2023 Spring CSE713 Study on Machine Learning Approach Using Fingerprint Recognition System

1st Md. Sajeebul Islam Sk.
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
sajeebul.islam.sk@g.bracu.ac.bd

2nd Iffat Ara Jui
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
iffat.ara.jui@g.bracu.ac.bd

3rd Fatema Tuz Zohora
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
p.f.zohora@gmail.com

4th Jubayer Ishfaq Rafid
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
jubayer.ishfaq.rafid@g.bracu.ac.bd

5th Md Sabbir Hossain
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
md.sabbir.hossain1@g.bracu.ac.bd

6th Annajiat Alim Rasel
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*—The pattern of Fingerprint are permanent and unchangeable on each finger during all the life. In situations where algorithmic methods are either unavailable or too computationally intensive, neural networks allow for the solution of issues. An application for neural networks that seems appropriate is the issue of feature extraction and classification. When compared to traditional technique, they give a substantial speed advantage. Fingerprints are distinctive and remain enduring throughout a person's life. The automatic fingerprint recognition system based on ridges and it's characteristics known as minutiae. It is extremely important to mark these minutiae accurately. In this work we have used ridge termination and ridge bifurcation as minutiae for fingerprint recognition system. At the time of analysis the approaches of attributes impart better result. With this technique recognition rate of this intended method of fingerprint recognition system using neural network is quite impressive. From the extraction outcome we may infer about a very affirmative impact of neural network on recognition rate.**

***Index Terms*—Fingerprints, Feature extraction, Matching, Neural Network**

I. Introduction

Fingerprint recognition refers to the automated method of identifying or confirming the identity of an individual based on the comparisson of two fingerprints. Fingerprint recognition is one of the most well known biometrics, and it is by far the most used biometric solution for authentication on computerized systems. The reasons for fingerprint recognition being so popular are the ease of acquisition, established use and acceptance when compared to other biometrics, and the fact that there are numerous (ten) sources of this biometric on each individual.A fingerprint is one of the more popularly used biometrics used in-person identification [1]. This is because fingerprints are easy to collect, examine, and classify. No two persons have been found with the same fingerprints and are found to be unique. Fingerprint characteristics never change throughout the age of a person. There are three basic fingerprint patterns: the arch, the loop, and the whorl. These patterns are defined by structures known as cores and deltas. The core of the print is the central area. A delta is a triangle-shaped area of a fingerprint where the ridge formation changes direction. The structure of The fingerprint and how The machine divides The overall fingerprint structure In different layers to identify it and learn the basic structure Of it: it is generally trained using A basic structural division of a fingerprint,(1) Global structure, (2) Low level structure,and (3) Low level structure.The first one represents the overall shape of the finger. Thus, the second one represents the valleys and ridges format at local intersecting region, and the later. i.e the low level structure represents the sweat pores on the fingerprint skin. Manual observations of a fingerprint are prone to inconsistency and can lead to errors [2].

manual fingerprint matching is time-consuming and may lead to errors.The most widely used system is automatic fingerprint identification system (AFIS) which has replaced human experts in fingerprint recognition as well as classification.

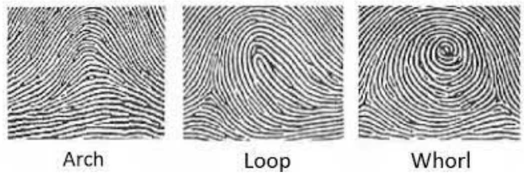

Fig. 1. Types of fingerprints

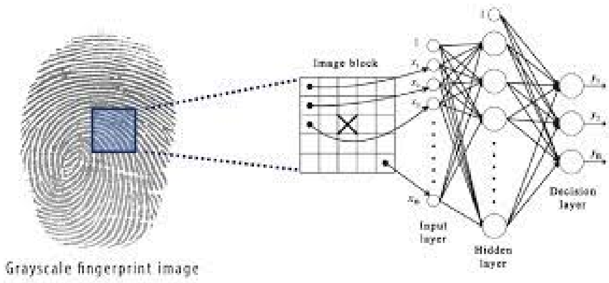

Fig. 2. Fingerprint Recognition System

it begins with the (I) enrollment phase, which basically involves the registration phase where the individual identity(the fingerprint structure) is fed to the machine for it to learn and later identify, the second phase is called the identification phase, however, responsible for extracting The individual identity from The database according to the user claimed identity.

