

DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068



**Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING**

Major Project Phase-II Report

**(LIVENESS DETECTION OF FINGERPRINT USING SVM AND
CNN)**

By

Nikitha D Reddy- ENG18CS0193

Sreesh Surendran- ENG18CS0281

Tejaswini R- ENG18CS0298

Under the supervision of

**Asst. Prof. Anusha Ashok
Professor, Department of CSE**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,
BANGALORE**

(2021-2022)



DAYANANDA SAGAR UNIVERSITY

School of Engineering

Department of Computer Science & Engineering

Kudlu Gate, Bangalore – 560068

Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled “**LIVENESS DETECTION OF FINGERPRINTS USING SVM AND CNN**” is carried out by **NIKITHA D REDDY (ENG18CS0193)**, **SREESH SURENDRAN (ENG18CS0281)**, and **TEJASWINI R (ENG18CS0298)**, bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2021-2022**.

Prof. Anusha Ashok	Dr. Girisha G S	Dr. A Srinivas
Assistant Professor Dept. of CSE, School of Engineering Dayananda Sagar University	Chairman CSE School of Engineering Dayananda Sagar University	Dean School of Engineering Dayananda Sagar University
Date:	Date:	Date:

Name of the Examiner

Signature of Examiner

1.

2.

DECLARATION

We, **NIKITHA D REDDY (ENG18CS0193), SREESH SURENDRAN (ENG18CS0281), TEJASWINI R (ENG18CS0298)**, are students of the eighth semester B.Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the phase-II project titled “**Liveness Detection of fingerprints using SVM and CNN**” has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2021-2022**.

Student	Signature
Name: NIKITHA D REDDY USN: ENG18CS0193	
Name: SREESH SURENDRAN USN: ENG18CS0281	
Name: TEJASWINI R. USN: ENG18CS0298	
Place: BENGALURU Date:	

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. A Srinivas. Dean, School of Engineering & Technology, Dayananda Sagar University** for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman, Computer Science, and Engineering, Dayananda Sagar University**, for providing the right academic guidance that made our task possible.

We would like to thank our guide **Prof. Anusha Ashok, Assistant Professor Dept. of Computer Science and Engineering, Dayananda Sagar University**, for sparing her valuable time to extend help in every step of our project work, which paved the way for smooth progress and the fruitful culmination of the project.

We would like to thank our **Project Coordinators, Dr. Meenakshi Malhotra and Dr. Bharanidharan** and all the staff members of Computer Science and Engineering for their support. We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

	Page
LIST OF ABBREVIATIONS	vi
LIST OF FIGURES	vii
ABSTRACT.....	viii
CHAPTER 1 INTRODUCTION.....	1
1.1. DESCRIPTION	1
1.2. FIGURES AND FLOWCHARTS.....	2
1.3 SCOPE.....	3
CHAPTER 2 PROBLEM DEFINITION	4
2.1 PURPOSE.....	4
2.2 INTENDED AUDIENCE.....	4
CHAPTER 3 LITERATURE SURVEY.....	5
CHAPTER 4 PROJECT DESCRIPTION.....	9
4.1. PROPOSED DESIGN	10
CHAPTER 5 REQUIREMENTS.....	11
5.1. FUNCTIONAL REQUIREMENTS	11
5.2. EXTERNAL INTERFACE REQUIREMENTS.....	11
CHAPTER 6 METHODOLOGY.....	12
6.1. PROPOSED SETUP.....	13
6.2. ALGORITHMS EMPLOYED.....	14
CHAPTER 7 EXPERIMENTATION.....	15
CHAPTER 8 TESTING AND RESULTS	16
8.1. BACKEND	16
8.2. FRONTEND	19
CHAPTER 9 CONCLUSION.....	23
REFERENCES... ..	24

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
DL	Deep Learning
GUI	Graphical User Interface
SVM	Support vector machine
CNN	Convolutional Neural Networks

LIST OF FIGURES

Fig. No.	Description of the figure	Page No.
1.2.1	Phase-I Architecture	2
1.2.2	Phase-II Architecture	3
6.1.1	Proposed Setup	13
8.1.1	Loading the dataset and required libraries	16
8.1.2	Cleaning the data using Otsu's Thresholding	16
8.1.3	Sweat Pore Extraction	13
8.1.4	Showing the feature extraction output	17
8.1.5	Results of SVM model	18
8.1.6	Accuracy graph of both SVM and CNN	18
8.2.1	Home Page	19
8.2.2	Login Page	19
8.2.3	User Registration Page	20
8.2.4	To upload the fingerprint images	20
8.2.5	Pre-loaded fingerprint images stored in the database	21
8.2.6	Detection of fake fingerprint	21
8.2.7	Prediction of Live Fingerprint	22

ABSTRACT

The ability of a system to determine whether a fingerprint, face, or other biometric is real (from a living person present at the point of capture) or fake in biometrics is known as liveness detection (from a spoof artefact or lifeless body part). The most used biometric recognition technique is fingerprinting. As a result, safeguarding the system from spoof assaults becomes quite difficult. The use of a fake fingerprint, however, is frequently used in spoofing attacks. Spoof fingerprint detection has become more crucial recently with the expansion of biometric authentication systems. In this research, we employ CNN and SVM to determine whether a fingerprint is live.

CHAPTER 1: INTRODUCTION

"The identification of an individual based on biological attributes, such as fingerprints, iris patterns, and facial features," according to the definition of biometrics (also known as biometry). Biometric systems are emerging technologies that enable an individual's verification based on physiological or behavioural traits, such as identifying faces, fingerprints, irises, hand geometry, palms, voices, gait, and handwriting signatures, among other things. Fingerprint recognition is the most popular and effective of these biometric identifiers. However, fingerprint scanners' security has been questioned.

Previous research has demonstrated that a well-duplicated synthetic finger made of silicone rubber, Play-Doh, wax, clay, gelatine, or, in the worst scenario, dismembered fingers can trick a number of fingerprint scanners. These compounds are moisture-based and can be imaged by most fingerprint scanners. From the standpoint of security and accountability, it is critical that a biometric system be able to detect forged biometric samples. Anti-spoofing techniques are primarily categorised into two groups:

- Detection of fingerprint spoofing using hardware
- Detection of fingerprint spoofing using software

Liveness detection is an anti-spoofing technique that ensures that only "genuine" fingerprints may generate enrolment, verification, and identity templates.

1.1. DESCRIPTION

Biometric systems are becoming increasingly popular. Fingerprint scanning is the most widely used biometric identification method available today. However, the security of fingerprint scanners has been questioned, and prior research have shown that counterfeit fingerprints, or replicas of real fingerprints, can trick scanners. The state of fingerprint systems is changing, and this study will look at the current condition.

