SHRI G. S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE

FACE BUNCH GRAPH

Author:
AMAN GOYAL
SAMIDHA MASATKAR
SHYAM RATHORE

Supervisor:
MR. K. K. SHARMA

A PROJECT REPORT

submitted to ***Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal***

In partial fulfillment of the requirements

for the degree of Bachelor of Engineering in the

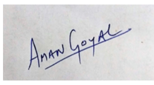
September 3, 2020

Declaration

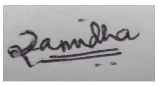
We, Aman Goyal, Samidha Masatkar, Shyam Rathore, declare that this project report titled, "Face Bunch Graph" and the work presented in it are our own. We confirm that:

- This work was done wholly while in candidature for a bachelor degree at this University.
- Where any part of this project has previously not been submitted for a degree or any other qualification at this University or any other institution.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this project is entirely our own work.
- We have acknowledged all main sources of help.

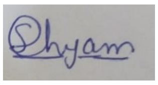
Signed:
Aman Goyal



Samidha Masatkar



Shyam Rathore



Date: 3rd September, 2020

Acknowledgements

We would like to express our sincere gratitude to all those who has given us the opportunity to accomplish this project. We would like to thank director of our institute Dr. R.K. Saxena and the Head of our Department Dr. Sunita Verma for giving us this opportunity to develop this project "Face Bunch Graph" and we would also like to thank our mentors Prof K. K. Sharma for guiding us and checking us throughout the development of this project. Special thanks to various faculty members for patiently answering our queries and providing us with suitable information and statistics.

Aman Goyal 0801IT161011
Samidha Masatkar 0801IT161066
Shyam Rathore 0801IT161082

Contents

List of Figures

Chapter 1

Introduction

1.1 Objective

To address image variation due to differences in facial expression, head pose, position, and size(to name only the most important). Our task is thus a typical discrimination in the presence of variance problem, where one has to try to collapse the variance and to emphasize discriminating features[2].

1.2 Scope

Our project mainly consists of three parts: face detection, face extraction and face recognition. The performance of the proposed method will be tested upon the standard face database namely databasename database. Extensive experiments are yet to be apply to demonstrate the effectiveness of method.

1.3 Problems in existing system

1. Illumination: a slight change in lighting conditions has always been known to cause a major impact on its results. If the illumination tends to vary, then;even if the same individual gets captured with the same sensor and with an almost identical facial expression and pose, the results that emerge may appear quite different.

2. Background: The placement of the subject also serves as a significant contributor to the limitations. A facial recognition system might not produce the same results outdoor compared to what it produces indoors because the factors- impacting its performance- change as soon as the locations change. Additional factors, such as individual expressions, aging etc. contribute significantly to these variations.

3. Occlusion: Occlusions of the face such as beard, moustache, accessories(goggles, caps, mask etc.) also meddle with the evaluation of a face recognition system. Presence of such components make the subject diverse and hence it becomes difficult for the system to operate in a non-simulated environment.

4. Complexity: Existing state-of-the-art methods of facial recognition rely on "too-deep" Convolution Neural Network(CNN) architecture which are very complex and unsuitable for real-time performance on embedded devices.[1][3]

1.4 Organization of the Report

Chapters

- Chapter 1: Introduction to the thesis topic
- Chapter 2: Literature Survey
- Chapter 3: Face Bunch Graph
- Chapter 4: System Analysis
- Chapter 5: System Design
- Chapter 6: Result and Conclusion

Chapter 2

Literature Survey

Face Recognition is such a challenging yet interesting problem that it has attracted many researchers from different backgrounds. It is due to this fact that the literature on face recognition is vast and diverse. The earliest work in face recognition can be traced back at least to the 1950's additionally; the research on automatic machine recognition of faces actually started in 1970's, but a fully automatic face recognition system based on neural network was reported back in 1997.[5]

The aim of all the researchers was to make face recognition as automated and accurate as possible through various types of inputs such as static images, video clips, etc.so as to increase its application in real world. Computational methods of face recognition need to address numerous challenges. These type of difficulties appear because faces are need to be represented in such a way that best utilizes the available face information to define a specific face from all the other faces in the database. Also, extracting such detailed facial features can be used in slandering search and enhancing recognition.
The problems of automatic face recognition involves three key steps:
1. face detection
2. feature extraction
3. face recognition

We present a system for recognizing human faces from single images out of a large database containing one image per person. The task is difficult because of image variation in terms of position, size, expression and pose. Our task is thus a typical discrimination-in-the-presence-of-variance problem, where one has to try to collapse the variance and to emphasize discriminating features.This is generally only possible with the help of information about the structure of variations to be expected.

In short, our system is based to a maximum on a general data structure- graphs labeled with wavelet responses- and general transformation properties. These are designer provided,but due to their generality and simplicity the necessary effort is minimal. Our system comes close to the natural model by needing only a small number of examples to handle the complex task of face recognition.

