A biometric based case study: Unrevealing the truth behind the features learned by Convolutional Neural Networks

Anonymous ICB 2018 submission

Abstract

Voluminous biometric projects like Aadhar and FBI next generation Identification (NGI) triggers the need of such biometric systems that are reliable as well as can be reterieved in a short span of time. Today the total number of fingerprints enrolled under Aadhar scheme is more than 1.1 billion, in such a scenario we require algorithms that are easily scalable. In this paper we present a novel deep convolutional neural network based architecture for hashing fingerprint database. The prodigious strength of this paper lies in the fact that we are not extracting minutia patterns from fingerprints for generating the hash-function.

1. Introduction

In today's era it is difficult to imagine image processing applications without deep-learning. Infact it is right worthy to say that current era is an era of deep-learning. Many researchers have contributed to enrich this field with novel ideas and techniques. Particularly the work done by Hiltons in backpropogation and Lecuns in convolutional neural network is praise worthy. The success of deep-learning poses again an interesting old theory question why deep networks better than shallow networks? Do deep-learning really learn locality of the function estimation at each layer? Many researchers try to relate these questions with the computational complexity of human brains. Some of them have suggested that deep neural networks try to mimic the behaviour of visual cortex. Although still we are lagging far behind in terms of computational complexity of human visual cortex.

Identifying different objects in a scene classification problem is a mundane task. But in biometrics trait classification problems like identifying different fingerprint sensors & iris sensors; three class iris classification into no-lens, soft-lens and cosmetic-lens; and classifying a genuine biometric trait from a spoofed biometric trait we require fine grain granular level feature analysis. In biometric domain our main concern is to achieve identification accuracy as close to 100% as possible.

Motivation The foremost important point while deciding the convolutional neural network architecture is to consider the complexity of the classification problem in hand. No doubt that deep neural networks has the ability to transmute input feature vectors into higher dimensional spaces which makes them paramount in classification. But there is no free lunch theorem for this exceptionally high performance in deep neural network architecture, as massive amounts of training data is a pre-requisite criteria for achieving it. On the other hand shallow network performs quite well if you have limited amount of training data. Therefore it is most important to understand the kind of discriminative features required for classification before freezing the architecture.

Problem Statement In our work, we are trying to investigate the features learned by convolutional neural network architecture for this we have taken finger-print sensor classification as a case study. Fingerprint sensors can be classified into various categories *e.g.* (i) basis of imaging technology they are classified as optical, capacitive and thermal; (ii) basis of user interaction they are classified as press, sweep and non-contacted ones. Fig.1 shows fingerprint images captured from different types of sensors. It is evident from Fig.1, that image quality largely depends upon underlying sensor employed.

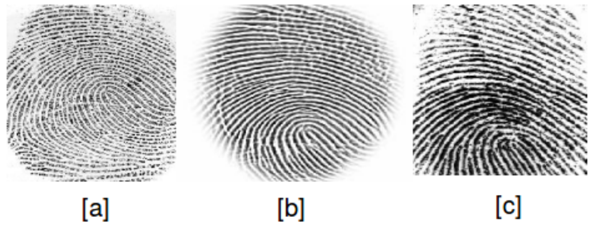

Figure 1. Example of fingerprint images taken from different sensors (a) Futonic, (b) Lumidigm, (c) SecuGen

In such a heterogenous sensor environment, it is crucial to identify the source sensor by which the acquired image is captured. This is essentially required to handle sensor inter-operability issues and further in identifying various attacks on biometric systems, where biometric templates can be modified or mis-used. Another interesting application of sensor identification is in establishing the sequence of commands for law enforcement to identify spurious activities in online systems. An image can be altered or fabricated during the acquisition phase, transmission or during storage. In order to understand whether the image has been fabricated or not it is necessary to know the source that generates the image.

Related Work : Not much work in the literature has been done so far for understanding the power of the deep-architecture through the features learnt by different layers. A review by Pinkus in the year 1999 concludes that ”there seems to be a reason for assuming that the two hidden layer model may be significantly more promising than the single hidden layer“. N. Mhaskar et al. proposed a theoretical framework for understanding the classes of functions which could be approximated well enough by the DCNN’s. In another work by Poggio et al. they have tried to find the conditions under which deep-learning is exponentially better than the shallow-learning. In our work we have tried to find the features learnt by convolutional neural network under different scenarios by taking fingerprint sensor classification as a case study.

