

**COLLEGE OF COMPUTING AND INFORMATION SCIENCE**

**BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING YR 2**

**(BSSE II)**

**RECESS PROJECT MODELS:**

**GENDER CLASSIFICATION MODEL**

**FINGERPRINT IDENTIFICATION MODEL**

**FINGERPRINT DIFFERENTIATION**

**06/08/2023**

**GROUP O**

|  |  |  |
| --- | --- | --- |
| **NAME** | **STUDENT No.** | **REGISTRATION No.** |
| SSEKITOOLEKO PETER CLAVER | 2200706919 | 22/U/6919 |
| NAGGITA ETHEL | 2100707022 | 21/U/07022/PS |
| BABIRYE HOPE MUGWERI | 2100707873 | 21/U/07873/PS |
| NAKISANZE DEZIRANTA | 2100706325 | 21/U/06325/PS |
|  |  |  |

Contents

[Abstract: 2](#_Toc142571019)

[Terms and definitions: 2](#_Toc142571020)

[1. Introduction: 3](#_Toc142571021)

[2. Background and Related Work: 4](#_Toc142571022)

[3. Methodology: 5](#_Toc142571023)

[3.1 Gender Classification model 5](#_Toc142571024)

[3.1.1 Model structure 5](#_Toc142571025)

[3.1.2 Model Compilation and Training Setup 6](#_Toc142571026)

[3.1.3 Summary 6](#_Toc142571027)

[3.1.4 Data augmentation 6](#_Toc142571028)

[3.1.5 Oversampling 6](#_Toc142571029)

[3.2 Fingerprint identification model 8](#_Toc142571030)

[3.2.1 Model structure 8](#_Toc142571031)

[3.2.2 Model Compilation 9](#_Toc142571032)

[3.3 Fingerprint differentiation model 9](#_Toc142571033)

[3.3.1 Model Structure 9](#_Toc142571034)

[4. Data Overview: 9](#_Toc142571035)

[4.1 Gender classification model 10](#_Toc142571036)

[4.2 Fingerprint identification model 10](#_Toc142571037)

[4.3 Fingerprint Differentiation Model 11](#_Toc142571038)

[5. Experiments: 11](#_Toc142571039)

[5.1 Gender classification model 11](#_Toc142571040)

[5.2 Fingerprint identification model 13](#_Toc142571041)

[5.3 Fingerprint differentiation model 13](#_Toc142571042)

[6.1 Gender classification model 13](#_Toc142571043)

[6.1.1 Easy Difficulty Level: 13](#_Toc142571044)

[6.1.2 Medium Difficulty Level: 14](#_Toc142571045)

[6.1.3 Hard Difficulty Level: 14](#_Toc142571046)

[5.4 Fingerprint identification model 14](#_Toc142571047)

[5.4.1 Easy Difficulty Level 14](#_Toc142571048)

[5.4.2 Medium Difficulty Level 15](#_Toc142571049)

[6.2.3Hard Difficulty level: 16](#_Toc142571050)

[5.5 Fingerprint differentiation model 16](#_Toc142571051)

[5.5.1 Easy Difficulty level 17](#_Toc142571052)

[5.5.2 Medium Difficulty level 18](#_Toc142571053)

[6.3.3 Hard Difficulty level 18](#_Toc142571054)

[7. Analysis and Discussion: 18](#_Toc142571055)

[a) Initial Model vs. Modified Model: 18](#_Toc142571056)

[b) Easy, Medium, and Hard Difficulty Levels: 18](#_Toc142571057)

[c) Mitigating Imbalance and Overfitting: 19](#_Toc142571058)

[8. Recommendations: 19](#_Toc142571059)

[9. Conclusion: 19](#_Toc142571060)

[10.Appendix 19](#_Toc142571061)

LIST OF FIGURES

[Figure 1 Data representation before oversampling 7](#_Toc142571062)

[Figure 2 Data representation after oversampling 7](#_Toc142571063)

[Figure 3 Bar graph showing number of real and altered fingerprints 10](#_Toc142571064)

[Figure 4 Bar graph showing number of real, medium and hard altered fingerprints 11](#_Toc142571065)

[Figure 5 Training log showing the last training epoch 11](#_Toc142571066)

[Figure 6 Showing Classification report after regularization 12](#_Toc142571067)

[Figure 7 Showing Classification report after augmentation 12](#_Toc142571068)

[Figure 8 Showing Classification report after oversampling 12](#_Toc142571069)

[Figure 9 classification report before employing ResNet NN 13](#_Toc142571070)

[Figure 10 Classification report for easy difficulty level 13](#_Toc142571071)

[Figure 11 Classification report for medium-altered 14](#_Toc142571072)

[Figure 12 Classification report for hard-altered fingerprints 14](#_Toc142571073)