Chapter 3

Face Bunch Graph

All human faces share a similar topological structure. Faces are represented as graphs with nodes positioned at fiducial points(nose, eyes, ears,...) and edges labeled with 2-D distance vectors. Each node contains a set of 40 complex wavelet coefficients at different scales and orientations(phase and amplitude). They are called jets. Recognition is based on labeled graphs. A labeled graph is a set of nodes connected by edges, nodes are labeled with jets, and edges are labeled with distances.[4]

Three major extensions to this system in order to handle larger galleries and larger variations in pose, and to increase the matching accuracy, which provides the potential for further techniques to improve recognition rates.
(i) The phase of the complex wavelet coefficients to achieve a more accurate location of the nodes and to disambiguate patterns which would be similar in their coefficient magnitudes.
(ii) Employ object adapted graphs, so that nodes refer to specific facial landmarks, called fiducial points. The correct correspondence between two faces can then be found across large viewpoint changes.
(iii) Introduced a new data structure, called the bunch graph, which serves as a generalised representation of faces by combining jets of a small set of individual faces.
This allows the system to find the fiducial points in one matching process, which eliminates the need for matching each model graph individually. This reduces computational effort significantly.

The Face Bunch Graph Algorithm treats one vector per feature of the face. Feature of the face are the fiducial points. This has the advantage that changes in one feature (eyes open, closed) does not necessarily mean that the person is not recognised anymore. In addition this algorithm makes it possible to recognise faces up to a rotation of 22 degrees.[4][1]

Drawbacks of this algorithm are that it is very sensitive to lighting conditions and that a lot of graphs have to be placed manually on the face but with the make of features, being the output of band pass filters, and these are closely related to derivatives and are therefore less sensitive to lighting change.

3.1 Preprocessing with wavelets

The representation of local features is based on the wavelet transform. Wavelets are biologically motivated convolution kernels in the shape of plane waves restricted by a Gaussian envelope function. The set of convolution coefficients for kernels of different orientations and frequencies at one image pixel is called a jet.

3.1.1 Jets

A jet describes a small patch of grey values in an image I($\bar{x}$) around a given pixel $\bar{x}$=(x,y). It is based on a wavelet transform, defined as convolution

$$\mathcal{J}_j(\vec{x}) = \int \mathcal{I}(\vec{x}')\psi_j(\vec{x}-\vec{x}')d^2\vec{x}'$$

with a family of kernels

$$\psi_j(\vec{x}) = \frac{k_j^2}{\sigma^2}\exp\left(-\frac{k_j^2x^2}{2\sigma^2}\right)\left[\exp(i\vec{k}_j\vec{x}) - \exp\left(-\frac{\sigma^2}{2}\right)\right]$$

This is known as a wavelet transform because the family of kernels is self-similar, all kernels being generated from one mother wavelet by dilation and rotation.

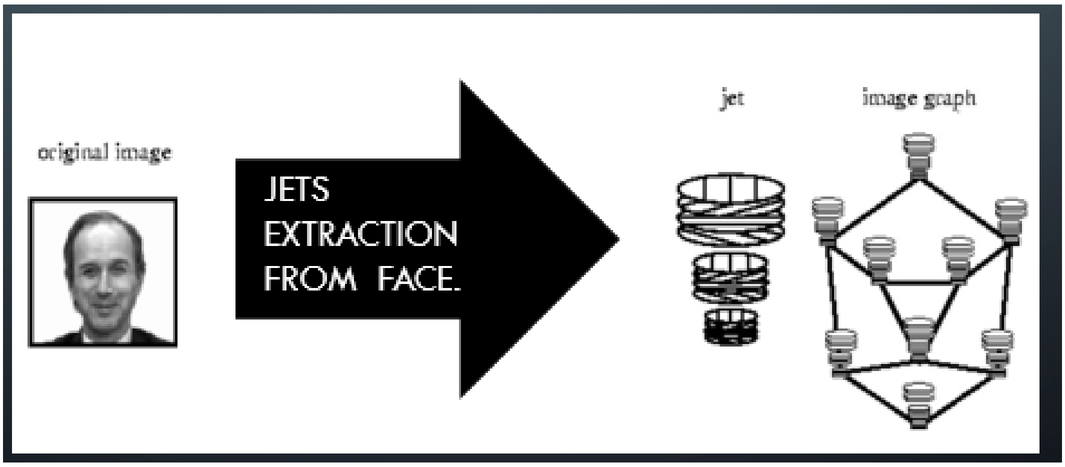


FIGURE 3.1: The graph representation of a face is based on the wavelet transform, a convolution with a set of wavelet kernels.

A jet J is defined as the set J_j of 40 complex coefficients obtained for one image point.

3.1.2 Comparing Jets

Due to phase rotation, jets taken from image points only a few pixels apart from each other have very different coefficients, although representing almost the same local feature. This can cause severe problems for matching. We therefore either ignore the phase or compensate for its variation explicitly. The similarity function

$$S_a(\mathcal{J},\mathcal{J}') = \frac{\sum_j a_j a'_j}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}},$$

Using phases has two potential advantages. Firstly, phase information is required to discriminate between patterns with similar magnitudes, should they occur, and secondly, since phase varies so quickly with location, it provides a means for accurate jet localization in an image. Assuming that two jets J and $J\prime$ refer to object locations with small relative displacement $\bar{d}$, the phase shift can be approximately compensated for by the terms $\bar{d}k_j$, leading to a phase-sensitive similarity function

$$S_\phi(\mathcal{J},\mathcal{J}') = \frac{\sum_j a_j a'_j \cos(\phi_j - \phi'_j - \vec{d}\,\vec{k}_j)}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}}.$$

To compute it, the displacement $\bar{d}$ has to be estimated. This can be done by maximizing S_θ in its Taylor expansion, as explained in the following section. It is actually a great advantage of this second similarity function that it yields this displacement information.

