IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Optimized Deep Features For High-Precision Fingerprint Image Segmentation

[1]Reji C Joy

[1]Professor

[1]Department of Computer Applications

[1]Vidya Academy of Science and Technology, Thrissur, India.

Abstract: The segmentation of fingerprints is a crucial step in automating the identification process. It is necessary to separate the ridge and valley structure of the fingerprint from the background, which often contains irrelevant data that can interfere with accurate feature extraction. In the proposed approach, fingerprint segmentation is treated as a classification problem, where the input image is classified into either the foreground or background class. To achieve this, an unsupervised learning algorithm called Stacked Sparse Autoencoder is employed to learn deep features that effectively distinguish the background region from the foreground. These deep features are then inputted into the SVM classifier. The results of the experiments demonstrate that the proposed method achieves state-of-the-art performance across various applications.

***Index Terms* -** Autoencoder, fingerprint segmentation, morphology.

I. INTRODUCTION

The progress in technology and the necessity to ensure security in different application domains have opened up opportunities for the individual recognition and identification system through biometric features. The term 'Biometric' refers to the various automated methods that uniquely identify individuals based on measurable physiological or behavioral characteristics. Among the various biometric features like iris, face, fingerprint, voice, gait, etc., fingerprint stands out as the most common and popular choice. This is primarily due to the lower cost of fingerprint sensors and the fact that it is the only trace left by criminals after committing an offense. Achieving accurate automatic personal identification is crucial in application domains such as electronic-commerce, automatic banking, and national ID cards. Therefore, the utilization of a fingerprint as a biometric in an identification system is gaining significant importance.

Fig. 1 A fingerprint image

Each fingerprint consists of unique patterns of curved lines known as ridges located between two adjacent curved lines or a valley. A captured impression of the finger can be divided into two parts: the foreground, which represents the desired image, and the background, which typically contains unwanted noise along the image border (Fig 1). In order to match fingerprints, feature points are extracted from the acquired image. The two most commonly utilized feature points are minutiae and singular points. The feature extraction algorithm retrieves numerous features from the foreground, as well as false features from the background.

Scores of studies have been mentioned in the literature associated with fingerprint image segmentation. Bezen [1] has proposed a method which uses local pixel features like local mean, local variance and local coherence. Later these feature sets are linearly combined to obtain the desired result.

In [2], the given input fingerprint image is used to extract Gabor features. This method is capable of effectively extracting Gabor features from the foreground area. However, it is not economically viable. Another approach involves extracting a combination of local mean and local variance of gradient magnitude from the input fingerprint image. This technique also allows for the extraction of Gabor features from the foreground area. However, it suffers from the same drawback of being economically unfeasible. Fingerprint image segmentation is achieved by utilizing a combination of local mean and local variance of gradient magnitude [3]. In [4], a new 2D Feature Extraction technique for fingerprints is proposed, which involves using minutiae points and their intersections. In [5], the construction of a neural network layer known as MaxPoolingFragment is described, which is subsequently followed by a back-propagation procedure. This efficient algorithm utilizes MaxPooling Convolutional Networks to segment images through training. In [6], a modified SVM-based approach is introduced, leveraging the characteristics of support vectors to eliminate redundant training vectors simultaneously. This method proves effective in reducing the number of input training vectors while preserving the support vectors, particularly in real image scenarios. One notable advantage of this method is its low computational cost.

This paper presents a new approach to fingerprint segmentation using self-taught deep features. The self-taught learning technique proves to be valuable in various image processing domains, as it enables the extraction of high-level features from fingerprint images. In this study, a stacked sparse autoencoder is employed to automatically learn these features from the fingerprint image. These deep features possess significant discriminative power, allowing for the differentiation of foreground and background regions in the fingerprint image. To segment these regions, the deep features are inputted into a support vector machine classifier.

