

Face Recognition and Biometric-(Fingerprint Detection)

Miss Snehal B. Jagtap
Reserch Scholar
Mrs.A.N.Mulla
Assistant Professor,
Dept of Computer Science and Engineering,
ADCET, Ashta, India
snehaljagtap.131990@gmail.com

Abstract-In this work, the *host-cpu* performs auxiliary operations for face detection and fingerprint recognition algorithms OpenCV library is used for initial processing of the image and trained cascaded Haar-classifier.

Viola-Jones algorithm is used to detect the frontal faces of the stored images. The process is carried out in different steps using Viola-Jones.

SIFT (Scale-Invariant Feature Transform) algorithm is used to detect whether two fingerprints are matched or not. The images are stored in the database. The keypoints are extracted and then these keypoints are matched within the given threshold.

The experiments are carried out to validate the results and scalability in terms of image size is shown

Keywords_ Face Detection, OpenCV, Fingerprint Detection, SIFT

I. INTRODUCTION

"Face Recognition" is a very active area in the Computer Vision and Biometrics fields, as it has been studied vigorously for 25 years and is finally producing applications in security, robotics, human-computer-interfaces, digital cameras, games and entertainment. The human face poses even more problems than other objects since the human face is a dynamic object that comes in many forms and colors [7]. Face detection in image sequence has been an active research area in the computer vision field in recent years due to its potential applications such as monitoring and surveillance human computer interfaces, smart rooms, intelligent robots, and biomedical image analysis. Face detection (Fig.1) is based on identifying and locating a human

face in images regardless of size, position, and condition. Numerous approaches have been proposed for face detection in images. Simple features such as color, motion, and texture are used for the face detection in early researches. However, these methods break down easily because of the complexity of the real world. Face detection proposed by Viola and Jones is most popular among the face detection approaches based on statistic methods. [8]

There are many different algorithms exist to perform face detection, each has its own weaknesses and strengths. Some use flesh tones, some use contours, and other are even more complex involving templates, neural networks, or filters. These algorithms suffer from the same problem; they are computationally expensive [2]. An image is only a collection of color and/or light intensity values. Analyzing these pixels for face detection is time consuming and difficult to accomplish because of the wide variations of shape and pigmentation within a human face. Pixels often require reanalysis for scaling and precision. Viola and Jones devised an algorithm, called Haar Classifiers, to rapidly detect any object, including human faces, using AdaBoost classifier cascades that are based on Haar-like features and not pixels [9].



Figure 1: Face detection Process

The second part of this objective is about Biometric that is fingerprint recognition which have been using for over a century. It can be used in forensic science to support criminal investigations, biometric systems such as civilian and commercial identification devices for person identification. It is one of the most significant biometric technologies which have drawn a substantial amount of attention recently [1, 3]. A fingerprint is comprised of ridges and valleys. The ridges are the dark area of the fingerprint and the valleys are the white area that exists between the ridges. The fingerprint of an individual is unique and remains unchanged of over a lifetime. The uniqueness of a fingerprint is exclusively determined by the local ridge characteristics and their relationships [1].

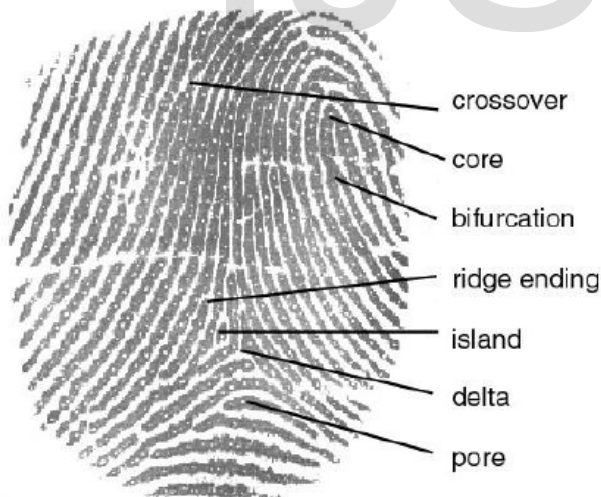


Figure 2: Fingerprint Structure

Figure 2 show that the structure of a fingerprint, it consists of crossover, core, bifurcation, ridge ending, island, delta and pore. The Fingerprint Recognition is a process of determining whether two sets of fingerprint ridge detail are from the same person. There are multiple approaches that are used in many different ways for fingerprint recognition which are minutiae,

correlation, and ridge pattern. These types of approaches can be broadly categorized as minutiae based or texture based.

