# Fingerprint Enhancement Using Unsupervised Hierarchical Feature Learning

Mihir Sahasrabudhe*
Anoop M. Namboodiri
Centre for Visual Information Technology, IIIT - Hyderabad
{mihir.s@research., anoop@}iiit.ac.in

## ABSTRACT

We present an approach for learning low- and high-level fingerprint structures in an unsupervised manner, which we use for enhancement of fingerprint images and estimation of orientation fields, frequency images, and region masks. We incorporate the use of a convolutional deep belief network to learn features from greyscale, clean fingerprint images. We also show that reconstruction performed by the learnt network works as a suitable enhancement of the fingerprint, and hierarchical probabilistic inference is able to estimate overall fingerprint structures as well. Our approach performs better than Gabor-based enhancement and short time Fourier transform-assisted enhancement on images it was trained on. We further use information from the learnt features in first layer, which are short and oriented ridge structures, to extract the orientation field, frequency image, and region mask of input fingerprints.

## Keywords

Biometrics, Fingerprint Enhancement, Deep Learning, Feature Learning

## 1. INTRODUCTION

Fingerprint recognition is the most widely used biometric to identify individuals. With a history of over a century, fingerprint recognition has been observed to be applied to a host of systems and tasks. The list includes authentication systems, forensic science, and attendance systems, to name a few [14].

The increase in popularity further emphasizes the need that fingerprint recognition tasks should be made as robust as possible. As expected, much has already been achieved in this field ([1, 2, 4, 5, 8, 9, 15, 20, 24]). Having started out as a manual task, where fingerprint experts would sit down with pairs of fingerprint images and try to find a match between them, recognition tasks have been taken over by computers since their advent. Advances in image processing have enabled development of robust algorithms, and technology has allowed these algorithms to be available to end-users at a small cost. However, this easy availability has been possible only because extensive research has tackled challenges faced during fingerprint recognition. A prominent and ever-present challenge is tolerance of noise. Most widely available fingerprint sensors induce at least one kind of noise in the images they output. This noise ranges from skin conditions such as dirt, injuries, cuts and bruises, and moisture, to that introduced by the sensors themselves, typically via. a worn-out sensor-surface [14]. It is then essential to remove these types of noises and extract the fingerprint image without losing information.

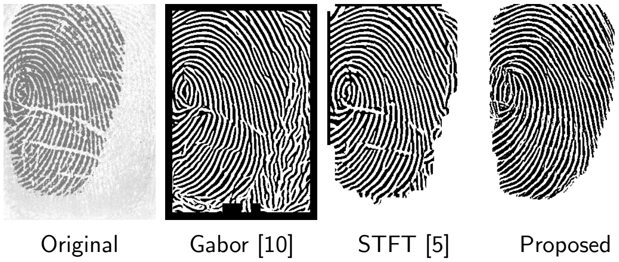

**Figure 1: Enhancement of a noisy fingerprint image using the proposed algorithm, compared to [10] and [5].**

## 2. RELATED WORK

The orientation field of a fingerprint image is very essential in enhancing the fingerprint. Filtering the image using Gabor filters based on ridge direction and frequency at a point gives a robust enhancement of the print. Since Hong, Wan and Jain proposed the use of Gabor filters [10], several modifications of the approach have been developed. Zhu, Yin and Zhang [24] achieve faster enhancement using circular Gabor filters. Yang *et al.* [23] developed a modified version of this technique to obtain more consistent enhancement, without damaging fingerprint structure. Bernard *et al.* [1] use wavelet filtering to arrive at a multiscale approach. Algorithms that pursue other filtering techniques have been developed, but could not gather as much momentum as Gabor filters. Greenberg *et al.* [9], for instance, used Weiner and anisotropic filtering. However, the task of orientation field estimation still remains very important to fingerprint

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

*ICVGIP* '14, December 14 - 18 2014, Bangalore, India
Copyright 2014 ACM 978-1-4503-3061-9/14/12 ...$15.00
http://dx.doi.org/10.1145/2683483.2683485.

enhancement, especially for very noisy images and latent prints [8]. Several frontiers have been explored to achieve robust estimation of orientation fields.

Global models try to model whole fingerprint structures, instead of using local ridge structures to estimate orientation fields. Popular approaches in this category employ Markov random fields (MRFs) to arrive at a fingerprint structure that minimises energy. Some examples of these are Dass [6], and Lee and Prabhakar [4]. Reddy and Namboodiri [20] improve upon these further by employing hierarchical MRFs that use loopy belief propagation.

Research has explored the frequency domain, in contrast to the spatial domain, to enhance fingerprints. The most popular work in this category, by Chikkerur *et al.* [5], uses the analysis of Short Time Fourier transform (STFT). A given fingerprint image is divided into regions, and intrinsic properties of the fingerprint are estimated for these regions. Intrinsic properties include the orientation in those regions, the local ridge frequency, and the mask that indicates the presence of a fingerprint. Based on these, filters are constructed and applied. Results of STFT analysis are comparable to Gabor-based enhancements, which opens up new prospects. In a previous work [21], we had proposed the use of unsupervised feature extraction using continuous restricted Boltzmann machines to reconstruct local, noisy orientation fields.

