BIOMETRIC FINGER VEINS AUTHENTICATION USING DEEP LEARNING ALGORITHM

A PROJECT REPORT

Submitted by

NIVEDHITAA RAVICHANDRAN [211417104172]

PRATISHA B [211417104195]

PREETHI M [211417104198]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600 025

AUGUST 2021

BONAFIDE CERTIFICATE

Certified that this project report **"Biometric Finger Veins Authentication using Deep Learning Algorithm"** is the bonafide work of **"NIVEDHITAA RAVICHANDRAN [211417104172], PRATISHA B [211417104195], PREETHI M [211417104198]"** who carried out the project work under my supervision.

SIGNATURE

Dr.S.MURUGAVALLI,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600123.

SIGNATURE

V. SATHIYA PREIYA
SUPERVISOR
ASSOCIATE PROFESSOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on 05.08.2021.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like express our heartfelt and sincere thanks to our Directors **Tmt. C. VIJAYARAJESWARI**, **Dr. C. SAKTHIKUMAR, M.E., Ph.D.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E., M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr. K. Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S. MURUGAVALLI , M.E., Ph.D.,** forthe support extended throughout the project.

We would like to thank **Mrs. V. SATHIYA PREIYA** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

NIVEDHITAA RAVICHANDRAN

PRATISHA B

PREETHI M

ABSTRACT

The target of this project was the design and development of a finger vein identification system that could be used by a limited number of users in a networked environment. Finger vein authentication can be a leading biometric technology nowadays in terms of security and convenience, since it introduces the features inside the human body. An image of a finger captured by the web camera under the IR light transmission contains not only the vein pattern itself, but also shade produced by various thickness of the finger muscles, bones, and tissue networks surrounding the vein. We introduce preliminary process to enhance the image quality worsened by light effect and noise produced by the web camera, then segment the vein pattern by using adaptive threshold method and matched them using improved template matching. The experimental result shows that even the image quality is not good, as long as our veins are clear and also with some appropriate process it still can be used as the means of personal identification. Such tools include image acquisition, pre-processing, feature extraction and matching methods to extract and analyze object patterns.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.NO.	ABBREVATION	EXPLANATION
1.	EDA	Exploratory Data Analysis
2.	CNN	Convolution Neural network
3.	SVM	Support Vector Machine
4.	ANN	Artificial Neural Network
5.	UML	Unified Modelling Language
6.	API	Application Programming Interface
7.	NIR	Near Infra-Red

CHAPTER 1

INTRODUCTION

1.1 Overview

Biometric authentication is a security process that relies on the uniquebiological characteristics of an individual to verify that he is who he says he is. Biometric authentication technology compares biometric data capture to stored, confirmed authentic data in a database. Finger vein recognition is a method that specifiesan individual using the vein pattern inside one’s fingers. Finger vein patterns are unique to each individual, even among identical twins. The false acceptance rate is very low (close to zero). Placing a hand or finger is less intrusive compared to other biometric technologies. Because veins are located inside the body, it is extremely difficult to read or steal and hence cannot be duplicated. Finger veins are less likely to be influenced by changes in the weather or physical condition of the individual.

1.2 Problem Definition

Finger vein patterns are unique to each individual, even among identical twins. Finger vein authentication works by utilizing the differences of vein patterns for personal identification. The authentication process is carried out by comparing the vein pattern that is previously captured during enrollment to the vein pattern that is taken during authentication, and checking if they match. To achieve this target, it’s necessary to make a combination between hardware design and software procedures. Through hardware design, the system may instruct the user to perform properly to make the pattern recognized easily. Through software procedures, the system may correct the problems related to the acquisition of the pattern, relying on algorithms that aid to solve irregularities. Furthermore, the algorithm is able to create the score of similarity in biometrical character from extractedfingerprints, the result of which is assumed to be accurate with infinite decimal.

CHAPTER 2

LITERATURE SURVEY

- **Finger Vein Recognition with Anatomy Structure Analysis**

Finger vein recognition has received a lot of attention recently and is viewed as a promising biometric trait. In related methods, vein pattern-based methods explore intrinsic finger vein recognition, but their performance remains unsatisfactory owing to defective vein networks and weak matching. One important reason may be the neglect of deep analysis of the vein anatomy structure. By comprehensively exploring the anatomy structure and imaging characteristic of vein patterns, this paper proposes a novel finger vein recognition framework, including an anatomy structure analysis-based vein extraction (ASAVE) algorithm and an integration matching strategy. Specifically, the vein pattern is extracted by the orientation map-guided curvature based on the valley- or half valley-shaped cross-sectional profile. In addition, the extracted vein pattern is further thinned and refined to obtain a reliable vein network.

- **Tri-branch Vein Structure Assisted Finger Vein Recognition**

In template matching of finger vein recognition, the probe will be accepted if the number of its overlapped vein points with the enrolled user is larger than the predefined threshold. However, the acceptance may be false owing to ignoring the structure of the vein pattern. We find that local vein branches near the bifurcation point of vein pattern vary largely between the imposter images. So, this paper tries to explore this kind of local vein structure to improve the recognition performance of the template matching. The bifurcation point and its local vein branches, named tri-branch vein structure, are extracted from the vein pattern, and fused with the whole vein pattern by a user-specific threshold-based filter framework. The experimental results on two public databases prove the effectiveness of the proposed framework for improving the performance of vein pattern-based finger vein recognition

- **Cancelable Permutation-Based Indexing for Secure and Efficient Biometric Identification**

  This paper schemes transform biometric features and perform pattern matching without restoring the original features. Although they strongly prevent the leakage of the original features, the response time can be very long in a large-scale biometric identification system. Most of the existing indexing schemes cannot be used to speed up the biometric identification system over networks since a biometric index leaks some information about the original feature. Secure and efficient indexing is a major challenge in large-scale biometric identification over networks. In this paper, we propose a novel indexing scheme that is promising with regard to both security and efficiency. The proposed indexing scheme transforms a permutation-based index, which is the state-of-the-art index in the field of similarity search, and performs a query search without recovering the original index.

- **Design and Development of low-cost Sensor to capture ventral and dorsal Finger-vein for Biometric Authentication**

  Biometrics-based authentication of subjects is widely deployed in several real-life applications. Among various biometric characteristics, finger-vein characteristic has demonstrated both reliable and highly accurate authentication for access control in secured applications. However, most of these systems are based on commercial sensors where the image level data is not available for academic research. In this paper, we present the design and development of a low-cost finger-vein sensor based on a single camera that can capture finger-vein images from dorsal and ventral part of the finger with high quality. The system consists of multiple Near-Infra-Red (NIR) light sources to illuminate the finger from both sides (left and right) and top. The camera in the sensor is also coupled with the custom designed physical structure to facilitate high reflectance of the emitted light and distribute the light uniformly on the finger to capture good quality dorsal and ventral finger-vein pattern.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

These biometric recognition approaches can be divided into two categories:

1. Extrinsic biometric features (palm print, iris, fingerprint, face)
2. Intrinsic biometric features (palm vein, hand vein, and finger vein)

Existing Approaches

- vessel extraction
- subspace-learning-based approaches
- statistical-based techniques
- local-invariant based methods

3.2 Proposed System

Finger vein recognition has emerged as the robust biometric modality because of their unique vein pattern that can be captured using near infrared spectrum. The large-scale finger vein based biometric solutions demand the need of searching the probe finger vein sample against the large collection of gallery samples. In order to improve the reliability in searching for the suitable identity in the large-scale finger vein database, it is essential to introduce the finger vein indexing and retrieval scheme. It proposes a biometric system to identify people based on the pattern of finger vein. The system uses a database of human index finger images acquired on infrared range. The present proposal has applied Sobel detector, enhancement filter and a binarization process to get the vein pattern. The proposed system is implemented using novel finger vein recognition algorithm. Dimension and Gabor filter are the algorithms used for feature extraction and using the distance classifier the matching of the extracted feature is done.

3.3 Requirement Analysis and Specification

3.3.1 Input Requirements

- Disk Space 32 GB or more,10 GB or more for Foundation Edition
- Processor 1.4 GHz 64 bit
- Memory 512 MB
- Display (800 × 600) Capable video adapter and monitor
- Python

3.3.2 Output Requirements

- Jupyter Notebook

3.3.3 Functional Requirements

Functional requirements is the inputs to the software system, its behavior, and outputs.

- Load the dataset.
- Analyze the dataset.
- Create model and train the model.
- Test for the accuracy.
- Evaluate and Predict the class.

3.4 Technology Stack

- **Python**

Python is a free, open-source programming language. Python is also a cross-platform compatible language. Python is also a great visualization tool. It provides libraries such as Matplotlib, seaborn and bokeh to create stunning visualizations. In addition, Python is the most popular language for machine learning and deep learning. As a matter of fact, today, all top organizations are investing in Python to implement machine learning in the back-end.

- **NumPy**

  NumPy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. NumPy is a Python library that is the core library for scientific computing in Python. It contains a collection of tools and techniques that can be used to solve on a computer mathematical models of problems in Science and Engineering. One of these tools is a high-performance multidimensional array object that is a powerful data structure for efficient computation of arrays and matrices.

- **Scikit-learn**

  Scikit-learn (formerly **scikits.learn** and also known as **sklearn**) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn integrates well with many other Python libraries, such as matplotlib and plotly for plotting, NumPy for array vectorization, pandas data frames, SciPy, and many more.

- **TensorFlow & Keras**

  TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. It was created and is maintained by Google and released under the Apache 2.0 open-source license. The API is nominally for the Python programming language, although there is access to the underlying C++ API. Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

- **Jupyter Notebook**

The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. A Notebook extension (nb extension) is a JavaScript module that you load in most of the views in the Notebook's frontend.