1.2. FIGURES AND FLOWCHARTS

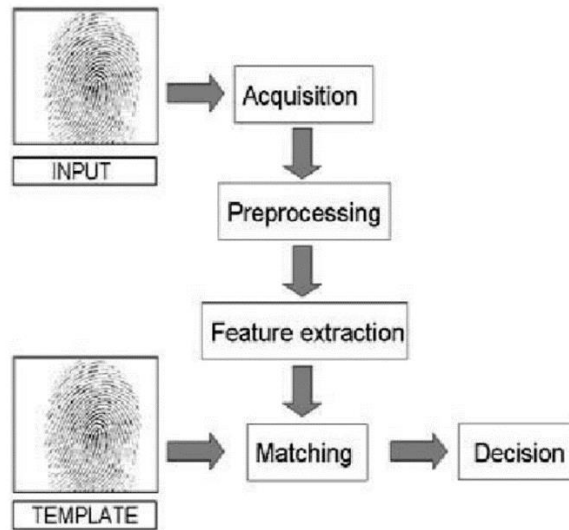


Fig 1.2.1: Phase – I Architecture

The proposed methodology is broken down into two phases: testing and training. The training phase comprises of two sets of sampled photos that have been pre-processed to improve image quality. The feature Extraction Algorithm is used to extract enhanced image features.

Finally, these features are chained (combined) and trained using a Support Vector Machine (SVM) before being saved in the Knowledge Base. Similarly, the input images are pre-processed during the testing phase to improve image quality. The improved image is given through a multi-feature extractor, which extracts the necessary feature from it. To categorise the final class as Live or Spoof, enhancing image features are retrieved using the Multi feature Extraction Algorithm and classified using the Support Vector Machine (SVM).

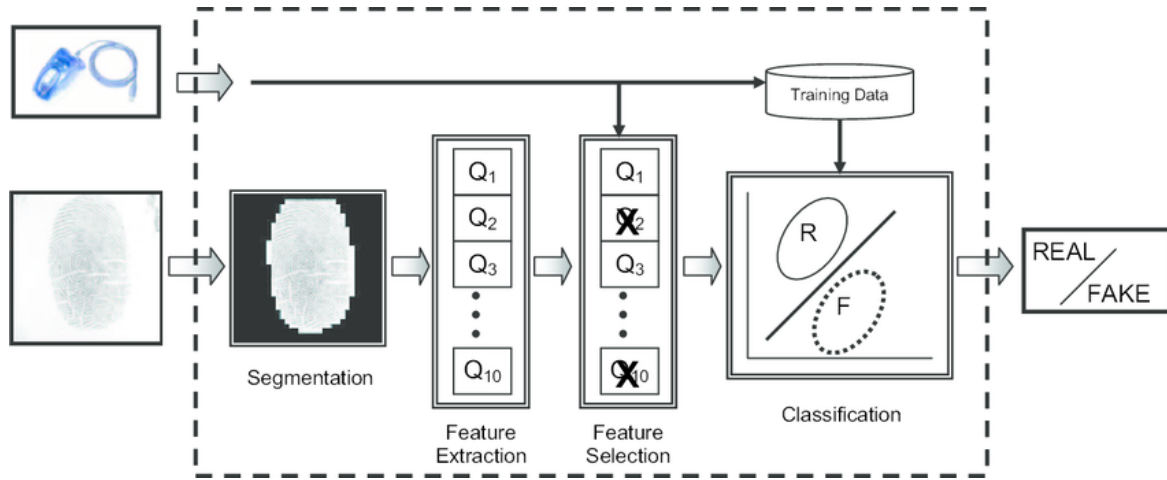


Fig 1.2.2: Phase – II Architecture

These features are chained (combined) and trained using Convolutional Neural Networks (CNN), which are then stored in a Knowledge Base. Similarly, during the testing phase, the input photographs are pre-processed to improve image quality. A multi-feature extractor extracts the required feature from the improved image and provides it. Image attributes are extracted using the Feature Extraction Algorithm and classified using Convolutional Neural Networks to categorise the final class as Live or Spoof (CNN).

1.3. SCOPE

Many biometric services are being developed and tested right now. However, these biometric technologies will be widely used in a few years. Plastic cards will soon fade into the background, and fingerprint scans will become a commonplace procedure. Liveness To protect biometric authentication systems from fraud, fingerprint detection is required. A fraudster could, for example, employ a false fingerprint to gain unauthorised access to accounts or data. Thus, for a safe facial authentication application, liveness detection is necessary for fraud prevention.

CHAPTER 2: PROBLEM DEFINITION

Because of its uniqueness and consistency over time, fingerprint recognition is one of the most widely utilised biometrics technologies for identification and authentication systems. Despite its widespread use in a wide range of large-scale and diverse person identification systems, the system has significant obstacles, particularly in single mode representation biometric systems where matching process mistakes occur owing to distortions and noisy data.

As a result, the system's accuracy has suffered a severe decline. In unimodal biometric systems, researchers at various levels have presented many methodologies and algorithms; yet, the system's efficiency and precision remain a major difficulty. Because biometric systems are resource expensive in terms of processing speed and accuracy, an effective solution for fingerprint recognition system performance is required.

2.1. PURPOSE

The goal of the study is to detect the liveness of fingerprints and classify them as real or phoney. Liveness To protect biometric authentication systems from fraud, fingerprint detection is required. (Avoid forgeries.) A fraudster could, for example, employ a false fingerprint to gain unauthorised access to accounts or data. Thus, for a safe facial authentication application, liveness detection is necessary for fraud prevention.

2.2. INTENDED AUDIENCE

This project is aimed towards the following individuals:

- Fingerprint recognition system manufacturers.
- Businesses are considering implementing a fingerprint recognition system.
- People who use fingerprint recognition software.
- Researchers interested in continuing their work in the field of fingerprint recognition systems, particularly in the areas of liveness detection and artificial fingerprint assaults.
- Computer science, information technology, and other related students with an interest in security, particularly biometrics.

