# DETECTION OF MOULDED FINGERPRINT

## MINOR PROJECT

## FINAL REPORT


Submitted By:

**Aditri Srivastava (9919103092)**

**Aryan Kumar (9919103093)**

**Shazia Salim (9919103099)**

**Shantanu Goyal (9919103103)**

Under Supervision Of:

**Dr. Raju Pal**

विद्या तत्व ज्योतिसमः

**Department of CSE/IT**

**Jaypee Institute of Information Technology University, Noida**


**DECEMBER 2021**

**ACKNOWLEDGEMENT**

I would like to place on record my deep sense of gratitude to Dr. RAJU PAL assistant professor (Senior grade) Jaypee Institute of Information Technology, India for his generous guidance, help and useful suggestions.

I also wish to extend my thanks to my project partners and other classmates for their insightful comments and constructive suggestions to improve the quality of this project work.

**Signature(s) of Students**

Aditri Srivastava (9919103092)
Aryan Kumar (9919103093)
Shazia Salim (9919103099)
Shantanu Goyal (9919103103)

# DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and beliefs, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place:Noida

Date:01-12-2021

Name:Aditri Srivastava

Enrolment No.:9919103092

Name:Aryan Kumar

Enrolment No.:9919103093

Name:Shazia Salim

Enrolment No.:9919103099

Name:Shantanu Goyal

Enrolment No.:9919103103

# CERTIFICATE

This is to certify that the work titled “Detection of Moulded Fingerprint” submitted by Name of Students of B.Tech of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Digital Signature of Supervisor

Dr. Raju Pal

Assistant Professor (Sr. Grade)

01-12-2021

# ABSTRACT

Fingerprint alteration is a challenge that poses enormous security risks. As a result, many research efforts in the scientific community have attempted to address the issue. However, non-existence of publicly available datasets that contain obfuscation and distortion of fingerprints makes it difficult to identify the type of alteration and thus the study and development of mechanism to correct the alteration and correctly identify individuals. In this work we present the publicly available Sokoto Coventry Fingerprint Dataset (SOCOFing Dataset) with unique attributes. SOCOFing is made up of 6,000 fingerprint images from 600 African subjects and contains unique attributes such as labels for gender, hand and finger name as well as synthetically altered versions with three different levels of alteration for obliteration, central rotation, and z-cut. . Moreover, we propose a Convolutional Neural Network (CNN) to identify images as real and altered with gender classification.

# TABLE OF CONTENTS


| Content | Page No. |
|---|---|
| Acknowledgement | **1** |
| Abstract | **4** |
| List of Figures | **6** |
| **Chapter 1: INTRODUCTION** | **7** |
| 1.1 Convolutional Neural Network | **7** |
| 1.2 Common uses for CNNs | **7** |
| 1.3 Layers used to build ConvNets | **8** |
| **Chapter 2: BACKGROUND STUDY** | **10** |
| 2.1 Literature Survey for Detection of Mouled Fingerprints | **10** |
| 2.2 Literature Survey for Detection of gender using Fingerprints | **12** |
| **Chapter 3: REQUIREMENT ANALYSIS** | **13** |
| **Chapter 4: DETAILED DESIGN** | **14** |
| 4.1 Workflow for Model | **14** |
| 4.2 Workflow for Flask | **14** |
| **Chapter 5: IMPLEMENTATION** | **15** |
| 5.1 Real or Artificial Moulded fingerprints | **15** |
| 5.2 Gender Classification | **17** |
| 5.3 Integration with Flask | **18** |
| **Chapter 6: EXPERIMENTAL RESULTS AND ANALYSIS** | **21** |
| **Conclusion** | **24** |
| **Future Scope** | **24** |
| **References** | **25** |

