{\partial \text{Verif}(f_i, f_j, y_{ij}, \theta_{ve})}{\partial f_i}$$

$$\nabla_{f_j} \theta_{ve} = \frac{\partial \text{Ident}(f_j, l_j, \theta_w)}{\partial f_j} + \lambda \cdot \frac{\partial \text{Verif}(f_i, f_j, y_{ij}, \theta_{ve})}{\partial f_j}$$

$\nabla \theta_c = \nabla_{f_i} \cdot \frac{\partial \text{Conv}(\theta_{xi}, \theta_c)}{\partial \theta_c} + \nabla_{f_j} \cdot \frac{\partial \text{Conv}(\theta_{xj}, \theta_c)}{\partial \theta_c}$ update $\theta_{id} = \theta_{id} - \eta(t) \cdot \nabla \theta_{id}$, $\theta_{ve} = \theta_{ve} - \eta(t) \cdot \nabla \theta_{ve}$, and $\theta_c = \theta_c - \eta(t) \cdot \nabla \theta_c$. end while output θ_c

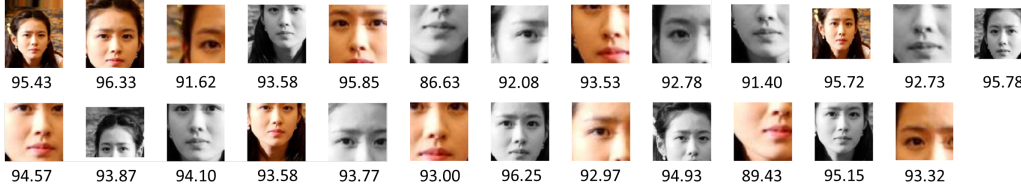


Figure 2: Patches selected for feature extraction. The Joint Bayesian [3] face verification accuracy (%) using features extracted from each individual patch is shown below.

2 To reduce the verification error of the preceding N training pairings, use the feature distances kufi fake. Tab. 1 does not cover updating m since it is too complicated.

3 Verification of human face

DeepID2 characteristics are embedded into the face recognition verification pipeline of face recognition, feature extraction, and face verification to evaluate the feature learning technique stated in Sec. 2. We start by detecting 21 face landmarks with the recently developed SDM method [24]. The facial photos are then aligned globally using similarity transformations based on the discovered landmarks. According to the globally aligned faces and the location of the facial landmarks, we cropped 400 face patches with varying positions, scales, colour channels, and horizontal flips. As a result, a total of 200 deep ConvNets extract 400 DeepID2 feature vectors, which are trained to extract two 160-dimensional DeepID2 vectors on one face patch and its horizontal flipped counterpart, respectively. We use the forward-backward greedy algorithm [25] to select a small number of effective and complementary DeepID2 feature vectors (25 in our experiment) to reduce redundancy among the large number of DeepID2 features and make our system practical. This saves the majority of the feature extraction time during the test. Figure 2 depicts all 25 patches that were chosen, with 25 160-dimensional DeepID2 feature vectors extracted and concatenated to form a 4000-dimensional DeepID2 feature vector. PCA compresses the 4000-dimensional vector to 180 dimensions for face verification. Based on the extracted DeepID2 characteristics, we learned the Joint Bayesian model [3] for face verification. The joint Bayesian model has been used to correctly model the joint probability of two faces being the same or distinct people.