CHAPTER 3: LITERATURE SURVEY

SL. NO:	Paper name:	Description	Inference
1.	<p>Evaluation of Fingerprint Liveness Detection by Machine Learning Approach - A Systematic View</p> <p>Published on: Jan, 2021</p> <p>By Dr. Edriss Eisa Babikir Adam, and Prof. Sathesh,</p>	<p>The software requirements for the fingerprint analysis is based on Image quality, perspiration, fusion, and skin deformation</p>	<p>Inferred the ideas used to acquire the sign of live moment such as Use of trained software processing algorithms.</p> <p>Use of previous trained or captured of physique moment</p>
2.	<p>Fingerprint liveness detection using local quality features</p> <p>Published on: June 8, 2018.</p> <p>Authors: Ram Prakash Sharma, Somnath Dey</p>	<p>The proposed system extracts 8 sensors independent quality features on a local level containing minute details of the ridge-valley structure of real and fake fingerprints. These local quality features constitute a 13-dimensional feature vector. The system is tested on a publicly available dataset of LivDet 2009 competition.</p>	<p>The experimental results exhibit supremacy of the proposed method over current state-of-the-art approaches providing least average classification error of 5.3% for LivDet 2009. Additionally, effectiveness of the best performing features over LivDet 2009 is evaluated on the latest LivDet 2015 dataset which contain fingerprints fabricated using unknown spoof materials. An average classification error rate of 4.22% is achieved in comparison with 4.49% obtained by the LivDet 2015 winner. Further, the proposed system utilizes a single fingerprint image, which results in faster implications and makes it more user-friendly.</p>

3.	<p>Fingerprint Liveness Detection with Feature Level Fusion Techniques using SVM and Deep Neural Network</p> <p>Published in: 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)</p>	<p>Experiment is done with Support Vector Machine (SVM) and Deep Neural Network (DNN) classifiers over the LivDet 2013 database.</p>	<p>It is found that the classification accuracy of the proposed detection technique is higher for DNN as compared to SVM classifier for different feature level fusion techniques.</p>
4.	<p>Protecting Against Fingerprint Spoofing in Mobile Devices</p> <p>Synaptics 2016</p>	<p>This paper discusses ways to protect against fingerprint spoofing in mobile devices and provides an overview of different ways biometric authentication can be hacked, as well as how anti-spoofing technologies can help defend against such threats.</p>	<p>We focus on anti-spoofing technology to detect when a fake finger is being used. Because fingerprints are the most common form of biometric authentication on mobile devices today, there are two aspects of fingerprint spoofing: the techniques hackers use to spoof fingerprints; and the ways device manufacturers can defend against these spoofs.</p>

5.	<p>Fingerprint Spoof Detection Using Quality Features</p> <p>Published on: 31st October, 2015.</p> <p>By: Arunalatha G, M. & Ezhilarasan Research Scholar, Department of Computer Science and Engineering Professor, Department of Information Technology Pondicherry Engineering College, Puducherry</p>	<p>The fingerprint spoof detection is performed by measuring the following quality features of fingerprint. They are Spatial Coherence, Clustering Factor, Gabor Features, Uniformity of Frequency field, Ridge frequency, Direction map and Contrast map. This approach is based on fingerprint image quality.</p>	<p>This technique is software based as it requires no external hardware. This approach is inexpensive.</p>
6.	<p>Using convolutional Neural networks for image recognition</p> <p>By Samer Hijazi, Rishi Kumar, and Chris Rowen, IP Group, Cadence</p>	<p>This white paper covers the basics of CNNs including a description of the various layers used. Using traffic sign recognition as an example, we discuss the challenges of the general problem and introduce algorithms and implementation software developed by Cadence that can trade off computational burden and energy for a modest degradation in sign recognition rates</p>	<p>In our paper we propose a fingerprint liveness-detection method based on convolutional neural network (CNN) features extracted from fingerprint patches. Firstly, fingerprints are segmented, and then data augmentation is performed to increase the size of training data. Secondly, on the augmented fingerprint, locations of patches are determined through normal distributions of segmented areas of the fingerprint image.</p>

7.	<p>Liveness Detection for Fingerprint Scanners Based on the Statistics of Wavelet Signal Processing</p> <p>Published on : 2006</p> <p>By: Bozhao Tan and Stephanie Schuckers Department of Electrical and Computer Engineering, Clarkson University, Potsdam, NY 13699</p>	<p>Statistical features are extracted for multiresolution scales to discriminate between live and non-live fingers. Based on these features, we use a classification tree to generate the decision rules for the liveness classification. We test this method on the dataset which contains about 58 live, 80 spoof (50 made from Play-Doh and 30 made from gelatin), and 25 cadaver subjects for 3 different scanners. Also, we test this method on a second dataset which contains 33 live and 33 spoof (made from gelatin) subjects.</p>	<p>Results show that it is possible for the capacitive DC and optical scanners to detect vitality using a single fingerprint based on the perspiration pattern specific to the live fingers. The method is purely software based and application of this liveness detection method can protect fingerprint scanners from spoof attacks</p>
----	--	---	--

CHAPTER 4: PROJECT DESCRIPTION

Because of its permanence and uniqueness, fingerprint recognition is the most extensively used biometric technology for personal identification systems. However, certain forms of attacks can compromise biometric systems. Spoofing is the deception of a biometric system by an unauthorized individual using a phone input that mimics one of the authorized person's biometric inputs. Spoof detection adds an additional layer of security to biometrics. The use of a fake replica of a biometric in an attempt to evade a biometric sensor is known as a spoof attack, which is a subset of presentation attacks. Liveness detection, also known as presentation attack detection, distinguishes between real and fake biometric traits. It is based on the idea that additional data can be gathered in addition to the data obtained by a standard authentication system to determine whether a biometric measure is genuine.

It prevents forgeries by extracting the fingerprint's features, which are then classified as live or false using a support vector machine (SVM) and a Convolutional Neural Network (CNN), with the best model chosen.

This liveness detection technology is entirely software-based, and it can defend fingerprint scanners from spoof attacks. We trained SVM and CNN models, calculated their accuracy rates, and then chose the model that predicted or assessed the liveness of fingerprints with the highest accuracy. To identify the liveness of the fingerprints, a webpage is developed, and the most correct model is chosen and passed onto the website.

4.1. PROPOSED DESIGN

To begin, we'll use Kaggle to select an appropriate dataset that includes both spoofed and live fingerprints for preliminary analysis. The Support Vector Machine (SVM) would then be implemented using a smaller dataset for analysis. The SVM Model would next be trained and tested using the earlier dataset obtained from Kaggle. Now, the SVM Model will use some unique properties of fingerprints, such as sweat pores and ridges, to estimate the accuracy and correctness of the fingerprints, and it will anticipate the outcomes accordingly.

Second, the initial implementation of Convolutional Neural Networks (CNN) would be done with a smaller dataset for analysis reasons. The CNN Model would next be trained and tested using the dataset obtained from Kaggle. Now, the CNN Model will estimate the accuracy of fingerprints based on specific unique fingerprint traits such as sweat pores, ridges, and so on, and the results will be anticipated appropriately. The best model with highest accuracy is chosen and a frontend website is developed in order to predict the liveness of the fingerprints.

Then, as the major goal of our project, we will conduct a comparison analysis of both models and select the model that produces the most precise and correct findings.

