1 INTRODUCTION

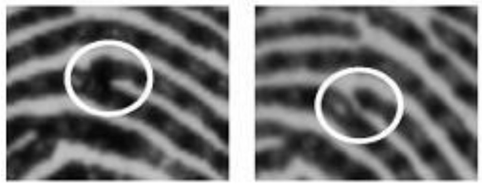
In this era, automatic access of people to services becomes important. As a result, the new technological area has established known as biometric recognition or biometrics [3]. The main goal of biometrics is to discriminate subjects for an application based on various signals obtained from physical or behavioural traits, such as fingerprint, face etc. It has several advantages when compared to classical security methods. In biometric system, we have to remember the large PIN code that may easily forget or a key that may lose or stolen [4] .Nowadays fingerprint can be spoofed easily using materials like gelatin, silicon, and wooden glue. Therefore, an effective fingerprint system will have the ability to distinguish a spoof from an authentic finger.

In biometric system liveness detection is used to determine whether the biometric being captured is the actual measurement from an authorized, live person who is present at the time when it is captured. No every system is efficient to prevent all attacks. But these liveness algorithms will reduce the spoofing [5].

Now there are several fingerprint liveness detection algorithms available, and they are mainly classified into two groups: Hardware and software. The hardware consists of a specified device attached to the sensor to detect particular features of a living traits such as blood pressure, skin distortion, or the odour. Software approach detects fake traits once the fingerprint sample is obtained from a standard sensor [6].

From the fingerprint sample, we will extract the features that we can use to distinguish between real and fake fingers. Some technique uses features such as ridge strength, continuity, and clarity of fingerprint [6]. The features exhibit uniqueness defined by type, position and orientation from fingerprint to fingerprint and they are classified into global and local features. Global features are those characteristics of the fingerprint that could be seen with the naked eye. They are the features that are characterized by the attributes that capture the global spatial relationships of a fingerprint. Global features include ridge pattern, type, orientation, spatial frequency, curvature, position and count. Others are type lines, core and delta areas.

The Local Features are also known as Minutia Points. They are the tiny, unique characteristics of fingerprint ridges that are used for positive identification. Local features contain the information that is in a local area only and invariant with respect to global transformation. It is possible for two or more impressions of the same finger to have identical global features but still differ because they have local features (minutia points) that are different. In Fig. 1.1, ridge patterns (a) and (b) are two different impressions of the same finger (person). A local feature is read as bifurcation in (a) while it appears as a ridge ending in (b)



(a) Bifurcation (b) Ridge ending

Fig. 1.1 Different minutiae for different impressions of the same finger

2 BACKGROUND

2.1 CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (ConvNets or CNNs) [3] are a classification of Neural Networks that have demonstrated exceptionally compelling in zones, for example, picture acknowledgment and grouping. ConvNets have been effective in distinguishing confronts, protests and activity signs separated from the controlling vision in robots and cars. ConvNets, in this way, are a critical instrument for most machine learning professionals today. Be that as it may, understanding ConvNets and figuring out how to utilize them interestingly can some of the time be a scary affair. The main role of this blog entry is to build up a comprehension of how Convolutional Neural Networks take a shot at the pictures. Convolution neural systems are basic devices for profound learning, and are particularly suited for picture acknowledgment. You can develop a ConvNet engineering, prepare a system, and utilize the prepared system to anticipate class names or numeric reactions. You can likewise separate elements from a pretrained system, and utilize these elements to prepare a straight classifier. Neural Network Toolbox likewise empowers you to perform exchange realizing; that is, retrain the last completely associated layer of a current ConvNet on new information.

A few smart errands, for example, visual recognition, dialect understanding, and sound-related discernment require development of good inward portrayals of components. These components must be invariant to superfluous varieties of the information. These components should likewise protect important data. A noteworthy impediment in machine learning was to decide how to learn such great elements consequently.

2.1.1 HISTORICAL BACKGROUND

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras.

Many important advances have been boosted by the use of inexpensive computer emulations. Following an initial period of enthusiasm, the field survived a period of frustration and disrepute. During this period when funding and professional support was minimal, important advances were made by relatively few researchers. These pioneers were able to develop convincing technology which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding. The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not allow them to do too much.

In 1949, Donald Hebb published *The Organization of Behavior*, which outlined a law for synaptic neuron learning. This law, later known as Hebbian Learning in honor of Donald Hebb is one of the simplest and most straight-forward learning rules for artificial neural networks.In 1951, Marvin Minsky created the first ANN while working at Princeton.In 1958 *The Computer and the Brain* was published posthumously, a year after John von Neumann’s death. In that book, von Neumann proposed many radical changes to the way in which researchers had been modeling the brain.