4 Main experiment to include

The LFW data [11], which would be the de - facto test set for face recognition in unconstrained settings, is used to provide face verification findings. It includes 13,233 face photographs from 5749 people gathered from the internet. Algorithms often report both mean face validation accuracy as well as the ROC curve on 6,000 given face pairings in LFW for comparison purposes. Although it is a good test set, it is insufficient for training because the majority of identifiers in LFW only have one face image. As a result, as with many previous high-performance face validation systems [4, 2, 21, 20, 13], we rely on a bigger outside dataset for training. The Celeb Faces+ dataset [20], which contains 202,599 face photos of 10,177 identities (celebrities), is used for training.. DeepID2 features are learned

from the face images of 8192 identities randomly sampled from Clubfaces+ The surviving detected faces of 1985 personalities (made reference to as Celeb Faces) can be used for selecting features and having to learn the face verification models, whereas the remaining detected faces of 1985 personalities (referred to as Celeb Faces+A) are used for selecting features and having to learn the face verifying models (Joint Bayesian). Celeb Faces is utilized as a validation set to determine the learning rate, train epochs, and hyper - parameters when training DeepID2 features on Celeb Faces+A. For feature selection, Celeb Faces is split into training of 1485 ids as well as a testing set of 500 identities. Finally, we evaluate the Joint Probabilistic model on LFW using chosen DeepID2 features after training it on the whole Celeb Faces+B set. Using a single neural network Conv Net to retrieve DeepID2 characteristics from the full face region, we first examine various aspects of extracting features from Sec. 4.1 through Sec. 4.3.

4.1 Process of verifying the document

By ranging from 0 to +, we study the effects of identification and authentication signals on feature learning. At = 0, the validation signal is no longer active, and only the identifying signal is active. As, the verification signal progressively takes over control of the training process. Only the confirmation signal remains at the other end of the + scale. Eq. (2)'s L2 norm confirmation loss is employed for training. Figure 3 compares the learnt DeepID2 features with the L2 norm and the Dual Bayesian model to show the face verification reliability on the test set. It is evident that neither the identifying signal nor the confirmation signal is the best for learning features. Instead, effective characteristics are the result of a well-balanced blend of the two.

This phenomenon can be explained from the view of inter- and intra-personal variations, which could be approximated by LDA. According to LDA, the inter-personal scatter matrix is $S_{inter} =$

$$\sum_{i=1}^c n_i \cdot (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T, \text{ where } \bar{x}_i \text{ is the mean feature of the } i\text{-th identity, } \bar{x} \text{ is the mean of the}$$

n_i is the amount of face photographs for the i -th identity in the overall collection. D_i is the collection of features of the i -th identification, $\bar{x}_i$ is the associated mean, and c is the variety of different identities in the intra-personal scatter matrix. The eigenvalues of the related scatter matrices are given in Fig. 5 as inter- and intra-personal variances. Different variation patterns are represented by the corresponding eigenvectors. In recognition, both the amplitude and variety of feature variations are important. When all of the feature variances are concentrated on a tiny proportion of eigenvectors, it means that intra- or cross variation diversity is low. = 0, 0.05, and + are used to learn the features, accordingly. The variances of each provided characteristic are normalised using the mean feature.

The learnt features contain both inter- and intra-personal differences when only the identifying signal is employed (= 0), as evidenced by the lengthy tails of the red curve in both pictures. While inter-personal differences help discern various identities, significant and diversified intra-personal differences are distracting and make with someone difficult. The diversity of inter-personal differences remains intact when both identification and authentication signals are utilised with proper weighting (= 0.05), although variances in a few primary directions get even larger, as illustrated by the green curve on the left compared to a red curve. At same time, the diversity and size of intra-personal variances diminish, as seen by the green curve on the right. As a result, bothFace verification becomes easier as inter- and intra-personal variances vary in a positive direction. When is increased to infinity, both intra and inter variations compress to variants in only a few primary directions, because different features cannot be generated without the identifying signal. Due to a lack of diversity on the inter-

