Designing Deep Learning Architecture for Latent Fingerprint Matching

Yenumula B Reddy

Grambling State University, Grambling, LA 71245, USA

Keywords: fingerprints; latent fingerprints; image recognition; neural networks; deep learning;

Abstract

Image recognition using partial information is one of the significant problems in current computer vision research. Computer vision has an important role in medical, robotics, economics, and crime-related subjects. Complete fingerprint or face is usually not available at a crime scene or camera image. The full image or partial image of crime is common in most of the crime data. The crime branch identifies the related match of criminal's data with the help of collected image or partial image data. In this research, we introduce the fingerprint background, classification, and deep learning artificial neural network model to identify the closest match image (fingerprint) for a given partial image (latent fingerprint) data..

1 Introduction

Fingerprint, facial, e-commerce, and healthcare data are the examples of big data. Classification of such data is an important research in the current technology. Existing machine learning algorithms help but take a longer time to process large sets of data. Storage, classification, and analysis of fingerprint data support for fast processing. Processing and automatic identification of a fingerprint image with accuracy and speed are the crucial part in crime detection. Currently, criminal detection uses face, fingerprint, vein & blood flow, and type of technical activity (computer keystrokes and intruding). Deep learning (DL) models are in the research and experimental stage. Deep learning packages are in the research and development using GPU (Graphics Processing Units) technology and available soon for criminal detection on a real-time basis.

Storing of the image of a fingerprint takes three steps. First, create a high-quality image with a high contrast between ridges and valleys before input the fingerprint image. Second, convert the fingerprint image into vector form and improve the smoothing quality of the fingerprint ridges. Third, identify the core point (turning point of innermost ridges) and delta point (two ridges running side-by-side). Finally, classify as Plain Arch, Tented Arch, Left Loop, Right Loop, and Whorl.

Collecting the fingerprint image with minimum noise is important. The image collection includes block-wise (foreground) and bit-wise (background) regions [1 - 2]. Block-wise has noise and good for automated processing. Bit-wise segmentation is a noiseless and time-consuming process. Collect the image and apply the enhancement algorithm to create a quality of image [3 - 4]. Image enhancement can be the distinct region, recoverable region, and unrecoverable region. Images can be recovered depending upon the ridge and valley structures. The image quality may not be good all the time, and different regions have different quality. Due to many of the quality facts, it is hard to classify the images with standard rules [5].

Neural networks are the alternative method to train the example fingerprints and retrieve the matched fingerprint with presented input fingerprint. The neural networks learn from its coarse-grained features. Several approaches developed from the fingerprint patterns like ridge orientation, and minutiae detection [6]. The ridge orientation fingerprint patterns then used as inputs to neural network classifiers. Fields [7], proposed the training of two disjoint neural networks and passed the output to the third neural network to produce the desired result. This process is tedious and time-consuming. It requires much effort for an efficient classifier.

The feature extraction and classification require image acquisition, pre-processing and segmentation. Image acquisition includes collecting several samples to create a fingerprint database. The fingerprint database is useful for training and testing. Pre-processing enhances the borders using high-pass filtering and high-frequency emphasis [8]. The segmentation process converted from grayscale to binary through fixed threshold value and submitted to Morphological operations of erosion and dilation. The process eliminates the black area at boundaries and keeps the region of interest in a required rectangular block. The normalized images are input to the wavelet transform according to FBI suggestions. The remaining process includes statistical analysis for validity depends upon proposed algorithms before submitting to neural networks.