2.1.2 WHY USE CONVOLUTIONAL NEURAL NETWORKS ?

Convolutional neural systems utilize pictures straightforwardly as information. The convolution arranges plays out the capacities that are performed by cells in the visual cortex, for example, extricating basic visual elements like situated edges, end-focuses, corners, and so forth. Convolutional neural systems comprise of convolutional layers which remove helpful data from the information and take out superfluous fluctuation. Each phase in a convolutional system is made out of a channel bank, and highlight pooling layers. With numerous stages, a convolutional system can learn multi-level chains of importance of elements. Today convolutional neural systems are connected in an assortment of territories, including common dialect preparing, picture and example acknowledgment, video examination and discourse acknowledgment.

3 LITERATURE REVIEW

The existing liveness detection techniques are :

1. CNN Random.

   The CNN with random weights [1] uses only random filter weights drawn from a Gaussian distribution. Although the filter weights can be learned, filters with random weights can perform well and they do not need to be learned. It uses a convolutional network with random weights as the feature extractor, the dimensions are further reduced using PCA and a SVM classifier with RBF kernels used as the classifier. An extensive search for hyperparameter fine-tune was performed automatically on more than 2000 combinations of hyper-parameters.

2. Local Binary Patterns

   Local Binary Patterns (LBP) [4] are a local texture descriptor that have performed well in various computer vision applications, including texture classification and segmentation, image retrieval, surface inspection, and face detection. It is a widely used method for fingerprint liveness detection and it is used in this work as a baseline method. In its original version, the LBP operator assigns a label to every pixel of an image by thresholding each of the 8 neighbors of the 3x3-neighborhood with the center pixel value and considering the result as a unique 8-bit code representing the 256 possible neighborhood combinations. As the comparison with the neighborhood is performed with the central pixel, the LBP is an illumination invariant descriptor. The operator can be extended to use neighborhoods of different sizes. Another extension to the original operator is the definition of so-called uniform patterns, which can be used to reduce the length of the feature vector and implement a simple rotation invariant descriptor.

4 PROPOSED METHOD

The proposed system uses two techniques for the liveness detection. The two techniques are:

1. Convolutional Neural Network
2. Fingerprint Image Enhancement

The first technique is convolutional neural network which is state of art technique in image recognition. It involve mainly three steps:

i. Convolution
ii. Subsampling
iii. Pooling

The second step is fingerprint image enhancement. The steps used to enhance a fingerprint image are:

i. Image Segmentation
ii. Image Local Normalization
iii. Orientation Estimation
iv. Ridge frequency Estimation
v. Gabor Filtering
vi. Image Binarization/thinning

