Fake Profile Detection in Social Media Using Image processing and Machine learning

by

Shuva Sen
16101202
Mohammad Intisarul Islam
16301145
Samiha Sofrana Azim
17101290
Fatema Akhtar Norin
16301172
Samiha Tasnim Shuha
17201070

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Shuva Sen

Shuva Sen
16101202

Intisarul

Mohammad Intisarul Islam
16301145

Samiha

Samiha Sofrana Azim
17101290

Norin

Fatema Akhtar Norin
16301172

Shuha

Samiha Tasnim Shuha
17201070

Approval

The thesis/project titled "Fake Profile Detection in Social Media Using Image processing and Machine learning" submitted by

1. Shuva Sen (16101202)
2. Mohammad Intisarul Islam (16301145)
3. Samiha Sofrana Azim (17101290)
4. Fatema Akhtar Norin (16301172)
5. Samiha Tasnim Shuha (17201070)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 02, 2021.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain
Assistant Professor
Department of Computer Science and Engineering
Brac University

Co-supervisor:
(Member)

Nazmus Sakeef
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Almost everybody has a social media presence in today's technologically advanced world. As a result, making fake accounts is very easy. The term "fake profile" refers to a person who may pretend to be someone else. These accounts are mostly used to impersonate others and defame them. Furthermore, a fake account can be used for various reasons, including igniting political feuds, spreading misleading facts, and disseminating news about current sensitive topics. Since fake profiles pose such a serious threat to everyone, a model was proposed that might aid in the reduction of fake profiles. It can assist with identifying accounts that could be accused of being fraudulent, such as those without a profile photo. To ensure that each user has a unique profile, machine learning and image recognition was used in our model. Our model attempted to discourage users from creating accounts using the photo or knowledge of another person. To do this, One Time Password (OTP) was implemented so that fake users can not get the chance to create an account by using another person's name. Fake accounts needed to identify by using deep learning from a real dataset of people's answers. To detect the false results, the k-means algorithm was implemented on our dataset. When the k-means clustering algorithm was used on the dataset, it was discovered that our detection accuracy was 75.30 percent.

Keywords: Fake profile; Machine Learning; image recognition; One Time Password ; k-means; accuracy ; fraud

Acknowledgement

First of all, we would like to praise the Almighty for whom our thesis has been completed without any major hurdle.

Secondly, we would like to thank our esteemed supervisor – Dr. Muhammad Iqbal Hossain, for his invaluable supervision, support, and tutelage throughout our entire research. He convincingly guided and encouraged us to be professional and do the right thing even when the road got tough. Without his persistent help, the goal of this project would not have been realized. We would also like to thank Nazmus Sakeef for his kind support and guidance.

And finally, to our parents, without their comprehensive support, it may not be possible.

Without their kind support and prayer, this work would not have been possible. Our appreciation also goes out to everyone who helped us in any manner during our studies.

Table of Contents

List of Figures

Chapter 1

Introduction

Online social networks such as Facebook, Twitter, and Instagram have become a constant necessity for our generation, allowing us to connect with people all over the world quickly and easily. Everyone, from millennia-ls to the elderly, is using online social networking sites to improve their lives. They use these social networking platforms to discover and make friends by making and posting personal memories, photos, videos, and chats. Students, for example, use social networking features to acquire more expertise and equip themselves with a variety of specialties. Tutors use online social networks to connect with their students and help the students in their learning process. Many businesses have used a variety of websites to market and sell their products and services online. Public agencies use social media to effectively provide government programs and keep citizens updated of different circumstances. Every kind of person is relying on social networking platforms for numerous reasons. Hence, to get connected with every other person, they need to share their personal information to create their profile which is available on social networks. Having more than 2.4 billion monthly active users, Facebook is recognised as the most famous worldwide used social network. It allows people worldwide to connect by exchanging images, texts, and remarks. Individuals use Facebook all around the world to communicate with others for their purposes. Any good practice, though, comes with its own set of issues. Unfortunately, often people use Facebook in an unacceptable manner. They created accounts by using other people's information with a tendency of harassment, spreading fake news and creating panic among the people. They even aim to embarrass celebrities by making profiles of their names and personal details. According to Facebook's study, the social media platform has deleted 5.4 billion false profiles in 2019. According to Facebook, about 5 percent of monthly active users were fraudulent. According to the study, one out of every ten likes on a Facebook post might be a response from a false account.