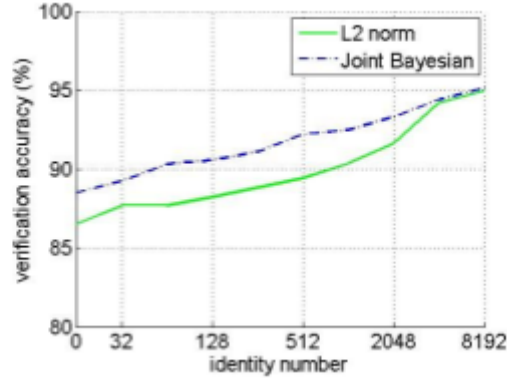
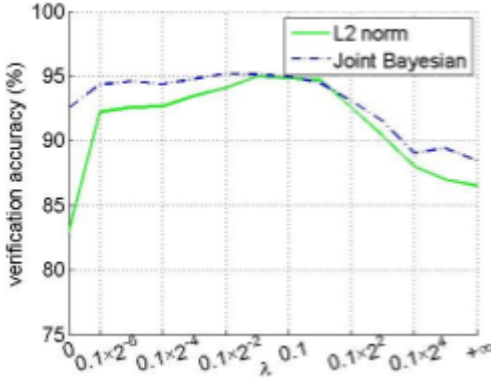


Figure 3: Face verification accuracy by varying the weighting parameter λ . λ is plotted in log scale. Figure 4: Face verification accuracy of DeepID2 by varying the number of training identities (shown in log scale) used for face identification. The result may be further improved with more than 8192 identities.

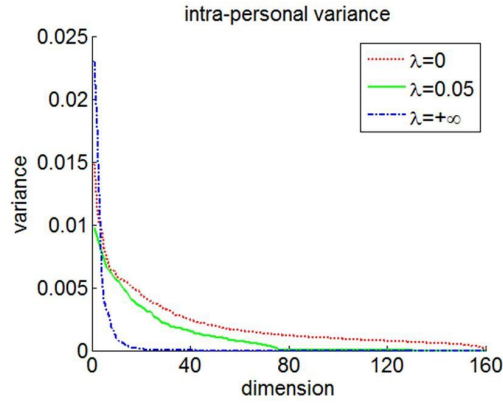
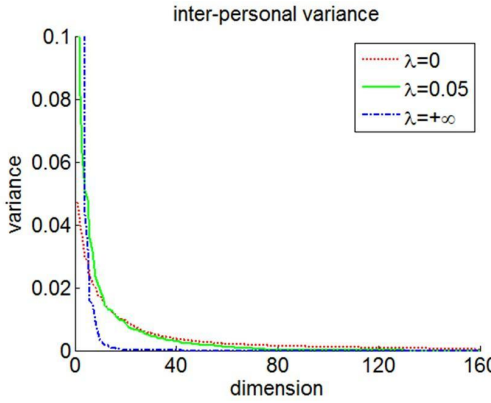


Figure 5: Spectrum of eigenvalues of the inter- and intra-personal scatter matrices. Best viewed in color.

personal variations, distinguishing different identities becomes difficult. Therefore the performance degrades significantly.

Figure 6 shows the first two PCA dimensions of features learned with $\lambda = 0, 0.05$, and $+\infty$, respectively. These features come from the six identities with the largest numbers of face images in LFW, and are marked by different colors. The figure further verifies our observations. When $\lambda = 0$ (left), different clusters are mixed together due to the large intra-personal variations, although the cluster centers are actually different. When λ increases to 0.05 (middle), intra-personal variations are significantly reduced and the clusters become distinguishable. When λ further increases towards infinity (right), although the intra-personal variations further decrease, the cluster centers also begin to collapse and some clusters become significantly overlapped (as the red, blue, and cyan clusters in Fig. 6 right), making it hard to distinguish again.