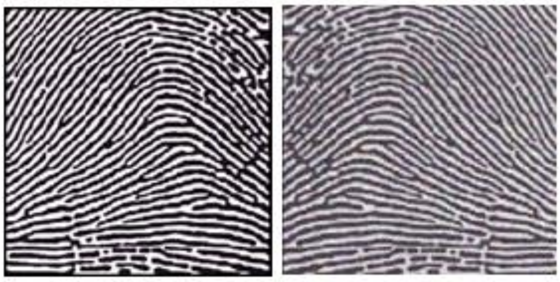
Performance in training and matching the closest fingerprints from a database requires special models. Deep learning is neural network based model with many hidden layers and composed of multiple levels of nonlinear operations. Deep networks are too difficult to train. The goal is to learn feature hierarchies where higher levels of the hierarchy (edges, local shapes, and image parts) formed by lower levels features (raw input). The MapReduce technique is an alternate technique to DL to classify the input data and identify the closest match image. The classification requires a different technique for MapReduce that is a future research topic.

The recent research shows that GPU-based deep learning is in research and production [9 - 12]. Currently, developers are working DL-based implementation in Social media, medical, energy, entertainment, Defense, intelligence, and media. Fingerprint data is one of the recent introductions of GPU-based DL model for real-time response. The requirement is to retrieve the matched fingerprint using latent fingerprint (partial image) data. The proposed research is an attempt to use GPU-based model for deep learning neural networks to identify the exact match fingerprint using latent fingerprint data.

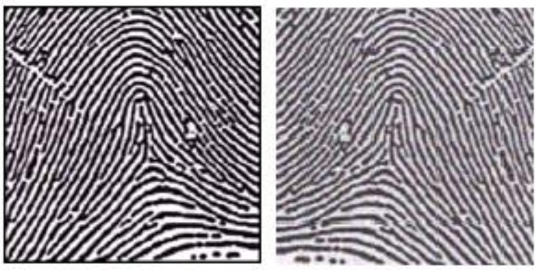
2 Fingerprint Basics

Criminal investigators identified that no two individuals have the same fingerprint, fingerprint remains unchanged (may change due to diseases or permanent scars) for life, and the ridge patterns help for identification. Individual fingerprints (dactylograms) are collected and stored in a fingerprint database. Fingerprints classified into three groups. The three groups include arches, loops, and whorls depend upon their visual patterns. Each group has been subdivided into smaller groups as shown below.

- Arch: Plain Arch and Tented Arch (Figure 1)
- Loop: Radial Loop and Ulnar Loop (Figure 2)
- Whorl: Plain Whorl, Central Pocket Whorl, Double Loop Whorl, and Accidental Whorl (Figure 3)



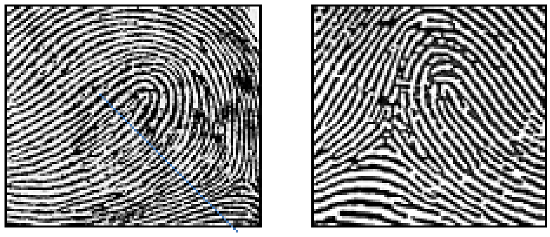
(a) Plain Arch



(b) Tented Arch

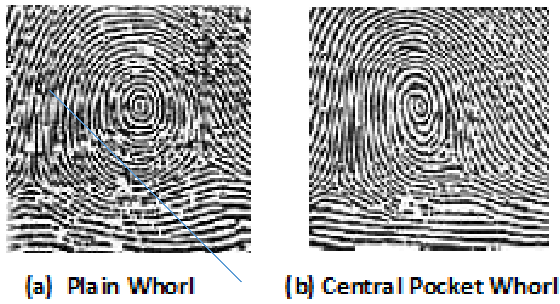
Finger 1: Arch

The simplest fingerprint is arch. They formed by ridges and enter one side (left or right) and exit another side. No deltas are present in the arches (Figure 1). Loops contain delta and ridges that enter one side and leave another side. These patterns are names of the radial loop (left or right) and ulnar loop (left or right) depends upon their positions related to the radius and ulna bones (Figure 2).

