Chapter 1

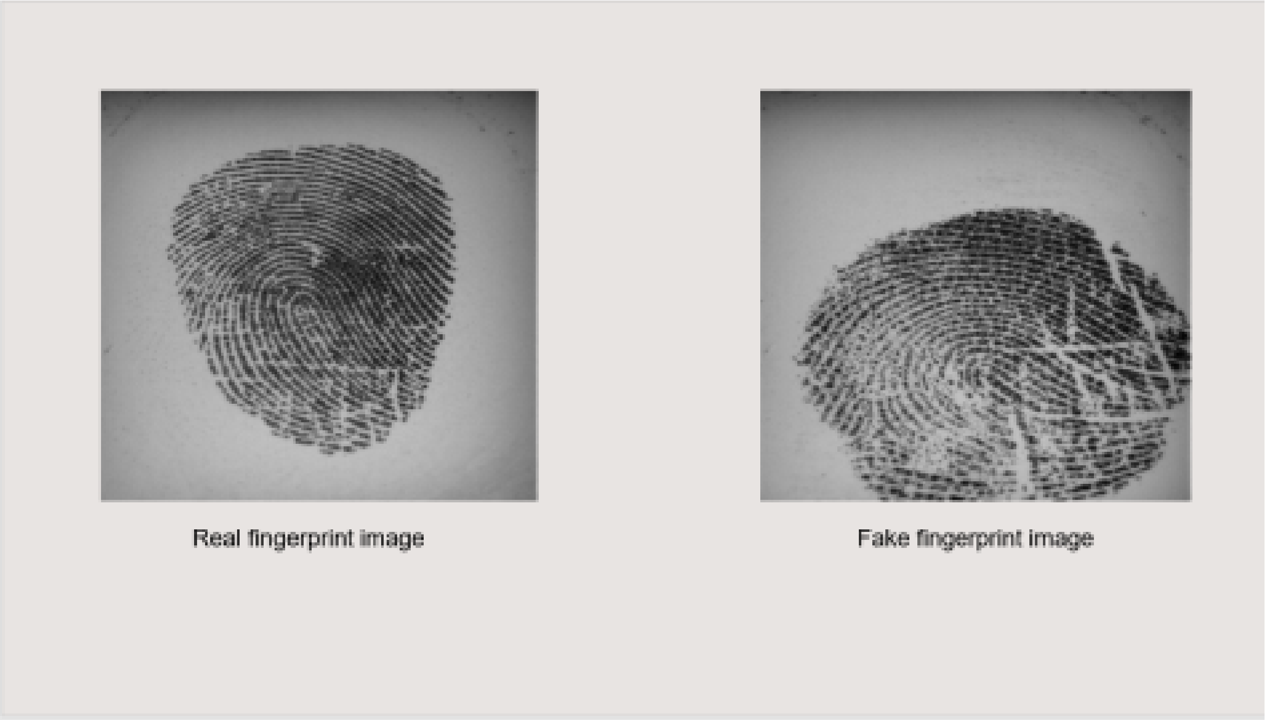
Introduction

1.1 What is liveness or spoof detection?

In biometrics, biometric recognition is the ability of a system to determine whether a fingerprint is real (from a living person present at the time of collection) or fake (from a fake artifact or inanimate body part). This is biometric authentication in which a replica of an individual's unique biometric (such as a fingerprint or a 3D silicone mask) is presented to a biometric scanner to fool system-imposed identification and authentication procedures, or go around It includes a set of technical features to combat spoofing attacks. . Liveness Check collects data from scanners and biometric readers, then uses algorithms to analyze the data to verify whether the source is from fake shows.

1.2 What are the known fingerprint spoofs?

- Fingerprint spoofing attacks (also known as presentation attacks or PA in tech nical terms) use mimics (presentation attack vehicles or PAIs) made from a variety of readily available materials such as glue, wax, clay, and silicone. The first fingerprint fraud was reported in 1998 by NETWORK Computing.
- These attacks have been the subject of many experiments and publications over the past decades on fingerprint forgery detection techniques and techniques, also known as presentation attack detection (PAD). .



Real fingerprint image

Fake fingerprint image

Figure 1: Real and fake fingerprints

1.3 How can fake fingerprints be created?

Fake fingerprints can be created using a variety of materials and techniques. Here are the general methods:

Latex: One common method involves creating a replica of an actual fingerprint using latex or a similar material. The residue can then be used to make a fake fingerprint that can be placed on the surface of the water to fool the fingerprint sensor.

Gelatin: Gelatin is another material that can be used to make fake fingerprints. A mold is made from a real fingerprint and gelatin is poured into the mold to make a fake fingerprint.

Duct tape: Duct tape can also be used to create fake fingerprints. Tape is applied to the surface, then dust or powder is applied to create the imprint.

3D Printing: With the increasing availability of 3D printers, it has become easier to create 3D fingerprint models that can be used to create fake fingerprints.

Artificial skin: Some researchers have created artificial skin that mimics the texture and appearance of human skin, including fingerprints. These artificial fingerprints can be used to create fake fingerprints that are difficult to detect.

It should be noted that making fake fingerprints is illegal and can be used for criminal purposes such as identity theft and fraud. To ensure the security and integrity of biometric identification systems, it is important to develop reliable methods to detect fake fingerprints.

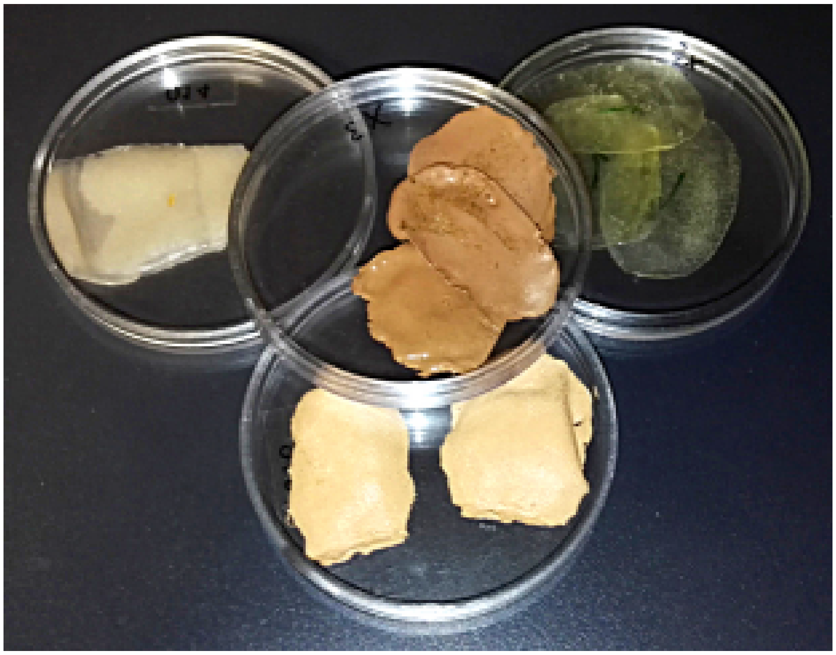


Figure 2: Example of creating fake fingerprint

1.4 How can fake fingerprints be used?

- Identity theft: Criminals can use fake fingerprints to impersonate others and gain access to personal information and assets. For example, a fake fingerprint can be used to unlock your smartphone, access your bank account or break into your secure building.

