Fingerprint Minutiae Extraction using Deep Learning

Luke Nicholas Darlow
Modelling and Digital Science, Council for Scientific and Industrial Research, South Africa
LDarlow@csir.co.za

Benjamin Rosman
Modelling and Digital Science, Council for Scientific and Industrial Research, South Africa
and
School of Computer Science and Applied Mathematics, University of the Witwatersrand, South Africa
BRosman@csir.co.za

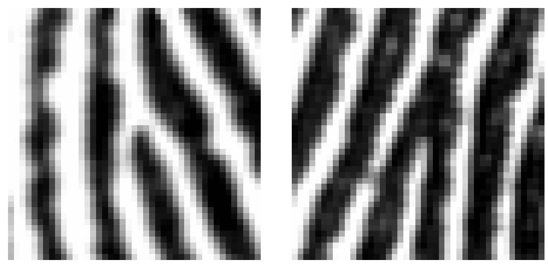
Abstract

The high variability of fingerprint data (owing to, e.g., differences in quality, moisture conditions, and scanners) makes the task of minutiae extraction challenging, particularly when approached from a stance that relies on tunable algorithmic components, such as image enhancement. We pose minutiae extraction as a machine learning problem and propose a deep neural network – MENet, for Minutiae Extraction Network – to learn a data-driven representation of minutiae points. By using the existing capabilities of several minutiae extraction algorithms, we establish a voting scheme to construct training data, and so train MENet in an automated fashion on a large dataset for robustness and portability, thus eliminating the need for tedious manual data labelling. We present a post-processing procedure that determines precise minutiae locations from the output of MENet. We show that MENet performs favourably in comparisons against existing minutiae extractors.

1. Introduction

At its core, fingerprint recognition is a pattern recognition problem [1]. Although automatic fingerprint recognition systems have been around for several decades, the problem is still not entirely solved. This is the result of a number of difficulties, both in the problem itself, namely the high intra-class variability (the same fingerprints can look very different between impressions) and high inter-class similarity (two different fingerprints can yield similar features), as well as practical issues including uncooperative data subjects, elastic distortion during scanning, inconsistent moisture conditions, and damaged fingerprints.

Many approaches have been taken to address these problems. A standard fingerprint recognition process typically involves the steps of fingerprint enhancement (including image enhancement and thinning), feature detection, and classification. The most commonly used and standardized feature is that of minutiae points. It is minutiae point extraction that is the focus of our research presented herein.



(a) Ridge Ending (b) Bifurcation

Figure 1. Examples of minutiae points.

Minutiae points are, in a simplified sense, points where fingerprint ridge-lines either end (ridge-endings) or split (bifurcations). These are the features that have an origin and long history of use in forensic science when comparing fingerprints. They are exemplified in Figure 1. Although other types of minutiae points exist, they remain defined by the same pattern (endings or splits). Bansal *et. al* [2] provides a thorough review and explanation of minutiae extraction.

Minutiae detection itself is a difficult problem, subject to the large amount of variation inherent in fingerprint images, the circumstantial variation in the manners in which minutiae points present themselves, and ridge-valley thickness and consistency variations.

As a result, minutiae extraction is difficult to generalize. Moreover, the critical dependency on sub-components of the conventional minutiae extraction process (such as image enhancement), coupled with the aforementioned sources of data-variation, has the unfortunate effect of causing parameter tuning to become unnecessarily important.

These inherent uncertainty-based challenges suggest the opportunity for addressing minutiae extraction as a *learning* problem. Posing it in this manner does away with the requirement for enhancement procedures wherein the solution must be manually optimized for each use case – these can instead be learned from the (sufficiently generalized) fingerprint data itself.

Extending this argument further, we posit that while the approaches taken for minutiae extraction in the past have proven to be functionally useful, it is unclear that the processes followed – such as fingerprint enhancement, binarization, thinning, and conventional minutiae extraction – are indeed optimal. We thus also seek to learn an appropriate minutiae extraction pipeline in a data-driven manner.

To this end, we draw on deep learning approaches, based on recent success in other image recognition tasks and high profile applications [3]. In particular, deep networks have a great capacity for representation learning [4], which directly addresses our desideratum of learning the appropriate lower-level features and steps required for this problem. Thus, instead of carefully tweaking algorithmic components and their accompanying parameters, we propose learning, using deep neural networks, the data-informed representation of minutiae points. As a result, this research endeavours to explore the application of deep learning to minutiae extraction. Moreover, unlike earlier research in this domain, we take the perspective that the amount of, and variation within, training data is paramount to the success and general applicability of a learning approach. Thus, we present an automated labelling procedure that allows for the easy use and augmentation of large fingerprint datasets.

In this paper, we propose a model for minutiae detection based on deep convolutional neural networks and show that it outperforms commercially available systems on standard datasets considering several metrics.

