COMPUTER VISION PROJECT

# SOKOTO COVENTRY

## ENHANCING FINGERPRINT RECOGNITION

*LEVERAGING SYNTHETIC ALTERATIONS FOR ROBUST ALGORITHMS*

NAJEEB
GODCARES
ANKIT

# WHY IS IT EVEN NEEDED?

1. Fingerprint recognition technology has numerous **real-world applications, including access control, identity verification, law enforcement, and forensic analysis.** By developing more accurate and reliable fingerprint recognition systems, we can enhance security, streamline authentication processes, and improve overall efficiency in various domains.
2. Biometric authentication, such as fingerprint recognition, **offers a higher level of security compared to traditional methods like passwords or PINs.** However, to ensure the effectiveness and reliability of biometric systems, it is crucial to thoroughly evaluate them using diverse and comprehensive datasets like SOCOFing.This helps identify potential vulnerabilities and ensure that the systems are resistant to spoofing attacks and other forms of manipulation.
3. Research and Development: Datasets like SOCOFing **provide researchers and developers with a standardized and well-documented resource for developing, testing, and evaluating algorithms and models** related to fingerprint recognition and biometric security. By having access to a diverse range of fingerprint images with known attributes and alterations, researchers can explore various techniques and methodologies to improve the accuracy and robustness of fingerprint recognition systems.
4. **Benchmarking**: Having a benchmark dataset like SOCOFing **allows researchers to compare the performance of different algorithms and approaches** in a consistent and fair manner. This enables the community to track progress over time and identify areas where further improvements are needed.

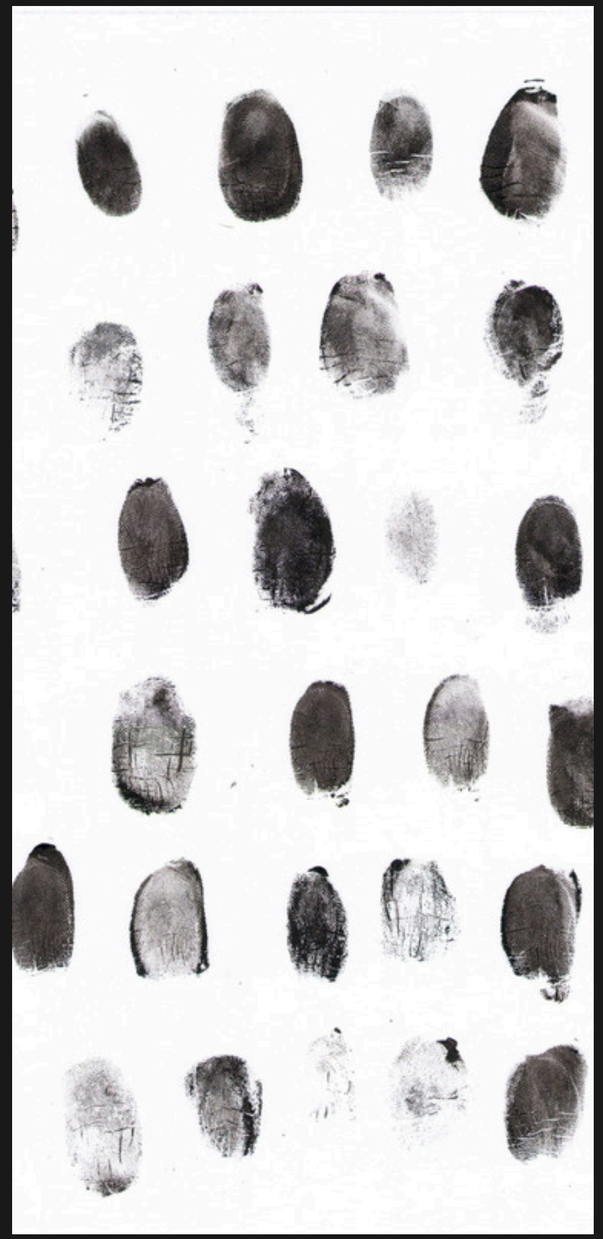
# NOT CONVINCED?

01


**Spoofing Attacks**: In spoofing attacks, adversaries attempt to deceive fingerprint recognition systems by presenting altered or synthetic fingerprints, such as artificial fingers made from molds or lifted prints from surfaces. These attacks exploit vulnerabilities in the system's ability to distinguish between genuine and fake fingerprints.

02

**Adversarial Examples**: Adversarial examples are inputs to machine learning models that are intentionally crafted to cause misclassification or errors. In the context of fingerprint recognition, adversaries may generate synthetic alterations that exploit weaknesses in the underlying algorithms, leading to incorrect identification or authentication.



