An Automated Object Tracking Model

Project Thesis

Submitted By

14-27908-3	Arefin, Mehedi Ibtesham
14-27673-3	Islam Md. Hasibul
15-28599-1	Hasan, Md. Naymur
13-23991-2	Huq, Nazmul

Department of Computer Science

Faculty of Science & IT

American International University Bangladesh

November, 2018

Declaration

We declare that this thesis is our original work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Arefin, Mehedi Ibtesham

14-27908-3

CSE

Islam Md. Hasibul

14-27673-3

CSE

Hasan, Md. Naymur

15-28599-1

CSE

Huq, Nazmul

13-23991-2

CS

Approval

The thesis titled "An Automated Object Tracking Model" has been submitted to the following respected members of the board of examiners of the department of computer science in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science on (31st October, 2018) and has been accepted as satisfactory.

Sabbir Ahmed
Assistant Professor & Supervisor
Department of Computer Science
American International University-Bangladesh

Md. Ezazul Islam
Assistant Professor & External
Department of Computer Science
American International University-Bangladesh

Dr. Mahbubul Syed
Associate Professor & Head
Department of Computer Science
American International University-Bangladesh

Professor Dr. Tafazzal Hossain
Dean
Faculty of Science & Information Technology
American International University-Bangladesh

Dr. Carmen Z. Lamagna
Vice Chancellor
American International University-Bangladesh

Table of Contents

List of Figures

Abstract

As the popularity of face recognition's huge applications successes are uprising, the demand of researching and funding behind it is increasing every second. A face detection and identification system tracks the subject's head and matches by comparing characteristics of the faces of those known individuals previously trained. This approach treats face recognition as two-dimensional recognition problem. The individuals' faces are first labeled and trained as datasets and thus new pictures are then identified as possibilities/percentage from all the trained datasets available. TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays.

Acknowledgement

First of all, we are thankful to Almighty Allah for the good health and comfort that were essential to complete this book. Our sincere thanks go to our honorable supervisor **Sabbir Ahmed** sir for providing us with all the necessary facilities and continuous support throughout the whole thesis work. We are also grateful to all of our valuable faculty members without whom we wouldn’t have come this far. Thanks to our parents for always being there throughout the whole undergraduate program and for the unceasing encouragement, support and attention.

We also place on record, our sense of appreciation to each and every person who have both directly and indirectly lent their hands in this venture.

Chapter 1

1. Introduction

We live in modern age where security is the most important thing. For security purpose different methods have come forward and have been applied. Now carrying ID card is not the trend for authorizing people. Things like magnetic card or chips, passwords and pins can be stolen or forgotten. For that something need to be develop which is reliable, trustworthy, fast and secure. [2] In business world different type of organizations has security system for authentication of identity of the individuals to get access to that particular organization like biometric technique. [1]

1.1 Biometric technique

A wide variety of systems requires reliable personal recognition schemes to either confirm or determine the identity of an individual requesting their services. The purpose of such schemes is to ensure that the rendered services are accessed only by a legitimate user and no one else. Examples of such applications include secure access to buildings, computer systems, laptops, cellular phones, and ATMs. A biometric system is essentially a pattern recognition system that operates by acquiring biometric data from an individual, extracting a feature set from the acquired data, and comparing this feature set against the template set in the database [22]. A biometric identification procedure is a technique with technological support for authorizations and identifications of personnel. Biometric techniques are created with biological characteristics of any human. The characteristics of a person cannot be stolen easily. That’s why biometric techniques are installed in small offices for security reason. Biometric authentication needs biometric devices [2] [22].

Biometric device uses person’s biological characteristics like fingerprints, face, iris pattern, hand geometry or retina image; or a behavioral characteristic such as voice, gait or signature for authorization or identification of a person. These characteristics are individual and cannot change or forgotten. These will be worked as password or pins. Biometric devices give a greater level of security for organizations [2].

1.1.1 Biometric works

As there are different type of biometric devices and their applications are different but the basic structure of working are same of all devices which are enrollment, storage and comparing.

In enrollment user will first time introduced to the biometric devices and device will take a certain characteristics of that user like fingerprint with fingerprint scanner or retina print with iris scanner or some other told previously. It should be checked that the print has taken perfectly. If the print is not clear and perfect, the authentication and recognition of that person will be harder and device may not authorize him [22].

In storage step the sample print which has taken in previous step will be saved in the database of biometric device by extracting individual characteristics from that sample and assigning individual username or id.

In comparing stage whenever an individual wants to gain access through biometric security system they have to verify their identities by again scanning their physical or biological traits and if they match to the ones which are already stored in its database they are granted access otherwise access is denied [2].

1.2 Types of recognition using biometric device

Fingerprint:

This is the most common authentication process which is used throughout the world. In which the fingerprint of a person is taken a sensor or fingerprint scanner as a sample for authentication.

Iris:

In this method the way of identification of a person is the images of the irises from his eyes which are also unique. A CCD digital camera is used to capture the images of the iris as like the center of the pupil, the edge of the pupil, the edge of the iris. The patterns are analyzed and convert them into a code. The chances of mismatching an image of a person with another person are almost none or negligible [2] but time consuming and uses in very high level security. In some cases fingerprint, iris and password authentication are used together in high security cases for authorizations.

Voice:

In this method voice scanner scans the sound of the voice and characteristics are differentiate on the base of vocal frequency and pattern.

Face Recognition:

This is the most recent and most usable technology which takes human face as input and analyzes the unique shape, pattern and positioning of facial features. Face reorganization is not often used because the cameras for face recognition are specially designed with the principle of artificial intelligence.

There are problem lies with face recognition is that people also change their style like beard, moustache etc. [2][20].

There are two methods for face recognition. One is video capturing and second is thermal imaging. Video capturing is used commonly.

Face recognition is the only biometric that can be used in two modalities—logon and continuous monitoring.

Logon modality is used as a perimeter defense mechanism, where an authorized individual gains entry to a network or session after a one-time logon process.

A continuous monitoring mode where persons are continuously authenticated for ensuring that at all times the individual in front of the computer or handheld device continues to be the same authorized person who logged in[21].

1.3 Thermal imaging:

Using thermal infrared imagery yields higher performance [3]. Identification using infrared thermal imaging is working as a new method and a recent phenomenon [8].

Thermal images are converted into digital representations by measuring the intensity of each pixel corresponding to the level of thermal energy for a corresponding portion of the image. An image containing the elemental shapes can be compared and then correlated with unique structural features of an individual [9].

For example, Rice, in U.S. Pat. No. 4,699,149, taught the scanning of subcutaneous blood vessels by measuring the inflection of incident radiation as a technique for determining a person's identification signature. Active heating of the area being scanned was required in order to get an accurate scan [4].

There are some issues faced with thermal imaging system like automatically positioning camera, biosensor, and identification accuracy enhancement using class sorting and identifying facial features from individuals wearing glasses [10].

For overcoming thus issues facial recognition proposed a way where that individual’s photo will be taken secretly at a certain position and after scaling the image will be matched to the database with an identification number whether the person is enrolled or not.