CHAPTER 5: REQUIREMENTS

The functional requirements are the presence of the application called Jupyter Notebook and the installation of several libraries

5.1. FUNCTIONAL REQUIREMENTS

This implementation should facilitate the users to detect liveness of fingerprints as close to the real-life as possible. This implementation should be able to detect liveness of the fingerprints when used in real-time.

5.2. EXTERNAL INTERFACE REQUIREMENTS

5.2.1. SOFTWARE INTERFACES:

- Operating System: Windows 10 and above
- Domain: Machine Learning
- Programming Language: Python
- Tools: Anaconda, Jupyter Notebook, Google Collab
- Libraries: Pandas, NumPy, Matplotlib, CV2, Keras, SciPy and TensorFlow

5.2.2. HARDWARE INTERFACES:

- Processor: Intel core i3 equivalent and higher
- Memory: 8GB of memory.
- Hard Disk: 1TB of storage.

CHAPTER 6: METHODOLOGY

Proposed methodology is basically divided into two phases testing phase and training phase. Training phase consists of two Set of sampled images, these images are pre-processed to enhance the quality of images. Obtain enhance image features are extracted by using feature Extraction Algorithm (Grey Level Co-occurrence Matrix, Curvelet Transform, Ridge clarity Factor and Ridge Continuity).

Finally, these features are chained (combine) and trained by Support Vector Machine (SVM) and stored in Knowledge base. Similarly at the testing phase the input Images are Pre-processed to enhance the Quality of image. The enhanced image fed as input to multi feature extractor to extract the desired feature of enhanced image. Obtain enhance image features are extracted by using Multi feature Extraction Algorithm (Grey Level Co-occurrence Matrix, Curvelet Transform, Ridge clarity Factor and Ridge Continuity) and classified using Support Vector Machine (SVM) to classify the final class as Live or Spoof.

Convolutional Neural Networks (CNN) are used to chain (combine) and train these features, which are then stored in a Knowledge Base. Similarly, the input images are pre-processed during the testing phase to improve image quality. The improved image is given through a multi-feature extractor, which extracts the necessary feature from it. To categorise the final class as Live or Spoof, image characteristics are retrieved using the Feature Extraction Algorithm and classified using Convolutional Neural Networks (CNN).

6.1. PROPOSED SETUP

Figure 6.1.1 shows the proposed system. Testing and training are the two parts of the suggested technique. Two sets of sampled photos have been pre-processed to increase image quality in the training phase. To extract enhanced picture features, the feature Extraction Algorithm is utilised.

Before being saved in the Knowledge Base, these features are chained (combined) and trained using specific algorithms. Similarly, during the testing phase, the input photographs are pre-processed to improve image quality. A multi-feature extractor extracts the required feature from the improved image and provides it. Enhancing picture features are extracted using the Multi feature Extraction Algorithm and categorised using the algorithms above to classify the final class as Live or Spoof.

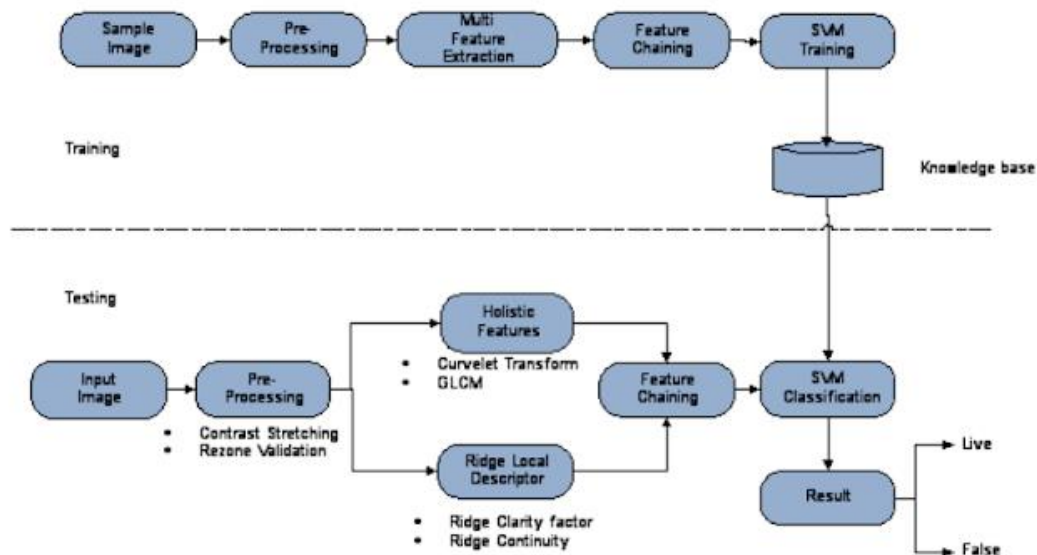


Figure 6.1.1

6.2. ALGORITHMS EMPLOYED

We have used Support Vector Machine (SVM) and Convolutional Neural Networks (CNN) algorithms in Machine Learning to extract the features of the fingerprints in order to classify them into real or fake fingerprints.

- **Support Vector Machine (SVM):** Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for classification problems in Machine Learning.

Equation: The distance of a hyperplane equation: $w^T\Phi(x) + b = 0$ from a given point vector $\Phi(x_0)$ can be easily written as:

$$d_H(\phi(x_0)) = \frac{|w^T(\phi(x_0)) + b|}{\|w\|_2}$$

- **Convolutional Neural Networks (CNN):** Convolutional Neural Networks or CNNs have fundamentally changed our approach towards image recognition as they can detect patterns and make sense of them. They are considered the most effective architecture for image classification, retrieval and detection tasks as the accuracy of their results is very high. They have broad applications in real-world tests, where they produce high-quality results and can do a good job of localizing and identifying where in an image a person/car/bird, etc., are. This aspect has made them the go-to method for predictions involving any image as an input

Equation: The output size O is given by this formula:

$$O = \frac{n - f + 2p}{s} + 1$$

CHAPTER 7: EXPERIMENTATION

In practically every effort or real-world situation, there will be a struggle or setback. To address these issues and continue the search for a solution, the following measures must be followed. During the course of our project, we encountered a few challenges. In our study, we used CNN and SVM to categorise fingerprints as false or authentic. We had trouble acquiring access to the dataset at the beginning because fingerprint files are highly sensitive and confidential, but our guide helped us out and provided us with the dataset to work with.

We started with SVM and discovered that the testing and training accuracy was too low to be useful in the real world, but CNN was the answer because it provided us accuracy of about 95. The model's training took a long time because the dataset was so large as it had both live and fake fingerprints were classified according to the spoofing technique used. In the frontend part, the prediction of the fingerprint is bit slow because of the complexity of the model and huge data being trained to the model, it takes time for the actual testing of the model and predicting the result. Understanding these structures took a long time and was tough.