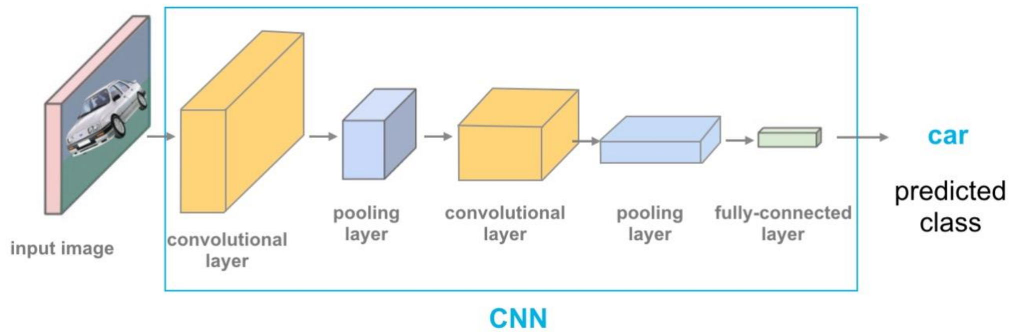
# LIST OF FIGURES


| Figure Title | Page No. |
|---|---|
| 1.a CNN Working | 7 |
| 1.b Max Pooling | 8 |
| 1.c Fully Connected Layer | 9 |
| 4.a Model Workflow | 14 |
| 4.b Flask Workflow | 14 |
| 5.1.a Fingerprint Classification Model Summary | 16 |
| 5.2.a Gender Classification Model Summary | 17 |
| 5.3.a Homepage(index.html) | 18 |
| 5.3.b Predicted Altered Fingerprint | 18 |
| 5.3.c Predicted Real Fingerprint | 19 |
| 5.3.d Predicted Female Fingerprint | 19 |
| 5.3.e Predicted Male Fingerprint | 20 |
| 6.1.a Research Paper Model Evaluation | 21 |
| 6.1.b Research Paper Model Accuracy Graph | 21 |
| 6.1.c Research Paper Model Loss Graph | 21 |
| 6.2.a Fingerprint Classification Model Evaluation | 22 |
| 6.2.b Fingerprint Classification Model Accuracy Graph | 22 |
| 6.2.c Fingerprint Classification Model Loss Graph | 22 |
| 6.3.a Gender Classification Model Evaluation | 23 |
| 6.3.b Gender Classification Model Accuracy Graph | 23 |
| 6.3.c Gender Classification Model Loss Graph | 23 |

# INTRODUCTION

## 1.1 Convolutional Neural Network (CNN)

A Convolutional neural network (CNN) is a neural network that has one or more convolutional layers and are used mainly for image processing, classification, segmentation and also for other auto correlated data. A convolution is essentially sliding a filter over the input. One helpful way to think about convolutions is this quote from Dr Prasad Samarakoon: “A convolution can be thought as “looking at a function’s surroundings to make better/accurate predictions of its outcome.” Rather than looking at an entire image at once to find certain features it can be more effective to look at smaller portions of the image.




**1.a CNN Working**

## 1.2 Common uses for CNNs

The most common use for CNNs is image classification, for example identifying satellite images that contain roads or classifying handwritten letters and digits. There are other quite mainstream tasks such as image segmentation and signal processing, for which CNNs perform well at. CNNs have been used for understanding in Natural Language Processing (NLP) and speech recognition, although often for NLP Recurrent Neural Nets (RNNs) are used. A CNN can also be implemented as a U-Net architecture, which are essentially two almost mirrored CNNs resulting in a CNN whose architecture can be presented in a U shape. U-nets are used where the output needs to be of similar size to the input such as segmentation and image improvement. More and more diverse and interesting uses are being found for CNN architectures. An example of a non-image based application is “The Unreasonable Effectiveness of Convolutional Neural Networks in Population Genetic Inference” by Lex Flagel et al. This is used to perform selective sweeps, finding gene flow, inferring population size changes, inferring rate of recombination. CNNs are also being used in astrophysics to interpret radio telescope data to predict the likely visual image to represent the data.

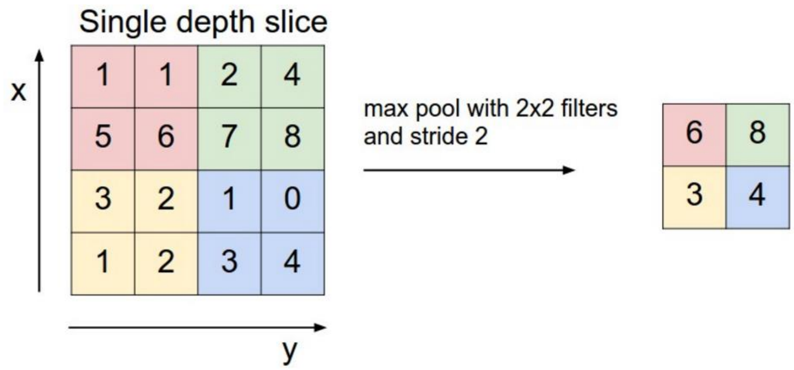
## 1.3 Layers used to build ConvNets

A convert is a sequence of layers, and every layer transforms one volume to another through a differentiable function.

### Types of layers:

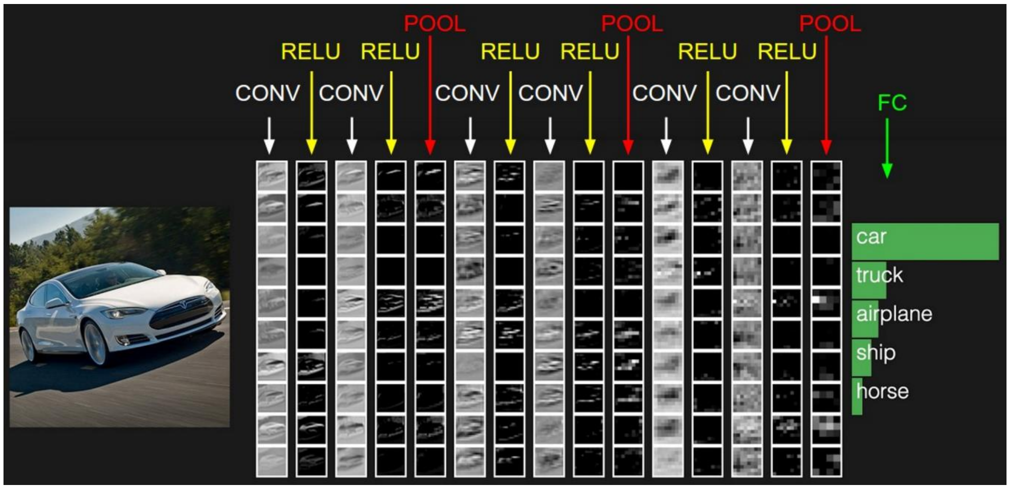
Let's take an example by running a covnets on an image of dimension 32 x 32 x 3.