4.1 STRUCTURE

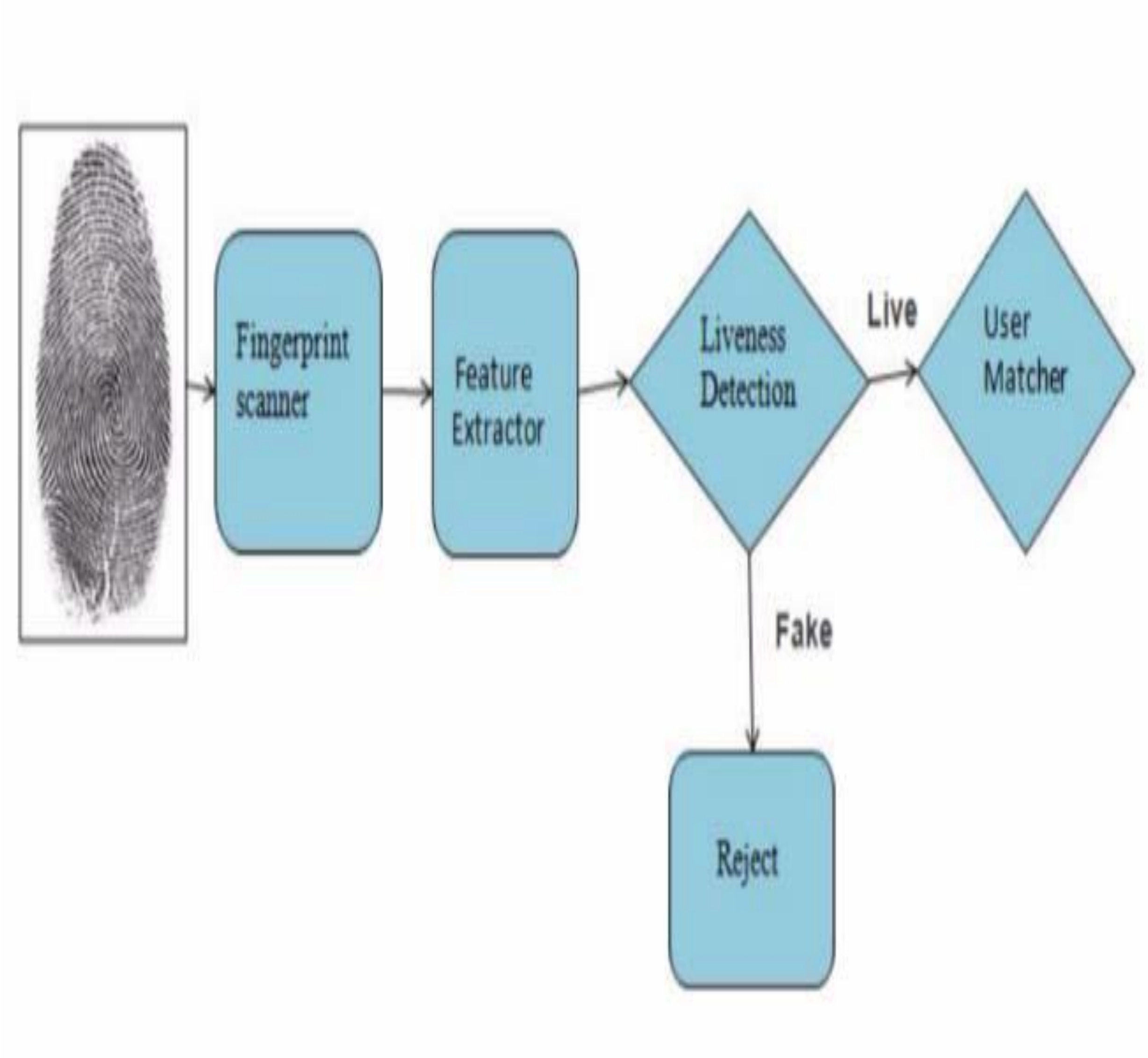


Fig. 4.1 The Overall Structure Of The Liveness Detection Technique

4.2 IMPLEMENTATION TECHNIQUES

4.2.1 CONVOLUTIONAL NEURAL NETWORK

Convolutional Networks [8] have demonstrated state-of the-art performance in a variety of image recognition benchmarks, such as MNIST , CIFAR-10, CIFAR-100, SVHN, and

ImageNet. A classical convolutional network is composed of alternating layers of convolution and local pooling (i.e., subsampling). The aim of a convolutional layer is to extract patterns found within local regions of the inputted images that are common throughout the dataset by convolving a template over the inputted image pixels and outputting this as a feature map c, for each filter in the layer. A non-linear function f(c) is then applied element-wise to each feature map c : a = f(c). A range of functions can be used for f(c), with max(0; c) a common choice. The resulting activations f(c) are then passed to the pooling layer. This aggregates the information within a set of small local regions, R, producing a pooled feature map s (normally of smaller size) as the output. Denoting the aggregation function as pool(), for each feature map c we have: $S_j = \text{pool}(f(c_i))\ \forall i \in R_j$, where Rj is the pooling region j in feature map c and i is the index of each element within it [1]. Among the various types of pooling, max-pooling is commonly used, which selects the maximum value of the region Rj .

The motivation behind pooling is that the activations in the pooled maps are less sensitive to the precise locations of structures within the image than the original feature map c. In a multi-layer model, the convolutional layers, which take the pooled maps as input, can thus extract features that are increasingly invariant to local transformations of the input image [9][10]. This is important for classification tasks, since these transformations obfuscate the object identity. Achieving invariance to changes in position or lighting conditions, robustness to clutter, and compactness of representation, are all common goals of pooling.

Fig 4.2 illustrates the feed-forward pass of a single layer convolutional network. The input sample is convoluted with three random filters of size 5x5 (enlarged to make visualization easier), generating 3 convoluted images, which are then subject to non-linear function max(x; 0), followed by a max-pooling operation, and subsampled by a factor of 2.

