University of Mumbai

Face recognition with the help of Image Classification

Submitted at the end of semester VI in partial fulfilment of requirements
For the degree of

Bachelors in Technology

By

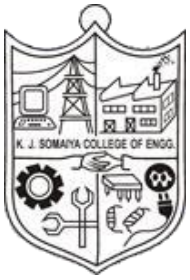
SANKALP JAIN
Roll No: 1813084

BABITA RATUDI
Roll No: 1813090

SUPARSHWA PATIL
Roll No: 1813103

Guide

PROF. RUCHIRA JADHAV



Department of Electronics and Telecommunication Engineering
K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch 2018 -2022

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

CERTIFICATE

This is to certify that the dissertation report entitled submitted **FACE RECOGNITION WITH THE HELP OF IMAGE CLASSIFICATION** by at the end of semester VI of TY B. Tech is a bonafide record for partial fulfilment of requirements for the degree of Bachelors of Technology in Electronics and Telecommunication Engineering of University of Mumbai

Guide/Examiner1

Expert/Examiner2

Date: 07th May, 2021

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

CERTIFICATE OF APPROVAL OF EXAMINERS

We certify that this dissertation report entitled **FACE RECOGNITION WITH THE HELP OF IMAGE CLASSIFICATION** is bonafide record of project work done by during semester VI. This project work is submitted at the end of semester VI in partial fulfilment of requirements for the degree of Bachelors of Technology in Electronics and Telecommunication Engineering of University of Mumbai.

_____

Internal Examiners

_____

External/Internal Examiners

Date: 07th May, 2021

Place: Mumbai-77

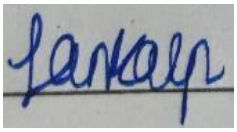
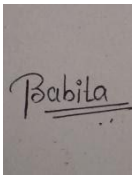
K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

DECLARATION

We declare that this written report submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

 **Signature of the Student** **Roll No.: 1813084**	 **Signature of the Student** **Roll No.: 1813090**
 **Signature of the Student** **Roll No.: 1813103**	

Date: 07th May, 2021

Place: Mumbai-77

ABSTRACT

Support Vector Machines (SVMs) have been recently proposed as a new technique for pattern recognition. In this project, the SVMs are used to tackle the face recognition problem. We will illustrate the potential of SVMs on the dataset, which consists of 800 images of 4 celebrities, containing quite a high degree of variability in expression, pose, and facial details. In this project we used OpenCV (haarcascade) to filter the images not having both eyes and face and used wavelet transform to extract more facial features to feed as input to the machine learning algorithm. We used HTML, CSS, JS for frontend and python flask as backend for the project.

CONTENTS

CHAPTER 1

INTRODUCTION

This chapter presents a brief idea of the project and motivation behind this project and also presents scope of the project. This provides a brief description of project.

1.1 BACKGROUND

Our project topic is **'FACE RECOGNITION WITH THE HELP OF IMAGE CLASSIFICATION.'** In this project, we have trained a ML model using the best fit algorithm (SVM) for face recognition which is incorporated with the FLASK server as a backend technology to create a fully functional responsive website. The user can upload any image of the celebrity shown on the site. The uploaded image is then pre-processed in the backend and then the corresponding recognised celebrity result is displayed. If the image is not detectable then the same will be conveyed to the user. The pre-requisites for this project are ML for training the model, Flask for the backend and HTML5, CSS3 and JS for the frontend part along with basic knowledge of Bootstrap framework to make the website responsive.

1.2 MOTIVATION:

Face recognition technology can be used in wide range of applications such as identity authentication, access control, and surveillance. Interests and research activities in face recognition have increased significantly over the past few years. We wanted to create on a project that can help identification of human beings/personalities and the same ideology can be applied on objects, animals, birds, plants etc. which has many real-time applications in the world.

1.3 SCOPE OF THE PROJECT:

The website is flexible to be upgraded and updated at any course of time. This can be made more and more feasible to the owner of the system. The current project is trained for humans specifically celebrities but it can also be trained and used for recognizing criminals from the database. We can also incorporate google lens or camera for scanning images as input along with the option of uploading it in the dropbox. We can even add a different section for historical figures/personalities or Hollywood celebrities in extension with Bollywood celebrities. This project can be used for educational purposes in museums, schools etc.

1.4 BRIEF DESCRIPTION OF PROJECT UNDERTAKEN

Initially, we will import all the necessary libraries needed for the project in Jupyter Notebook. We will then download the required images of celebrities with the help of image scrapper coded in python. When we take a look at any image, most of the time we identify a person using his/her face. An image might contain faces of more than one celebrity, also the face can be obstructed. Hence, we will be preprocessing all the raw images in the dataset.

The first step in our pre-processing pipeline will be to detect the face from an image. Once the face is detected, we will then detect the facial features i.e. eyes etc., we keep that image only if both the eyes are detected otherwise, we will discard it. For this we will be using haarcascade and wavelet transform. After preprocessing if there are images which do not belong to a particular celebrity in their respective folder then that we will have to remove those images from our preprocessed dataset manually.

After getting the needed dataset we applied wavelet transform on it to extract more features from the facial images. The preprocessed data will be trained for which we will be applying various supervised learning algorithms with suitable kernel. We will be applying grid search cv with logistic regression, random forest, SVM algorithms to check for best accuracy and best set of parameters from the given parameter values.

Based on the best suitable algorithm we will be saving the model as pickle file and dictionary of celebrities as json files which will be later used in backend flask server. We have planned to use dropbox to collect input images from user in frontend and then the base64 string is passed to backend flask server. There the trained model is applied on the input image and then predicted celebrity along with accuracy is sent to frontend to display. We will be using different frontend technologies like HTML5, CSS3, JAVASCRIPT and Bootstrap to make a responsive and working website as final product which will be deployed on local server.