2. Related Work

We are not the first to propose a learning approach to fingerprint recognition. Most notably, Jiang *et. al* [5] presented a two-phase deep convolutional neural network architecture that is applied directly to fingerprint images. The first part of their approach used a neural network, 'JudgeNet', to detect the presence of minutiae in large overlapping regions (63×63 pixels). The output of JudgeNet was a binary classifier. 'LocatNet' was another deep convolutionary neural network designed to accurately locate the pixel location (in a 27×27 pixel centralized box, accurate to 9×9 pixels) within a smaller region (45×45 pixels).

They applied data augmentation by rotating the training patches, blurred exterior regions of patches to remove irrelevant details and improve performance, and performed model averaging (using the same architectures).

Their approach resulted in accurate minutiae extraction in terms of precision, recall, and qualitative assessment. Furthermore, they showed how machine learning can be effectively applied to the minutiae extraction problem. However, there are a number of shortcomings to address.

They performed pre-processing (image normalization) to eliminate the differences between scanning devices. This normalization of the input data, along with the limited diversity and quantity of said input data (200 fingerprints from the Chinese criminal investigation fingerprint database), begs questions of model robustness and overfitting.

Much of this limitation is as a consequence of the effort required to label data as a human being. The impracticality of labelling vast numbers of fingerprints for minutiae is pointed out by the researchers. Conversely, the approach we disclose in this paper is novel in that it uses an automatic labelling procedure that employs the functionality of commercial software. This means that the primary limit to the training data becomes the availability of fingerprints.

The decoupling of human effort in this application has the desirable effect of making it possible to train a neural network toward robustness and portability. This is simply because the input data can be augmented and tailored according to the desired outcomes or evidenced shortcomings. Extending this idea of vast and extensible training data, we incorporate noise and contrast data augmentation that *caters specifically for fingerprint regions of very poor quality*, while Jiang *et. al* stated that the only regions their approach had difficulty with were fringe (i.e., the outer edges of fingerprints) and noisy regions.

The architecture we present herein is similar to the above-mentioned research, but different in that we use a single deep neural network to determine the probability that the central pixel is a minutiae point. Our post-processing procedure is what yields precise locations.

Maio *et. al* [6] used a neural network to filter minutiae after detection to improve the certainty regarding minutiae detection using a different minutiae extractor. Leung *et. al* [7] also used a neural network to extract minutiae on thinned fingerprint images. Gour *et. al* [8] used a neural network to confirm minutiae extraction success.

Tang *et. al* [9] presented an approach for minutiae extraction on latent fingerprints using a fully convolutional neural network. This is a difficult problem in that these images are often occluded by the graphics on the surfaces on which they are found. The results they achieved were comparable to the state-of-the-art which showed how deep neural networks can be used to solve complex problems in this domain. Sankaran *et. al* [10] used stacked autoencoders (a class of neural networks) to detect minutiae in latent fingerprints. This problem and the complexities involved therein, however, are outside of the scope of our research.