The existing sensor identification techniques can be grouped into three main categories based on : (i) hand crafted features, (ii) sensor pattern noise and (iii) colour filter. Bayrem et al. [4] proposed a method for sensor identification based on measuring the interpolation artifacts occurred in image using color filter arrays. Lukas et al. [10] proposed a technique in which sensor is identified by measuring the pixel non uniformity (PNU) noise of each image using wavelet based denoising. Further Barlow et al. [3] used a variant of PNU technique known as photo response non-uniformity ($PRNU$) for fingerprint sensor identification. Agarwal et al. [1] used handcrafted features that includes features based on entropy, texture, image quality and statistics for sensor recognition. Recently, Sudipta et al. [2] identified sensors from NIR iris images. In their work they have reported that enhanced Sensor Pattern Noise Scheme (SPN), works better for detecting image sensor than maximum likelihood and phase based SPN methods. Uhl and Holler [17] have also used $PRNU$, to identify NIR iris sensor from their images. Table 1, summarizes the related work done in fingerprint sensor classification.

Author	Significant Contribution
Ross & Jain [15]	Optical versus Solid State
Bartlow [3]	Photo Response Non Uniformity
Modi [12]	False non match rate, minutia count
Jia [8]	Cross Database(Fingerpass)
Lungini [9]	Optical fingerprint sensor interoperability
Agarwal [1]	Combining handcrafted features
Debiasi [6]	Multiple PRNU enhancements

Table 1. Summarized sensor classification literature review

Contribution: In this paper we have proposed a Convolutional Neural Network based architecture for fingerprint sensor classification. The main contribution of this paper is three fold, that is summarized in the following section.

1. An architecture based on Deep Convolutional Neural Network is proposed that is capable of detecting input fingerprint sensor by systematically pruning and training two different types of convolutional neural networks VGG and ResNet50 namely.

2. In-depth feature analysis is done to understand the real-insight of features learned by different layers.

3. A highly generalized deep convolutional neural network based architecture has been proposed.

The rest of the paper is organized in the following manner Section 2 presents the proposed architecture framework, Section 3 discusses the experimental results, and Section 4 finally concludes our paper.

2. Proposed Architecture

In an image classification problem the task is to predict a label of the input image among the set of the predefined labels. Traditional methods of classification uses hand-crafted features like HOG and SIFT, but these methods encode low-level characterstics and thus not able to distinguish well in case of fine grain classification problems. In present world deep learning based methods are the state-of the art methods that are capabale of encoding higher level characterstics.

Extensive experimentation has been done in order to decide the suitable network for our fine grain finger-print sensor classification problem. We have considered two networks (a) A shallow network (VGG) (b) A deep network (ResNet50) in order to understand the type of features, classification accuracy and generalization ability trade-off between shallow and deeper networks. As our finger-print sensor classification problem is not a trivial one, we are required to extract features at granular level. Another important point of consideration here is that our fingerprint image size is small and using a network having large kernel size will not be too useful. Considering all the above points in mind we conducted two set of experimentations.

Shallow Network(VGG) In the first set of experimentation we have used a variant of VGG-19 a popular deep-convolutional neural network model. The foremost advantage of using this network is its small kernel filter size of $3{*}3$ which tries to learn high-level features at granular levels.

VGG-19 network is divided into 5-blocks with 19 weight layers. It takes an input image of size $224 * 224$. Due to the limited amount of training data at our disposal we are incapable of re-training it from scratch. After doing experimentation we have found that the block-5 is not adding any discriminative information for our fingerprint sensor classification problem so we systematically prune it. While fine tuning this network we have used $adam$ optimizer with a mini-batch size of 64, initial learning rate has been set as 0.001 for 30 epoches. All the parameters that are used in this work are calculated empirically over a small validation set.

Deeper Network(ResNet50) In the second set of experimentation our network is inspired by ResNet50 architecture. We had taken this network deliberately in order to know whether the presence of identity connections between the layers in the residual network architecture gives us an advantage in learning discriminative features for fingerprint sensor classification. We took pre-trained ResNet50 model on ImageNet images and performed extensive experimentation over it using multi-sensor fingerprint images in order to fine tune it. The existing ResNet50 model consists of five main branches as described below.

