Web Application for Signature Extraction and Authentication

A

Project Report

submitted in partial fulfillment of the
requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Specialization in

CCVT

by

Name	**Roll No.**
Vaibhav Tomar	R110217175
Vinay Chaudhary	R110217182
Vipin Uniyal	R110217185

under the guidance of

Mr. Amrendra Tripathi

Assistant Professor (SS)
Department of Virtualization

UPES

Department of Virtualization
School of Computer Science
University of Petroleum & Energy Studies
Bidholi, Via Prem Nagar, Dehradun, UK
November – 2020

UPES
COLLEGE
OF ENGINEERING
STUDIES

The innovation driven
E-School

CANDIDATE'S DECLARATION

We hereby certify that the project work entitled **"Signature Extraction & Verification"** in partial fulfilment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in CCVT and submitted to the Department of Virtualization at School of Computer Science, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **August, 2020** to **November, 2020** under the supervision of **Mr. Amrendra Tripathi, Assistant Professor (SS), Department of Virtualization, SoCS**. The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

Vaibhav Tomar [Roll No. 175]	Vinay Chaudhary [Roll No. 182]	Vipin Uniyal [Roll No. 185]

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: December 2020

Mr. Amrendra Tripathi
Project Guide

Dr. Deepshikha Bhargava
Head – Department of Virtualization
School of Computer Science
University of Petroleum & Energy Studies
Dehradun – 248 001 (Uttarakhand)

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide **Mr. Amrendra Tripathi**, for all the advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank our Head of the Department, **Dr. Deepshikha Bhargava**, for her great support in doing our project in **Area** at **SoCS**.

We are also grateful to **Dr. Manish Prateek Professor and Director SoCS**, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

Vaibhav Tomar	**Vinay Chaudhary**	**Vipin Uniyal**
500060797 **R110217175**	**500061178** **R110217182**	**500060225** **R110217185**

ABSTRACT

Signatures are one of the most important techniques for biometric authentication. There are two kinds of signatures nowadays, offline (static) and dynamic (online).

There are greater distinctive characteristics of online signatures, but there are less distinctive characteristics of offline signatures. Offline signatures are also harder to check. Furthermore, the most significant downside of offline signatures is that even the most talented signer does not sign them the same way. That is called Intra-personal variability. Both of these make checking the offline signature a difficult problem for researchers.

In this research, we proposed an offline signature verification approach based on Deep Learning (DL) to prevent signature fraud by malicious individuals.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Biometric systems are primarily used in two situations: identification and verification. In the identification, the user of the system provides some biometric samples and ensures his or her identity. The task of such a system is to identify whether the user is genuinely he or she claims to be. In the case of verification, the system processes the data provided by the user and compares the data sample among all registered users in the system and then verifies the identity of that user.

The handwritten signatures are one of the ubiquitous and important attributes of biometric identification, mainly due to their widespread use in the financial, administrative, and legal fields to verify an individual's identity. One of the reasons for its widespread use is that the process of collecting handwritten signatures does not involve the physical presence of the person and people are familiar with their use in their day-to-day lives. The signature verification system is designed to automatically identify whether the handwritten signature samples are authentic or not. In simpler words, they are used to discriminate genuine signatures from the forged ones.

Counterfeiting signatures basically termed as forgeries are usually divided into three categories: simple, random, and simulated or skilled forgery. For simple forgery, the counterfeiter knows the user name but does not know the user's signature. In this case, the forgery may have more similarities to the real signature, especially for users who use a full or partial full name signature. In the case of random forgery, the counterfeiter does not have information about the user or his signature but uses some random or his own signature. In this case, the semantic meaning of the forgery is different from the user's real signature, resulting in a very different overall shape. In skilled forgery, counterfeiters can access a user's name and signature, often imitating the user's signature. This results in a higher similarity between counterfeits and real signatures, making them more difficult to detect.

The signature verification system is divided into two types: static referred to as offline and dynamic referred to as online. In the online mode, an acquisition device is used, such as a digitized table or other biometric information to get the user's signature. The data is collected sequentially over time, including the position of the pen and, in some cases, other information, such as the pen's tilt, pressure, and so on. In the verification of offline signatures, the signature is obtained after the user completes his or her writing process; a digital image scanned from the paper is represented in this case.

