Review on ”SP-NET: One Shot Fingerprint Singular-Point Detector”

Anand Tadepalli, Anshuman Singh

April 2021

1 Introduction

Authentication has always been a problem ever since information became important to our society. This is to ensure that the person who would like to access certain information is really who he/she claims to be. This has commonly been solved by using an password or a pin which must be entered by the user to authenticate themselves. This method has its own problems such as leaking of the password, forgetting the password, etc. One powerful alternative is the usage of bio-metrics, which are unique to an individual. Features such as fingerprint, iris, palm and other features are not only unique to each person but cannot be lost.

Fingerprints are impression left by the ridges of a human finger on a surface. They are unique to every individual and unlike other security very tedious to replicate and they cannot be forgotten, so fingerprint authentication has proven to be more secure and convenient. Fingerprints have been used to identify people for decades, long before the development of computers and even though different bio-metric technologies have been developed over the years fingerprint recognition is most commonly used because of its lower price.

A Singular-point refers to the area of a fingerprint where the curvatures of the ridges change rapidly. This can also be attributed to being the center or core of the fingerprint as it is usually the point around which the ridges circulate around. With its curvature properties, Singular point plays a major role in feature extraction as it is the basis of minutiae matching. Hence the importance for their correct detection.

This aim of this paper is to verify the results published by the paper “SP-NET: One Shot Fingerprint Singular-Point Detector” [4] which uses two merged networks viz: Macro-Localization Network (MLN) and Micro-Regression Network (MRN). Additionally, MLN is an encoder-decoder network along with three stacked hourglasses as a bottleneck.

2 Related Work

This paper [6], presents a mask that locates the core point simply from the ridge orientation map. The introduced algorithm detects the core point at the end of the discontinuous line appearing in the orientation map presented by a grey-scale. A property is presented and supported with a mathematical proof to verify that the singular regions are located at the end of this discontinuous line. The proposed core point detection algorithm in [4] follows four steps: normalization, ridge orientation estimation, smoothing and core point detection. The fingerprint images are acquired by the scanner that is not good quality images. The discrepancy in gray-level besides the ridges and valleys is inaccurate because the quality of an images is poor.

The [10] proposed the normalization process is widely accepted method. The image smoothing is done after normalization using Gaussian smoothing technique [10] based on the orientation consistency. By smoothing, accuracy of the orientation map and CP detection is increased. The process of the algorithm is: Compute the orientation base on image gray gradient and convert the orientation filed into a continuous vector filed. Using Gaussian low pass filter generate new Orientation filed with the result. To verify or identify an individual local features are used while the global features describe local orientation of ridges and valleys using orientation fields. In fingerprint recognition orientation estimation process plays vital role. The orientation field indicate the direction of ridges. The impact of orientation estimation process is to affect the other succeeding processes like image enhancement, singular point detection and classification of an image and matching. The classification of fingerprint is done using core point detection based on the global patterns of ridges. It is useful in matching when finger image is rotated. For that in global pattern core and delta points are used to rotate the image. The Orientation Field, Multi-Resolution, Curvature and Poincare Index are existing core point detection algorithms mentioned in [5]. The existing methods doesn’t provide accurate and estimated outcome for noisy images.

A technique to detect singular point utilizing a sliding window by viewing local region of the image has been proposed in [13]. A window of fixed size, slides through the fingerprint image, to locate the singular point in a partially extracted fingerprint image. The paper uses a complex filter core location method [5] instead of Poincare Index. The approach on FVC2002 database take 0.2 seconds lesser than a procedure that uses the whole fingerprint image for singular point detection.

[15] proposes a one-stage detector called Real One-Stage Effort (ROSE) to detect fingerprint singular points. It uses multi-scale spatial attention, a Gaussian heatmap and a variant of focal loss are applied together.