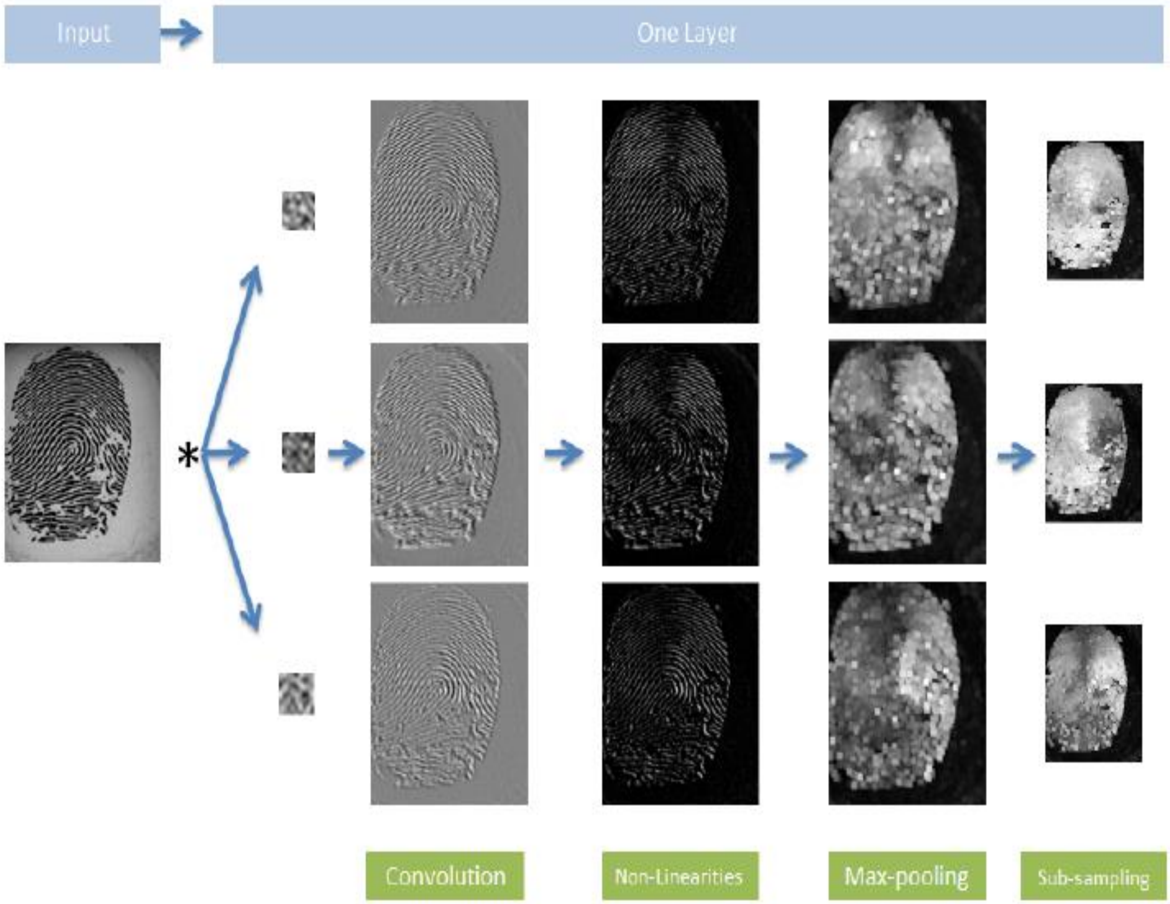


Fig. 4.2 Illustration of a chain of operations performed by a single layer convolutional network in a sample image.

4.2.1.1 CONVOLUTION

The layer which receives an input signal is called convolution filter. Convolution is said to be the process where the network labels the input signal by referring to the things it has learned in the past. The reference signal is mixed in, or convolved with, the inputted signal.

Convolution has the nice property of being translational invariant. This means that each convolution filter represents a feature of interest, and the CNN algorithm learns which features comprise the resulting reference. The output signal strength is independent on the location of the features, but on whether the features are present.

4.2.1.2 SUBSAMPLING

Inputs after passing through the convolution layer can be "smoothened" to reduce the sensitivity of the filters against noise and variations. This process is called subsampling, and it can be obtained by taking average or taking the maximum over a sample of the signals. Examples of subsampling methods (for image signals) include reducing the size of the image or reducing the color contrast across RGB channels. Downsampling is used here which is associated with the process of resampling in a multi-rate digital signal processing system. It is used by various authors to describe the entire process, which includes lowpass filtering, or just the part of the process that does not include filtering. When downsampling (decimation) is performed on a sequence of samples of a signal or other continuous function, it produces an approximation of the sequence that would have been obtained by sampling the signal at a lower rate (or density, as in the case of a photograph).

4.2.1.3 POOLING

Three different types of pooling/sub-sampling are commonly used: average, max and stochastic pooling. Average Pooling outputs the average of the activations units in each patch. Max Pooling outputs the maximum value of each patch (shown in Fig. 4.3) [7]. Stochastic Pooling randomly draws a value from each patch based on a distribution where higher pixel values have higher probabilities of being chosen. The patches obtained from the previous convolutional layer are used for pooling process.

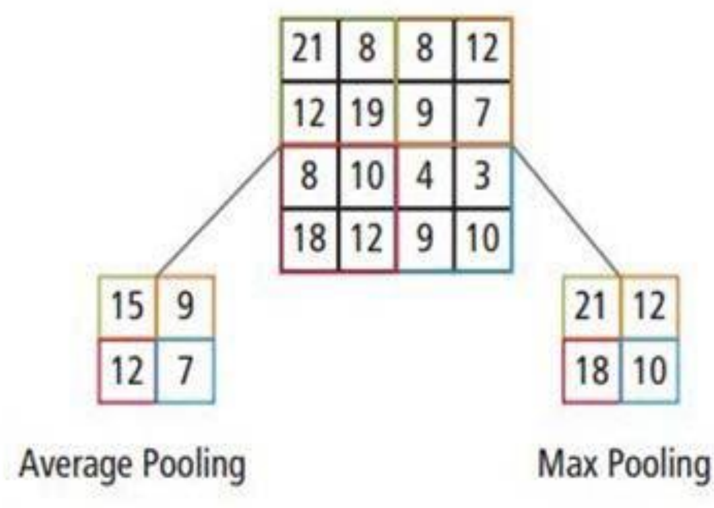


Fig. 4.3 Pictorial representation of max pooling and average pooling

4.2.1 FINGERPRINT IMAGE ENHANCEMENT

Reliable and sound verification of fingerprints in any Automatic Fingerprint Identification Systems (AFIS) is always preceded with a proper detection and extraction of its features. A fingerprint image is firstly enhanced before the features contained in it could be detected or extracted. A well enhanced image will provide a clear separation between the valid and spurious features. Spurious features are those minutiae points that are created due to noise or artifacts and they are not actually part of the fingerprint [2]. The overview of the algorithm is shown in Fig. 4.4. Its main steps include image segmentation, local normalization, filtering and binarization/thinning.