CHAPTER 2

LITERATURE REVIEW

This chapter presents the literature review done before working on the project

- Support Vector Machines (SVM) have been recently developed in the framework of statistical learning theory (Vapnik, 1998) (Cortes and Vapnik, 1995), and have been successfully applied to a number of applications, ranging from time series prediction (Fernandez, 1999), to face recognition (Tefas et al., 1999), to biological data processing for medical diagnosis (Veropoulos et al., 1999). Their theoretical foundations and their experimental success encourage further research on their characteristics, as well as their further use.

- An SVM using a kernel motivated from Fisher Linear Discriminant was shown to outperform the standard linear SVM for a face recognition task in (Tefas et al., 1999).

- Intuitively, given a set of points belonging to two classes, an SVM finds the hyperplane that separates the largest possible fraction of points of the same class on the same side, while maximizing the distance from either class to the hyperplane. According to Vapnik, this hyperplane is called Optimal Separating Hyperplane (OSH) which minimizes the risk of misclassifying not only the examples in the training set but also the unseen examples of the test set.

- In geometric feature-based methods, facial features such as eyes, nose, mouth, and chin are detected.

- It is difficult to discriminate or recognize different persons (hundreds or thousands) by their faces because of the similarity of faces. In this research, we focus on the face recognition problem, and show that the discrimination functions learned by SVMs can give much higher recognition accuracy than the popular standard eigenface approach.

- Wavelet is an increasingly popular tool in image processing and computer vision. Many applications, such as compression, detection, recognition, image retrieval have been investigated.

- Wavelet transform has nice features of space-frequency localization and multiresolution. The main reasons for Wavelet transform popularity lie in its complete theoretical framework, the great flexibility for choosing bases and the low computational complexity.

- We have many frameworks for styling and adding custom CSS like Semantic UI, Materialize CSS, Bootstrap etc. We will be using Bootstrap in our project because it is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

- Facial feature points encode critical information about face shape and face shape deformation.

- SVM take cares of outliers better than KNN. SVM outperforms KNN when there are large features and lesser training data.

- The SVM-Linear, SVM-RBF and CNN model is used to extract useful high-level features automatically given that it provides results comparable with each other, including hyperspectral image classification

- From all the available backend frameworks like Node js, Django, Flask etc. We have selected FLASK because:

1. It is easy to use
2. Built in development server and debugger
3. Integrated unit testing support.
4. RESTful request dispatching
5. Extensively documented.

CHAPTER 3

PROJECT DESIGN

This chapter explains briefly the problem statement, objectives and the research done for the project

3.1 PROBLEM STATEMENT:

Given still or video images of a celebrity, identify or verify the person in the images using a stored database of faces.

3.2 ANALYSIS OF PROBLEM STATEMENT:

A complete face recognition system includes face detection, face preprocessing and face recognition processes. Therefore, it is necessary to extract the face region from the face detection process and separate the face from the background pattern, which provides the basis for the subsequent extraction of the face difference features. Face recognition of the separated faces is a process of feature extraction and contrast identification of the normalized face images in order to obtain the identity of human faces in the images.

3.3 OBJECTIVES:

The objective of this project is to train the model to successfully identify the image given as an input and show the % similarity with all the celebrities stored in the database. We also want to create a fully functional dynamic website for the same. We main aim is to achieve decent accuracy. In this project our objective is to learn:

1. Implementation of machine learning algorithms in real life projects.
2. Extraction of images using image scrapper.
3. How image classification is done using OpenCV.
4. Data cleaning, preprocessing of raw dataset.
5. How to Feature engineering is performed on the raw dataset.
6. Implementation of concepts like Wavelets transform for better facial extraction.
7. How images can be used to train SVM classifier along with other supervised learning classifiers.
8. Deployment of machine learning projects on the responsive website using Flask as a server and using HTML5, CSS3, JS to build the frontend of the website.

3.4 TOOLS AND IMAGE DATASET

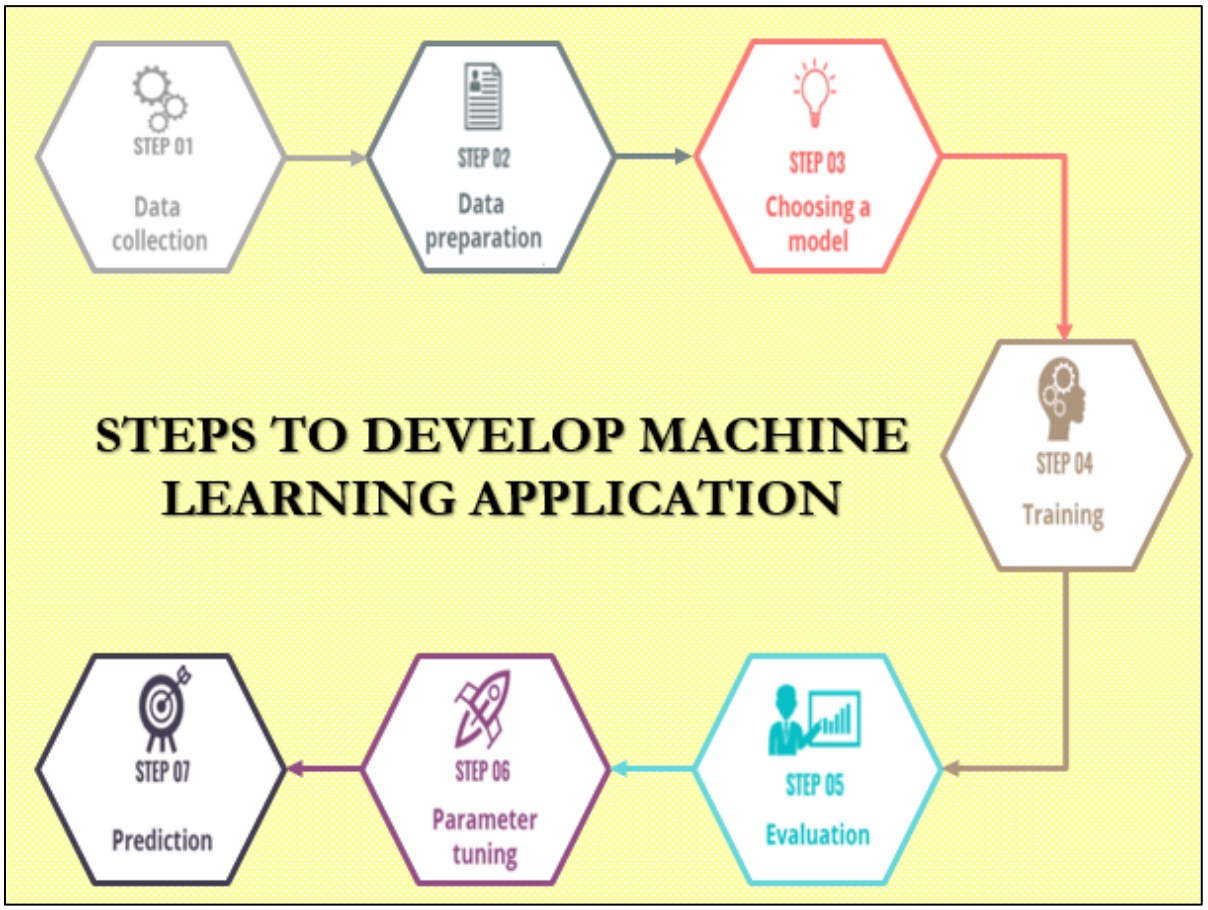
The tools which we have used in our projects are