# DATASET OVERVIEW

- Subjects: 600 African subjects
- Fingerprints per Subject: 10
- Demographic: Subjects aged 18 years or older
- Attributes: Gender, hand, and finger name labels

# SYNTHETIC ALTERATIONS

- Types: Obliteration, Central Rotation, Z-cut
- Levels: Easy, Medium, Hard
- Tool: STRANGE toolbox
- Total Altered Images: 49,273

# SYNTHETIC ALTERATIONS

## Obliteration

obliteration can involve obscuring or blurring parts of the fingerprint pattern, making it difficult for traditional fingerprint recognition algorithms to accurately match or identify the fingerprint.

## Central Rotation

Central rotation involves rotating the central part of the fingerprint image around a specific point.

## Z-cut

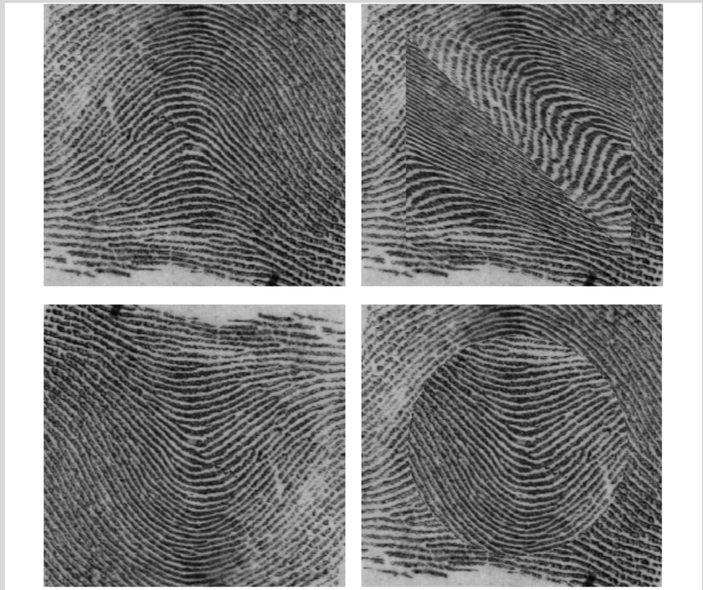


Figure 3. An original fingerprint and its altered versions: 'Z' cut.

Z-cut alteration involves creating a vertical cut through the center of the fingerprint image and displacing one side of the cut relative to the other.


# DATA COLLECTION

- Sensors: Hamster plus (HSDU03PTM) and SecuGen SDU03PTM sensor scanners
- Image Resolution: 1 × 96 × 103 (gray × width × height)

# NOMENCLATURE

- Example Filename: "001_M_Left_little_finger_Obl.bmp"
- 001: Subject Number
- M: Gender (Male)
- Left: Hand
- Little: Finger Name
- Obl: Alteration Type (Obliteration)
- .bmp: File Extension