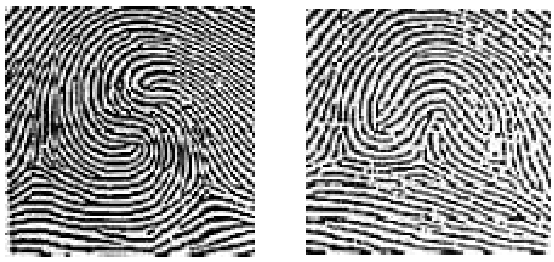


(a) Radial loop (left loop) (b) Ulnar Loop (right loop)

Figure 2: Loops



(a) Plain Whorl (b) Central Pocket Whorl



(c) Double Loop Whorl (d) Accidental Whorl

Figure 3: Whorls

Whorls have at least one ridge and two deltas. It is likely accidental to have more than two deltas to a whorl. Draw a line between two deltas. If the line touches some of the curved ridges, it is a plain whorl otherwise it is a central pocket whorl (Figure 3a, 3b). Whorls may be a double loop or accidental. Double loop whorls are made up of any two loops combined into one print. Accidental whorls contain two or more patterns (does not include plain arch). Such patterns may not fall under any of the different categories (Figure 3c, 3d).

Fundamental and composite ridge characteristics are called minutiae. A ridge is a curved line in a finger image. The minutiae are the specific points in a finger (at structure change). Ridges are upper skin layer segments of the finger.

The valleys are the lower segments. Minutiae are a transform on the ridges of the fingerprint. Minutiae are usually ridge endings and bifurcations. The position, angle, and form determine the minutiae type. Figure 4 provides the minutiae characteristics. Figure 5 provides the detailed examples of minutiae.

Figure 4: Minutiae Characteristics

3 Fingerprint Classification

Classification and sub-classification of fingerprints involve the careful placing of fingerprint data in a file for feature retrieval. The classification reduces the search space. For example, if N is the size of the database of 10 fingers of each individual and D is the number of classes, the search space is calculated as below.

No classification: N^{10}
Classification: $(N/D)^{10}$

The primary classifications include Henry classification, Juan Vucetich Classification, and Battley Single-Fingerprint System. These three systems lead to the development of National Crime Information Center (NCIC) classification and Integrated Automated Fingerprint Identification System (IAFIS). The brief description of the three systems helps us to classify the fingerprints for a Deep Learning project.

A. Henry Fingerprint Classification Systems (HFCS)

Henry classification system gives a number to each finger based on finger pattern type. The right thumb takes the number 1 and left thumb has number 6. The right hand has the number 1 to 5 and left-hand fingers are numbered 6 to 10. Table 1 shows the numbers assigned to each finger.

Finger	1	2	3	4	5	6	7	8	9	10
Number	16	16	8	8	4	4	2	2	1	1

Table 1. Henry Fingerprint Classification

Depending on the arch, loop or whorl to the fingers, the letters ‘A’, ‘L’, or ‘W’ are given as code. Using this code the finger value is calculated. If the finger 6 has whorl the value is 4 and whorl in finger 2 takes the value 16. Table 2 provides the presence of whorl in each finger.

The numerator takes the whorl values of even fingers +1 and denominator takes the whorl values of odd fingers + 1. The grouping is calculated as below.

$$\text{Primary Group Ratio} = \frac{1 + \text{Sum of even numbered Fingers}}{1 + \text{Sum of odd numbered Fingers}} \quad (1)$$

Sir Francis Galton formulated the classification system based on HFCS. Galton grouped the letters for the right hand’s index, middle, and fingers followed by same fingers of left hand. He then appended the right-hand thumb, right-hand little finger, left-hand thumb, and left-hand little finger. This classification then recorded for each criminal for later investigation.