- Branch-1 is the initial one that convolves the input image of size $224 * 224$ with 64 kernels and gives an output image of size $112 * 112$. After that max-pooling has been done to avoid over-fitting and to reduce the amount of parameters and in turn computation. The max-pooling layer gives an output of size $55 * 55$.
- Branch-2 consists of three sub-branches branch-2a, branch-2b and branch-2c. At the end of branch-2 an output of size $55 * 55$ has been obtained.
- Branch-3 of ResNet50 model consists of four further branches branch-3a, branch-3b, branch-3c and branch-3d. At the end of branch-3 an output of size $28 * 28$ is generated.
- Branch-4 of ResNet50 model consists of five further branches namely branch-4a, branch-4b, branch-4c, branch-4d, branch-4e and branch-4f and at the end output of size $14 * 14$ is generated.

ResNet Description and Parameterization: The first layer of proposed architecture is an input layer which takes an input image of size $224 * 224$. Given the fingerprint image and its corresponding label, the first convolutional layer $CONV1$, filters the input image of size $224 * 224$ using 64 kernels and pool it to an output of size $112 * 112$. The filters of $CONV1$ layer generally detects the edges and colors of the input image. The output of the $CONV1$ is connected to max-pooling layer. After that $Branch-2$, $Branch-3$ and $Branch - 4$ blocks of ResNet50 architecture has been utilized. At the end a fully connected layer with 2048 neuron has been added along with a dropout layer in the model with a probability of 0.4, in order to avoid over-fitting as well as to force the network to learn only robust features. Finally $softmax$ function has been used to generate the probability distribution by minimizing the categorical cross-entropy loss-function. While fine tuning this network we have used $adam$ optimizer with a mini-batch size of 64, initial learning rate has been set as 0.001 for 30 epoches. All the parameters that are used in this work are calculated empirically over a small validation set.

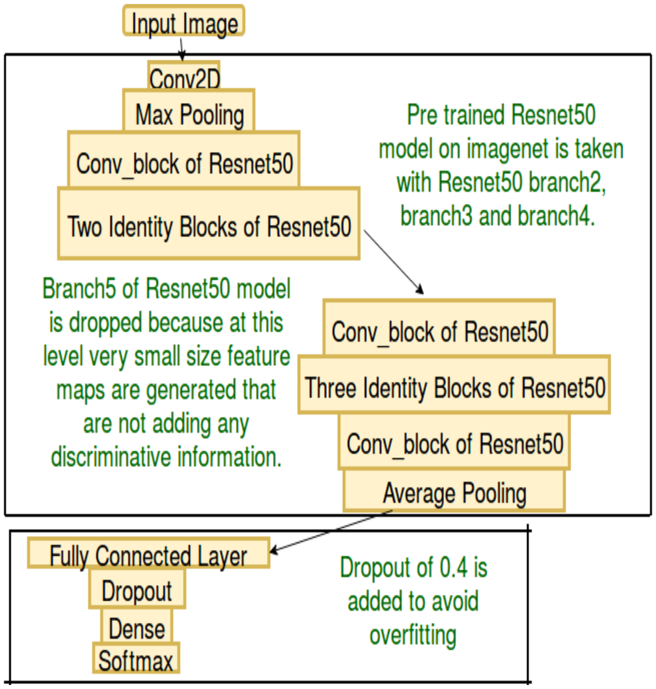
Input Image
Conv2D
Max Pooling
Conv_block of Resnet50
Two Identity Blocks of Resnet50
Pre trained Resnet50 model on imagenet is taken with Resnet50 branch2, branch3 and branch4.
Branch5 of Resnet50 model is dropped because at this level very small size feature maps are generated that are not adding any discriminative information.
Conv_block of Resnet50
Three Identity Blocks of Resnet50
Conv_block of Resnet50
Average Pooling
Fully Connected Layer
Dropout
Dense
Softmax
Dropout of 0.4 is added to avoid overfitting

Figure 2. Block diagram of the proposed Architecture.