[Figure 13 Classification report for easy-altered fingerprints 14](#_Toc142571074)

[Figure 14 AUC-ROC Curve showing performance of SubjectID model 15](#_Toc142571075)

[Figure 15 CMC Curve showing performance of the SubjectID model 15](#_Toc142571076)

[Figure 16 Classification report for medium altered fingerprints 16](#_Toc142571077)

[Figure 17 Classification report for hard-altered fingerprints 16](#_Toc142571078)

[Figure 18 classification report for easy altered 17](#_Toc142571079)

[Figure 19 line graph showing training and validation losses for easy altered 17](#_Toc142571080)

[Figure 20 line graphs showing training and validation accuracies of easy altered 17](#_Toc142571081)

[Figure 21 classification report for medium-altered 18](#_Toc142571082)

[Figure 22 Classification report for hard altered 18](#_Toc142571083)

# Abstract:

This report presents a comprehensive analysis of a machine-learning project aimed at gender classification using altered fingerprint images, fingerprint identification and a model to differentiate between real and altered fingerprints. The project involves data preprocessing, model training, and evaluation. Various strategies, including regularization, data augmentation, early stopping, dropout and random oversampling, were employed to address challenges posed by class imbalance and overfitting.

# Terms and definitions:

* **Validation Accuracy:**

The accuracy of the model's predictions on a separate validation dataset. It indicates the model's performance on unseen data and its ability to generalize.

* **Validation Loss:**

A measure of the difference between the predicted values and the actual values in the validation dataset. It helps in assessing the model's generalization capability.

* **Training Accuracy:**

The accuracy of the model's predictions on the training dataset. It shows how well the model fits the training data and learns from it.

* **Training Loss:**

A measure of the difference between the predicted values and the actual values in the training dataset. It helps in understanding how well the model is learning the training data.

* **F1 Score:**

A metric that combines both precision and recall. It provides a balanced measure of a model's accuracy by considering both false positives and false negatives.

* **Precision:**

The ratio of correctly predicted positive observations to the total predicted positives. It measures how accurate the positive predictions are.

* **Recall:**

The ratio of correctly predicted positive observations to all observations in actual class. It measures the model's ability to identify all relevant instances.

* **Dataset:**

A collection of data used to train, validate, and test machine learning models. It forms the foundation for training and evaluating the model's performance.

* **Augmentation:**

A technique that involves artificially increasing the diversity of the dataset by applying transformations to the original data, such as rotations, flips, and noise addition.

* **Oversampling:**

A method to address the class imbalance by increasing the number of instances in the minority class. It involves creating synthetic instances to balance class distribution.

* **Preprocessing:**

The steps are taken to prepare and clean the raw data before feeding it into a machine-learning model. This includes tasks like data normalization, resizing, and cleaning.

* **Regularization:**

Techniques are employed to prevent overfitting by adding penalties or constraints to the model's parameters. It helps the model generalize better to unseen data.

* **Overfitting:**

A scenario where a model learns the training data too well, capturing noise and irrelevant patterns. This can lead to poor performance on new data.

* **Deep Learning:**

A subfield of machine learning that involves the use of neural networks with multiple layers. It's particularly effective for complex tasks like image and speech recognition.

* **Neural Networks:**

Computational models inspired by the human brain's structure and function. They consist of interconnected nodes or "neurons" that process and transmit information.

* **Layering:**

Organizing neural networks into sequential layers, each performing specific operations. These layers include input, hidden, and output layers.

* **Early stopping:**

This is the process of stopping the training process before the model starts overfitting by monitoring a specific metric for example validation loss and terminating training when the metric stops improving.

* **Dropout:**

This is a regularization technique where a random subset of units from the neural network is temporarily removed (dropped out) during each training iteration to prevent overfitting.

# Introduction:

We were presented with a fingerprint dataset to perform machine learning and our objectives were to create a gender classification model, an identification model as well as a model to differentiate between real and altered fingerprints. We analyzed the number of real and altered fingerprints. The number of easy, medium and hard altered fingerprints because we needed to know the quantity and kind of data we were dealing with. We developed these models using Convolutional Neural Networks (CNN) because these models are well suited for extracting meaningful features from images and their ability to automatically learn hierarchical features from images among others. We also employed ResNet Neural Networks for our model to differentiate between real and altered fingerprints. This was to improve the performance of our model because ResNet networks addresses the vanishing gradient problem and can be used for deeper networks among others.

Gender classification using fingerprint images is a challenging problem with applications in security and identification systems. This report outlines our approach to tackling this problem using machine learning techniques. We employed some techniques from various fields of machine learning like deep learning, neural networks, and layering. This required usage of several libraries to assist in the study and analysis.