In [7], the author proposes a new customized Convolutional Neural Network called SinNet,for segmenting the accurate singular point area, differing from other detection methods, treats singular points detection as a semantic segmentation problem and uses only a few samples for training. The network extracts the singular regions of interest and then use a blob detection (SimpleBlobDetector) to locate the singular points quickly. The experiments are carried out on the test dataset

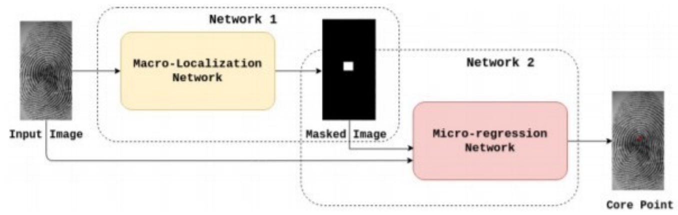


Figure 1: Block Diagram of Proposed SP-Net (dotted lines shows logical parts)

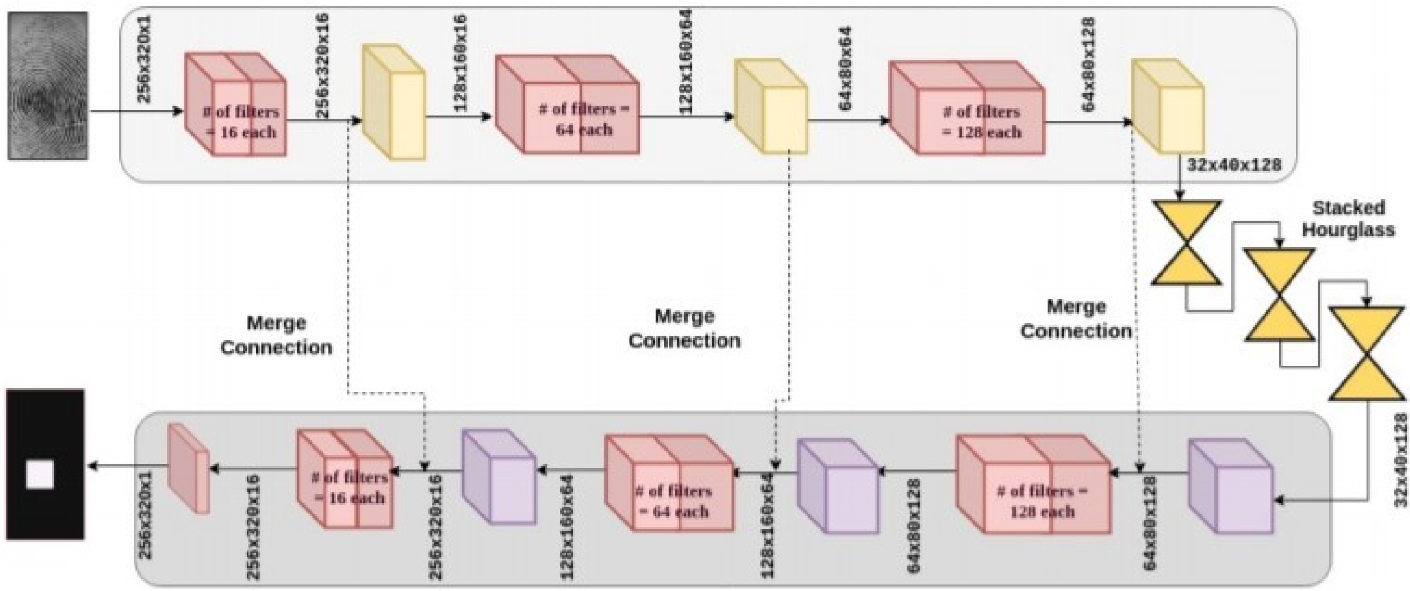


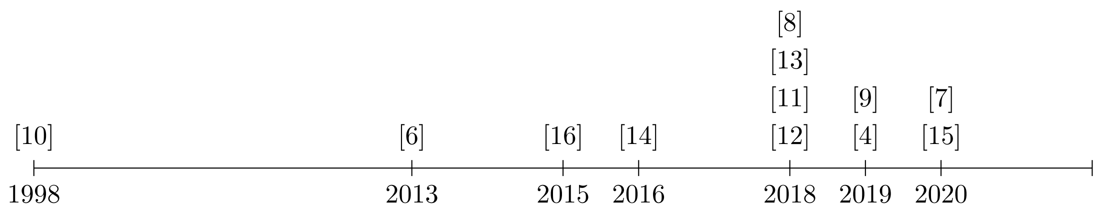
Figure 2: Architecture of Proposed Macro Localization Network

from SPD2010. SPD2010 benchmark database [3] has been which has 500 fingerprint images with 355×390 pixel resolution captured by an optical scanner (Microsoft Fingerprint Reader - model 1033).