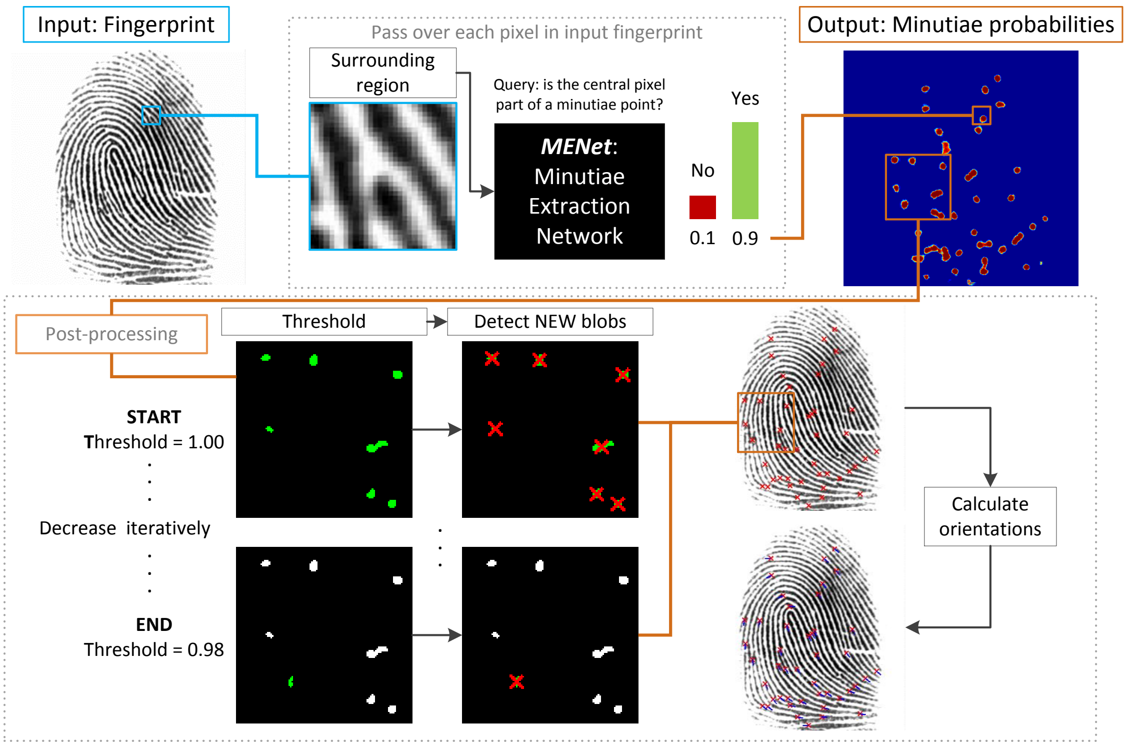


Figure 2. Minutiae extraction using MENet – an overview

Neural network approaches have also been taken regarding fingerprint comparisons, thus evidencing the utility of this technique. Werner *et. al* [11] used a neural network to increase the efficiency of a comparison algorithm. Jea and Govindaraju [12] used a neural network to compare minutiae from partially overlapping fingerprints, while Abdullah [13] used a neural network to compare fingerprints based on a constructed twelve parameter feature vector.

There have been many uses of neural networks in the fingerprint domain. However, certain considerations must be made. The recent popularity of neural networks in many domains has much to do with the availability of large amounts of data, and the advances in training methods and network architectures. That said, much of the fingerprint related research is limited in that they either use pre-processing, train using a very small dataset, or use a shallow network architecture [5]. Considering the recent advances in deep learning, minutiae extraction must be reassessed using methodologies that use a wholly data-driven approach. Paying heed to the need for data-driven perspectives and deeper architectures in the application of neural networks, the research presented in this paper details a manner in which large amounts of fingerprint data can be used to build a robust minutiae extractor.

The following section outlines the approach we have taken to apply deep learning to minutiae extraction.

3. Approach

We term our proposed convolutional neural network model *MENet*, for Minutiae Extraction Network. The contribution of this research is in the conjunction of our model with the automated supervised training procedure and the post-processing. *Moreover, the intention of this paper is to present and expound upon the utility of these learning tools within the pattern-recognition domain of fingerprint biometrics.*

The following section provides an overview of our approach.

3.1. Overview

Figure 2 outlines our proposed methodology. Each pixel of an input fingerprint is processed by considering a window surrounding said pixel (exemplified by the surrounding region highlighted in blue). This window can vary in size but is set according to the architecture of the neural network.

Our MENet model, described in Section 3.2, has as output two *softmax* normalized probabilities corresponding to 'yes' and 'no'. The *softmax* normalization function gives the probability that the central pixel belongs to one of these classes, indicating the predicted presence or absence of a minutiae point, respectively.

The resultant probability map (prior to post-processing, Figure 2) captures an estimate, over the entire fingerprint,

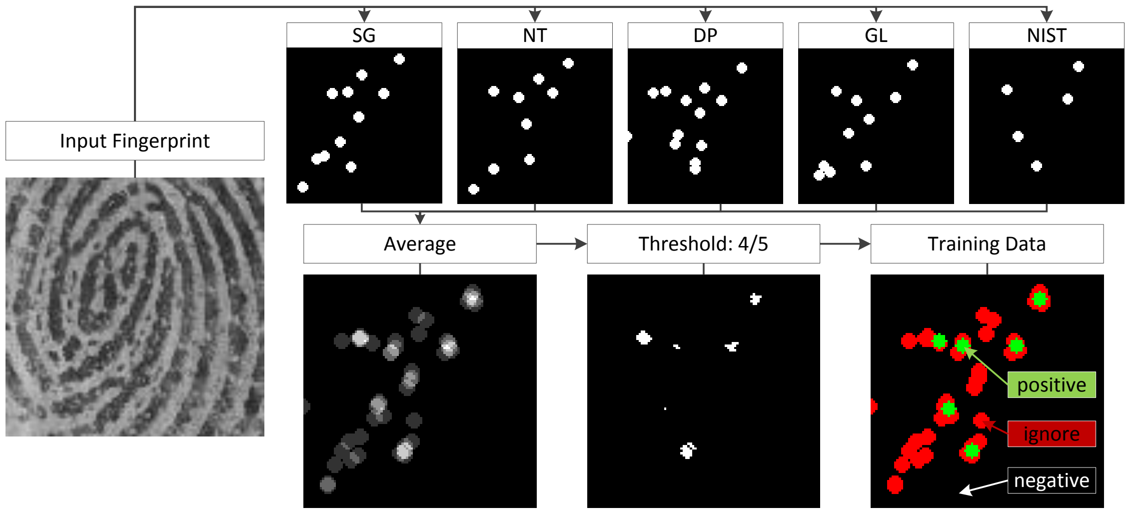


Figure 3. Automated labelling of training data. Note: the displayed is merely a portion of a fingerprint.

of minutiae point positions. In order to capture the precise location of minutiae points from this map, a post-processing procedure is followed (the lower half of Figure 2).