3.2 Face Representation

3.2.1 Individual Faces

For faces, we have defined a set of fiducial points, e.g. the pupils, the corners of the mouth, the tip of the nose, the top and bottom of the ears, etc. a labeled graph G representing a face consists of N nodes on these fiducial points at positions $\bar{x_n}$, n = 1,....,N and E edges between them. The nodes are labeled with jets J_n. The edges are labeled with distances $\triangle\bar{x_e}$=$\bar{x_n}$-$\bar{x_{n'}}$, e = 1,....,E, where edge connects node $n\prime$ with n .Hence the edge labels are two-dimensional vectors.

Graphs for different head pose differ in geometry and local features. Although the

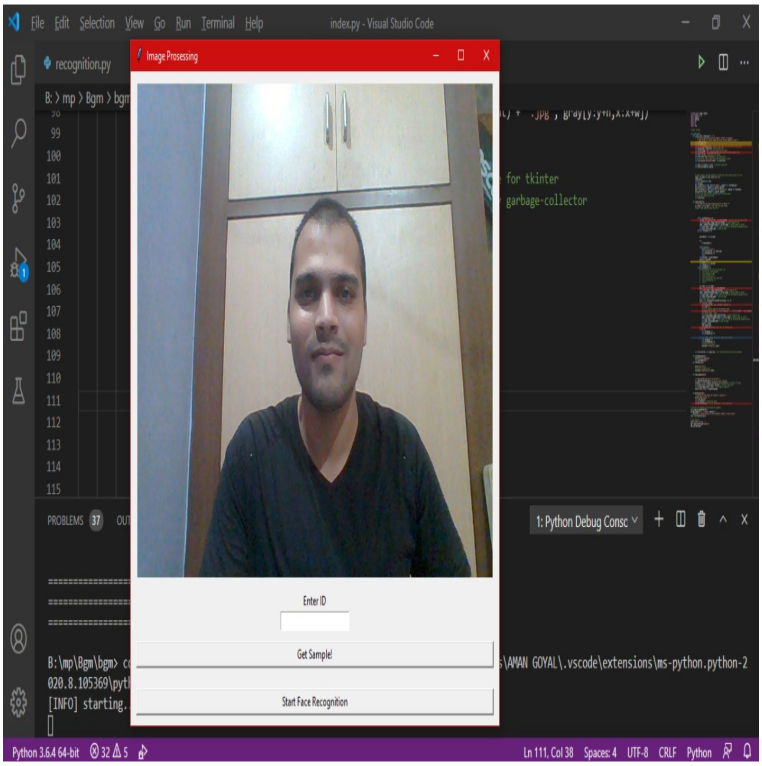


FIGURE 3.2: Image showing a individual face.

fiducial points refer to corresponding object locations, some may be occluded, and jets as well as distances vary due to rotation in depth. To be able to compare graphs for different poses, we have manually defined pointers to associate corresponding nodes in the different graphs.

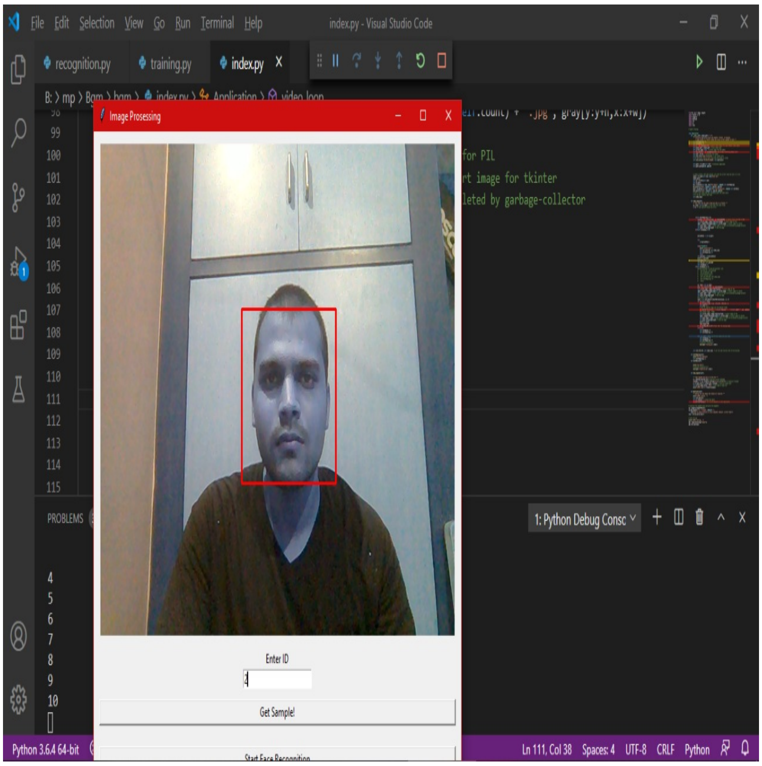


FIGURE 3.3: Image when fiducial points are identified.

3.2.2 Face Bunch Graphs

To find fiducial points in new faces, one needs a general representation rather than models of individual faces. This representation should cover a wide range of possible variations in the appearance of faces, such as differently shaped eyes, mouths, or noses, different types of beards, variations due to sex, age, race, etc. It is obvious that it would be too expensive to cover each feature combination by a separate graph. We instead combine a representative set of individual model graphs into a stack-like structure, called a face bunch graph(FBG). Each model has the same grid like structure and the nodes refer to identical fiducial points. A set of jets referring to one fiducial point is called a bunch. An eye bunch, for instance, may include jets from closed, open, female, and male eyes, etc., to cover these local variations.