Additionally, a post-processing technique utilizing morphological operators is applied to identify the optimal fingerprint ridge-like regions. The entire work is divided into four sections: Section 2 outlines the feature extraction method employed for segmentation, Section 3 details the system flow and architecture of the proposed fingerprint image segmentation method, Section 4 presents the test results and performance analysis, and Section 5 concludes the paper.

II. DEEP FEATURES FOR HIGH PRECISION FINGERPRINT IMAGE SEGMENTATION

The distinctive arrangement of ridges and valleys in a fingerprint image causes variations in intensity, giving rise to a recognizable texture. This particular attribute of a fingerprint image frequently leads to a spatial correlation with nearby image intensities. By leveraging deep neural network architecture, it becomes possible to detect and extract these spatial relationships, thereby discerning the foreground of the fingerprint from its background.

2.1 Autoencoder

An autoencoder is specifically created to encode the provided input x into the hidden representation h through an encoding function, and then reconstructs the input using a decoder function. To achieve this, a network structure comprising of an input layer, hidden layer, and output layer is utilized. Consequently, an autoencoder qualifies as an unsupervised learning algorithm since it does not rely on labels during training.

Given an input $x \in R^l$ the encoder function takes the form:

$$h = S(Wx + b) \qquad (1)$$

where $S(\cdot)$ is a sigmoid function,

A decoder function has the form:

$$\hat{x} = S(W'h + b') \quad (2)$$

where W' and b' are the weight and bias parameters to the decoder function respectively.

Given that we are utilizing the autoencoder as a means of extracting features, this simple identity mapping fails to offer the distinguishing patterns necessary for effectively classifying the foreground and background of the fingerprint. Nevertheless, by imposing restrictions on the network, such as limiting the quantity of hidden units, the autoencoder begins to acquire a condensed representation and uncovers intriguing patterns within the input. The fundamental autoencoder may also result in overfitting of the data. This issue can be mitigated by incorporating a weight-decay regularization term into the objective function, which promotes smaller weights.

The compression task would pose a significant challenge if the input were entirely random. Nevertheless, due to the strong correlation among textures in a fingerprint image, the aforementioned model can acquire knowledge about these correlations and utilize them as features for fingerprint segmentation.

2.2 Sparse Autoencoder

The basic autoencoder can be subjected to a sparsity constraint, which compels it to uncover valuable features from the input data [7]. This variant of autoencoder, known as sparse autoencoder, incorporates sparsity constraints in the hidden nodes. In a neural network model, a neuron is deemed active when its output value reaches 1 and inactive when it reaches 0 [8], [9].

Stacked Sparse Autoencoder is a variant of neural network with multiple layers of sparse autoencoder. Here, the output of one layer forms the input of the adjacent layer [7]. By stacking all the layers, a stacked autoencoder behaves like a deep network to identify some complex features in the input data. In this work, two hidden sparse autoencoder networks are connected with an input layer and an output layer. Finally, all the different layers are stacked to form a deep network. The architecture of SSAE is shown in Fig 2. As we are interested only in the feature extraction, the decoder parts of the autoencoder are not represented.