1.4 Video capturing:

Recognition with facial pattern and shape is the most recent and new effective method for security, authentication and identification. It has low error rate less than 1 percent and it is not intrusive [5].

For accessing the organization surveillance camera will be on the place where the face of the human should correctly capture in the camera perfectly. There will be identification number on accordance of that person face and information. The information will be matched in database that person authentication.

There are many ways in recognition of human face. In one way human face can be captured by the camera and match with existing database information. It takes a huge database and on other hand camera should be moving in a way where it can capture image. That means camera has to program in such a way that it always target the face where it can take better shot and match. The shots that camera take must be saved in the database which will take a huge amount of space.

On other hand detection can be done in surveillance cameras where the cameras can save both the shots and videos in database. In this case the cameras have artificial intelligence algorithm programmed. There are many system program which helps in automated detection and person authentication.

Jefry, in U.S. Pat. No. 5991429A shows a system of authentication with a moving camera which captures the face as it needs and match with enrolled data with an id number whether the man face is covered with glass or disguise.

Different companies have provided digital signatures from video images which work like fingerprint which is unique and relatively easy to look up in database against id.

There are much facial recognition software likes Visionics' FaceIt system. This system measures a face according to its peaks and valleys—such as the tip of the nose, the depth of the eye sockets—which are known as nodal points. A typical human face has 80 nodal points and precise recognition can be achieved with as few as 14 to 22 utilizing the FaceIt system. [5].

Visage Technology of Littleton, Mass., has a slightly different model. Its software compares faces to 128 archetypes it has on record. The Visage Technology has been

utilized to date in the identification of criminals, for access control, for transaction security and for identity fraud prevention.

In the past, law enforcement officials often have no more than a facial image to link a suspect to a particular crime or previous event. Database search with textual entries are all the things but by conducting facial image searching it has become more easy and possible for searching about a victim over the hundreds of databases within a minute and get all the information about that person [5].

In current facial recognition system algorithm is the main part which is mathematical techniques for encoding faces. In this case a system maps a face and creates face print and an id for that and stores it. After that it matches the face print with other faces in database.

David, in U.S. Pat. No. 7634662B2 proposed a system for associating facial recognition system in multimedia surveillance system using IP network. In this camera will view the scene, processor will analyze the video signals and then few works done by it like facial separation, facial signal generation and database creation, database lookup necessary alarm generating. This system can be work on IP cameras stored images captured by motion and also in live video. The system image server will have images which will be generated by camera motion. Video streams will be produced by motion camera. A facial database processor contains a stored database of the Facial Signatures and associated "mug shots" of some previously-defined persons. A facial processor will detect faces from camera's captured footage. Providing signatures and matching with previous. IP cameras will connected through LAN and WAN. Facial recognition will be integrated in IP camera network. All the stored video footage will be shown in same IP network. The amount of captured data in image database is huge.

Rowley's neural network based face detection (IEE page 23-38) shows a neural network based frontal face detection. In this system bootstrap algorithm has been used for negative. False detection adding was done. A retinal connected neural network examines small windows of an image and decides whether each window contains a face [6].

Laurenz on CAIP1997: Page: 456-463 shows the detection of human faces with a single image from a larger database which is indeed a difficult task because of image variations in terms of position, size, expression and pose. In this system Gabor Wavelets transform base had been followed in which faces are represented by labeled graphs. Recognition is based on similarity between graphed image and gallery of model graph. To handle the large gallery and increase the matching accuracy besides the algorithm, an object adapted graph called fiducial points and a new data structure which serves as a generalized representation of faces by combining jets of a small set of individual faces called bunch graph had been introduced. This eliminated the need of matching each model graph individually [7]. But this system does not meet the need of person detection from video footage.

In a hybrid neural-network for human face recognition which combines local image sampling, a self-organizing map (SOM) neural network, and a convolutional neural network.

Recognizing face in video there are multiple works. Uncontrolled videos of faces, the `YouTube Faces' database, along with benchmark, pair-matching tests [13].

A manifold approach is also proposed for face recognition from low quality videos. Super-resolution allows more accurate identification of individuals from low-resolution surveillance footage. Optical flow-based super-resolution method is benchmarked against

Baker et al.’s *hallucination* and Schultz et al.’s super-resolution techniques on images from the Terrascope and XM2VTS databases. Super-resolution system can improve the discriminability of surveillance video and enhance face recognition accuracy [18].

NEC's Neo Face v3.1 and Google-owned Pittsburgh Pattern Recognition v5.2.2 are two automated face recognition system for detecting Boston Marathon Bombings criminal. One million mug shots have been taken and searched by FBI in different video databases [16].

An efficient patch based face image quality assessment algorithm was proposed for video based face recognition which quantifies the similarity of a face image to a probabilistic face model, representing an `ideal' face. Image characteristics that affect recognition are taken into account; including variations in geometric alignment (shift, rotation and scale), sharpness and head pose and cast shadows. FERET and PIE datasets have shown that the proposed algorithm is able to identify images which are simultaneously the most frontal, aligned, sharp and well illuminated [18].

A system for long-term tracking of a human face in unconstrained videos is built on Tracking-Learning-Detection (TLD) approach. The off-line trained detector localizes frontal faces and the online trained validator decides which faces correspond to the tracked subject [19]. An image-based visual hull (IBVH) is computed from a set of monocular views and used to render virtual views for tracking and recognition. For Optimal face recognition, we place virtual cameras to capture frontal face appearance; for gait recognition we place virtual cameras to capture a side-view of the person. Multiple cameras can be rendered simultaneously, and camera position is dynamically updated as the person moves through the workspace [23].

There are also systems proposed by H. Hongo in which multiple camera tracks human face and hands with a standard color. For detecting face and hand gestures using linear discriminant analysis four directional features were proposed [24].

All the system has some expected result but all has faced some issues in recognition of face. Most of all show the way of capturing high quality video footage image. Live camera has been used, hand gestures have been analyzed and above all integrating the facial recognition system with the ip network has also done. But in most of the companies are using non intelligence camera which is not moving with motion. Surveillance cameras with ip connected stores the video footage. If we want to find a person we cannot find that automatically. This is a manual work process which is time consuming. We have to check from given time to time. There is no certain software on this that we

can find the person from previous stored footage using the database which will save time. However in existing IP cameras continuous footage without occurring network configuration change there is no detection system from stored video by running a process easily.