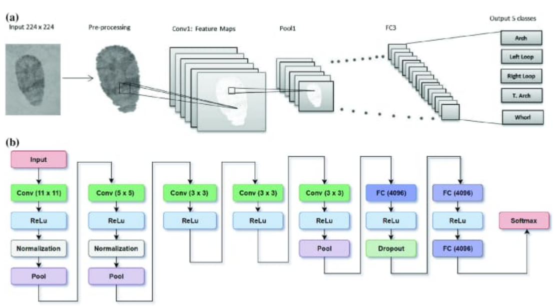

Fig. 3. Fingerprint Recognition

II. LITERATURE REVIEW

All the fingerprint identification different types of work have been done so far. We have gone through various research papers, till today the methods used different algorithms. These algorithms intend to profit from on this uniqueness to develop the efficiency and provision for matching accuracy the fingerprint the fingerprint recognition and confirmation [3]. In this paper neural network back propagation for trained the finger print classifier to identify the fingerprints with time efficient preprocessing.

Biometric measures the uniqueness of an individual based on the physiological, biological and behavioral properties of their body. This tool presents itself as a reliable scientific method to identify and authenticate an individual which are fundamental security principles. The present work makes a contribution on fingerprint recognition which is robust method and has the advantage of security, many attacks are emerging. In field of fingerprint there are fake fingerprints are made by using printed fingerprints, silicone, wood glue or other products [4]. An alternative to classical fingerprint biometrics research is being done on finger veins [5]. This technology uses infrared to observe vein features [6].

The use of infrared has the advantage of extracting characteristics from the skin of finger and from the veins of the fingers to avoid forgery, because veins cannot be forged. More and more studies are made on multimodal biometrics [7] [8]. In this studies, at least two modalities are used. Faced with the complexity and precision of the textures of the sweat bread [9] and fingerprints several studies are interested in the dual of iris paints [10].

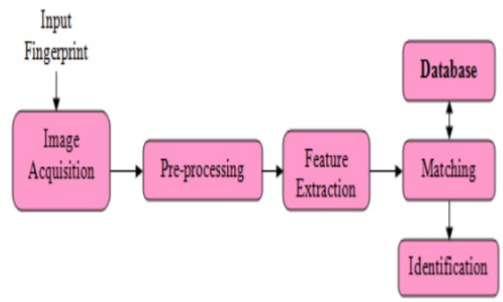

Fig. 4. Fundamental steps of finger print identification system

Recent research focuses on deep learning, which is widely used in computer vision and mainly in biometrics [11]. It's success is due to it's robustness with better recognition scores. The study done in this work shows the performance of convolutional neural networks in fingerprint recognition CNN [12] were applied to the fingerprint database.

III. METHODOLOGY

In our proposed work there are four sets of data, each consisting of a set of grayscale images and their corresponding labels. The sets are named xreal, xeasy, xmedium, and xhard, and their corresponding labels are yreal, yeasy, ymedium, and yhard, respectively.

- The npz file format is a compressed format that can store multiple arrays in a single file. Here, each npz file contains a single array named 'data', which is loaded and assigned to a variable.
- The first image and label from each set using a subplot with four columns and one row. Each subplot has a title displaying the label and displays the corresponding image using the imshow function from matplotlib, with a grayscale colormap.