- Access control bypass: Fake fingerprints can be used to bypass access control sys tems that rely on biometrics. For example, criminals can create fake fingerprints to gain access to high-security areas or steal valuable equipment.

- Financial fraud: Criminals can use fake fingerprints to gain unauthorized access to other people’s financial accounts or make fraudulent transactions. For exam ple, a fake fingerprint can be created to authorize a wire transfer or withdraw cash from an ATM.

- Espionage: Spies and other attackers can use fake fingerprints to gain unautho rized access to sensitive information and facilities. For example, spies may use fake fingerprints to gain access to government agencies and research centers.

1.5 How to detect liveness in fingerprint presentation at tacks?

Technics to overcome presentation attacks on biometric fingerprint systems are di vided into static and dynamic methods.

1.5.1 Static methods

- They compare a point with other fingerprints. It can descry the absence of details in fake fingerprints, similar as holes, differences in patterns, and unusual features(similar as air bubbles) compared to real fingerprints. The discovery of tampered fingerprints similar as noise and smirches has long been known in forensics.

- Uprooted from a single point prisoner, static features similar as skin pliant ness, sweat-grounded features, textural features similar as smoothness(aka face roughness) and morphology can be used. For illustration, natural skin is gener ally smoother than accoutrements similar as gelatin and silicone polymers.

1.5.2 Dynamic methods

- Multiple frames of fingerprints (aka fusion) and perform a deeper analysis to detect signs of life in the captured fingerprints.

- Skin distortion analysis: The skin turns white under pressure. This effect is seen when the tip of the finger is pressed and the bleeding stops due to tissue compression. In addition, the user can intentionally increase the deformation of the skin by moving the user's finger while pressing on the surface of the scanner.

- Blood flow detection. The idea then's to capture the blood inflow in the skin to distinguish a live cutlet from an artificial cutlet.

- Active Severance Discovery: Active pores and ionic sweat fluid only live in living fritters and are delicate to reproduce.

.

Chapter 2

Machine learning methods and models

- Machine learning is a branch of computer science that focuses on developing al gorithms and models that can learn and make predictions or decisions based on data. It is part of artificial intelligence (AI), which allows computers to automat ically improve their abilities on problems without being explicitly programmed.

- In other words, machine learning is the process of training computers to rec ognize patterns and relationships in data by using statistical techniques such as regression, clustering, and classification. This technique allows machines to learn from examples and experiences, and then use that learning to improve their abilities on new problems or data.

2.1 Work flowchart

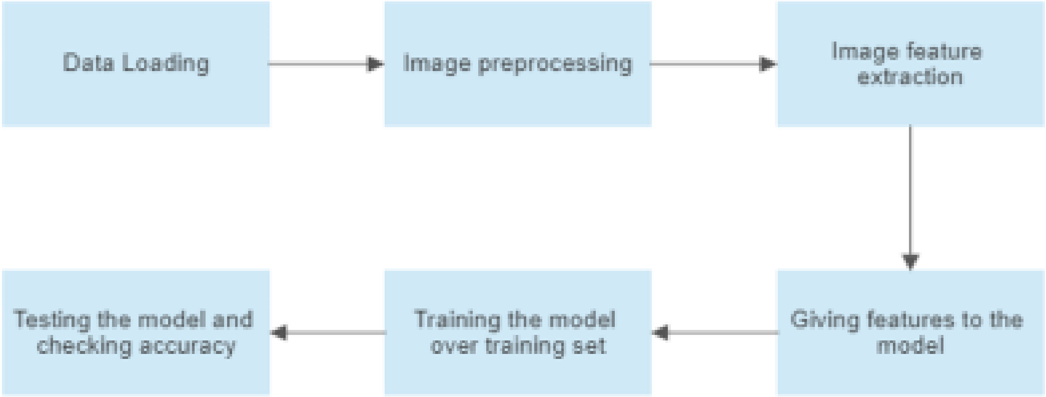

Data Loading
Image preprocessing
Image feature extraction
Giving features to the model
Training the model over training set
Testing the model and checking accuracy

Figure 2: Project Workflow

2.2 Image pre-processing

- Processing techniques such as image enhancement and smoothing, thresholding and edge detection are used to make features in the data more visible for better accuracy.

- Median and adaptive thresholding are two commonly used techniques for dividing an image into foreground and background based on pixel intensity.

- Average thresholding is a simple thresholding method that divides an image by comparing the intensity value of each pixel to a specified threshold value. The threshold value is usually set to the average intensity value of the image, hence the name ”threshold”.

- Adaptive thresholding can be more accurate than average thresholding, espe cially for images with non-uniform illumination or different contrast. However, it may not be suitable for more computationally intensive and real-time applica tions.

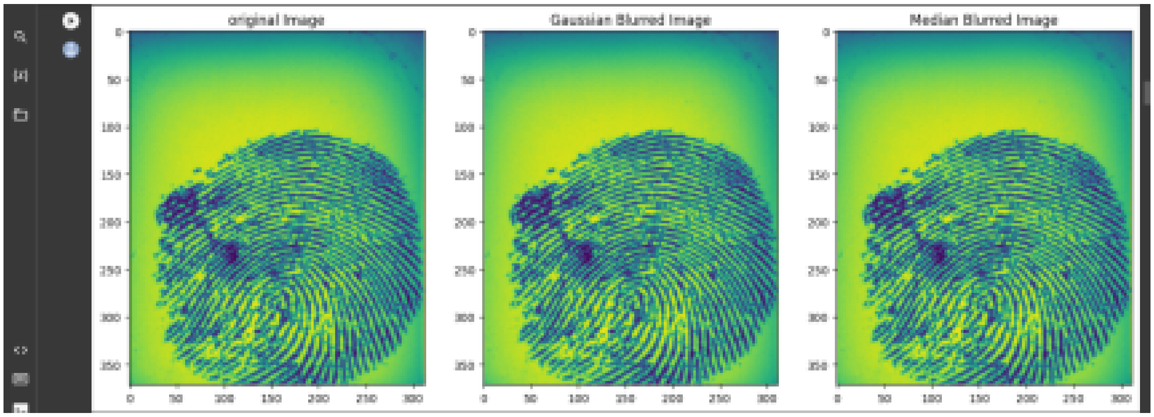


Figure 3: Average and Adaptive Thresholding

2.3 Feature Extraction

2.3.1 Minutiae Features of a fingerprint:

- Minutiae features are unique to each fingerprint and are unique features and features used to identify and compare fingerprints. These features include ridges, bifurcations, and dots, which are the most common and distinctive patterns in fingerprints.