- Microsoft Visual C++
- Jupyter Notebook from Anaconda Navigator
- Google Colab
- Git.
- PyCharm

The image dataset used for training in this project was scrapped using an image scrapper. Also, some photos were taken from the internet (not containing copyright issues) to test varying conditions.

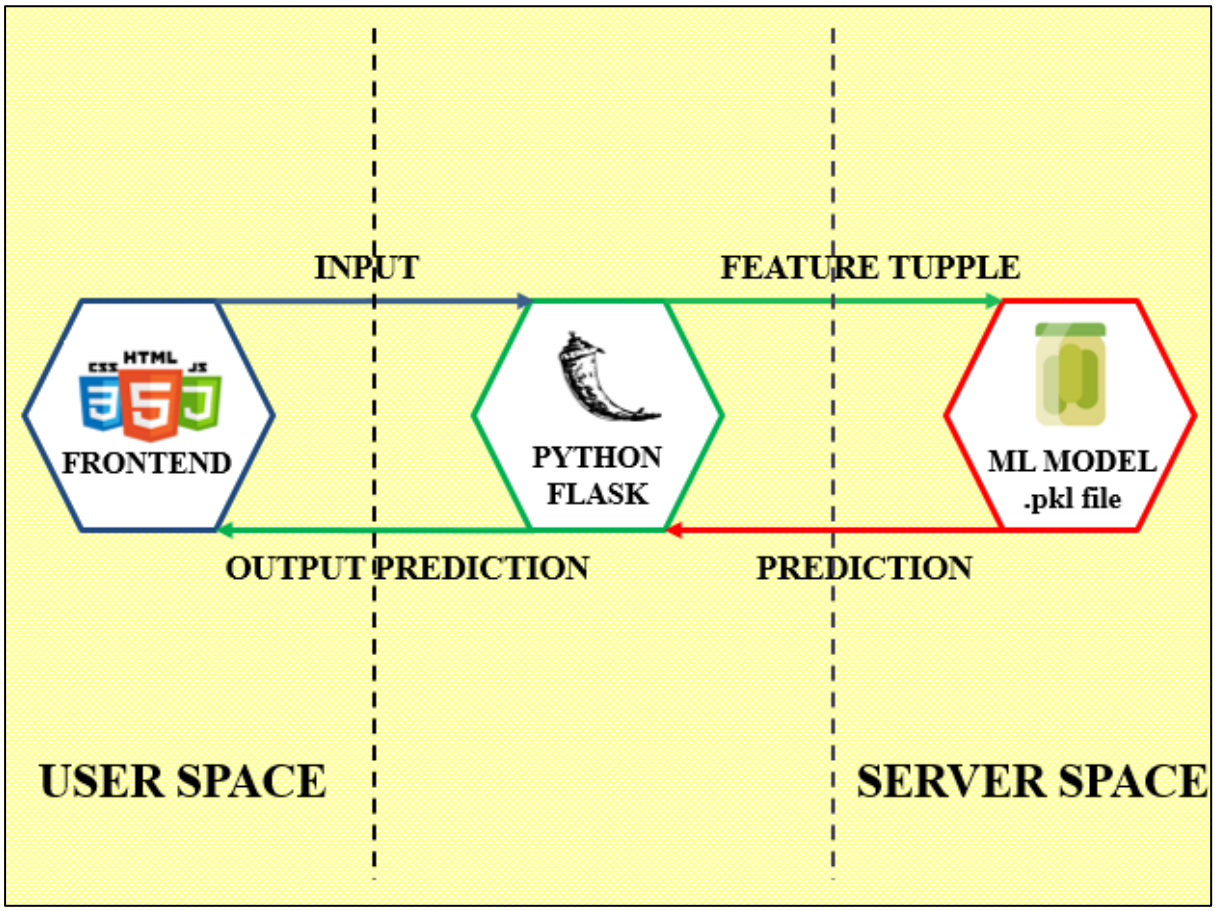
3.5 BLOCK DIAGRAM:

3.5.1 DEVELOPMENT OF ML MODEL



Made using Lucid Chart

3.5.2 BACKEND DEPLOYEMENT USING FLASK



Made using Lucid Chart

3.6 SUPERVISED LEARNING

Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights through a reinforcement learning process, which ensures that the model has been fitted appropriately.

Working of supervised learning: Supervised learning uses a training set to teach models to yield the desired output. This training dataset includes inputs and correct outputs, which allow the model to learn over time. The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized.