- The code is likely being used to visualize the different levels of difficulty of the image sets, with xreal being the most difficult and xeasy being the easiest.
- The xeasy, xmedium, and xhard image sets and their corresponding label sets into a single xdata and labeldata dataset using NumPy's concatenate function. The images and labels are concatenated along the first dimension, resulting in a new dataset with a size equal to the sum of the sizes of the individual datasets.
- Then the combined dataset is splited into a training set (xtrain and labeltrain) and a validation set (xval and labelval) using the traintestsplit function from the scikit-learn library. The testsize parameter is set to 0.1, indicating that the validation set should contain 10 percent of the total data.
- Finally, the code prints the shapes of the original and split datasets to confirm that the splitting was successful.
- Data augmentation is applied to a single image from the combined dataset xdata to create nine additional images, and then visualizes the original image and the augmented images using matplotlib's subplot function.
- The data augmentation is performed using the imgaug library, which provides a wide variety of image augmentation techniques. Here, the sequence of augmentations applied includes Gaussian blur, scaling, translation, and rotation. These augmentations are applied randomly and independently to each image, resulting in a diverse set of augmented images.
- The augmented images are stored in the augs variable using the augmentimages function of the imgaug library. The resulting augmented images are then plotted using matplotlib's subplot function, with the first subplot displaying the original image, and the remaining subplots displaying the augmented images.

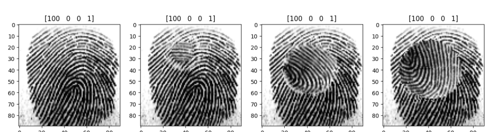

Fig. 5. Finger print identification system

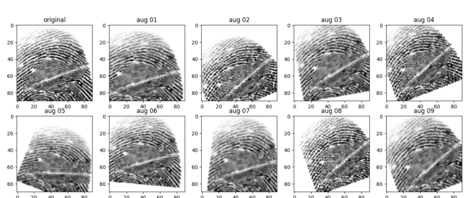

Fig. 6. Augmentation in Finger print identification system

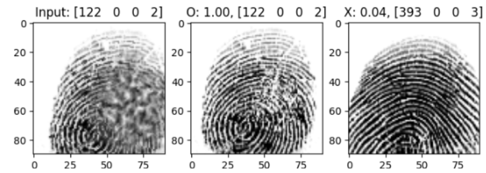

Fig. 7. Evaluationin Finger print identification system

- the data augmentation techniques are being used before training a model on the augmented data.
- A dictionary is created to lookup table that maps a string key to an integer label for the real image dataset (xreal). The labels are represented as a 6-digit string, with leading zeros added to ensure that each label has the same length.
- The code iterates through each label in the yreal array using a for loop, and converts it to a string using the astype method of the NumPy array. The resulting string is then zero-padded using the zfill method to ensure that it has 6 digits.
- The zero padded string is then used as the key in the labelrealdict dictionary, with the corresponding integer label (i) as the value. This results in a dictionary lookup table that can be used to quickly retrieve the integer label for a given string key.

```
model.summary()

Model: "model_1"
____________________________________________________________________________________
 Layer (type)                   Output Shape          Param #     Connected to
====================================================================================
 input_1 (InputLayer)           [(None, 90, 90, 1)]   0           []

 input_2 (InputLayer)           [(None, 90, 90, 1)]   0           []

 model (Functional)             (None, 22, 22, 32)    9568        ['input_1[0][0]',
                                                                   'input_2[0][0]']

 subtract (Subtract)            (None, 22, 22, 32)    0           ['model[0][0]',
                                                                   'model[1][0]']

 conv2d_2 (Conv2D)              (None, 22, 22, 32)    9248        ['subtract[0][0]']

 max_pooling2d_2 (MaxPooling2D)  (None, 11, 11, 32)   0           ['conv2d_2[0][0]']

 flatten (Flatten)              (None, 3872)          0           ['max_pooling2d_2[0][0]']

 dense (Dense)                  (None, 64)            247872      ['flatten[0][0]']

 dense_1 (Dense)                (None, 1)             65          ['dense[0][0]']

====================================================================================
Total params: 266,753
Trainable params: 266,753
Non-trainable params: 0
```

Fig. 8. Model of the finger print identification system

- This type of label dictionary lookup table can be useful for tasks such as image classification, where the label for an image is often represented as a string or other non-integer value. By creating a lookup table that maps these non-integer labels to integer labels, it is easier to use them in machine learning algorithms that require integer labels, such as neural networks.
- A custom data generator class DataGenerator is for