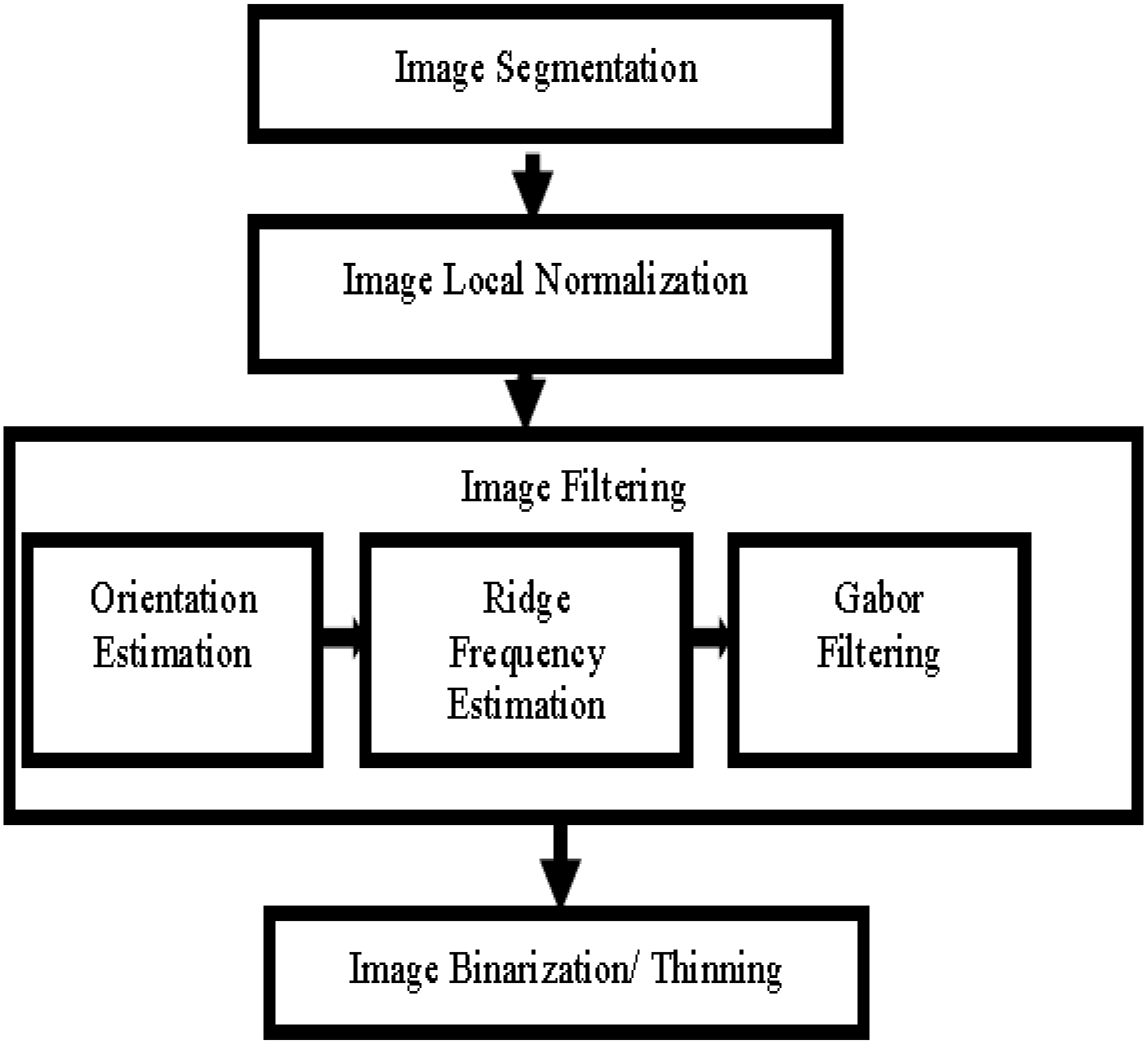


Fig. 4.4 The notional diagram of the fingerprint enhancement algorithm

4.2.2.1 IMAGE SEGMENTATION

One of the most important pre-processing steps in an Automatic Fingerprint Identification System is the fingerprint segmentation. It separates a fingerprint area (foreground) from the image background. An accurate segmentation of a fingerprint greatly reduces the computation time of the successive processing steps and discard many invalid minutiae. The two regions that represents a fingerprint image are the foreground region and the background region. The regions having the ridges and valleys are the foreground regions. As shown in Fig.4.5, the ridges are darker and raised regions of a fingerprint image and the valleys are the white and low regions between the ridges.