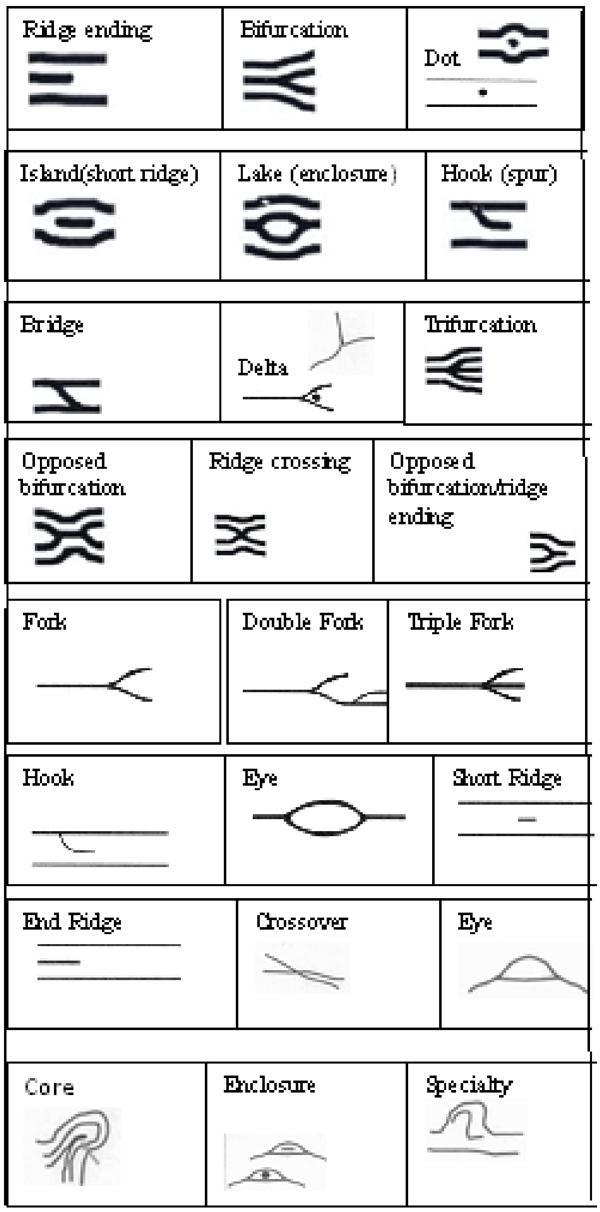


Figure 5: Name of Minutiae and its identification on Fingerprint

HFIS influenced AFIS highly during its first introduction of AFIS technology. Initially, AFIS emulated HFCS process. The current automatic process does not require HFCS emulation.

Finger	Number	Value if Whorl
Right thumb	1	16
Right Index	2	16
Right Middle	3	8
Right Ring	4	8
Right Little	5	4
Left Thumb	6	4
Left Index	7	2
Left Middle	8	2
Left Ring	9	1
Left Little	10	1

Table 2. Henry's Primary Values

B. Juan Vucetich Classification

Vucetich extended the Galton's three pattern system (arch, loop, and whorl). He divided the loop into an internal loop (left-scope) and external loop (right-scope) categories. Table 3 explains the Vucetich's pattern-type that is the extension of Galton's 3 letter classification. The system starts with right-hand thumb and ends with the left-hand little finger.

Pattern	Thumbs	Other Fingers
Arch	A	1
Internal loop	I	2
External loop	E	3
Whorl	W	4

Table 3. Vucetich's pattern-type Symbols

Vucetich classification consists of primary and secondary classification. In the primary, the right-hand fingers dealt as series and left-hand fingers as a section. The code is A1141 represents the thumb has an arch, and other fingers have an arch, arch, whorl, and arch. The secondary classification takes the numbers 5, 6, 7, 8, and 9. The description of second level values depends upon the primary pattern. For example, superscript 5 in whorl means normal, 6 – sinuous, 7-ovoid, 8-hooked, and 9-all others. Similarly, for the arch, the internal loop, and external loop the superscript has a description. The superscript value for ridge count spread is always a maximum limit as shown in Table 4. Vucetich's formula uses right-hand values as a numerator and left-hand values as a denominator. The equation (2) explains the Vucetich's secondary classification. The equation (2) explains that a person right-hand has external right slope loops, and left-hand fingers have internal left slope loops.

$$\frac{E^{(20)}3^{(10)}3^{(5)}3^{(15)}3^{(10)}}{I^{(10)}2^{(5)}2^{(10)}2^{(10)}2^{(5)}} \qquad (2)$$