How large should an FBG be and which models should be included depends first of all on the variability of faces one wants to represent. If the faces are of many different races, facial expression, age, etc., the FBG must contain any different models to cope with this variability. The required FBG size also increases with desired matching accuracy for finding the fiducial points in a new face. The accuracy can be estimated by matching the FBG to face images for which the fiducial points have been verified manually; FBG does not depends on gallery size. In general, the models in the FBG should be as different as possible to reduce redundancy and maximize variability. Here we used FBGs with 30 models fro the normalization stage and 70

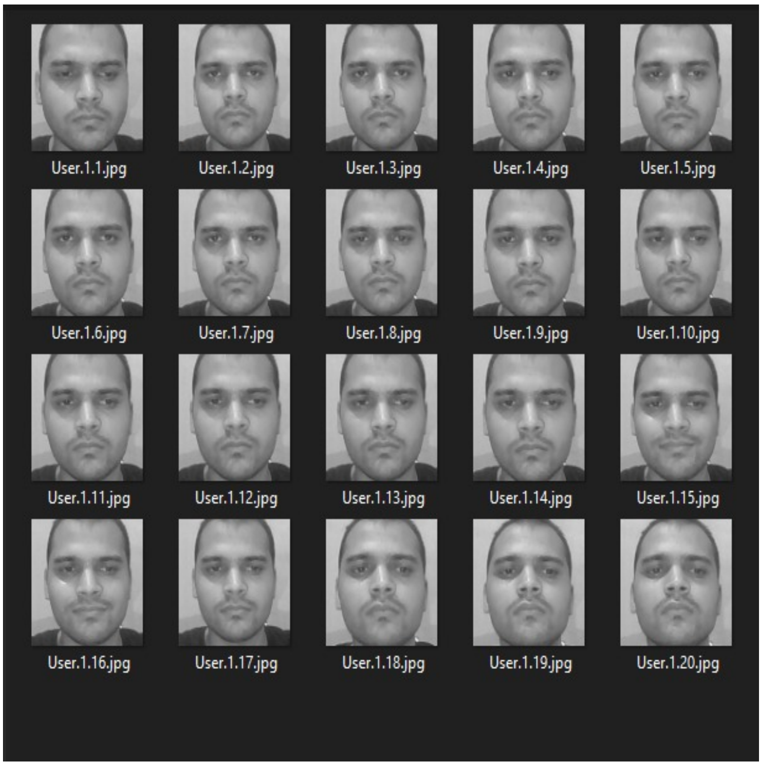


FIGURE 3.4: Sample faces from our database.

models for the final graph extraction stage. These sizes seemed to give sufficient accuracy and reliability. We selected the models arbitrarily and did not optimize for maximal variability.

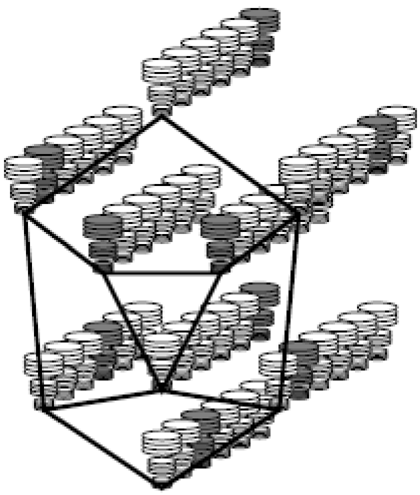


FIGURE 3.5: The Face Bunch Graph (FBG) serves as a representation of faces in general. It is designed to cover all possible variations in the appearance of faces. The FBG combines information from a number of face graphs. Its nodes are labelled with sets of jets, called bunches, and its edges are labeled with averages of distance vectors. During comparison to an image, the best fitting jet in each bunch, indicated by grey shading, is selected independently

3.3 Generating Face Representations By Face Bunch Graph algorithm

So far we have only described how individual faces and general knowledge about faces are represented by labelled graphs and the FBG, respectively. We are now going to explain how these graphs are generated. The simplest method is to do so manually. We have used this method to generate initial graphs for the system, one graph for each pose, together with pointers to indicate which pairs of nodes in graphs for different poses correspond to each other. Once the system has an FBG (possibly consisting of only one manually defined model), graphs for new images can be generated automatically by Face Bunch Graph Algorithm. Initially, when the FBG contains only a few faces, it is necessary to review and correct the resulting matches, but once the FBG is rich enough (approximately 70 graphs) one can rely on the matching and generate large galleries of model graphs automatically.

3.3.1 Manual Definition of Graph

Manual definition of graphs is done in three steps. First, we mark a set of fiducial points for a given image. Most of these are positioned at well-defined features which are easy to locate, such as left and right pupil, the corners of the mouth, the tip of the nose, the top and bottom of the ears, the top of the head, and the tip of the chin. These points were selected to make manual positioning easy and reliable. Additional fiducial points are positioned at the center of gravity of certain easy-to-locate fiducial points. This allows automatic selection of fiducial points in regions where well-defined features are missing, e.g. at the cheeks or the forehead. Then, edges are drawn between fiducial points and edge labels are automatically computed as the differences between node positions. Finally, the wavelet transform provides the jets for the nodes.