In this paper the work is organized as follows: we start by pre-processing the image uploaded by the user and convert it into the list of binary values of the pixels. The dataset we are using is the CEDAR signature dataset that is one of the benchmark datasets for signature recognition.

2. LITERATURE REVIEW

Countless studies on scene text detection and recognition have been developed and published in the recent years. One of the most common procedures used for the extraction of text from scene images is through an end-to-end recognition of scene text, which is divided into two sub-processes:

Scene text detection

Also known as ‘localization’, it is used to detect regions of text and extract these regions from the input image.

In this process, the locations where the text is present in the image are identified. [1] Before feeding the image to the feature detector model, the image is preprocessed. Preprocessing is done by converting the image to grayscale, defining an optimum resolution for the feature detector in which it has to be sent into, detecting edges in the image, etc. [2]

Scene text recognition

In screen text recognition, the extracted text regions are processed and extracted text strings are returned.

After the identification of text regions, the process of recognizing the text information present in those regions is executed. The text regions of these images are sent into deep learning models that are vigorously trained to recognize characters, and consequently words. [1]

The output of these models contains the words or characters that were present in the image.

The process of verification of the extracted signatures has been achieved through the use of Convolutional Neural Networks (CNN), since neural networks are commonly used for both writer dependent and independent systems. [3] This project makes the use of the Siamese Neural Networks, which uses twin CNNs, for verifying the authenticity of the signatures. [4]

3. OBJECTIVE

The aim of this project is to create a web application that will check the authenticity of signatures with the help of deep learning.

Given a dataset ***A*** (Used for learning) that contains real signatures from a group of users, using which we can train the model and store it in set ***t1*** (train data). The model is then used for authentication; the user tries to claim his or her identity and provides a signature image. The model then tries to classify the signature, whether it belongs to the genuine category or forged category. To calculate the accuracy of the system, we consider a test set ***t2*** (test data) consisting of forgeries and real signatures.

4. PROBLEM STATEMENT

One of the major challenges of offline signature verification is having very high variation in different signature samples.

The offline signature is usually captured by a scanner or any other type of imaging devices, which basically produces two dimensional signature images. As signature verification has been a popular research topic through decades and substantial efforts are made both on offline as well as on online signature verification purpose.

There are many cases where authenticating offline signatures is the only option such as check transaction and document verification. Because of its broader application area, in this paper, we focus on the more challenging task- automatic offline signature verification.

5. METHODOLOGY

This project consists of two main parts:

Preprocessing of extracted image and modeling of dataset to train the system and testing accuracy. Using Euclidean distance as the distance metric we will be comparing the original and processed signature images.

Pre-Processing:

Pre-processing plays a vital role in signature validation. Even between a person's real signatures, the signature image can show changes in pen weight, scale, rotation, and so on. Let's summarize the main pre-processing techniques:

Signature extraction: This is the first step to include finding and extract signatures from documents. This is a particularly challenging issue in bank checks. Signatures are usually written at the top of the complex background. However, most signature verification studies do not consider this step because they already consider the signature extracted from the document.

Noise Removal: Scanned signature images usually included Noise. The common strategy for resolving this issue is applying a noise-removal filter to the image.

Invert Image: Invert original signature images to get the features easily highlighted and recognizable (such as geometric, texture, depth, wavelet etc).

Resizing: In this project, deep learning is used for offline signature verification. A neural network usually requires images of the same size because of batch training, but the signature images we consider have different sizes. Using bilinear interpolation, we resize all the images to 155x220. We then invert the images such that there are 0 values for the background pixels. In addition, by splitting the pixel values with the standard deviation of the images' pixel values in a dataset, we normalize each image.

Modeling of the dataset:

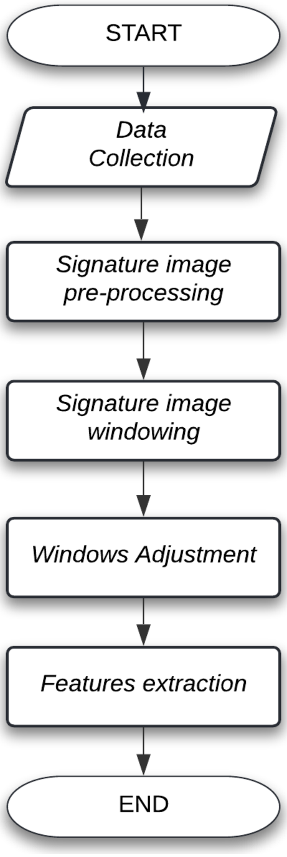
CNN and Siamese Network: The Siamese neural network is a network architecture class that usually involves two similar sub-networks. With the same parameters and shared weights, the twin CNNs has the same configuration. The upgrade parameter is replicated in all sub-networks.[5] A loss function at the top connects these sub-networks, which computes a metric of similarities involving the Euclidean distance between the representation of the feature on each side of the Siamese network.