The output probability map is smoothed using a 3×3 median filter. This is thresholded at iteratively lower thresholds in order to extract blobs, the centres of mass of which are the minutiae points. These are shown as red crosses in Figure 2. At the final threshold value, all minutiae with at least that level of certainty will have been detected.**The reason for this is that each detected minutiae point can, in this manner, have a corresponding probability estimation (confidence) at which it was detected**. Simply thresholding at the final (lowest) threshold would result in the same minutiae detection but would not yield any confidence estimates. These could be used as a measure of minutiae quality in future research.

The orientation of these minutiae are calculated by a standard method of local orientation estimation using the principal axis of variation of the image gradients [14]. Minutiae orientations are crucial for fingerprint comparisons. That said, using the simple aforementioned approach suffices since it was used consistently throughout testing which allows for fair evaluations.

The following section describes our proposed deep learning architecture. Section 3.3 then details our approach to collecting training data and the training procedure used.

3.2. The MENet Architecture

As discussed in Section 1, the issue of selecting appropriate intermediate representations (in this case to identify minutiae points from raw pixels) is a generally difficult one. The strength of deep learning has largely been enabled through its success in learning representations, where each layer of such a neural network can be considered a progressively more abstract representation: ranging from low-level input pixels to a high-level classification decision.

MENet was constructed with 5 convolutional layers, followed by 2 fully connected layers of 1024 nodes, and finally a *softmax* output layer. Each convolutional layer consists of 32 5×5 filters, and the first two of these layers use pooling. All units apart from the output layer use ReLU activation functions. The input to MENet was obtained by moving a 30×30 sliding window over the full input fingerprint. TensorFlow [15] was used for implementation.

This architecture was empirically found to outperform shallower networks on the same fingerprint data. A major concern with deep neural networks is that they require large amounts of training data, which in this case involves labelling. Section 3.3 next discusses our approach to automating this procedure, and covers the training of MENet.

3.3. Data Generation and Training

In order to provide optimal conditions when training a deep neural network, the manner in which training data is constructed, organised, and provided is an important consideration. This section details our proposed approach.

First, our training data consisted of all real constituent fingerprints from the 2000 [16], 2002 [17], and 2004 [18] fingerprint verification competitions. This resulted in a database of 7920 fingerprints. These fingerprints encapsulate a wide variety of real world scenarios. That is, some fingerprints are of an exceptionally good quality, while others display severe quality degradation owing to either a damaged fingerprint, a poor scan (since the scanning procedure was not always systematically controlled), wet or dry fingers, or elastic distortion.

Second, training data was derived by applying five commercially available minutiae extractors. These are parts of the following systems/software development kits (SDKs):

- NIST: the National Institute of Standards and Technology's [19] *mindtct*, using a quality threshold of 30% to alleviate the high number of false minutiae detected;
- DP: Crossmatch's DigitalPersona [20] system;
- SG: the SecuGen [21] software development kit (SDK);
- NT: the Neurotechnology [22] SDK; and
- GL: Griaule Biometrics' [23] SDK.

We drew on these tools to drive generation of training data, as detailed in Figure 3 which depicts the process by which regions of a fingerprint are marked as minutiae points. The minutiae locations are extracted from the five aforementioned minutiae extractors. An encompassing circle (for each minutiae point) is drawn on an accompanying 'mask' image for each extractor. These masks are then averaged and thresholded. The threshold is set at $(n-1)/n$ where n is the number of minutiae extractors used.

The thresholded result is closed using image dilation and erosion. Blobs are detected and any separate regions are encircled as minutiae regions for training. The result of this procedure can be seen in Figure 3 as the 'Training Data'. The green circles are used as positive training cases and the black region (the background) is used as negative training cases. The red, however, is ignored during training. It is this functionality of ignoring uncertain regions that allows us to train the neural network to a satisfactory degree. **Note that this procedure requires no manual labelling.** Although better approaches to combining minutiae from various sources may exist, the technique described herein is simple and yields reliable training data for this research.

For training, batches are constructed by first randomly selecting a fingerprint and then randomly selecting locations from it. The method above is used to define the corresponding labels. An equal number of positive and negative training cases are always provided for unbiased training.

In order to improve the robustness of the model trained on this data, **data augmentation** was used. A pseudo-random selection of grayscale colour inversion, contrast degradations, contrast improvements, and noise degradations were applied to every fingerprint selected during training. The parameters for this augmentation were also randomly selected within limits.

The input data (the FVC datasets) was split into training (80%) and testing (20%) sets. Training using the backpropogation algorithm with stochastic gradient descent continued with a batch size of 500 and 50% dropout for approximately 1.3 million epochs. Owing to the similarity of minutiae between both training and testing data, there was no sharp testing-accuracy loss over time. Thus, when convergence was maintained for a sufficient period, training was stopped.

4. Experimental Results and Discussion

This section exhibits the results obtained for minutiae extraction. Three experiments were carried out, all of which were designed to test the accuracy of minutiae location detection using our model, MENet, compared to the five commercial minutiae extractors:

1. *Minutiae extraction versus ground truth minutiae*. A set of 100 fingerprints were randomly chosen from the testing dataset and minutiae point locations were manually extracted. This is discussed in Section 4.1.
2. *Biometric performance evaluation*. Minutiae were extracted and compared from two FVC datasets. This is detailed in Section 4.2.
3. *Qualitative assessment*. Two examples were chosen from the above-mentioned hand-labelled dataset in order to understand under which circumstances minutiae extraction succeeds and/or fails. This is shown and discussed in Section 4.3.