3.3.2 The Graph Similarity Function

A key role in Face Bunch Graph Algorithm is played by a function evaluating the graph similarity between an image graph and the FBG of identical pose. It depends on the jet similarities and the distortion of the image grid relative to the FBG grid. For an image graph G_I with nodes n= 1,...,N and edges e= 1,...,E and an FBG B with model graphs m= 1,...,M the similarity is defined as

$$S_B(\mathcal{G}^I, \mathcal{B}) = \frac{1}{N} \sum_n \max_m \left(S_\phi(\mathcal{J}_n^I, \mathcal{J}_n^{Bm}) \right) - \frac{\lambda}{E} \sum_e \frac{(\Delta \vec{x}_e^I - \Delta \vec{x}_e^B)^2}{(\Delta \vec{x}_e^B)^2},$$

3.3.3 Matching Procedure

The goal of Face Bunch Graph Algorithm on a probe image is to find the fiducial points and thus to extract from the image a graph which maximizes the similarity with the FBG. In practice, one has to apply a heuristic algorithm to come close to the optimum within a reasonable time. We use a coarse to fine approach in which we introduce the degrees of freedom of the FBG progressively: translation, scale, aspect ratio, and finally local distortions. We similarly introduce phase information and increase the focus of displacement estimation: no phase, phase with focus 1, and then phase with focus 1 up to 5. The matching schedule described here assumes faces of known pose and approximately standard size, so that only one FBG is required.

Step 1: Find approximate face position

Condense the FGB into an average graph by taking the average magnitudes of the jets in each bunch of the FBG (or, alternatively, select one arbitrary graph as a representative). Use this as a rigid model (=) and evaluate its similarity at each location of a square lattice with a spacing of 4 pixels. At this step the similarity function is without phase issued instead of S. Repeat the scanning around the best fitting position with a spacing of 1 pixel.The best fitting position finally serves as the starting point for the next step.

Step 2: Refine position and size.

Now the FBG is used without averaging, varying it in position and size.Check the four different positions (3, 3) pixels displaced from the position found in Step 1, and at each position checks two different sizes which have the same center position, a factor of 1:18 smaller or larger than the FBG average size. This is without effect on the metric similarity, since the vectors xBe are transformed accordingly. We still keep = . For each of these eight variations, the best fitting jet for each node is selected and its displacement is computed. This is done with a focus of 1, i.e., the displacements may be of a magnitude up to eight pixels. The grids are then rescaled and repositioned to minimize the square sum over the displacements. Keep the best of the eight variations as the starting point for the next step.

Step 3: Refine size and find aspect ratio.

A similar relaxation process as described for Step 2 is applied,but relaxing the x- and y-dimensions independently. In addition, the focus is increased successively from 1 to 5.

Step 4: Local distortion.

In a pseudo-random sequence the position of each individual image node is varied to further increase the similarity to the FBG. Now the metric similarity is taken into account by setting = 2 and using the vectors xBe as obtained in Step 3. In this step only those positions are considered for which the estimated displacement vector is small ($d < 1$, see Eq. (8)). For this local distortion the focus again increases from 1 to 5. The resulting graph is called the image graph and is stored as a representation of the individual face of the image.

3.3.4 Schedule of Graph extraction

To minimize computing effort and to optimize reliability, we extract a face representation in two stages, each of which uses a matching procedure as described in the

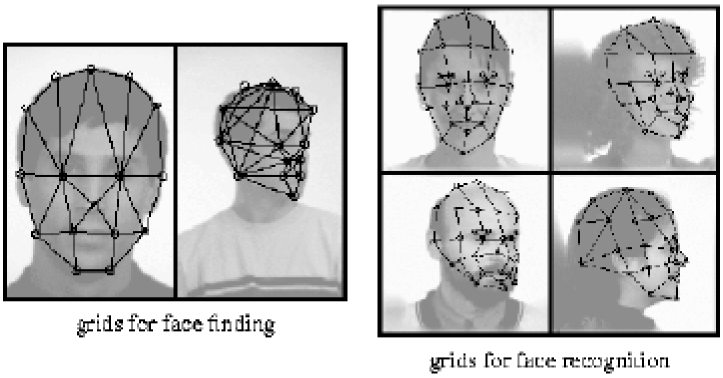


FIGURE 3.6: Object-adapted grids for different poses. One can see that, in general, the matching finds the fiducial points quite accurately. But mismatches occurred, for example, for bearded man. The chin was not found accurately; the leftmost node and the node below it should be at the top and the bottom of the ear respectively.

previous section. The first stage, called the normalization stage and has the purpose of estimating the position and size of the face in the