- Ridges are the ends of the ridges in the fingerprint, and bifurcations are the points where the ridges split into two branches. Points are small ridges that do not form terminations or bifurcations. These minute features are stable, consistent and highly discriminative, making them suitable for fingerprint recognition.

- Minutiae-based fingerprint recognition systems use this feature to extract fin gerprint images to represent a template or fingerprint. This template is then compared with other templates in the database to determine if it matches.

- The accuracy and reliability of a fingerprint recognition system depends on the quality and quantity of minute features extracted from the fingerprint image.

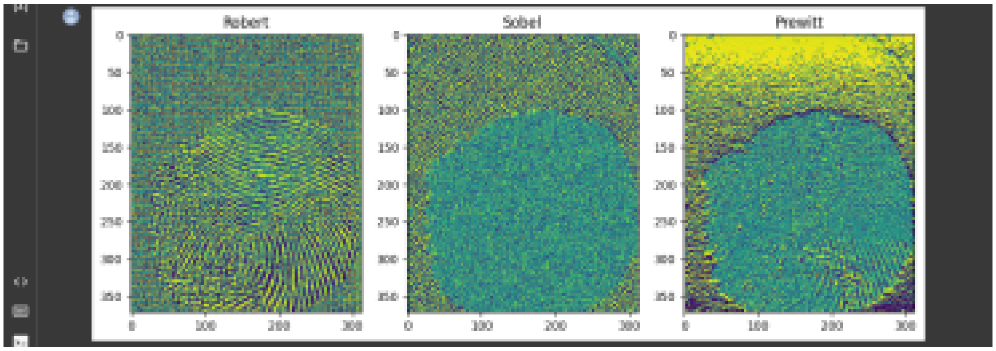


Figure 4: Robert, Sobel and Prewitt filters for edge detection

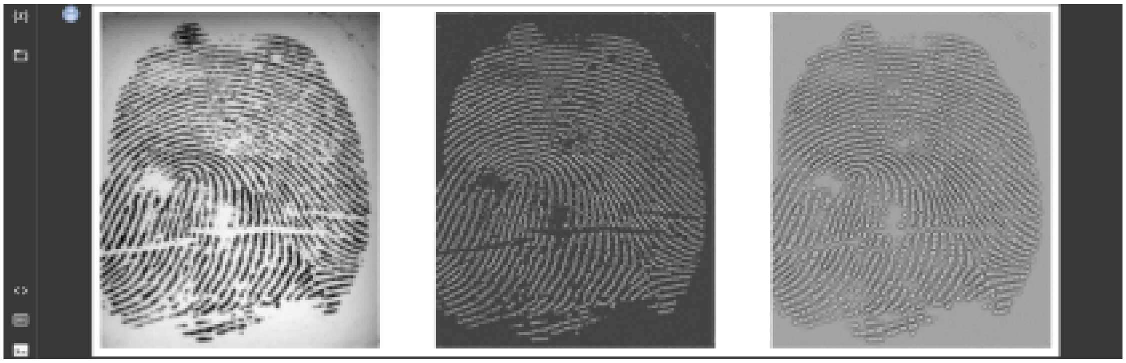


Figure 5: Detecting Ridges

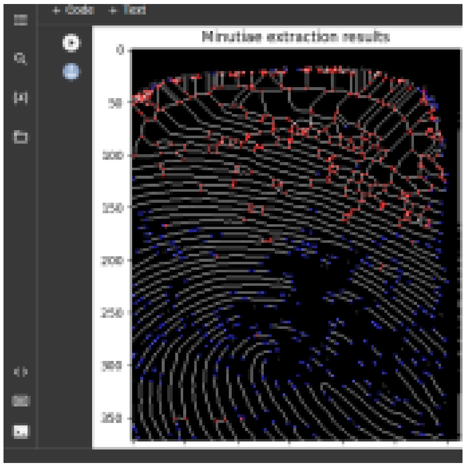


Figure 6: Minutiae Features

2.3.2 LBP features of images:

- Local binary pattern (LBP) is a texture descriptor that can be used in image classification problems. LBP works by comparing the intensity value of a pixel with its surrounding neighbors and encoding the result as a binary pattern. The extracted patterns can then be used as features for image classification problems.

  To use the LBP feature for image classification, the image is first divided into sub-regions and the LBP pattern is calculated for each pixel in each region. The LBP pattern is calculated by limiting the difference between the intensity value of the central pixel and its neighboring neighbors and encoding the result as a binary value. The resulting binary values are combined to form a histogram of LBP patterns for each region.

2.4 SVM

SVM (Support Vector Machine) works by finding the best hyperplane that divides two data points with the maximum distance. In binary image classification, data points correspond to image features such as pixel values, textures, or edges, and two classes correspond to two classification categories. To use SVM for binary image classification, the image is first processed and features are extracted from it. Common feature extraction techniques include edge detection, texture analysis, and histogram based features. The extracted features are used to train the SVM model using a set of labeled images.

During training, the SVM algorithm learns the best hyperplane that can separate two nodes of the node class with the maximum distance. Once the SVM model is trained, it can be used to classify new, unseen images into one of two categories based on their features. Image features are converted into feature vectors and fed into a trained SVM model, which then predicts class labels based on the position of the feature vector relative to the hyperplane.

SVM can achieve high accuracy in binary image classification problems, especially when the feature extraction process is carefully designed to capture different visual features of the two classes. SVM can be combined with other methods such as image segmentation and object detection to improve its performance in more complex image classification problems.

2.5 KNN

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm that can also be used in binary image classification problems. KNN is a non-parametric algorithm that works by finding the nearest neighbors of a data point based on distance criteria and then assigning the most common class among those nearest neighbors.

To use KNN for binary image classification, the image is first processed and features such as pixel values, texture or edge features are extracted. The extracted features are used to train the KNN model using a set of labeled images.

During training, the KNN algorithm creates a database of feature vectors and corresponding class labels. When a new image is presented to the algorithm, its features are converted into feature vectors and compared with the feature vectors in the database using distance measures such as Euclidean distance or Manhattan distance. The nearest neighbor of the new image is then determined, and the class label of the new image is based on the most common class label among its nearest neighbors.

2.6 Naive Bayes

Naive Bayes is a probabilistic machine learning algorithm that can also be used in binary image classification problems. Naive Bayes works by calculating the probability of two classes given a set of features and then dividing the image into the class with the highest probability.

To use Naive Bayes for binary image classification, an image is first processed and features such as pixel values, textures or edge features are extracted. The extracted features are used to train a Naive Bayes model using a set of labeled images.

During training, the Naive Bayes algorithm calculates the probability of each given feature in each class, as well as the prior probability of each class. This probability is used to calculate the posterior probability of each class, given a set of features for a new image, using Bayes theorem. This image is then classified into the class with the highest posterior probability.