1.5 Tensor Flow

Tensor flow is an open-source software library for dataflow programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google, often replacing its closed-source predecessor, DistBelief. [46]

1.6 Brief of Tensor Flow

Tensor flow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on single devices, Tensor flow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). [46]Tensor flow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

Tensor flow computations are expressed as stateful dataflow graphs. The name Tensor flow derives from the operations that such neural networks perform on multidimensional data arrays. These arrays are referred to as "tensors". In June 2016, Jeff Dean stated that 1,500 repositories on GitHub mentioned Tensor flow, of which only 5 were from Google. [46]

1.7 Tensor Processing Unit

A **tensor processing unit** (**TPU**) is an AI accelerator application-specific integrated circuit (ASIC) developed by Google specifically for neural network machine learning. The chip has been specifically designed for Google's Tensor flow framework, Compared to a graphics processing unit, it is designed for a high volume of low precision computation (e.g. as little as 8-bit precision) with higher IOPS per watt, and lacks hardware for rasterization/texture mapping. The TPU ASICs are mounted in a heat sink assembly, which can fit in a hard drive slot within a data center rack, according to Google Distinguished Hardware Engineer Norman Jouppi. There are three generations of TPU. The last was released in May 8, 2018.

1.8 Hardware

Google says, “Each chip consists of two compute cores called Tensor Cores. A Tensor Core consists of scalar, vector and matrix units (MXU). In addition, 16 GB of on-chip memory (HBM) is associated with each Tensor Core.”

1.9 Software

Google says they have a TPU Estimator. You cannot download and use the GPU-enabled version of Tensor flow, which is different than regular Tensor flow in that it uses the CUDA SDK for that part of that code that is written in C and C++. There is no separate TPU-enabled version of Tensor flow. And unlike GPU, there appears to be no way to explicitly tell the code to use the TPU device, like in this code snippet that multiplies two matrices using GPU device. [46]

1.10 Tensor Flow works

In order to understand Tensor Flow one needs to understand Tensors and Graphs.

1.11 Tensors

Tensors are geometric objects that describe linear relations between geometric vectors, scalars, and other tensors. Elementary examples of such relations include the dot product, the cross product, and linear maps. Geometric vectors, often used in physics and engineering applications, and scalars themselves are also tensors.

1.12 Tensors in Tensor Flow

The big revelation is what NumPy lacks is creating Tensors. We can convert tensors to NumPy and viceversa. That is possible since the constructs are defined definitely as arrays/matrices.

1.13 Graphs

A computational graph is a series of TensorFlow operations arranged into a graph of nodes. Each node takes zero or more tensors as inputs and produces a tensor as an output. One type of node is a constant. Like all TensorFlow constants, it takes no inputs, and it outputs a value it stores internally.

1.14 Variables

Variables are in memory buffers containing tensors. They must be explicitly initialized and can be saved to disk during and after training. We can later restore saved values to exercise or analyze the model. Variable initializes must be run explicitly before other ops in your model can run. The easiest way to do that is to add an op that runs all the variable initializes, and run that op before using the model. We can conclude that placeholder is a way to define variables without actually defining the values to be passed to it when we create a computational graph.

tf.placeholder() is the norm, used by all the Tensor flow folks writing code daily.

For a more in depth reading: I/O for Tensor flow.

1.15 Working of Tensor Flow

When we create a tensor and declare it to be a variable, Tensor Flow creates several graph structures in our computation graph. It is also important to point out that just by creating a tensor; Tensor Flow is not adding anything to the computational graph. Tensor Flow does this only after creating available out of the tensor.

The main way to create a variable is by using the Variable () function, which takes a tensor as an input and outputs a variable. This is the declaration and we still need to

initialize the variable. Initializing is what puts the variable with the corresponding methods on the computational graph.

```
zero_tsr = tf.zeros([row_dim, col_dim])
```

Many algorithms depend on matrix operations. Tensor Flow gives us easy-to-use operations to perform such matrix calculations.

```
my_var = tf.Variable(tf.zeros([2,3]))

sess = tf.Session()

initialize_op = tf.global_variables_initializer ()

sess.run(initialize_op)
```

Besides the standard arithmetic operations, Tensor Flow provides us with more operations that we should be aware of. We need to know how to use them before proceeding. Again, we can create a graph session by running the following code:

```
import tensorflow as tf

sess = tf.Session()
```

When we start to use neural networks, we will use activation functions regularly because activation functions are a mandatory part of any neural network. The goal of the activation function is to adjust weight and bias. In Tensor Flow, activation functions are non-linear operations that act on tensors. They are functions that operate in a similar way to the previous mathematical operations. Activation functions serve many purposes, but a few main concepts is that they introduce a non-linearity into the graph while normalizing the outputs.

Chapter 2

2. Literature Review of Face Recognition Techniques

This chapter gives an overview on major face recognition techniques that are used on frontal faces. It also goes through the advantages and disadvantages of this techniques. The considered methods are Eigenfaces, Neural Networks, Dynamic Link Architecture, Hidden Markov model, Geometrical Feature Matching and Template Matching.

2.1 Eigen faces

In 1991, Turk and Pentland used PCA projection as the feature vectors to solve the face recognition problem. They used Euclidian distance as similarity function [18]. Later this system was called eigenface method which was the first eigenspace based face recognition system. After that many eigen-space based face recognition approach were proposed with different projection methods and similarity functions.

In mathematical term eigenfaces are the principal component of the distribution of faces or the eigenvectors of the covariance matrix of the set of face images. Reference [20] used eigenfaces, which was motivated by the technique of Kirby and Sirovich, for face detection and identification. The eigenvectors are ordered to represent different amounts of the variation, respectively, among the faces. Each face can be represented exactly by a linear combination of the eigenfaces. It can also be approximated using only the “best” eigenvectors with the largest eigenvalues. The best M eigenfaces construct an M dimensional space, i.e., the “face space”. The authors reported 96 percent, 85 percent, and 64 percent correct classifications averaged over lighting, orientation, and size variations, respectively. Their database contained 2,500 images of 16 individuals.

As the images include a large quantity of background area, the above results are influenced by background. The authors explained the robust performance of the system under different lighting conditions by significant correlation between images with changes in illumination. However, [21] showed that the correlation between images of the whole faces is not efficient for satisfactory recognition performance. Illumination normalization [19] is usually necessary for the eigenfaces approach.

Reference [22] proposed a new method to compute the covariance matrix using three images each was taken in different lighting conditions to account for arbitrary

illumination effects, if the object is Lambertian. Reference [23] extended their early work on eigenface to eigenfeatures corresponding to face components, such as eyes, nose, and mouth. They used a modular eigenspace which was composed of the above eigenfeatures (i.e., eigeneyes, eigennose, and eigenmouth). This method would be less sensitive to appearance changes than the standard eigenface method. The system achieved a recognition rate of 95 percent on the FERET database of 7,562 images of approximately 3,000 individuals. In summary, eigenface appears as a fast, simple, and practical method. However, in general, it does not provide invariance over changes in scale and lighting conditions.

2.2 Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. Neural networks-based approaches are learned from the example-images and rely on the techniques from machine learning to find the relevant characteristics of face images. The learned characteristics, in the form of discriminant functions (i.e. non-linear decision surfaces), are subsequently used for face recognition. Conventionally, face images are projected to a low-dimensional feature space and nonlinear decision surface is formed using multilayer neural networks for classifications and recognition [24]. Neural networks have also been used successfully for face recognition problem [25],[26],[24]. The advantage of using the neural networks for face recognition is that the networks can be trained to capture more knowledge about the variation of face patterns, and thereby achieving good generalization [27]. The main drawback of this technique is that the networks have to be extensively tuned to get exceptional performance. Among the neural networks approaches for face recognition, multilayer perceptron (MLP) with back propagation (BP) algorithm has been mostly used [25]. However, the convergence of the MLP networks is slow and the global minima of the error space may not be always achieved [25]. On the other hand, the RBF neural networks have fast learning ability [29] and best approximation property [13]. So, in recent times, many researches have used RBF networks for face recognition and show in figure 1 [31], [32], [27].