4.1. Minutiae Extraction vs. Ground Truth

A set of 100 fingerprints were hand-labelled for this assessment. This set was randomly selected from the testing dataset (a subset of the collated FVC dataset) and contained both good and poor quality fingerprints. These fingerprints were carefully individually labelled, yielding a total of 4730 minutiae points.

Each minutiae extractor (i.e., the commercial extractors and MENet) was used to process the same fingerprints and comparisons were made (manually/visually by us against the ground truth) in order to determine: (1) the number of correct minutiae extracted relative to the ground truth, (2) the number of minutiae missed relative to the ground truth; and (3) the number of false minutiae. This is presented in Tables 1 and 2.

Table 1. Minutiae Extraction Results. Red highlights the worst results and green highlights the best results achieved.

Input	**Correct %**	**Missed %**	**False %**	**EER: 2002**	**EER: 2004**
	of the ground truth				
NIST	67.1	32.9	22.7	2.778	15.075
DP	74.1	25.9	21.3	1.181	6.072
SG	74.9	25.1	16.2	1.191	8.876
NT	74.2	25.8	15.9	1.299	8.986
GL	65.0	35.0	19.7	2.292	19.766
MENet	85.8	14.2	18.6	0.781	5.450

Table 2. NFIQ breakdown. Red highlights the worst results and green highlights the best results achieved.

NFIQ	Input	Correct % of the ground truth	Missed % of the ground truth	False %
1 (45% of data)	NIST	68.6	31.4	16.2
	DP	75.2	24.8	15.6
	SG	76.0	24.0	10.6
	NT	74.7	25.3	11.1
	GL	65.8	34.2	13.1
	MENet	86.8	13.2	14.0
2 (33% of data)	NIST	68.8	31.2	21.5
	DP	76.6	23.4	19.1
	SG	74.5	25.5	13.9
	NT	74.9	25.1	14.4
	GL	66.9	33.1	17.4
	MENet	86.8	13.2	18.3
3 (15% of data)	NIST	61.1	38.9	40.8
	DP	66.2	33.8	35.1
	SG	75.0	25.0	29.8
	NT	75.0	25.0	21.7
	GL	61.9	38.1	36.0
	MENet	83.5	16.5	28.4
4 (2% of data)	NIST	47.5	52.5	40.0
	DP	60.4	39.6	52.0
	SG	61.4	38.6	28.7
	NT	59.4	40.6	44.4
	GL	45.5	54.5	53.1
	MENet	72.3	27.7	41.1
5 (5% of data)	NIST	62.1	37.9	40.5
	DP	72.7	27.3	40.6
	SG	69.6	30.4	44.0
	NT	67.1	32.9	46.0
	GL	59.0	41.0	39.5
	MENet	77.0	23.0	33.0

As is shown in Table 1, MENet performed favourably when compared to the ground truth minutiae. It only missed 14.2% of these, while the closest competitor – SecuGen – missed 25.1%. Although it was not the top performer with regard to false minutiae detection, it still managed to yield a success rate above 80%.

Table 2 gives a breakdown of the minutiae detection accuracy grouped by the NIST NFIQ score. NFIQ is a quality assessment ranging from 1 (best) to 5 (worst). MENet performed best regarding correctly detected minutiae in all cases. What is most noteworthy here is the accuracy regarding poor-quality fingerprints. MENet was still able to detect 77.0% of the minutiae in the poorest quality fingerprints. These test cases were randomly selected.

The following section details how this minutiae extraction success relates to biometric comparison performance.

In order to ensure that this assessment remained objective, only the locations of the extracted minutiae varied. Locations were extracted using the commercial extractors and MENet. The orientation was calculated [14] and a minutiae file was constructed. This was then used to compare fingerprints in the FVC 2002 DB1A [17] and FVC 2004 DB1A [18] databases. These databases consist of 100 fingers with 8 impressions for each. The former consists of better quality fingerprints and is generally considered an 'easier' database to test against. The latter contains dry and wet impressions as well as distorted impressions, making it a more challenging database regarding biometric evaluation. Testing was limited to these two databases for space considerations and because they are sufficiently diverse.

The Minutiae Cylinder Code [24] (MCC) [1] was used to perform these assessments. MCC is robust against false minutiae. A comparison protocol [25] was followed in order to calculate the True Positive and False Positive Rates. Each impression was compared against all impressions of the same finger for 2800 genuine comparisons and the first impression of each fingerprint was compared against all other first impressions for 4950 impostor comparisons.

It must be noted at this stage that NT failed to extract minutiae in 1 case for the 2002 DB1A database, and in 18 cases for the 2004 DB1A database. This resulted in 7 missing genuine comparisons on the 2002 DB1A database. For the 2004 DB1A database, 120 genuine comparisons and 99 impostor comparisons were consequently missing. These 'failed to enroll' cases were accounted for during experimentation by making the score equal to zero.