1. Input Layer: This layer holds the raw input of the image with width 32, height 32, and depth 3.
2. Convolution Layer: This layer computes the output volume by computing the dot product between all filters and image patches. Suppose we use a total of 12 filters for this layer we'll get output volume of dimension 32 x 32 x 12.
3. Activation Function Layer: This layer will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are RELU: max(0, x), Sigmoid: 1/(1+e^-x), Tanh, Leaky RELU, etc. The volume remains unchanged hence output volume will have dimension 32 x 32 x 12.
4. Pool Layer: This layer is periodically inserted in the covenants and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 16x16x12.




**1.b Max Pooling**

5. Fully-Connected Layer: This layer is a regular neural network layer that takes input from the previous layer and computes the class scores and outputs the 1-D array of size equal to the number of classes.




**1.c Fully Connected Layer**

# BACKGROUND STUDY

## 1. Literature Survey for Detection of Molded Fingerprints

### a. Evaluation of Fingerprint Liveness Detection by Machine Learning Approach - A Systematic View - by Dr. Edriss Eisa Babikir Adam

This paper focuses on the implementation and evaluation of suitable machine learning algorithms to detect fingerprint liveness. The support vector machine (SVM) classifiers work with indiscriminate loads and confined grayscale array values. This leads to aliveness report of fingerprints for detection purposes. This research paper uses a dataset having 16K images from various sensors and 2K images from spoof fingerprint images. The introduction of segmentation of the image can give better clarity in feature extraction. To assess the liveness of a spoof fingerprint, this analysis used two separate approaches such as a single classifier and an ensemble approach classifier

### b. Fingerprint Recognition System

In this Paper the basic fingerprint recognition system consists of four stages: firstly, the sensor which is used for enrolment & recognition to capture the biometric data. Secondly, the pre-processing stage is used to remove unwanted data and increase the clarity of ridge structure by using enhancement techniques. Thirdly, the feature extraction stage which takes the input from the output of the pre-processing stage to extract the fingerprint features. Fourthly, the matching stage is to compare the acquired feature with the template in the database. Finally, the database which stores the features for the matching stags. In the final stage, they used KNN for matching.

### c. Fingerprint Alterations Type Detection Using Deep Convolutional Neural Network

In this work, they have presented the publicly available Coventry Fingerprints Dataset (SOCOFing Dataset) with unique attributes such as ten fingerprints for 611 different subjects, gender, hand and finger name for each image, among others. They also provide a total of 55,249 images with three levels of alteration for z-cut, obliteration, and central rotation synthetic alterations, which are the most common types of obfuscation and distortion. Moreover, we propose a Convolutional Neural Network (CNN) to identify this type of alteration. The proposed CNN model achieves a classification accuracy rate of 98.55%. Results are also compared with a residual CNN model pre-trained on ImageNet which produces an accuracy of 99.88%.

### d. Fingerprint classification using deep learning approach

In this paper, they have presented a classification method to identify detailed fingerprint information using a deep learning approach. The proposed method aimed to distinguish the specific fingerprint information such as left-right hand classification, sweat-pore classification, scratch classification and fingers classification. Due to high personalization and security issue, they privately constructed our own dataset of fingerprint images. Five state-of-the-art deep learning models such as classic CNN, Alexnet, VGG-16, Yolo-v2 and Resnet-50 were adapted to be trained from scratch for those four categories. In their experimental tests, they received the results as follows. The Yolo-v2 model provided the highest accuracy of 90.98%, 78.68% and 66.55% for the left-right hand, scratch and fingers classification, respectively. For sweat-pore classification, the Resnet-50 model provided the highest accuracy of 91.29%. It is also worth noted that both Yolo-v2 and Resnet-50 took at most 250.37 ms per image.

**e. A CNN-based Framework for Comparison of Fingerprints**

In this paper there is a comparison of 2D fingerprint images with moulded fingerprints ,which offer deformation-free acquisition of fingerprint features. Convolutional neural networks (CNN) have shown remarkable capabilities in biometrics recognition. However, there has been almost nil attempt to match fingerprint images using CNN based approaches. This paper develops a CNN-based framework to accurately match original and moulded fingerprint images. Our framework firstly trains a multi-Siamese CNN using fingerprint minutiae, respective ridge map and specific region of ridge map. This network is used to generate deep fingerprint representation using a distance-aware loss function. Deep fingerprint representations generated in such multi-Siamese network are concatenated for more accurate cross comparison. The proposed approach for cross-fingerprint comparison is evaluated on two publicly available databases containing original fingerprints and respective moulded fingerprints. Their experiments presented in this paper consistently achieve outperforming results, over several popular deep learning architectures.