This paper [9], proposes an adaptive method to detect Singular-points. The algorithm consists of three stages. First, the image is enhanced by removing the background of the fingerprint using singular value decomposition. Next, blurring detection and boundary segmentation is used to detect the region of impression. Finally, an adaptive method based on wavelet extrema and the Henry system for core point detection is proposed. Experiments conducted using the FVC2002 DB1 and DB2 databases [1].

In this study [12], uses Faster-RCNN with an orientation constraint, in a two-step process, to obtain orientation information of singular points. The paper also designs a convolutional neural network (ConvNet) for singular points detection according to the characteristics of fingerprint images and the existing works helping the proposed method to extract the singular-point without preprocessing. this trains and tests on FVC2002 DB1 and NIST SD4 datasets [2].

In this paper [11],the proposed method claims to efficiently calculate fingerprint blocks orientation using the pretrained neural network. The same neural network is applied again to define singular points at pixel level. Although the training may be time consuming, the authors claim that the method outperforms algorithms that calculate pixel level orientation in real time. In addition, the proposed model is rotation insensitive and testing was performed on NIST-29 database.



3 Proposed Methodology

There are two major components in this model that are stacked one over the other. The first component applies segmentation to find the probable region of singular point, called the Macro Localization Network. The second component applies regression to identify the coordinate of the singular point on the fingerprint from the segmented area. The advantage of the proposed model lies in its ability to determine the location of the singular point efficiently in one pass, from the input image to the output coordinates. A block diagram depicting the proposed model is given in ¡Reference diagram of the complete network¿.

Macro Localization Network (MLN). It is an encoder-decoder network based on the U-Net architecture [16], that maps the input fingerprint image to an image containing a highlighted region having the highest probability of containing the

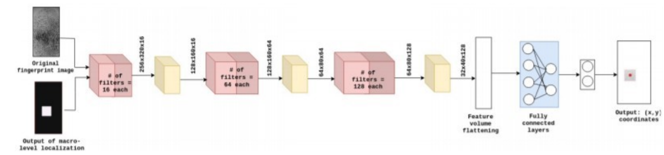


Figure 3: Architecture of Proposed Micro Regression Network

singular point. The encoder consists of 3×3 convolutional layer of filter sizes of 16, 64 and 128. Every pair of convolutional layer is followed by a 2×2 max-pool layer. The encoder serves as a feature extractor. At the end of the encoder, a bottleneck is introduced in the form of three-stacked hourglass network which feeds into the decoder. An hourglass network [14] has the ability to capture features at multiple scale and combining them to make pixel-wise predictions. This is made possible by the use of skip-connections that conserve information at every scale. An hourglass network consists of a series of convolution layers followed by a max-pool. This is repeated to process features till a very low scale. The lowest resolution features are then upsampled. Skip connections are introduced to merge the feature maps at different scales. The architecture of the single hourglass network is separately shown in ¡reference to hourglass diagram¿. Adding hourglass modules one after the other allows for re-assessment of the features across the whole image by going to-and-fro between the scales. This helps in conserving spatial relationship among features resulting in better and robust localization, especially in the noisy images. The output of the bottleneck is passed to the decoder that performs segmentation on the input image. The decoder first performs up-sampling using a 3×3 transposed convolution layer. Its output is concatenated with the corresponding feature map of the encoder using merge connections. The merge connections preserve the essential spatial information of the input image that may have lost in the encoder. The concatenated feature map is passed to a pair of 3×3 convolution layer followed by ReLU activation. This is repeated for different filters of size 128, 64 and 16. The complete architecture of the network is shown in ¡ref to diagram of MLN¿. Here red, yellow and violet blocks represent 3×3 convolution, 2×2 max-pooling, and 3×3 up-sampling blocks respectively.