In this paper, we look to use a deep, generative neural network to achieve fingerprint image enhancement. Strides have been made in deep learning over the past decade, and a host of models suited to different tasks have emerged. Deep belief networks (DBNs) arose as suitable many-layered (deep) models to learn patterns. DBNs have the restricted Boltzmann machines (RBMs) as their building blocks. As newer methods to train neural networks surfaced [3], it became more feasible to use neural networks in pattern recognition.

For this paper, we were interested in a generative, deep model that is capable of working directly on pixels and extracting features in an unsupervised manner. Convolutional networks fit the choice very well, in that they are scalable to large images, feature detection performed by them is robust, and they now are known to perform very well at several real-world classification problems ([12]). However, convolutional neural networks themselves are not generative models. Instead, we use convolutional deep belief network (CDBN) in this work. Desjardins and Bengio [7], developed convolutional RBMs, and were closely followed by Lee *et al.* [13], who introduced probabilistic max pooling and hierarchical probabilistic inference - both of which proved essential in making the convolutional DBNs generative models.

# 3. CONVOLUTIONAL DEEP BELIEF NETWORK (CDBN)

To maintain continuity in this manuscript, we give a brief description of the convolutional deep belief network in this section. We will start with the basic building block for a convolutional DBN - the convolutional RBM - and move on to the CDBN, later to computing the network's representation of the image, and then reconstructing from this representation using the learnt parameters.

## 3.1 Convolutional RBM

A convolutional RBM is a neural network that, like an RBM, has a visible layer and a hidden layer. In addition to an RBM, though, it also has a pooling layer. The visible layer of an RBM is shown images from which features need to be learnt. This data, called *training data*, is a an essential factor in learning representations. A detailed description of the training data we used for our learning is given in Section 4.1.

From images shown to the visible layer, the network draws an inference, in that it computes states of neurons in the hidden layer. Let the visible layer be called $V$. The hidden layer, $H$, consists of $K$ groups of units. With each group, $k$, of hidden units, we associate a weight $W^k$ that connects the visible units with the group.

Neurons in the $k$-th group of hidden units are obtained by a 'valid' convolution of the visible layer with the weight $W^k$:

$$I\{H^k\} = \left[\tilde{W}^k * v\right] + b_k \tag{1}$$

Each block in this group of units is connected to exactly one unit in the $k$-th group of pooling units. This sub-sampling step is called *probabilistic max pooling.* Unlike other pooling algorithms used mostly in convolutional neural networks (for example, *max pooling*), this method is probabilistic in nature.

We perform alternate Gibbs sampling to get the activations of visible and hidden neurons. Training is done with an update to the weights that is proportional to the difference of positive and negative associations, like the RBM. The associations corresponding to a weight are calculated by convolving the visible layer, which is the same for all weights, with the group of hidden units corresponding to that weight. To approximate the objective function, contrastive divergence learning with 1 step (CD-1) was employed throughout this paper. To ensure that the representation that we learn is sparse, a sparsity update is done at every step of learning determined by the average of probabilities associated with every neuron in a group of hidden units. We also employ L2 regularisation to prevent overfitting of the model.

## 3.2 Converting to a Deep Network

With this building block at our disposal, we can now stack multiple convolutional RBMs on top of each other to build a convolutional DBN. In such a model, the pooling units of a layer in the DBN (which is a CRBM) serve as visible units for the next layer (Figure 2). We refer to one such stacked CRBM as a *layer* in the Convolutional DBN (CDBN).

Training the convolutional DBN involves layer-wise training of each layer using the algorithm stated in the previous section. Once this training is complete, pooling units are generated using the learnt weights on training images, which are in-turn used to train the next layer.

Notations concerning the parameters of a convolutional DBN have been illustrated in Figure 2.

## 3.3 Hierarchical Probabilistic Inference

Hierarchical probabilistic inference (HPI) is an algorithm to reconstruct an image using the network's representation of it. The term *hierarchical* says we use data from higher layers to affect values of hidden units in lower layers, and in-turn the reconstructions of images. The algorithm is *probabilistic* because it uses probabilistic max-pooling, described in Section 3.1, to draw an inference. To use data from higher

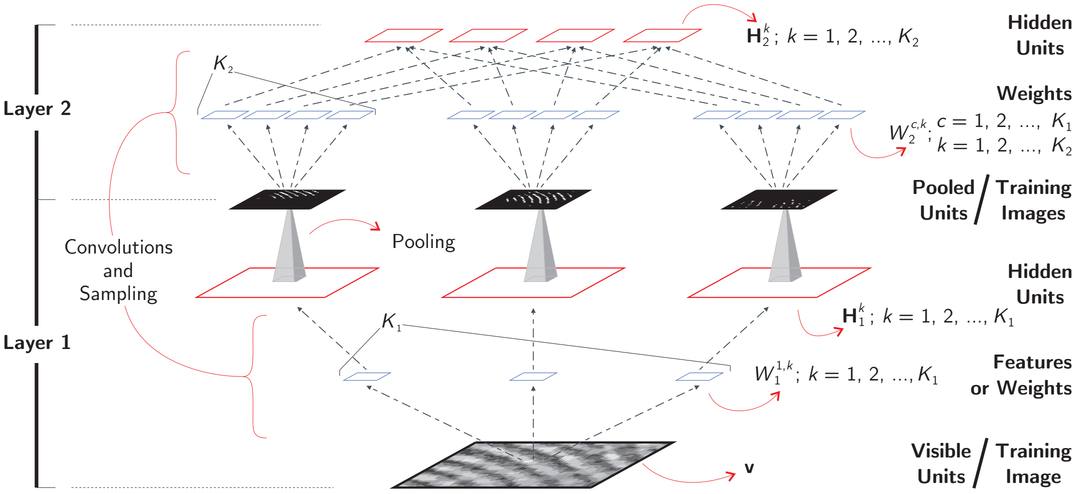
Layer 2
Layer 1
$K_2$
$\mathbf{H}_2^k$; $k = 1, 2, \ldots, K_2$
Hidden Units
Weights
$W_2^{c,k}$; $c = 1, 2, \ldots, K_1$; $k = 1, 2, \ldots, K_2$
Pooled Units / Training Images
Convolutions and Sampling
Pooling
Hidden Units
$\mathbf{H}_1^k$; $k = 1, 2, \ldots, K_1$
$K_1$
$W_1^{1,k}$; $k = 1, 2, \ldots, K_1$
Features or Weights
Visible Units / Training Image
$\mathbf{v}$