CHAPTER 8: TESTING AND RESULTS

8.1. BACKEND

```
In [1]: import numpy as np
import glob
import random
import imageio
import PIL, cv2
import pandas as pd
%matplotlib inline
import matplotlib.pyplot as plt
from skimage.morphology import convex_hull_image, erosion
from skimage.morphology import square
import matplotlib.image as mpimg
import skimage
import math
from scipy.ndimage.filters import convolve
from PIL import Image, ImageFilter
from skimage.feature import hessian_matrix, hessian_matrix_eigvals

In [2]: DATA_DIR = "/Users/Admin/OneDrive/Desktop/finger/Training2/Digital_Persona/Live/"
list_dirs = list(glob.glob(DATA_DIR+"*.png"))
num_images = len(list_dirs)
```

Fig 8.1.1: Loading the dataset and required libraries

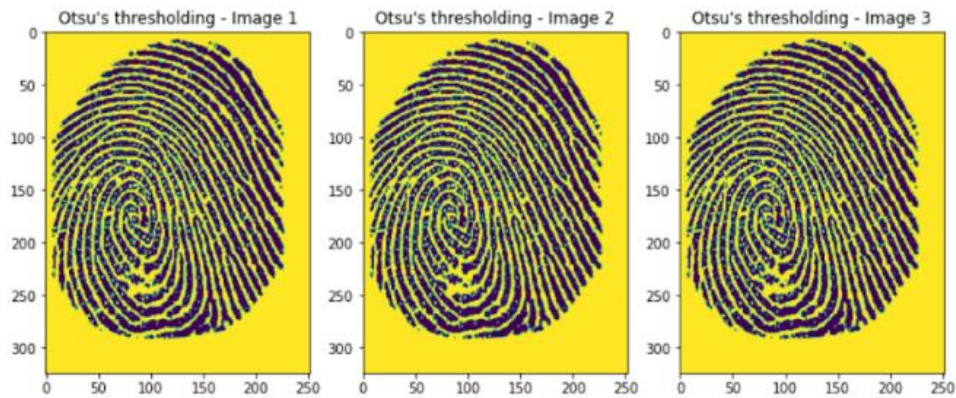


Fig 8.1.2: Cleaning the data using Otsu's Thresholding


```
In [1]: import cv2
import numpy as np
from matplotlib import pyplot as plt
import matplotlib
import scipy.ndimage as ndi
from skimage import transform, io, measure, color, morphology
import os
import cal_TDR_FDR
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False

In [2]: def normalize(im):
    im = im.astype(np.float)
    min_ = np.min(im)
    im = im - min_

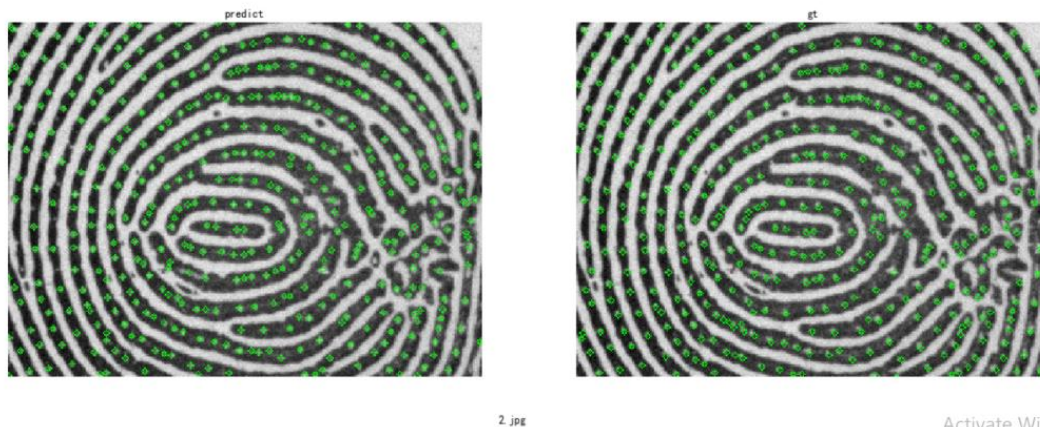
    min_ = 0
    max_ = np.max(im)
    im = im/max_ *255
    return im.astype(np.uint8)

def binarize_otsu(im):
    threshold, img_res = cv2.threshold(im,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
    return threshold, img_res

def plot_gray_img(im,title):
```

Activate Window
Go to Settings to activate

Fig 8.1.3: Sweat Pore Extraction



Activate Window
Go to Settings to activate

Fig 8.1.4: Showing the feature extraction output (Sweat Pores)

Report

```
In [11]: print("Classification report for - \n{}:\n{}\n".format(
          clf, metrics.classification_report(y_test, y_pred)))
```

Classification report for -
GridSearchCV(estimator=SVC(),
param_grid=[{'C': [1, 10, 100, 1000], 'kernel': ['linear']},
{'C': [1, 10, 100, 1000], 'gamma': [0.001, 0.0001],
'kernel': ['rbf']}]):

	precision	recall	f1-score	support
0	0.56	0.50	0.53	64
1	0.49	0.30	0.37	73
2	0.64	0.44	0.53	81
3	0.78	0.95	0.86	311
4	0.69	0.58	0.63	71
accuracy			0.71	600
macro avg	0.63	0.56	0.58	600
weighted avg	0.69	0.71	0.69	600

```
In [13]: import pickle
          filename = 'finalized_model.sav'
          pickle.dump(clf, open(filename, 'wb'))

          # some time later...

          # Load the model from disk
          loaded_model = pickle.load(open(filename, 'rb'))
          result = loaded_model.score(X_test, y_test)
          print(result)

          0.7133333333333334
```