To identify current fake accounts and prevent fake users from creating new ones, a deep learning algorithm was used to detect fake accounts along with the concept of image processing and OTP to prevent fake users from creating new ones. the k-means algorithm was applied to distinguish fake data from a real dataset by calculating euclidean distances and modifying the centroid points.

1.1 Motivation

The various social platforms have been an integral aspect of people's everyday lives. Everyone's public interaction has evolved in the modern era to being connected to online interpersonal organisations. These platforms have brought about a significant change in the way we track our civic activity. It has proven to be easier to add new companions and keep in contact with them and their posts. Almost everyone has an account on various social networks and sometimes more than one account or on different platforms. In the recent era of technology as the applications and the utilization's increase in our daily life, we continuously post some unwanted and unaware stuff in social networking and create a mess in the social platform. As a result, creating a fictitious account is quite easy. In the name of a female, there are over 200 million fake Facebook profiles. As a result, controlling these accounts is difficult, and money laundering, smuggling, and other anti-social activities result from these false accounts.Furthermore, these fictitious identities are being used to spread rumours, hate speech, or even post pictures or thoughts of other people without their permission. In this case, we will consider false accounts and individuals who use this site to launder money and put a stop to such a serious situation.

To date, no one has come up with a feasible solution to these problems like fake profiles, online impersonation, etc. We hope to include a mechanism in this project that allows for the automated identification of false accounts, ensuring that people's social lives are protected. We also hope that we can make it easy for sites to handle the large number of profiles that cannot be handled manually by using this automatic detection strategy. We hope to reduce it so that no fake news or information cannot be spread from any fake accounts. It will also help to reduce identity theft.

We hope to make contributions as follows:

i) Develop an innovative method for accurately detect fake account among social network users, based on various activity collection and analysis.

ii) Provide a detailed analysis of those fake accounts we detected based on the collected dataset.

1.2 Problem Statement

Fake profiles are being used to spread rumours and hate speech and post pictures and thoughts of other people without their permission. This type of action entails the mass production of false identities to launch an online assault on social media. We will also create multiple accounts in our app for comparison. For this, a database will be introduced where all the information about currents users will be stored.The identification would be focused on the users' Facebook behavior and interactions with other users and their user feed info. We often use image recognition to determine whether or not different accounts have the same images. We will focus on which features are missing and different in the fake accounts by comparing our accounts features and dataset features from the database using image processing and algorithms.

1.3 Objective and Contributions

Our preliminary plan is to detect fake accounts from various social media platforms starting from Facebook.

i) We constructed a database with some fundamental information about the users to distinguish the characteristics of real accounts.
ii)We additionally used image processing to classify the images being used to open a new account.
iii) We used face detection and object detection features of OpenCV to do so.
iv) We used the K means algorithm to divide our database into different clusters.
v) Lastly, we added a one-time password(OTP) system to assure additional security to the accounts.

1.4 Thesis structure

This report describes a model which has been designed to find out the fake accounts in social media and also showed a way of also preventing it.

Firstly, the Introduction Chapter (Chapter 1) indicates the motivation and inspiration behind our work. The summary of our work and the main purpose of our work is stated in this chapter.

Then, in the background chapter (Chapter 2), we basically wanted to summarize some of the previous works where other people also worked on the topic of detecting fake accounts in social media. These papers were studied and analyzed thoroughly to have a rough idea about the different direction towards the solution of the problem. Moreover, the algorithms used in the papers are also discussed.

Next, the proposed model chapter (Chapter 3) indicates our dataset was briefly discussed and also the process of the dataset preprocessing. Furthermore, K-means algorithm was also described and how the algorithm was implemented on our dataset to find out the fake and real data. The idea of image processing and OTP were also shared in this section which was actually implemented to prevent the user from creating a fake account.

After that, in the Experimentation and Result Analysis Chapter (Chapter 4), the flowchart of our implementation for the detection through algorithm was written. The flowchart of the implementation of image processing and OTP was also drawn and described shortly. The accuracy of our detection part was shown in the result analysis chapter.

Lastly, in the conclusion part (Chapter 5), the significance of our model and also the process is shortly explained.