# DATASET

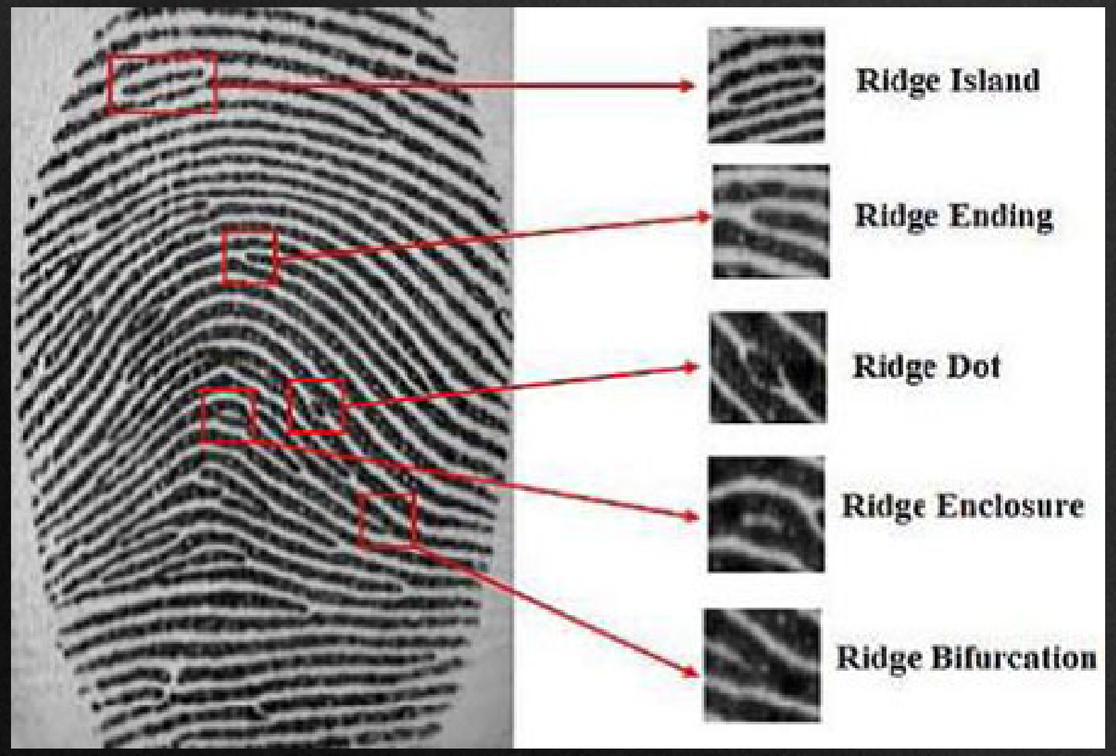
Real

Altered-Easy

Altered-Medium

Altered-Hard

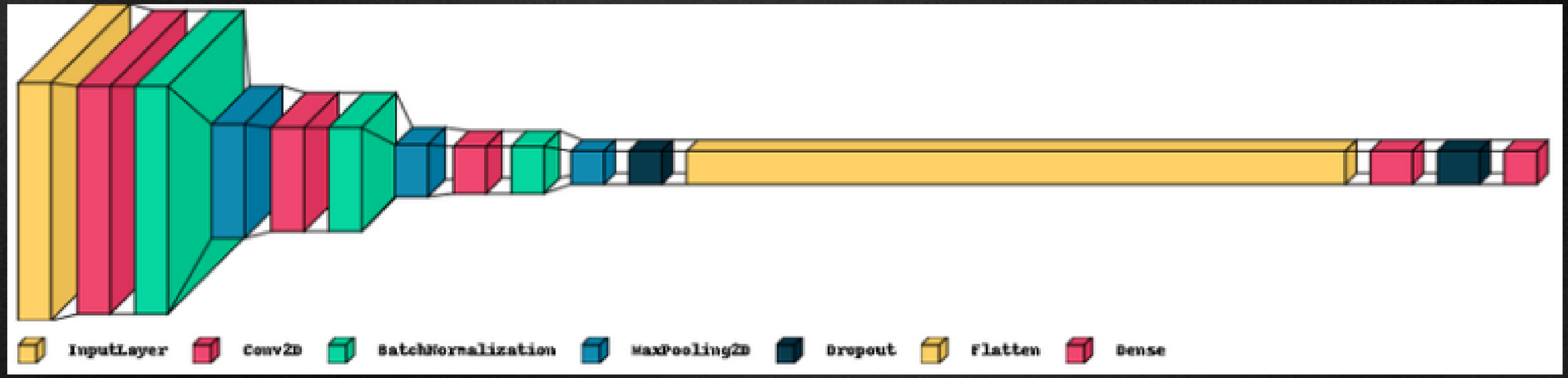
# FINGER MINUTIAE




**Minutiae characteristics**

| Minutiae | example | minutiae | example |
|---|---|---|---|
| ridge ending | | bridge | |
| bifurcation | | double bifurcation | |
| dot | | trifurcation | |
| island (short ridge) | | opposed bifurcations | |
| lake (enclosure) | | ridge crossing | |
| hook (spur) | | opposed bifurcation/ridge ending | |

# ARCHITECTURE



# ARCHITECTURE

```
# FIRST CONVOLUTIONAL BLOCK
X = LAYERS.CONV2D(32, 5, ACTIVATION='RELU')(INPUTS)
X = LAYERS.BATCHNORMALIZATION()(X)
X = LAYERS.MAXPOOLING2D()(X)

# SECOND CONVOLUTIONAL BLOCK
X = LAYERS.CONV2D(64, 5, ACTIVATION='RELU')(X)
X = LAYERS.BATCHNORMALIZATION()(X)
X = LAYERS.MAXPOOLING2D()(X)

# THIRD CONVOLUTIONAL BLOCK
X = LAYERS.CONV2D(128, 3, ACTIVATION='RELU')(X)
X = LAYERS.BATCHNORMALIZATION()(X)
X = LAYERS.MAXPOOLING2D()(X)
X = LAYERS.DROPOUT(0.25)(X)

# FLATTEN AND DENSE LAYERS
X = LAYERS.FLATTEN()(X)
X = LAYERS.DENSE(256, ACTIVATION='RELU')(X)
X = LAYERS.DROPOUT(0.5)(X)
OUTPUTS = LAYERS.DENSE(NUM_CLASSES, ACTIVATION='SOFTMAX')(X)
```