C. Battley Single-Fingerprint System

Battley's proposal includes 10 patterns followed by additional subdivisions. The subdivisions depend upon the pattern designation. They include radial or ulnar inclination, ridge counts, ridge tracings, core (s) format, and the position of the delta (s), and circle readings. For example, the circle reading subdivision calculated using concentric circles of the print. The center placed at designated point of the impression and the circle readings were taken based on the position of a particular formation. Each fingerprint of the criminal was classified and compared. Battley's system requires a great deal of labour and collection became too large. It is tough to collect and process such databases. At the same time, the system is inefficient for latent fingerprints collected at crime scene.

Table 4. Vucetich's ridge count values

Ridge count Spread	Superscript Value
1 – 5	5
6 – 10	10
11 - 15	15
16 – 20	20
Over 20	25

D. National Crime Information Center (NCIC) and Integrated Automated Fingerprint Identification System (IAFIS) classification

The turnaround time is lengthy with the manual process. Therefore, the computing process became unavoidable to minimize the response time. The computer-based classification has 3 phases called primary, secondary, and tertiary. Initially, punch cards were used to store and extract the individual fingerprints. The method was not successful.

The NCIC recognized the need and created the centralized system in 1971. The NCIC system used the eliminating process to search for a potential suspect. The classification used 20 characters for coding of 10 fingers beginning with right thumb to left little finger. Table 5 shows the NCIC classification codes.

Ulnar loop ridge count (actual ridge count)	01-49
Plain arch	AA
Plain whorl, inner tracing	PI
Plain whorl, meet tracing	PM
Central pocket whorl, outer tracing	CO
Double loop whorl, inner tracing	dI
Double loop whorl, meet tracing	dM
Accidental whorl, outer tracing	XO
Missing or amputated finger	XX

Table 5. NCIC Classification Codes

In 1980 AFIS instituted the automatic fingerprint classification with the name Automated Fingerprint Identification System (AFIS). The system was based on the computerized extraction of minutiae. The extraction creates computer determined pattern with minutiae location and direction. Table 6 contains the AFIS pattern classification. There are many algorithms available today to compare and print the nearest match.

Arch	AU
Left-slant loop	LS
Right-slant loop	RS
Whorl	WU
Amputation	xx
Complete Scar	SR
Unable to classify	UC
Unable to print	UP

Table 6. AFIS Pattern Classification Codes

4 Neural Networks and Deep Learning

Machine learning (ML) is the part of Artificial Intelligence. Systems learn using the neural network model to recognize the images that are important without telling what the image is ahead of time. Neural networks used in this process have limited hidden layers (processing power requirement depends upon the number of hidden layers). Current ML systems need vast computational processing to extract values from Big Data such as Facebook (approximately 350 M images uploaded per day), Walmart (2.5 petabytes customer data per hour), and YouTube (100 hours of video every minute) for analysis. The simple neural network models using backpropagation, feed forward neural works, Adaptive resonance theory, and Boltzmann neural network with limited hidden layers may not produce required results for Big Data. They can produce with significant error rate. Therefore, we need an extended model called deep learning model.

Deep learning algorithms based on artificial neural networks to extract complex data representation and identify more complex relations between input and output. Deep learning is a set of algorithms to model high-level abstractions in data by using model architectures in machine learning that are composed of multiple non-linear transformations. The deep neural networks (DNN) in image recognition require special design framework (includes a sufficient number of hidden layers and input parameters) and ad3equate input images for training. Big Data processing requires the combination of DNN and GPU-based architecture for real-time analysis of data.

The DNN has pre-training and post-training states. In the pre-training state, the system learns by parts with supervised or unsupervised learning mode. In the post-training state, the system uses backpropagation model. The GPU-based systems used to program DNN with special algorithms which are made to do multicore the common parallel processing.