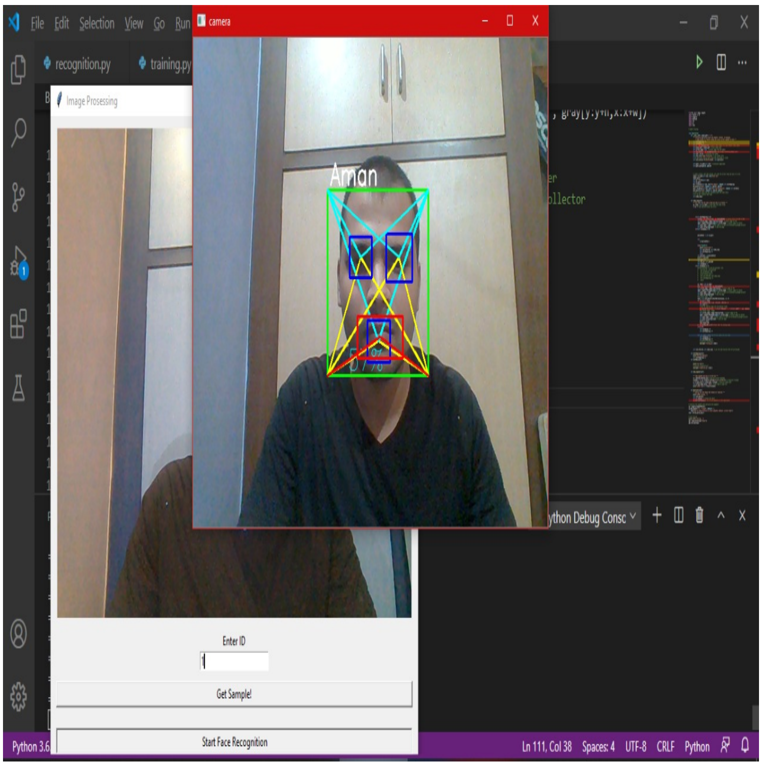


FIGURE 3.7: Grids to find the face.

original image, so that the image can be scaled and cut to standard size. The second stage takes this image as input and extracts a precise image graph appropriate for face recognition purposes. The two stages differ in emphasis. The first one has to deal with greater uncertainty about size and position of the head and has to optimize the reliability with which it finds the face, but there is no need to find fiducial points with any precision or extract data important for face recognition. The second stage can start with little uncertainty about position and size of the head, but has to extract a detailed face graph with high precision.

3.4 Recognition

After having extracted model graphs from the gallery images and image graphs from the probe images, recognition is possible with relatively little computational effort by comparing an image graph to all model graphs and picking the one with the highest similarity value. A comparison against a gallery of 250 individuals took slightly less than a second. The similarity function we use here for comparing graphs is an average over the similarities between pairs of corresponding jets. For image and model graphs referring to different pose, we compare jets according to the manually provided correspondences. If G^I is the image graph, G^M is the model graph, and node $n_{n'}$ in the model graph corresponds to node $n^{'}$ in the image graph, we define graph similarity as

$$\mathcal{S}_{\mathcal{G}}(\mathcal{G}^{\mathcal{I}}, \mathcal{G}^{\mathcal{M}}) = \frac{1}{N'} \sum_{n'} \mathcal{S}_a(\mathcal{J}^{\mathcal{I}}_{n'}, \mathcal{J}^{\mathcal{M}}_{n_{n'}}),$$

Chapter 4

System Analysis

4.1 Information Flow Representation

UML is a way of visualizing software programs using collection of diagrams. This chapter includes the required UML diagrams along with brief introduction to it.

4.1.1 Usecase Diagram

A use case specifies the behavior of a system or a part of a system, and is a description of a set of sequences of actions, including variants, that a system performs to yield an observable result of value to an actor.
Use case diagrams describe what a system does from the standpoint of an external observer.
In our use case diagram, we have two actors namely user and system admin. The use cases defined for user are:
• Request for matching
• Provide sample matching

The use cases defined for system admin are:
• Train model
• Detection and Recognition
• Delete Sample from Database
• Update Database

4.1.2 Sequence Diagram

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. It shows a set of objects and the messages sent and received by those objects. Graphically, a sequence diagram is a table that shows objects arranged along the X axis and messages, ordered in increasing time, along the Y axis. It describes an interaction by focusing on the sequence of messages that are exchanged, along with their corresponding occurrence specifications on the lifelines.

In our application, we have four objects - user, camera, database and model. The following sequence of actions takes place:
• The user first opens application.
• The device then helps the user in starting the camera.
• Then the device asks for uploading or scanning of the image.
• The user scans/uploads the image.
• Then the device gives that image to the model (which is a pre-trained model).

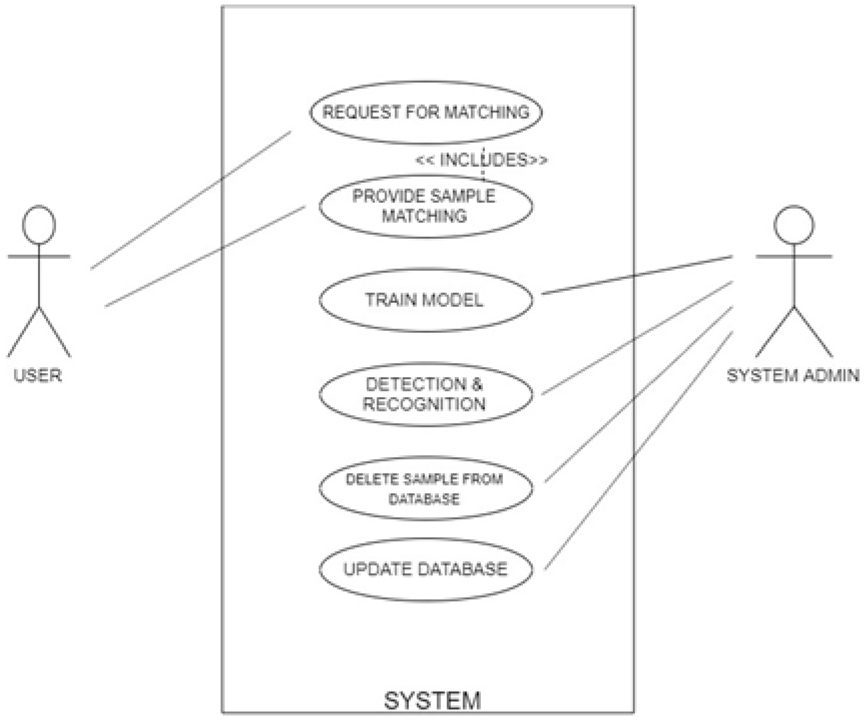


FIGURE 4.1: Usecase Diagram

- The model then processes the image and finally sends the results to the device.
- The device then shows the results to the user.