Classification: Classification uses an algorithm to accurately assign test data into specific categories. It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labeled or defined. Common classification

algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbor, and random forest, which are described in more detail below.

Supervised Learning Algorithms: Various algorithms and computation techniques are used in supervised machine learning processes. Below are brief explanations of some of the most commonly used learning methods, typically calculated through use of programs like R or Python.

1. Neural networks

Primarily leveraged for deep learning algorithms, neural networks process training data by mimicking the interconnectivity of the human brain through layers of nodes. Each node is made up of inputs, weights, a bias (or threshold), and an output. If that output value exceeds a given threshold, its "fires" or activates the node, passing data to the next layer in the network. Neural networks learn this mapping function through supervised learning, adjusting based on the loss function through the process of gradient descent. When the cost function is at or near zero, we can be confident in the model's accuracy to yield the correct answer.

2. Naive Bayes

Naive Bayes is classification approach that adopts the principle of class conditional independence from the Bayes Theorem. This means that the presence of one feature does not impact the presence of another in the probability of a given outcome, and each predictor has an equal effect on that result. There are three types of Naïve Bayes classifiers: Multinomial Naïve Bayes, Bernoulli Naïve Bayes, and Gaussian Naïve Bayes. This technique is primarily used in text classification, spam identification, and recommendation systems.

3. Linear regression

Linear regression is used to identify the relationship between a dependent variable and one or more independent variables and is typically leveraged to make predictions about future outcomes. When there is only one independent variable and one dependent variable, it is known as simple linear regression. As the number of independent variables increases, it is referred to as multiple linear regression. For each type of linear regression, it seeks to plot a line of best fit, which is calculated through the method of least squares. However, unlike other regression models, this line is straight when plotted on a graph.

4. Logistic regression

While linear regression is leveraged when dependent variables are continuous, logistical regression is selected when the dependent variable is categorical, meaning they have binary outputs, such as "true" and "false" or "yes" and "no." While both regression models seek to understand relationships between data inputs, logistic regression is mainly used to solve binary classification problems, such as spam identification.

5. Support vector machine (SVM)

A support vector machine is a popular supervised learning model developed by Vladimir Vapnik, used for both data classification and regression. That said, it is typically leveraged for classification problems, constructing a hyperplane where the distance between two classes of data points is at its maximum. This hyperplane is known as the decision boundary, separating the classes of data points (e.g., oranges vs. apples) on either side of the plane.

6. K-nearest neighbor

K-nearest neighbor, also known as the KNN algorithm, is a non-parametric algorithm that classifies data points based on their proximity and association to other available data. This algorithm assumes that similar data points can be found near each other. As a result, it seeks to calculate the distance between data points, usually through Euclidean distance, and then it assigns a category based on the most frequent category or average. Its ease of use and low calculation time make it a preferred algorithm by data scientists, but as the test dataset grows, the processing time lengthens, making it less appealing for classification tasks. KNN is typically used for recommendation engines and image recognition.

7. Random forest

Random forest is another flexible supervised machine learning algorithm used for both classification and regression purposes. The "forest" references a collection of uncorrelated decision trees, which are then merged together to reduce variance and create more accurate data predictions.

3.7 TUNING PARAMETERS

In machine learning, kernel machines are a class of algorithms for pattern analysis, whose best-known member is the support-vector machine. The general task of pattern analysis is to find and study general types of relations in datasets. Choosing the right kernel is crucial, because if the transformation is incorrect, then the model can have very poor results. As a rule of thumb, always check if the data is linear and, in that case, always use linear SVM (linear kernel). Linear SVM is a parametric model, but an RBF kernel SVM isn't, so the complexity of the latter grows with the size of the training set. Not only is more expensive to train an RBF kernel SVM, but one also has to keep the kernel matrix around, and the projection into this "infinite" higher dimensional space where the data becomes linearly separable is more expensive as well during prediction. Furthermore, one has more hyperparameters to tune, so model selection is more expensive as well and finally, it's much easier to overfit a complex model.

3.8 SUPPORT VECTOR MACHINE

In machine learning, task of deducing a category from supervised training data is known as Supervised Learning. In supervised learning the training data consist of a set of training examples, where each example is a pair consisting of an input and an anticipated output value. A supervised learning algorithm analyzes the training data and then predicts the correct output categorization for given data-set input. For e.g. Teacher teaches student to identify apple and oranges by giving some features of that. Next time when student sees apple or orange, he can easily classify the object based on his learning from his teacher, this is called supervised learning. He can identify the object only if it is apple or orange, but if the given object was grapes the student cannot identify it.

ADVANTAGES AND DISADVANTAGES OF SUPPORT VECTOR MACHINE

ADVANTAGES:

- There are many folds advantages of using the supervised learning approach of Support Vector Machine (SVM).
- They are very effective when we have very high dimensional spaces. Also, when number of dimensions becomes greater than the existing number of samples, in such cases too SVM is found to be very effective.
- SVM uses a subset of training point also known as support vectors to classify different objects hence it is memory efficient.
- Support Vector Machines are versatile, for different decision function we can define different kernel as long as they provide correct result.
- Depending upon our requirement and application we can choose types of kernel which is most productive for our application.

DISADVANTAGES:

- The disadvantage of SVM is that if the number of features is much greater than the number of samples, the method is likely to give poor performances.
- SVM gives efficient result for small training samples as compared to large ones.
- SVMs do not directly provide probability estimates, so these must be calculated using indirect techniques. Also, we can have Non-traditional data like strings and trees as input to SVM instead of featured vectors.
- Lastly selecting appropriate kernel for the project is a big issue which depends upon user's requirement.