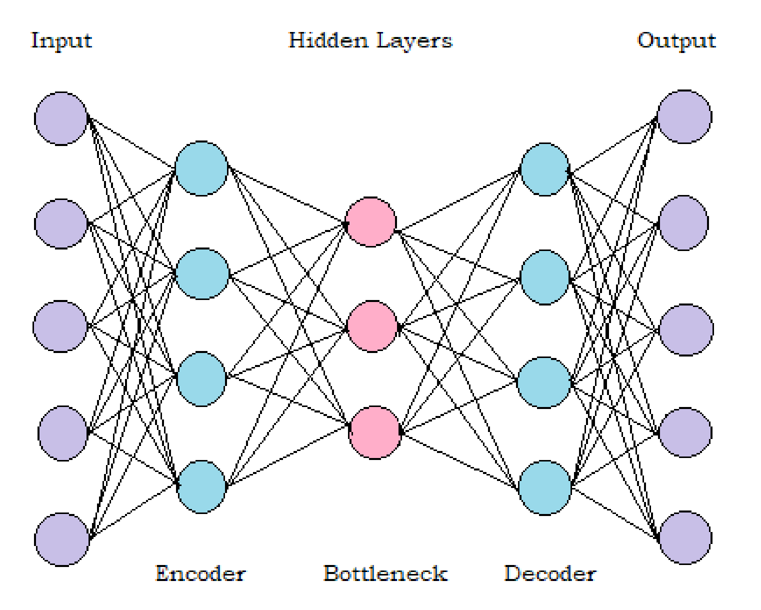


Fig. 2 Structure of Stacked Sparse Autoencoder

III. RESEARCH METHODOLOGY

The concept of fingerprint segmentation involves distinguishing between the foreground and background areas in a fingerprint image. Thus, in this study, we approach the task of fingerprint segmentation as a classification issue. Our proposed method consists of two phases. In the initial phase, a deep autoencoder is employed, which is a self-taught machine learning mechanism, to extract features. These extracted features offer more personalized representations compared to traditional feature extraction methods. In the second phase, these high-level features are inputted into a supervised learning algorithm to potentially classify the fingerprint regions. Numerous general supervised learning techniques [10] and fingerprint segmentation-specific methods [11] have been suggested in the literature. However, for the research, we have opted to utilize the Support Vector Machine (SVM) classifier, as the fingerprint segmentation problem strictly involves two classes. In the subsequent section, we provide a brief introduction to SVM.

3.1 Support Vector Machine (SVM)

During the testing phase, SVM employs the subsequent decision function to determine the class to which the new test data belongs.

$$f(X) = sgn(\sum_{i=1}^{N} a_i y_i \cdot K'(X, X_i) + b) \quad (3)$$

where N is the number of data points in the test input and a_i are the coefficients of the trained data set that describes the separating hyper plane. Here, $a_i > 0$ are called support vectors. When it is hard to find a separating hyperplane to separate the input test data, SVM can use a non-linear transformation. sgn(ψ) is the sign function.

3.2 Stacked Sparse Autoencoder (SSAE) with SVM

In this research paper, the concept of SSAE, which allows for the direct generation of features from image patches. These features serve as the feature vector for classification purposes. The size of each image patch is carefully determined to encompass a significant amount of fingerprint textural content while minimizing the likelihood of both foreground and background appearing within the same patch. The SSAE network consists of an input layer that feeds the image patches, as well as two hidden sparse autoencoders that produce the hidden representation. These layers are then stacked to create a deep network. Consequently, our deep network, initially trained with fingerprint image patches as input, produces a high-level feature representation known as the deep feature. This deep feature, along with the labeled data, is subsequently provided as input to the Support Vector Machine (SVM) for the final classification process.

During the initial stage of training, we implemented a greedy layer-wise pre-training approach in deep learning to generate the deep feature. The diagram in Fig 2 illustrates the architecture of this training process. The first Stacked Autoencoder (SAE) receives image patches from the input layer in order to learn the primary feature representation by adjusting the weight values. Consequently, it produces the first layer feature. These primary features are then passed on to the second SAE, which generates the higher level features in the output layer. As this strategy is an unsupervised learning technique, the higher level features, also known as the deep features [16], are not associated with any specific labels. In the subsequent phase, the deep features, combined with the labeled data, are utilized as input features for a supervised learning algorithm, such as Support Vector Machine (SVM), to perform the final classification.