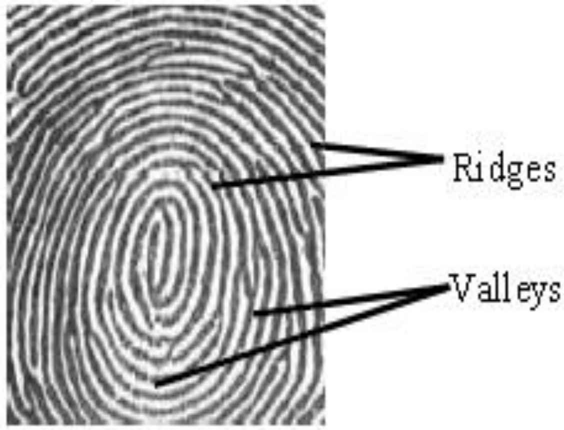


Fig. 4.5 Ridges and valleys on a fingerprint image

The foreground regions also known as the Region of Interest (RoI) and the background regions are the outside regions, where the noises injected into the image during enrollment are found are given in the image presented in Fig. 4.6.

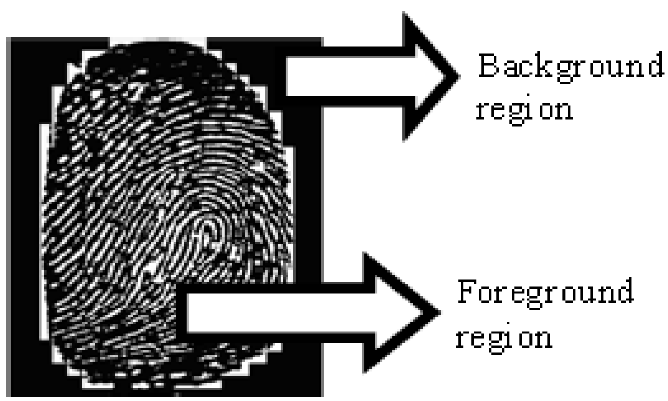


Fig. 4.6 A fingerprint image and its foreground and background regions

The main purpose of segmentation is to reduce the load associated with image enhancement by keeping the focus only on the foreground regions and the background regions are ignored. The background regions possess very low grey-level variance values while the foreground regions possess very high grey-level variance values. A block processing approach used in [11]-[12] is adopted in this research for obtaining the grey-level variance values. The approach firstly divides the image into blocks of size W x W and then the variance V(k) for each of the pixels in block k is obtained.

4.2.2.2 IMAGE LOCAL NORMALISATION

On the segmented fingerprint image ridge structure normalization is performed in order to standardize the level of variations in the image grey-level values. As a result of normalization, we can make fall the grey-level values within a certain range, so that it is good enough to improve the image contrast and brightness. The first task of image normalization implemented and adopted for the research is the division of the segmented image into blocks of size S x S. Then for each pixel, grey-level value is compared with the average greylevel value for the host block. The result of the comparison produced a normalized grey-level value N(i,j), for a pixel I(i,j) belonging to a block having average grey-level value of M.

4.2.2.3 ORIENTATION ESTIMATION

Orientation estimation is the first of the prerequisites for fingerprint image filtering. The ridges form patterns in each image flow in different directions. The orientation of a ridge at location x,y is the direction of its flow over a range of pixels as shown in Fig. 4.7[2]. The Least Square Mean (LSM) fingerprint ridge orientation estimation algorithm proposed and implemented was slightly modified and used in this research.

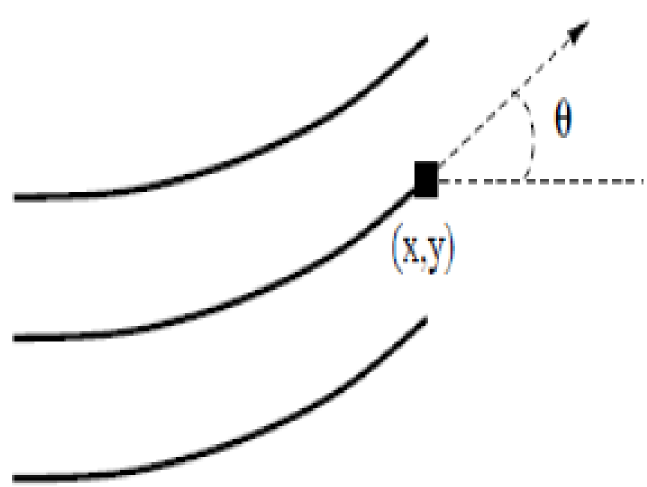


Fig. 4.7 The orientation of a ridge pixel in a fingerprint

The algorithm has the following steps:

STEP 1: Firstly, blocks of size S×S were formed from the normalized fingerprint image.

STEP 2: The gradients (p, q) were computed for each pixel, (p,q) in each block, as the gradient magnitudes in the x and y directions, respectively.

STEP 3: By using its S×S neighbourhood, the local orientation of a pixel was computed for a fingerprint image.