CHAPTER 4

SYSTEM DESIGN

4.1 UML Diagrams

4.1.1 Use case diagram

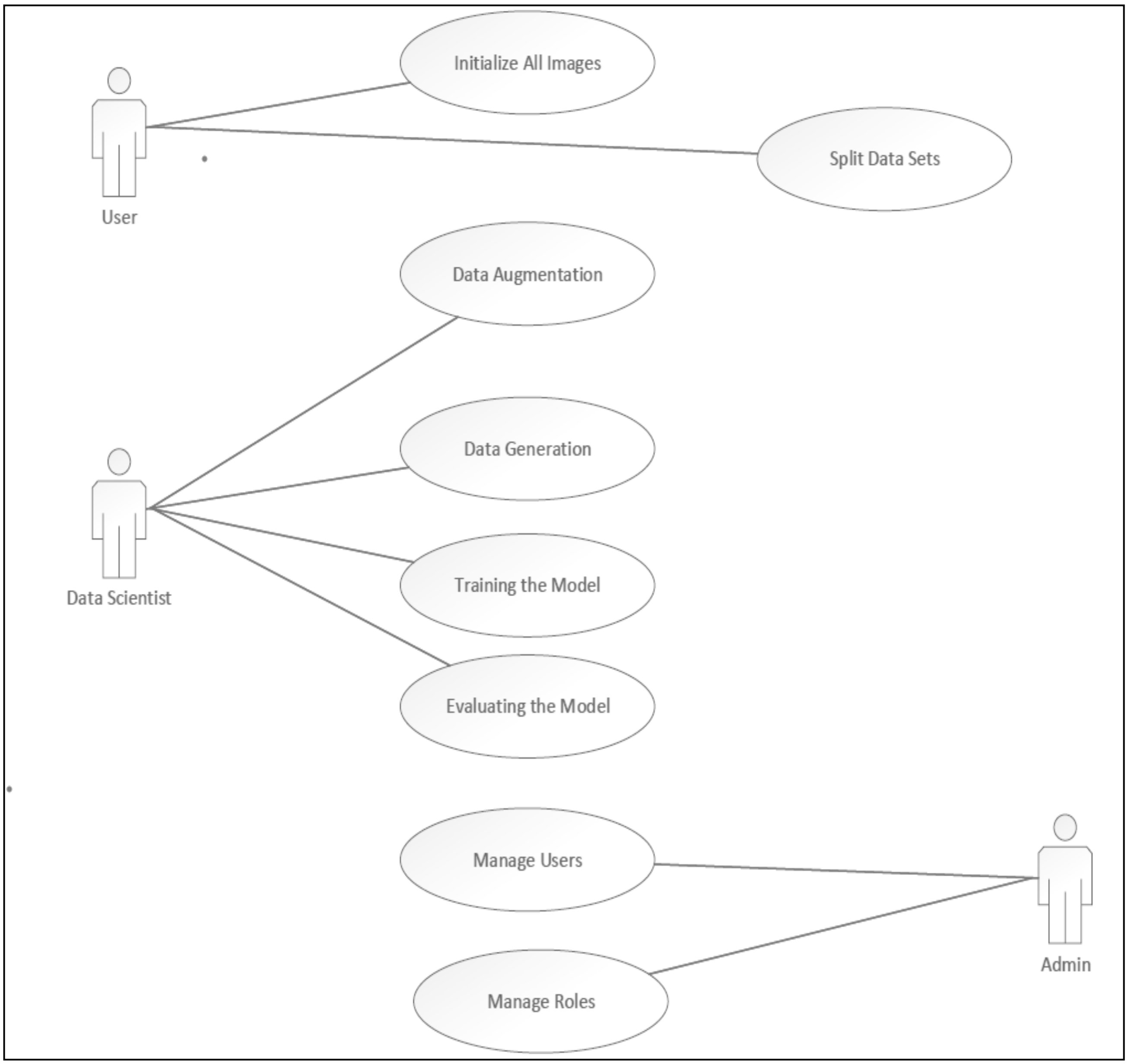


Fig.4.1.1 Use Case Diagram

4.1.2 Activity diagram

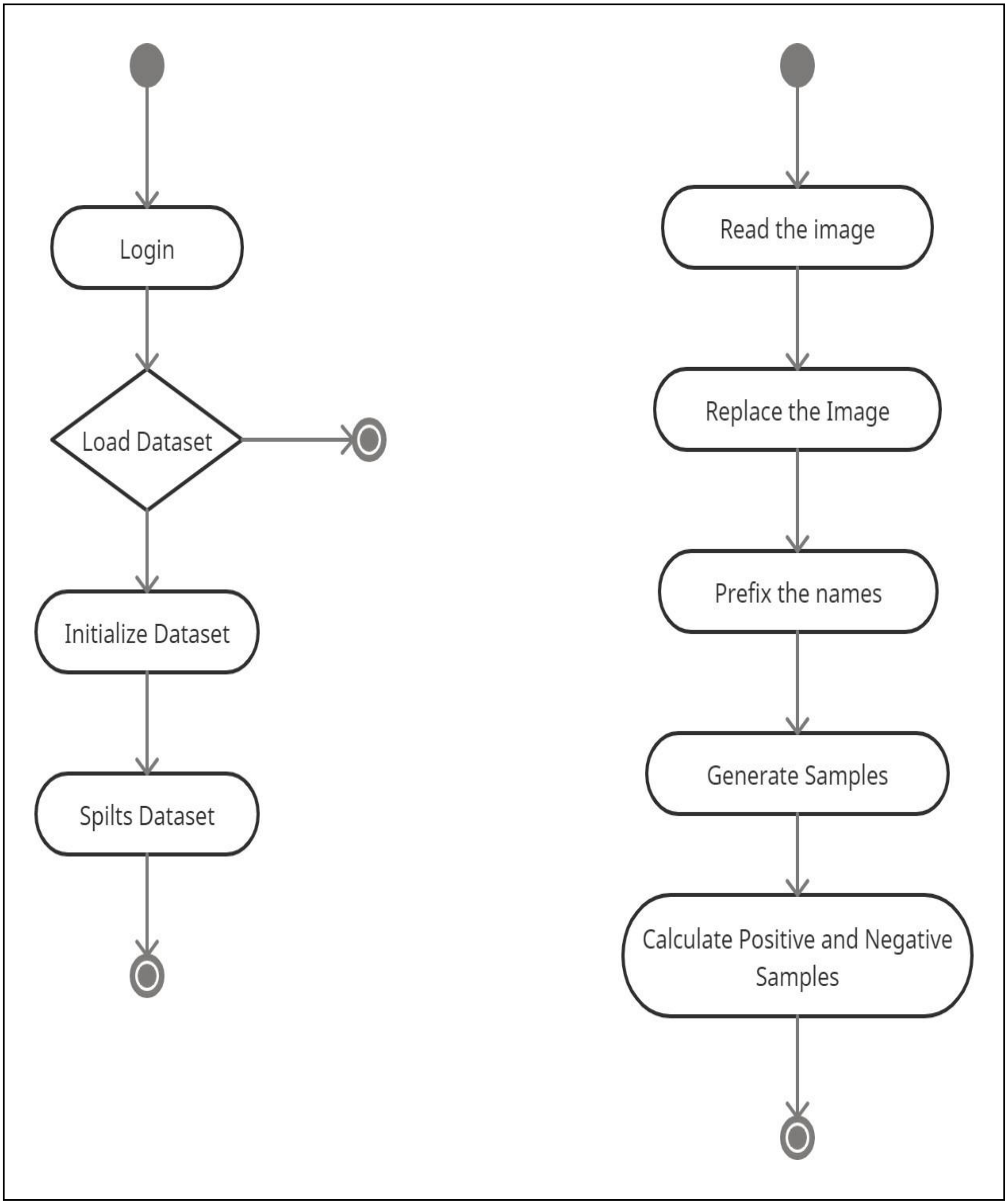


Fig. 4.1.2 Activity Diagram

4.1.3 Sequence diagram

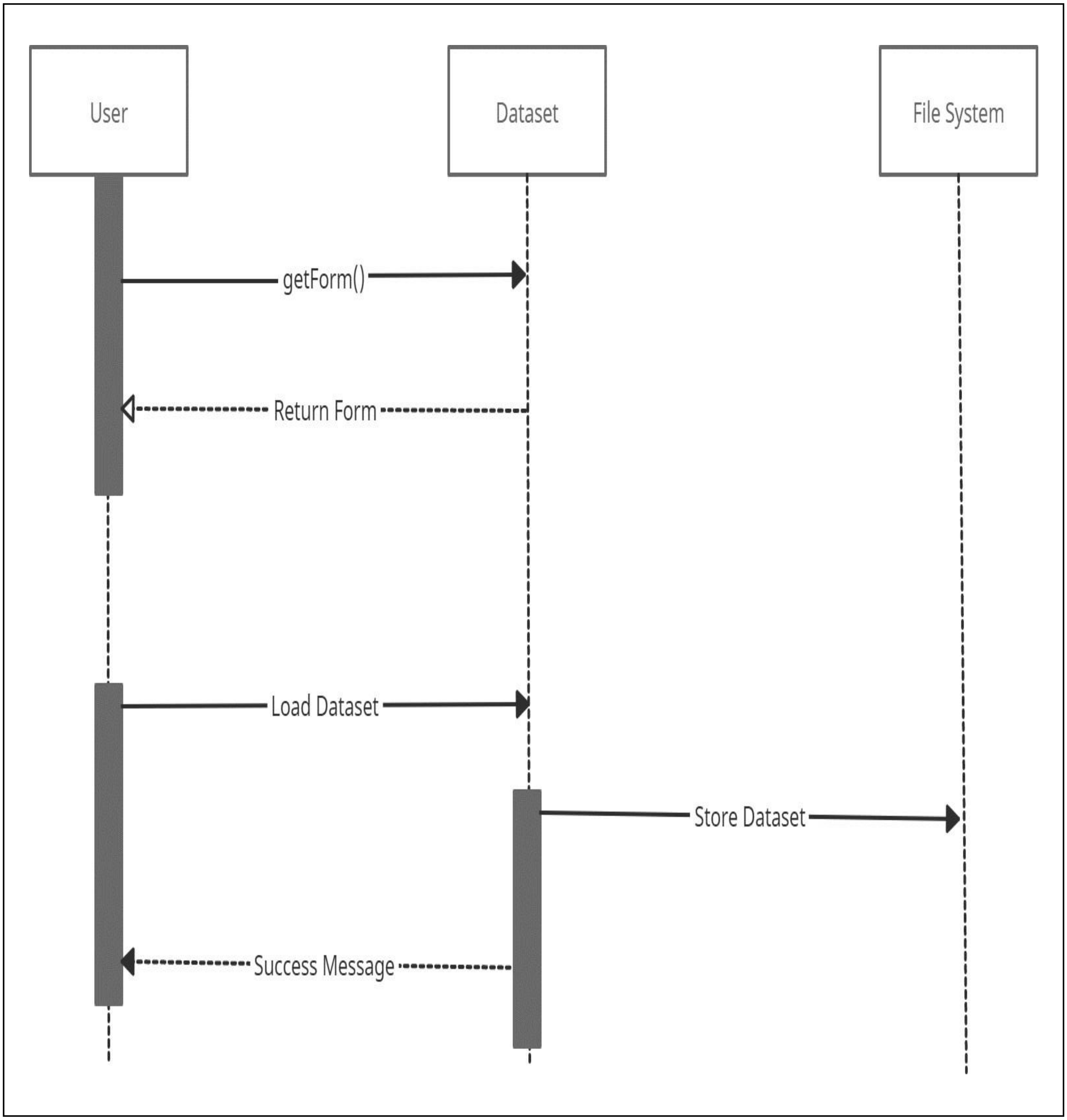


Fig. 4.1.3 Sequence Diagram

4.1.4 Class diagram

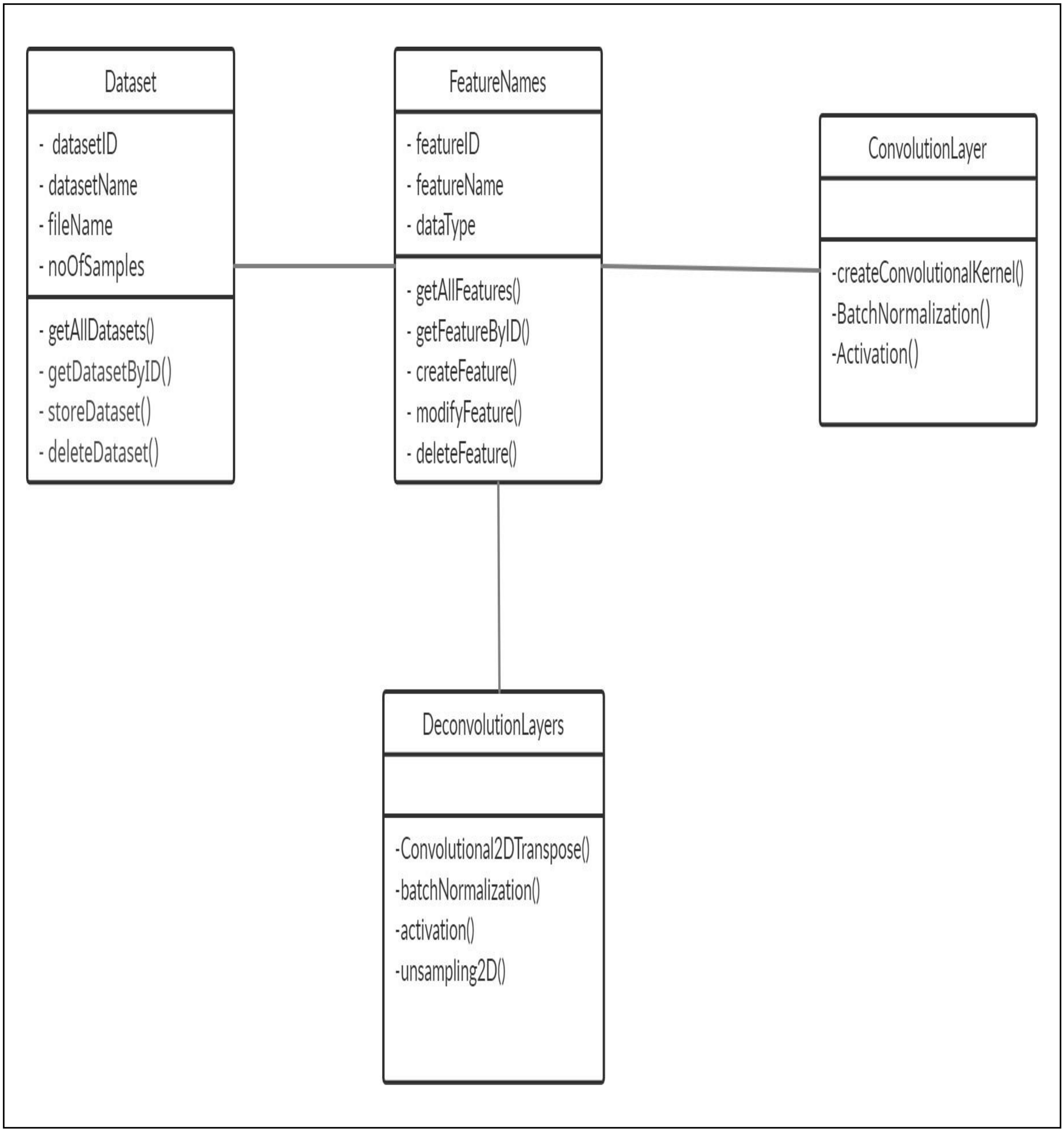


Fig. 4.1.4 Class Diagram

4.1.5 Dataflow diagram

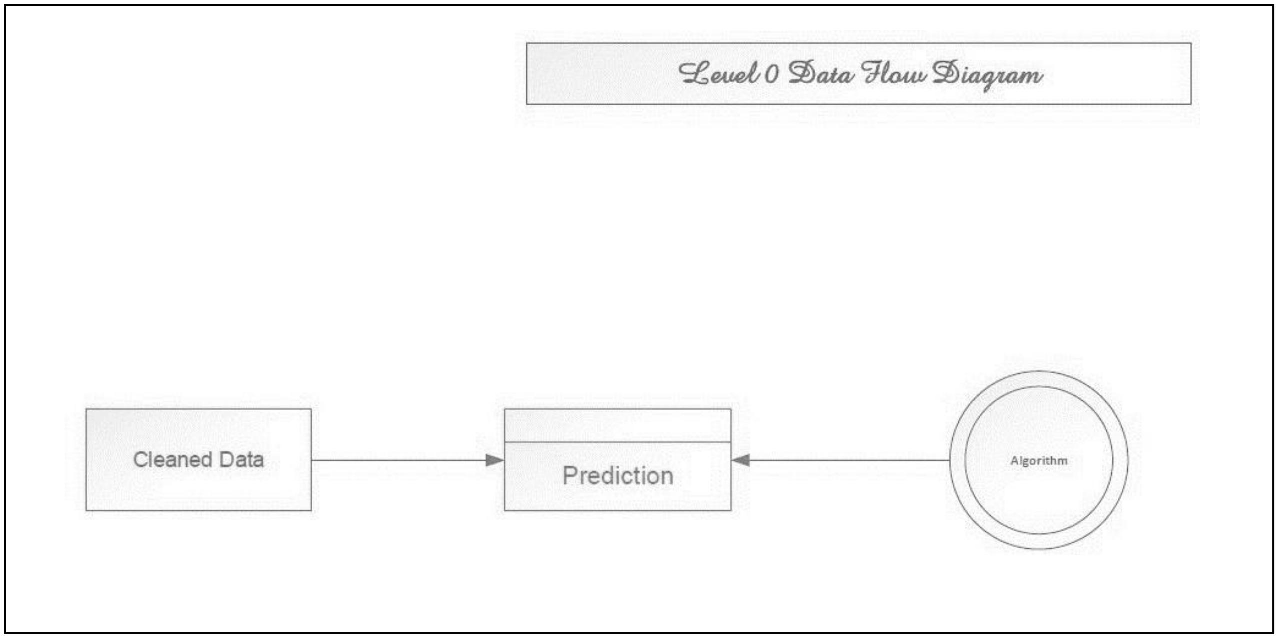


Fig. 4.1.5a Dataflow Diagram Level 0

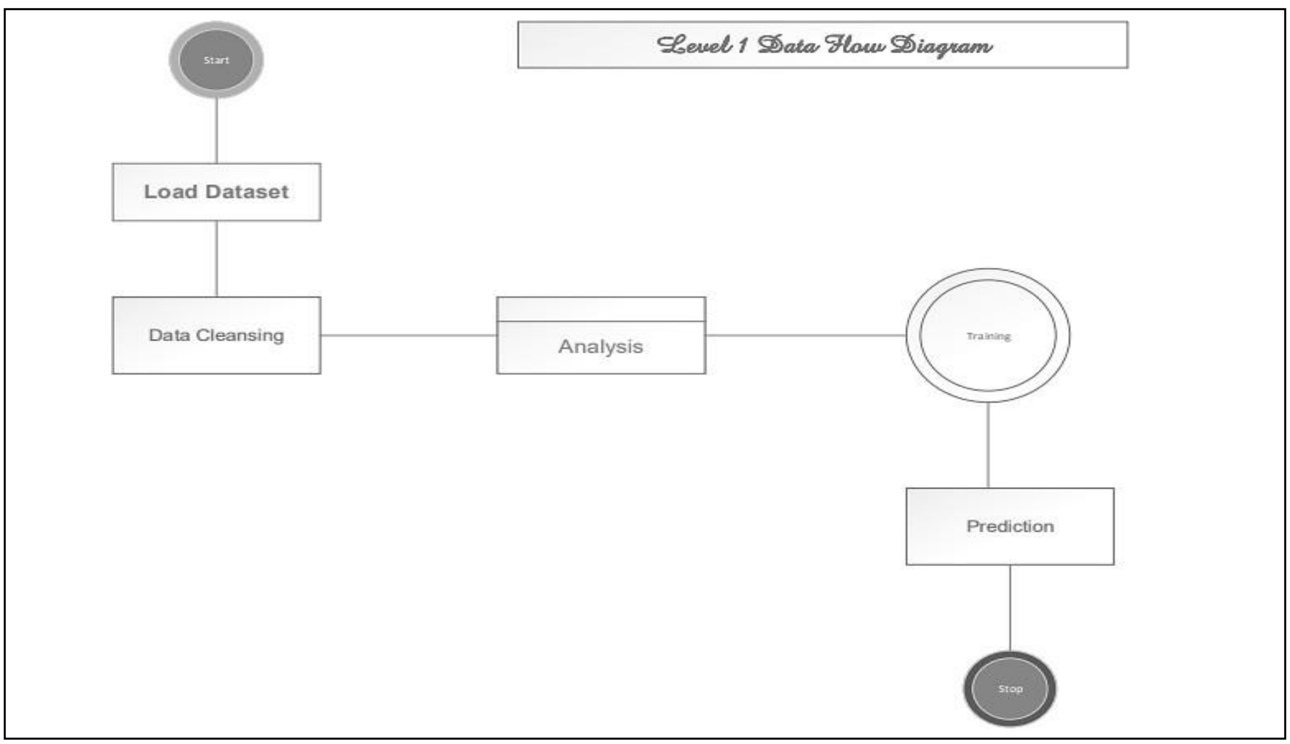


Fig. 4.1.5b Dataflow Diagram Level 1

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 Architecture Overview

Initially the dataset is divided into three namely: Training Data, Testing Data and Validation Data. The images from the Testing Dataset undergo feature extraction process to extract all the required features to analyze the image and to produce the appropriate result. Then these data will be processed further for the feature selection and the system is trained using deep learning algorithm. Once training is complete, the system is ready for authenticating the users.