## CONVOLUTION BLOCKS

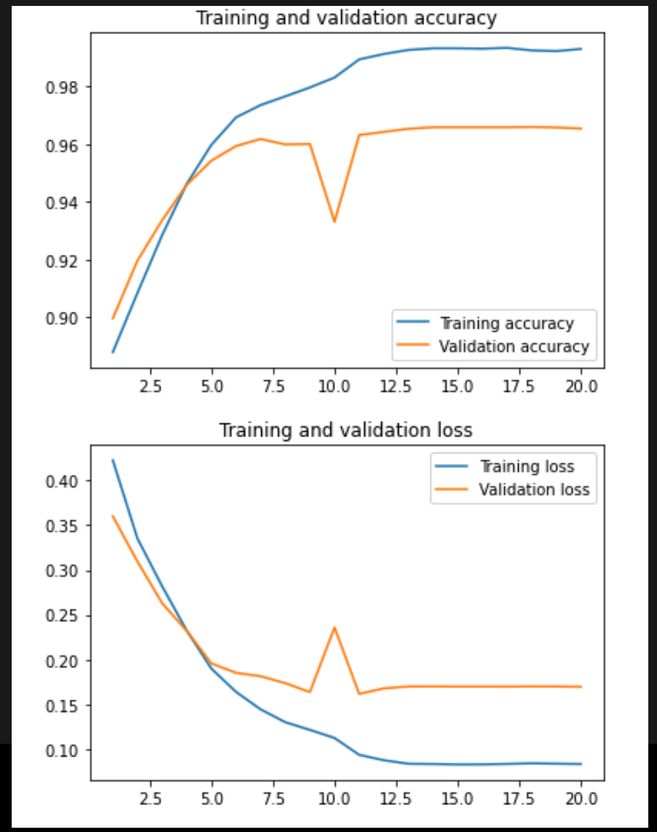
**Hierarchical Feature Learning**
**Extracting Richer set of features at deeper layers**
**Capacity to Capture Variation**

## BATCH NORMALIZATION

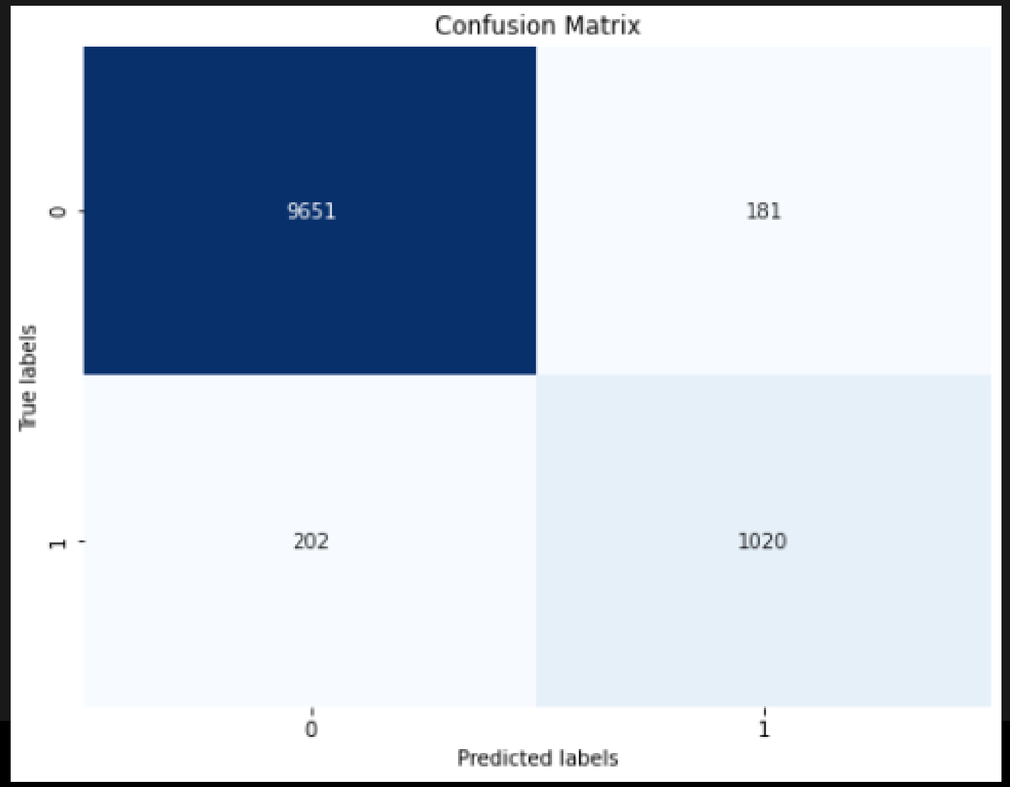
**speeds up training**
**allows suboptimal starts;**
**makes initial weight less important**
**act as a regularizer**