Datasets used for training:

CEDAR: CEDAR signature database contains signatures of 55 signers belonging to various cultural and professional backgrounds.

Web app creation:

After implementing the model and training it, we need to make a web application using python flask framework. We hosted this on our local machine. The functionality of a web application is simple. Firstly the user will upload his/her three genuine signatures and then upload a test signature image to verify it. We used a database to retrieve and store images.



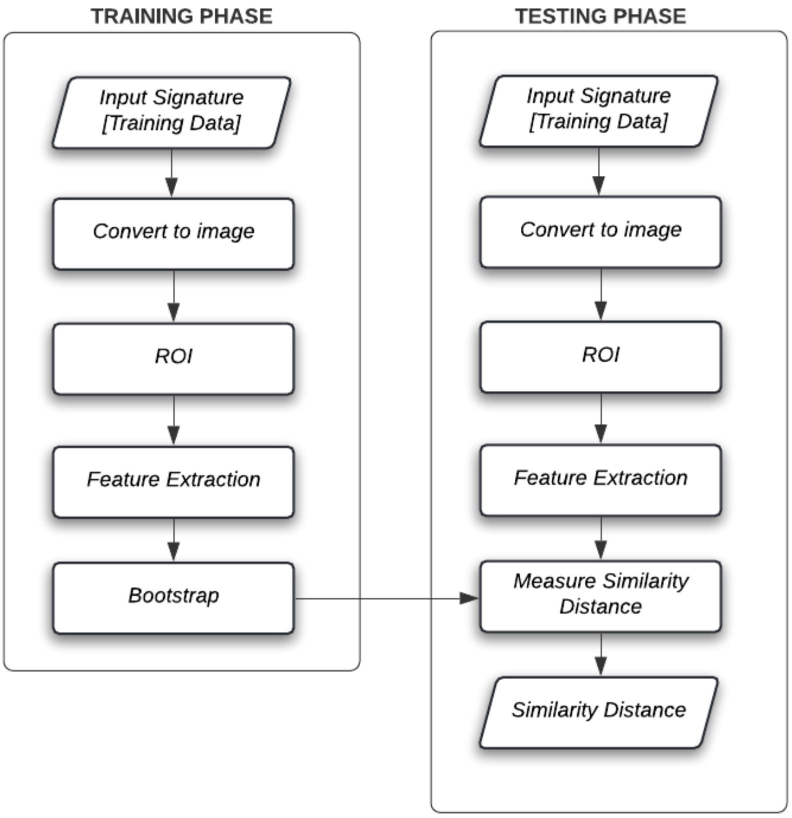


Fig 1: Signature Extraction from given image

Fig 2: Training & Testing phases for Signature verification process

6. ALGORITHM

The algorithm for the system:

Step 1: Download dataset from the website:
http://www.cedar.buffalo.edu/NIJ/data/signatures.rar

Step 2: Data Preprocessing:

- Convert image to image array
- Convert image array to image tensor
- Invert image using PIL(Python Imaging Library)
- Return inverted image array

Step 3: Model Creation:

- Derive pyTorch class Module which is base class for all neural networks and extract the features of images and then get Euclidean distance and return it as distance metric.
- The model used is Siamese Convolutional Neural Network.
- Calculate similarity score.
- Loss function (contrastive loss) is used to find the accuracy.

Step 4: Model Training:

- Training of the model is done until fixed epoch, the preprocessing helps in increasing accuracy.

Step 5: Test model accuracy

Step 6: Web Application Creation:

- Using the flask framework the web interface is created and deployed on a local server.