4.2 Providing the rich identification for the whole process

We are investigating how would the identity information contained in the identification supervisory signal influence the learned features. In particular, we experiment with an exponentially increased number of identities used for identification during training from 32 to 8192, while the verification signal is generated from the 8192 training identities all the time. shows

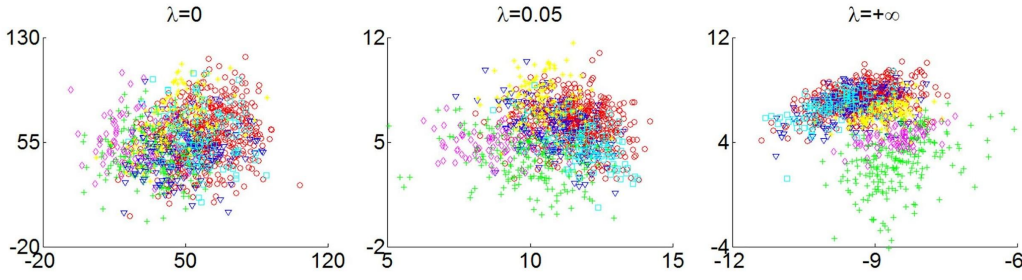


Figure 6: The first two PCA dimensions of DeepID2 features extracted from six identities in LFW.

Table 2: Comparison of different verification signals.

verification signal	L2	L2+	L2-	L1	cosine	none
L2 norm (%)	94.95	94.43	86.23	92.92	87.07	86.43
Joint Bayesian (%)	95.12	94.87	92.98	94.13	93.38	92.73

We are investigating how would the identity information contained in the identification supervisory signal influence the learned features. In particular, we experiment with an exponentially increased number of identities used for identification during training from 32 to 8192, while the verification signal is generated from the 8192 training identities all the time Shows.

4.3 Verifying the signals sent

The verification signal with moderate of intensity mainly takes the effect of reducing the intra-personal variations. To further verify this, We compare our L2 norm verification signal of all the sample pairs with those that only constrain either the positive or negative sample pairs, denoted as L2+ and L2-, respectively. That is L2+ only decreases the distances between DeepID2 features of the same identity, while L2- only increases the distances between DeepID2 features of different identities if they are smaller than the margin. The face verification accuracies of the learned DeepID2 features on the test set, measured by the L2 norm and Joint Bayesianas well as no verification signal (none). The identification signal is the same (classifying the 8192 respectively, are shown in Table 2. It also compares with the L1 norm and cosine verification signals, identities)for all the comparisons. Deep ID2 features learned with the L2+ verification signal are only slightly worse than those learned with L2. In contrast, the L2- verification of signal helps little in feature learning and gives almost the same result as no verification signal is used. This is strong evidence that the effect of the verification signal is mainly reducing the intra-personal variations. Another observation of face verification accuracy improves in general whenever the verification signal is added in addition to the identification signal. However, the L2 norm is better than the other compared with verification metrics. This may be due to that all the other constraints are weaker than L2 and less effective in reducing the intra-personal variations. For example, the cosine similarity only constrains of angle, but not the magnitude.

4.4 Final process for verifying and identifying the person

Before a learning Joint Bayesian, DeepID2 features are first projected to 180 dimensions by PCA. 6000 given face pairs in LFW, where the log-likelihood ratio given by Joint Bayesian is compared After PCA, the Joint Bayesian model is trained on the entire CelebFaces+B data and tested on the to a threshold optimized on the training data for face verification. Tab. 3 shows the face verification used to extract those DeepID2 features from each face with a single Titan GPU. We achieve 98.97% of accuracy

with all the 25 selected face patches. The feature extraction process is also efficient and accuracy with an increasing number of face patches to extract DeepID2 features, as well as the time takes only 35 ms for each face image. The face verification of accuracy of each individual face patch is provided in Fig. 2. The short DeepID2 signature is extremely efficient of face identification and face image search when matching a query image with a large number of candidates.

Table 3: Face verification accuracy with DeepID2 features extracted from an increasing number of face patches.