**Figure 2: A convolutional deep belief network with 2 layers. The further layers work on the data that is found in the previous layer's pooling units. The notation $W_c^{a,b}$ refers to the $b$-th weight corresponding to the $a$-th channel in the $c$-th layer, where a *channel* is one of the components of the input. We use greyscale images and so have only one channel in the visible units of the first layer.**

layers, hidden neurons in a layer are modified to receive inputs from the layer's visible units, as well as from the next layer's hidden units. This redefines the probability of a neuron firing according to the total input received by it [13].

# 4. ENHANCING IMAGES USING A CDBN

In this section, we will describe the training and use of a CDBN for fingerprint image enhancement.

## 4.1 The Training Data

The training data used is very important to learning. The network learns features depending on the training data. The reconstruction performed by this network also relies heavily on the training data. As we are trying to perform reconstruction of fingerprint images which should serve as enhancement, we use clean fingerprint images to train our network. We hand-picked images from standard fingerprint datasets (FVC 2000 Db1_a [16], FVC 2002 Db1_a [17], and NIST-4 special database [22]). While selecting images, we followed some rules: (a) significant variety in ridge orientations and ridge frequencies should be present in the image; (b) the images should have the least patches of noise, i.e., inconsistent ridges, cuts, bruises, smudges, wet fingers, pores, dirt, stray ink, lettering, and noise from sensors; (c) the images should have uniform contrast; and (d) the images should have uniform moisture, and should not be too wet or too dry. Further, the selected images were trimmed so as to include minimal regions that do not belong to the fingerprint. Before being used for training, the training data was normalised [14, 10] and whitened [19] before it is fed to the network. Whitening ensures that the training images have equal variance in different directions, and hence helps the gradient descent learning in its search for a solution.

## 4.2 Training on Fingerprint Images

We train the convolutional deep belief network on the preprocessed images. We used a two-layered network with greedy layer-wise training. Greedy layer-wise training refers to training each layer of the convolutional DBN by considering it to be a convolutional RBM. We start with the first layer, and the training data for this layer is our set of training images. Once the first layer is trained, we compute the first layer's inference of all images of the training data, and pool the hidden units to generate the training data for the next layer. The same process is then done with the second layer. To enhance a fingerprint using a learnt network, we used hierarchical probabilistic inference.

The initial weights are drawn from normally distributed random numbers and then multiplied by a factor of 0.01. Further, the visible biases are set to zero initially, and the hidden biases are also drawn from normally distributed random numbers but are multiplied with a factor of −0.1.

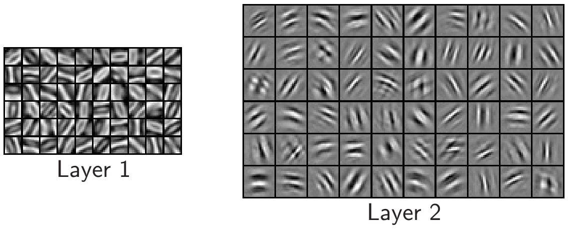

**Figure 3: Weights learnt by the first and second layers. The training data contained fingerprints with various ridge frequencies, which is visible in the diversity in ridge frequencies in the learnt weights.**

The first layer learns single, oriented ridges. We empirically chose the number of features in the first layer to be 60 because the training has combinations of many orientations and frequencies. The weights learnt by the first layer are shown in Figure 3. The size of weights used in the first layer was 10 pixels × 10 pixels. To achieve this set of weights, we used a target sparsity of 0.003.

The second layer learnt higher level features than the first. This is because of a pooling operation on the hidden units of the first layer which down-samples training images after the hidden units. This factor by which the image is downsampled was set to 2 in our implementation. Further, the second layer had 60 features, each of size 10 pixels × 10 pixels. A weight now covers a larger portion of the input image. Weights learnt in the second layer are visualised in Figure 3. A target sparsity of 0.005 was used for this layer. The second layer is essential in drawing hierarchical probabilistic inference of an image, and “filling up” regions of the fingerprint image which could not be reconstructed by the first layer alone.

### 4.3 Reconstructing Fingerprints

Now that we have trained a model, we can use it to reconstruct (enhance) fingerprint images. These reconstructions serve as the enhancement which we are targeting. To reconstruct an image, we show it to the visible units of the first layer, and calculate the network’s representation of it. We then reconstruct the image from this representation using hierarchical probabilistic inference, in which we iterate over 20 steps of block Gibbs sampling (Figure 4).