In deep learning, we use a neural network architecture with several layers of nodes between input and output. These layers between input and output do feature identification and processing in a series of stages similar to human brain. Multilayer neural networks are in use for several years. The good algorithms in neural networks can work less than five layers. Further, current algorithms are not suitable for several hidden layers (many layers between input and output) due to the nature of multi-layer design.

Currently, we are using the machine learning models includes Backpropagation, Support Vector Machine (SVM), Gaussian Mixture Models (GMM), and Restricted Boltzmann Machines (RBM) as building blocks. These models have limited modeling capability concepts and cannot make use of unlabeled data. The new models require approximate complex decision boundary (fewer computational units for same functional mapping), hierarchical learning (increasingly complicated features), and work well in different domains (examples: vision, audio, and multimedia).

The hierarchical learning is easier to monitor since it has low-level to high-level structure. The RBM finds a good initial set of weights to tune the local search procedures. The Backpropagation model can be used to fine-tune the model to be better at discrimination. The current algorithms are useful for learning weights with one or two hidden layers. Special algorithms are required for more hidden layers and multi-layer architecture. Therefore, we restructure the neural networks with several layers to process in a series of stages just like our brain do. The new model is called deep learning.

The deep neural networks (DNN) require design framework (includes a sufficient number of hidden layers and input parameters) and adequate input images for training. Proposed DNN is the combination of parallel computing (GPU-based architecture), parallel algorithms, and Big Data. The DNN has pre-training and post-training states. In the pre-training state, the system learns by parts with supervised or unsupervised learning mode. In the post-training state, the system uses backpropagation model. Since GPUs programmed for DNN, the algorithms are made to do multicore the common parallel processing.

5 Designing Deep Neural Networks

The traditional multi-layer perception (MLP) was successfully used for image recognition. The experience shows that the model suffers from the curse of dimensionality. Therefore, it does not scale well to higher resolution images. The new theory is DNN that constructs the original image from different parts of that image similar to the visual cortex. The CIFAR-10 dataset image of size 32x32x3 (32 wide, 32 high, and 3 color channels) requires 32*32*3 = 3072 weights. A 200x200x3 pixel image requires 120,000 weights. Full connectivity of such neurons to emulate the behavior of a visual cortex leads to overfitting in the traditional MLP. Therefore, we suggest Convolutional Neural Network (CNN) architecture formed by a stack of distinct layers that transform the input volume to output volume through a differentiable function [14 -17].

Each layer in CNN have many grids. Each grid has several neurons and takes input from all grids of the previous layer. The weights for each neuron in the current section is same. After each CNN layer there is pooling layer (filter) to produce a resolution of the future map. In pooling, each grid produces a value. The value may be average or maximum or linear combination of grid values. Figure 6 shows the CNN architecture and Figure 7 the example of maximum pooling.

The pooling layer controls overfitting of the number of parameters and amount of computation. The pooling layer operates independently on every depth slice of input and sizes spatially. Recent trends tried to discard pooling filter due to the control of the size of the CNN architecture and used many other filters. Rectified Linear Units (ReLU) is one of them that applies the non-saturating activation function f(x) = max (0, x). The ReLU increases the nonlinear properties of decision function and overall network without affecting the receptive fields of the convolution layer. Other filter includes hyperbolic tangent f(x) = tanh (x) and sigmoid function f(x) = (1 + e-x)-1. The following are few of the CNN architectures.

INPUT -> FC Implements a linear classifier
INPUT -> CONV -> ReLU -> FC
INPUT -> [CONV -> ReLU -> POOL] * 2 -> FC -> ReLU -< FC
(Single CONV layer between every pool)

INPUT -> [CONV -> ReLU -> CONV -> ReLU -> Pool] * 3 -> [FC -< ReLU] * 2 -> FC
(Here there are two CONV layers stacked before every pool)

6 Deep Neural Networks for detecting Image anomalies

Photo accuracy is important in the magazine, industries, and Government. Most of the photos received undergo some editing. Poorly edited images (anomalies) need to be flagged and filtered before they enter into production. Convolutional Neural Network (CNN) architecture helps to recognize well-edited images from the stream of image data. Well-designed CNN filters anomalies that occur in the stream of video data. The problem can be extended to detect the anomalies in Big Data (unlabeled and unstructured) generated by sensors and characterize the temporal variance of a site to determine the pattern of life.