3.9 OPEN CV

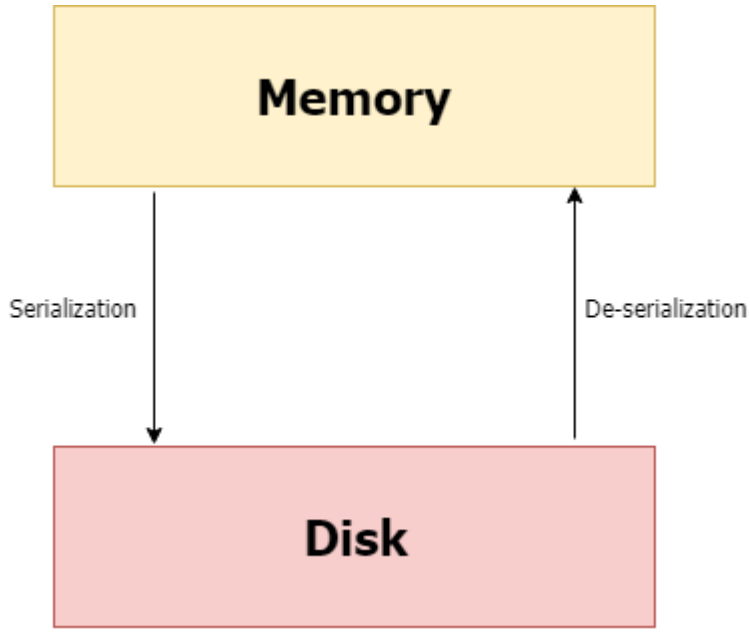
OpenCV acronym for Open Source Computer Vision Library is a library containing functions for computer vision. It is developed by Intel and now handled and supported by Willow Garage. The library is functional cross platform and runs on Windows, Android, FreeBSD, Maemo, iOS, OpenBSD, Linux and Mac OS. The current release of the library is obtained from the Sourceforge and they also provide the binaries for the user, so that they can develop according to their requirements. OpenCV makes use of CMake to compile source files to start using the library [12]. The main focus of this library is on the real-time image processing functionality and implementing the machine learning algorithms. By using this we can improve the cost of computation and take an initiative to advance the CPU – intensive applications. The areas of application where openCV can be useful are facial recognition system, mobile robotics, gesture recognition, segmentation, object identification, motion tracking and many more. OpenCv also includes a statistical machine learning library that supports the above areas of application. The name of the functions that supports this library are decision tree learning, expectation maximization, gradient boosting trees, Naïve Bayes classifier, k- nearest neighbor, artificial neural network, support vector machine (SVM) and many more

3.10 HAARCASCADE CLASSIFIER

In statistical model-based training, we take multiple positive and negative samples and extract different features from these samples. These distinctive features are then compressed into statistical model parameters which are used as special property to classify different objects. By making adjustments in these parameters we can improve the accuracy of classification for these algorithms. The fundamental concept for detecting objects from images for Haar classifier is the Haar-like features. These features exploit the difference in contrast values between contiguously grouped pixels instead of using the intensity values of that particular pixel. These contrasting values between the grouped pixels are used to detect relative light and dark spot from the images. These two to three contiguous groups with a comparatively contrasting values form a Haar-like feature. In images we have object of different sizes, these Haar features can be scaled by increasing or decreasing the size of the grouped pixel being examined. This scaling of the pixels makes it possible to detect and extract objects with varying sizes. In opencv, we first take the positive and negative samples and form the corresponding database. After that we create a vector file and then build the classifier. After building the classifier we check for the performance and accuracy of the training dataset. If we find satisfying results, we create a xml file which contains all the features for detecting objects. This is the whole procedure for detecting faces or any objects using Haar classifier. Afterwards we retrieve the classifier's data from the xml file and use this data to classify objects for the testing data set.

3.11 SERIALIZING/DE-SERIALIZING

In simple words serializing is a way to write a python object on the disk that can be transferred anywhere and later de-serialized (read) back by a python script.



Serialization, De-Serialization

The model is converted which is in the form of a python object into a character stream using pickling. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script. We will save our trained model to the disk using the *pickle* library. *Pickle* is used to serializing and de-serializing a Python object structure.

CHAPTER 4

IMPLEMENTATION

This chapter explains briefly the problem statement, objectives and the research done for the project

4.1 TRAINING OF ML MODEL

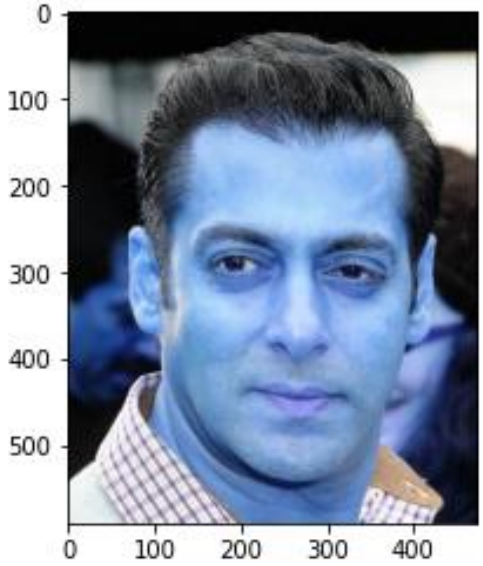
4.1.1 COLLECTION OF DATA

500 images of each celebrity were collected by web scrapping and extracting data using python image scrapper.

```
Duckduckgo search: Salman Khan
Downloading results into /content/drive/MyDrive/Colab Notebooks/images/Salman Khan
100.00% [500/500 01:02<00:00 Images downloaded]
[PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/001_89d74bef.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/002_c9c64e09.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/003_104e531f.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/004_ee7703f7.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/005_8c69e83e.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/006_42909575.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/007_536a0c75.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/008_cfce55b9.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/009_5457a2c4.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/010_0868aead.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/011_13092f07.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/012_58f021eb.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/013_20280290.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/014_ae96f04d.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/015_53abd028.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/016_e31a532e.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/017_1381bdfa.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/018_04ced675.jpg'),
 PosixPath('/content/drive/MyDrive/Colab Notebooks/images/Salman Khan/019_739ca917.jpg'),
```