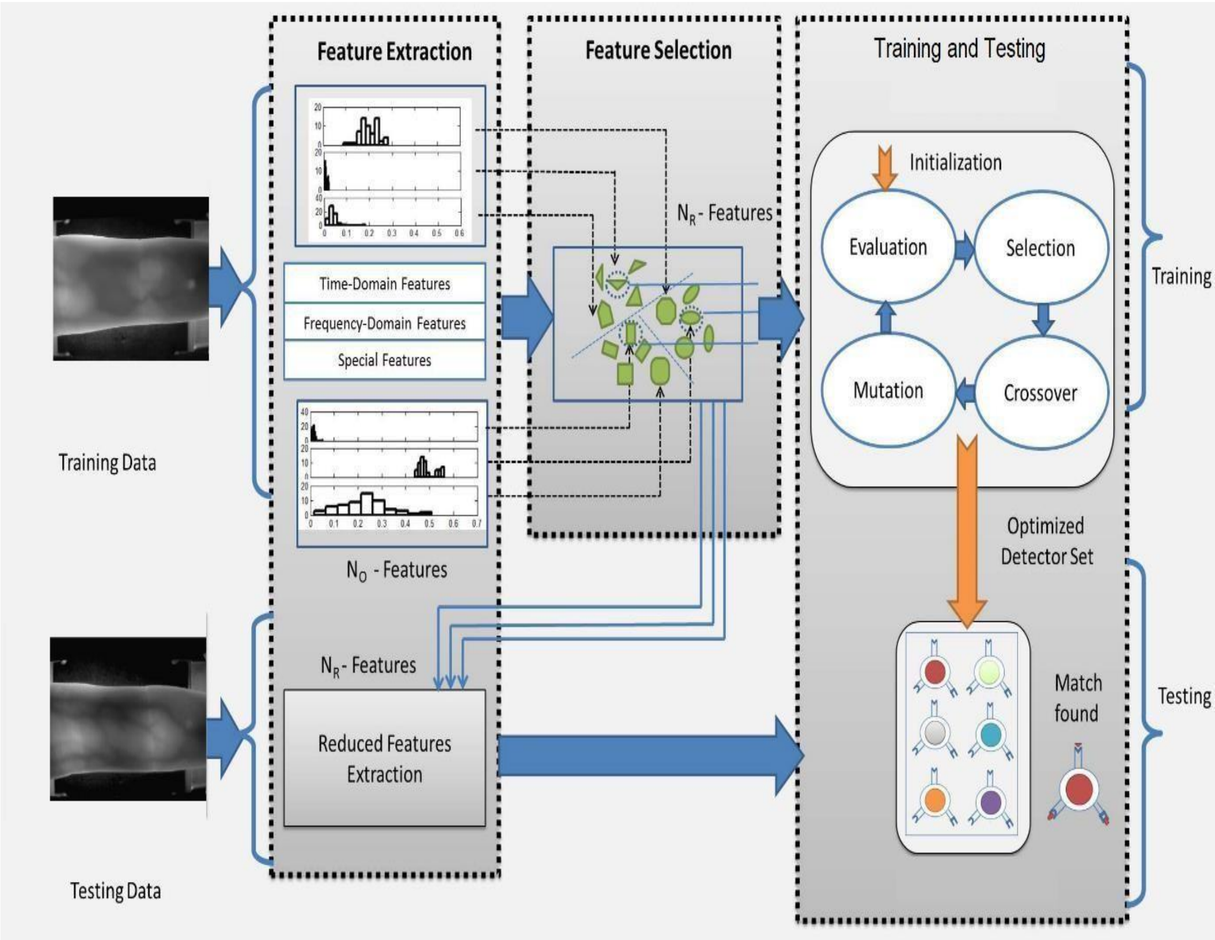


Fig. 5.1 System architecture of Finger Vein Authentication

5.2 Module Design Specification

The Proposed System consists of the following Modules:

Module 1: Exploratory Data Evaluation

Module 2: Pre-processing

Module 3: Feature Engineering

Module 4: Prediction

Exploratory data evaluation

Exploratory Data Analysis (EDA) is the first step in data analysis process. Exploratory Data Analysis is valuable to data science projects since it allows to get closer to the certainty that the future results will be valid, correctly interpreted, and applicable to the desired business contexts. Such level of certainty can be achieved only after raw data is validated and checked for anomalies, ensuring that the data set was collected without errors. EDA also helps to find insights that were not evident or worth investigating to business stakeholders and data scientists but can be very informative about a particular business.

EDA is performed in order to define and refine the selection of feature variables that will be used for machine learning. Once data scientists become familiar with the data set, they often have to return to feature engineering step, since the initial features may turn out not to be serving their intended purpose. Once the EDA stage is complete, data scientists get a firm feature set they need for supervised and unsupervised machine learning.

Pre-processing

Sometimes you may find some data are missing in the dataset. We need to be equipped to handle the problem when we come across them. One of the most common idea to handle the problem is to take a mean of all the values of the same column and have it to replace the missing data.

The library that we are going to use for the task is called Scikit Learn preprocessing. It contains a class called Imputer which will help us take care of the missing data. Sometimes our data is in qualitative form, that is we have texts as our data. We can find categories in text form. Now it gets complicated for machines to understand texts and process them, rather than numbers, since the models are based on mathematical equations and calculations. Therefore, we have to encode the categorical data. Now we need to split our dataset into two sets — a Training set and a Test set. We will train our machine learning models on our training set, i.e our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to check how accurately it can predict. A general rule of the thumb is to allocate 80% of the dataset to training set and the remaining 20% to test set. For this task, we will import test_train_split from model_selection library of scikit.

Feature selection

Feature selection is a process where you automatically select those features in your data that contribute most to the prediction variable or output in which you are interested.

Having irrelevant features in your data can decrease the accuracy of many models, especially linear algorithms like linear and logistic regression.

Three benefits of performing feature selection before modeling your data are:

- Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.
- Improves Accuracy: Less misleading data means modeling accuracy improves.
- Reduces Training Time: Less data means that algorithms train faster.

The number of pixels in an image is the same as the size of the image for grayscale images we can find the pixel features by reshaping the shape of the image and returning the array form of the image. Edges in an image are the corners where the pixel change drastically, as the images are stored in array form, we can visualize different values and see where the change in pixel value is higher but doing it manually takes time.

Prediction

Once training is complete, it’s time to see if the model is any good, using Evaluation. This is where that dataset that we set aside earlier comes into play. Evaluation allows us to test our model against data that has never been used for training. This metric allows us to see how the model might perform against data that it has not yet seen. This is meant to be representative of how the model might perform in the real world.

A good rule of thumb I use for a training-evaluation split somewhere on the order of 80/20 or 70/30. Much of this depends on the size of the original source dataset. If you have a lot of data, perhaps you don’t need as big of a fraction for the evaluation dataset. Once you’ve done evaluation, it’s possible that you want to see if you can further improve your training in any way. We can do this by tuning our parameters. There were a few parameters we implicitly assumed when we did our training, and now is a good time to go back and test those assumptions and try other values.

5.3 PROGRAM DESIGN LANGUAGE:

Convolution Neural Network (CNN) is a class of Deep learning algorithms. CNN is an algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. CNN works by extracting features from the image any CNN consists of the following:

1. The input layer which is a gray scale image.
2. The output layer which is a binary or multi-class labels.
3. Hidden layers consisting of convolution layers, ReLu (rectified linear unit) layers, the pooling layers, and a fully connected Neural Network.