2.7 Decision Trees

A decision tree is a machine learning algorithm that can also be used in binary im age classification problems. Decision trees are used by sequentially partitioning the database based on feature values until the samples at each leaf node belong to the same class.

To use Decision Trees for Binary Image Classification, an image is first processed and features such as pixel values, textures or edge features are extracted. The ex tracted features are used to train a Decision Tree model using a set of labeled images.

During training, the Decision Tree algorithm sequentially partitions the database based on feature values, so each partition increases the purity of the resulting parti tion. The purity of a partition is usually measured by a measure, such as the Gini coefficient, which determines whether the samples in a partition belong to the same class. Split continues until the sample at each point of the leaf belongs to the same class or until the stopping criteria is met.

2.8 Logistic Regression

Logistic regression is a statistical machine learning algorithm that can also be used in binary image classification problems. Logistic regression works by estimating the probability of an image belonging to one of the two classes based on its features, then assigning the image to the class with the highest probability.

To use logistic regression for binary image classification, the image is first processed and features such as pixel values, texture or edge features are extracted. The extracted features are used to train a logistic regression model using a set of labeled images.

During training, the logistic regression algorithm uses a logistic function to esti mate the probability of an image belonging to one of the two classes based on its fea tures. The logistic function maps input characteristics to probability values between 0 and 1 and can be interpreted as the probability that the positive class belongs to the class. The logistic regression algorithm then learns parameters from the logistic function that maximizes the likelihood of the training data given the parameters.

Chapter 3

Convolutional Neural Networks

- Fake fingerprint detection using CNNs involves training a deep neural network model to determine whether the fingerprint is fake or live. CNNs are particularly suitable for this problem because they can learn a hierarchical representation of the input data suitable for image classification problems.
- Model Architecture: An architecture can consist of several convolutional layers, pooling layers, and fully connected layers. The transaction layer extracts fea tures from the input image and the basic layer represents the feature map to reduce the computational complexity. The fully connected layer performs the final classification of the input image.
- Training: The CNN model is trained on a pre-processed database using a pre distributed and stochastic gradient stream. The goal of the training process is to reduce the classification errors in the training database.
- Testing: After the model is trained, it is evaluated on a test database to mea sure its performance in detecting fake fingerprints. Model performance can be measured by metrics such as accuracy, precision, recall, and F1-score.