**f. A Survey on Anti-Spoofing Schemes for Fingerprint Recognition Systems**

The art of attacking a biometric system has gained sophistication over the past several years. One such attack involves the use of fake fingers or spoofs in order to defeat the biometric recognition system. The success of spoof attacks has been demonstrated by several researchers. Artificial fingerprints are usually made of materials which can be scanned by existing commercial fingerprint scanners. Thus, there is a need for developing robust liveness detection or anti-spoofing schemes in order to maintain the integrity of fingerprint recognition systems. we reviewed different types of spoof attacks and discussed the various countermeasures that have been developed in the literature to detect or deflect such attacks. The pros and cons of some of these countermeasures were presented. Databases and performance metrics used to evaluate the efficacy of these countermeasures were also discussed. We then presented methods for combining liveness values with match scores and quality measures. Finally, we discussed some of the open challenges in this field. As fingerprint verification systems become widely used, it is necessary to make them resilient to spoof attacks. The advent of mobile biometrics and remote authentication further reinforces the need to design robust anti-spoofing schemes for fingerprints and other biometric modalities.

g. **Anti-spoofing method for fingerprint recognition using patch based deep learning machine**

   A novel model is proposed. It is based on DRBM and deep Boltzmann machine It deals with complex texture pattern in a strong multi layer model After training a DBM deep feature of prints are extracted KNN classifier is applied The experiment results demonstrate that the Deep learning model is robust against different kinds of spoof forgeries

h. **Fingerprint Liveness Detection using CNN features of random sample patches**

   First patches must be extracted from fingerprint area In learning phase for ex 100 patches are extracted from a image than CNN composed of alternating layers of convolution and pooling is used to train model Than tested in training phase Avg classification error of 3.42% is achieved in results.

## 2. Literature Survey for Detection of gender using Fingerprints

a. **Gender Classification Based on Fingerprint Analysis**

   Gender classification plays an active role in several applications such as biometrics, criminology, surveillance, human computer interaction, and commercial profiling. Though biometric traits such as face, gait, iris and hand shape are used for gender classification in the past, majority of the work is based on face as it contains more prominent features than others. In this paper we have analyzed fingerprints for gender classification with a hope that it has great potential for future research. We have employed a three convolutional layer CNN with rectified linear (ReLu) and tanh activation functions on NIST database which contains a set of 4000 images and achieved 99% accuracy. Performance of the proposed system demonstrated that fingerprints contain vital features to discriminate against the gender of a person.

# REQUIREMENT ANALYSIS

**Visual Studio Code:** Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

**Collab Notebook**: is an excellent open-source web application that allows you to create and share documents that contain live code, equations, visualizations and are used for data cleaning and transformation, numerical simulation, statistical modelling, data visualization and machine learning.

Python Libraries for Exploratory Data Analysis: NumPy, Pandas, Matplotlib, Keras, Tensorflow and CV2.

Python Libraries for machine learning: Scikitlearn is an excellent source for implementing machine learning algorithms.

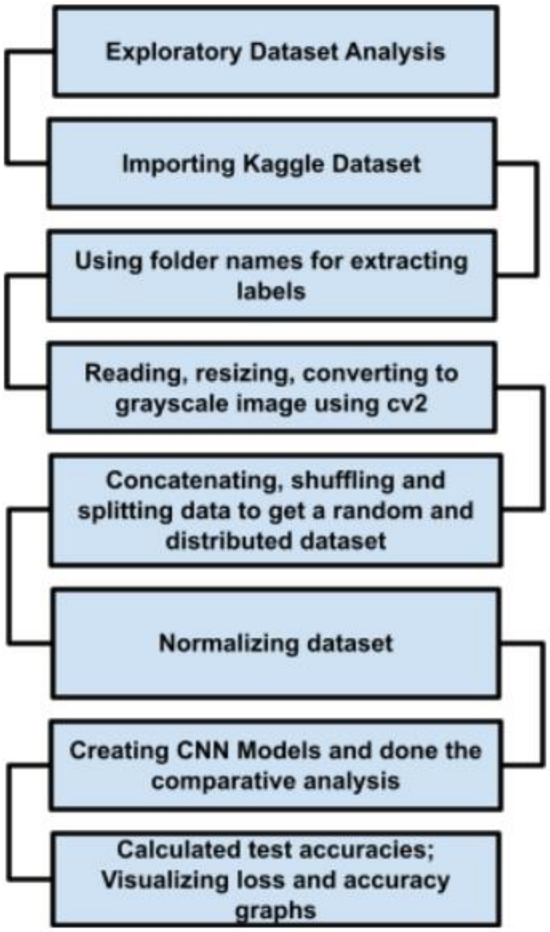
**Flask:** is an API of Python that allows us to build up web-applications.

**Hardware requirements:**

- 32- or 64-bit computer.
- Minimum 3 GB disk space to download and install anaconda.