Majority of these applications are time sensitive and needs continuous interactions with users, which enforces the application to perform at real time. However, CPU single thread implementation of face detection consumes lot of time, and despite various optimizations, it performs poorly at real time.

II. IMPLEMENTATION

A) Implementation of Face detection

There are many algorithms which implement the face-detection as a binary pattern classification. In this approach the content of a particular part of an image is transformed into features, after that a classifier which is trained on example faces decides whether that particular region of the image is a face. Often, a window-sliding technique is used. That is, the classifier is employed to classify the particular portions of an image, at all locations and scales, as either faces or non-face. Those images which have a plain or a static background are easy to process. In this implementation background is removed and only the faces will be left, assuming the image only consists of frontal face.

Face Detection using Viola-Jones

In face detection a photo is searched to find any face (surrounded by red rectangle), then image processing cleans up the facial image for easier recognition. The OpenCV library makes it fairly easy to detect a frontal face in an image using its Haar Cascade Face Detector (also known as the Viola-Jones method).

The Viola-Jones detector is a strong, binary classifier build of several weak detectors. Each weak detector is an extremely simple binary classifier. During the learning stage, a cascade of weak detectors is trained so as to gain the desired hit rate / miss rate using AdaBoost to detect objects, the original image is partitioned in several rectangular patches, each of which is submitted to the cascade. If a rectangular image patch passes through all of the cascade stages, then it is classified as "positive". The process is repeated at different scales until face is detected.

Algorithm (I): Viola Jones Algorithm

- Integral image for feature extraction
- Ada-Boost for feature selection
- Attentional cascade for fast rejection of non

face sub-windows

Implementation of Viola Jones Algorithm

The features used by the detection algorithm, involve the sums of image pixels within rectangular regions. As such, they bear some similarity to Haar basis functions, which were used earlier in the realm of image-based object detection. However, as the features used by Viola and Jones all depend on more than one rectangular region, they are usually more complex. The value of a feature is simply the sum of the pixels within clear rectangles subtracted from the sum of the pixels within dark shaded rectangles.

In order to train the different stages of the cascaded classifier the AdaBoost algorithm requires to be fed with positive examples – that is, images of faces. The facial images are collected manually and put into dataset and this dataset is used for implementation.

a) Integral Image

The first step of the Viola-Jones face detection algorithm is to take the input image from dataset and turn into an integral image. This is done by making each pixel equal to the entire sum of all pixels above and to the left of the concerned pixel. This allows for the calculation of the sum of all pixels inside any given rectangle using only four values. These values are the pixels in the integral image that coincide with the corners of the rectangle in the input image.

b) Feature Extraction

The Viola-Jones face detector analyzes a given sub-window using features consisting of two or more rectangles. The different types of features are shown in Figure

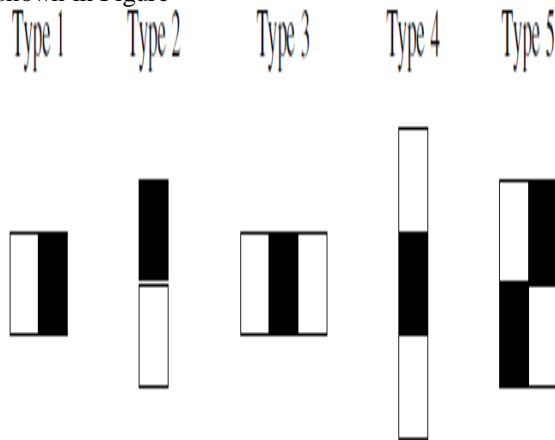


Figure 3: The different types of features

Each feature results in a single value which is calculated by subtracting the sum of the white rectangle(s) from the sum of the black rectangle(s).

c) AdaBoost Algorithm

An important part of the modified AdaBoost algorithm is the determination of the best feature, polarity and threshold. This means that the determination of each new weak classifier involves evaluating each feature on all the training examples in order to find the best performing feature. This is expected to be the most time consuming part of the training procedure. The steps for adaboost are

- Scan through the image, pick a window and rescale it to 24x24,
- Pass it to the strong classifier for detection.
- Report face, if the output is positive

d) The Cascaded Classifier

The cascaded classifier is composed of stages each containing a strong classifier. The job of each stage is to determine whether a given sub-window is definitely not a face or maybe a face. When a sub-window is classified to be a non-face by a given stage it is immediately discarded. Conversely a sub-window classified as a maybe-face is passed on to the next stage in the cascade. This sub window passed through each section of image and minimizes the false positive rate (i.e. non faces for example hand) and detects the face. This is repeated until the face found. In this method the false positive rate is very low as compare to others.