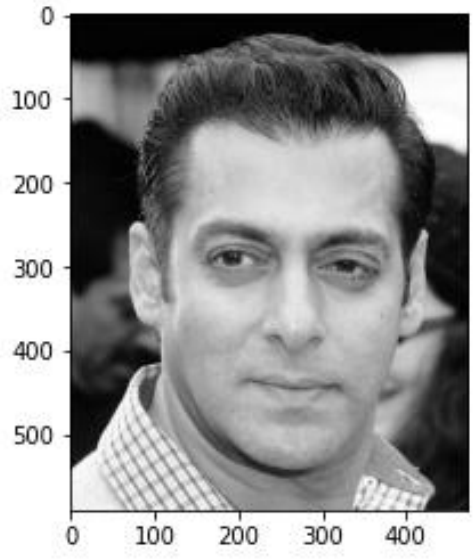
4.1.2 READING IMAGE OF CELEBRITY

cv2.imread() method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.



4.1.3 IMAGE MANIPULATION WITH RGB2GREY

Image data is represented as a matrix, where the depth is the number of channels. An RGB image has three channels (red, green, and blue) whereas the returned greyscale image has only one channel. Accordingly, the original color image has the dimensions 100x100x3 but after calling rgb2grey, the resulting greyscale image has only one channel, making the dimensions 100x100x1

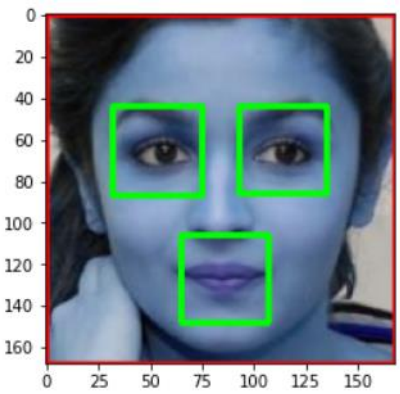


4.1.4 ANALYSE THE COLLECTED DATA

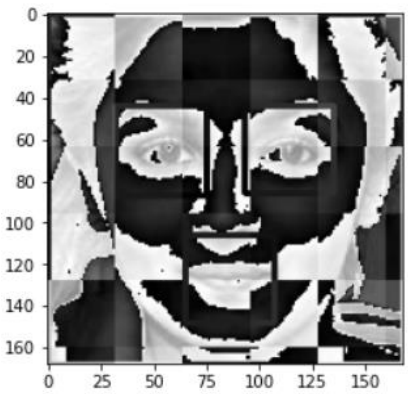
Once the data was collected, we needed to make sure it was in a useable format. The benefit of having this standard format was that we could mix and match algorithms and data sources. We needed to do some algorithm-specific formatting here. Some algorithms need features in a special format. When we look at any image, most of the time we identify a person using a face. An image might contain multiple faces, also the face can be obstructed and not clear. For optimal face detection, we needed the facial features of the input data to be clearly visible.

4.1.5 PREPARATION OF INPUT DATA

The first step in our pre-processing pipeline is to detect faces from an image and we have done that using Haarcascades (OpenCV). Once face is detected, we will detect eyes, if two eyes are detected then only, we keep that image otherwise discard it. But it will even detect the faces which are with the desired person in the images. So, we have to remove them manually from the pre-processed dataset. In wavelet transformed image, you can see edges clearly and that can give us clues on various facial features such as eyes, nose, lips etc.



Detection of facial features



Wavelet transformation

4.1.6 SPLIT INTO TRAIN AND TEST SETS

train_test_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. The train-test split is a technique for evaluating the performance of a machine learning algorithm. It can be used for classification or regression problems and can be used for any supervised learning algorithm. The procedure involves taking a dataset. The data are converted into train and test sets. The training data will include 75% of images and the remaining 25% was used to test the model. To avoid getting different values for train and test every time you select the random_state you want to use. This will also have an impact on the evaluation metrics as if the random_state is not selected the values from the evaluation will differ. The values 0, 1 are the most common values used.

4.1.7 TRAIN THE ALGORITHM

The training data is an initial set of data used to help a program understand how to apply technologies like neural networks to learn and produce sophisticated results. Images in cropped folder were used for model training. We used these raw images along with wavelet transformed images to train our classifier. We used SVM with rbf kernel tuned with heuristic finetuning to train the data.

4.1.8 TEST THE ALGORITHM

The test set is a set of observations used to evaluate the performance of the model using some performance metric. It is important that no observations from the training set are included in the test set. After training the model we tested its precision and recall values.

4.1.9 TEST FOR DIFFERENT ALGORITHMS

GridSearchCV is a function that comes in Scikit-learn's (or SK-learn) model_selection package. This function helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. We used GridSearchCV to test accuracy for different algorithms and to select the algorithm having the best accuracy and get the best set of values for the parameters. The algorithms which we implemented were SVM, Logisitc Regression and Random Forest.

4.1.10 ANALYSE THE RESULTS

A heatmap contains values representing various shades of the same colour for each value to be plotted. Usually the darker shades of the chart represent higher values than the lighter shade. For a very different value a completely different colour can also be used. We plotted the heat map of confusion matrix using the seaborn library to analyse the trained model's efficiency.

4.1.11 SCORE MODEL

Accuracy of the matrix is always calculated by taking average values present in the main diagonal i.e. Classification Accuracy. We calculate this by calculating the ratio of correct predictions by a total number of input Samples. It works great if there are an equal number of samples for each class.

4.1.12 SAVED THE MODEL

We will save our trained model to the disk using the *pickle* library. *Pickle* is used to serializing and de-serializing a Python object structure. In which python object is converted into the byte stream. *dump()* method dumps the object into the file specified in the arguments. We saved trained model the model as pickle file and labels(dictionary) as json file which will be used in the backend functionality of this application by the flask server.

4.2 INCOPORATION WITH FLASK

Flask is a micro web framework written in Python. It is classified as microframework because because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

4.2.1 Server.py

This contains Flask APIs that receives input images from the user in the form of base 64 string, through GUI or API calls, computes the predicted value based on our model and returns it.