Chapter 4

Results

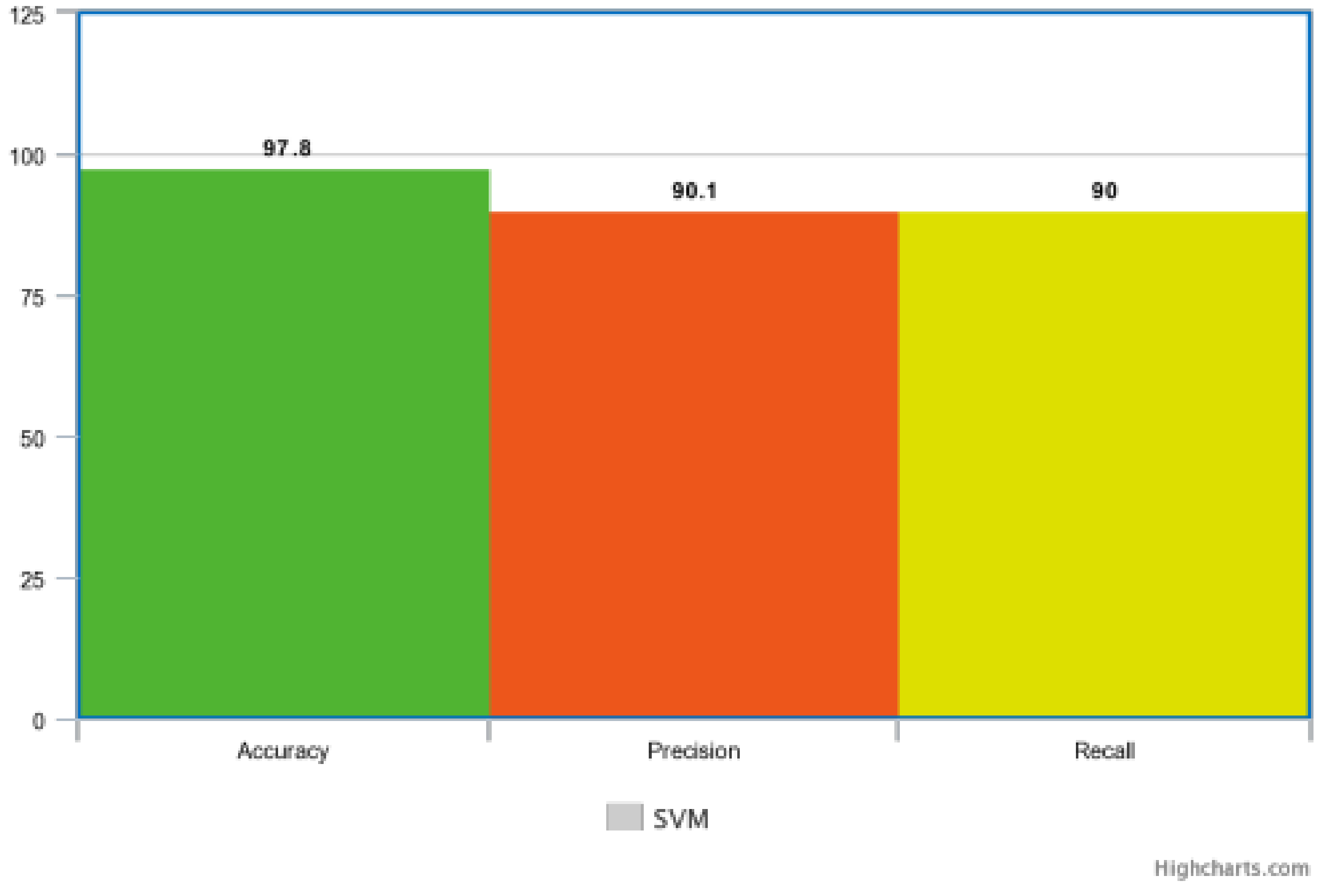


Figure 7: SVM using LBP features

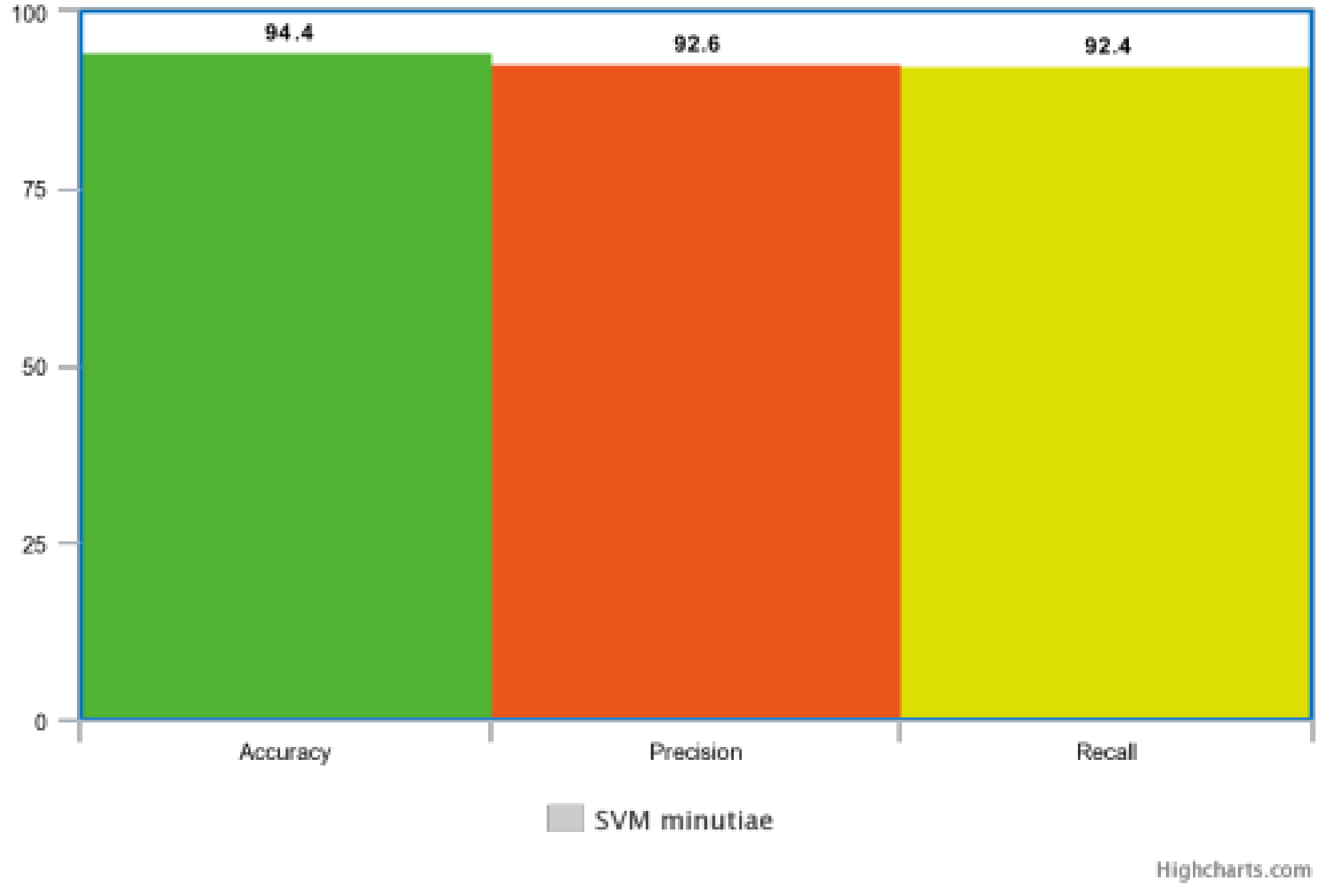


Figure 8: SVM using minutiae features

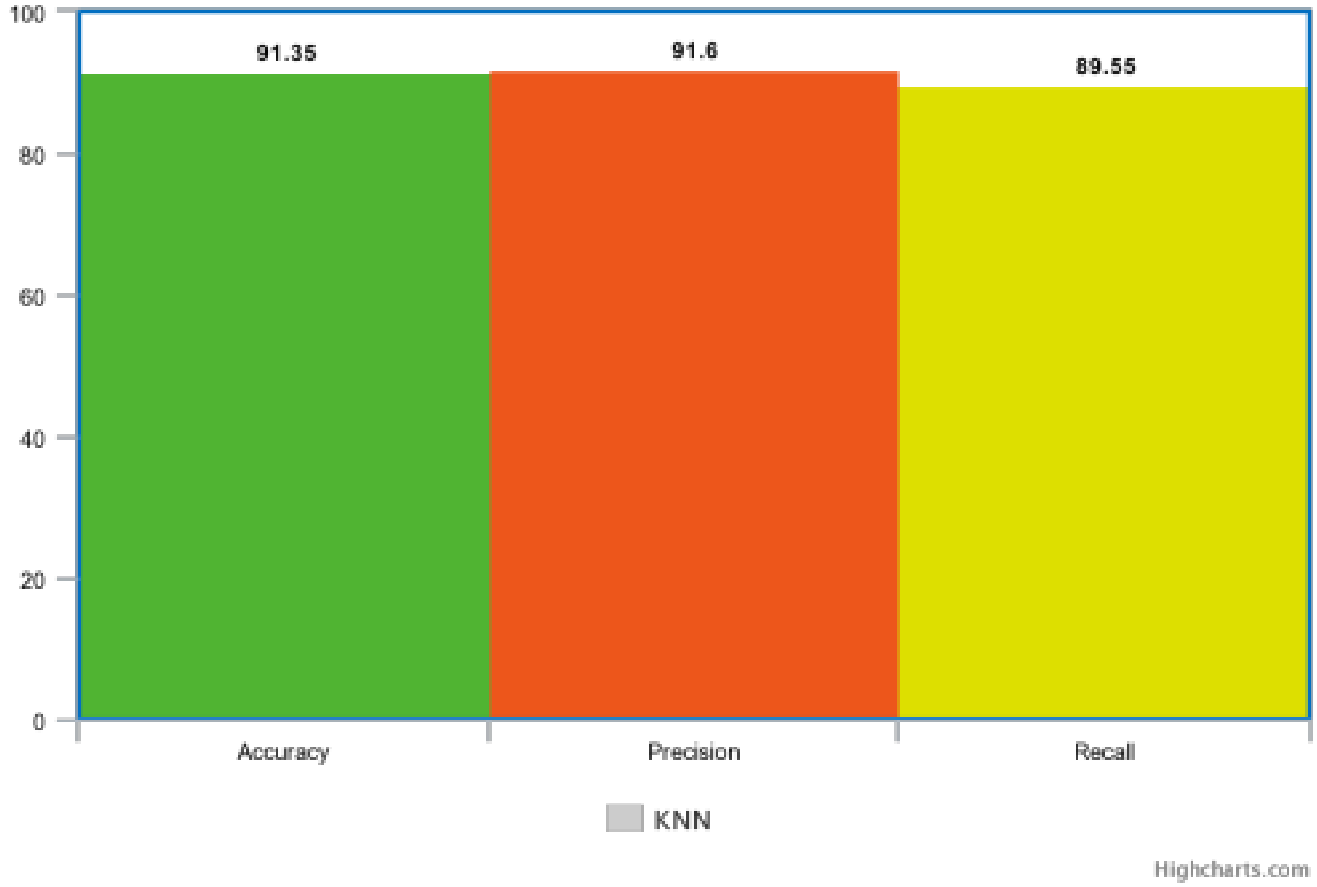


Figure 9: KNN algorithm result

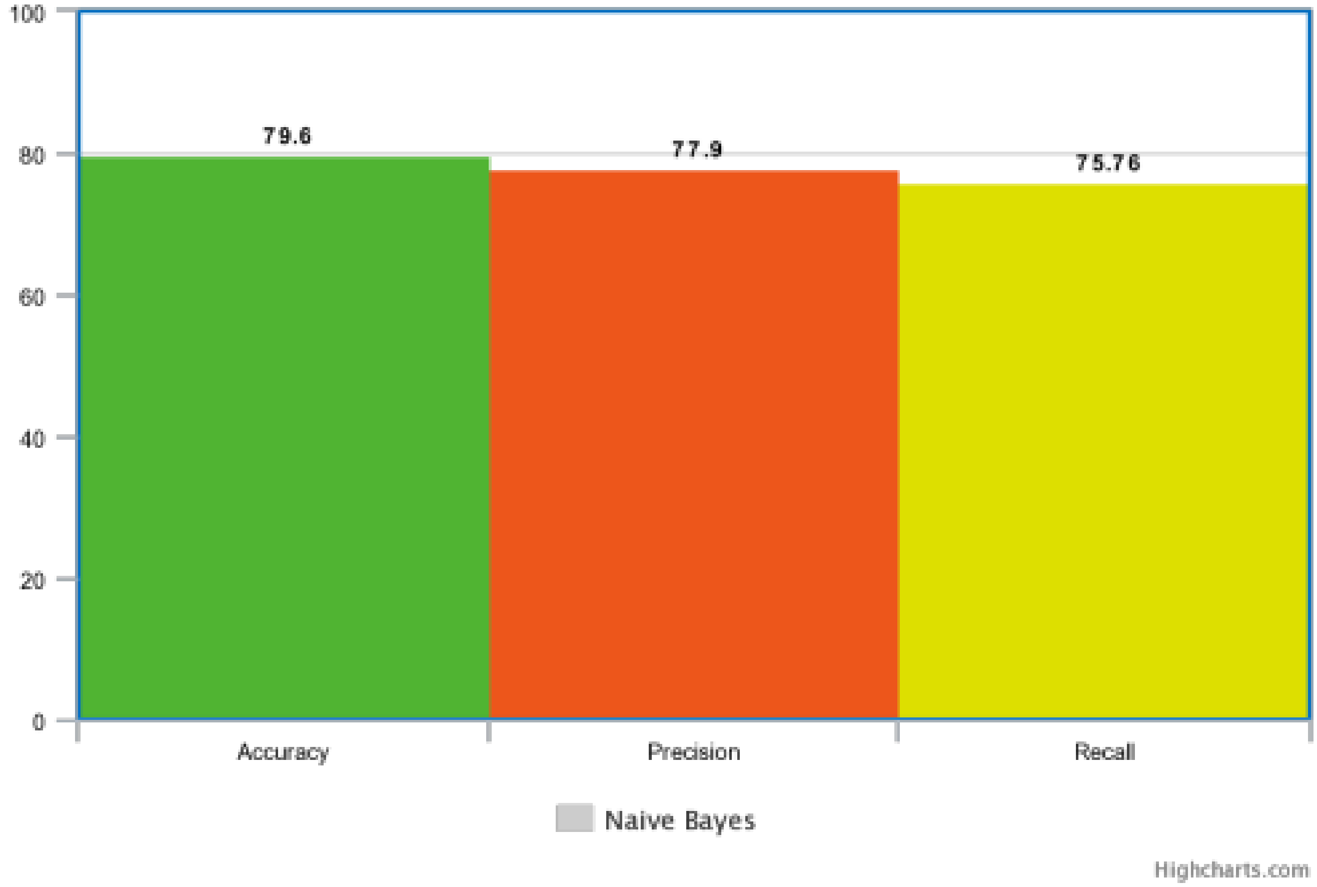


Figure 10: Naive Bayes algorithm result

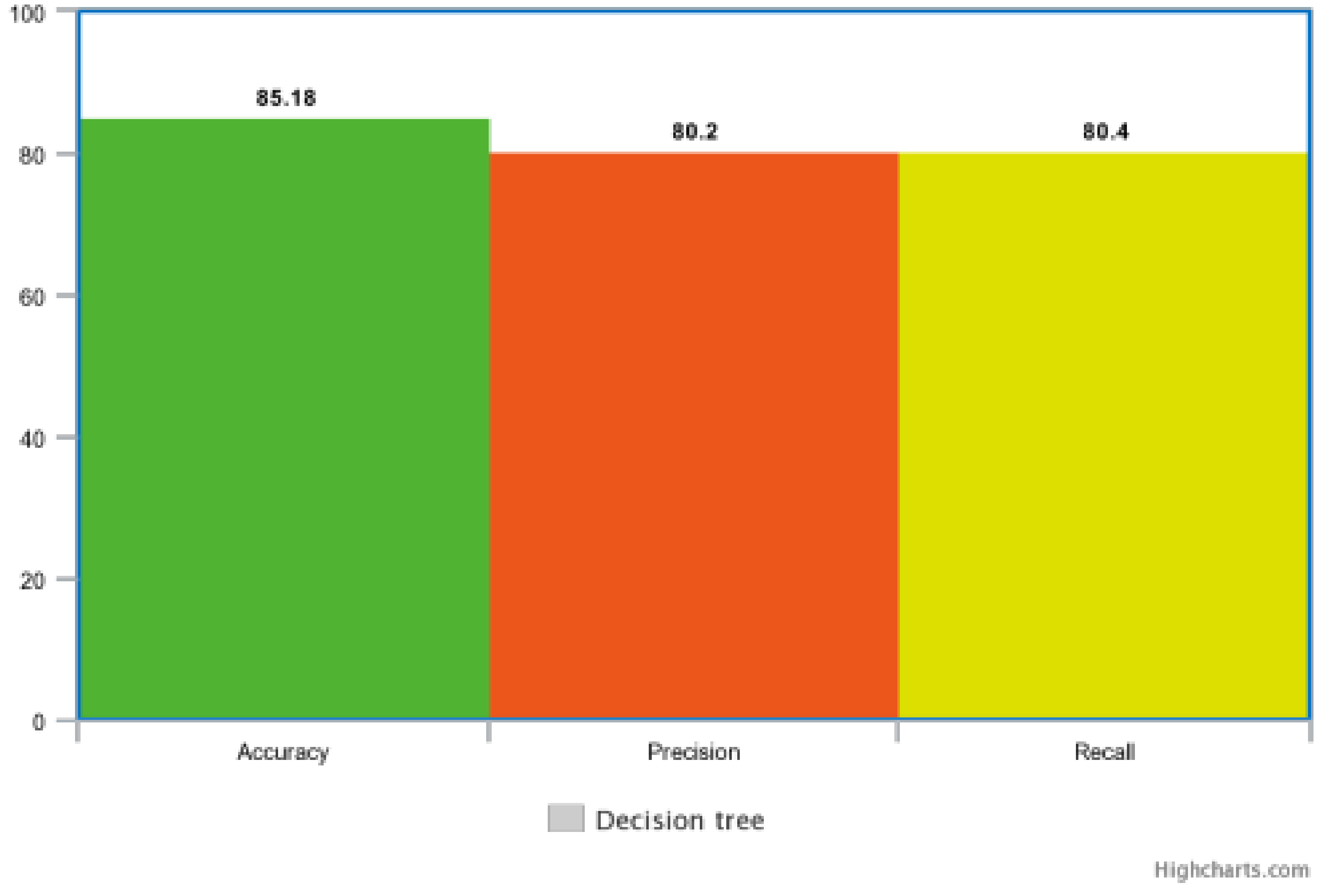


Figure 11: Decision Tree algorithm result

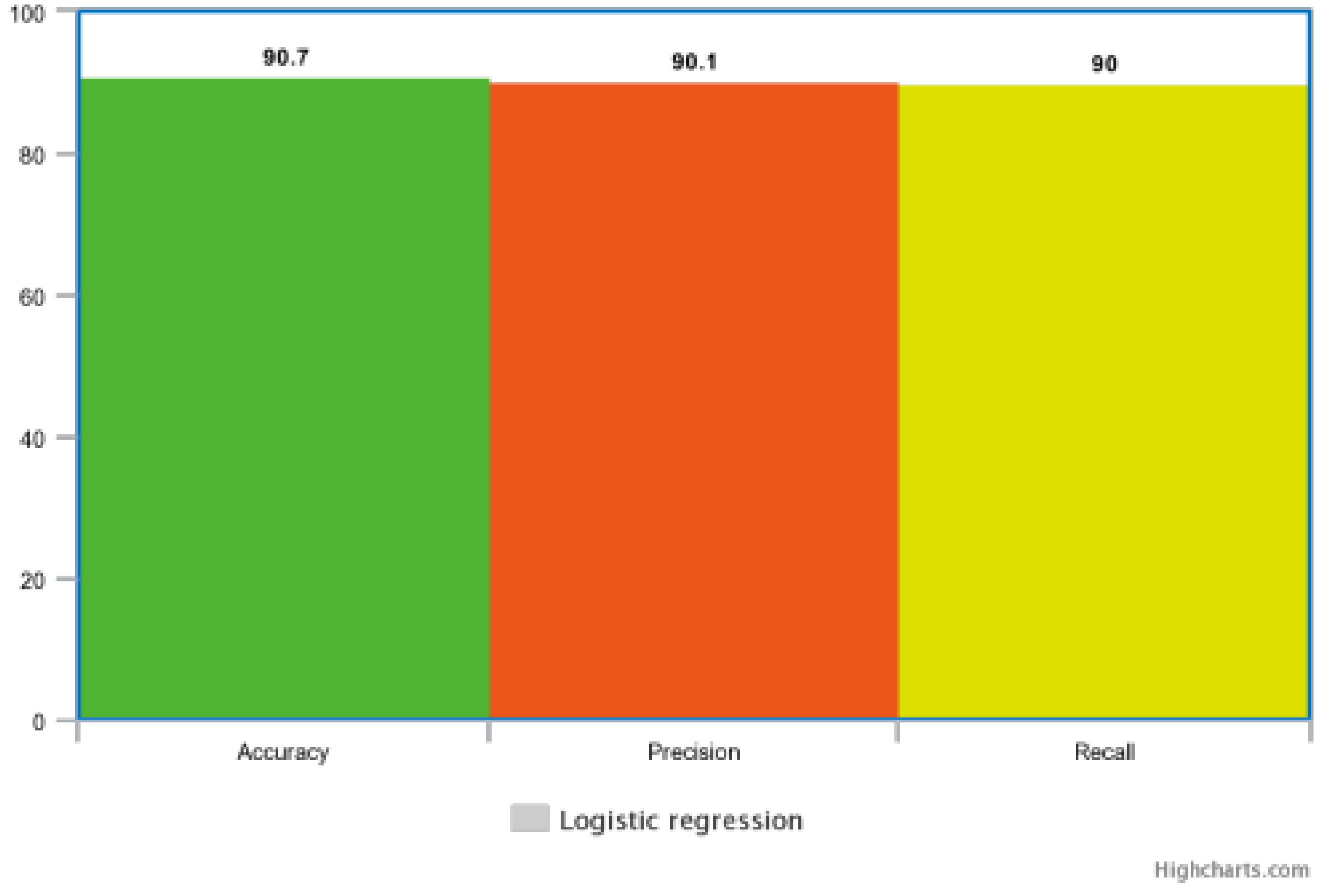


Figure 12: Logistic Regression algorithm result

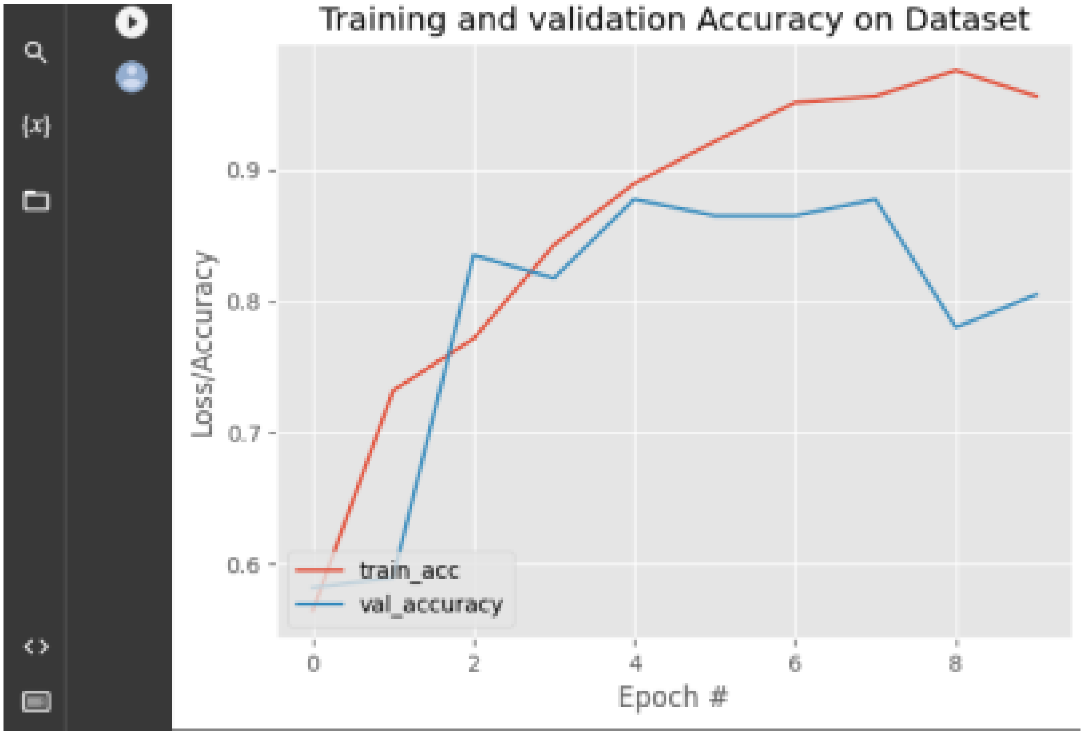


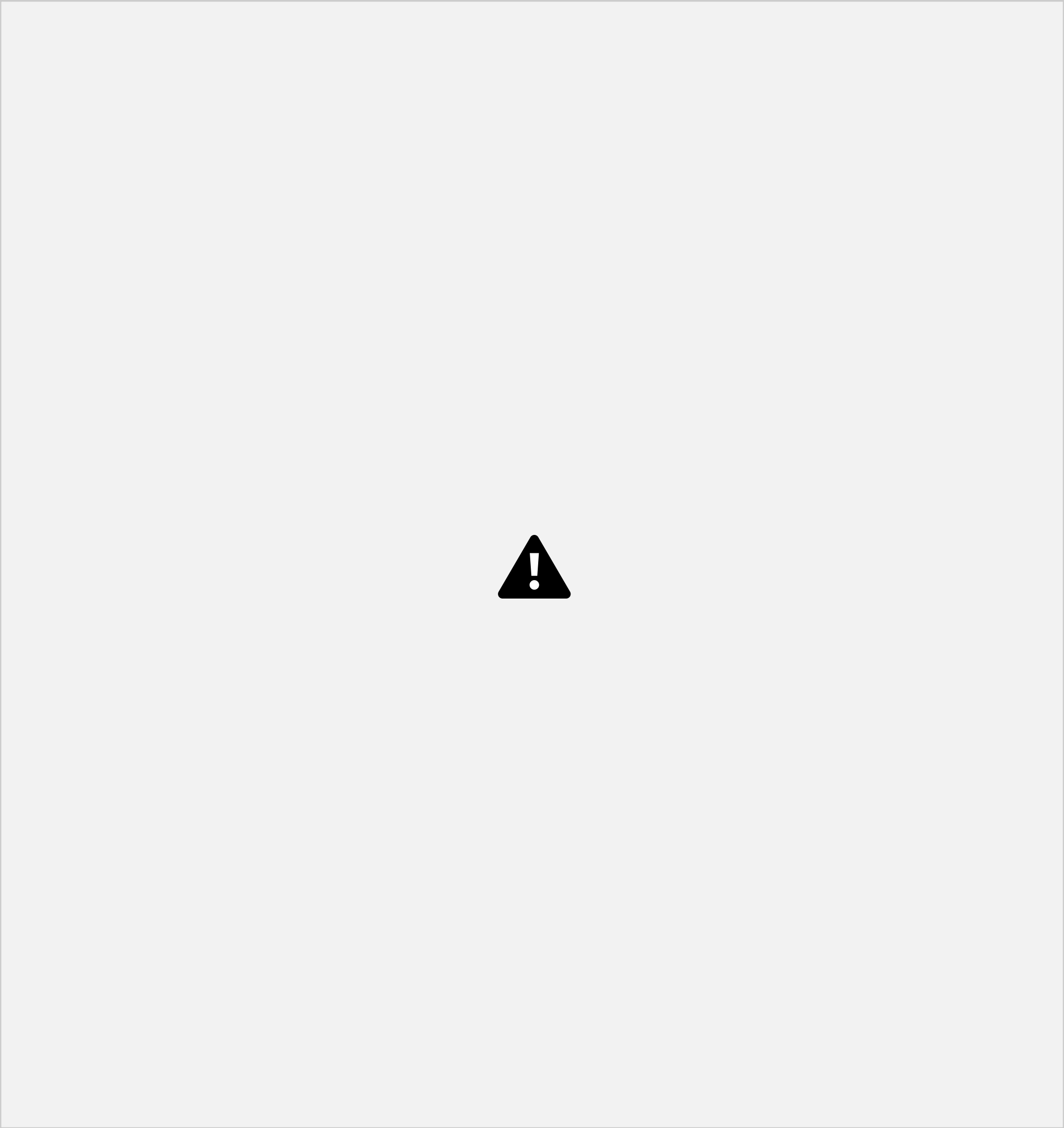
Figure 13: CNN algorithm result

Overall Comparative Analysis

So here we conclude that after applying various machine learning models, SVM using LBP gave the best accuracy of 97.8

4.1 Checking for live fingerprint

On checking for live fingerprint, model predicted that the input is 100 percent live.