# patches	1	2	4	8	16	25
accuracy (%)	95.43	97.28	97.75	98.55	98.93	98.97
time (ms)	1.7	3.4	6.1	11	23	35

Table 4: Accuracy comparison with the previous best results on LFW.

method	accuracy (%)
High-dim LBP [4]	9517± 1.13
TL Joint Bayesian [2]	9633± 1.08
DeepFace [21]	9735± 0.25
DeepID [20]	9745± 0.26
GaussianFace [13]	9852± 0.66
DeepID2	9915± 0.13

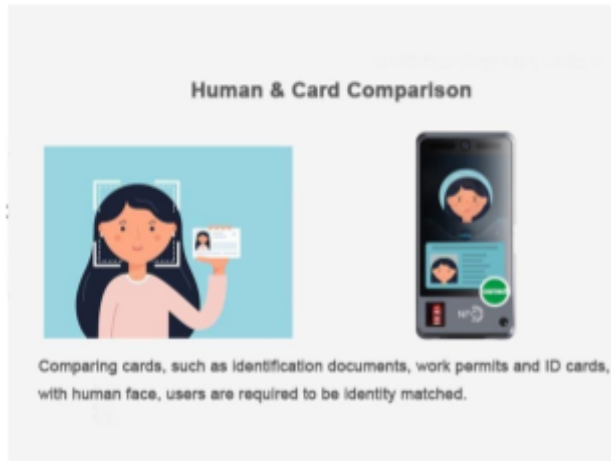


Figure 7: ROC comparison with the previous best results on LFW. Best viewed in color.

To further exploit the rich pool of DeepID2 features extracted from a large number of patches, we repeat the feature selection algorithm another six times, each time choosing a DeepID2 feature from the patches that have not been selected by previous feature selection steps. Then we learned the Joint Bayesian model on each of the seven groups of selected features, respectively. We fuse the seven Joint Bayesian scores a each pair of compared faces by further learning an SVM. In this way, we achieve an even higher 99.15% face verification accuracy. The accuracy and ROC comparison with previous state-of-the-art methods on LFW are shown in Tab. 4 and Fig. 7, respectively. We achieve the best results and improve previous results with a large margin.

5.Main conclusion for whole process

This paper has been shown that the effect of the face identification and verification supervisory signals on the deep feature of representation coincides with the two aspects of constructing ideal features for face recognition, i.e.,

increasing the inter-personal variations of reducing the intra-personal variations, and the combination of the two supervisory signals lead to the significantly better features than either one of them. When embedding the learned features of the traditional face verification pipeline, we achieved an extremely effective system with the 99.15% face verification accuracy on LFW. The arXiv report of paper was published in June 2014 .

References

- [1] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *PAMI*, 19:711–720, 1997.
- [2] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun. A practical transfer learning algorithm for face verification. In *Proc. ICCV*, 2013.
- [3] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *Proc. ECCV*, 2012.
- [4] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Proc. CVPR*, 2013.
- [5] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. CVPR*, 2005.
- [6] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? Metric learning approaches for face identification. In *Proc. ICCV*, 2009.
- [7] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. CVPR*, 2006.
- [8] J. Hu, J. Lu, and Y.-P. Tan. Discriminative deep metric learning for face verification in the wild. In *Proc. CVPR*, 2014.
- [9] C. Huang, S. Zhu, and K. Yu. Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval. *NEC Technical Report TR115*, 2011.
- [10] G. B. Huang, H. Lee, and E. Learned-Miller. Learning hierarchical representations for face verification with convolutional deep belief networks. In *Proc. CVPR*, 2012.
- [11] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments. Technical Report 0749, University of Massachusetts, Amherst, 2007.
- [12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.
- [13] C. Lu and X. Tang. Surpassing human-level face verification performance on LFW with GaussianFace. Technical report, arXiv:1404.3840, 2014.
- [14] A. Mignon and F. Jurie. PCCA: A new approach for distance learning from sparse pairwise constraints. In *Proc. CVPR*, 2012.
- [15] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *Proc. ICML*, 2009.
- [16] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. *PR*, 33:1771–1782, 2000.
- [17] V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proc. ICML*, 2010.
- [18] Y. Sun, X. Wang, and X. Tang. Hybrid deep learning for face verification. In *Proc. ICCV*, 2013.