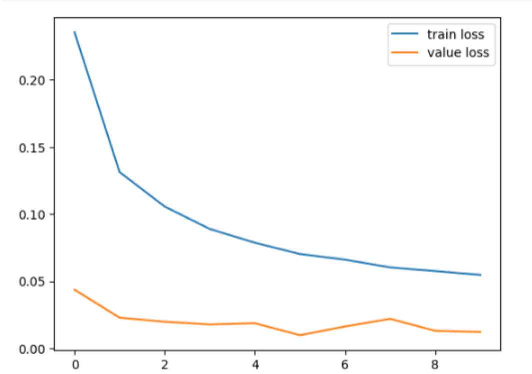

Fig. 9. Train loss and Value loss in finger print identification system

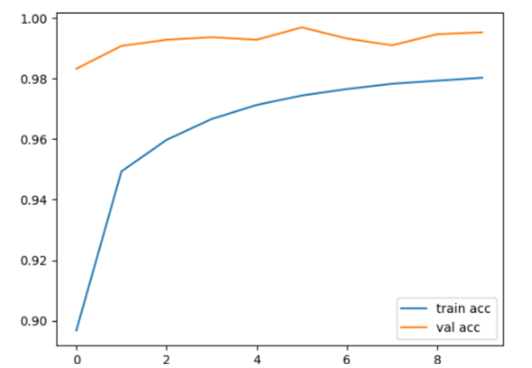

Fig. 10. Train accuracy and Value accuracy in finger print identification system

training a Siamese neural network. The class takes in the training images x, corresponding labels label, a dictionary lookup table labelrealdict for mapping the labels to their index in the real label array yreal, and some additional parameters such as batch size and shuffle flag.

- The len() method returns the number of batches per epoch, which is calculated by dividing the total number of samples by the batch size.
- The getitem() method generates one batch of data. It first selects a batch of x1 images and their corresponding labels from the training set. It then initializes empty arrays for x2 images and their corresponding labels. It applies image augmentation to the x1 images if the shuffle flag is set to True.
- For each image in the batch, it selects a random image from the training set to form a pair. If the random image matches the same person as the current image, it is labeled as 1.0 (matched pair). Otherwise, it is labeled as 0.0 (unmatched pair). The selected x2 image and the corresponding label are added to the batch.
- The method returns the batch as a tuple of two inputs (i.e., x1 and x2) and one output (ybatch).
- The onepochend() method shuffles the x and label arrays if the shuffle flag is set to True.
- The history object returned by the fitgenerator method contains information about the training process, such as the loss and accuracy on the training and validation sets at each epoch. You can use this information to visualize the training progress and make any necessary adjustments to the model or training process.

IV. Implementation

A. *Convolution Neural Network*

The Convolution Neural Network introduced here for Fingerprint Recognition system as a trainable deep neural network. The proposed system enhances the fingerprint image during preprocessing and matches the fingerprint image in recognition phase. CNN approaches are used to realize recognition and preprocessing phases [13]. Ridgeline thinning is performed using a skeletonization method. Quality improved extracted lines are further processed thus enhancing the overall system performance.

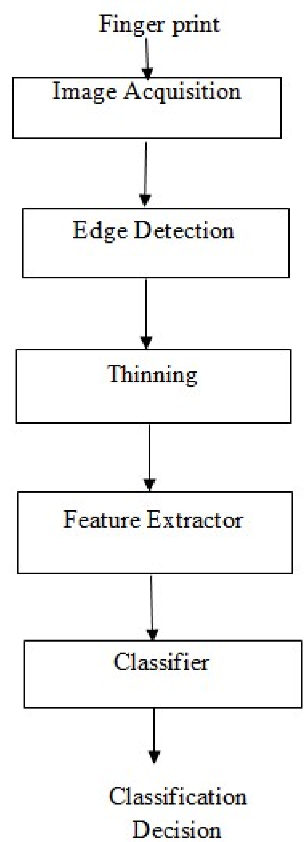

Fig. 11. Convolution Neural Network

Fingerprint Recognition Module is represented in Fig. 12. This module comprises feature model which accepts user fingerprint to process and verify. It determines 1 for same fingerprints and 0 for different fingerprint.
Title of figures [101 0 1 2] means, left to right order:

- Subject id
- Gender (male 0, female 1)

- Left or Right hand (left 0, right 1)
- Finger index (0-4)

Applied some augmentation (gaussian blur, zoom, translation, rotation) to input (left image) for wild environment.