ResNet Pruning : ResNet50 is a very deep network with 50 weight layers trained over 1000 ImageNet classes. Since we have less number of classes and that too with few thousands of images, we have decide to prune the network systematically in order to avoid over-fitting. It is also famous for its notorious training hence extensive experimentation has been done to $fine - tune$ as well as systematically prune existing ResNet50 model. During experimentation we found that $Branch-2$ and $Branch-4$ of ResNet50 are extremely important for learning discriminative information, which is necessary for differentiating between images acquired from different fingerprint sensors. We also have observed that $Branch - 5$ and $Branch - 3$ have "similar" contribution in final classification for our specific fingerprint sensor classification problem. Hence one can drop anyone layer, but dropping both have caused drastic performance deterioration. Hence we have dropped $Branch - 5$ because generated feature map size was only $7 * 7$. Fig. 2

shows the proposed network architecture, which has been designed in a manner so that it can classify commonly used fingerprint sensors. In this model, we have dropped $Branch - 5$, since in our case this branch was not learning much discriminative information. By doing so, we have decreased the computation time while retaining the performance.

Observation: While implementing the two set of models described above we have observed that although the ResNet is much deeper model than the VGG but the model size of ResNet is much smaller than the VGG due to the usage of global average pooling layers in comparision to the fully connected layers.

Network Implementation Details: All implemented is done in python using keras library and tensorflow as a backend. All the implementation is done on Intel(R) Xenon(R) CPU E5-2630 V4 at 2.20 GHz.

3. Experimental Analysis

In this section, we provide the details about the database and the testing protocol used in this work. We have tested our proposed architecture on publicly available databases as well as over largest available IITK database which has more than $40,000$ images. We have used single fingerprint images as well as four slap fingerprint images.

The IITK dataset of single fingerprints consists of $41,129$ images collected using three different types of sensors *viz.*, (i) Futronic (FS88H), (ii) Lumidigm V310 (V31X) and (iii) SecuGen Hamster I. All of them are of same 500 DIP, but the light source and the image size generated is different. The IITK dataset of four slap fingerprints consists of $26,215$ images collected using three different types of sensors *viz.*, (i) Crossmatch, (ii) Sagem and (iii) Morpho. Proposed architecture has also been tested upon four publicly available benchmark single fingerprint databases *viz.*, (i) FVC2002 [11], (ii) FVC2004[7] and (iii) FVC2006 [5] (iv) IITD-MOLF. Few sample images of FVC datasets are shown in Fig.3. All FVC dataset consists of images collected from four different types of sensors. The detail description about the dataset considered has been given in Table 2.

For our work we have considered only the set-A of FVC datasets. More information about these databases can be obtained from the reference papers [5], [11], [7]. We have trained the proposed model by taking only 10% of the single finger print images as training set and rest 90% as testing set, thus adopting a very difficult protocol.

3.1. Single Slap Fingerprint Experimentation:

In this section we present the classification results of the proposed two different architectures (deep, shallow) on single fingerprint images. The results are computed for two different testing strategies namely (a) Intra sensor classification (b) Multi sensor classification [13]. The Correct Classification Rate (CCR%) is computed for performance evaluation, higher the value better is the result.

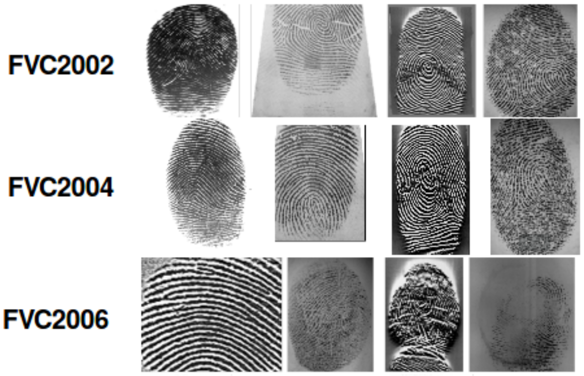

Figure 3. Sample images from FVC dataset.

Database	Sensor Model	Images
FVC 2002 [11] $(SingleFinger)$	Optical Sensor ”TouchView” Optical Sensor ”FX2000” Capacitive Sensor ”100SC” Synthetic fingerprint generation	3,200
FVC 2004 [7] $(SingleFinger)$	Optical Sensor ”V300” Optical Sensor ”U4000” Thermal Sweeping Sensor Synthetic fingerprint generation	3,143
FVC 2006 [5] $(SingleFinger)$	Electric Field Sensor Optical Sensor Thermal sweeping Sensor SFinGe V3.0	6,720
IIITD-MOLF [5] $(SingleFinger)$	Lumidigm Venus IP65 Shell Secugen Hamster-IV CrossMatch L-Scan Patrol	16,400
IITK-S Dataset $(SingleFinger)$	Futronic FS88H Lumidigm V310(V31X) SecuGen Hamster I	41,129
IITK-F Dataset **(Four Slap Finger)**	Crossmatch Sagem Morpho	26,215