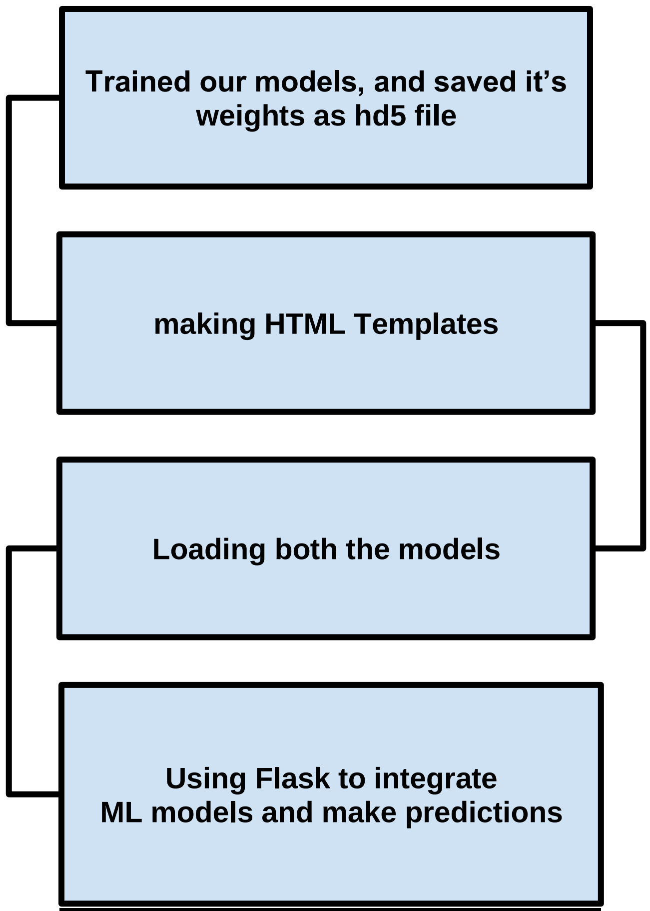
## DETAILED DESIGN

**Workflow for Model:**




**4.a Model Workflow**

**Workflow for flask:**




**4.b Flask Workflow**

# IMPLEMENTATION

## 1. Real or Artificial moulded fingerprints

In this work, we propose a deep learning CNN model for feature extraction and classification. Deep CNN have proven to be efficient in image processing related tasks and therefore are suitable for detecting fingerprints alteration types. We train and evaluate this model on the real and synthetically altered images of the SOCOFing Dataset described above. Each class, including real images, is randomly split into 80% training and 20% testing sub- sets. The images are also resized to 96 x 96.

Convolutional neural networks retain spatial information through filter kernels. In this work we exploit this unique ability of CNN to train a model to classify images from the SOCOFing Dataset into two categories: altered and real where real images are those without any alterations.

The deep CNN model has five convolutional layers with 20 3x3, 40 3x3, 60 3x3, 80 3x3 filter kernels. All convolutional layers use a stride of one and zero padding of size two. Moreover, the output of every convolutional layer is shaped by a rectifier linear unit (ReLU) function. Max pooling is applied to the first three convolutional layers for dimensionality reduction. These convolutional layers are followed by two fully connected layers with 1000 and 100 hidden units respectively. Furthermore, we employ batch normalization to standardize the distribution of each input feature across all the layers.

The deep CNN is trained using stochastic gradient descent (SGD). We trained on min batches of size 70 and set the learning rate, *LR*, to 0.01. *LR* was decayed with a factor of 0.01 according to our base research paper. The loss is defined by a SoftMax operator. The training was done for 100 epochs as further training led to overfitting.