STEP 4: This is slightly modified by dividing the image into S×S blocks and then computing the local orientation of each block centered at the pixel I(i , j) .

STEP 5: Then this orientation image is converted into a continuous vector field.

STEP 6: Then the Gaussian smoothing is performed on a vector field.

STEP 7: The orientation field O of the block centered at pixel (i,j) is finally smoothed.

4.2.2.4 RIDGE FREQUENCY ESTIMATION

For fingerprint image filtering, the second prerequisite is the ridge frequency estimation. In any fingerprint image, the ridge frequency image is the sum total of the local frequency of the ridges. The ridge frequency is obtained from the extraction of the ridge map from the image. This extraction involves the following steps:

- The consistency of the orientation field obtained from the first prerequisite in the local neighbourhood of a pixel (p,q) is computed.
- If the consistency level is below a previously fixed threshold Fc, then until the consistency is above Fc, the local orientations in this region are re-estimated at a lower image resolution level. After the orientation field is obtained, two adaptive filters are applied to the image. The two filters are capable of stressing under different condition the local maximum grey level values along the normal direction of the local ridge orientation.

4.2.2.5 GABOR FILTERING

After obtaining the prerequisites, Gabor filtering is used to improve or enhance the fingerprint image to a finer structure. It involves the removal of noise and artifacts. a Gabor filter, named after Dennis Gabor, is a linear filter used for texture analysis, which means that it basically analyzes whether there are any specific frequency content in the image in specific directions in a localized region around the point or region of analysis. Frequency and orientation representations of Gabor filters are claimed by many contemporary vision scientists to be similar to those of the human visual system, though there is no empirical evidence and no functional rationale to support the idea. They have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave.

4.2.2.6 IMAGE BINARIZATION/THINNING

The image after Gabor filtering stage is binarized and thinned to make it more suitable for feature extraction. The Method sets the threshold (T) for making each and every cluster in the image as tight as possible, thereby minimizing their overlap. Here T is taken as zero(0). To determine the actual value of T, the following operations are performed on set of presumed threshold values:

a) The pixels are separated into two clusters according to the threshold.
b) The mean of each cluster are determined.
c) The difference between the means is squared.
d) The product of the number of pixels in one cluster and the number in the other is determined.

The success of these operations depends on the difference between the means of the clusters. The optimal threshold is the one that maximizes the between-class variance or, conversely, the one that minimizes the within-class variance. The within-class variance of each of the cluster is then calculated as the weighted sum of the variances.

5 COMPARISON CHART

This proposed system is compared with two other existing liveness detection techniques:-

- CNN
- LBP

And the proposed system provides about 98% accurate result.