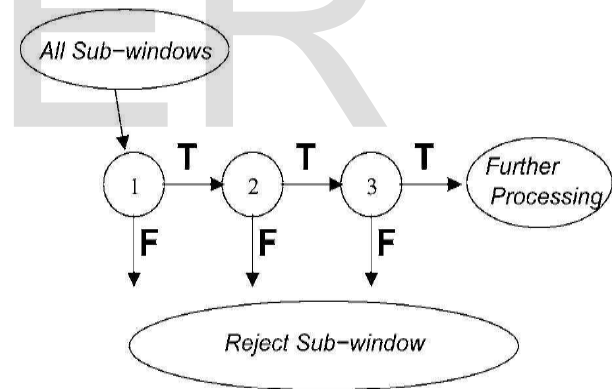


Figure 4: Flow of Cascade Classifier

After applying these steps the image is found consisting faces surrounded by rectangles and this is the face detection.

The height and width of image is calculated. The time required to detect the faces in facial images from dataset are also computed for finding time complexity of face detection using sequential algorithm.

B) Implementation of Biometrics-Fingered Print Recognition

Scale Invariant Feature Transform (SIFT) [11, 12, 13], and it's used for extending characteristic

feature points of fingerprint beyond minutiae points. SIFT approach has been adopting in other object recognition problems. It transforms an image into a collection of local feature vectors and each of these feature vectors is invariant to image translation, scaling and rotation. These features are extracted to perform matching in scale space through a staged filtering approach, and are robust to changes in illumination, noise, occlusion and minor changes in viewpoint. Also, they are highly distinctive and allowed for correct object recognition with low probability of mismatch and are easy to match against database of local features [6].

It is expected that in the domain of fingerprint recognition, this method is also stable and reliable, effective and efficient, and the features points are robust to the fingerprint quality and deformation variation.

Algorithm(II): Scale Invariant Feature Transform (SIFT)

There are following steps in SIFT algorithm are as follows:

1. Preprocessing: There are two steps to perform preprocessing, adjusting the graylevel distribution and removing noisy SIFT feature points.
2. Scale-space Extrema Detection : this stage identified interest key points that are invariant to scale and orientation in scale-space by using difference-of Gaussian (DoG) function
3. Keypoint Localization: In this stage, the final keypoints are selected and determined based on the stability of the candidate keypoints
4. Orientation Assignment: In this stage, the final keypoints are selected and determined based on the stability of the candidate keypoints. In order to perform a detailed fit to the nearby data for location, scale and ratio of curvatures, the candidate keypoints with low contrast or are poorly localized along an edge will be eliminated
5. Keypoint Descriptor: During the previous stages, a stable location, scale and orientation for each keypoint have been detected and determined. This stage measures the local image gradients at selected scale in the region around each keypoint, and computes a descriptor for the local image region that is highly distinctive
6. Keypoint Matching: The final stage is to compare the keypoints that we have detected from previous stages.

Implementation of SIFT Algorithm Preprocessing

This stage is to initialize the original input image. In order to obtain better matching performance, the input image has to be processed in two steps:

- i) Converting to grayscale,
- ii) Apply Gaussian-smoothing.