Fingerprint identification is a critical component of modern biometric authentication systems, enabling secure and efficient user verification. In this report, we present a state-of-the-art fingerprint identification model that not only accurately identifies individuals but also provides valuable information about the subject’s identity (ID) and the specific finger used for authentication. This model is built using advanced machine learning techniques, specifically Convolutional Neural Networks (CNNs), which have demonstrated exceptional performance. This model contains two models, one for subject ID identification while the other for finger name identification.

Fingerprint differentiation helps in biometric systems reliable as it helps to distinguish between the genuine(real) and tampered(altered) images

The fingerprint differentiation model created helps to distinguish the altered fingerprints from the real fingerprints with the given dataset.

To make the model work, we use the ResNet to handle the training of our data

In our model will discuss the dataset used for training and validation, the ResNets model for our training, and the evaluation metrics used to assess the performance of our model

# 2. Background and Related Work:

The use of fingerprint recognition for gender classification has gained attention due to its potential applications. These applications include biometric security systems and biometric user authentication systems. With the addition of the gender classification models to the fingerprint recognition models results show a better performance of the overall model in terms of F1 score, recall, and accuracy. Previous studies have focused on addressing class imbalance and overfitting, which are common challenges in such tasks.

Fingerprint identification has emerged as a fundamental pillar in the field of biometric authentication, offering a unique and reliable method for verifying individual identity. The distinct ridge patterns and minutiae points found in fingerprints have been harnessed for decades to enable secure access control, criminal investigation and various identity verification applications. Such models are related to traditional minutiae-based approaches, Siamese networks for similarity learning, transfer learning, fingernail detection and classification as well as performance evaluation metrics used like F1 score, accuracy, precision, recall derived from the confusion matrix.

Fingerprint differentiation is reliable for identifying of individuals since the there is a raising concern about security and integrity of biometric systems. This has raised concerns of coming uo with ways of differentiating between the real and altered fingerprints to make fingerprint authentication reliable.

With the rise of deep learning, convolutional neutral networks and was used to handle the challenge, but ResNets emerged as the powerful architecture due to its ability to effectively handle deep networks and capture the intricate features

# 3. Methodology:

## 3.1 Gender Classification model

### 3.1.1 Model structure

Our approach involved the implementation of a Convolutional Neural Network (CNN) architecture for feature extraction from fingerprint images. To combat overfitting, we incorporated L2 regularization for Dense layers and Dropout regularization for Convolutional layers for the

1. **Input Layer:**
   1. Conv2D Layer: 32 filters, kernel size 3x3, 'same' padding, ReLU activation, and 'he\_uniform' kernel initializer.
   2. Input Shape: (96, 96, 1) - representing a grayscale image with a resolution of 96x96 pixels.
2. **MaxPooling2D Layer:**
   1. Pooling size: 2x2.
   2. Dropout Layer: Dropout rate of 0.25.
   3. Purpose: Introduces pooling to downsample the spatial dimensions and reduce computational complexity. Dropout helps mitigate overfitting.
3. **Convolutional Layer:**
   1. 32 filters, kernel size 3x3, 'same' padding, ReLU activation, and 'he\_uniform' kernel initializer.
   2. Purpose: Extract complex features from the downsampled representation.
4. **MaxPooling2D Layer:**
   1. Pooling size: 2x2.
   2. Dropout Layer: Dropout rate of 0.25.
   3. Purpose: Further downsample the feature maps while introducing regularization through dropout.
5. **Flatten Layer:**
   1. Converts the 2D feature maps into a 1D feature vector, preparing for fully connected layers.
6. **Dense Layer:**
   1. 128 units, ReLU activation, 'he\_uniform' kernel initializer.
   2. L2 Regularization: Regularizes the layer with L2 regularization strength of 0.01.
   3. Purpose: Adds a dense, fully connected layer with L2 regularization to prevent overfitting and improve generalization.
7. **Dense Layer:**
   1. 1 unit with sigmoid activation.
   2. Purpose: The final layer produces a binary classification output, indicating whether the fingerprint image is male or female.

### 3.1.2 Model Compilation and Training Setup

* After defining the architecture, the model is compiled using the Adam optimizer with a learning rate of 1e-3 and binary cross-entropy loss function. The chosen optimizer is known for its efficiency in training deep neural networks.
* To prevent overfitting and ensure the model generalizes well, an Early Stopping callback is employed. This callback monitors the validation loss and halts training if the loss does not improve over a certain number of epochs (patience set to 10).

### 3.1.3 Summary

* The model aims to classify fingerprint images into male or female categories.
* The model architecture incorporates convolutional layers to learn hierarchical features from fingerprint images. MaxPooling layers downsample the feature maps, and dropout layers introduce regularization to mitigate overfitting. A fully connected layer with L2 regularization ensures better generalization.
* This architecture's combination of convolutional, pooling, and regularization layers creates a balance between capturing intricate features and preventing overfitting. The subsequent compilation and training setup enhance the model's convergence and generalization abilities.