The consequent receiver operative curves are presented in Figures 4 and 5. These are presented with a log-scale for the False Positive Rate in order to better accentuate the differences between algorithm performance. In both cases MENet performed best. Furthermore, Table 1 shows that the equal error rates (EERs) yielded by MENet are superior to those yielded by the other minutiae extractors.

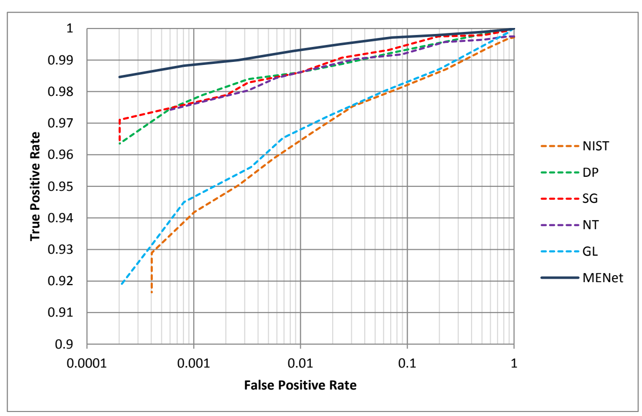


Figure 4. Receiver operating curves: FVC 2002, database 1A

[1] MCC SDK from http://biolab.csr.unibo.it/mccsdk.html

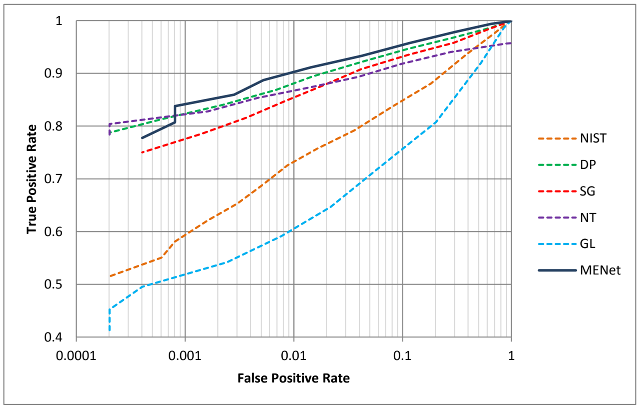


Figure 5. Receiver operating curves: FVC 2004, database 1A

4.2. Biometric Evaluation

The improved accuracy of MENet had the consequence of a higher number of true minutiae detected. This evidently improved the performance of MCC. It must be noted, however, the standardization of minutiae orientation detection (to a range of $0° - 180°$ [14]) may have a negative impact on the performance in all cases. Additional features such as minutiae type may also improve comparisons. That, however, is not the focus of this research. Instead, we are attempting to best detect the presence and consequent location of minutiae points.

Although the two test databases were involved in training MENet, only a *maximum of 80% of possibly augmented data* was seen. Moreover, owing to the voting system used for data generation, the majority of challenging minutiae would never have been seen. Nevertheless, future work will involve testing on databases never seen during training. Moreover, it may even be advisable to create new testing datasets using scanners that were not used for the training data.

4.3. Qualitative Assessment

This section contains two fingerprint examples and the result of minutiae extraction. Figure 6 exhibits minutiae extraction on a poor quality fingerprint, while Figure 7 exhibits minutiae extraction on a good quality fingerprint.

Considering the minutiae extracted by NIST and DP on the poor quality fingerprint (Figures 6(a) and (b), respectively), it is evident that dryness has an impact in the success of these extractors. MENet was able to detect the highest number of correct minutiae in this case.

The minutiae extraction on the good quality fingerprint serves to exemplify the tension between undetected minutiae and falsely detected minutiae. For instance, while NIST only missed four minutiae, it detected many false minutiae. MENet performed best in this case, with the lowest number