We observe the reconstructions computed by the network at every iteration of block Gibbs sampling, and we see that the second layer progressively reconstructs regions that the first layer alone was unable to represent (Figure 5).

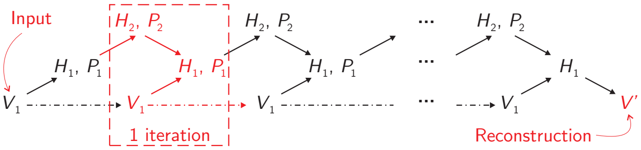

**Figure 4: Enhancing a fingerprint using a learnt model. Solid arrows indicate drawing an inference of the quantities they are directed at, using the quantities at their bases. Dotted arrows simply carry-forward the current values of variables.**

### 4.4 Computing Intrinsic Images

Now that we are able to perform a reconstruction of the fingerprint image using the convolutional deep belief network, we show a method to estimate intrinsic images of the fingerprint, viz., the orientation field, the frequency image, and the region mask, using weights learnt in the first layer. First layer features are oriented ridges, and hence have a ridge orientation and ridge frequency associated with them. The associated orientation and frequency for a weight can be found using the Fourier transform of the weight. To get a reasonable association of a weight with an orientation and frequency, we calculate a weighted average of the orientations and frequencies given by the Fourier transform, weighed according to the energy of the transform corresponding to every orientation. Chikkerur, Cartwright, and Govindaraju [5] show a method to use the Fourier transform for this task. We borrowed from their method to associate orientations and frequency with a weight. If $\Phi_k$ is the shifted Fourier transform of weight $W_1^{1,k}$ and $\theta$ is the matrix signifying the orientation angle for every point in $\Phi_k$, the orientation associated with $W_1^{1,k}$ is given by

$$d_k = \frac{1}{2}\arctan\frac{\sum\sin\left(2\theta\cdot|\Phi_k|^2\right)}{\sum\cos\left(2\theta\cdot|\Phi_k|^2\right)}, \tag{2}$$

where $\cdot$ is element-wise multiplication of two matrices.

Frequency, $f_k$, for a feature can be calculated from the Fourier transform too. We use the distance of peaks in the power spectrum from the centre - $R$ - instead of $\theta$.

$$f_k = \frac{1}{W_s}\frac{\sum R\cdot|\Phi_k|^2}{\sum|\Phi_k|^2}, \tag{3}$$

where $W_s$ is the size of a weight in the first layer, and also the size of the computed Fourier transform of a weight. Once we have associated an orientation and a frequency with every weight in the first layer, we perform hierarchical probabilistic inference. The reconstruction, $\mathbf{v}'$, of an image is given by $\mathbf{v}' = \sum_{k=1}^{K} W_1^{1,k} * \mathbf{H}_1^k$. To calculate the orientation field, we calculate a weighted average of the orientations from $K$ groups of hidden units, weighed according to $W_1^{1,k} * \mathbf{H}_1^k$. An estimate of the orientation field ($\mathbf{D}'$) is obtained using Equations 4, 5, and 6:

$$\mathbf{D}_s = \left(\sum_{k=1}^{K}\sin(2d_k)\left(W_1^{1,k} * \mathbf{H}_1^k\right)\right)\oslash\mathbf{v}' \tag{4}$$

$$\mathbf{D}_c = \left(\sum_{k=1}^{K}\cos(2d_k)\left(W_1^{1,k} * \mathbf{H}_1^k\right)\right)\oslash\mathbf{v}' \tag{5}$$

$$\mathbf{D}' = \frac{1}{2}\arctan\left(\mathbf{D}_s \oslash \mathbf{D}_c\right), \tag{6}$$

where $\oslash$ is element-wise division. We use sines and cosines of $d_k$ so that we average a continuous variable (the orientation field values jump abruptly from 0 to $\pi$). Finally, we smoothen the complete orientation image using a 10 × 10 Gaussian smoothing kernel. The final orientation field image, $\mathbf{D}$, is given by:

$$\mathbf{D} = \frac{1}{2}\arctan\left(\frac{G * \sin(2\mathbf{D}')}{G * \cos(2\mathbf{D}')}\right) \tag{7}$$

The frequency image can be estimated in a similar way, using $f_k$ instead of $d_k$:

$$\mathbf{F}' = \left(\sum_{k=1}^{K} f_k\left(W_1^{1,k} * \mathbf{H}_1^k\right)\right)\oslash\mathbf{v}' \tag{8}$$

To estimate the region mask, we note that the network performs reconstructions of only those regions that match a learnt fingerprint pattern. Hence, the region mask can be thought of consisting of all those pixels where the convolutional DBN performs reconstruction. A pixel $(x, y)$ is set to 1 in the region mask, $\mathbf{M}$, if there is a response from the network at that pixel. The region mask can thus be defined by Equation 9.

$$[\mathbf{M}]_{x,y} = \begin{cases} 1; & \text{if } [\mathbf{v}']_{x,y} \neq 0 \\ 0; & \text{if } [\mathbf{v}']_{x,y} = 0 \end{cases} \tag{9}$$

## 5. EXPERIMENTS AND RESULTS

In this section, we evaluate our model in terms of the performance of a matching algorithm on it. We will also compare our results with existing Gabor-based and STFT-based enhancement algorithms. We will plot Receiver Operating Characteristics (ROC) for the performances on some standard FVC (Fingerprint Verification Competition) datasets ([16, 17]).