We will use the flask web framework to handle the POST requests that we will get from the *app.js*.

In the following section of the code, we have created the instance of the *Flask()*
Here, we have bounded */api* with the method *classify_image()*. In which predict method gets the data from the json passed by the scratch.classify_image() method takes input from the json and converts it into 2D *numpy array* the results are stored into the variable named *response* and we return this variable after converting it into the json object using flasks *jsonify()* method.

Finally, we ran our server. There we had used port 5500 and had set *debug=True* so that if we got any error then we could debug it and solve it.

4.2.2 Scratch.py

This contains code for the machine learning model to predict celebrity of the input image based on the trained data.

We loaded the saved model into the *__model*. In this section we checked whether the input image is suitable for image classification using OpenCV haarcascade (detected face and eyes) and was further preprocessed into the required shape.

4.2.3 Wavelet.py

This contains code for wavelet transform that is applied on the input image for better feature extraction and to know more about the facial features. In wavelet transform we converted the color images into gray scale images and normalized the images. This code was applied on the image in scratch.py.

4.3 OBSERVATION:

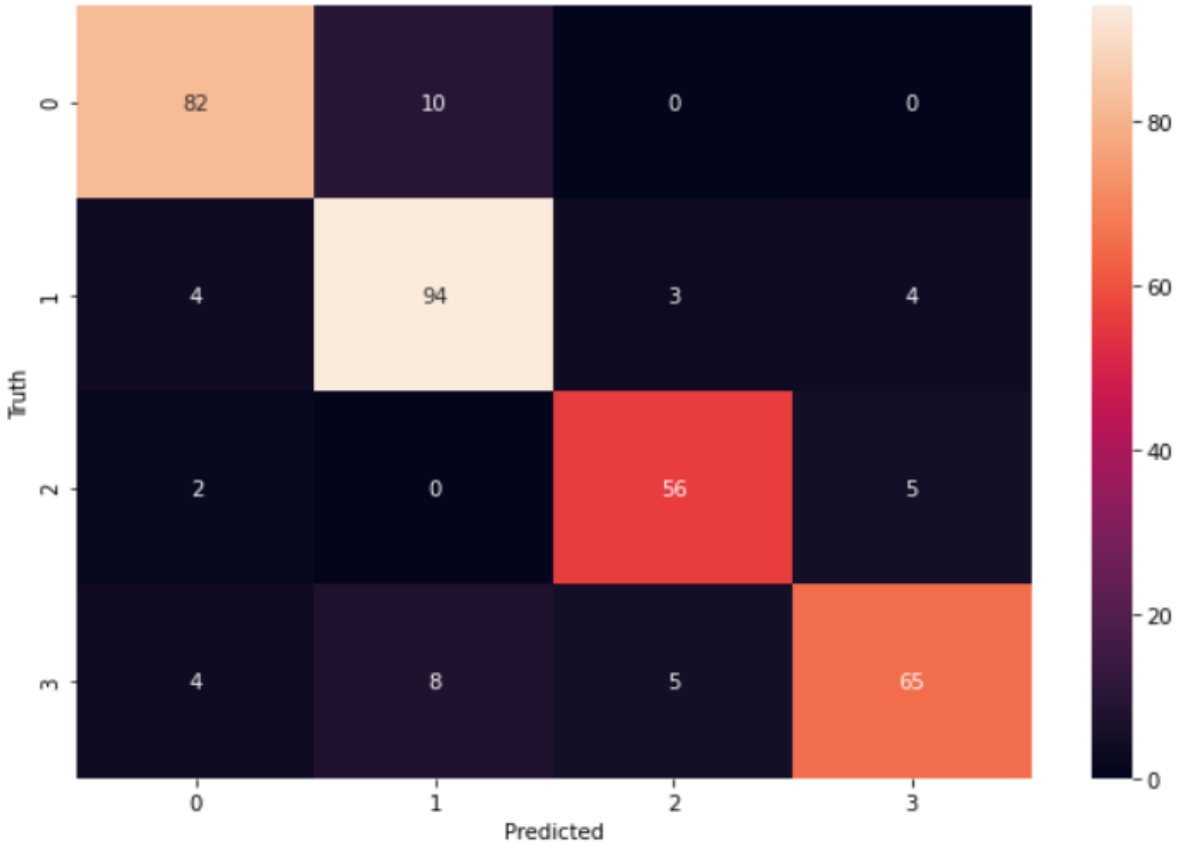
Accuracy:

```
0.8888888888888888
```

Confusion Matrix:

	precision	recall	f1-score	support
0	0.94	0.90	0.92	92
1	0.85	0.91	0.88	105
2	0.85	0.97	0.90	63
3	0.93	0.78	0.85	82
accuracy			0.89	342
macro avg	0.89	0.89	0.89	342
weighted avg	0.89	0.89	0.89	342

Heat Map:



CHAPTER 5

CONCLUSION

This chapter presents the conclusion, result and future scope of the implemented project.

5.1 CONCLUSION

This project has helped us get first hand experience how machine learning algorithms are being applied in real time projects. We also learnt OpenCV, haarcasde as well as wavelet transform for extracting features out of human face. Based on our research image recognition and authentication is a vast topic and has a great scope. Thus, to implement and test the working of the trained model, we incorporated with it with a functional website made using HTML5, CSS3, JAVASCRIPT and Bootstrap and Flask in backend. Overall, we had developed knowledge about application of Machine learning as well as website development.