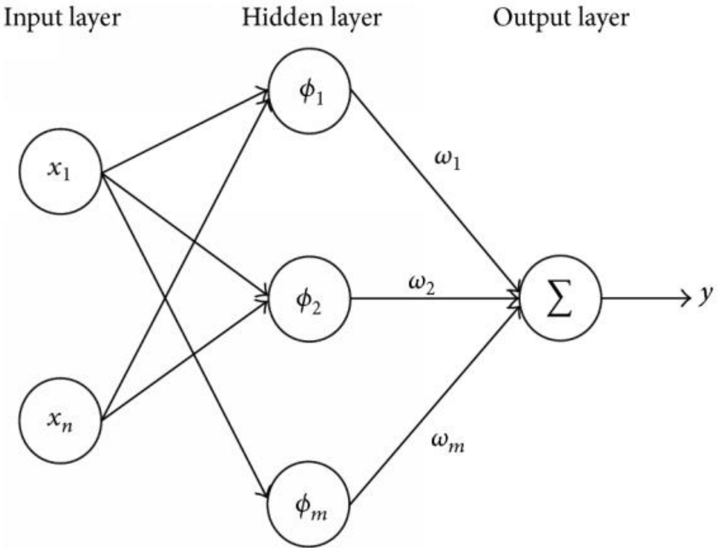


Figure 1 Structure of RBF neural network.

However, their success rates are not as promising as the error rates vary from 5 to 9% under variation of pose, orientation, scale and light [30]. This may be due to the fact that the selection of the centers of the hidden layer neurons might not have been done by capturing the knowledge about the distribution of training patterns and variations of face pose, orientation and lighting.

2.3 Graph Matching

Graph matching is another approach to face recognition. Reference [33] presented a dynamic link structure for distortion invariant object recognition which employed elastic graph matching to find the closest stored graph. Dynamic link architecture is an extension to classical artificial neural networks. Memorized objects are represented by sparse graphs, whose vertices are labeled with a multiresolution description in terms of a local power spectrum and whose edges are labeled with geometrical distance vectors. Object recognition can be formulated as elastic graph matching which is performed by stochastic optimization of a matching cost function. They reported good results on a database of 87 people and a small set of office items comprising different expressions with a rotation of 15 degrees.

The matching process is computationally expensive, taking about 25 seconds to compare with 87 stored objects on a parallel machine with 23 transputers. Reference [34] extended the technique and matched human faces against a gallery of 112 neutral frontal view faces. Probe images were distorted due to rotation in depth and changing facial expression. Encouraging results on faces with large rotation angles were obtained. They reported recognition rates of 86.5% and 66.4% for the matching tests of 111 faces of 15 degree rotation and 110 faces of 30 degree rotation to a gallery of 112 neutral frontal views. In general, dynamic link architecture is superior to other face recognition techniques in terms of rotation invariance; however, the matching process is computationally expensive.

2.4 Hidden Markov Model

Stochastic modeling of nonstationary vector time series based on (HMM) has been very successful for speech applications. Reference [35] applied this method to human face recognition. Faces were intuitively divided into regions such as the eyes, nose, mouth, etc., which can be associated with the states of a hidden Markov model. Since HMMs require a one-dimensional observation sequence and images are two-dimensional, the images should be converted into either 1D temporal sequences or 1D spatial sequences.

In [36], a spatial observation sequence was extracted from a face image by using a band sampling technique. Each face image was represented by a 1D vector series of pixel observation. Each observation vector is a block of L lines and there is an M lines overlap between successive observations. An unknown test image is first sampled to an observation sequence. Then, it is matched against every HMMs in the model face database (each HMM represents a different subject). The match with the highest likelihood is considered the best match and the relevant model reveals the identity of the test face.

The recognition rate of HMM approach is 87% using ORL database consisting of 400 images of 40 individuals. A pseudo 2D HMM [36] was reported to achieve a 95% recognition rate in their preliminary experiments. Its classification time and training time were not given (believed to be very expensive). The choice of parameters had been based on subjective intuition.

2.5 Geometrical Feature Matching

Geometrical feature matching techniques are based on the computation of a set of geometrical features from the picture of a face. The fact that face recognition is possible even at coarse resolution as low as 8x6 pixels [37] when the single facial features are hardly revealed in detail, implies that the overall geometrical configuration of the face features is sufficient for recognition. The overall configuration can be described by a vector representing the position and size of the main facial features, such as eyes and eyebrows, nose, mouth, and the shape of face outline.

One of the pioneering works on automated face recognition by using geometrical features was done by [38] in 1973. Their system achieved a peak performance of 75% recognition rate on a database of 20 people using two images per person, one as the model and the other as the test image. References [39, 40] showed that a face recognition program provided with features extracted manually could perform recognition apparently with satisfactory results. Reference [41] automatically extracted a set of geometrical features from the picture of a face, such as nose width and length, mouth position, and chin shape.

There were 35 features extracted form a 35 dimensional vector. The recognition was then performed with a Bayes classifier. They reported a recognition rate of 90% on a database of 47 people.

Reference [42] introduced a mixture-distance technique which achieved 95% recognition rate on a query database of 685 individuals. Each face was represented by 30 manually extracted distances. Reference [43] used Gabor wavelet decomposition to detect feature points for each face image which greatly reduced the storage requirement for the database. Typically, 35-45 feature points per face were generated. The matching process utilized the information presented in a topological graphic representation of the feature points. After compensating for different centroid location, two cost values, the topological cost, and similarity cost, were evaluated. The recognition accuracy in terms of the best match to the right person was 86% and 94% of the correct person's faces was in the top three candidate matches.

In summary, geometrical feature matching based on precisely measured distances between features may be most useful for finding possible matches in a large database such as a Mug shot album. However, it will be dependent on the accuracy of the feature location algorithms. Current automated face feature location algorithms do not provide a high degree of accuracy and require considerable computational time.

2.6 Template Matching

A simple version of template matching is that a test image represented as a two-dimensional array of intensity values is compared using a suitable metric, such as the Euclidean distance, with a single template representing the whole face. There are several other more sophisticated versions of template matching on face recognition. One can use more than one face template from different viewpoints to represent an individual's face.

A face from a single viewpoint can also be represented by a set of multiple distinctive smaller templates [41, 44]. The face image of gray levels may also be properly processed before matching [45]. In [45], Bruneli and Poggio automatically selected a set of four features templates, i.e., the eyes, nose, mouth, and the whole face, for all of the available faces. They compared the performance of their geometrical matching algorithm and template matching algorithm on the same database of faces which contains 188 images of 47 individuals. The template matching was superior in recognition (100 percent recognition rate) to geometrical matching (90 percent recognition rate) and was also simpler. Since the principal components (also known as eigenfaces or eigenfeatures) are linear combinations of the templates in the data basis, the technique cannot achieve better results than correlation [45], but it may be less computationally expensive.