Table 2. Detail Database Specifications

Database	Sensor Type	Shallow	Deep
FVC 2002 [11]	DB1	100	100
	DB2	99.17	96.79
	DB3	100	99.72
	DB4	99.72	99.45
	Aggregate	**99.72**	**98.99**
FVC 2004[7]	DB1	100	100
	DB2	100	100
	DB3	99.85	93.78
	DB4	100	99.17
	Aggregate	**99.96**	**98.23**
FVC 2006[5]	DB1	100	99.87
	DB2	100	99.28
	DB3	100	99.80
	DB4	99.93	100
	Aggregate	**99.98**	**99.73**
IIITD-MOLF	DB1	100	100
	DB2	100	100
	DB3	99.92	100
	Aggregate	**99.97**	**100**
IITK S Dataset	Futronic	99.45	99.34
	Lumidigm	100	100
	SecuGen	100	100
	Aggregate	**99.82**	**99.78**

Table 3. Intra-Sensor qualitative performance in CCR(%) on proposed architectures (where IITK S dataset is single fingerprint dataset and DB1 means the images from the first sensor of the corresponding dataset and same nomenclature is used for other datasets)

3.1.1 Intra-sensor Classification

In this type of classification training and testing is performed on images acquired from same type of sensor model. Table 3, indicates the computed performance of two different proposed architectures on various datasets. The main phenomenon that we observed is that we are getting almost same type of CCR(%) inspite of the fact of using two different kind of varied architectures. This can be attributed to the fact that fingerprint classification problem is a fine grain problem which does not require to learn features at the finest granular level and thus even shallow network is giving remarkable performance.

3.1.2 Multi-sensor Classification

In multi-sensor classification, data is fused together from various fingerprint sensors. We did this purposefully in order to check the generalizability of our proposed architectures. We have merged data from all FVC datasets *i.e.* FVC 2002[11], FVC 2004[7] and FVC 2006[5]. The combined dataset consists of $13,063$ images resulting from 12 different sensors and trained our network with 12 output neurons. We have trained our proposed model on 1306 images and tested on remaining $11,757$ images (*i.e.* 10% training and 90% testing). Table 4 indicates the computed performance in terms of CCR(%). We have observed that the obtained results are quite remarkable which depicts the high generalizability of our proposed architectures. In this case also shallow networks are learning discriminative features quite well.

Database	Sensor Type	Shallow	Deep
FVC Combined [11]	2DB1	98.75	99.29
	2DB2	97.63	98.37
	2DB3	100	99.57
	2DB4	99.58	99.72
	4DB1	99.31	99.16
	4DB2	100	100
	4DB3	100	99.40
	4DB4	100	97.47
	6DB1	100	100
	6DB2	100	100
	6DB3	99.86	98.56
	6DB4	99.80	100
	Aggregate	**99.58**	**99.29**

Table 4. Multi-sensor qualitative performance in CCR(%) on proposed architectures (here in xDBy: x stands for FVC dataset number and y stands for sensor type of the corresponding dataset(eg. x=2 & y=1 means first sensor of FVC-2002 dataset)

3.2. Four Slap Fingerprint Experimentation :

We have used one state of the art localization network *viz.*, Faster R-CNN [14] that is based on region proposals for extracting the ROI from four slap fingerprints. The ground truth for finger bounding boxes have been generated using method suggested in [16] which was around 92% accurate. Fig. 5 shows the ROI extracted on Four Slap fingerprint dataset. Extensive experimentation has been done in order to select the appropriate threshold and other network parameters for extracting the correct ROI. We have trained Faster R-CNN from scratch, for four slap fingerprint. Fig. 4, shows the graph that can summarize the performance of the the trained localization network where x axis is threshold applied over IOU (Intersection over Union) computed between predicted and actual bounding box, while y axis represents the accuracy at some threshold.