Fig 8.1.5: Results of SVM Model

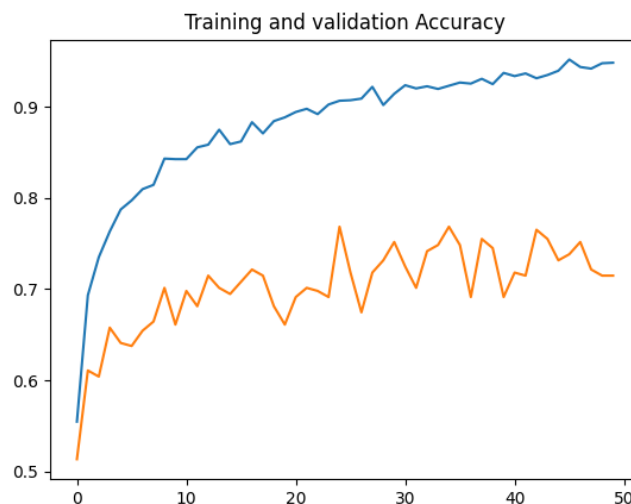


Fig 8.1.6: Accuracy graph of both SVM and CNN

8.2. FRONTEND

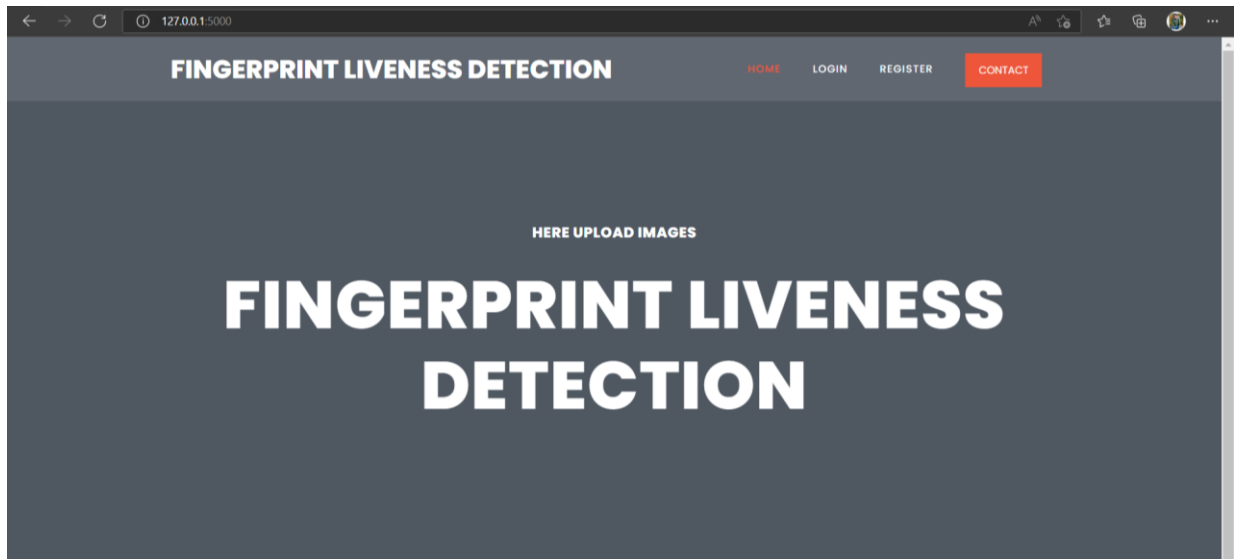