# RESULTS

Real and Altered Classification Accuracy: 96.53518795967102 %



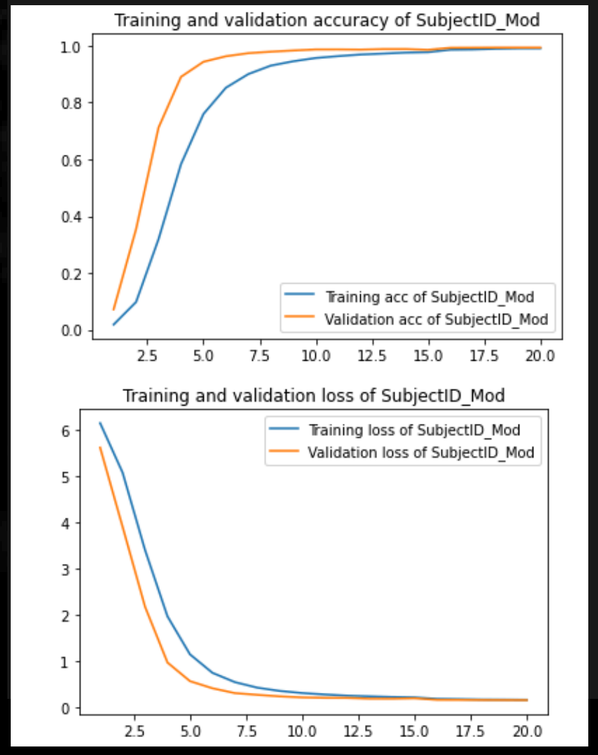

Finger Num Model - Confusion Matrix



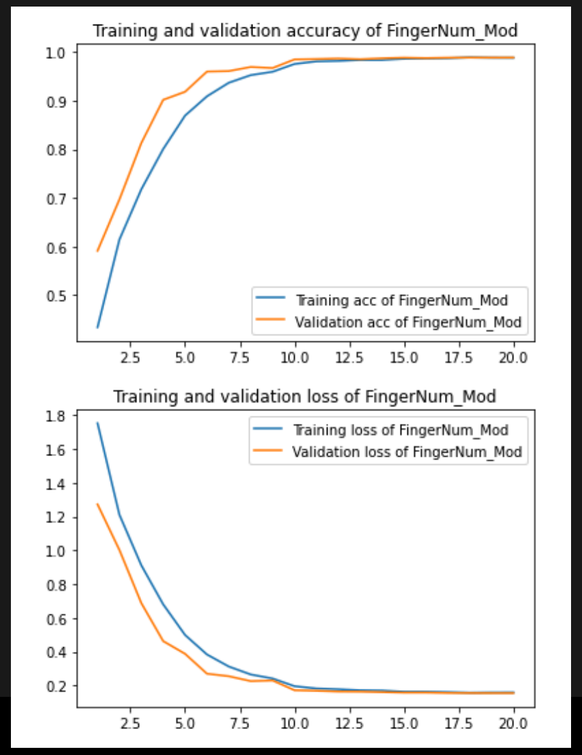
| Dataset | Feature Shape | Label Shape | Description |
|---|---|---|---|
| SubjectID (full) | (49270, 96, 96, 1) | (49270, 600) | 49270 grayscale images (96x96) with labels indicating 600 subjects |
| SubjectID (Train) | (39416, 96, 96, 1) | (39416, 600) | Training set for SubjectID classification |
| SubjectID (Validation) | (9854, 96, 96, 1) | (9854, 600) | Validation set for SubjectID classification |
| SubjectID (Test) | (6000, 96, 96, 1) | (6000, 600) | Test set for SubjectID classification |
| fingerNum (full) | (49270, 96, 96, 1) | (49270, 10) | 49270 grayscale images (96x96) with labels indicating 10 finger types |
| fingerNum (Train) | (39416, 96, 96, 1) | (39416, 10) | Training set for fingerNum classification |
| fingerNum (Validation) | (9854, 96, 96, 1) | (9854, 10) | Validation set for fingerNum classification |
| fingerNum (Test) | (6000, 96, 96, 1) | (6000, 10) | Test set for fingerNum classification |