Four Slab Sensor Detection : For detecting the sensor of four slap fingerprints we have used the same proposed architecture (as discussed previously). We fed a four slap fingerprint image (sized 1672×1572) to our trained model as well as four single fingerprint images (extracted using the Faster-RCNN). Finally $softmax$ function has been used to generate the probability distribution. Each single fingerprint

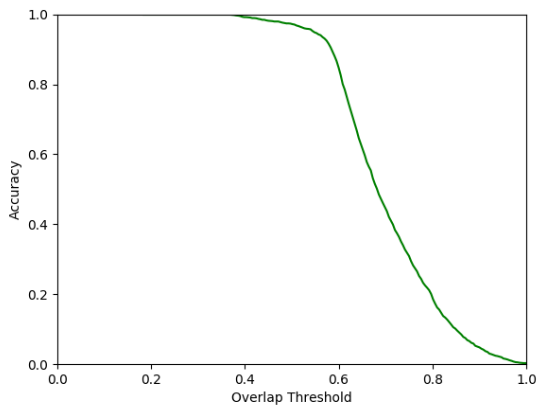

Figure 4. Accuracy Graph

have probability distribution corresponding to its sensors. In order to predict the corresponding sensor for four slap fingerprint the aggregated sum of all probabilities has been calculated and maximum among them is assigned as its label class.

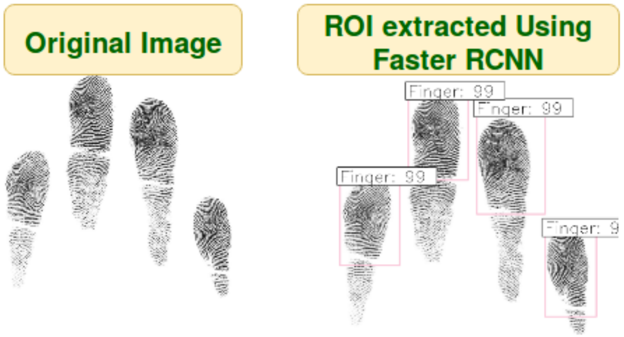

Figure 5. Four Slap Fingerprint (a) Original image, (b) Corresponding ROI extracted using FPSegNet.

Four Slap Fingerprint Results : We have used the same deep network for predicting the sensor of the four slap fingerprint images. In total we have $26,215$ images of four slap fingerprints, out of these, when we have used $15,729$ images as whole for training the network, obtained CCR(%) are as follows Crossmatch-Crossmatch 99.98, Sagem-Sagem 96.82, Morpho-Morpho 80.9. In the second set of experiment, we have used only 4080 images for training the network. In this case we have only shown the four single fingerprint images (extracted using retrained-Faster-RCNN), corresponding to the single four slap finger print for training the network. In this case the CCR(%) are as follows, Crossmatch-Crossmatch 100, Sagem-Sagem 94.76, Morpho-Morpho 65.2. In the first set of experiments we are getting low CCR(%) for Morpho due to big re-sizing, one have to do before feeding it to our network (from $1672 * 1572$ to $224 * 224$), hence a lot of discriminative information got lost during size reduction. In order to address this we have consider the second set of experimentation but due to the restricted computational resources at the dispose we have to train our network only over 1360 images per sensor.

3.3. Comparative Analysis :

In[1], fingerprint sensor classification has been done *via.* combination of handcrafted features. They have combined four databases namely FVC 2002, FVC 2006, IIIT-D MOLF and CASIA Cross Sensor Fingerprint dataset. In total they have $29,320$ images out of which they have used 3000 images for training and remaining for testing. They have achieved an accuracy of about 96.52%. In our experimentation we have also considered training on 10% dataset and testing on remaining 90% dataset. We have also considered FVC 2004 dataset which is quite challenging, and four- slap finger-print dataset of IIT-K . In all the cases the achieved aggregated accuracy is more than 98% and above. We are not comparing our results with [1] because of the unavailability of CASIA Cross Sensor fingerprint dataset .To the best of our knowledge this is the first attempt in which Deep Convolutional Neural Network is used for identifying the sensor of the underlying fingerprint image.

3.4. Robustness or Generalization Analysis

In order to gain further insights in understanding the effectiveness and generalization of the proposed architecture we introduce some artifacts in the original image in the form of random-noise, rotation and occlusion. All these artifacts are done on a small validation-set comprising of 100 images for each sensor corresponding to IITK single fingerprint dataset.