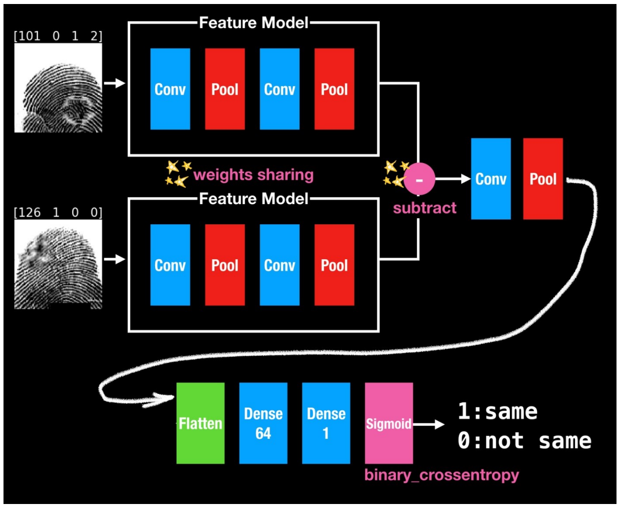

Fig. 12. Fingerprint Recognition module

B. Testing

Different test scenarios in unit testing are:

- TC 001: This unit test case is to test for different fingerprint (Thumb): Image is matched to the images in the database resulting success.
- TC 002: The last unit test case is to test for partial fingerprint (Thumb): Image is matched to the images in the database resulting success.

Different individual test cases are grouped to form a larger group. This is also used in order to fixes errors. If no errors arise in the unit testing phase, then the next phase of testing would be to do integrated testing.

There are only three test cases in integrated testing.

- TC 001: Test case used to check if the user is authenticated, after any random input image, Results success.
- TC 002: Test case used to check if the user is authenticated, after any random fingerprint image, Results success.
- TC 003: This test case is used in order to check if the user is authenticated, after any partial fingerprint image/input image, Results success.

System testing is a testing process where in a completely integrated system is tested. This is done in the final stages of development of a system. If this testing technique is done first and an error arises, the entire system must be stripped down and testing process must begin from unit testing.
The system testing test cases are:

- TC 001: This testing process is carried out in order to check if a valid user is authenticated. This test case is a success.
- TC 002: This testing process is carried out in order to check if an invalid user is authenticated. This test case is a success.

V. Result and Discussion

CNN algorithm is successful in identifying the input fingerprint from the altered fingerprints and even partial matches were considered. The CNN was implemented using Python (Jupyter Notebook) and the open-source platform Keras for the TensorFlow machine learning toolbox. It is run in Google Colaboratory (Colab). Performance analysis of fingerprint recognition is shown in Fig. 13. Let's assume that left image

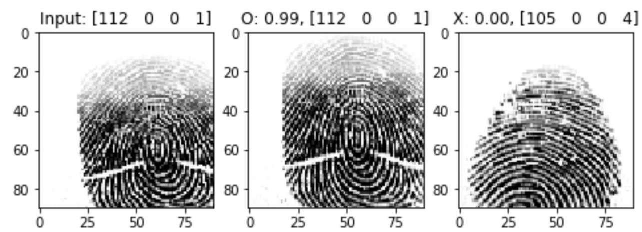

Fig. 13. Performance analysis of fingerprint recognition

is a new input from user, center and right images are stored in database.
Title of figures [122 0 0 1] means, left to right order:

- subject id
- gender (male 0, female 1)
- left or right hand (left 0, right 1)
- finger index (0-4)

Applied some augmentation(gaussian blur, zoom, translation, rotation..) to input(left image) for wild environment.Center image is the answer so it has 99 percent confidence, on the other hand, right image is wrong so that has 0 percent.
Over fitting, slower learning speeds and increase in the number of parameters are some of the limitations of using CNN for fingerprint recognition [14]. Batch normalization method is used to overcome the drawbacks. Identifying the right dataset for the machine learning model is difficult. Analysis can only be done if the hands are laid on the right kind of data. Attaining access to different types of data in suitable format is very challenging and time-consuming. Insufficient volume of data, hidden data and less variety in the kind of data are some of the issues that arise during data collection [15].