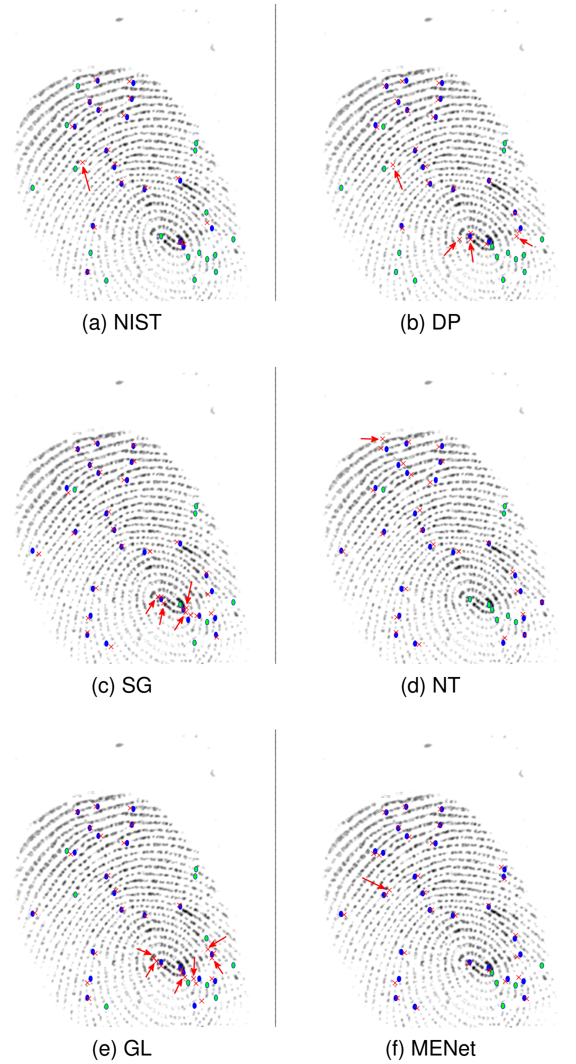
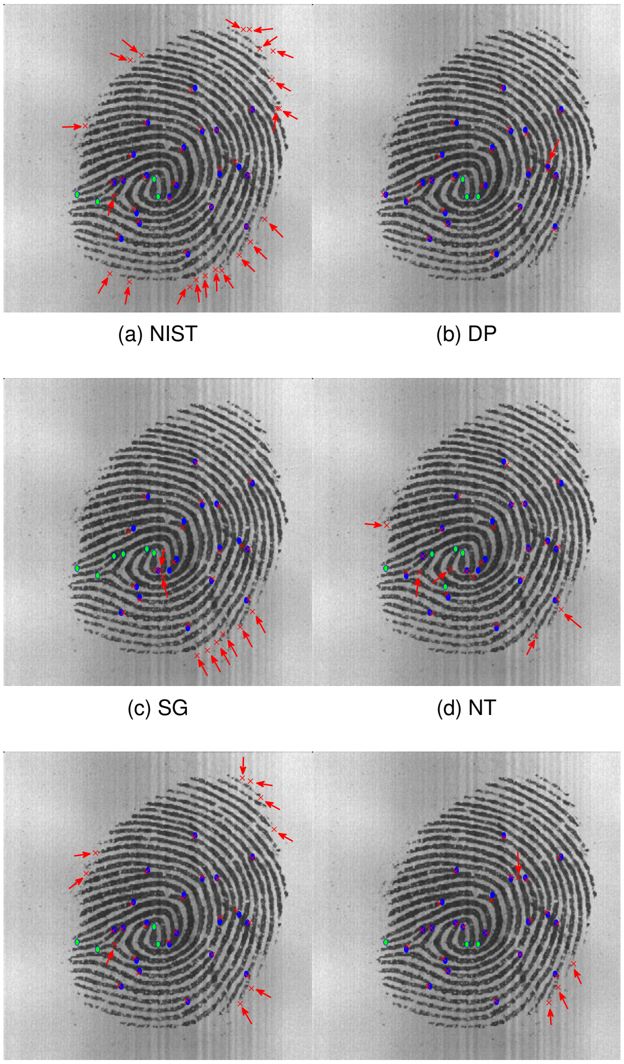


Figure 6. Minutiae Extraction, example 1: a dry fingerprint exhibiting poor quality regions. The circles are the ground truth minutiae: blue circles are detected, and green circles are the undetected minutiae. The red arrows indicate false minutiae.

of undetected and false minutiae. An ideal situation would be the co-minimization of both false and undetected minutiae, but this is a very challenging task. Future work may see an improved post-processing procedure for MENet that accounts for local certainty and quality measures when determining blob thresholds (see Figure 2).

5. Conclusion

This paper addressed the problem of minutiae extraction for fingerprint comparison, by posing it as a machine learning problem *and providing training data in an automated fashion*. To this end we developed MENet: a deep convolutional neural network model that was shown to outperform



(a) NIST (b) DP

(c) SG (d) NT

(e) GL (f) MENet

Figure 7. Minutiae Extraction, example 2: a good quality fingerprint. The circles are the ground truth minutiae: blue circles are detected, and green circles are the undetected minutiae. The red arrows indicate false minutiae.

commercially available algorithms on benchmark datasets using several standard assessment criteria.

With superior equal error rates on two FVC datasets and a minutiae miss-rate of only 14.2%, the minutiae extraction model presented in this research exhibits detection accuracy that may serve to augment and improve the existing fingerprint identification pipeline.

Moreover, and possibly most importantly, the approach taken in this research has been one that utilizes the strengths of existing algorithms and the availability of large real-world fingerprint datasets. By learning directly from the data, robust and accurate models can be learned to either emulate or improve upon existing solutions. By collating the 2000, 2002, and 2004 FVC datasets and using an automated voting system for minutiae labelling prior to training, the tedious task of manually labelling minutiae was circumvented. This is significant in that it causes the primary limitation on training data (which has a direct impact on model performance) to be the availability of fingerprints, and not the availability of time of human experts.