7. SYSTEM REQUIREMENTS

Python libraries used in development:
pyTorch, opencv, keras, flask

Software specification requirements:

Operating System	: *Windows 7 or above / Ubuntu 16.0 or above*
Compiler	: *python 3 compiler [for backend processing] with flask module*
Web Browser	: *Google Chrome, Mozilla Firefox [preferred]*

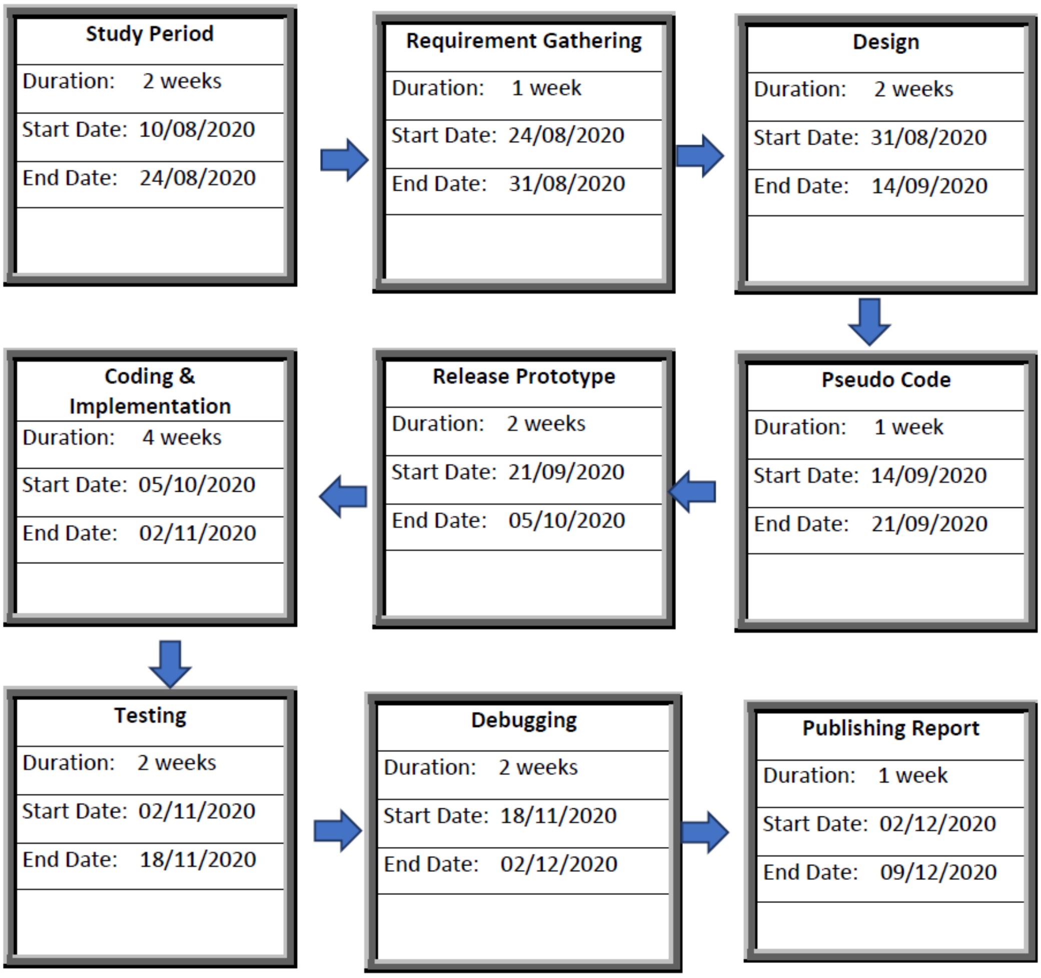
Hardware specification requirements:

Minimum requirements:	***Recommended requirements:***
Processor: Intel Core2 Duo E8400 **Processor Clock Speed:** 1.6 GHz **RAM:** 2 GB or Above **Internet Connection:** 512 Kbps	**Processor:** Intel Core i5-9400F **Processor Clock Speed:** 4.1 GHz **RAM:** 4GB or Above **Internet Connection:** Broadband Connection 10 Mbps

Table 1: System requirements

8. PROJECT SCHEDULE

PERT Chart:



Study Period
Duration: 2 weeks
Start Date: 10/08/2020
End Date: 24/08/2020

Requirement Gathering
Duration: 1 week
Start Date: 24/08/2020
End Date: 31/08/2020