CNNs are used for image classification and recognition because of its high accuracy. The CNN follows a hierarchical model which works on building a network, like a funnel, and finally gives out a fully-connected layer where all the neurons are connected to each other and the output is processed.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 SERVER-SIDE CODING:

```
from IPython.display import clear_output

clear_output()

import numpy as np

from tqdm import tqdm

import cv2

import os

import shutil

import itertools

import imutils

import matplotlib.pyplot as plt

from sklearn.preprocessing import LabelBinarizer

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score, confusion_matrix

import plotly.graph_objs as go

from plotly.offline import init_notebook_mode, iplot

from plotly import tools

from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D

from tensorflow.keras.layers import Activation, Dropout, Flatten, Dense

from tensorflow.keras import backend as K

from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input

from tensorflow.keras import layers

from tensorflow.keras.models import Model, Sequential

from tensorflow.keras.optimizers import Adam, RMSprop

from tensorflow.keras.callbacks import EarlyStopping

from sklearn.utils import Bunch

from sklearn import svm, metrics, datasets

from sklearn.model_selection import GridSearchCV, train_test_split

from skimage.io import imread

from skimage.transform import resize

from pathlib import Path

from sklearn import tree

from sklearn.neighbors import KNeighborsClassifier

init_notebook_mode(connected=True)

RANDOM_SEED = 123

IMG_PATH = 'Dataset/'
```

```
image = cv2.imread('Dataset/001/index_1.bmp')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.figure(figsize=(20, 20))
plt.subplot(1,  2,  1)
plt.title("Original")
plt.imshow(image)
kernel_sharpening = np.array([[-1,-1,-1],
                              [-1,9,-1],
                              [-1,-1,-1]])
sharpened = cv2.filter2D(image, -1, kernel_sharpening)
plt.subplot(1, 2, 2)
plt.title("Image Sharpening")
plt.imshow(sharpened)
plt.show()
image = cv2.imread('Dataset/002/index_2.bmp', 0)
plt.figure(figsize=(30, 30))
plt.subplot(3,  2,  1)
plt.title("Original")
plt.imshow(image)
ret,thresh1 = cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)
```

```
plt.subplot(3, 2, 2)

plt.title("Threshold Binary")

plt.imshow(thresh1)

image = cv2.GaussianBlur(image, (3, 3), 0)

thresh = cv2.adaptiveThreshold(image, 255, cv2.ADAPTIVE_THRESH_MEAN_C,
cv2.THRESH_BINARY, 3, 5)

plt.subplot(3, 2, 3)

plt.title("Adaptive Mean Thresholding")

plt.imshow(thresh)

_, th2 = cv2.threshold(image, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

plt.subplot(3, 2, 4)

plt.title("Otsu's Thresholding")

plt.imshow(th2)

plt.subplot(3, 2, 5)

blur = cv2.GaussianBlur(image, (5,5), 0)

_, th3 = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

plt.title("Guassian Otsu's Thresholding")

plt.imshow(th3)

plt.show()

image = cv2.imread('Dataset/003/index_3.bmp')
```

```
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

plt.figure(figsize=(20, 20))

plt.subplot(3,  2,  1)

plt.title("Original")

plt.imshow(image)

kernel = np.ones((5,5), np.uint8)

erosion = cv2.erode(image, kernel, iterations = 1)

plt.subplot(3, 2, 2)

plt.title("Erosion")

plt.imshow(erosion)

dilation = cv2.dilate(image, kernel, iterations = 1)

plt.subplot(3, 2, 3)

plt.title("Dilation")

plt.imshow(dilation)

opening = cv2.morphologyEx(image, cv2.MORPH_OPEN, kernel)

plt.subplot(3, 2, 4)

plt.title("Opening")

plt.imshow(opening)

closing = cv2.morphologyEx(image, cv2.MORPH_CLOSE, kernel)

plt.subplot(3, 2, 5)
```

```
plt.title("Closing")

plt.imshow(closing)

image = cv2.imread('Dataset/004/middle_1.bmp', 0)

image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

height, width,_ = image.shape

sobel_x = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=5)

sobel_y = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=5)

plt.figure(figsize=(20, 20))

plt.subplot(3, 2, 1)

plt.title("Original")

plt.imshow(image)

plt.subplot(3, 2, 2)

plt.title("Sobel X")

plt.imshow(sobel_x)

plt.subplot(3, 2, 3)

plt.title("Sobel Y")

plt.imshow(sobel_y)

sobel_OR = cv2.bitwise_or(sobel_x, sobel_y)

plt.subplot(3, 2, 4)

plt.title("sobel_OR")
```

```
plt.imshow(sobel_OR)

laplacian = cv2.Laplacian(image, cv2.CV_64F)

plt.subplot(3, 2, 5)

plt.title("Laplacian")

plt.imshow(laplacian)

canny = cv2.Canny(image, 50, 120)

plt.subplot(3, 2, 6)

plt.title("Canny")

plt.imshow(canny)

image = cv2.imread('Dataset/005/index_6.bmp')

image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

plt.figure(figsize=(20, 20))

plt.subplot(2,  2,  1)

plt.title("Original")

plt.imshow(image)

image_scaled = cv2.resize(image, None, fx=0.75, fy=0.75)

plt.subplot(2, 2, 2)

plt.title("Scaling - Linear Interpolation")

plt.imshow(image_scaled)

img_scaled = cv2.resize(image, None, fx=2, fy=2, interpolation = cv2.INTER_CUBIC)
```

```
plt.subplot(2, 2, 3)

plt.title("Scaling - Cubic Interpolation")

plt.imshow(img_scaled)

img_scaled = cv2.resize(image, (900, 400), interpolation = cv2.INTER_AREA)

plt.subplot(2, 2, 4)

plt.title("Scaling - Skewed Size")

plt.imshow(img_scaled)
```

CHAPTER 7

SYSTEM TESTING

CONFUSION MATRIX

N=65	True Positive	True Negative
Predicted Positive	50	4
Predicted Negative	6	5

7.1 CNN Algorithm Confusion Matrix

N=65	True Positive	True Negative
Predicted Positive	45	11
Predicted Negative	4	5

7.2 SVM Algorithm Confusion Matrix

N=65	True Positive	True Negative
Predicted Positive	45	9
Predicted Negative	5	6

7.3 ANN Algorithm Confusion Matrix

7.1 PERFORMANCE ANALYSIS MEASURES

Measure	CNN Value	SVM Value	ANN Value
Sensitivity	0.8929	0.9184	0.9000
Specificity	0.5556	0.3125	0.4000
Precision	0.9259	0.8036	0.8333
Negative PredictiveValue	0.4545	0.5556	0.5455
False Positive Rate	0.4444	0.6875	0.6000
False Discovery Rate	0.0741	0.1964	0.1667
False Negative Rate	0.1071	0.0816	0.1000
Accuracy	0.8462	0.7682	0.7846
F1 Score	0.9091	0.8571	0.8645
Matthews CorrelationCoefficient	0.4130	0.2879	0.3371

CHAPTER 8

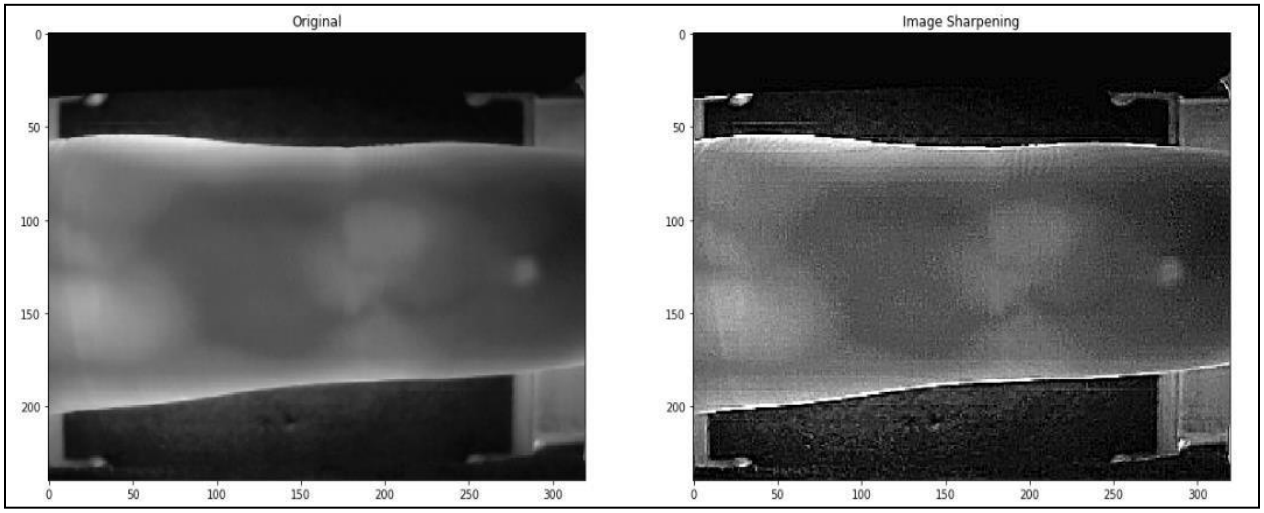
CONCLUSION

8.1 CONCLUSION AND FUTURE ENHANCEMENT

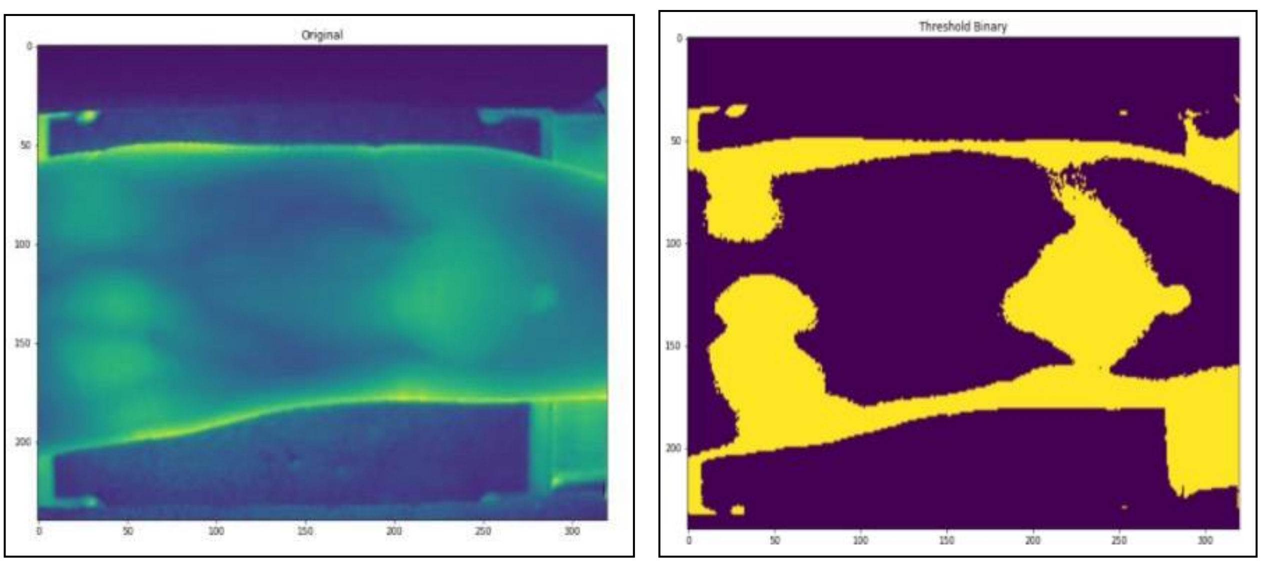
The proposed method is suitable for quantification in the early stage and for use in large cohorts. It may be expected that system can be detected timelier with our quantitative measurements than by ordinal scores. Our future research will notably include applying the automatic approach to large-scale cohorts, so as to examine whether the approach has an increased sensitivity to change.