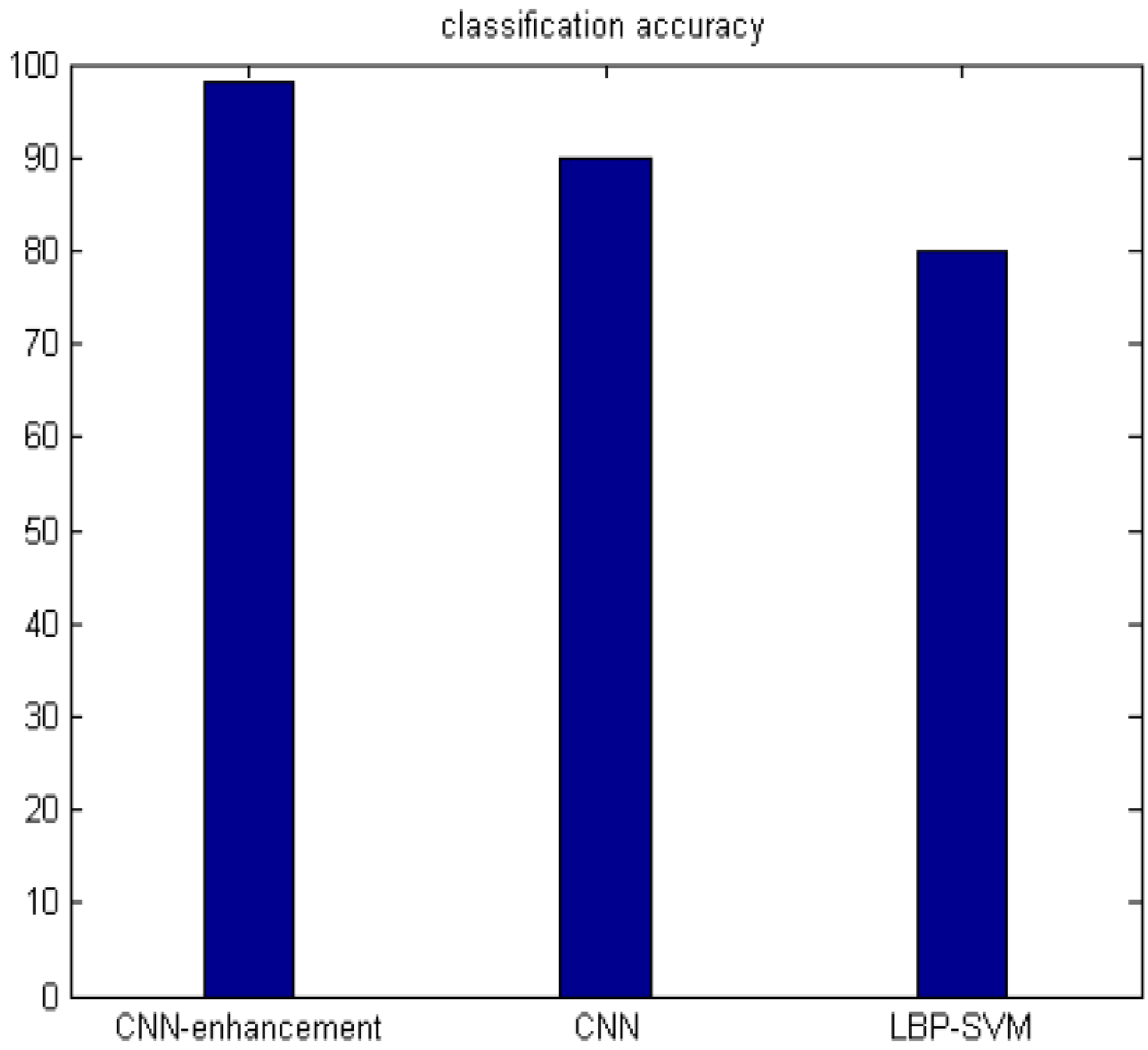


Fig 5.1 Comparison Chart

6 CONCLUSION

Convolution neural networks are one of the essential tools for deep learning, and are suitable for image recognition. We can construct a ConvNet architecture, train a network, and use that trained network to predict the class labels or numeric responses. We can also extract features from a pretrained network, and use such features to train a classifier. Neural Network Toolbox also enables us to perform a transfer learning; ie., retrain the last fully connected layer of an existing ConvNet on new data. Here such a Convolutional Neural Network is used for classifying live fingerprints from the fake ones. By this project an efficient fingerprint liveness detection technique is designed and implemented. Along with CNN, a fingerprint image enhancement technique is used in order to increase the accuracy of the system.

The results of the experiments conducted for image segmentation, normalization, ridge orientation estimation, ridge frequency estimation, Gabor filtering, binarization and thinning on synthetic and real fingerprint images reveal that with free or minimal noise level, the algorithms perform well. Improved performance is specifically noticed for the modified ridge orientation estimation algorithm. It is also established that each stage of the enhancement process is important for obtaining a perfectly enhanced image that is acceptable and presentable to the features extraction stage. The results obtained from the final stage of thinning show that the connectivity of the image ridge structure has been preserved and improved at each stage.

7 REFERENCES

[1] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado, “Fingerprint liveness detection using convolutional neural networks,” IEEE Transactions on Information Forensics and Security, vol. 11, no. 6, pp.1206–1213, 2016.

[2] Babatunde, Iwasokun Gabriel, et al. "Fingerprint image enhancement: Segmentation to thinning." (2012).

[3] Browne, Matthew, and Saeed Shiry Ghidary. "Convolutional neural networks for image processing: an application in robot vision." Australasian Joint Conference on Artificial Intelligence.Springer Berlin Heidelberg, 2003.

[4] Huang, Di, et al. "Local binary patterns and its application to facial image analysis: a survey." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 41.6 (2011): 765-781.

[5] Johnson, Peter, and Stephanie Schuckers. "Fingerprint pore characteristics for liveness detection." Biometrics Special Interest Group (BIOSIG), 2014 International Conference of the.IEEE, 2014.

[6] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado, “Evaluating software-based fingerprint liveness detection using convolutional networks and local binary patterns,” in Biometric Measurements and Systems for Security and Medical Appli-cations (BIOMS) Proceedings, 2014 IEEE Workshop on. IEEE, 2014, pp. 22–29.

[7] Lazimul Limnd T.P, Binoy D.L, “A Survey On Various Feature Selection Methods For Fingerprint Liveness Detection Techniques,” in II National Conference on Recent Trends in Computational Intelligence & Image Processing, RCIP 2017.

[8] Y. LeCun et al., “Generalization and network design strategies,” Connections in Perspective. North-Holland, Amsterdam, pp. 143–55, 1989.

[9] Y.-L. Boureau, J. Ponce, and Y. LeCun, “A theoretical analysis of feature pooling in visual recognition,” in Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 111–118.

[10] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," arXiv preprint arXiv:1301.3557, 2013.

[11] L. Hong, Y. Wan and A. Jain: 'Fingerprint image enhancement: Algorithm and performance evaluation'; Pattern Recognition and Image Processing Laboratory, Department of Computer Science, Michigan State University, 2006, pp1-30

[12] R. Thai: Fingerprint Image Enhancement and Minutiae Extraction, PhD Thesis Submitted to School of Computer Science and Software Engineering, 2003, University of Western Australia.