Input live fingerprint

Output

Chapter 5

Future Works and Conclusion

5.1 Future Works

Fake and direct fingerprint detection is an important research area with a wide range of practical applications to come. Future opportunities for fake and instant fingerprint detection:

Mobile devices: With the proliferation of mobile devices for biometric recognition, the need for reliable methods to detect fake and live fingerprints on these devices is increasing. Future research can be focused on developing lightweight and efficient algorithms for artificial and live fingerprint identification on mobile devices.

IoT devices: The Internet of Things (IoT) is growing rapidly, and many IoT de vices use biometric authentication for access control. Future research could focus on developing methods to detect fake and direct fingerprints on these devices, which often have limited resources and processing power.

Multi-modal biometrics: Combining different biometric methods, such as finger print and facial recognition, can improve the accuracy and reliability of biometric recognition systems. Future research may focus on developing methods to detect fake and direct fingerprints in multi-modal biometric systems.

Adversary attacks: Adversary attacks are a growing concern in the field of biomet ric security. Future research may focus on developing methods to detect and mitigate adversary attacks on biometric recognition systems, including identifying fake and live fingerprints.

Privacy Issues: Biometric data is highly sensitive and raises privacy issues. Future research may focus on developing methods to detect fake and live fingerprints that protect individual privacy and prevent misuse of biometric data. Overall, the future scope of fake and live fingerprint detection is wide and diverse,

and there is a need for continuous research and development in this area to ensure the security and privacy of biometric identification systems.

5.2 Conclusion

In conclusion, our project focuses on developing and evaluating a machine-based ap proach to detect fake and direct fingerprints. We examined various methods such as SVM, KNN, CNN, Decision tree and Naive Bayes and evaluated their performance using real and fake fingerprint databases.

Overall, our project has demonstrated the potential of machine learning-based approaches in detecting artificial and direct fingerprints and laid the foundation for future research in this area. Our research can help improve the security and reliability of biometric identification systems and ensure user privacy and security.