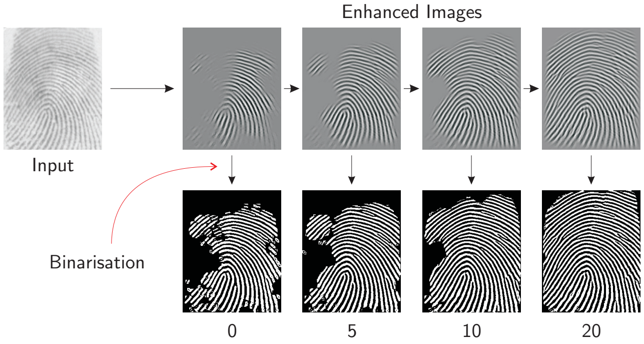

**Figure 5: Reconstructions of a fingerprint image as the number of iterations of alternate Gibbs sampling in HPI increases, shown at various iterations. 0 iterations correspond to reconstruction using only the first layer.**

## 5.1 Qualitative Analysis

We perform a qualitative analysis of our algorithm on images from three datasets: FVC2000 Db1_a and Db2_a [16], and FVC2002 Db3_a [17]. Images from the dataset were shown to the trained convolutional deep belief network, and reconstructions of these images were recorded.

Figure 5 is an example of the reconstruction performed by a trained network. "CDBN-$k$" denotes the reconstructions obtained using $k$ iterations of block Gibbs sampling in HPI. The binary images in the figure have been generated by thresholding the reconstructed images according to their mean values. Pixel values greater than the mean constitute valleys, while the rest of them constitute ridges.

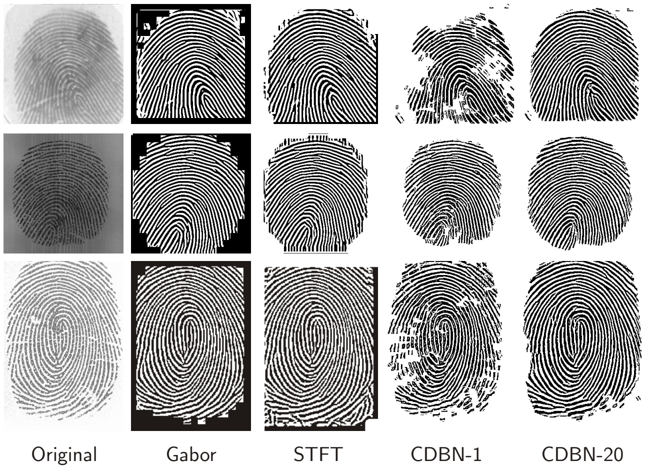

**Figure 6: A comparison of enhancements and binarisations performed using Gabor [10], STFT analysis [5], CDBN-1 and CDBN-20.**

Figure 6 shows a comparison between Gabor-based enhancement, STFT analysis, and reconstructions using the CDBNs for 1 and 20 iterations of Gibbs sampling in hierarchical probabilistic inference. We see that using higher-level data, we can "fill-up" regions of the fingerprint which the first layer is not able to comprehend. The first layer is "helped" by the higher layers in filling up this data. However, it is important that this filling up should not destroy information of minutiae present in the input image. To achieve this, we propose a modification to the hierarchical probabilistic inference (HPI) algorithm. HPI involves partitioning all variables in all the layers into two disjoint sets and performing block Gibbs sampling on these variables (Section 3.3). We propose that during each iteration of block Gibbs sampling, the visible units be replaced by the input image, instead of using the values obtained from other units in the network. We saw that this modification to the reconstruction algorithm keeps the fingerprint structure intact, and tremendously improves enhancement. We also do not get spurious minutiae, which result from incorrect enhancements of fingerprints.

Using HPI, we are able to remove minor creases and ridge discontinuities from fingerprint images. The hierarchical inference also smoothens ridges in the enhanced images by removing a majority of the pores.

## 5.2 Quantitative Analysis

The proposed algorithm was evaluated by conducting a matching exercise on fingerprints from standard datasets enhanced using the network. We used two datasets: FVC2000 Db1_a and Db2_a [16]. For minutiae extraction, we used the software FingerJetFXOSE developed by Digital Persona Inc. [1]. Fingerprint matching was performed using the NIST matcher, *bozorth3* [18]. To compare our approach with existing techniques, we have also stated the performance of Gabor-based enhancement [10] and short time Fourier transform analysis [5] on the same datasets. Receiver operating characteristics for the three algorithms are given in Figure 7. This graph describes reconstructions using the CDBN for four values of number of iterations in HPI. Equal error rates for the three approaches on these two datasets are tabulated in Table 1.