Then we tried to improve accuracy and got an improvement of 0.03 from 99.36% in the research paper model to 99.39% in our own model.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_33 (Conv2D)           (None, 200, 200, 20)      560
_________________________________________________________________
max_pooling2d_24 (MaxPooling (None, 100, 100, 20)      0
_________________________________________________________________
conv2d_34 (Conv2D)           (None, 100, 100, 40)      7240
_________________________________________________________________
max_pooling2d_25 (MaxPooling (None, 50, 50, 40)        0
_________________________________________________________________
conv2d_35 (Conv2D)           (None, 50, 50, 60)        21660
_________________________________________________________________
max_pooling2d_26 (MaxPooling (None, 25, 25, 60)        0
_________________________________________________________________
conv2d_36 (Conv2D)           (None, 25, 25, 80)        43280
_________________________________________________________________
flatten_5 (Flatten)          (None, 50000)             0
_________________________________________________________________
dense_16 (Dense)             (None, 1000)              50001000
_________________________________________________________________
dense_17 (Dense)             (None, 100)               100100
=================================================================
Total params: 50,173,840
Trainable params: 50,173,840
Non-trainable params: 0
```

**5.1.a Fingerprint Classification Model Summary**

## 2. Gender Classification

We train and evaluate this model on the real and synthetically altered images of the SOCOFing Dataset described above. We used file nomenclature to determine the gender of the person. We used images in the altered images as our training dataset and real images for the testing dataset. The images are resized to 96 x 96. We shuffle to get evenly distributed data.

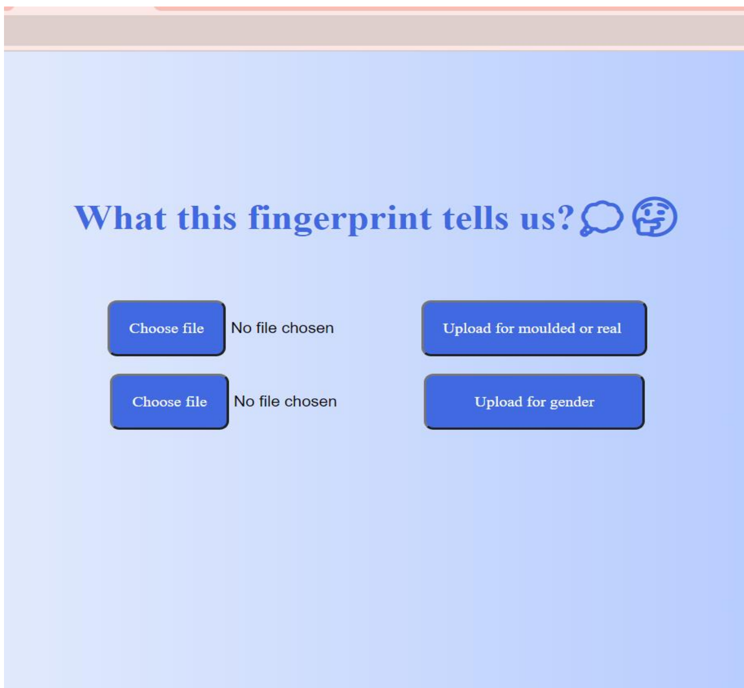
The deep CNN model has four convolutional layers with 20 3x3, 40 3x3, 60 3x3, 80 3x3 filter kernels. All convolutional layers use a stride of one and zero padding of size two. Moreover, the output of every convolutional layer is shaped by a rectifier linear unit (ReLU) function. Max pooling is applied after convolutional layers for dimensionality reduction. These convolutional layers are followed by two fully connected layers with 1000 and 10 hidden units respectively. The deep CNN is trained using adam optimizer and categorical cross-entropy. The training was done for 20 epochs.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 96, 96, 20)        200

 conv2d_1 (Conv2D)           (None, 96, 96, 40)        7240

 conv2d_2 (Conv2D)           (None, 96, 96, 60)        21660

 conv2d_3 (Conv2D)           (None, 96, 96, 80)        43280

 max_pooling2d (MaxPooling2D  (None, 48, 48, 80)       0
 )

 flatten (Flatten)           (None, 184320)            0

 dense (Dense)               (None, 100)               18432100

 dense_1 (Dense)             (None, 10)                1010

 dense_2 (Dense)             (None, 2)                 22

=================================================================
Total params: 18,505,512
Trainable params: 18,505,512
Non-trainable params: 0
```

**5.2.a Gender Classification Model Summary**

## 3. Integration with Flask

We made a home HTML page giving users two options for choosing between checking authentication of fingerprint and gender classification. After the user chooses the file, it goes to predict the HTML file which gives the prediction. While we trained our models, we saved our models weights as hd5 files. Hence, we loaded both the models and predict the result by choosing the maximum value in the array returned by the prediction function




**5.3.a Homepage(index.html)**


**5.3.b Predicted Altered Fingerprint**


**5.3.c Predicted real Fingerprint**


**5.3.d Predicted Female Fingerprint**

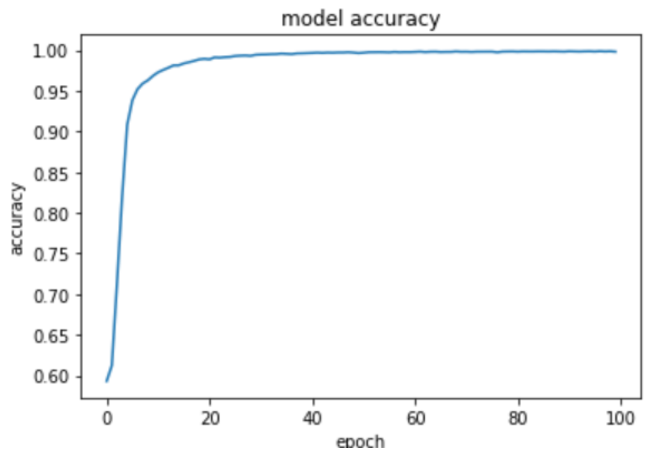

**5.3.e Predicted Male Fingerprint**

# EXPERIMENTAL RESULTS AND ANALYSIS

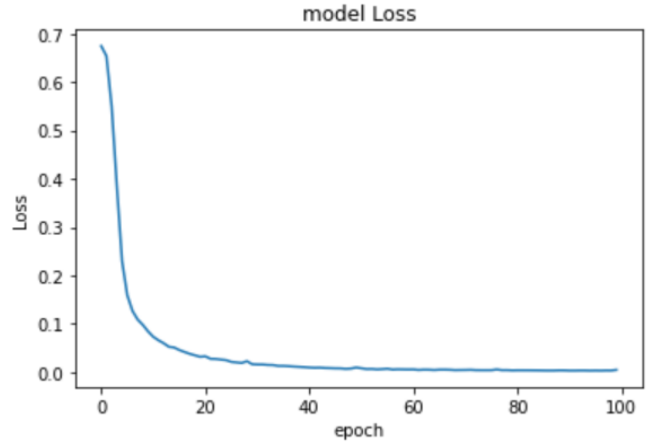
1. Implementation of the model in the research paper.

```
model.evaluate(test_data, test_labels)

346/346 [==============================] - 4s 10ms/step - loss: 0.0290 - accuracy: 0.9937
[0.029039721935987473, 0.9936674237251282]
```