Rotation In order to check the robustness of the proposed architecture the input fingerprint images have been rotated with different angles and the computed CCR% is shown in Table 5. It can be inferred from Table 5 as we increase the angle of rotation the performance of SecuGen sensor decreases drastically this can be attributed to the non-uniformity of the image captured through it as clearly visible from Fig.1

Occlusion In order to test whether our proposed architecture is invariant to occlusion or not. We consider a patch of $90 * 90$ in the input image and randomly occluded some percentage of pixels in it. The results in terms of CCR% has been shown in Table 6. Surprisingly it can be inferred from Table 6 that on increasing the occlusion in the image the performance of the network is not deteriorating. This abnormal behaviour compel us to think what exactly our network is learning. We will try to elaborate on this further

Rotation	2	4	6	8	15
Futonic	99	99	99	99	99
Lumidigm	100	100	100	100	100
SecuGen	94.5	84.5	80	76	54

Table 5. Rotation qualitative performance in CCR(%) on validation set for shallow network architecture (here 2, 4, 6, 8 and 15 represents the angle of rotation in clockwise as well as in anti-clockwise direction)

while visualizing the features learned by our network in the folowing sections.

Occlusion	0.1%	1%	5%	10%
Futonic	99	99	98	98
Lumidigm	100	100	100	99
SecuGen	99	100	100	100

Table 6. Occlusion qualitative performance in CCR(%) on validation set for shallow network architecture (here 0.1%,1%,5% and 10% are the percentage of pixels occluded in a region of $90 * 90$ patch)

Random Noise In real life scenarios it is very difficult to get a perfect condition. Most of the time we end up with noise affecting our ideal conditions. In such cases it is essential that our architecture is robust for noise upto certain range. Table 7 shows the result of adding random noise to our validation test data. It can be infered from Table 7 that on increasing the random noise the performance of the Lumidigm sensor is decaying to a large extent. As it is clearly evident from Fig.1 that the Lumidigm sensor captured image is highly uniform in texture, so any small alteration or artifact in its texture greatly affects our network performance.

Noise	0.01%	0.05%	0.1%	1%	5%
Futonic	99	99	99	98	93
Lumidigm	100	94	50	0	0
SecuGen	99	98	98	99	98

Table 7. Random noise qualitative performance in CCR(%) on validation set for shallow network architecture (here 0.01%, 0.05%, 0.1%, 1% and 5% are the percentage of the random noise inserted in the input image of size $224 * 224$)

3.5. Layer Specific Feature Analysis

3.6. Conclusion

Occlusion	0.1%	1%	5%	10%
Futonic	93	37	10	0.8
Lumidigm	87	2	0	0
SecuGen	99	100	100	98

Table 8. Occlusion qualitative performance in CCR(%) on experimental set for shallow network architecture (here 0.1%,1%,5% and 10% are the percentage of pixels occluded in a region of $90 * 90$ patch)

References

[1] A. Agarwal, R. Singh, and M. Vatsa. Fingerprint sensor classification via mélange of handcrafted features. In *23rd International Conference on Pattern Recognition, ICPR*, pages 3001–3006, 2016.

[2] S. Banerjee and A. Ross. From image to sensor: Comparative evaluation of multiple prnu estimation schemes for identifying sensors from nir iris images. In *5th International Workshop on Biometrics and Forensics (IWBF), (Coventry, UK),2017*.

[3] N. Bartlow, N. D. Kalka, B. Cukic, and A. Ross. Identifying sensors from fingerprint images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops , Miami, FL*, pages 78–84, 2009.

[4] S. Bayram, H. Sencar, N. Memon, and I. Avcibas. Source camera iden- tification based on cfa interpolation. In *IEEE International Conference on Image Processing*, volume 3, pages 69–72, 2005.

[5] R. Cappelli, M. Ferrara, A. Franco, and D. Maltoni. Fingerprint verification competition 2006. In *Biometric Technology Today,2007*, volume 15.

[6] L. Debiasi and A. Uhl. Blind biometric source sensor recognition using advanced prnu fingerprints. In *23rd European Signal Processing Conference (EUSIPCO),2015*.

[7] D.Maio, D.Maltoni, R.Cappelli, J. Wayman, and A.K.Jain. Combining multiple matchers for fingerprint verification: A case study in FVC2004. In *13th International Conference, Image Analysis and Processing ICIAP, Cagliari, Italy,2005*.

[8] X. Jia, X. Yang, Y. Zang, N. Zhang, and J. Tian. A cross-device matching fingerprint database from multi-type sensors. In *21st International Conference on Pattern Recognition*, pages 3001–3004, 2012.