### 3.1.4 Data augmentation

We also employed data augmentation to solve the issue of class imbalance as the female fingerprints were significantly lower than the male fingerprints.

### 3.1.5 Oversampling

We implemented the Random Oversampling technique during training. This approach rebalanced the gender distribution within each difficulty level, enhancing the model's ability to learn from both genders.

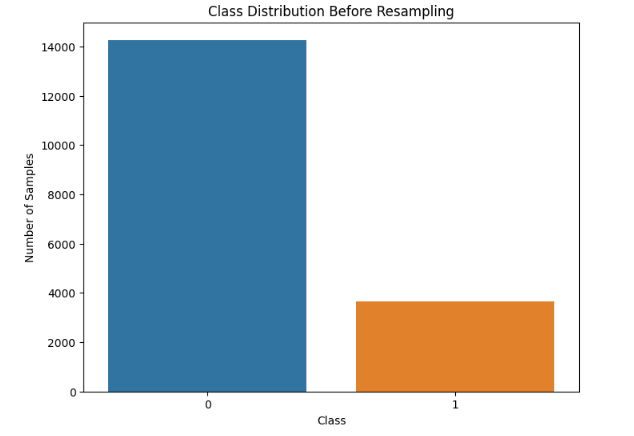


Figure 1 Data representation before oversampling

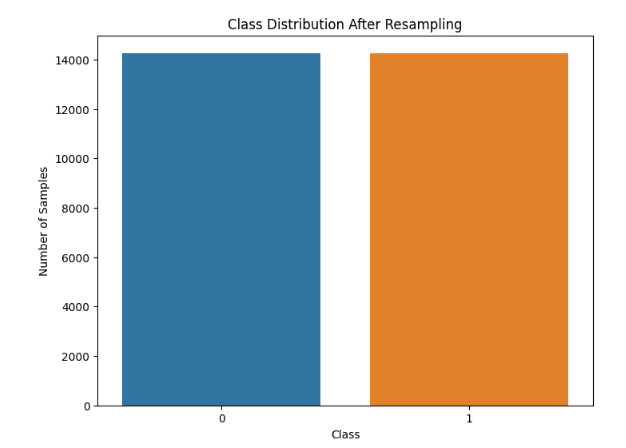


Figure 2 Data representation after oversampling

## 3.2 Fingerprint identification model

### 3.2.1 Model structure

This model defines two convolutional neural network (CNN) models using the Keras API with various techniques to prevent overfitting, including L2 regularization, batch normalization, and dropout. Here's a breakdown of the model structure for both "SubjectID\_Mod" and "FingerNum\_Mod" models.

1. **Common structure**
2. Input Shape: (96, 96, 1) - This means each input image has a resolution of 96x96 pixels with a single channel (grayscale).
3. Regularization: L2 regularization with a coefficient of 0.001 is applied to the kernel weights of convolutional layers.
4. Batch Normalization: Applied after each convolutional layer to stabilize and accelerate training by normalizing the activations.
5. MaxPooling: Max pooling layers with a pool size of (2, 2) are used to downsample the spatial dimensions of the feature maps.
6. Dropout: Dropout with a rate of 0.3 is applied after the last max pooling layer to randomly deactivate neurons during training, preventing overfitting.
7. Model “SubjectID\_Mod”
8. Convolutional Layer: 32 filters with a kernel size of (5, 5) and ReLU activation.
9. MaxPooling Layer: (2, 2) pool size.
10. Convolutional Layer: 64 filters with a kernel size of (5, 5) and ReLU activation.
11. MaxPooling Layer: (2, 2) pool size.
12. Convolutional Layer: 128 filters with a kernel size of (3, 3) and ReLU activation.
13. MaxPooling Layer: (2, 2) pool size.
14. Dropout Layer: Dropout with a rate of 0.3.
15. Flatten Layer: Flattens the output to a 1D vector.
16. Dense Layer: 256 units with ReLU activation.
17. Dropout Layer: Dropout with a rate of 0.4.
18. Dense Layer: Output layer with the number of units corresponding to the number of classes in the "SubjectID\_Mod" task (600), using softmax activation.
19. Model “FingerNum\_Mod”
20. Convolutional Layer: 32 filters with a kernel size of (5, 5) and ReLU activation.
21. MaxPooling Layer: (2, 2) pool size.
22. Convolutional Layer: 64 filters with a kernel size of (5, 5) and ReLU activation.
23. MaxPooling Layer: (2, 2) pool size.
24. Convolutional Layer: 128 filters with a kernel size of (3, 3) and ReLU activation.
25. MaxPooling Layer: (2, 2) pool size.
26. Dropout Layer: Dropout with a rate of 0.3.
27. Flatten Layer: Flattens the output to a 1D vector.
28. Dense Layer: 256 units with ReLU activation.
29. Dropout Layer: Dropout with a rate of 0.4.
30. Dense Layer: Output layer with the number of units corresponding to the number of classes in the "FingerNum\_Mod" task (10), using softmax activation.