**6.1.a Research Paper Model Evaluation**




**6.1.b Research paper Model Accuracy Graph**

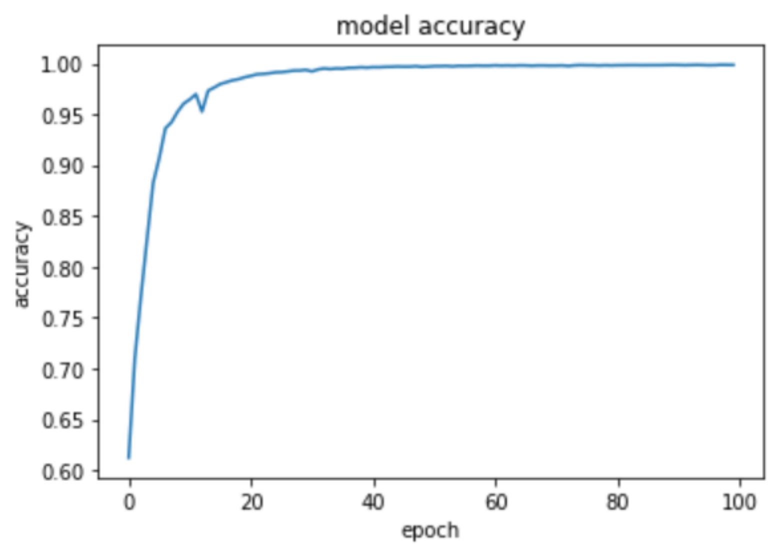



**6.1.c Research Paper Model Loss Graph**

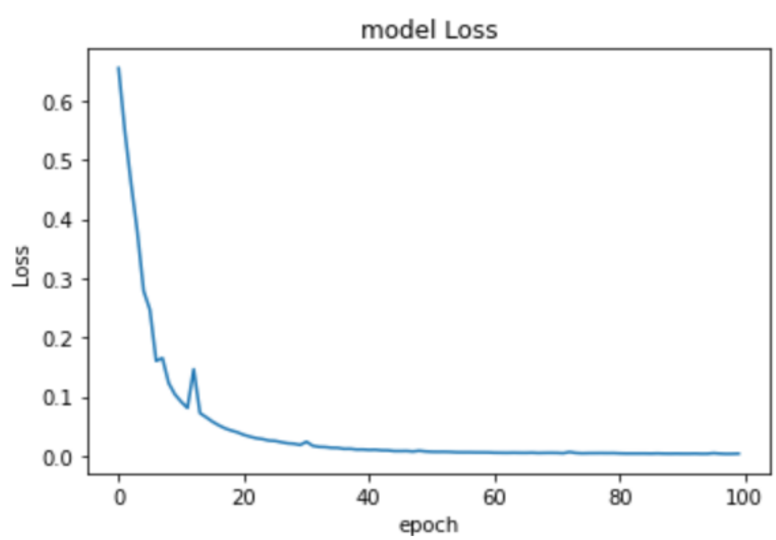
2. Making our own model and trying to improve the accuracy. Although it was not a significant change

```
model.evaluate(test_data, test_labels)

346/346 [==============================] - 4s 12ms/step - loss: 0.0234 - accuracy: 0.9939
[0.023423219099640846, 0.9939388632774353]
```