Although the speed of neural networks may concern the reader, it must be noted that the purpose of this research is to show the utility of neural networks in this domain. Moreover, speed and efficiency improvements are actively being researched. Network binarization or quantization [26], distilling the knowledge in a neural network [27], and speed-ups gained through the use of widely available GPU implementations are options for future implementations of MENet.

Future research will explore other aspects of fingerprint identification and comparison that are equally easily posed as machine learning problems.

Regarding the minutiae extraction model presented herein, future work will be undertaken to: (1) improve the post-processing procedure by local quality considerations; (2) derive a local quality estimation based on the variation patterns of the minutiae probability maps; (3) determine whether orientation information is inherently encoded within the the trained neural network; and (4) use MENet to solve other fingerprint-related problems such as fingerprint masking.

Finally, we will investigate addressing the full fingerprint comparison problem using deep learning alone. To this end, other techniques in neural network research such as Siamese networks [28] and attention modelling [29] will be investigated.

References

[1] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[2] R. Bansal, P. Sehgal, and P. Bedi, "Minutiae extraction from fingerprint images – a review," *arXiv preprint arXiv:1201.1422*, 2011.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[4] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[5] L. Jiang, T. Zhao, C. Bai, A. Yong, and M. Wu, "A direct fingerprint minutiae extraction approach based on convolutional neural networks," in *Proceedings of the International*

Joint Conference on Neural Networks. IEEE, 2016, pp. 571–578.

[6] D. Maio and D. Maltoni, “Neural network based minutiae filtering in fingerprints,” in *Proceedings of the Fourteenth International Conference on Patter Recognition*, vol. 2. IEEE, 1998, pp. 1654–1658.

[7] W. Leung, S. Leung, W. Lau, and A. Luk, “Fingerprint recognition using neural network,” in *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*. IEEE, 1991, pp. 226–235.

[8] B. Gour, T. Bandopadhyaya, and S. Sharma, “Fingerprint feature extraction using midpoint ridge contour method and neural network,” *International Journal of Computer Science and Network Security*, vol. 8, no. 7, pp. 99–103, 2008.

[9] Y. Tang, F. Gao, and J. Feng, “Latent fingerprint minutia extraction using fully convolutional network,” *arXiv preprint arXiv:1609.09850*, 2016.

[10] A. Sankaran, P. Pandey, M. Vatsa, and R. Singh, “On latent fingerprint minutiae extraction using stacked denoising sparse autoencoders,” in *Proceedings of the International Joint Conference on Biometrics*. IEEE, 2014, pp. 1–7.

[11] G. Á. Werner and L. Hanka, “Tuning an artificial neural network to increase the efficiency of a fingerprint matching algorithm,” in *Proceedings of the 14th International Symposium on Applied Machine Intelligence and Informatics*. IEEE, 2016, pp. 105–109.

[12] T.-Y. Jea and V. Govindaraju, “A minutia-based partial fingerprint recognition system,” *Pattern Recognition*, vol. 38, no. 10, pp. 1672–1684, 2005.

[13] H. A. Abdullah, “Fingerprint identification systems using neural network,” *Nahrain University, College of Engineering Journal*, vol. 15, no. 2, pp. 234–244, 2012.

[14] P. Kovesi. Fingerprint Image Enhancement and Minutiae Extraction. Accessed on 04 April 2017. [Online]. Available: http://www.peterkovesi.com/matlabfns

[15] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.

[16] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, “FVC2000: Fingerprint verification competition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 402–412, 2002.

[17] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. K. Jain, “FVC2002: Second fingerprint verification competition,” in *Proceedings of the 16th international conference on Pattern recognition*, vol. 3. IEEE, 2002, pp. 811–814.

[18] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain, “FVC2004: Third fingerprint verification competition,” *Biometric Authentication*, pp. 31–35, 2004.

[19] National Institute of Standards and Technology. Accessed on 04 April 2017. [Online]. Available: http://nist.gov

[20] Crossmatch. Digital Persona. Accessed on 04 April 2017. [Online]. Available: http://www.crossmatch.com

[21] SecuGen Corporation. Accessed on 04 April 2017. [Online]. Available: http://www.secugen.com

[22] Neurotechnology. Accessed on 04 April 2017. [Online]. Available: http://www.neurotechnology.com

[23] Griaule Biometrics. Accessed on 04 April 2017. [Online]. Available: http://www.griaulebiometrics.com

[24] R. Cappelli, M. Ferrara, and D. Maltoni, “Minutia cylinder-code: A new representation and matching technique for fingerprint recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2128–2141, 2010.

[25] FVC 2004 Performance Evaluation. Accessed on 05 April 2017. [Online]. Available: http://bias.csr.unibo.it/fvc2004/perfeval.asp

[26] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, “Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1,” *arXiv preprint arXiv:1602.02830*, 2016.

[27] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.

[28] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, “Signature verification using a ”siamese” time delay neural network,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 4, pp. 669–688, 1993.

[29] S.-h. Zhong, Y. Liu, Y. Zhang, and F.-l. Chung, “Attention Modeling for Face Recognition via Deep Learning,” in *Proceedings of the Cognitive Science Society*, 2012, pp. 2609–2614.