### 3.2.2 Model Compilation

The model is then compiled with Adam optimizer with a learning rate of 0.0001 and categorical cross entropy loss suitable for multi class classification tasks.

## 3.3 Fingerprint differentiation model

### 3.3.1 Model Structure

We begun by training the model using CNN but due to its poor performance, we resorted to using ResNets and it provided better results.

1. Conv2D: layer with 32 filters, 3x3 kernel size, ReLU activation, and ‘same’ padding.
2. BatchNormalization: layer of normalization.
3. MaxPooling2D: layer with a 2x2 pool size.
4. Dropout: layer for regularization.
5. Multiple instances of “resnet\_block” function: includes residual connections.
6. GlobalAveragePooling2D: layer to reduce spatial dimensions.
7. Dense layer: 128 units, ReLU activation and L2 regularization.
8. Dense layer: 1 unit and sigmoid activation.

# 4. Data Overview:

The dataset used was a folder containing real and altered fingerprint images of 600 subjects both male and female. The altered fingerprint subfolder consisted of easy, medium and hard altered fingerprints. Each subject had all their fingerprints recorded for all fingers. Each fingerprint was named with the subject ID, gender, the hand (left/right) and the finger name.

The training dataset consisted of altered fingerprint images categorized into three difficulty levels: easy, medium, and hard.

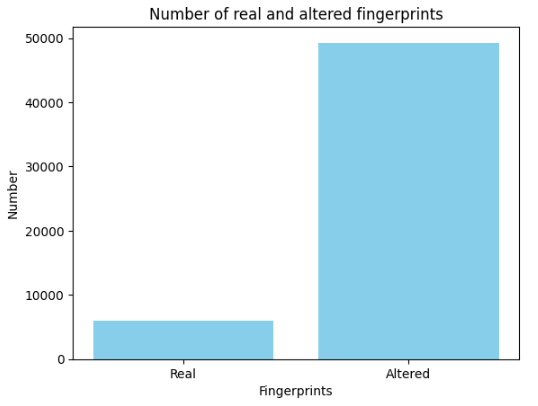


Figure 3 Bar graph showing number of real and altered fingerprints

## 4.1 Gender classification model

Each level exhibited a significant gender imbalance, with more male samples compared to females.

* Easy-altered Dataset: 3665 F, 14266 M
* Medium-altered Dataset: 3426 F, 13641 M
* Hard-altered Dataset: 2746 F, 11526 M

The testing dataset consisted of real fingerprint images which also exhibited a significant gender imbalance

* Test Dataset (Real Fingerprints): 1230 F, 4770 M

***Note:*** *unlike the training dataset, the testing dataset wasn’t exposed to random oversampling or data augmentation.*

## 4.2 Fingerprint identification model

The models all exhibited different accuracies in identifying real fingerprints using easy, medium and hard fingerprints for training. The easy, altered and hard fingerprints had different numbers with hard fingerprints being significantly less than the other two categories.

* Easy-altered Dataset: 17931
* Medium-altered Dataset: 17067
* Hard-altered Dataset: 14272

Testing data consisted of real fingerprints.

* Real Dataset: 6000

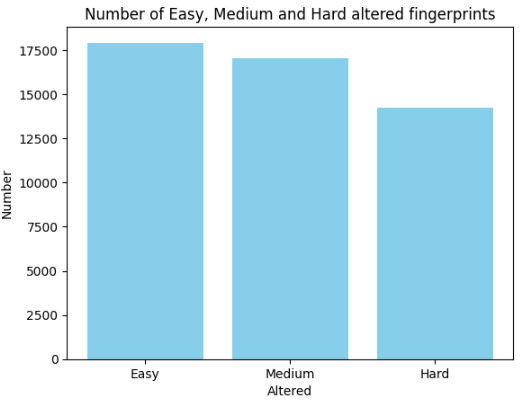


Figure 4 Bar graph showing number of real, medium and hard altered fingerprints

## 4.3 Fingerprint Differentiation Model

We used a dataset containing real and altered images.

We worked on one kind of altered images with the real images, and we split the altered type image folder and combined with the real to obtain the training and validation data.

For Altered vs Real, we used 8534 real samples and 8534 altered easy samples

# Experiments:

## 5.1 Gender classification model

Our model underwent training on all difficulty levels. The model first underwent training without the regularization techniques with the easily altered dataset where upon analysis of the training logs we noticed a very high training accuracy and fluctuating validation and training loss which we concluded that there must be overfitting.