Micro Regression Network (MRN). This network performs regression on the proposed region of interest by taking the original fingerprint image along with the output of the MLN i.e. image containing a probable region for singular point proposal, as input. The inputs are passed through three convolutional blocks with varying filter size 16, 64 and 128 wherein, every block is followed by ReLU activation function and a 2×2 max-pool layer. This is followed by a flattening and four fully connected layers. The last fully connected layer outputs two values representing the predicted (x, y) coordinates of the singular point. The architecture of the Micro Regression network is graphically shown in ¡ref to diagram of MRN¿.

All these component networks are stacked one over the other after training them individually to build a single network that predicts the location of the singular point in a given fingerprint image. For training the models, cross entropy loss is back-propagated by the macro-localization network to learn the proposed region for singular point presence while mean squared error is back-propagated to the regression network for learning singular point localization.

4 Our work and procedure

4.1 Fixing Compatibility Issues

The code that was provided for the purpose of this work, was using some python methods that were no longer supported by the frameworks it was written on. To make the code run, we were required to understand and make the appropriate corrections to the code for it to be compatible with current versions of the frameworks. The code-base being written in python 2, had to be rewritten into corresponding python 3 code. This was necessary because the machines that we had available no longer supported python 2.

4.2 Code Modifications for testing

In addition to the above mentioned changes, we had to modify parts of the code to enable us to test the code properly, such as changing the testing output data, changing filenames and directory paths, and changing save file configurations. There was some prior testing code that also had to be removed since that code seemed to be hampering our ability to review the paper.

4.3 Data-set Cleaning

After the appropriate modifications to the training code, we had to clean the data-sets that was provided to us as the data-sets were very inconsistent and had many missing files as well as many empty files. These files created lots of inconsistencies further down the pipeline such as lack of control over test and training sizes and the code has been written assuming the data-sets are consistent. For this we created a script to remove all files that were missing its ground truth or its data.

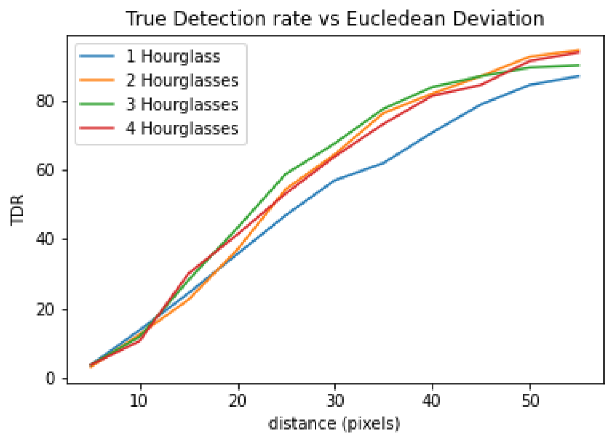


Figure 4: Caption

4.4 Training and Testing

With the data-sets and model code-base ready, we performed the training. This required us to utilize the services of Google Colab for more powerful CPUs and GPUs. We trained the model at its default setting of 100 epochs, with Adam optimizer and a learning rate 1e-5. With the varying sizes of the data-sets, we found that the model would reach a saturation point earlier rather than later in some cases and was appropriate to stop the training once the losses were stable for 10 epochs continuously. Additionally, for training and testing split, we split the FVC2002 data-set by choosing 2 random images per subject (8 images per subject, with 100 subjects) for testing and the rest for training. This method of data split could not be done with the other data-sets as there were too many missing data points in them. Hence, we chose to split the training and testing data into 80For testing, we evaluated the data on the True Detection Rate (TDR),

5 Results

5.1 Data-sets Used

FVC2002 DB1. It is a public database [1] widely used for fingerprint recognition benchmark. The database was originally used to host fingerprint verification competition. It contains 800 fingerprint images collected from 100 subjects. Each subject has provided eight fingerprint impressions of the same finger. The databases have been collected using an optical sensors namely, TouchView II by Identix. There exists huge intra-class variation among different fingerprint impressions of the same subject due to a difference in placement, pressure etc.

Sec 1 & 2, Fut 1 & 2, Lum 1 & 2 = FPL30K. This is an in-house database that has been collected from 855 subjects using three different sensors. Subjects are from rural population, involved in laborious field work. Each subject has provided nearly three fingerprint impressions of two of their fingers on each scanner in two phase having gap of two months. Ground truth has been manually generated for all 30,000 fingerprint images.