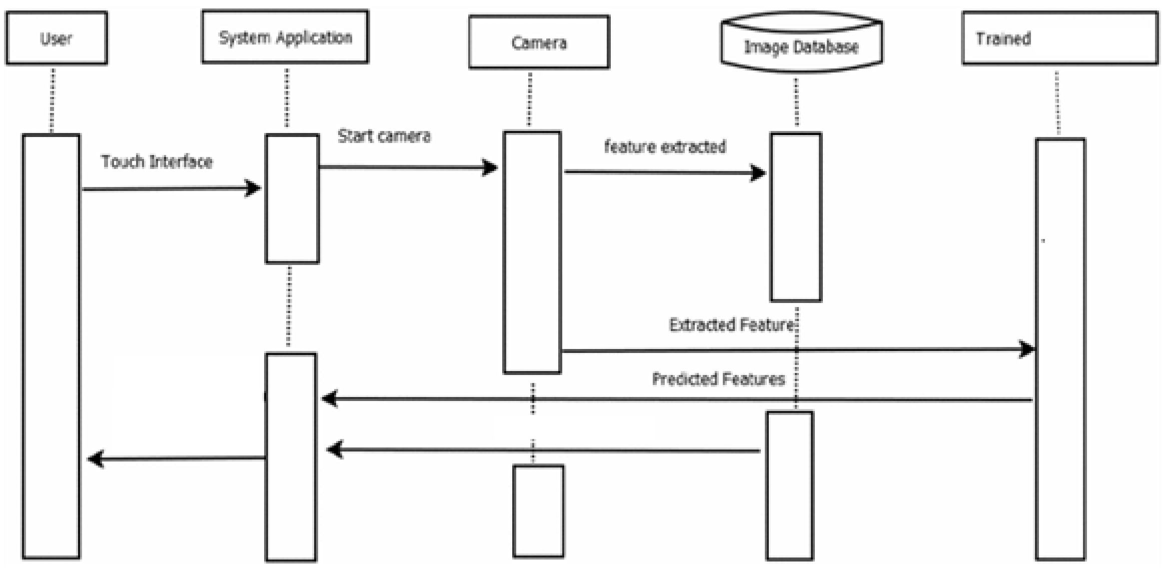


FIGURE 4.2: Sequence Diagram

Chapter 5

System Design

5.1 Proposed Methodology

Our system has an important core of structure which reflects the fact that the images of coherent objects tend to translate, scale, rotate, and deform in the image plane. Our basic object representation is the labeled graph; edges are labeled with distance information and nodes are labeled with wavelet responses locally bundled in jets. Stored model graphs can be matched to new images to generate image graphs, which can then be incorporated into a gallery and become model graphs. Wavelets as we use them are robust to moderate lighting changes and small shifts and deformations. Model graphs can easily be translated, scaled,oriented, or deformed during the matching process, thus compensating for a large part of the variance of the images. Unfortunately, having only one image for each person in the galleries does not provide sufficient information to handle rotation in depth analogously. However, we present results on recognition across different poses.

This general structure is useful for handling any kind of coherent object and may be sufficient for discriminating between structurally different object types. However, for in-class discrimination of objects, of which face recognition is an example, it is necessary to have information specific to the structure common to all objects in the class. This is crucial for the extraction of those structural traits from the image which are important for discrimination ("to know where to look and what to pay attention to"). In our system,class-specific information has the form of bunch graphs, one for each pose, which are stacks of a moderate number (70 in our experiments) of different faces, jet-sampled in an appropriate set of fiducial points (placed over eyes, mouth, contour, etc.). Bunch graphs are treated as combinatorial entities in which, for each fiducial point, a jet from a different sample face can be selected, thus creating a highly adaptable model. This model is matched to new facial images to reliably find the fiducial points in the image. Jets at these points and their relative positions are extracted and are combined into an image graph, a representation of the face which has no remaining variation due to size, position (or in-plane orientation, not implemented here).

A bunch graph is created in two stages. Its qualitative structure as a graph (a set of nodes plus edges) as well as the assignment of corresponding labels (jets and distances) for one initial image is designer-provided, whereas the bulk of the bunch graph is extracted semi-automatically from sample images by matching the embryonic bunch graph to them, less and less often intervening to correct incorrectly identified fiducial points.

Image graphs are rather robust to small in-depth rotations of the head. Larger rotation angles, i.e. different poses, are handled with the help of bunch graphs with a different graph structure and designer-provided correspondences between nodes in different poses.

After these preparations our system can extract from single images concise invariant face descriptions in the form of image graphs (called model graphs when in a gallery). They contain all information relevant for the face discrimination task. For the purpose of recognition, image graphs can be compared with model graphs at small computing cost by evaluating the mean jet similarity.