Figure 5 Training log showing the last training epoch

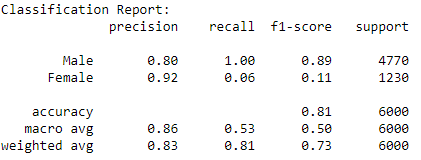
We employed L2 regularization for Dense layers and Dropout regularization for Convolutional layers to mitigate overfitting to make the model more robust as visualized in Figure 1. We then generated a report to show the model’s performance.  


Figure 6 Showing Classification report after regularization

The report showed a struggle by our model to correctly identify female fingerprints so we first employed data augmentation as a means to solve the issue at hand and generated the report below.

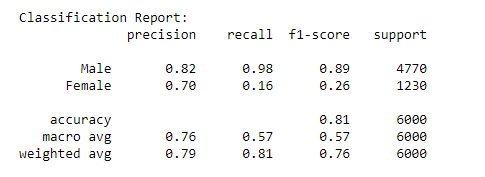


Figure 7 Showing Classification report after augmentation

The report showed that the model hadn’t improved in the direction of detecting female fingerprints correctly. Upon that conclusion, we went ahead to try random oversampling and we generated this report.

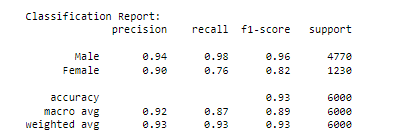


Figure 8 Showing Classification report after oversampling

The results showed a great improvement in our model’s overall performance and more so being able to correctly distinguish and identify the female fingerprints.

## Fingerprint identification model

With the overfitting techniques employed, the model performed very well with easy-altered data according to all the evaluation metrics used. The validation accuracy was high and close to the training accuracy, also the training loss kept on reducing.

The medium-altered dataset also performed in a similar way but with a little reduced accuracy. We therefore did not employ hyperparameter tunning.

The hard-altered dataset performed more poorly than the easy and medium altered because of the limited training data and maybe it’s level of difficulty.

## Fingerprint differentiation model

The initial CNN model performed poorly even after employing techniques like data splitting and over sampling to solve the imbalance between real and easy-altered fingerprints.

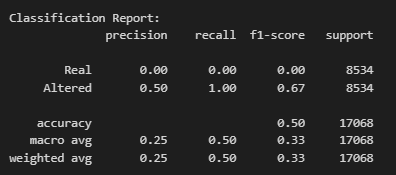


Figure 9 classification report before employing ResNet NN

To solve the above problem, we employed ResNet Neural Networks which improved our model performance.

## 6.1 Gender classification model

### 6.1.1 Easy Difficulty Level:

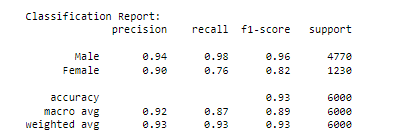


Figure 10 Classification report for easy difficulty level

### 6.1.2 Medium Difficulty Level:

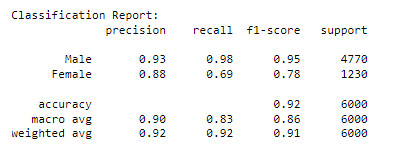


Figure 11 Classification report for medium-altered

### 6.1.3 Hard Difficulty Level:

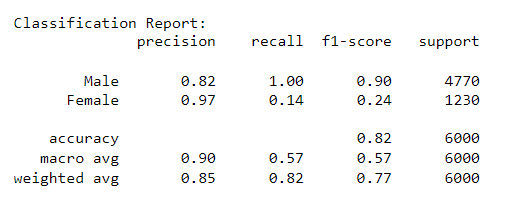


Figure 12 Classification report for hard-altered fingerprints

## Fingerprint identification model

* + 1. Easy Difficulty Level:

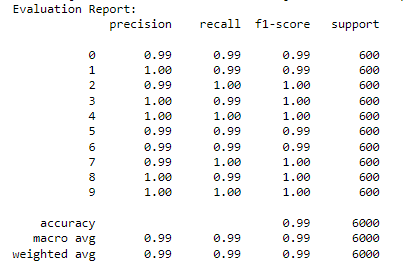


Figure 13 Classification report for easy-altered fingerprints

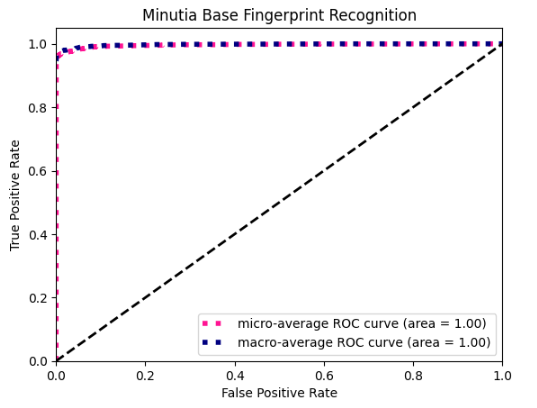


Figure 14 AUC-ROC Curve showing performance of SubjectID model

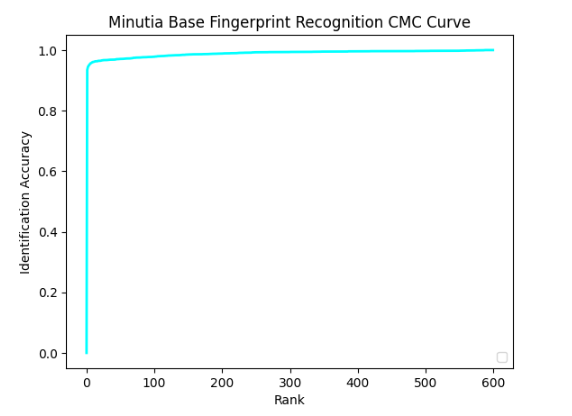


Figure 15 CMC Curve showing performance of the SubjectID model

* + 1. Medium Difficulty Level:

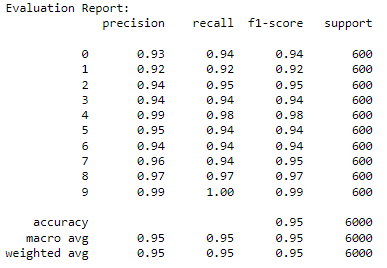


Figure 16 Classification report for medium altered fingerprints

### 6.2.3Hard Difficulty level:

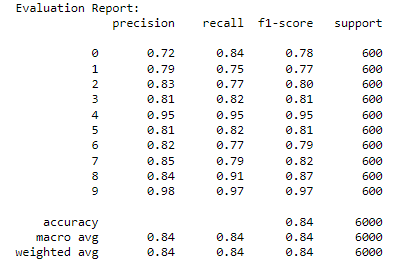


Figure 17 Classification report for hard-altered fingerprints

## Fingerprint differentiation model

### Easy Difficulty level

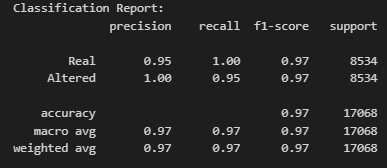


Figure 18 classification report for easy altered

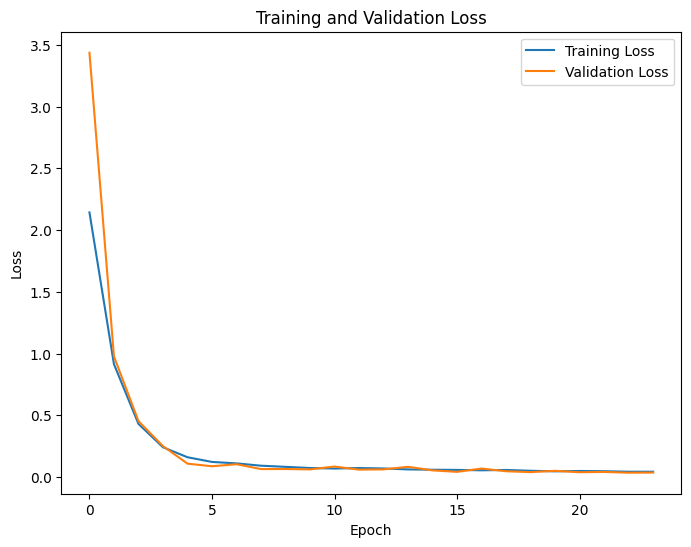


Figure 19 line graph showing training and validation losses for easy altered



Figure 20 line graphs showing training and validation accuracies of easy altered

### Medium Difficulty level

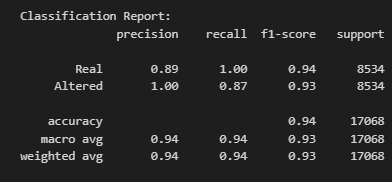


Figure 21 classification report for medium-altered

### 6.3.3 Hard Difficulty level

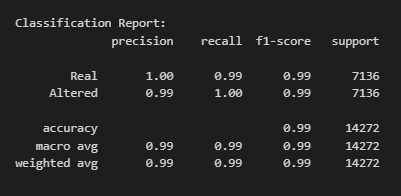


Figure 22 Classification report for hard altered

# 7. Analysis and Discussion:

## Initial Model vs. Modified Model:

* The initial model suffered from overfitting, as evidenced by a widening gap between training and validation accuracy for the gender classification model.
* The modified model, with regularization techniques, demonstrated improved generalization, as indicated by narrowed accuracy discrepancies.
* The ResNet Neural Network greatly improved the fingerprint differentiation model unlike the original CNN model.