5.2 Evaluation Parameter

True Detection Rate (TDR) has been taken as a measurement index to gauge the performance of the proposed model. The proposed model outputs (x, y) coordinates of the singular point corresponding to the fingerprint sample. The extracted singular point will be considered as a true singular point if the euclidean distance between the original and predicted coordinate is less than or equal to the given pixels. This can be formulated as given in Equation below, where ((Cp) x, (Cp) y) and ((Ca) x, (Ca) y) refer to the (x, y) coordinates of the predicted and ground truth singular point respectively.

$\sqrt{((C_p)^x - (C_a)^x)^2 + ((C_p)^y - (C_a)^y)^2} ipixels$

where i=5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55

5.3 Experimental Setup

The implementation of the proposed network has been done on a Google Colaboratory with graphics memory of 12 GB. The model has used the Adam optimizer with a learning rate of 0.0005. The network has been trained for 100 epochs with a mini-batch of 8 images (each of size 256×320).

6 Conclusion

We have verified that the usage of a 3 stacked hourglass networks in the Macro Localization is the optimal number, as it provides a higher accuracy and that using more or less hourglass networks.

[15]	2020	ROSE	True Detection Rate (<10) and Average Inference speed	FVC2002 DB1 (800 images) and NIST SD04 (4000 images).	TDR [NIST SD04 (cores)93.5 (deltas)95.1] [FCV2002 DB1 (cores)97.1 (deltas)98.6] Speed [20ms]
[7]	2020	SinNet	True Detection Rate (<10)	SPD2010(210 training and 290 testing	TDR[SPD2010 (corrected detection rate) 48 (cores) 68 (deltas) 63]
[9]	2019		True Detection Rate (<10)	FVC 2002 DB1 and DB2 databases	TDR[FVC 2002 DB1 (corrected detection rate) 90.72 (cores) 92.43 (deltas) 97.25][FVC 2002 DB2 (corrected detection rate) 89.92 (cores) 95.54 (deltas) 95.21]
[12]	2018		True Detection Rate (<10) while the difference between the predicted orientation predicted and the g truth is less than 20$\circ$	FVC2002 (800 images) [24], NIST sd04 (4000 images) [25] and Ten-Finger Card dataset (62,655 images) from a laboratory database	method achieves 96.03% detection rate for core points and 98.33% detection rate for delta points on FVC2002 DB1 dataset while 90.75% for core points and 94.87% on NIST SD4 dataset, which outperform other algorithms.

Hourglass	Pixel distance										
	5	10	15	20	25	30	35	40	45	50	55
1	3.75	13.75	24.375	35.625	46.875	56.875	61.875	70.625	78.75	84.375	86.875
2	3.125	12.5	22.5	36.875	54.375	64.375	76.25	81.875	86.875	92.5	94.375
3	3.75	11.875	28.125	43.125	58.75	67.5	77.5	83.75	86.875	89.375	90.0
4	3.75	10.625	30.0	41.25	53.125	63.745	73.125	81.25	84.375	91.25	93.75

Table 1: Table shown below shows the comparison between using different number of hourglass stacked in MLN

DatasetDatasetDatasetpt¡ -Datasetpt¿	Pixel distance										
	5	10	15	20	25	30	35	40	45	50	55
fcv2002	3.75	11.875	28.125	43.125	58.75	67.5	77.5	83.75	86.875	89.375	90.0
FUT Phase1	0.37	2.23	6.59	11.61	16.99	22.47	27.86	33.24	38.07	42.80	47.17
FUT Phase2	0.45	2.18	4.67	8.80	13.09	17.08	21.82	25.13	29.87	34.46	38.98
LUM Phase1	1.2	5.7	10.8	18.4	25.3	33.8	40.7	47.0	52.3	56.3	60.2
LUM Phase2	1.28	5.77	12.29	20.09	24.89	31.73	38.46	44.34	49.47	54.91	60.04
SEC Phase1	0.75	2.35	3.53	5.14	7.07	9.02	11.35	14.24	18.42	22.06	25.59
SEC Phase2	0.63	2.03	4.69	8.76	13.46	18.15	21.60	25.35	29.73	34.27	37.56

Table 2: Cross-dataset testing on model trained on fvc2002

References

[1] The fvc2002 database. http://bias.csr.unibo.it/fvc2002/default.asp.