3.3 Morphological Operations

Accurate segmentation of the fingerprint image into foreground and background is heavily reliant on the quality of the input image (refer to Figure 3(a)). When the classifier struggles to accurately determine the actual class, the segmentation process may result in the generation of spurious regions, where regions of one class appear within regions of another class (refer to Figure 3(b)). However, these unwanted regions can be eliminated by considering the smaller regions as part of the larger regions they belong to, thereby forming a meaningful cluster. In this study, morphological operations [12] are employed as a post-processing technique to obtain the most optimal cluster regions.These operations, which involve dilation and erosion, are utilized to remove the spurious regions [16]. A morphological method proposed in a previous work [13] is utilized as the post-processing method to determine the ideal cluster regions.

IV. Experimental Results

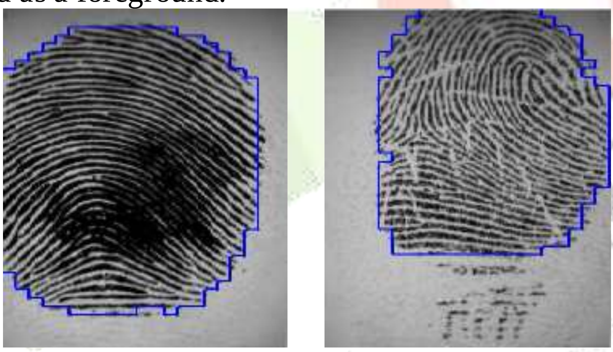
The performance evaluation of the proposed fingerprint segmentation algorithm was conducted using the standard FVC 2002 Db2_a dataset. This dataset comprises 800 images, each depicting one of eight different fingerprint impressions from 100 individuals. Each fingerprint image has a resolution of 500×500 pixels. To train the algorithm, the fingerprint images were manually divided into 28×28 image patches, from which 10 patches were randomly selected from both the foreground and background areas. Additionally, 5 image patches were chosen for testing. In total, 8000 image patches were used for training, while 4000 image patches were reserved for testing purposes. The performance of the SSAE deep network was assessed based on various parameters. Since there are no definitive guidelines for determining the optimal values of these parameters, random methods were employed for their selection. The design of our SSAE deep network allows the input layer to accept 28×28 image patches and generate a feature vector of size 50 for each input data set. This feature vector serves as the input for the SVM. During the training phase, a 5-fold cross-validation technique was applied to the training set.

The results of the proposed segmentation algorithm are compared to calculate the misclassification to quantify the algorithm performance. The misclassification can be quantified as:

$$\begin{aligned} p(\hat{\tau}_1|\hat{\tau}_0) &= \frac{b_{error}}{b_n} \\ p(\hat{\tau}_0|\hat{\tau}_1) &= \frac{f_{error}}{f_n} \\ p_{err} &= \frac{p(\hat{\tau}_1|\hat{\tau}_0) + p(\hat{\tau}_0|\hat{\tau}_!)}{2} \end{aligned} \tag{4}$$

The algorithm's performance is evaluated by considering the number of blocks that are incorrectly classified as backgrounds when they are actually foregrounds, denoted as $ferror$. The number of true foreground blocks that are correctly identified from the image, denoted as fn, is also taken into account. The probability of foreground classification error, represented as $p(\tau\hat{}1 \mid \tau 0)$, provides an estimation of the likelihood of misclassifying a foreground block as a background.

Similarly, the algorithm's accuracy can be assessed by examining the number of blocks that are mistakenly classified as foregrounds when they are true backgrounds, referred to as $berror$. The number of true background blocks that are correctly identified from the image, denoted as bn, is also considered. The probability of background classification error, denoted as $p(\tau\hat{}0 \mid \tau 1)$, provides an estimation of the likelihood of misclassifying a background as a foreground.



(a) (b)

Fig. 3 Results after the segmentation

In order to assess the accuracy of the suggested algorithm in classifying fingerprints, a set of 10 randomly chosen fingerprint images were obtained from the Db2_a dataset. Additionally, the morphological operations were performed. An overview of the error probabilities prior to and following the application of morphological operations on the same set of 10 fingerprint images is given. The findings indicate that the overall rate of misclassification is remarkably minimal.