One drawback of template matching is its computational complexity. Another problem lies in the description of these templates. Since the recognition system has to be tolerant to certain discrepancies between the template and the test image, this tolerance might average out the differences that make individual faces unique.

In general, template-based approaches compared to feature matching are a more logical approach. In summary, no existing technique is free from limitations. Further efforts are required to improve the performances of face recognition techniques, especially in the wide range of environments encountered in real world.

Chapter 3

3. Our Proposed System

This chapter proposes an intelligent surveillance system for human monitoring and visual surveillance to reduce human efforts. The proposed system collects video streams from the database captured through surveillance cameras, detects a target and produces an escape path of the target by analyzing adjacent cameras. After that this chapter includes comparisons between existing surveillance systems and our proposed system. Lastly it discusses the benefits and disadvantages.

1. Algorithm Flow Chart

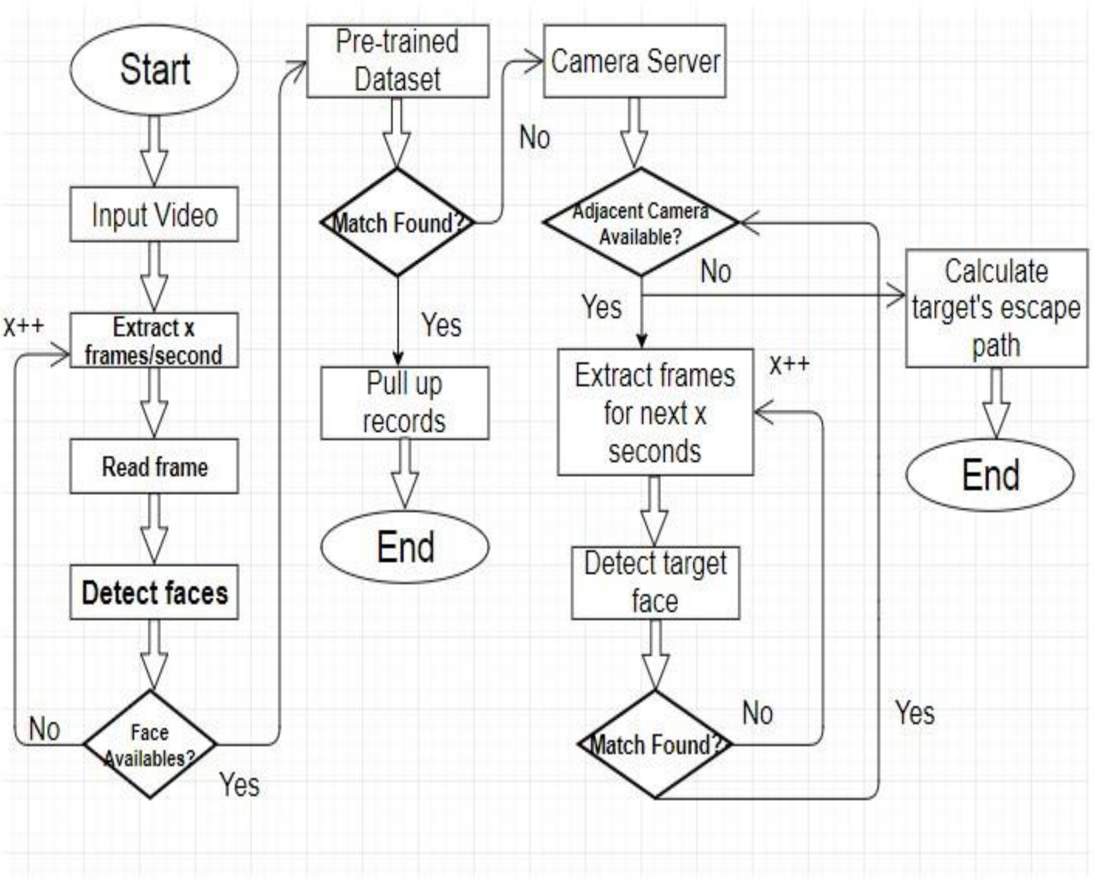


Figure 2 Flow chart of An Automated Object Tracking Model

The diagram in above describes our proposed system of an automated object tracking model. At first the video footage from the camera will be taken to our system as input. From the video footage every single images will be extracted to the software at x (where x is a user defined number) frames per seconds. After extracting, the images will be checked if there are any faces available. If no face is found in database then x is increased to check more accurately. If a face is found then the image will try to match the face with the pre-trained dataset from the tensor flow, if a match is found then the software will pull up the record of the target and his/her information. On the other hand, if, the system cannot find the image from the system, then the system will check from the camera server if there are any adjacent cameras. If the system finds any adjacent camera then the system will extract images from the video footage to the system for next x minutes (where x is a user defined number because the target would require some time to move from one camera to another camera, the time it took is considered x minutes). If the target of the respective image is found from the adjacent camera then it will check for its next adjacent camera, if not, x will be increased to check if the target has delayed to come to the viewport of next camera. If there is still no match found, then the system will check if there is any more adjacent camera left or not. If there is any adjacent camera then the process will continue until it finds no information of that target image. If the camera can’t find then the system will calculates the target’s escape path.

3.1 Data Flow Model

The proposed system primarily consists of camera, switch, time laps, video server, software unit and eventually users. The data flow from one component to another throughout the whole system. The data flow diagram is given below-