# RESULTS

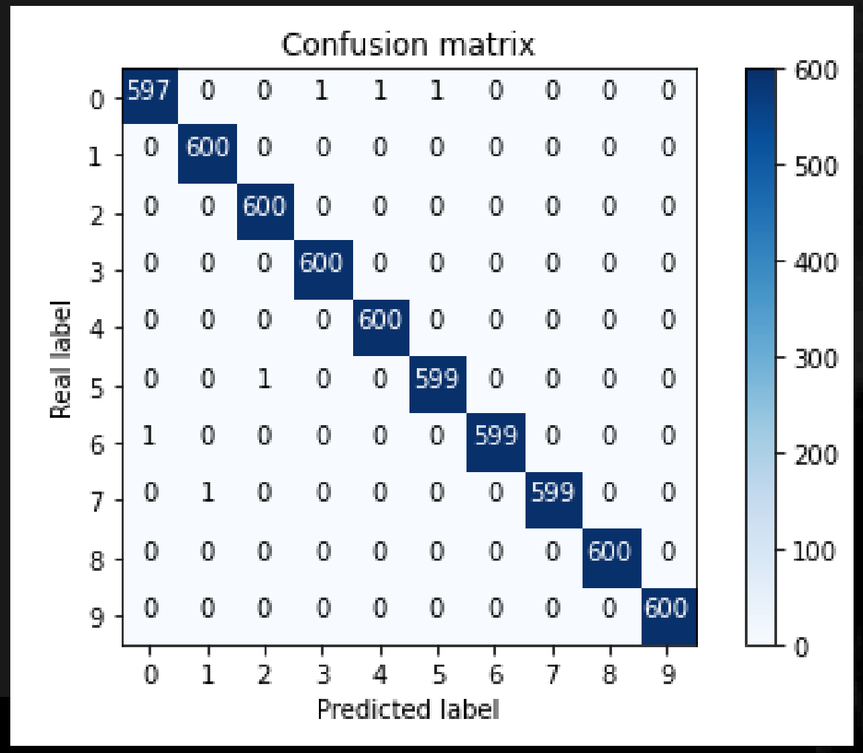
Id recognition accuracy: 99.75 %




Finger recognition accuracy: 99.90 %

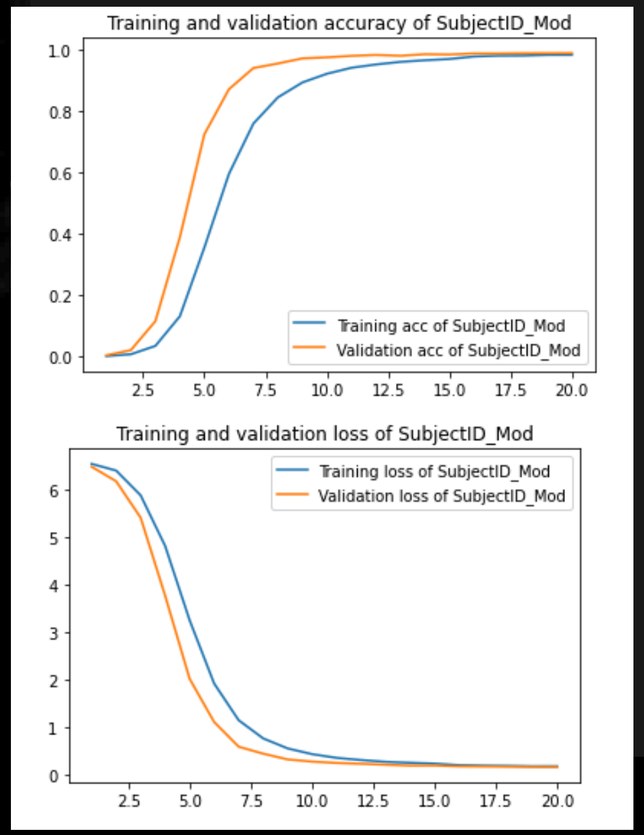



Finger Num Model - Confusion Matrix