It is worth noting that the enhancements performed by the CDBN are better on the dataset FVC 2000 Db1_a, the images of which were a part of the training dataset. Furthermore, the equal error rate for FVC 2002 Db3_a is comparable to Gabor and STFT analysis, even though images from this dataset were not a part of the training set. We have compared the results with the feature learning approach proposed by Sahasrabudhe and Namboodiri [21] too. We also observe that including the second layer in the CDBN substantially increases the matching accuracy and enhancement performance.

| **Method ↓** | **2000 1_a** | **2000 2_a** | **2002 3_a** |
|---|---|---|---|
| **Gabor [10]** | 8.47 | 7.71 | 24.34 |
| **STFT [5]** | 8.79 | 7.98 | 21.99 |
| **CRBMs [21]** | 7.10 | - | 22.65 |
| **CDBN-20** | 6.62 | 8.52 | 23.95 |
| **CDBN-10** | 8.19 | 9.14 | 25.00 |
| **CDBN-5** | 9.59 | 10.24 | 25.45 |
| **CDBN-1** | 10.57 | 10.66 | 24.48 |

**Table 1: Equal error rates (in percentage) for the performance of the proposed algorithm.**

Besides evaluating our approach using a fingerprint matching exercise, we conducted an evaluation using the detected minutiae too. The enhanced images were subject to feature extraction using FingerJetFXOSE (referenced in the previous section), and a count of the total detected minutiae, and

[1] `https://github.com/FingerJetFXOSE/FingerJetFXOSE`

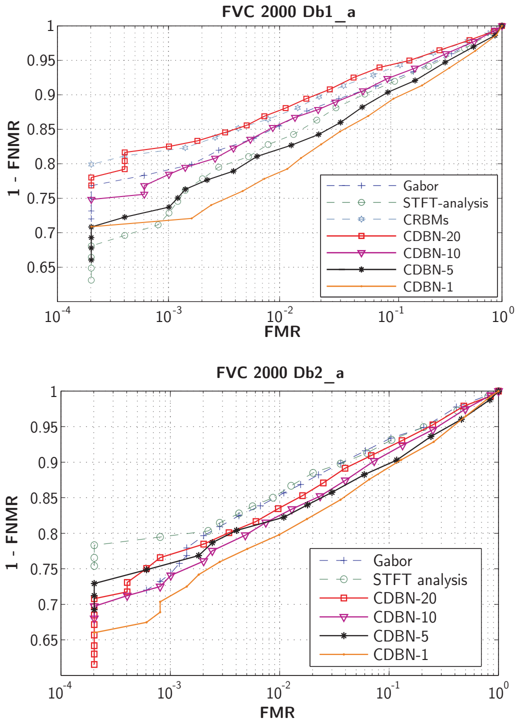

**Figure 7: Receiver Operating Characteristics plotted for the proposed algorithm, and compared with Gabor [10] and STFT-analysis [5]. The graphs plot the True Match Rate (1 - FNMR) against the False Match Rate (FMR).**

total spurious and missing minutiae extracted was made. We performed this exercise on the FVC 2002 Db3_a [17] dataset. To count the number of spurious and missing minutiae, we used ground truth data provided by Kayaoglu *et al.* [11] for this dataset. We record the observed values in Table 2.

| Method→ | Gabor | STFT | CRBMs | CDBN-20 |
|---|---|---|---|---|
| **Actual** | | $---19032---$ | | |
| **Detected** | 52560 | 48963 | 41764 | **31151** |
| **Spurious** | 37951 | 33096 | 26324 | **16211** |
| **Missing** | 4423 | **3165** | 3592 | 4092 |

**Table 2: A comparison of the number of spurious and missing minutiae detected by Gabor-based enhancement [10], STFT analysis [5], CRBMs [21], and the proposed approach on the FVC 2002 Db3_a dataset.**

It is interesting to note that images from FVC 2002 Db3_a were again not a part of the training set used to train the CDBN. Further, FVC 2002 Db3_a is a dataset with relatively noisy fingerprints. Our approach using the convolutional deep belief network led to fewer minutiae being detected than either STFT analysis or Gabor-based enhancement.

## 5.3 Varying the Number of Features in Layers

The number of features, or weights, in every layer plays a crucial role in the reconstructions achieved by that layer. A higher number of weights directly means that more features will be learnt from the training data, and hence, we will achieve better reconstructions. However, it is also essential to show that this holds true based on quantitative analysis. We experimented with several networks, varying the number of weights in each one of them. We compare our results with two more networks - one with 20 weights in layer 1 and 30 weights in layer 2, and another with 40 weights in layer 1 and 60 weights in layer 2. The weights learnt by these networks are displayed in Figure 8.

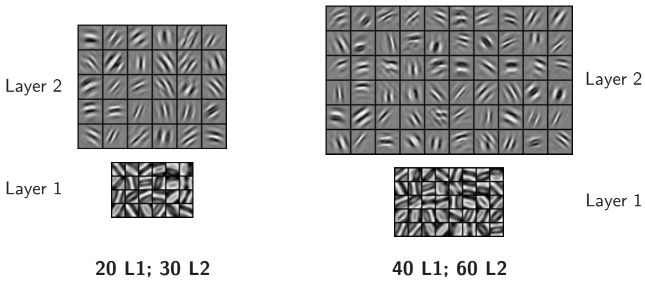

**Figure 8: Two networks trained on the same training data with different number of weights; with *left:* 20 features in layer 1, and 30 features in layer 2, and; *right:* 40 features in layer 1, and 60 features in layer 2.**

It can be seen from Figure 8 that having fewer weights in the first layer restricts the number of learnt oriented ridges, and hence the network is not able to capture all orientations of fingerprint ridges. It also restricts the number of frequencies. Overall, the learnt filters will show a good response at fewer places in test images, as compared to networks that have more weights.

| Network | FVC 2000 Db1_a | FVC 2000 Db2_b |
|---|---|---|
| 20×30 | 11.47 | 22.48 |
| 40×60 | 8.31 | 16.48 |
| 60×60 | 6.62 | 10.44 |

**Table 3: Equal error rates (in percentage) for three different networks on two datasets.**

We perform the matching exercise on images reconstructed by these two networks too. It is observed that as the number of weights in the layers increases, the quality of reconstruction also increases significantly. The ROC curves and EERs are superior for the network with 60 weights in both layers than the other two. Further, there is a significant qualitative improvement as the number of weights is increased. Figure 10 shows comparative ROC curves for the three networks, with equal error rates being tabulated in Table 3 (The notation $a \times b$ denotes a network with $a$ features in layer 1 and $b$ features in layer 2).