VI. Conclusion

Our work presented CNN based fingerprint methodology to enhance and match the fingerprints to improve the performance of a fingerprint recognition system. The work conducted in this paper is mainly devoted to fingerprint identification using a CNN network that can perform fingerprint classification by considering whole fingerprint images [16]. The proposed algorithm uses poor-quality original raw fingerprint images.

In order to reduce the expensive training cost, data resizing was applied. Hyper-parameter tuning, using various epoch numbers, was considered to improve the performance of classification. Training sets and size should be minimal for matching problems when compared to classification problem. Classification of whorls demands the need to expose the network to a large sample representative of whorl patterns. These inferences made by our study. Following these considerations, we would argue that the proposed method can achieve a very good performance compared to the traditional hand-crafted features method, despite the fact that it uses raw data.

VII. Future Enhancements

For future developments, we are interested in improving the performance of classification by using other pre-processing techniques correlated to extensive hyper-parameter tuning [14]. Additionally, other fingerprint databases will be used to assess the generalization capabilities of CNN architectures.

Further research on multimodal biometric with a federated setup with the help of the open source community and its users to obtain the list of resources for better monitoring and management of resources can lead to better access control and safety.

References

[1] A. K. Datta, H. C. Lee, R. Ramotowski, and R. Gaensslen, *Advances in fingerprint technology*. CRC press, 2001.

[2] B. T. Ulery, R. A. Hicklin, J. Buscaglia, and M. A. Roberts, "Accuracy and reliability of forensic latent fingerprint decisions," *Proceedings of the National Academy of Sciences*, vol. 108, no. 19, pp. 7733–7738, 2011.

[3] M. A. Medina-Pérez, A. M. Moreno, M. Á. F. Ballester, M. García-Borroto, O. Loyola-González, and L. Altamirano-Robles, "Latent fingerprint identification using deformable minutiae clustering," *Neurocomputing*, vol. 175, pp. 851–865, 2016.

[4] K. Cao and A. K. Jain, "Automated latent fingerprint recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 4, pp. 788–800, 2018.

[5] R. Cappelli, M. Ferrara, and D. Maltoni, "Fingerprint indexing based on minutia cylinder-code," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 1051–1057, 2010.

[6] U. U. Deshpande and V. Malemath, "Minu-extractnet: automatic latent fingerprint feature extraction system using deep convolutional neural network," pp. 44–56, 2021.

[7] J. Ezeobiejesi and B. Bhanu, "Patch based latent fingerprint matching using deep learning," pp. 2017–2021, 2018.

[8] J. Feng, J. Zhou, and A. K. Jain, "Orientation field estimation for latent fingerprint enhancement," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 4, pp. 925–940, 2012.

[9] M. D. Garris, "Rm mccabe, nist special database 27, fingerprint minutiae from latent and matching tenprint images," 2000.

[10] A. A. Paulino, J. Feng, and A. K. Jain, "Latent fingerprint matching using descriptor-based hough transform," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 31–45, 2012.

[11] D.-L. Nguyen, K. Cao, and A. K. Jain, "Robust minutiae extractor: Integrating deep networks and fingerprint domain knowledge," pp. 9–16, 2018.

[12] Y. Zhu, X. Yin, X. Jia, and J. Hu, "Latent fingerprint segmentation based on convolutional neural networks," pp. 1–6, 2017.

[13] W.-S. Jeon and S.-Y. Rhee, "Fingerprint pattern classification using convolution neural network," *international journal of fuzzy logic and intelligent systems*, vol. 17, no. 3, pp. 170–176, 2017.

[14] R. Abrishambaf, H. Demirel, and I. Kale, "A fully cnn based fingerprint recognition system," pp. 146–149, 2008.

[15] D. Cozzolino and L. Verdoliva, "Noiseprint: A cnn-based camera model fingerprint," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 144–159, 2019.

[16] M. H. Mohamed, "Fingerprint classification using deep convolutional neural network," *J. Electr. Electron. Eng*, vol. 9, pp. 147–152, 2021.