APPENDICES

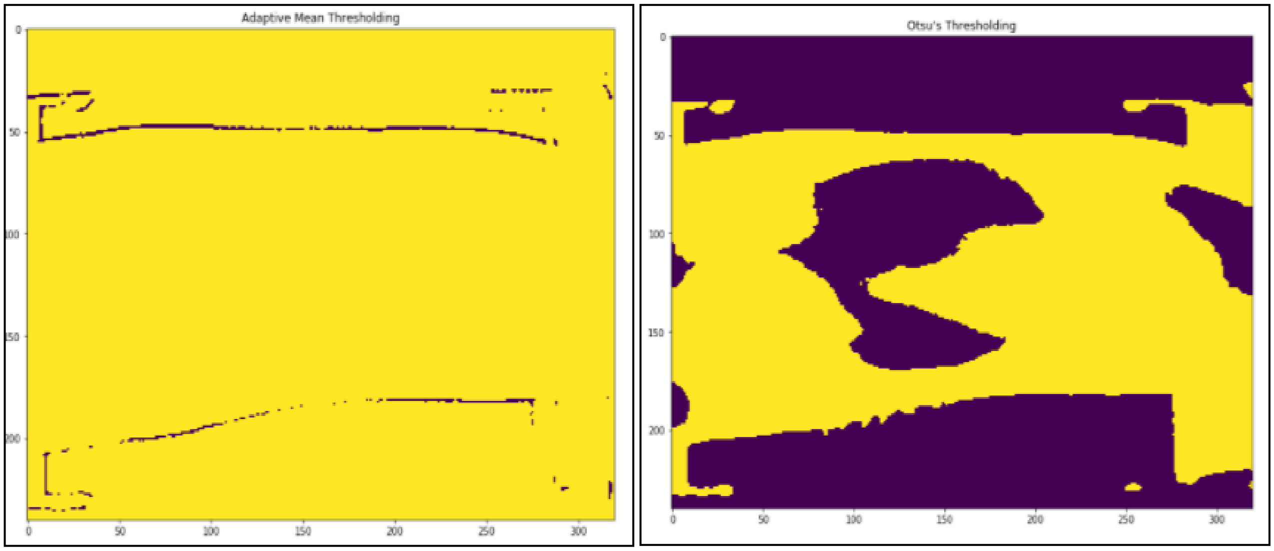
A.1 SAMPLE SCREENSHOTS



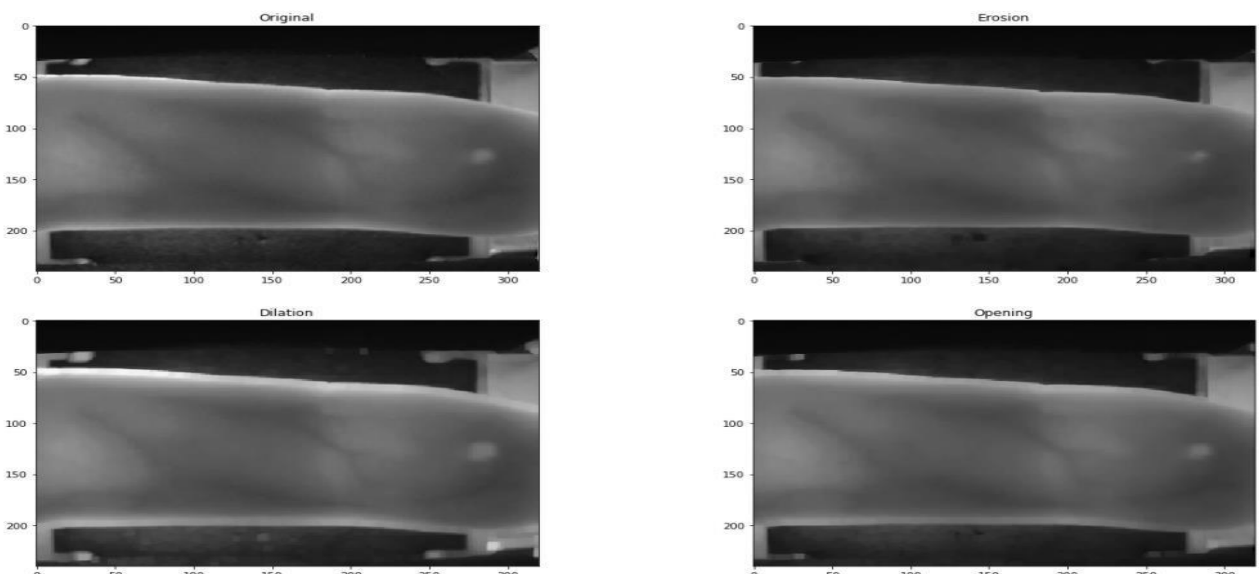
A.1.1 Sharpening of the identified veins



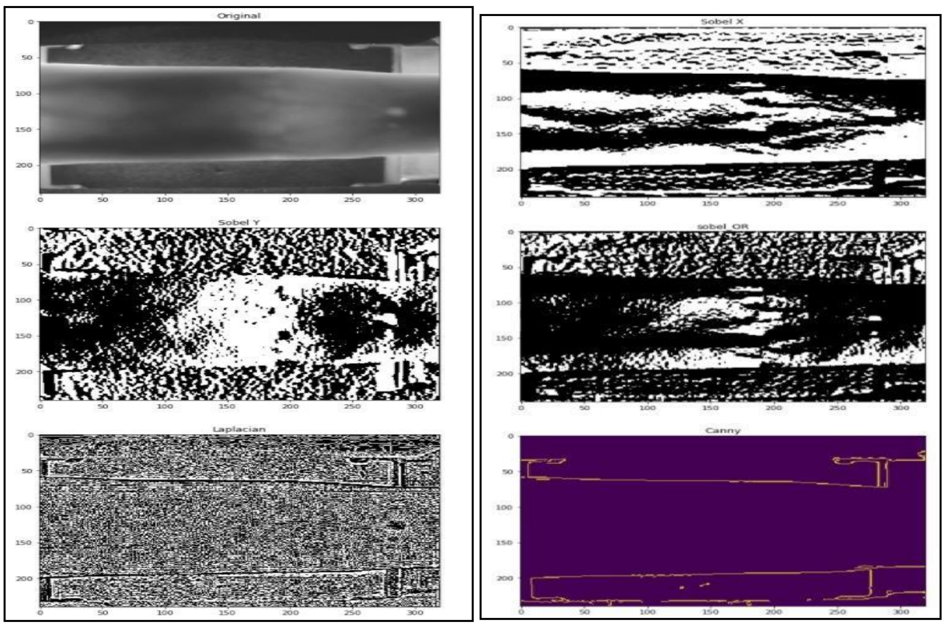
A.1.2 Threshold Binary done to create binary image



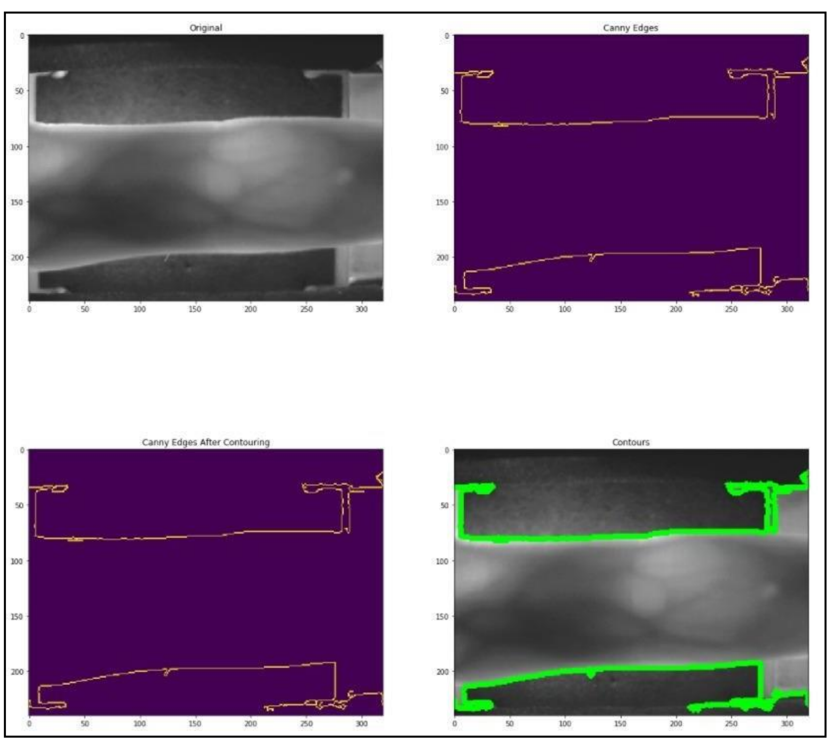
A.1.3 Adaptive Mean and Otsu's Threshold