CNN architecture comprises training one unsupervised feature extractor on small spatiotemporal patches extracted from sequences of video frames, and subsequently convolving this model with a larger region of the video frames. Figures 6 and 7 show the processing. The resulting feature vectors are then presented for processing layer to generate the output.

CNN requires some latent features to convolute that leads to processing layer. The independent component learning lacks to learn latent features and features reducing the dimensionality.

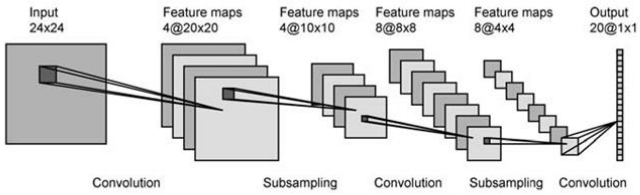


Figure 6: CNN Architecture

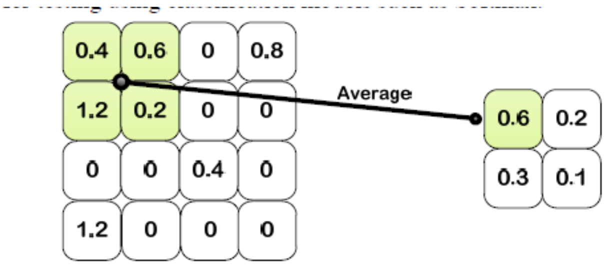


Figure 7: Average pooling

7 GPU Powered Deep Learning for Image Classification

To learn a large set of objects and identify an object from millions of images requires considerable learning capacity. We discussed in the previous sections that CNN architecture has such capability. The capability response limits with current computational power. Therefore, we recommend GPU powered deep learning model for fast response. The proposed GPU architecture is GEFORCE GTX 980M with Intel Core i7 has 1536 CUDA cores with memory clock 2500 MHz, Memory Bandwidth 160 GB/sec with Windows 10 operating system, NVIDIA CUDA Toolkit 7.5, and Caffe (https://github.com/Microsoft/caffe). Alternatively, we can use Ubuntu 14.0, NVIDIA Toolkit 7.5, and Caffe with Tesla K40c GPU card (2,880 CUDA cores) on Linux based system.

Figure 8 shows the latent of one finger divided into four groups for training purpose in image learning CNN amp. The number of images depends upon the training set. The average pooling for each image is taken and display the output. For test accuracy, modify the DNN, fix training set, and update training parameters. The training set of 256 images using K40 GPU was observed 9 times faster than dual core CPU system. Complete details of the training set and explanation is not provided in the current paper.

The following are in instructions to install cuDNN.

- Register as NVIDIA developer.
- Download cuDNN as per our application.
- Install CAFFE and connect the path.

The design of DNN for your need is the biggest problem Limit 5 layers during initial implementation. Monitor the DNN training visually in real-time.

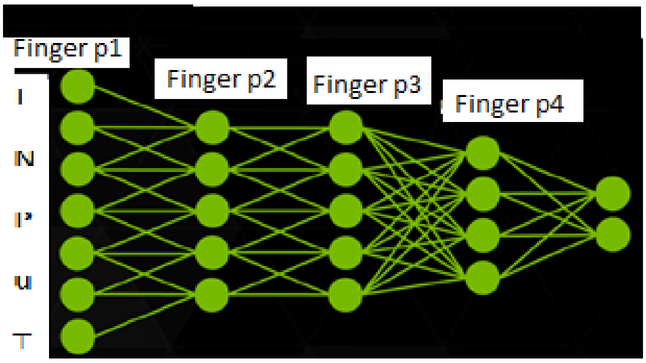


Figure 8: CNN Architecture for latent fingerprint images

8 Conclusions

The paper discusses the basics of fingerprints and fingerprint classification. The design in Figure 8 uses 4 layers of one grid. In fact, the GPU-based deep learning for fingerprints require 10 grids for 10 fingers, each grid has sub-grids. The sub-grids contain the latent of a fingerprint. Training of the latent prints and fusing of these latent prints to match a fingerprint is the focus of the research in progress. The fusion is similar to brain activity involved in human vision. Once the DNN is set and samples are trained, the work can be extended to detection of an image with partial image information.