In this report, we convert the input image into 8-bit grayscale, and then convert the scale from 8-bit to 32-bit with single precision floatingpoint numbers. In other words, all pixel values of the input image will be 32-bit floating point numbers, as it's easy to compare and detect the keypoints which will be done in the next stage. Finally, we apply Gaussian smoothing to the 32-bit image in order to reduce noise.

a) Scale – space Extrema Detection

This stage is to detect keypoints by using a cascade filtering approach which adopts efficient algorithms to identify candidate keypoint locations that are determined and examined in further stage. The keypoints are detected by identifying the locations and scales that is repeatedly assigned under different views of the same object. The purpose of detecting locations is to be invariant to scale change of the image. It is done by searching stable features across all possible scales, and this is known as scale space. The scale space of an image is produced from the convolution of a variable-scale Gaussian with an input image. Therefore, this stage identified interest key points that are invariant to scale and orientation in scale-space by using difference-of Gaussian (DoG) function [5]. Firstly, generating Gaussian pyramid, that is, the input image is firstly applied Gaussian smoothing using B to give an image A. The value of B is then increased to create the second smoothed image B. The DoG images are generated by subtracting two nearby scales which are separated by a constant multiplicative factor k. In other words, the DoG is obtained by subtracting image B from A. After each octave, the Gaussian image is down sampled by a factor of 2 and the process is repeated until the entire DoG pyramid is built up. The number of scales in each octave is determined by a integer number, s, of intervals, so $k = 2^{1/s}$. Therefore, there are $s + 3$ images in the stack of blurred images for each octave. After the DoG pyramid has been produced, the local extrema is detected by comparing a pixel to its 26 neighbors in 3x3 regions at the current and adjacent scales. These extrema are selected as candidate keypoints which will be filtered in the next stage.

b) Local Extrema Detection

Once the DoG images have been produced across the entire pyramid, the local maxima and minima can be detected based on these DoG images. For each DoG image in each octave, it compares each sample point to its eight neighbors in the current image and nine neighbors in the scale above and below

(Figure 5). The sample point is only selected if its pixel value is less than or larger than all of its neighbors pixel value. Another thing has to be stated is that in each octave, the top and bottom DoG images are only used for comparing. In other words, the middle DoG images are used for keypoint selection. This process has to be applied across the entire pyramid until all extrema are selected as candidate keypoints.

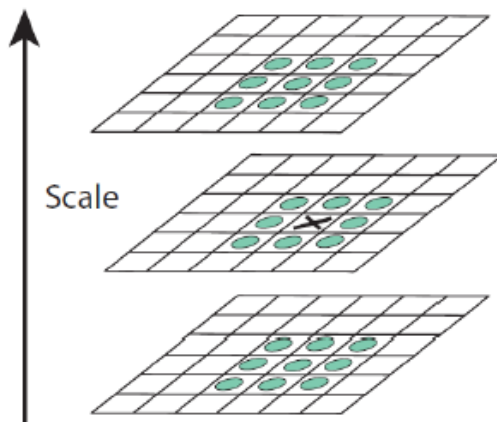


Figure 5: local extrema of DoG image are detected by comparing a pixel value (marked X) with its 26 neighbors (marked circle) in 3x3 regions from current image and adjacent scales

c) Keypoint Localization

This stage is to remove all unreliable keypoints. After all candidate keypoints has been selected during the previous stage, this stage performs a detailed fit to the nearby data for location, scale, and ratio of principal curvatures, that is, for each candidate keypoint, it will be eliminated if it has low contrast or is poorly localized along an edge.

d) Orientation Assignment

In order to achieve invariant to image rotation, each keypoint is assigned a consistent orientation based on local image properties. This approach contrasts with the approach from [10], in which each image property is based on a rotationally invariant measure. Therefore, this approach limits the descriptors that can be used and discards image information

d) Keypoint Descriptor

An image location, scale and orientation to each keypoint have been assigned during the previous operations. In this stage, a descriptor is computed for the local image region that is highly distinctive for each keypoint.

In order to achieve orientation invariance, the coordinates of descriptor and the gradient orientations are rotated relative to the keypoint orientation, and these gradients are performed during the previous stage. The image gradient magnitudes and orientations are sampled around keypoint location. These are illustrated with small arrows at each sample location on the left side of Figure 6. A Gaussian weighting function with B related to the scale of the keypoint, that is, one half the width of the descriptor window is used for assigning a weight to the magnitude of each sample point. A circular window on the left side of Figure 6 shows the Gaussian window. The right side of Figure 6 shows the keypoint descriptor. The orientation histogram is created over 4x4 sample region (shown on the left of the Figure 6), as it allows vital shift in gradient position. There are eight directions for each orientation histogram. The length of each arrow corresponds to the magnitude of that histogram entry. The same histogram will be produced if a sample gradient is shifted less than 4 sample positions.