A qualitative comparison of reconstruction of the same fingerprint using the three networks is given in Figure 11. As hypothesised, we find that using more weights in a layer increases the learning capacity of the convolutional deep belief network, and the quality of its reconstructions also increases.

## 5.4 Intrinsic Images

We can estimate the orientation field, frequency image, and region mask for an input fingerprint using the learnt weights. The estimation is done using equations described in Section 4.4. We only require the visible and hidden units

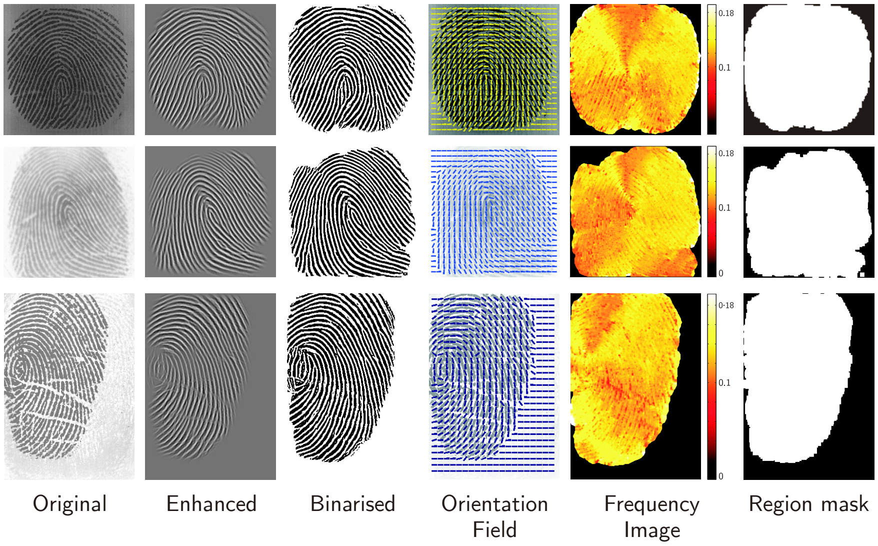

**Figure 9: Intrinsic images inferred using the convolutional DBN. Each row shows these intrinsic images generated from the left-most image of the row.**

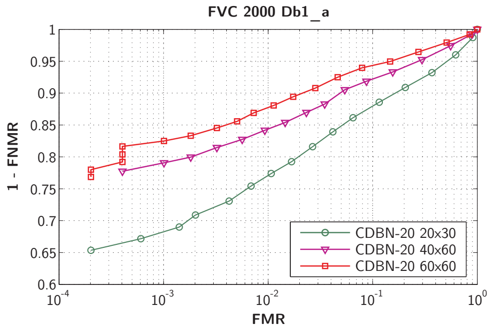

**Figure 10: ROC curves for three networks on the FVC 2000 Db1_a dataset.**

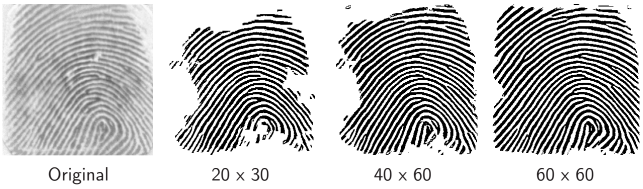

**Figure 11: Reconstructions of the same fingerprint using three different networks. All of the shown reconstructions are after 20 iterations of hierarchical probabilistic inference.**

in the first layer to compute the intrinsic images. Figure 9 shows the estimates of orientation field, frequency image, and region mask for some fingerprints.

We also see how the reconstruction is affected by the presence of more layers in the convolutional DBN, hence giving better estimates of orientation field, frequency image, and region mask.

Of particular importance here is the region mask obtained from the reconstructions. Several fingerprint segmentation algorithms used currently work by dividing the images into blocks. However, using a trained convolutional deep belief network, we are able to find the region mask with a precision of one pixel.

We evaluated the estimation of intrinsic images by comparing them with orientation field, frequency image and region mask computed in the Gabor-based ([10]) enhancement. This was done by replacing the orientation field, frequency image and region mask in Gabor-based enhancement by those computed by the convolutional DBN (CDBN-20). The usual Gabor filtering was then applied on fingerprint images. We show receiver operating characteristics (Figure 12) obtained by this modification to the Gabor-based enhancement, and compare them with CDBN-20, [10] and [5].