The singular points, also referred to as core and delta points, are one of the many crucial features of the fingerprint image in any AFIS. These singular points serve as distinctive landmarks within the fingerprint image, playing a significant role in fingerprint classification and matching. However, it is worth noting that singular points, particularly the delta points, are often located in close proximity to the image boundary. Consequently, it is of utmost importance for any segmentation algorithm to accurately identify and include these points during the segmentation process. To assess the effectiveness of the proposed algorithm, a test was conducted on the FVC2002 Db2_a dataset to determine whether the foreground region contains these feature points after undergoing segmentation.

V. CONCLUSION

A novel method is proposed for fingerprint image segmentation using deep features and SVM. In this study, stacked sparse auto encoders are utilized to learn deep features, which are subsequently inputted into an SVM classifier. The success of any AFIS heavily relies on the results of this preprocessing step, as it gathers reliable features. Initially, a greedy layer-wise training approach is employed to learn deep features in the first phase. Then, in the second phase, the deep features are processed by the SVM-supervised classifier. Experimental results demonstrate that the input fingerprint image achieves state-of-the-art outcomes across all domains of applications.

VI. References

[1] Bazen and S. Gerez, "Segmentation of fingerprint images," in ProRISC 2001 Workshop on Circuits, Systems and Signal Processing, Veldhoven, TheNetherlands, 2001.

[2] F. Alonso-Fernandez, J. Fierrez-Aguilar and J. Ortega-Garcia, "An enhanced Gabor filter-based segmentation algorithm for fingerprint recognition systems," in Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA2005), 2005.

[3] Saparudin, "Segmentation of Fingerprint Image Based on Gradient Magnitude and Coherence," IJECE, 2015.Nibras, "A Novel 2D feature extraction Method for Fingerprints Using Minutiae Points and Their Intersections," IJECE, 2017.

[4] M. Jonathan, "A fast learning algorithm for image segmentation with max-pooling convolutional networks," arXiv:1302.1690v1[cs.CV], 2013.

[5] Y. Zhiwen, "A modified support vector machine and its application to image segmentation," Image and Vision computing, 2011

[6] N. Andrew, "Cs294a lecture notes: Sparse autoencoder," 2010. [Online].

[7] N. Andrew, "Sparse Autoencoder," 2011. [Online]. Available: https://web.stanford.edu/class/cs294a/sparseAutoencoder_2011new.pdf. [Accessed 22 10 2017].

[8]N. Andrew, "Sparse Autoencoder," 2017. [Online]. Available: http://web.stanford.edu/class/archive/cs/cs294a/cs294a.1104/sparseAutoencoder.pdf. [Accessed 22 10 2017].

[9] Maltoni, D. Maio, A. Jain and S. Prabhakar, Handbook of fingerprint recognition (Second Edition), New York: Springer, 2009.

[10] R. Gonzalez and P. Wintz, Digital Image Processing.2nd Edition, Addison-Wesley, 1987, pp. 153-160.

[11] C. Reji and S. Hemalatha, "A Gradient Based Approach for Fingerprint Image Segmentation using Morphological Operators," International Journal of Engineering & Technology, 2018.

[12] C. Gravelines, "Deep learning via stacked sparse autoencoders for automated voxel-wise brain parcellation based on functional connectivity," Electron. Thesis Diss. Repos., pp. 1-75, 2014.

[13] V. Vapnik, Statistical Learning Theory, New York: Wiley, 1998.

[14] P. Soille, Morphological Image Analysis: Principles and Applications, Springer-Verlag, 1999.

[15] Y. Bengio, "Learning deep architectures for AI," Found. Trends. Mach. Learn., vol. Vol.2, pp. 1-127, 2009.

[16] C. J. Reji and M. Azath, "Fingerprint Image Segmentation using Textural Features," in Proceedings of International Conference on Computer Vision and Image Processing, Springer Nature, 2017.