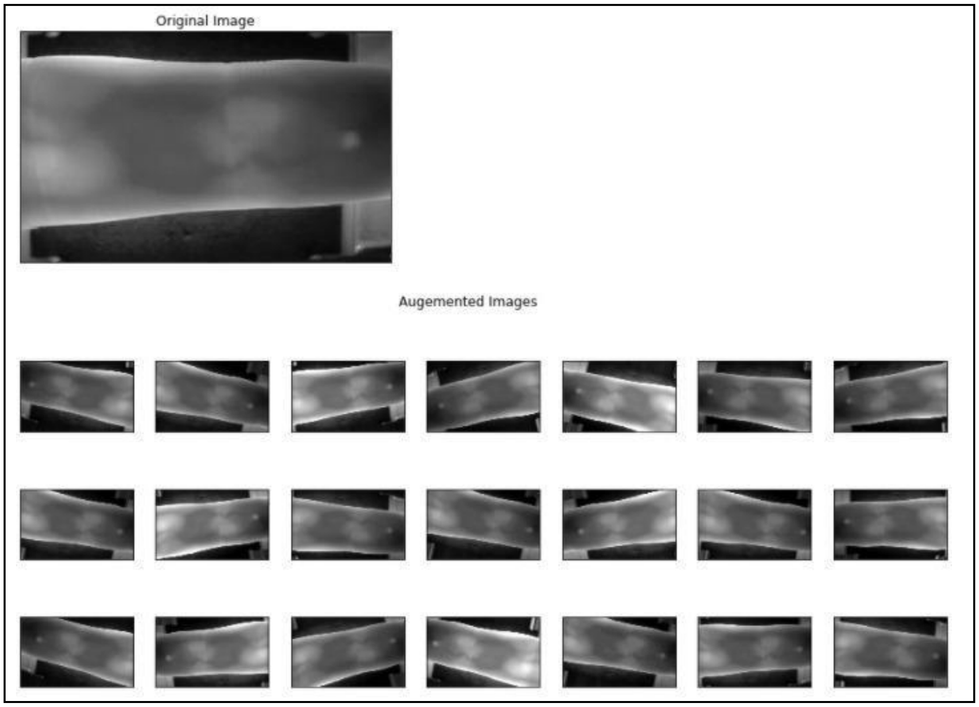
A.1.4 Applying binary erosion



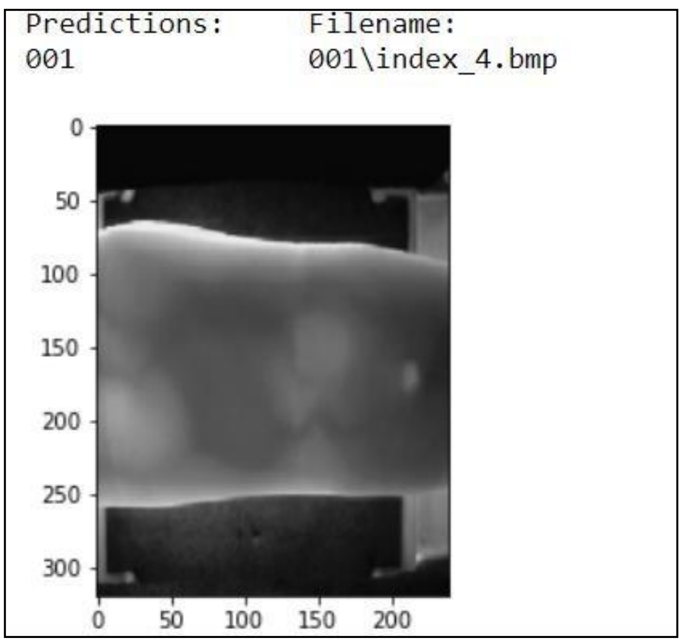
A.1.5 Sobel filter



A.1.6 Contouring filter



A.1.7 Augmentation of finger vein image



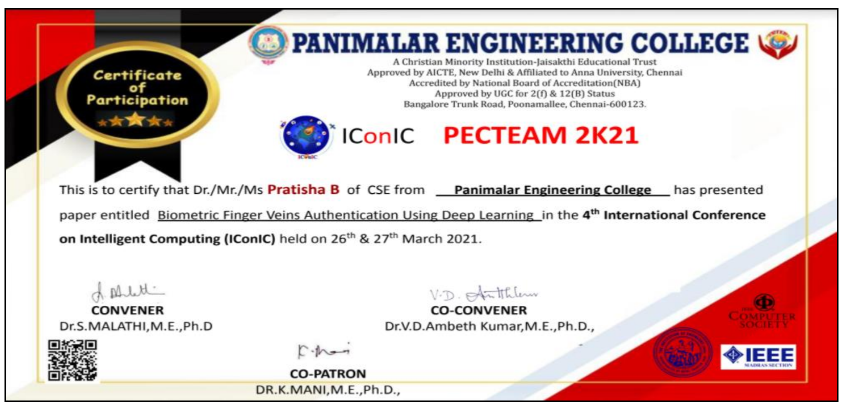
A.1.8 Prediction after algorithm is processed

A.2 Publications:

Journal Name: International Journal of Emerging Technologies and Innovative Research

Paper Title: Biometric Finger Veins Authentication using Deep Learning Algorithm

Published In: Volume 8 | Issue 4 | Year April-2021



Biometric Finger Veins Authentication using Deep Learning Algorithm

Dr.L.JabaSheela[1],V. Sathya Preiya[2], Nivedhitaa Ravichandran[3], Pratisha B[3], Preethi M[3]
Professor[1],Associate Professor[2], Student[3]
Department of Computer Science Engineering,
Panimalar Engineering College, Chennai,Tamilnadu, India.

Abstract— **Finger vein verification can be the most advanced biometric technology these days for safety and ease of use, as it incorporates features within the human body. The fingerprint image captured by the web camera under IR light transmission contains not only the vein pattern itself, but also the shadow produced by the various thicknesses of the finger muscles, bones, and the network networks around the vein. In this paper, we present the first process of enhancing a low-resolution image quality with the effect of light and sound produced by a webcam, and then separate the vein pattern using the corresponding adhesive method and compare it using advanced template matching. The test result shows that even image quality is not good, as long as our arteries are clear and the correct procedure can still be used as personal diagnostic methods. This paper provides detailed reviews of finger recognition algorithms. Such tools include image detection, pre-processing, feature removal and similar methods for extracting and analyzing object patterns. In addition, we document some of the novel's findings after a critical analysis of the comparative strategies of the proposed strategy.**

Keywords—Human interaction, Finger veins, Deep Learning

I. INTRODUCTION

With our progress toward a global knowledge society, the life of the average person is at the same time threatened by crime that can occur anywhere in the world. Terrorism can spread around the globe in an instant, escalating and intensifying the threat. Thus, biometrics systems, which are more precise and use part of the human body, have become the ideal response to these high safety requirements and are already universally accepted. Researchers have determined that the vascular pattern of a person's body is different from that of the individual, difficult to form, in contact with, unaffected by color and skin color, and does not change with age.

The goal of vein vein detection is that infrared light at a wavelength of 700nm - 1000nm can pass through most human tissues while hemoglobin in the blood can completely absorb infrared light. We align Near-Infrared (NIR) to the Far Infrared (FIR) method, as it can capture larger vein patterns more effectively. We use a light transfer process to capture the vein pattern. We place the finger under IR light, so that the vein pattern is captured as a shadow pattern with a web camera under the finger. A diagram of the completed authentication process can be seen in Figure 2. Vascular pattern identification is a relatively new research topic in biometric technology and all published articles have used sufficient space to prove that human arteries can be used for human self-identification purposes. Our aids in personal biometric recognition. In this paper, we studied the identification of arteries in the test and found that there were several difficulties: a. Photo captured a typical web camera contains the sound of salt and pepper and the distribution of gray matter between different experiments is not the same, because the web camera always does light adjustment.

b. Under normal circumstances, the vein size of the vein image of the vein is very small. We need a good boundary breakdown to get a functional binary image that provides enough detail of finger veins. c. The pressure applied to our finger will cause the inner vein to shrink or change. Therefore, we need to create a "weak" finger border to allow the user's finger to be in a "relaxed" position.

II. RELATED WORK

These biometric detection methods can be divided into two categories:

1. Extrinsic biometric features (palm print, iris, fingerprint, face)
2. Intrinsic biometric features (palm vein, handvein, and finger vein)

Existing Approaches

- vessel extraction
- subspace-learning-based approaches
- statistical-based techniques
- local-invariant based methods

Fingerprint recognition has received a lot of attention recently and is considered a promising biometric practice. In the accompanying modes, vein pattern patterns test the internal finger vein recognition, but their performance remains unsatisfactory due to flawed vein networks and weak similarities. One important reason may be the neglect of a thorough analysis of the structure of the vein anatomy. By carefully examining the anatomy structure and imaging image of vein patterns, this paper proposes a novel vein recognition framework, including an anatomic-based vein extraction (ASAVE) algorithm and a concatenated integration strategy. Specifically, the vein pattern is extracted by a curve directed at the map directed at the base of the vertical profile cut across the cross. In addition, the extracted vein pattern was further reduced and refined to obtain a reliable vein network. In addition to the vein network, the clear vein branches in the image were excavated from the vein pattern, called the vein spine.

In analogy, a vein vein is used in vein network measurements to overcome finger migration. The similarity of the measured vein networks is measured by the uniformity of the vein stretching and recurring continuously by combining the degree of fragmentation of the corresponding vein.

The vein pattern, originating in the fingertip, is continuous and interconnected in growth to the finger foot. In other words, in each branch and between different branches, the vein pattern is unbroken.

The outer diameter of the vein pattern varies, and gradually increases from the tip of the finger to the root of the finger and from the lower branch to the main branch. The difference between the outer diameters of the branches of the blood vessels near the near part and the middle part of the finger is found Each branch in the pattern of the pattern turns smooth and firm. There is no sudden or sharp difference in vein width and no burr or hole in the vein pattern.

In a fingerprint recognition template, an investigation will be accepted if the number of its vein points scattered by a registered user is greater than the previously defined limit. However, acceptance can be false due to ignoring the formation of the vein pattern. We find that the local vein branches near the bifurcation point of the vein pattern differ significantly between the fake images. Therefore, this paper attempts to explore this type of local vascular structure to improve the visual performance of template simulations. The bifurcation point and its local vein branches, named after the vein-branch structure, are extracted from the vein pattern, and integrated with the whole vein pattern in a user-based filtering framework. Two public knowledge test results confirm the effectiveness of the proposed framework for improving the performance of vein pattern- b a s e d finger vein recognition.

The tri-branch vein architecture is used at the first level of the framework to screen fraudsters and provide candidates with a choice of investigations. Affected by image quality, the similarities between the tri-branch vein class structures differ significantly from different users.

For one user, the corresponding ratio of tri-branch vein frames from two real images is 0.6, but points are about 0.3 for one user. Therefore, user-defined thresholds, not a single standard limit, are used for filtering. In the second level of the framework, the whole vein pattern is used to match the investigation with the candidates, given the first level, and the visual effect will be restored.

Reasons to use the arterial structure of a tri-branch branch, instead of the whole vein pattern, at the first level of the framework twice. Obviously, a three-branch vein structure costs less time and space costs than the entire vein pattern alike. It follows vein patterns several times. The tracking function detects the local black line (finger vein pattern) and then pixels the pixel with black lines. If a black line is not found, a new tracking point starts randomly in another position.

Manually set parameter

- Cannot perform well on low-quality images
- Ambient lighting conditions tough to identify
- Cannot recognize the images from poorly designed image capturing devices.

III. Proposed System

Fingerprint recognition has emerged as a strong biometric mode due to their unique vein pattern that can be captured using the infrared spectrum. Large finger-based solutions using biometric require the need to search for a probe vein sample against a large collection of gallery samples. To improve credibility in finding the right identity in a large database of fingerprints, it is important to introduce a vein identification system with finger and retrieval.

This paper proposes a biometric system for identifying humans based on a finger vein pattern. The system uses a database of human fingerprint images obtained at infrared range. The current proposal used a Sobel detector, an upgrade filter and a capture process to obtain a vein pattern.

The proposed system is implemented using the novel fingerprint recognition algorithm. Dimension and gabor filter are algorithms that are used for element extraction and are used to separate the distance to which the extracted element is performed.

The Deep Neural Network is used to recognize the Finger Vein more precisely than any other algorithms.

Advantages

- Supports how to integrate the results of model components
- Filters are automatically adjusted to extract the most useful information
- Measurement ensures that the output histogram is smooth
- Improved by the histogram measurement process
- Stages can be used in a picture.