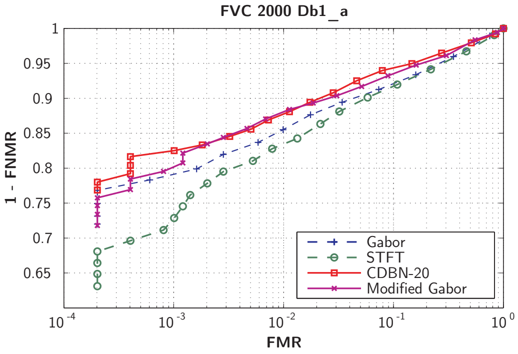

**Figure 12: ROC curves for the modified Gabor-based enhancement compared with Gabor, STFT and CDBN-20**

The recorded EER for this technique was 7.40, which is better than 8.79 and 8.47 for STFT-analysis, Gabor enhancement, respectively. Hence, replacing the intrinsic images in Gabor-based enhancement using those computed from the network improved the fingerprint matching results. This shows that the orientation field computed by the network is better than the gradient-based approach adopted by Gabor enhancement. However, CDBN-20, with an EER of 6.62, shows better results than the "modified" technique. This can be attributed to better filtering on part of the convolutional DBN.

## 6. CONCLUSION AND FUTURE WORK

In this paper, we presented the use of convolutional deep belief network for unsupervised feature learning of fingerprint images. As convolutional deep belief networks are generative models, we are able to perform reconstructions of fingerprint images using what the network learns. These reconstructions serve as enhancements of these images. Due to the convolutional nature of the CDBN, which goes hand-in-hand with the traditional contextual filtering approach applied for fingerprint enhancement so extensively, the reconstructions using a learnt network are also in-line with Gabor- and Fourier transform-based enhancements. We have also shown that adding higher layers to our network significantly improves the quality of enhancement. Experiments showed that the matching accuracy on enhanced images was better than Gabor-based enhancement on three datasets.

Future work on this path includes adding more layers to the network so that extremely noisy fingerprints can also be recovered. This can also branch out into several other areas of fingerprint recognition, e.g., classification (using extracted features), fingerprint segmentation and minutiae extraction (segmentation of minutiae points only), and synthetic fingerprint generation (by setting a representation and computing an image). It would be interesting to evaluate the usefulness of such a system on a large scale, efficient architecture, with a large training set and more features and layers.

## 7. REFERENCES

[1] S. Bernard, N. Boujemaa, D. Vitale, and C. Bricot. Fingerprint segmentation using the phase of multiscale gabor wavelets. In *The 5th Asian Conference on Computer Vision, Melbourne, Australia*. Citeseer, 2002.

[2] R. Cappelli, M. Ferrara, and D. Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 2128–2141, 2010.

[3] M. A. Carreira-Perpinan and G. E. Hinton. On contrastive divergence learning, 2005.

[4] K. chih Lee and S. Prabhakar. Probabilistic orientation field estimation for fingerprint enhancement and verification. In *Biometrics Symposium, 2008. BSYM '08*, pages 41–46, Sept 2008.

[5] S. Chikkerur, A. N. Cartwright, and V. Govindaraju. Fingerprint enhancement using stft analysis. *Pattern Recognition*, 40(1):198–211, 2007.

[6] S. Dass. Markov random field models for directional field and singularity extraction in fingerprint images. *Image Processing, IEEE Transactions on*, 13(10):1358–1367, 2004.

[7] G. Desjardins and Y. Bengio. Empirical evaluation of convolutional rbms for vision. *DIRO, Université de Montréal*, 2008.

[8] J. Feng, J. Zhou, and A. Jain. Orientation field estimation for latent fingerprint enhancement. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(4):925–940, April 2013.

[9] S. Greenberg, M. Aladjem, D. Kogan, and I. Dimitrov. Fingerprint image enhancement using filtering techniques. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 3, pages 322–325. IEEE, 2000.

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

[11] M. Kayaoglu, B. Topcu, and U. Uludag. Standard fingerprint databases: Manual minutiae labeling and matcher performance analyses. *CoRR*, abs/1305.1443, 2013.

[12] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.

[13] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009.

[14] D. Maio and A. K. Jain. *Handbook of Fingerprint Recognition*. Springer, 2009.

[15] D. Maio and D. Maltoni. Neural network based minutiae filtering in fingerprints. In *Pattern Recognition, 1998. Proceedings. 14th International Conference on*, pages 1654–1658, 8 1998.

[16] D. Maio, D. Maltoni, D. Cappelli, J. L. Wayman, and A. K. Jain. Fvc2000: Fingerprint verification competition, August 2000.

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

[18] NIST. NIST biometric image software. `http://nist.gov/itl/iad/ig/nbis.cfm`.

[19] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311 – 3325, 1997.

[20] R. K. N. V. Rama and A. M. Namboodiri. Fingerprint enhancement using hierarchical markov random fields. In *Proceedings of the 2011 International Joint Conference on Biometrics*, Washington, DC, USA, 2011.

[21] M. Sahasrabudhe and A. M. Namboodiri. Learning Fingerprint Orientation Fields Using Continuous Restricted Boltzmann Machines. In *Proceedings of the 2nd Asian Conference on Pattern Recognition*, 2013.

[22] C. I. Watson and C. L. Wilson. NIST special database 4, fingerprint database, March 1992.

[23] J. Yang, L. Liu, T. Jiang, and Y. Fan. A modified gabor filter design method for fingerprint image enhancement. *Pattern Recognition Letters*, 24(12):1805–1817, 2003.

[24] E. Zhu, J. Yin, and G. Zhang. Fingerprint enhancement using circular gabor filter. In *Image Analysis and Recognition*, pages 750–758. Springer, 2004.