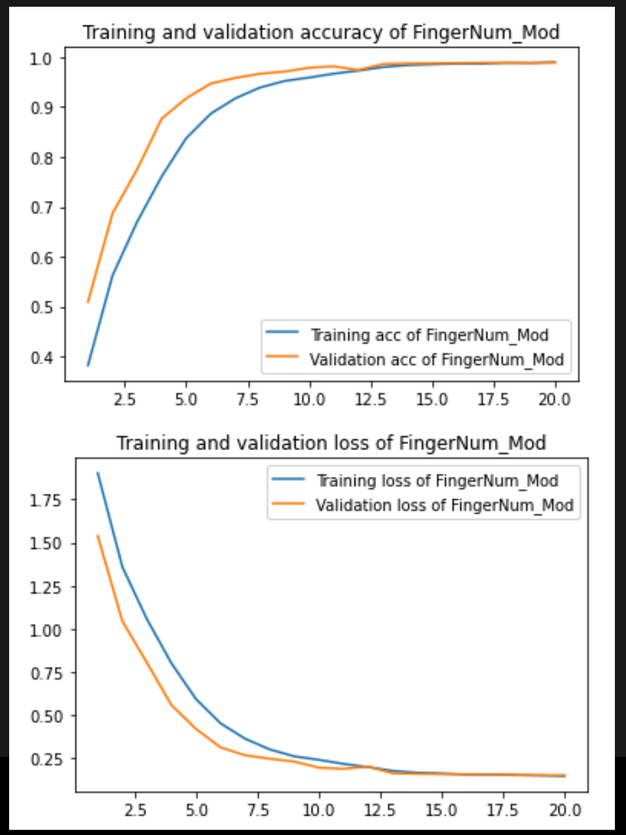


# RESULTS W/ PREPROCESSING

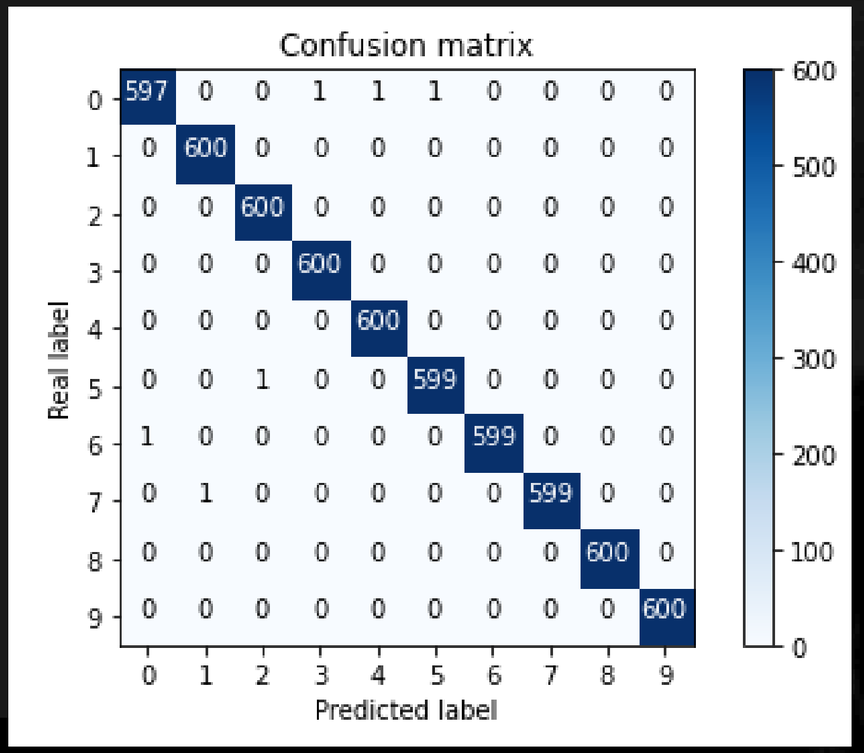
Id recognition accuracy: 99.75 %



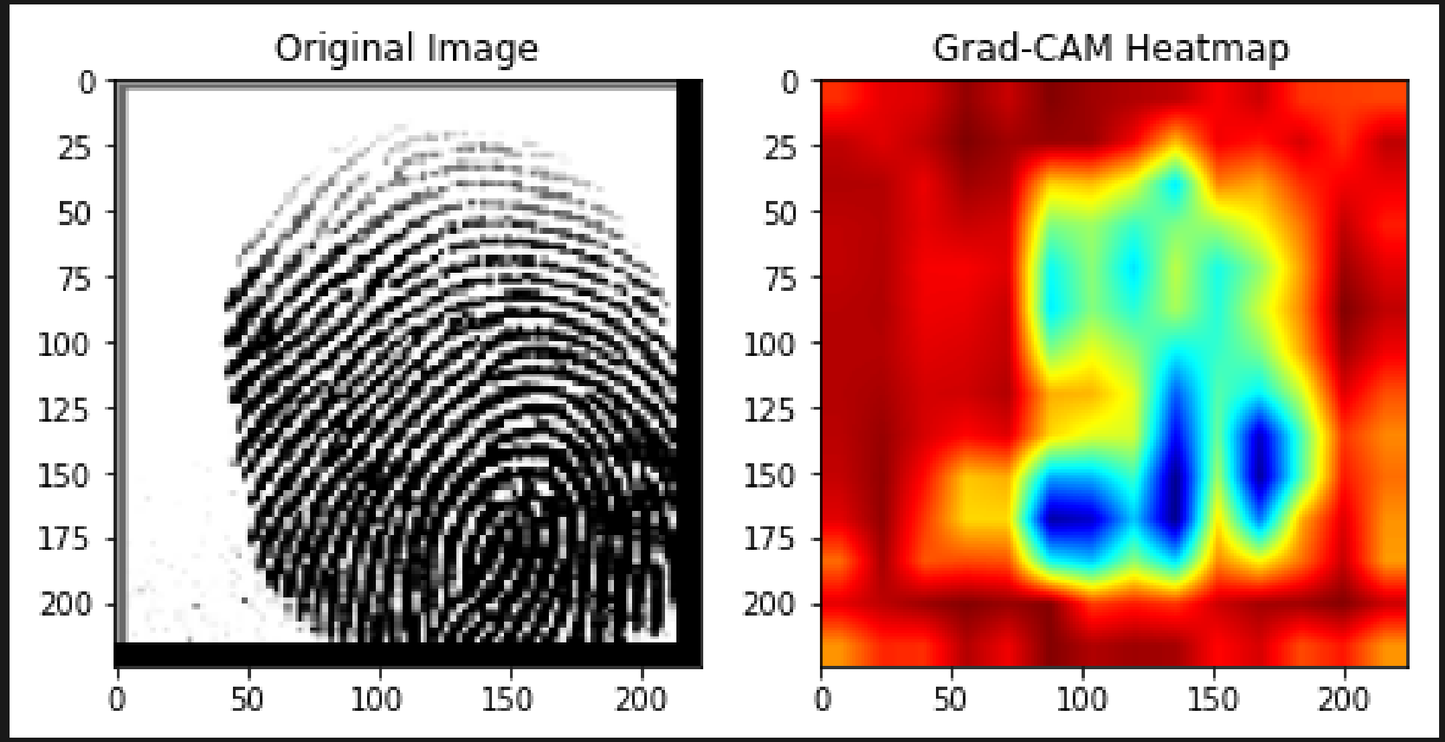