Fig 8.2.1: Home Page

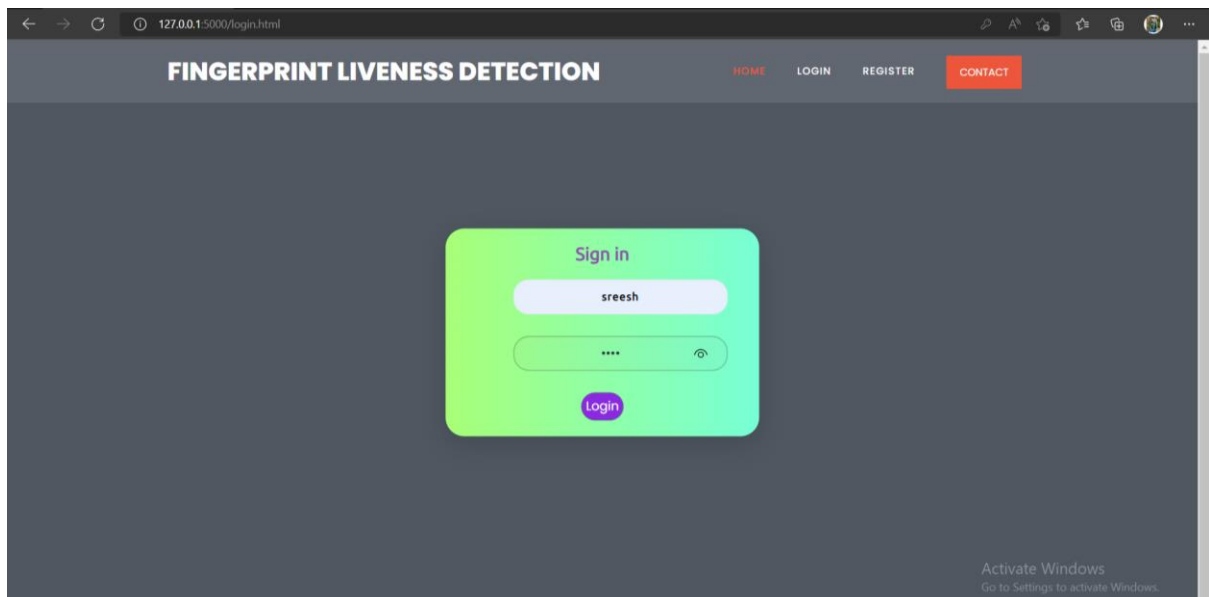
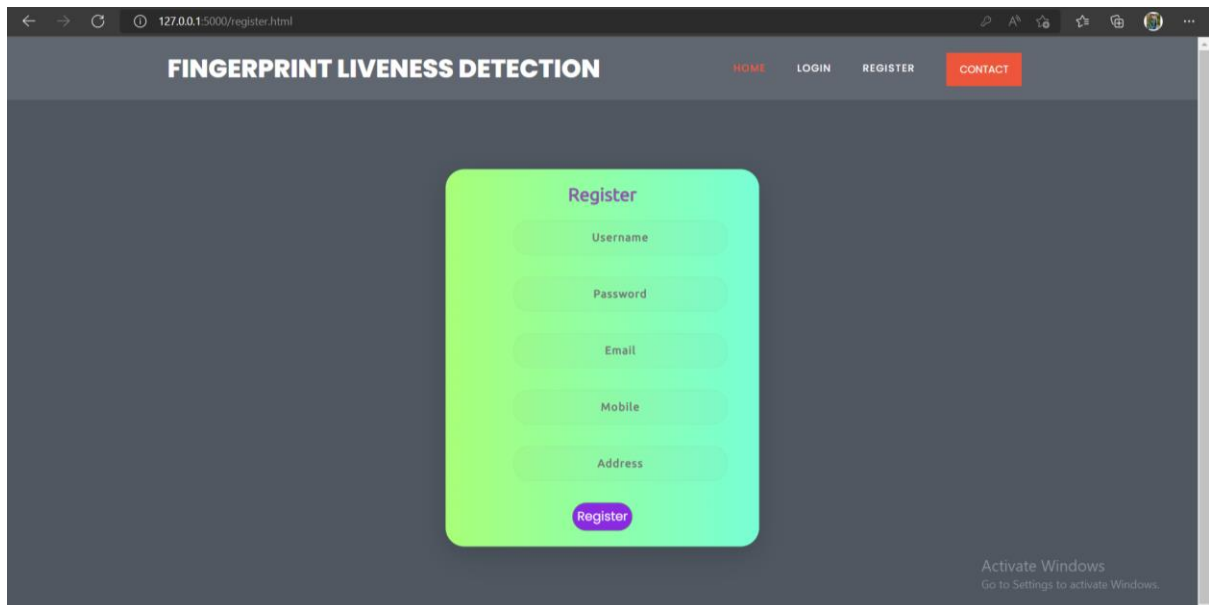
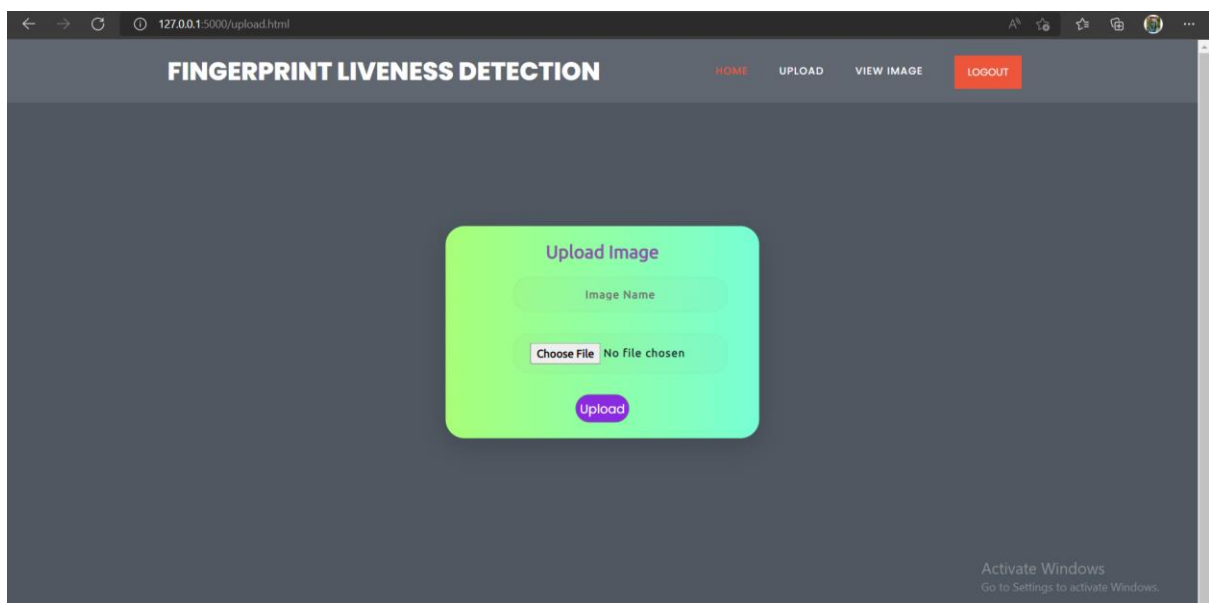