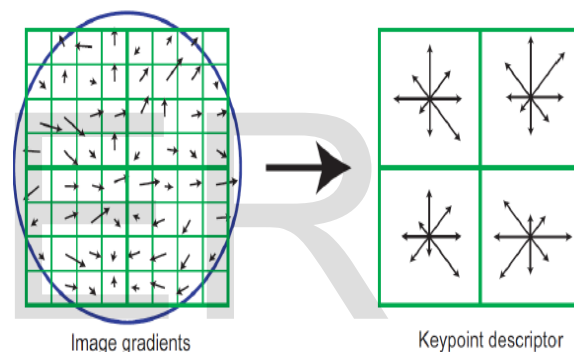


Figure 6: A keypoint descriptor is produced by computing the gradient magnitude and orientation at each sample point in a region around the keypoint location (shown on the left). The overlaid circle is a Gaussian window. The keypoint descriptor is shown on the right side.

Overall, the descriptor constitutes from a vector which contains the values of all the orientation histogram entries, which corresponds to the lengths of the arrows (shown on the right side of Figure 6).

e) Keypoint Matching

The best way to match each keypoint is to identify its nearest neighbor in the keypoints database. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector. However, due to the features arise from background clutter, there are many features from an image will not have correct match. Therefore, we have to discard some features that do not have any good match to the database. A better measurement is adopted by comparing the distance of the closest neighbor to that of the second closest neighbor. In

order to achieve reliable matching, correct matches need to have the closest neighbor significantly closer than the closest incorrect match which supports the performance of this measurement. The probability of matching is determined by the ratio of distance from the closest neighbor to the distance of the second closest. The ratio of distance is selected between 0.0 and 1.0.

III EXPERIMENTAL RESULTS

We evaluated our face detector under using a static image database. These are described in detail below. 42 frontal face images having 165 frontal faces were chosen and used for assessing the accuracy and speed of our face detector. We can notice that in many of the images, the faces are not well illuminated and some of the faces are not purely frontal. Due to these reasons, 100% detection accuracy is not achieved. As the dataset contains images at different sizes, we report the detection speed in terms of number of pixels processed per second. The results of our detectors on dataset are shown as

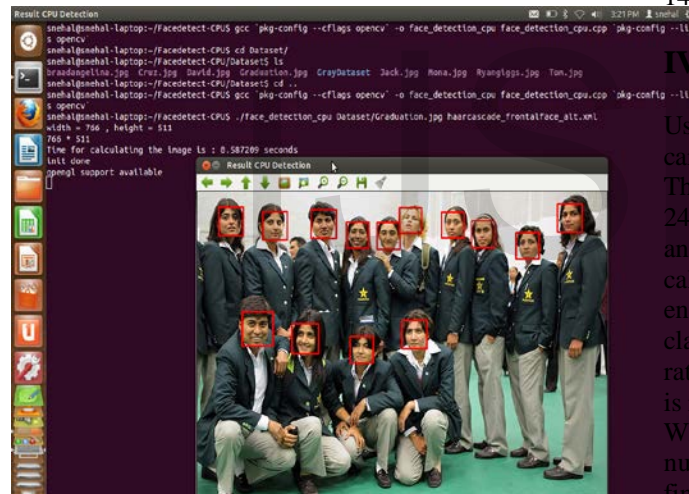


Figure 7: Face Detection for group image

The result shows the face detection of group image. The width=766 and height=511. The time taken to calculating the image is 0.587209 seconds

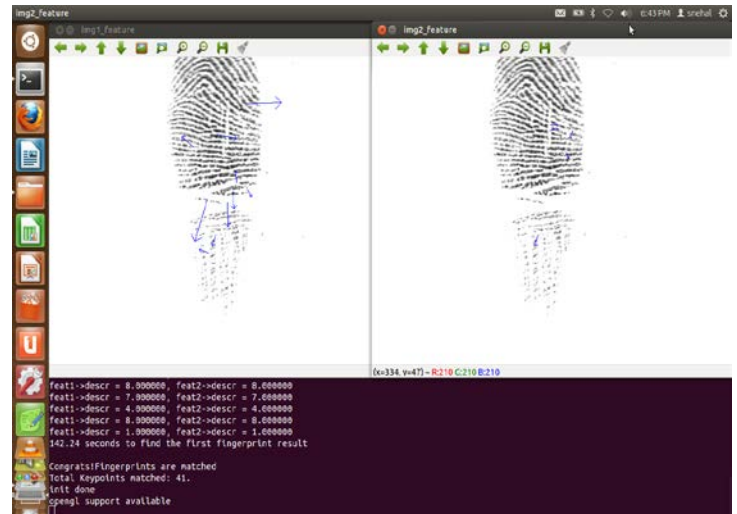


Figure 8: Fingerprint detection

This shows two images which are matched. Total keypoints matched are 41.