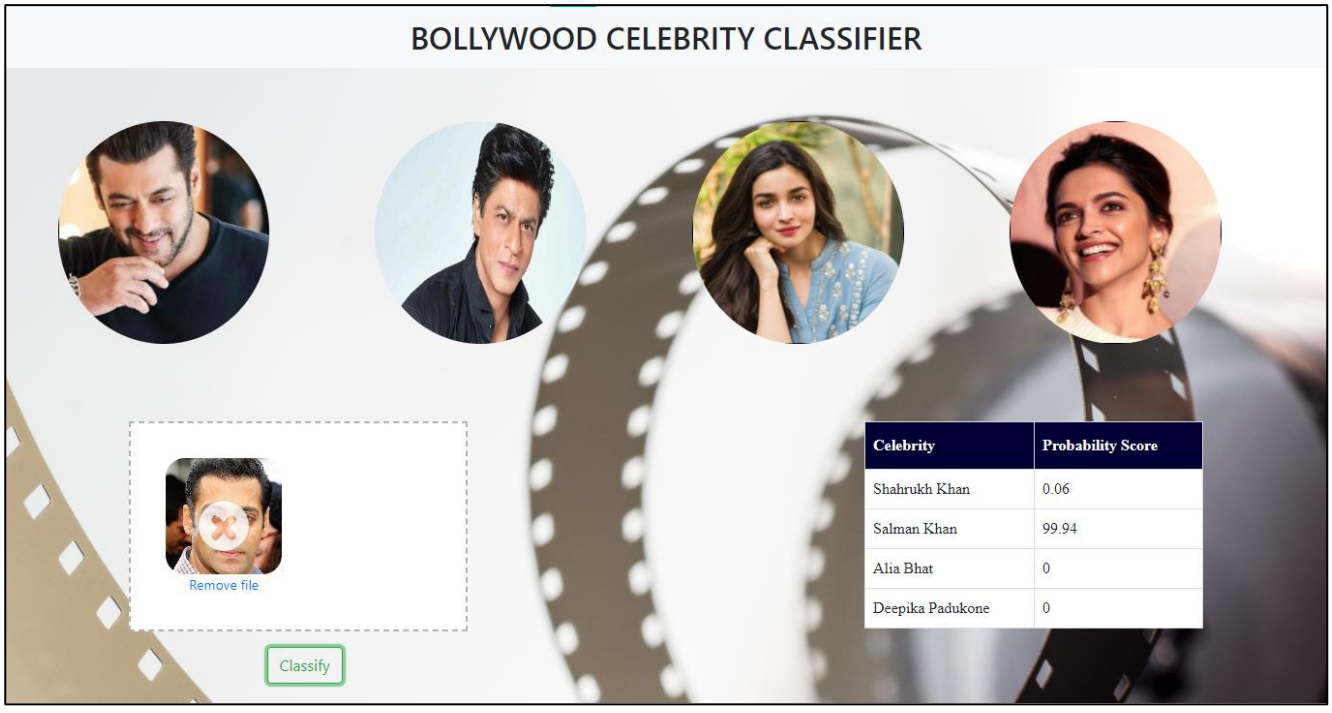
5.2 RESULT

We got 89% of accuracy of the classification with the highest Precision rate of 94% and Recall rate of 90% by using SVM with the best parameter determined by GridSearchCV and Flask as a backend server to connect algorithms to the front end of a website.

Final Layout of Website

INPUT IMAGE: DEEPIKA PADUKONE

INPUT IMAGE: SALMAN KHAN



INPUT IMAGE: SHAHRUKH KHAN

INPUT IMAGE: ALIA BHAT

ERROR IS DISPLAYED IF IMAGE IS NOT DETECTABLE

RESPONSIVE WEBSITE

5.3 FUTURE SCOPE:

- The idea of project can be used for recognizing criminals from the database of police.
- The idea of the project can be extended to Historical figures/personalities or Hollywood celebrities in addition with Bollywood celebrities.
- The idea of project can be used for educational purposes in museums.
- The idea can even be extended to animal hospitals, agriculture-based industries to detect agro based products, breed of animals, flowers and related details effectively which may save a lot of time.
- Image recognition can be done for side profile of celebrity.
- It can be implemented for bio-medical report identification.
- This project can be updated to identify multiple faces in a single photo and provide the corresponding results for all the detected faces.
- Face Authentication can also be implemented for security purposes as well as for general use like attendance for colleges, meetings etc.

CHAPTER 5

BIBLIOGRAPHY

This chapter contains the citations and references used while working on this project

[1] https://ieeexplore.ieee.org/document/8390191/

[2]https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1322&context=etd_projects

[3] https://jakevdp.github.io/PythonDataScienceHandbook/05.14-image-features.html

[4] http://www2.ece.ohio-state.edu/~aleix/CVPR09.pdf

[5] https://towardsdatascience.com/svm-and-kernel-svm-fed02bef1200

[6] https://medium.com/@dataturks/understanding-svms-for-image-classification-cf4f01232700

[7] https://towardsdatascience.com/logistic-regression-for-facial-recognition-ab051acf6e4

[8] https://rpubs.com/Sharon_1684/454441

[9] https://www.researchgate.net/publication/221621494

[10]https://scikitlearn.org/stable/auto_examples/applications/plot_face_recognition.html#sphx-glr-auto-examples-applications-plot-face-recognition-py

[11] https://www.worldscientific.com/doi/abs/10.1142/S0218001402001575

[12]https://www.researchgate.net/publication/246548380_Wavelet_Transform_based_Feature_Extraction_for_Face_Recognition

[13] Face Recognition by Support Vector Machines: Guodong Guo, Stan Z. Li, and Kapluk Chan

[14] Hayder Hasan et al 2019 IOP Conf. Ser.: Earth Environ. Sci. 357 012035

[15] Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 3, No. 1, March 2015, pp. 16~20
ISSN: 2089-3272

[16] Wavelets and Support Vector Machines for Texture Classification: Kashif Mahmood Rajpoot; Nasir Mahmood Rajpoot

CHAPTER 5

ACKNOWLEDGMENT

This project would not have been possible without the kind support of many individuals. We would like to extend our thanks to all of them.

We are highly indebted to PROF. RUCHIRA JADHAV for her guidance and constant supervision as well as for providing insightful information regarding the project, and also for her support in the completion of the project.