Chapter 2

Background

2.1 Literature Review

In recent times, many researches have been captivated by removing fake accounts; thus, extensive research has been carried out to detect false accounts. Also, different perspectives have been proposed to expose the fake profiles based on attribute comparability, the similarity of community of friends, inquiry of profile for a time interval, the similarity of attribute together with IP address. [10]. It brings out false and fraud accounts to generate a balance in the dataset by making use of re sampling methods. The list of the algorithms they used in this model is- supervised ML, map reduction, pattern recognition approach and unsupervised two-layer meta-classifier method. PCA algorithm, SMOTE, Medium Gaussian SVM, Regression, Logistic Algorithms, Various classifier algorithms. In this proposed model, linear SVM gives 95.8 percent accuracy, medium Gaussian SVM provides 97.6 percent, and logistic regression gives 96.6 percent. Moreover, Graph Analysis is utilized in many applications, such as displaying circuit diagrams to discover SHAPE, image matching, and social network analysis graphs are analyzed to decode most of the social network issues. It was discovered that the Medium Gaussian SVM algorithm anticipates fake profiles with a high area under the curve=1 and a low false-positive rate=0.02.

The emergence of online social media, such as Facebook, has become an essential element in human communication. User participation in such websites produces a tremendous impact on human society. But nowadays, a fake account has been a severe problem since the creation of websites with member login features. A fake account is specified as an account that does not act like an average user, which indicates that the account has been involved in a specific activity not represented by the user, such as spamming with advertisements. To detect fake accounts, the activity of each account will be collected, and after that, the models will be trained by using these data.

Moreover, accounts are extracted by the words, comments, and posts they are using in their timeline. In this model, Random forest along with C4.5 and adaptive boosting with decision stump are used as a second classifier that is generated behind it to focus on the instances in the training data, in case the accuracy of the first classifier is less effective. [8]. All the classifiers output real-time prediction and probability scores that can rank each account in the dataset. ROC curve (Receiver Operating Characteristics curve) has been generated to measure the classifiers' performance, along with some other metrics such as precision, recall, F-1 score, etc. This detection of real-world data performs very well, and the models do not over fit according to the result.

On the proposed model, the main attention is on identifying and discovering fake accounts on Facebook. First of all, user feed data is analyzed that is the way location is executed dependent on the client's customary activities and their correspondence with Facebook clients.[7]. The recognition of several attributes are required which distinguish user actions.

Moreover, they break down the fake accounts into two groups-user misclassified accounts and undesirable accounts.

(i) User-misclassified accounts act for the personal profiles which are generated by users for a business, organization, or non-human entity such as a pet

(ii) Undesirable accounts are the user profiles which violates Facebook terms of service, including spamming and this is done intentionally for specific purposes.

A set of 17 attributes are named and measured, which dictates the actions and behavior of Facebook users. Then, these attributes are preceded as input in setting up learning models. 12 supervised machine learning classification algorithms are operated which are prepared by Weka, namely, k-Nearest Neighbor, Naïve Bayes, Decision Tree (J48, C5.0, Reduced Error Pruning Trees Classification (REPT), Random Tree, Random Forest), Decision Rule Based (OneR, RIPPER (JRip)), Support Vector Machines (SVM, Sequential Minimal Optimization (SMO)). After implementing all these algorithms, the lowest accuracy rate of 58 percent was resulted in Naive Bayes but it could detect 92 percent of the actual fake accounts, the highest of them all and OneR. 62-73 percent of the existing fake profiles were identified by all the classifiers.

Our daily life has been surrounded by online Social networks. The growing demand for online social networks has led to expanding disgrace and dishonor in digital social media in the shape of fake profiles, viral marketing, and security breach attacks, etc. Supervised Machine learning algorithms are used to dig out fake profiles [12]. Supervised Machine learning algorithms take input from a dataset, and taking one value from the dataset anticipates the other dataset values. The dataset has been observed for a long period of time then they collect all the dataset. There are a total of five machine learning algorithms and a skin detection algorithm which has been applied to find out in decent pictures from account holders. Additionally, a deep learning method has been executed for recognizing the picture whether it is an image of a human face or not. If the portrayal contains a human face, it will go under skin detection, where the percentage of skin present in the image will be computed. Using all these algorithms, we obtained 80 percent accuracy from ML, and the rest of the classifier has 60-80 percent accuracy and the error rate is 20 percent. For the upcoming research, a more elaborated algorithm for skin detection can be executed. Researchers are planning to work on natural language processing (NLP) techniques to detect fake accounts more precisely.