The time taken to calculate the fingerprint result is 142.24 seconds.

IV CONCLUSION

Face Detection is widely used application. Using Viola Jones algorithm face detection process is carried out. In that the Haar like features are extracted. The integral image is calculated using input image and 24 by 24 sub window is passed through out the image and resulted image is passed to the cascade classifier to calculate the false positives. But the result is not good enough so the images are passed through the AdaBoost classifiers. The AdaBoost minimizes the false positive rate and pass the image to next classifier. The process is repeated until Face is found.

When using SIFT in fingerprint recognition, the number of keypoints extracted based on the quality of a fingerprint image. Therefore, the image enhancement is a key step in preprocessing stage, as it can improve the fingerprint ridge structure and remove all noise around the boundary region which is different for every fingerprint impression even for the same finger. The preprocess is really significant for SIFT based object recognition, as it influences the keypoints detection which has direct association with the final matching result because the gradient and magnitude are calculated based on the keypoint location, particularly when we compare same fingerprint with different impression. Thus, The solid SIFT features we detect, the more accurate matching the SIFT approach will perform. In real time fingerprint recognition, SIFT algorithm should be applied with other approach according to the huge computation, which means it is better to use other approach such as Moment Invariant

to target all matched fingerprints from database, then applying SIFT algorithm to filter these matched fingerprints, and targeting the matched fingerprint by using SIFT eventually.

V FUTURE WORK

Almost all of these applications are very much time sensitive and needs continuous interactions with users, which needs the application to perform at real time. But with CPU single thread implementation of face detection time taken to detect the face is very high. With all the optimization, its performance is very bad in real time. One possible solution is to parallelize the detection algorithm. Parallelization with the help of CUDA enabled GPU's increases the performance of detection. With the arrival of General Purpose GPU (GPGPU) and growing support for various parallel programming languages like CUDA, OpenCL, we can easily use GPU for normal computational tasks.

One of the objectives is to implement a Viola-Jones based face detection system and Principal Component Analysis based face recognition system and fingerprint recognition system using GPUs of HPC GPU Cluster. It will attempt to analyze the strength of proposed system.

VI REFERENCES

- [1] H.C.Lee and R. E. Gaensslen. *Advance in Fingerprint Technology*. Elsevier, New York, 1991.
- [2] AT&T Laboratories Cambridge and Cambridge University Computer Laboratory. <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.htm>
- [3] E. Newham. *The Biometric Report*. SJB Services, New York, 1995. Face Detection and Face Recognition by Shervin Emami 2012
- [4] Junguk Cho, Shahnam Mirzaei, Jason Oberg and Ryan Kastner. *FPGA-Based Face Detection System Using Haar Classifiers*.
- [5] David B. Kirk., Wen-mei W. Hwu, *Programming Massively Parallel Processors – A Hands-on Approach*, Morgan Kaufmann, Elsevier (2010)
- [6] NVIDIA CUDA Compute Unified Device Architecture Programming Guide, V. 4.0,
 - a. NVIDIA “CUDA ZONE” <http://www.nvidia.com/page.home.html> (2011)
- [7] C. Schmid, and R. Mohr. Local gray value invariants for image retrieval. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 19(5): 530-534. 1997.
- [8] D. Lowe, “Object Recognition from Local Scale-Invariant Features,” *International Conference on Computer Vision*, September, 2004 *Machine Intelligence*, vol. 19, 302-314, 1997.
- [9] D. Lowe, “Distinctive image features from scale-invariant key points,” *International Journal of Computer Vision*, 60(2), 91-110, 1999.
- [10] Park, U.; Pankanti, S. & Jain, A.K. (2008). *Fingerprint Verification Using SIFT Features*. *Proceedings of SPIE Defense and Security Symposium*, 0277-786X, SPIE, Orlando, Florida