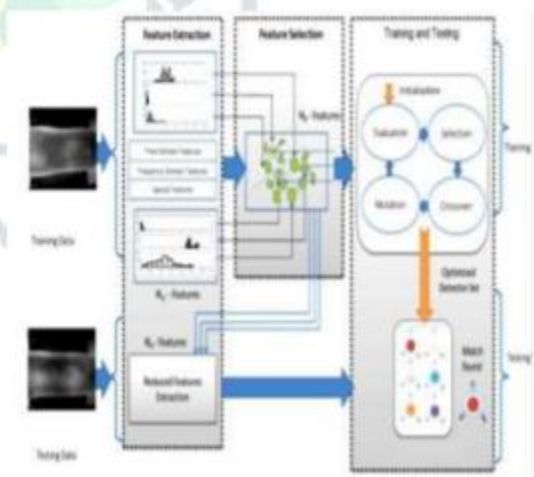


Fig 1. System Architecture

A. Exploratory Data Analysis

Analysis of test data often falls into two ways. First of all, each method is not a click or click. And secondly, each method is not divisive or multivariate (usually bivariate). Non-graphical methods often involve summative calculations, while graphical methods obviously summarize data in a logical or figurative way.

Different methods look at one variable (data column) at a time, while multivariate methods look at two or more variables simultaneously to test the relationship. Normally our multivariate EDA will be bivariate (looking at two variables), but sometimes it will involve three or more variables. It is almost always a good idea to make an unapproved EDA for each multivariate EDA item before making a multivariate EDA.

Data visualization is a method that uses multiple combinations of static views and interactions within a specific context to help people understand and make sense of big data. Details are often displayed in a media format that reflects patterns, styles and combinations that may not be obvious.

B. Dataset Processing

The File Handling Package is one of the most complete python progress bar packages and is useful in those situations you want to create your own scripts that keep users informed of the status of your application. The package works on any platform (Linux, Windows, Mac, FreeBSD, NetBSD, Solaris / SunOS) on any console or GUI, and is friendly to IPththon / Jupyter..

C. Feature Selection:

The number of pixels in an image is the same as the size of a gray image that we can obtain pixel features by resizing the image and restoring the mage form. The edges of the picture are at the corners there the pixel is very flexible, as the images are stored in the same format we can see at different values and see that the change in pixel value is high but manually doing it takes time.

D. Prediction

The ImageDataGenerator section allows for up to 90 degrees rotation, horizontal flip, flexibility and vertical positioning. We need to use training reinforcement over a test set. ImageDataGenerator will generate a stream of unpopular images during training.

We will describe the operational functions of the Exponential Linear Unit (ELU) One layer that is fully integrated after the last major integration. Parameter = 'same' parameter. This simply means that the output volume pieces will be the same size as the input.

Batch configuration provides a way to use data processing, such as a standard school, for hidden network layers. It typically produces the hidden layer effects of each subgroup (hence the name) in a way, which keeps its activation value close to 0, and its standard deviation is closer to 1. layers. Networks have a fast stop train and can use high levels of learning.

IV. EXPERIMENTAL RESULTS

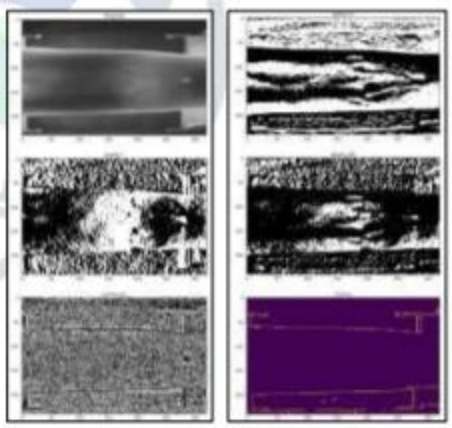


Fig 2. Preprocessed Image

V CONCLUSION

The proposed method is suitable for measurement in the first phase and is used in large batches. It can be expected that the system can be accessed in a timely manner with our measurement rather than ordinal measurements. This paper presented an analysis of the anatomy structure in the removal of the finger network and simulation and proposed an effective framework for finger recognition. Contributions for this paper can be summarized as follows. An algorithm has been used in the extraction of the vein pattern, including the curvature method directed at the shape map and the anatomy structure based on the vein network refinement.

REFERENCES

[1] *D.* De Santos-Sierra, M. F. Arriaga-Gómez, G. Bailador, And C. Sánchez-Ávila, "Low Computational Cost Multilayer Graph-Based Segmentation Algorithms For Hand Recognition On Mobile Phones," In Proc. Int. Carnahan Conf. Security Technol. (Iccst), Rome, Italy, 2014, Pp. 1–5.

[2] W. Kang And Q. Wu, "Pose-Invariant Hand Shape Recognition Based On Finger Geometry," Ieee Trans. Syst., Man, Cybern., Syst., Vol. 44, No. 11, Pp. 1510–1521, Nov. 2014.

[3] B. P. Nguyen, W.-L. Tay, And C.-K. Chui, "Robust Biometric Recognition From Palm Depth Images For Gloved Hands," Ieee Trans. Human–Mach. Syst., Vol. 45, No. 6, Pp. 799–804, Dec. 2015.Morales Et Al., "Synthesis Of Large Scale Hand-Shape Databases For Biometric Applications," Pattern Recognit. Lett., Vol. 68 , No. 1, Pp. 183–189, 2015.

[4] R. M. Luque-Baena, D. Elizondo, E. López-Rubio, E. J. Palomo, And T. Watson, "Assessment Of Geometric Features For Individual Identification And Verification In Biometric Hand Systems," Expert Syst. Appl., Vol. 40, No. 9, Pp. 3580–3594, 2013.S. Marcel, M. S. Nixon, And S. Z. Li, Handbook Of Biometric Anti-Spoofing: Springer, 2014.

[5] D. Gragnaniello, C. Sansone, And L. Verdoliva, "Iris Liveness Detection For Mobile Devices Based On Local Descriptors," Pattern Recognition Letters, Vol. 57, Pp. 81-87, 2015.L. Yang, G. Yang, Y. Yin, And X. Xi, "Finger Vein Recognition With Anatomy Structure Analysis," Ieee Trans. Circuits Syst. Video. Technol., 2017.

[6] L. Yang, G. Yang, L. Zhou, And Y. Yin, "Superpixel Based Finger Vein Roi Extraction With Sensor Interoperability," In Proc. 8th Int. Conf. Biometrics (Icb), Phuket, May. 2015, Pp. 444–451.

[7] L. Yang, G. Yang, Y. Yin, And R. Xiao, "Sliding Window-Based Region Of Interest Extraction For Finger Vein Images," Sensors, Vol. 13, No. 3, Pp.3799–3815, 2013.

[8] Nalini, K., and L. Jaba Sheela. "A survey on datamining in cyber bullying." International Journal on Recent and Innovation Trends in Computing and Communication 2.7 (2014): 1865-1869.

[9] Sheela, L. Jaba, and V. Shanthi. "DIMAR-Discovering interesting medical association rules form MRI scans." 2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. Vol. 2. IEEE, 2009.

[10] K. B. Raja, R. Raghavendra, And C. Busch, "Video Presentation Attack Detection In Visible Spectrum Iris Recognition Using Magnified Phase Information," Ieee Transactions On Information Forensics And Security, Vol. 10, No. 10, Pp. 2048-2056, 2015.

REFERENCES

[1] D. De Santos-Sierra, M. F. Arriaga-Gómez, G. Bailador, And C. Sánchez-Ávila, "Low Computational Cost Multilayer Graph- Based Segmentation Algorithms For Hand Recognition On Mobile Phones," In Proc. Int. Carnahan Conf. Security Technol. (Iccst), Rome, Italy, 2014, Pp. 1–5.

[2] W. Kang And Q. Wu, "Pose-Invariant Hand Shape Recognition Based On Finger Geometry," Ieee Trans. Syst., Man, Cybern., Syst., Vol. 44, No. 11, Pp. 1510–1521, Nov. 2014.

[3] B. P. Nguyen, W.-L. Tay, And C.-K. Chui, "Robust Biometric Recognition From Palm Depth Images For Gloved Hands," Ieee Trans. Human–Mach. Syst., Vol. 45, No. 6, Pp. 799–804, Dec. 2015.Morales Et Al., "Synthesis Of Large Scale Hand-Shape Databases For Biometric Applications," Pattern Recognit. Lett., Vol. 68 , No. 1, Pp. 183–189, 2015.

[4] R. M. Luque-Baena, D. Elizondo, E. López- Rubio, E. J. Palomo, And T. Watson, "Assessment Of Geometric Features For Individual Identification And Verification In Biometric Hand Systems," Expert Syst. Appl., Vol. 40, No. 9, Pp. 3580–3594, 2013.S. Marcel, M. S. Nixon, And S. Z. Li, Handbook of Biometric Anti-Spoofing: Springer, 2014.

[5] D. Gragnaniello, C. Sansone, And L. Verdoliva, "Iris Liveness Detection For Mobile Devices Based On Local Descriptors," Pattern Recognition Letters, Vol. 57, Pp. 81-87, 2015.L. Yang, G. Yang,

[6] Y. Yin, And X. Xi, "Finger Vein Recognition With Anatomy Structure Analysis," Ieee Trans. Circuits Syst. Video. Technol., 2017.

[7] L. Yang, G. Yang, L. Zhou, And Y. Yin, “Superpixel Based Finger Vein Roi Extraction With Sensor Interoperability,” In Proc. 8th Int. Conf. Biometrics (Icb), Phuket, May. 2015, Pp. 444–451.

[8] L. Yang, G. Yang, Y. Yin, And R. Xiao, “Sliding Window-Based Region OfInterest Extraction For Finger Vein Images,” Sensors, Vol. 13, No. 3, Pp.3799–3815, 2013.

[9] Nalini, K., and L. Jaba Sheela. "A survey on datamining in cyber bullying.” International Journal on Recent and Innovation Trends in Computing and Communication 2.7 (2014): 1865-1869.

[10] Sheela, L. Jaba, and V. Shanthi. "DIMAR- Discovering interesting medical association rules form MRI scans." 2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. Vol. 2. IEEE, 2009.

[11] K. B. Raja, R. Raghavendra, And C. Busch, “Video Presentation Attack Detection In Visible Spectrum Iris Recognition Using Magnified Phase Information,” Ieee Transactions On Information Forensics And Security, Vol.10, No. 10, Pp. 2048-2056, 2015.