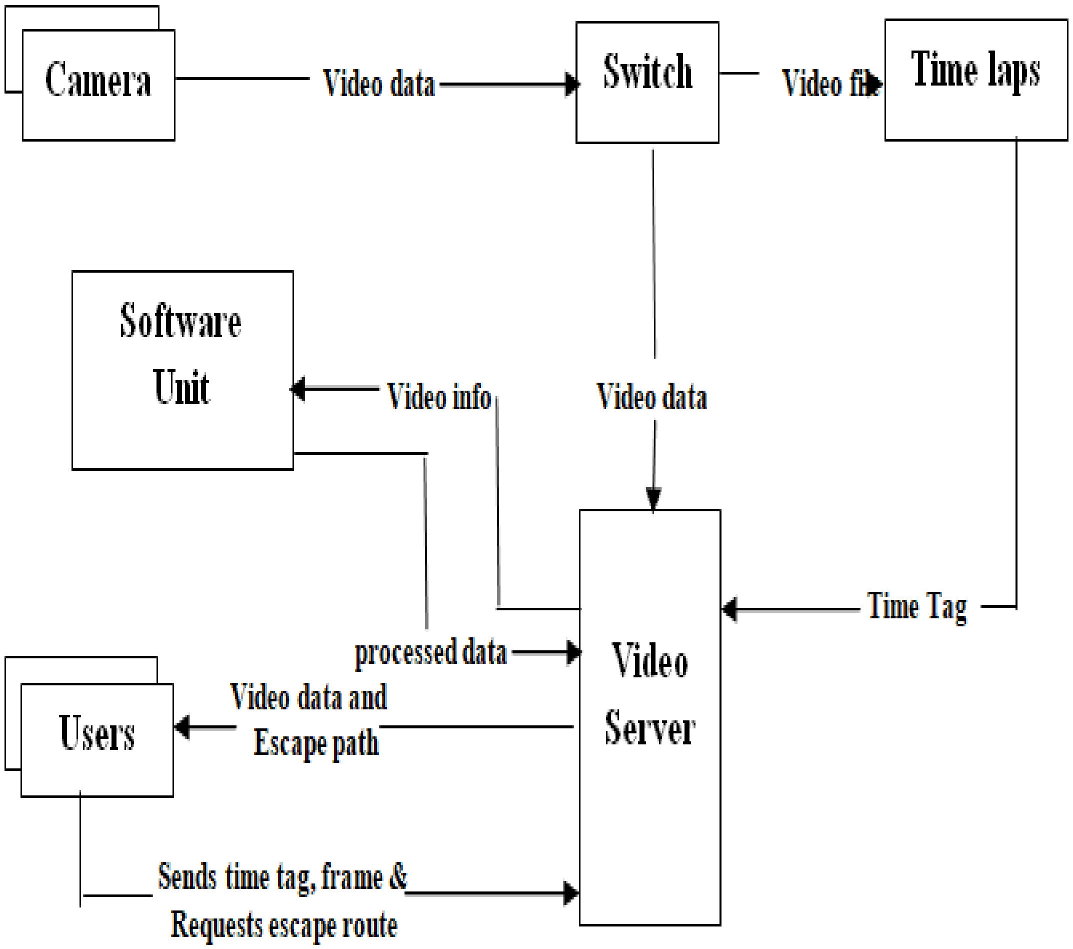


Figure 3 Data flow model of the proposed system.

Here we see that cameras shoot videos and send them to switch. After that switch sends the video files to time laps to extract the frames and collect time tags. At the same time switch also sends the video data to video server to store. On the other hand time laps send the time tag to video server to store them in respect to video frames. After that if a user sends requests to video server with time tag and frame the server sends them to the software unit. The software unit utilizes the video server data and detects a person or persons and process escape route and sends the video server. Eventually the video server returns the user pc fully processed video information and escape route.

3.2 System Architecture

This section gives an overview on the system architecture of the Object Detection Model Surveillance System.

It describes:

- A general description of the system
- The logical components of the layers and top-level components
- The physical Architecture of the Hardware on which runs the software

3.2.1 Architecture

The main goal of the proposed system is to detect and recognize a person from a specific time frame and present the person's trail.

The main purpose of the proposed system is for surveillance. Since it is specially designed to detect thief and show an escape route, it is suitable for any institution where many people comes and goes. Such as universities, hospitals, corporate offices even a street.

The system has mainly four layers

- **Networking layer**: Collects the videos through camera nodes.
- **Data layer:** Manages the system databases. Stores the video files from the networking layer and maintain timeframe.
- **Application processing layer:** Application specific functionality such as detection, recognition or producing escape route.
- **Presentation layer:** Presents the results of a computation to system users collecting user input

3.2.2 Physical Architecture Overview

As mentioned earlier the proposed system consists of four layers. Each layer contains some components.

- Networking layer consists of cameras at different paths and switches.
- Data layer is consists of time lapse and video server.
- Application processing layer is mainly the software that is installed in the video server
- Presentation layer is mainly the user interface (surveillance computers)

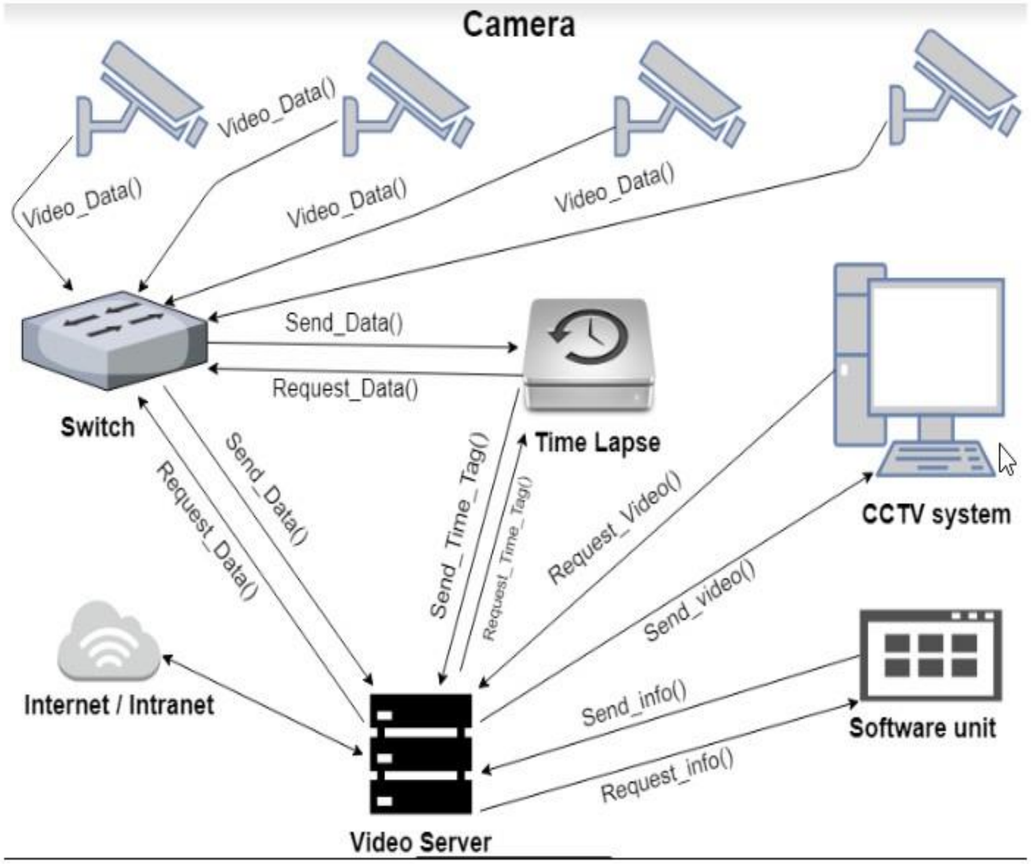


Figure 4 System Architecture of an Automated Object Tracking Model

3.2.2.1 Camera

In this proposed system close circuit cameras are used. These are the vital component through which the system collects the video data of the surveillance area. There should be many cameras at different points. The accuracy of escape route depends on the number of cameras. More accuracy is obtained with more cameras.

3.2.2.2 Switch

Cameras are connected through switches to send the video data to the video server and time lapse.

3.2.2.3 Time lapse

Time lapse is a server where each frame is extracted from the video files and sent to the video server with a time tag.