Acknowledgements

The author wishes to express appreciation to Dr. Connie Walton, Director of Sponsored Programs and Dr. Danny Hubbard, Associate Dean of College of Arts and Sciences, Grambling State University for their continuous support.

References

[1] F. Ahmad and D. Mohammad. "A review on fingerprint classification technique" comp technology and development, vol.2, pp-411-415 (2009).

[2] W. Yifei and A. Jain. "Fingerprint image enhancement: algorithm and performance evaluation", IEEE transactions on pattern analysis and machine intelligence, vol. 20, no. 8, August 1998.

[3] L. Hong and A. Jain. "Classification of Fingerprint Images", Proceedings of 11th Scandividian Conference on Image Analysis (1999).

[4] J. Guo, Y. liu, J. Chang and J. Lee, "Fingerprint classification based on decision tree from singular points and orientation field", Expert Systems with Applications 41(2), pp. 752-764 (2014).

[5] D. Parekh and R. Vig, "Survey on Parameters of Fingerprint Classification Methods Based On Algorithmic Flow", Vol. 2, No.3, August 2011.

[6] Fields, "Expert system with applications", vol. 41, issue2, pp.-752-764(February 2014)).

[7] H. V. Neto and D. L. Borges, "Fingerprint Classification with Neural Networks", Brazi;ian Symposium on Neural Networks, pp. 66- 80 (1997).

[8] R. Gonzalez and R. Woods, "Digital Image Processing", 3rd Edition, Addison-Wesley, 1992.

[9] G. Hinton, S. Osindero and Y. W. The, "A fast learning algorithm for deep belief nets. Neural Computation", Vol. 18, pp. 1527-1554 (2006).

[10] Y. Bengio, P. Lamblin, P. Popovici and H. Larochelle, "Greedy Layer-Wise Training of Deep Networks", Advances in Neural Information Processing Systems, Vol. 19 (2007).

[11] M. Priyanka, "A Review on Classification of Fingerprint Images", IOSR J. Electronics and Communication Eng. (IOSR-JECE), Vo;. 9, issue 3, pp. 61-66 (2014).

[12] E. N. Alshemmary, "Classification of Fingerprint Images Using Neural Network Techniques", Journal of Engineering, Vol. 1, No. 3, pp. 40-48 (2012).

[13] S. R. Patil and S. R. Suralkar, "Fingerprint Classification Using Artificial Neural Networks", Int. J. of Eng. Tech. and Advanced Engineering, Vol. 2, Issue 10, pp. 513 – 517 (2012)

[14] R. Wang, C. Han, Y. Wu and T. Guo, "Fingerprint Classification Based on Depth Neural Networks", The Computing Research Repository (CoRR), pp. 1-10 (2014)

[15] X. Wang, L. Gao, S. Mao and S. Pandey, "DeepFi: Deep learning for indoor fingerprinting using channel state information", IEEE Wireless Communications and Networking Conference (WCNC), pp. 1666-1671 (2015).

[16] Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, T. Hirzel, A. Aspuru-Guzik and R. Adams, "Convolutional Networks on Graphs for Learning Molicular Fingurprints", Advances in Neural Information Processing Systems (NIPS), Vol3. pp. 2215-2223 (2015).

[17] R. F. Nogueira, "Fingerpring Liveness Detection using Convolutional Neural Networks", IEEE Transactions on Information Forensics and Security, Vol 11, Issue 6, pp. 1206-1213 (2016).