Fig 8.2.2: Login Page



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/register.html". The page title is "FINGERPRINT LIVENESS DETECTION". The navigation bar includes links for "HOME", "LOGIN", "REGISTER", and "CONTACT". The main content area features a registration form with the following fields: "Username", "Password", "Email", "Mobile", and "Address". A "Register" button is located at the bottom of the form. An "Activate Windows" watermark is visible in the bottom right corner.

Fig 8.2.3: User Registration Page



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/upload.html". The page title is "FINGERPRINT LIVENESS DETECTION". The navigation bar includes links for "HOME", "UPLOAD", "VIEW IMAGE", and "LOGOUT". The main content area features an "Upload Image" form with the following fields: "Image Name" and a "Choose File" button. An "Upload" button is located at the bottom of the form. An "Activate Windows" watermark is visible in the bottom right corner.

Fig 8.2.4: To upload the fingerprint images

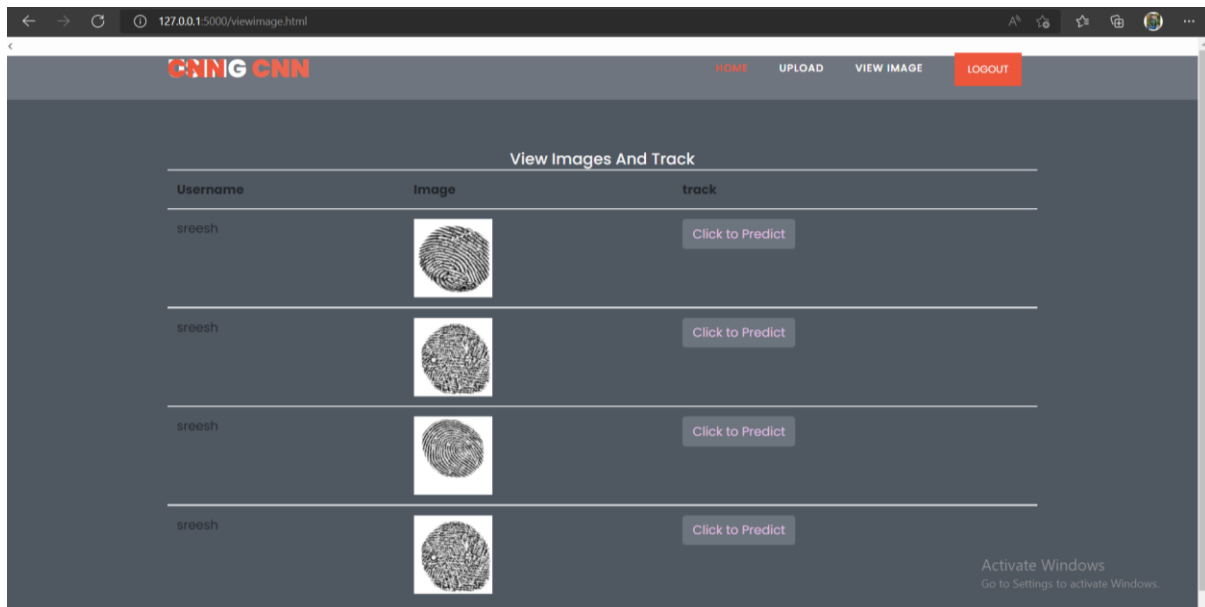


Fig 8.2.5: Pre-loaded fingerprint images that is stored as database

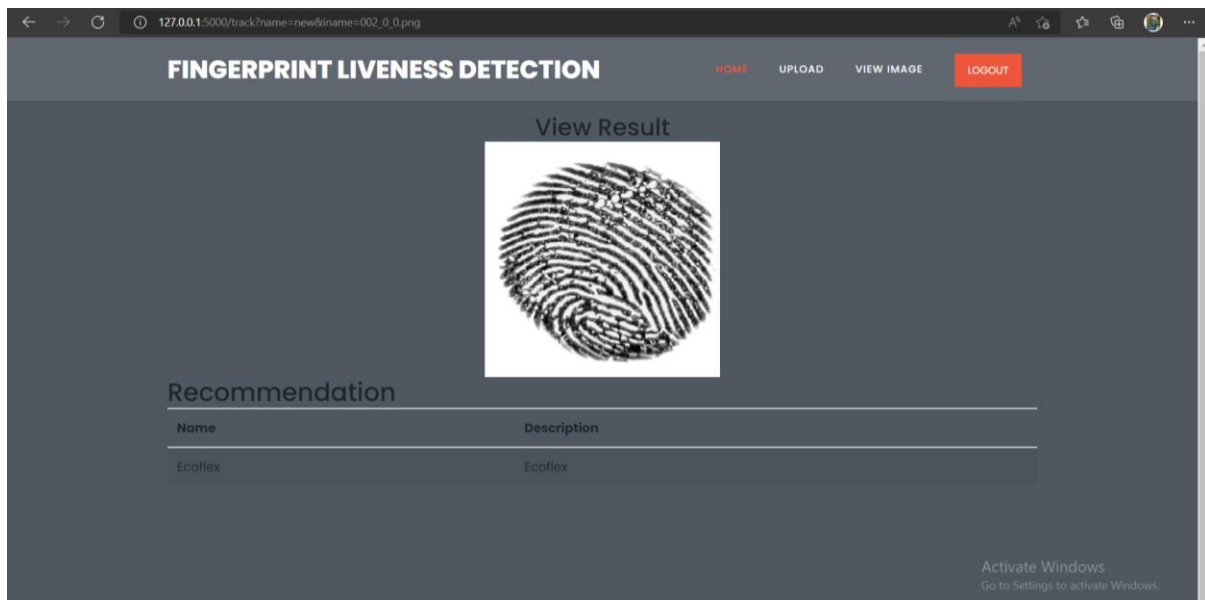


Fig 8.2.6: Detection of Fake Fingerprint(Ecoflex)

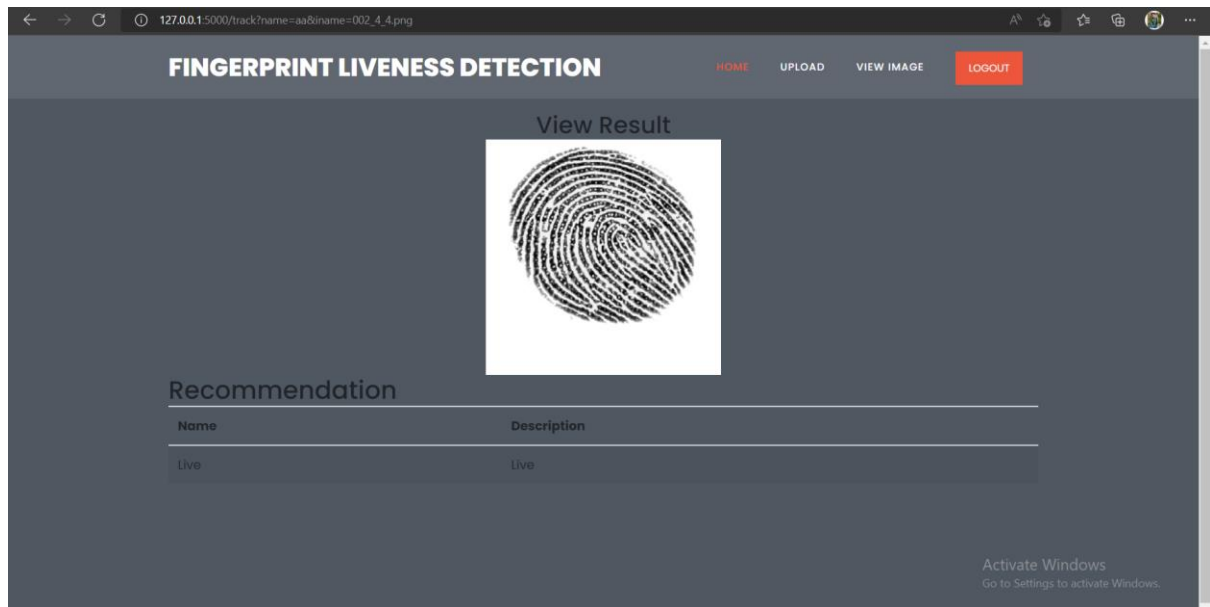


Fig 8.2.7: Prediction of Live Fingerprint

CHAPTER 9: CONCLUSION

In this study, we developed a comprehensive framework for real-time and accurate fingerprint liveness detection, as well as a detailed comparison of CNN and SVM models. Overall, we propose using machine learning to build a full framework for real-time and accurate liveness identification of fingerprints.

As a result, we can infer that a model with an accuracy of over 94% has been developed and deployed. This project also includes a graphical representation of a comparison between CNN and SVM models, which aids in effective analysis. A viable real-time application will arise from a low-cost, low-memory device using CNN.

REFERENCES

- [1] Bozhao Tan, Stephanie Schuckers, “Spoofing Protection for Fingerprint Scanner by Fusing Ridge Signal and Valley Noise” 2009 Department of Electrical and Computer Engineering Clarkson University
- [2] Jeena Sara Vijul, Sruthy S, “SVM AND RANDOM FOREST CLASSIFICATION METHODS FOR FINGERPRINT LIVENESS DETECTION” 2018, IRJET ISO 9001:2008 Certified Journal.
- [3] Chengsheng Yuan, Xinting Li, Q. M. Jonathan Wu, Jin Li, Xingming Sun, “Fingerprint Liveness Detection from Different Fingerprint Materials Using Convolutional Neural Network and Principal Component Analysis”2017, DOI :10.3970/cmc.2017.053.357
- [4] Bozhao Tan S. Schuckers, “Liveness Detection for Fingerprint Scanners Based on the Statistics of Wavelet Signal Processing” 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW 06) DOI: 10.1109/CVPRW.2006.120
- [5] Athira Raju Pillai Manju Manuel Y Premson, “Fingerprint Liveness Detection with Feature Level Fusion Techniques using SVM and Deep Neural Network” 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) DOI: 10.1109/RTEICT42901.2018.9012600