3.2.2.4 Video Server

The video server is the main server where database and the software is installed. Here the software fetch the specific data and produces the output result and send it to the user interface.

3.2.2.5 CCTV System

This system is mainly the user interface. This system may consist only one desktop/tablet/mobile or many devices connected through internet or intranet.

3.3 Comparison between Existing Systems and our Proposed System

Our proposed system has many similar functionalities as many existing surveillance systems have yet it is a unique surveillance system since it has many unique features. It is unique not only for its unique functionalities but also for many aspects such as cost effectiveness, installation difficulties, maintainability and algorithm accuracy.

Most of the existing detection surveillance systems only detects a person or detects motion. When our proposed system detects a person and recognize the person using machine learning. As a result our proposed system is more helpful and futuristic.

On the other hand our proposed system can detect from logically inter connected surveillance cameras and recognize the person from various camera through various angle which increases accuracy. While other existing systems uses only one camera to detect a person for a specific position.

The proposed surveillance system uses different cameras which are not connected physically but logically with a software which makes the system cheaper and easier to install. When more advanced surveillance systems uses physically connected cameras which are more costly and difficult to maintain.

Existing surveillance systems uses high-tech sensors and cameras to detect motion and persons which makes them costlier where our proposed system uses normal cheap surveillance cameras. And this approach makes our proposed system cheaper to install and maintain.

Our proposed surveillance system detects a person and recognize him/her and produce an escape route using several surveillance camera in different positions. And it is a unique feature of our proposed system.

Moreover our proposed system is more accurate in detecting and recognizing a person than many existing systems.

All the above scenario leads to a conclusion that our proposed system is cheaper to install and maintain but more accurate, useful and action specific. Furthermore, the use of more advanced artificial intelligence algorithms made it more futuristic than other existing systems.

3.4 Test result

Our proposed system took approximately 7 minutes to train about 100 pictures. The output time was 5 or 6 seconds for pre-trained dataset that contained about 100 pictures. Below is the system configuration that we used:

- Windows 10,

- 4th Generation Intel Core i7-4500U (1.8 GHz, 4MB L3 cache, 2 cores),

- Integrated: Intel HD Graphics 4400 (Core I processor),

- Discrete: AMD Radeon HD 8670M (1GB or 2GB DDR3, Switchable Graphics),

- Ram: 4GB DDR3L SDRAM (1600 MHz).

3.5 Issues faced

We had faced a lot of lacking of proper documentation about Tensor Flow. We had also faced difficulties of using Tensor Flow in windows operating system. Proper python script could not be found for our proposed system.

Chapter 4

4. Results

4.1 Python script

(To train dataset)

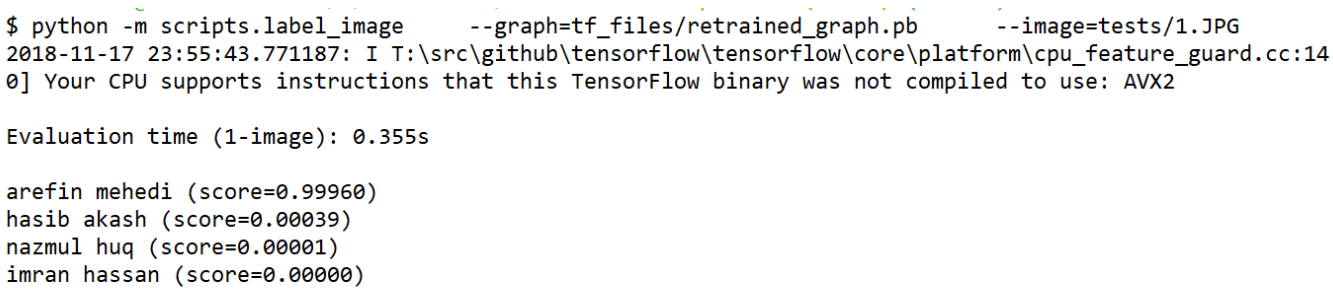
```
IMAGE_SIZE=224
ARCHITECTURE="mobilenet_0.50_${IMAGE_SIZE}"
    python -m scripts.retrain \
  --bottleneck_dir=tf_files/bottlenecks \
  --model_dir=tf_files/models/"${ARCHITECTURE}" \
  --summaries_dir=tf_files/training_summaries/"${ARCHITECTURE}" \
  --output_graph=tf_files/retrained_graph.pb \
  --output_labels=tf_files/retrained_labels.txt \
  --architecture="${ARCHITECTURE}" \
  --image_dir=tf_files/   YOUR_IMAGE_DIRECTORY_HERE
```

To test result

```
python -m scripts.label_image \
    --graph=tf_files/retrained_graph.pb  \
```

--image=**YOUR_PATH_TO_IMAGE_HERE**

Test result (screenshot)



```
$ python -m scripts.label_image     --graph=tf_files/retrained_graph.pb      --image=tests/1.JPG
2018-11-17 23:55:43.771187: I T:\src\github\tensorflow\tensorflow\core\platform\cpu_feature_guard.cc:14
0] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2

Evaluation time (1-image): 0.355s

arefin mehedi (score=0.99960)
hasib akash (score=0.00039)
nazmul huq (score=0.00001)
imran hassan (score=0.00000)
```

From the test result the –image=tests/1.jpg is the image of the random person and the result is shown for 4 person as the data set was trained for 4 person. If the data set had 100 person or more than that the result will show all the person's name with the probability from the descending order.

Chapter 4

5. Conclusion

Face detection has been and will be in future one of the most heated topics in terms of security. From CCTV cameras to latest smartphones, nowadays we cannot imagine security without being able to detect faces and differ person to person. But no system is entirely perfect and neither is ours. That is why we intend to improve our system in future. Currently our system is not so accurate when it comes to detect and identify faces from critical angles and the accuracy is also not up to the mark. These sectors need further research and study to properly identify faces with as much accuracy as possible. Also, when it comes to track an object from CCTV footage, posture detection becomes a basic requirement, which our system currently lacks. We have strong belief that by addressing these issues we can make our system actually implementable in various platforms.

References

[1] Jeffrey S. Coffin, Darryl Ingram, “Facial recognition system for security access and identification”, INFRARED INDENTIFICATION Inc, US (1)WO (1) , 1996.

[2] Information Security and Compliance (SOX, HIPPA, ISO-27001), On “Biometric Security System”, Project Report (PAPER-III).

[3] Diego A.Socolinsky, AndreaSelinger, Joshua D.Neuheisel, “Face recognition with visible and thermal infrared imagery”, Computer Vision and Image Understanding, Volume 91, Issues 1–2, July–August 2003, Pages :72-114

[4] David A. Monroe, Method for incorporating facial recognition technology in a multimedia surveillance system, e-Watch Inc, US (2), US7634662B2, 2002.