## Easy, Medium, and Hard Difficulty Levels:

1. Gender classification model

* The model excelled in distinguishing genders in easy and medium difficulty levels, showcasing balanced precision and recall.
* We still notice a small fall in the precision recall and F1 score which is attributed to the greater distortion in the medium-level fingerprints compared to the altered easy fingerprints.
* Challenges emerged in hard cases, particularly in accurately classifying female fingerprints as we see a significant fall in the F1 score and recall.
* This was partially attributed to the reduction in the training data at the altered hard level (from 17,000-14,000).  
  It is also attributed to the greater distortion of the fingerprints at the hard level compared to the medium level.

1. Fingerprint identification model

The models portrayed different accuracies at different difficulty levels with easy having the highest accuracy, followed by medium then hard.

1. Fingerprint differentiation model

The model portrayed the highest accuracy in differentiating hard-altered fingerprints, followed by medium then easy-altered fingerprints.

## Mitigating Imbalance and Overfitting:

* Random oversampling successfully balanced the gender distribution during training, mitigating the effects of class imbalance for the gender classification model.
* L2 regularization for Dense layers and Dropout regularization for Convolutional layers curbed overfitting, yielding better generalization for the above model.
* L2 regularization, drop out regularization, early stopping and batch normalization were deployed to curb overfitting for both fingerprint identification models at all levels.
* Data splitting, over sampling were employed to solve class imbalance for the fingerprint differentiation model. These were employed in the CNN model but did not perform well. ResNet Neural Networks were then employed which improved model performance.

# 8. Recommendations:

* Further fine-tuning and feature engineering could be explored to improve performance in hard difficulty levels.
* Analyzing misclassified examples from challenging cases can provide insights into specific patterns causing difficulties.
* Considering advanced techniques such as Synthetic Minority Over-sampling Technique (SMOTE) to tackle class imbalance more effectively.

# 9. Conclusion:

Our models’ performance in easy and medium difficulty levels is promising, demonstrating its ability to learn features effectively. However, challenges persist in correctly classifying fingerprints with hard-altered dataset for both gender classification and identification model. Iterative refinement and exploration of advanced techniques are recommended to bridge this gap.

# 10.Appendix

* **NumPy** (**import numpy as np**): NumPy is a popular library for numerical computing in Python. It provides support for multi-dimensional arrays and mathematical functions to operate on these arrays efficiently.
* **matplotlib.pyplot** (**import matplotlib.pyplot as plt**): Matplotlib is a widely used library for creating visualizations and plots in Python. The **pyplot** module within Matplotlib provides a simple interface to create various types of plots.
* **keras** and **tensorflow** (**import keras** and **import tensorflow as tf**): Keras is a high-level neural networks API written in Python. It allows easy and fast prototyping for building and training deep learning models. TensorFlow, on the other hand, is a popular deep-learning framework developed by Google. Keras is often integrated with TensorFlow to leverage its computational capabilities.
* **seaborn** (**import seaborn as sns**): Seaborn is a Python data visualization library based on Matplotlib. It provides an interface for creating more attractive and informative statistical graphics.
* **layers** and **Model** from **keras**: These are specific components from Keras used to define and build neural network architectures.
* **shuffle** from **sklearn.utils** (**from sklearn.utils import shuffle**): **shuffle** is a utility function from scikit-learn (sklearn) used for randomly shuffling data.
* **train\_test\_split** from **sklearn.model\_selection** (**from sklearn.model\_selection import train\_test\_split**): This function from scikit-learn is used to split the data into training and testing sets for model evaluation.
* **iaa** from **imgaug** (**from imgaug import augmenters as iaa**): ImgAug is a library for image augmentation, which is a technique used to increase the diversity of the training data by applying random transformations to the images.
* **os** (**import os**): The **os** module provides a way to interact with the operating system. It is commonly used to handle file and directory operations.
* **cv2** (**import cv2**): OpenCV (Open Source Computer Vision) is a popular computer vision library that provides tools for image and video processing.
* **random**: The **random** module is used for generating random numbers or making random selections.
* **from tensorflow.keras import Sequential**: This imports the **Sequential** class from TensorFlow's Keras API. **Sequential** is a linear stack of layers that can be used to build neural network models.
* **from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, BatchNormalization, Dropout**: These are various types of layers that will be used to build the CNN model.
* **from tensorflow.keras import layers**: This imports additional layers from TensorFlow's Keras API.
* **from tensorflow.keras import optimizers**: This imports the optimizers module from TensorFlow's Keras API, which contains different optimization algorithms like Adam, SGD, etc.
* **from sklearn.metrics import confusion\_matrix**: This imports the **confusion\_matrix** function from scikit-learn. The confusion matrix is used to evaluate the model's performance in a classification problem.