[2] Nist sd4 datasets. https://www.nist.gov/srd/nist-special-database-4.

[3] Spd2010 benchmark database. https://paginas.fe.up.pt/ spd2010/.

[4] Geetika Arora, Ranjeet Jha, Akash Agrawal, Kamlesh Tiwari, and Aditya Nigam. Sp-net: One shot fingerprint singular-point detector. 08 2019.

[5] Ayman Mohammad Bahaa-Eldin. A medium resolution fingerprint matching system. *Ain Shams Engineering Journal*, 4(3):393–408, 2013.

[6] G.A. Bahgat, A.H. Khalil, N.S. Abdel Kader, and S. Mashali. Fast and accurate algorithm for core point detection in fingerprint images. *Egyptian Informatics Journal*, 14(1):15–25, 2013.

[7] Jiong Chen, Heng Zhao, Zhicheng Cao, Fei Guo, and Liaojun Pang. A customized semantic segmentation network for the fingerprint singular point detection. *Applied Sciences*, 10(11), 2020.

[8] Xifeng Guo, En Zhu, and Jianping Yin. A fast and accurate method for detecting fingerprint reference point. *Neural Comput. Appl.*, 29(1):21–31, January 2018.

[9] Ngoc Tuyen Le, Duc Huy Le, Jing-Wein Wang, and Chih-Chiang Wang. Entropy-based clustering algorithm for fingerprint singular point detection. *Entropy*, 21(8), 2019.

[10] Lin Hong, Yifei Wan, and A. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):777–789, 1998.

[11] L. Liu. Fingerprint analysis and singular point definition by deep neural network. *International Journal of Machine Learning and Computing*, 8:625–629, 12 2018.

[12] Yonghong Liu, Baicun Zhou, Congying Han, Tiande Guo, and Qin Jin. A method for singular points detection based on faster-rcnn. *Applied Sciences*, 8:1853, 10 2018.

[13] Xiangshen Min, Xuefeng Zhang, and Fang Ren. Fingerprint core location algorithm based on sliding window. *Wuhan University Journal of Natural Sciences*, 23:195–200, 06 2018.

[14] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation, 2016.

[15] Liaojun Pang, Jiong Chen, Fei Guo, Zhicheng Cao, and Heng Zhao. Rose: Real one-stage effort to detect the fingerprint singular point based on multi-scale spatial attention, 2020.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

Dataset testedDataset testedDataset testedpt¡ -Dataset testedpt¿	Pixel distance										
	5	10	15	20	25	30	35	40	45	50	55
fcv2002	3.75	11.875	28.125	43.125	58.75	67.5	77.5	83.75	86.875	89.375	90.0
30K	10.537	29.748	49.534	65.273	76.503	84.584	89.870	92.953	94.799	96.205	97.052
FUT Phase1	13.09	33.52	50.79	66.02	78.18	84.31	90.25	93.31	94.80	96.38	97.77
FUT Phase2	35.289	66.29	76.82	83.52	87.81	90.519	92.927	94.206	95.33	96.237	97.516
LUM Phase1	53.0	78.5	87.2	91.0	93.4	96.3	97.2	98.2	98.9	99.1	99.1
LUM Phase2	50	83.547	89.85	92.949	94.872	95.299	96.474	97.65	98.077	98.398	98.825
SEC Phase1	16.059	39.29	58.137	71.734	80.406	86.081	89.721	92.933	94.432	95.503	96.466
SEC Phase2	27.86	61.66	75.12	83.72	88.42	92.96	95.77	96.71	97.34	98.60	99.53

Table 3: Testing results of model trained on 80% of dataset and tested on 20% of the dataset

data	Pixel distance										
	5	10	15	20	25	30	35	40	45	50	55
60-40	2.8125	11.25	25.3125	40	54.375	65.9375	73.75	84.0625	90.625	93.4375	95
50-50	3.5	10.75	20.75	35.75	51.25	60.25	71.5	80.25	86.25	91	93.25
80-20	6.875	11.875	20	35	48.75	61.25	70.625	78.75	86.875	90.875	94.375

Table 4: Dataset split testing for FVC2002 DB1