[5] H.A. Rowley ; S. Baluja ; T. Kanade, Neural network-based face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 20 , Issue: 1 , Jan 1998), **Pages:** 23 - 38

[6] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, Christoph von der Malsburg, “Face recognition by elastic bunch graph matching”, International Conference on Computer Analysis of Images and Patterns, Computer Analysis of Images and Patterns, volume 1296, pp 456-463, 1997

[7] Diego A.Socolinsky, AndreaSelinger, Joshua D.Neuheisel, “Face recognition with visible and thermal infrared imagery”, Computer Vision and Image Understanding,Volume 91, Issues 1–2, Pages 72-114, July–August 2003.

[8] D.A. Socolinsky, L.B. Wolff, J.D. Neuheisel , C.K. Eveland, “Illumination invariant face recognition using thermal infrared imagery”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, 15 April 2003.

[9] D.A. Socolinsky , A. Selinger, “A comparative analysis of face recognition performance with visible and thermal infrared imagery”, IEEE Explorer, Object recognition supported by user interaction for service robots, 2003.

[10] Lior Wolf , Tal Hassner , Itay Maoz, “Face recognition in unconstrained videos with matched background similarity”, CVPR 2011, IEEE, August 2011.

[11] Joshua C. Klontz ; Anil K. Jain, “A Case Study of Automated Face Recognition: The Boston Marathon Bombings Suspects” IEEE Computer ,Volume: 46 , Issue: 11 , Nov. 2013 , Pages**:** 91 – 94.

[12] Frank Lin, Clinton Fookes, Vinod Chandran, Sridha Sridharan, “Super-Resolved Faces for Improved Face Recognition from Surveillance Video” Lecture Notes in Computer Science book series, volume 4642.

[13] ErikHjelmåsn ,Boon KeeLow^{b} Face Detection: A Survey, Computer Vision and Image Understanding, Volume 83, Issue 3, September 2001, Pages 236-274

[14] Anil K. Jain, Arun Ross, and Salil Prabhakar, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 14, NO. 1, JANUARY 2004, PAGES: 4-20;

[15] Rabia Jafri* and Hamid R. Arabnia, “A Survey of Face Recognition Techniques”, Journal of Information Processing Systems, Vol.5, No.2, June 2009

[16] G. Shakhnarovich ; L. Lee ; T. Darrell, “Integrated face and gait recognition from multiple views”, IEEE, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001,15 April 2003.

[17] H. Hongo ; M. Ohya ; M. Yasumoto ; Y. Niwa ; K. Yamamoto, “Focus of attention for face and hand gesture recognition using multiple cameras” Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), 06 August 2002.

[18] Turk M. and Pentland A., Eigenfaces for Recognition, J. Cognitive Neuroscience,3(1), 1991, 71-86.

[19] M. Kirby and L. Sirovich, “Application of the Karhunen- Loève procedure for the characterisation of human faces,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, pp. 831-835, Dec. 1990.

[20] M. Turk and A. Pentland, “Eigenfaces for recognition,” J. Cognitive Neuroscience, vol. 3, pp. 71-86, 1991.

[21] M.A. Grudin, “A compact multi-level model for the recognition of facial images,” Ph.D. thesis, Liverpool John Moores Univ., 1997.

[22] L. Zhao and Y.H. Yang, “Theoretical analysis of illumination in pcabased vision systems,” Pattern Recognition, vol. 32, pp. 547-564, 1999.

[23] A. Pentland, B. Moghaddam, and T. Starner, “View-Based and modular eigenspaces for face recognition,” Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, pp. 84-91, 1994.

[24] Rowley H., Baluja S., and Kanade T., Neural network-based face detection, IEEE Trans. Pattern Anal. Mach. Intell, 20, 1998, 23-38.

[25] Er M.J., Wu S., Lu J., Toh H.L., Face recognition with radial basis function (RBF) neural networks, IEEE Trans. Neural Net,13, 2002, 697-710.

[26] Osuna E., Freund R., Girosi F., Training support vector machines: an application to face detection, in: Proceedings of the IEEE Conference Computer Vision and Pattern Recognition,1997, 130-136.

[27] Yang F., Paindovoine M., Implementation of an RBF neural network on embedded systems: real-time face tracking and identity verification, IEEE Trans. Neural Network, (14), 2003, 1162-1175.

[28] Valentin D., Abdi H., O’Toole A.J., and Cottrell G.W., Connectionist models for face processing: a survey, Pattern Recognition, 27, (1994), 1209–1230.

[29] Moody J., and Darken C.J., Fast learning in network of locallytuned processing units, Neural Computing, vol 1, page 281–294, 1989.

[30] Girosi F., and Poggio T., Networks and the best approximation property, Biol.Cybern, (63), 1990, 169-176.

[31] Howell, H. Buxton, Learning identity with radial basis function networks, Neurocomputing, l 20, 1998, 15-34.

[32] Ranganath S., and Arun K., Face recognition using transform features and neural networks, Pattern Recognition, l 30, 1997, 1615-1622.
[33] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. Von Der Malsburg, R.P. Wurtz, and M. Konen, “Distortion Invariant object recognition in the dynamic link architecture,” IEEE Trans. Computers, vol. 42, pp. 300-311, 1993.

[34] L. Wiskott and C. von der Malsburg, “Recognizing faces by dynamic link matching,” Neuroimage, vol. 4, pp. 514-518, 1996.

[35] F. Samaria and F. Fallside, “Face identification and feature extraction using hidden markov models,” Image Processing: Theory and Application, G. Vernazza, ed., Elsevier, 1993.

[36] F. Samaria and A.C. Harter, “Parameterisation of a stochastic model for human face identification,” Proc. Second IEEE Workshop Applications of Computer Vision, 1994

[37] S. Tamura, H. Kawa, and H. Mitsumoto, “Male/Female identification from 8_6 very low resolution face images by neural network,” Pattern Recognition, vol. 29, pp. 331-335, 1996.

[38] T. Kanade, “Picture processing by computer complex and recognition of human faces,” technical report, Dept. Information Science, Kyoto Univ., 1973.

[39] A.J. Goldstein, L.D. Harmon, and A.B. Lesk, “Identification of human faces,” Proc. IEEE, vol. 59, pp. 748, 1971.

[40] Y. Kaya and K. Kobayashi, “A basic study on human face recognition,” Frontiers of Pattern Recognition, S. Watanabe, ed., pp. 265, 1972.

[41] R. Bruneli and T. Poggio, “Face recognition: features versus templates,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, pp. 1042-1052, 1993.

[42] I.J. Cox, J. Ghosn, and P.N. Yianios, “Feature-Based face recognition using mixture-distance,” Computer Vision and Pattern Recognition, 1996.

[43] B.S. Manjunath, R. Chellappa, and C. von der Malsburg, “A Feature based approach to face recognition,” Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, pp. 373-378, 1992.

[44] R.J. Baron, “Mechanism of human facial recognition,” Int'l J. Man Machine Studies, vol. 15, pp. 137-178, 1981.

[45] M. Bichsel, “Strategies of robust object recognition for identification of human faces,” Ph.D. thesis, Eidgenossischen Technischen Hochschule, Zurich, 1991.

[46] TensorFlow: A System for Large-Scale Machine Learning Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, Google Brain