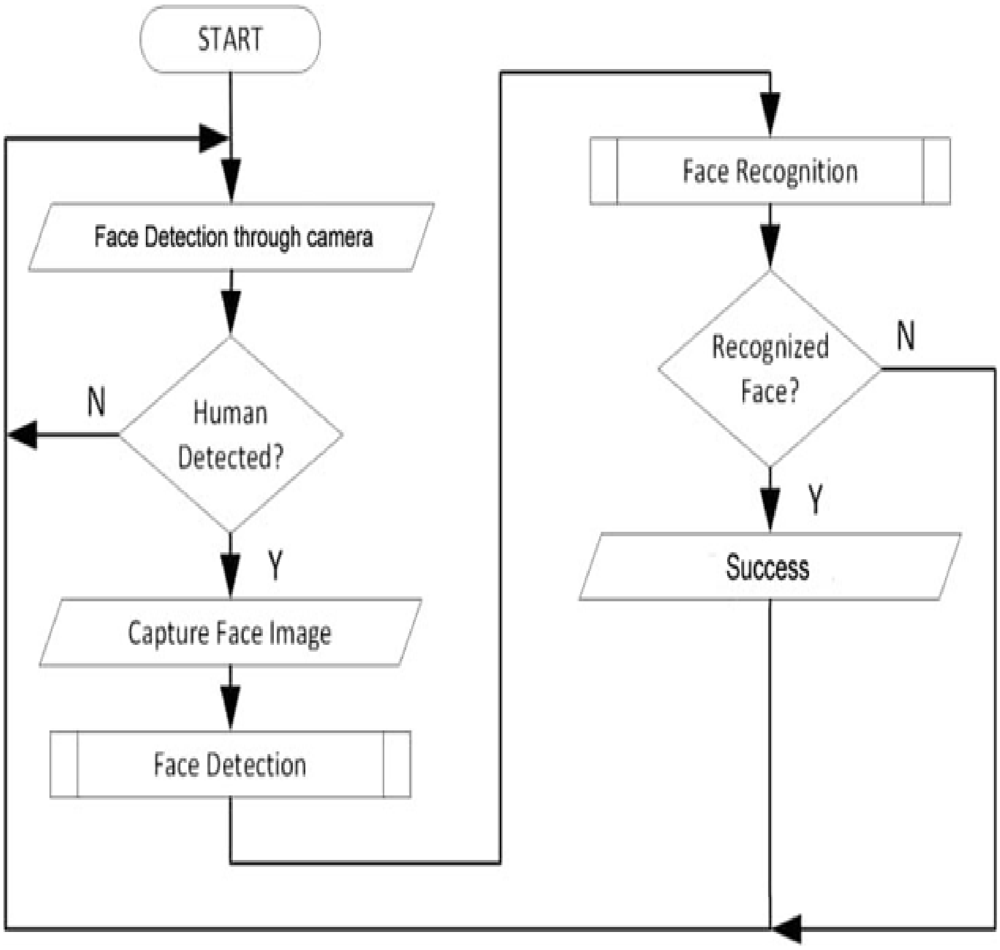


FIGURE 5.1: Workflow

Chapter 6

Result and Conclusion

6.1 Result

We tried to find out the accuracy of our system, so we checked it with different numbers of images(5, 10, 15, .., 30,). We found out that the accuracy of our system is highest when the image size is equal to 15.

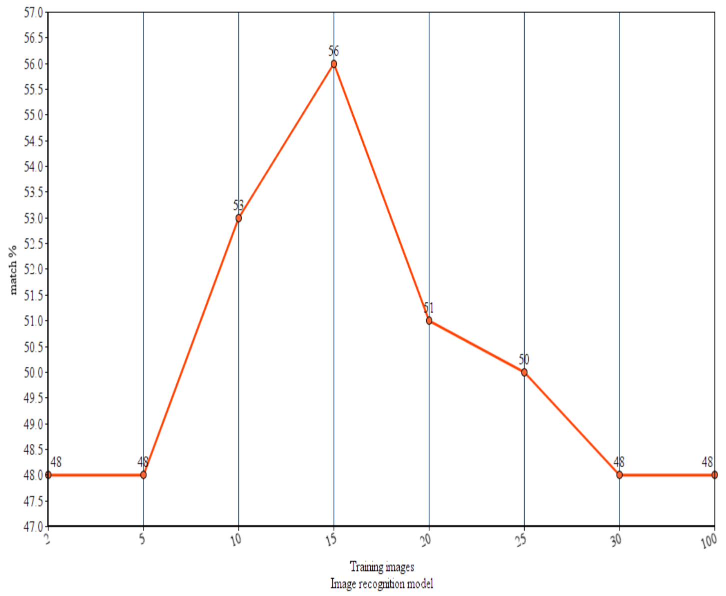


FIGURE 6.1: Graph showing the accuracy with different number of training sets.

6.2 Conclusion

In this work, we have tried to build an application which can detect faces and extract their features. We have implemented our project using Bunch graph method which can handle with issue of illumination in facial recognition.

Bibliography

[1] Debaditya Goswami et al. "Evaluation of face recognition system in heterogeneous environments (visible vs NIR)". In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE. 2011, pp. 2160–2167.

[2] P Jonathon Phillips et al. *An introduction to the good, the bad, & the ugly face recognition challenge problem*. IEEE, 2011.

[3] P Jonathon Phillips et al. "Face recognition accuracy of forensic examiners, superrecognizers, and face recognition algorithms". In: *Proceedings of the National Academy of Sciences* 115.24 (2018), pp. 6171–6176.

[4] Laurenz Wiskott et al. "Face recognition by elastic bunch graph matching". In: *IEEE Transactions on pattern analysis and machine intelligence* 19.7 (1997), pp. 775–779.

[5] Wenyi Zhao et al. "Face recognition: A literature survey". In: *ACM computing surveys (CSUR)* 35.4 (2003), pp. 399–458.