**6.2.a Fingerprint Classification Model Evaluation**




**6.2.b Fingerprint Classification model Accuracy Graph**

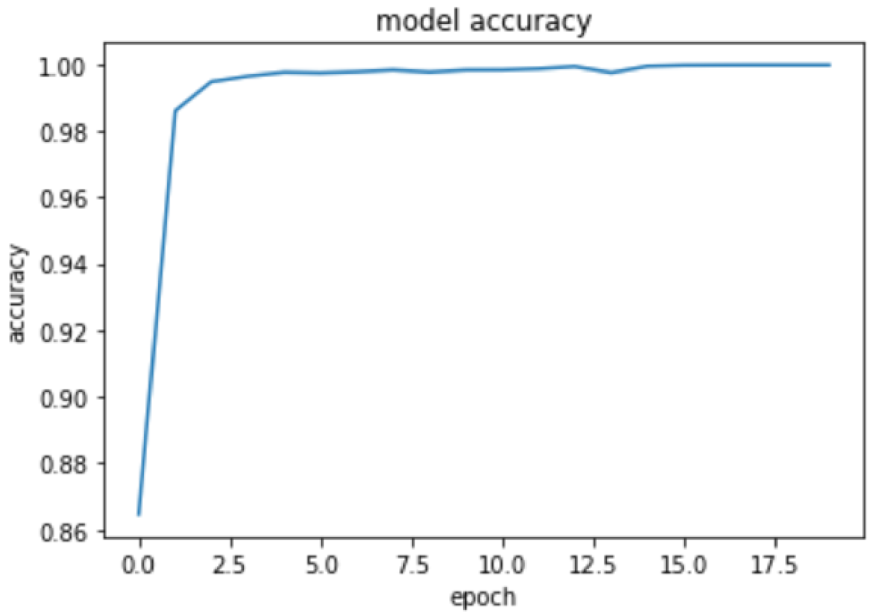



**6.2.c Fingerprint Classification model Loss Graph**

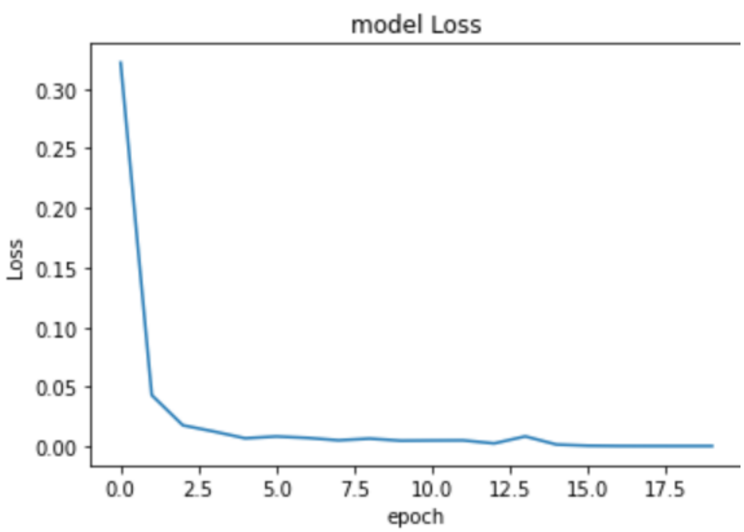
3. Using the same dataset to classify the fingerprints as male or female

```
model.evaluate(test_data, test_labels)

188/188 [==============================] - 8s 39ms/step - loss: 0.0214 - accuracy: 0.9912
[0.02137395553290844, 0.9911666512489319]
```

**6.3.a Gender Classification Model Evaluation**




**6.3.b Gender Classification Model Accuracy Graph**




**6.3. c Gender Classification Model Loss Graph**

# CONCLUSION

Using CNN and the given dataset we trained a model which will be helpful in predicting the authenticating fingerprint and the gender through fingerprint only. The proposed CNN model achieves a testing accuracy rate of 99.39% and the model for gender classification achieves a testing accuracy rate of 99.97%.

Github Link: https://github.com/ads-22/Moulded-Fingerprint-Detection

Google Collab Notebook:

Finger Classification:
https://colab.research.google.com/drive/1euxZykVrixTi5liJuUNAm7TN4zu3BPFg#scrollTo=-Z3mDczBC1NR

Gender Classification:
https://colab.research.google.com/drive/19qRJyVkqXNkvKhmkM6htgm-foImFJx2P?usp=sharing#scrollTo=gyDW_1MIa2U6

# FUTURE SCOPE

From the proposed work we can identify whether the fingerprint used is real or altered and the gender of the person whose fingerprint is used. We can build a software solution for biometric authentication that can tell the difference between a real live fingerprint and an artificially molded one, and throw an alert in case the latter is presented. Similarly, unscrupulous individuals and hostile actors when try to abuse such systems by using fake artificially molded synthetic fingerprints that are very similar to the actual fingerprint, we can prevent frauds. This project in the future with further knowledge and resources can be used for security purposes to avoid unauthorized access and for cybersecurity.

## REFERENCES


- http://irep.ntu.ac.uk/id/eprint/34736/1/12137_James.pdf
- https://link.springer.com/article/10.1007%2Fs11042-020-09314-6
- https://arxiv.org/ftp/arxiv/papers/1807/1807.10609.pdf
- https://www.researchgate.net/profile/Sathesh-Ammayappan/publication/349915135_Evaluation_of_Fingerprint_Liveness_Detection_by_Machine_Learning_Approach_-A_Systematic_View/links/6046ff90a6fdcc9c78219cc5/Evaluation-of-Fingerprint-Liveness-Detection-by-Machine-Learning-Approach-A-Systematic-View.pdf
- https://www.researchgate.net/profile/Mouad-Ali/publication/310953762_Overview_of_Fingerprint_Recognition_System/links/59de2a7aaca27247d7942263/Overview-of-Fingerprint-Recognition-System.pdf
- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.712.7277&rep=rep1&type=pdf
- https://drive.google.com/file/d/1JbE0O57E9o3co4ZljR5EuoVo7HzaCg07/view?usp=sharing
- https://reader.elsevier.com/reader/sd/pii/S2215098619300527?token=B6C14E1E14C851537416084DA969C8607E2BA19DC0E316CB3FB447370F72A0641C9485D31A727F33B0EF47720A820EFC&originRegion=eu-west-1&originCreation=20211025030154
- https://dl.gi.de/bitstream/handle/20.500.12116/1227/321.pdf?sequence=1