Design
Duration: 2 weeks
Start Date: 31/08/2020
End Date: 14/09/2020

Pseudo Code
Duration: 1 week
Start Date: 14/09/2020
End Date: 21/09/2020

Release Prototype
Duration: 2 weeks
Start Date: 21/09/2020
End Date: 05/10/2020

Coding & Implementation
Duration: 4 weeks
Start Date: 05/10/2020
End Date: 02/11/2020

Testing
Duration: 2 weeks
Start Date: 02/11/2020
End Date: 18/11/2020

Debugging
Duration: 2 weeks
Start Date: 18/11/2020
End Date: 02/12/2020

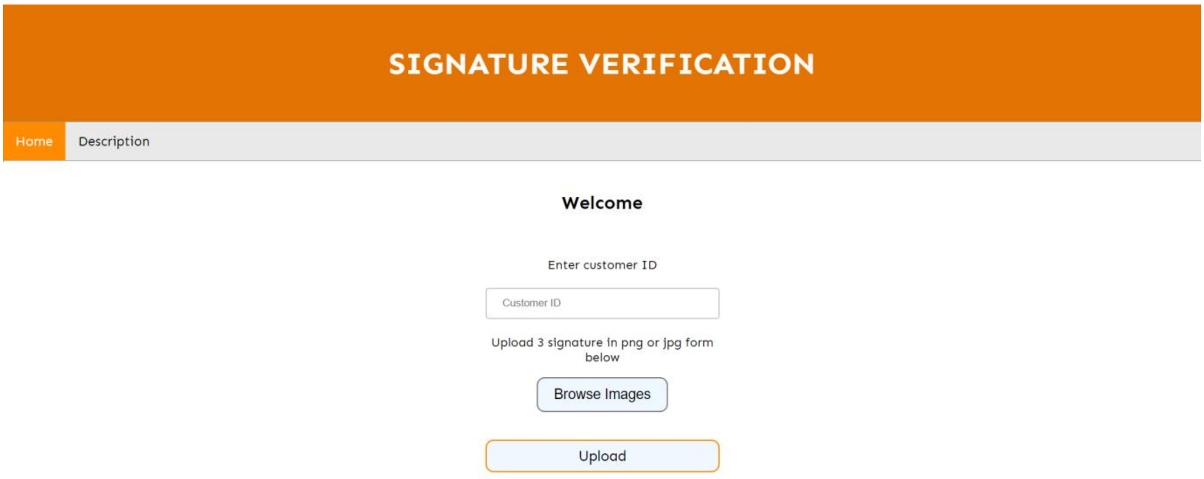
Publishing Report
Duration: 1 week
Start Date: 02/12/2020
End Date: 09/12/2020

Fig 3: PERT Chart

9. RESULT & ANALYSIS

The web application takes 3 original signature images and a test signature image as input from the user. It then checks whether the signature is original or a forged one. The working of the web application is explained below:

The home page contains a form where the user has to enter the customer's ID and select 3 original signature images as references to be compared against the test image, and click on 'upload'.