Finger recognition accuracy: 99.90 %



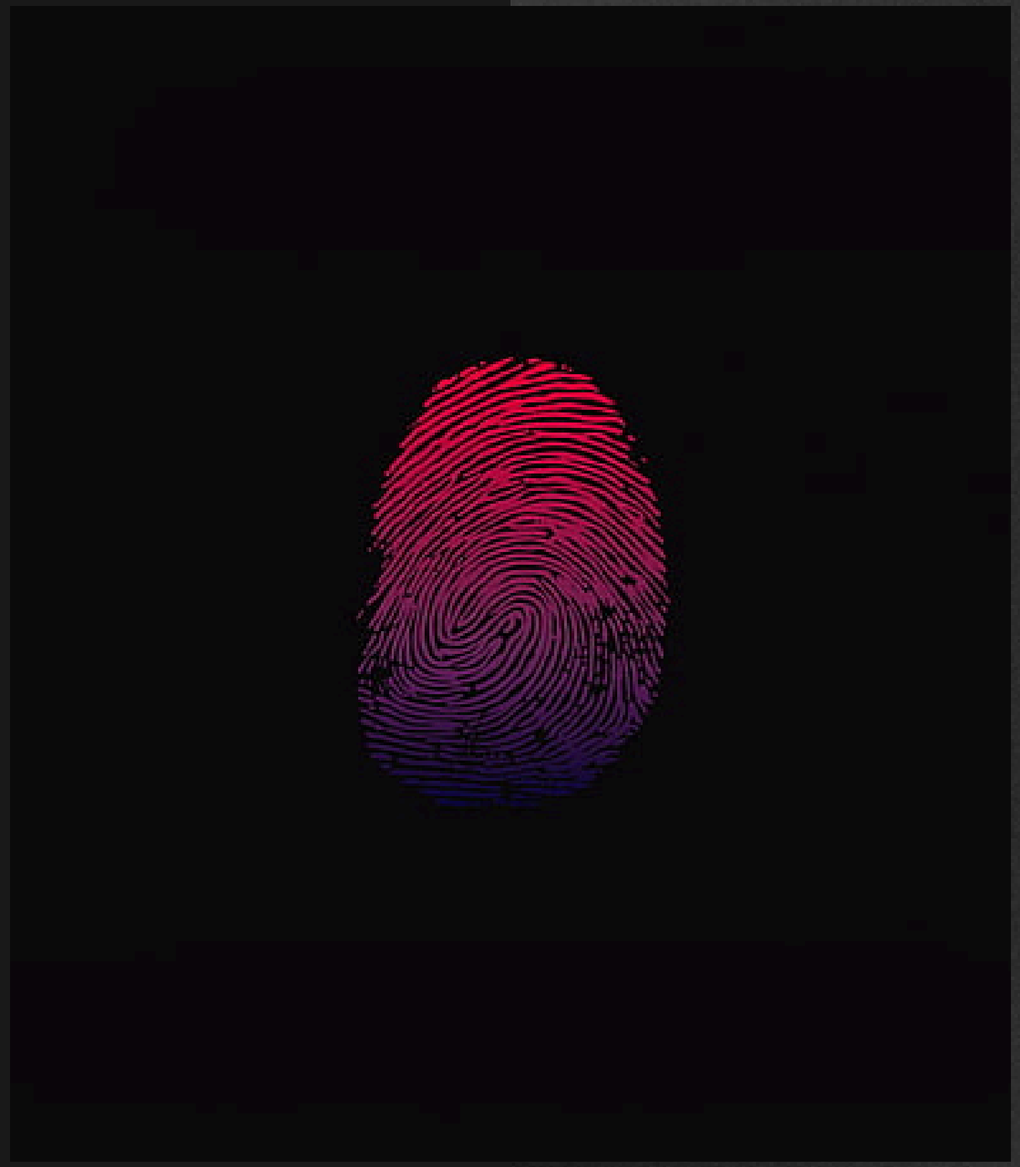

Finger Num Model - Confusion Matrix



# FUTURE SCOPE



# THANK YOU



**Presented By**

Najeeb saiyed
Godcares Ndubuisi
Ankit