[9] L. Lugini, E. Marasco, B. Cukic, and I. Gashi. Interoperability in fingerprint recognition: A large-scale empirical study. In *43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop*, 2013.

[10] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, pages 205–214, 2006.

[11] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. Fvc2002: Second fingerprint verification competition. In *16th Inter- national Conference on Pattern Recognition,2002*, volume 3.

[12] S. K. Modi, S. J. Elliott, and H. Kim. Statistical analysis of fingerprint sensor interoperability performance. *In IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*, 2009.

[13] R. Raghavendra, K. Raja, and C.Busch. Contlensnet: Robust iris contact lens detection using deep convolutional neural

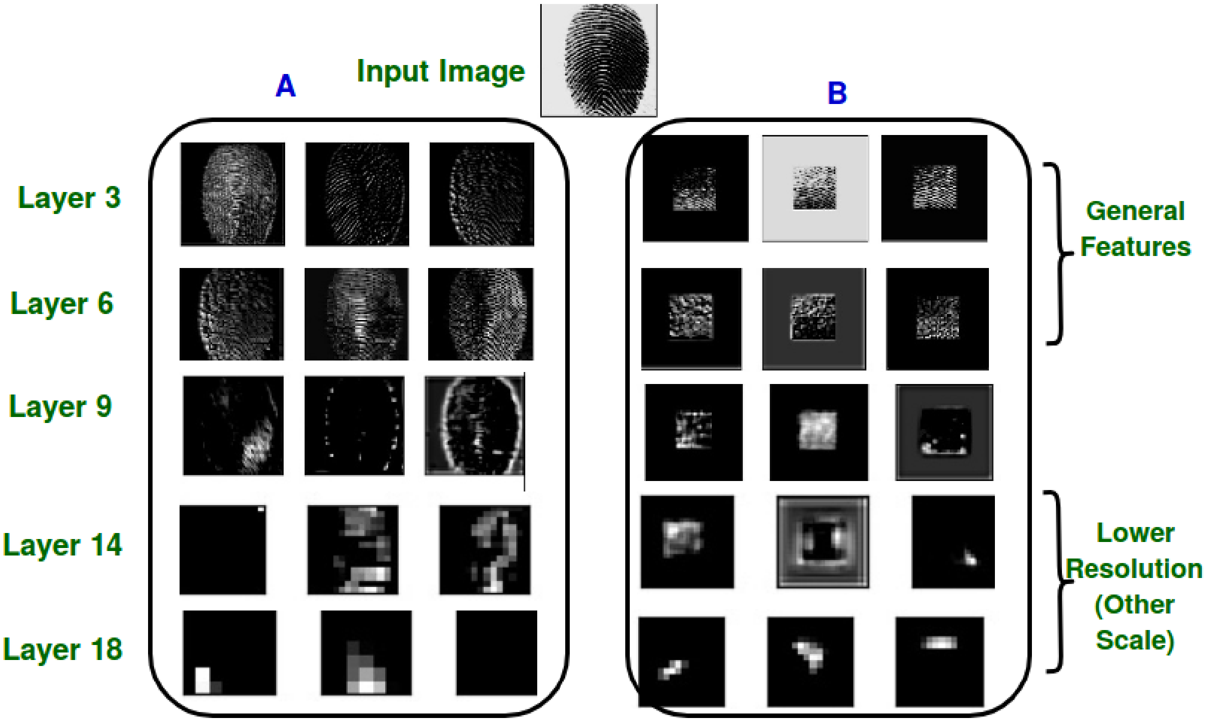

Figure 6. Comparative Feature Analysis

networks. In *IEEE Winter Conference on Applications of Computer Vision, WACV*.

[14] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1137–1149, 2017.

[15] A. Ross and A. K. Jain. Biometric sensor interoperability: A case study in fingerprints, 2004.

[16] N. Singh, A. Nigam, P. Gupta, and P. Gupta. Four slap fingerprint segmentation. In *8th International Conference, Intelligent Computing Theories and Applications ICIC, Huangshan, China*, pages 664–671, 2012.

[17] A. Uhl and Y. Hller. Iris-sensor authentication using camera prnu fingerprints. In *5th IAPR International Conference on Biometrics (ICB)*, pages 230–237, 2012.