SIGNATURE VERIFICATION

Welcome

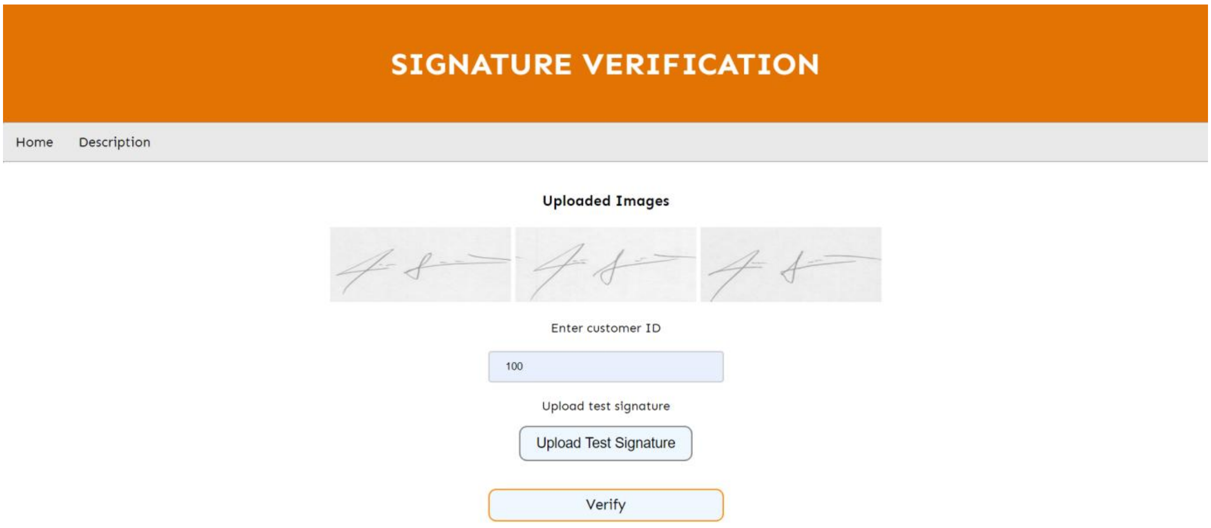
Enter customer ID

Customer ID

Upload 3 signature in png or jpg form below

Fig 4: Uploading 3 Original Signature Images

Upon successful upload, the images are displayed on the web page. The user now needs to upload the image to be tested and click on 'verify' to start the verification process.


SIGNATURE VERIFICATION

Uploaded Images

Enter customer ID

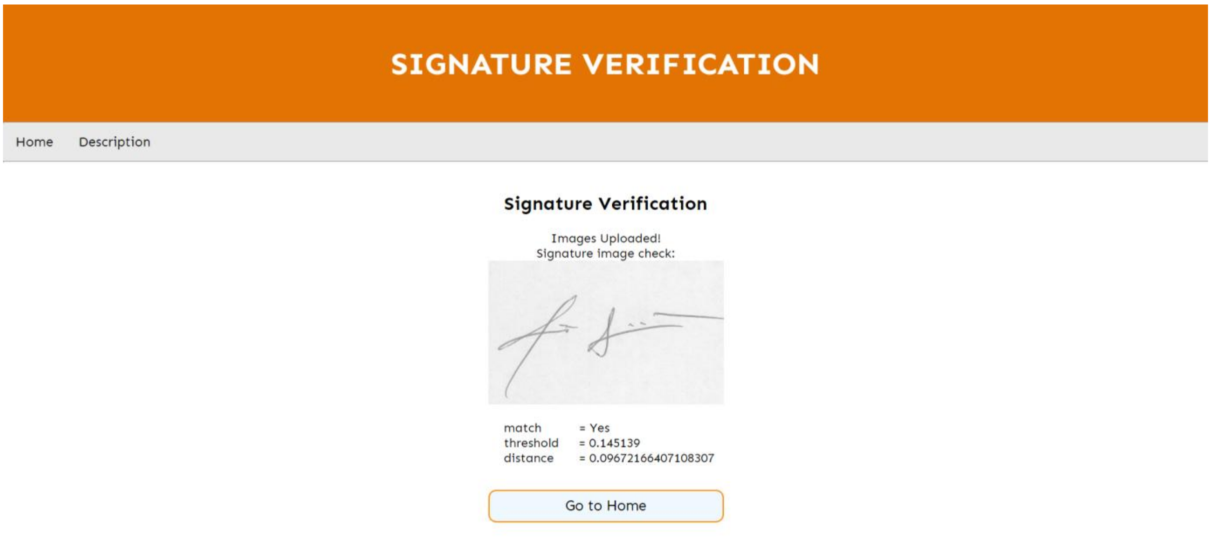
100

Upload test signature

Fig 5: Uploading Signature Image for Verification [Test Image]

The verification process runs in the backend as the test image is compared to the original reference images.

If the test image matches the original, the signature is original.
In this case, following statistics are displayed:



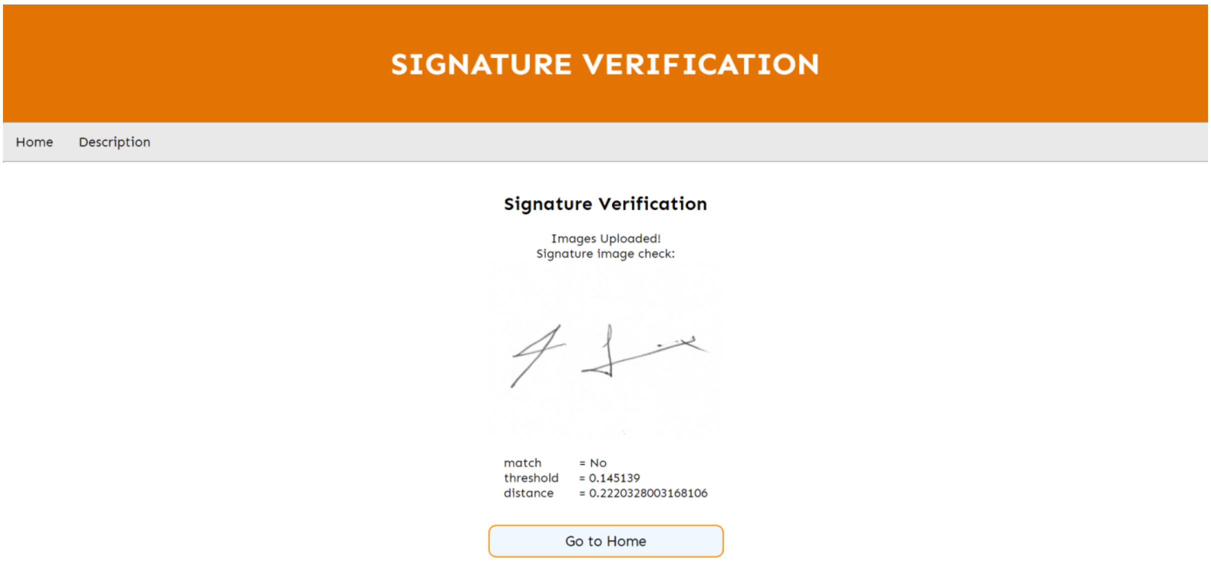
Signature Verification

Images Uploaded!
Signature image check:

```
match      = Yes
threshold  = 0.145139
distance   = 0.09672166407108307
```

Fig 6: Result [Matching Image]

If the test image does not match the original, the signature is forged.
In this case, following statistics are displayed:



Signature Verification

Images Uploaded!
Signature image check:

```
match      = No
threshold  = 0.145139
distance   = 0.2220328003168106
```

Fig 7: Result [Image does not match]

The user can now click on ‘Go to Home’ to return to the home page and test another set of images.

10. CONCLUSION

Over the last decade, researchers have proposed a large variety of methods for Offline Signature Verification. While distinguishing genuine signatures and skilled forgeries remains a challenging task, error rates have dropped significantly in the last few years, mostly due to advancements in Deep Learning applied to the task. Analyzing the recent contributions to the field, we can notice that they concentrate in the following categories:

- Obtaining better features - Several new feature extractors have been proposed for the task. Texture features (LBP variations), interest-point matching (SIFT, SURF) and directional features (HOG) have been successfully used to increase the accuracy of Offline Signature Verification Systems. More recently, feature learning methods have been successfully applied for the task, showing that features learned for a subset of users, generalize to new users, and even users from other datasets.

- Improving classification with limited number of samples - Given the severe constraints in practical applications, researchers have searched for ways to increase performance in cases where a small number of samples per user is available. In particular, the creation of dissimilarity-based writer-independent solutions and metric-learning solutions have shown to be promising to address this problem.

- Augmenting the datasets - Related to the problem of having low number of samples per user, some researchers have focused in generating synthetic signatures, in order to increase the number of samples available for training.

- Building model ensembles - In order to increase classification accuracy, and the robustness of the solutions, some researchers have investigated the creation of both static and dynamic ensembles of classifiers.

11. REFERENCES

This report was prepared with the help of following references **<u>(As per their appearance in the chapters):</u>**

[1] Bartz, Christian & Yang, Haojin & Meinel, Christoph. (2017). STN-OCR: A single Neural Network for Text Detection and Text Recognition.

[2] Vaibhav Goel, Vaibhav Kumar, Amandeep Singh Jaggi, Preeti Nagrath, "Text Extraction from Natural Scene Images using OpenCV and CNN", International Journal of Information Technology and Computer Science(IJITCS), Vol.11, No.9, pp.48-54, 2019. DOI: 10.5815/ijitcs.2019.09.06

[3] Dey, Sounak, et al. "Signet: Convolutional siamese network for writer independent offline signature verification." arXiv preprint arXiv: 1707.02131 (2017).

[4] L. G. Hafemann, R. Sabourin and L. S. Oliveira, "Offline handwritten signature verification — Literature review," 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, QC, 2017, pp. 1-8, doi: 10.1109/IPTA.2017.8310112.

[5] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: CVPR, 2005, pp. 539–546.