There are some papers and announced datasets about the spotting of fake engagement actions itself and the observation of users who are engaged in indecent activity in digital Social Networking like Facebook and Twitter.Our main attention is to detect the fake followers on Twitter by operating machine learning algorithms. From the point of view of fake engagement activity, Instagram is also examined in some projects. This paper proposes SVM, Neural network, SMOTE-NC, and Naïve Bayes with Gaussian distribution [11]. Through the application of different kinds of algorithms, it is a target to accomplish various features of the dataset (i.e., independence, reparability, complex relations) which is not being deeply examined in the literature and to find a suitable method of detecting the fake and self-operating accounts (known as a bot) of Instagram. To detect robotic accounts, to differentiate and check the effectiveness of the executed techniques; Precision, Recall, and F1 Score are valued in the evaluation metric. F1 score is more significant for the evaluation of performance because precision ignores the effect of FN, and recall ignores the impact of FP. F1 score observes both of them.

Now-a-days, the number of fake accounts on social media platforms (SMPs) are increasing and they are doing vicious work to harm and blackmail people. Regrettably, there is not enough research executed to expose fake profiles invented by humans, especially so on SMPs. Supervised machine learning, SVM, Naïve Bayes, and supervised. These features anticipated SMS spam to a great extent by operating supervised machine learning models like the random forest, decision trees, J48, logistic regression, and Naïve Bayes. It is compared from a one-class support vector machine model to a Naïve Bayes machine learning model. It shows how the one-class SVM outmatches the binary classification model when one of the classes is the minority. Their research also showed how bots are detected using clustering, a standard unsupervised machine learning method. The data is clustered based on a community with unsupervised machine learning. Unsupervised machine learning was favorably appealed by Gu el al, Wu et al, Yahyazadeh, and Abadi. As bots generally share similar features and characteristics and have the same motive, they can easily be detected with the help of clustering. All machine learning models were having very high accuracy, with the highest being 87.11 percent. These results could inaccurately specify that the supervised machine learning models are good predictors for identifying fraud by humans on SMPs.

2.2 Algorithm

2.2.1 K-Means clustering

Machine Learning algorithms are divided into two major groups- 1) Supervised and 2) Unsupervised. K-means clustering is one of the easiest and recommended unsupervised machine learning algorithms where the input data have an unlabeled response and make presumptions from dataset using only input vectors[9]. It performs the iteration that does the partition of the dataset into K pre-determined independent well-separated clusters where each data point belongs to one region[3].

It gathers the similarities to create data points and differentiate the clusters on the basis of dissimilarities.

The word ‘means’ in the K-means clustering algorithm does averaging of the data resulting in finding the centroid. So, the K-means clustering algorithm selects “K” number of centroids and then assigns each data point to the closest cluster for keeping the centroids size small as much as possible.It computes all of the centroids and iterates to find the optimal centroid.

K-means clustering is polished in various states in common sense life like scholarly execution, demonstrative frameworks, web search tools, remote sensor organizations, market division, report grouping, picture division, picture pressure, client division, examining the pattern on powerful information, vector quantization, highlight learning, bunch investigation, PC vision, stargazing, etc.

K-implies fundamentally manages its work by getting a significant instinct of the design of the information from the dataset that should be dissected. First of all, data need to be clustered and then predicted whether different model will be built for different subgroups or only one model is enough for all.

K-means is very fast, robust, easier to understand and robust. It is quicker in light of the fact that request for time intricacy is straight with the quantity of information. It would be quicker than Various leveled grouping on account of an enormous number of factors. In the event that k is little, it produces more tight bunches than Various leveled grouping. Its’ yield is emphatically affected by starting sources of info like number of bunches and request of information will firmly affect the last yield[4]. In any case, it requires the determination of the quantity of bunch focuses. It is exceptionally delicate to re-scaling and unfit to deal with the non-straight dataset, boisterous information and anomalies.

K-means uses various kinds of distance measures such as 1) Euclidean distance measure, 2) Manhattan distance measure 3) A squared Euclidean distance measure, 4) Cosine measure. Distance measure is finding the similarities between two data and determines the shape of the cluster.

First of all, we need to split the inputs into “K” clusters. Then we need to choose “K” random points as cluster centers which are called centroids. After that, we need to implement Euclidean distance by assigning each input to the closest cluster and identify new centroids by taking the average of the assigned points. Euclidean distance is the distance between the two points in Euclidean space. For example, if we have a point X and point Y, the Euclidean distance is an ordinary straight line between the two points. The formula of Euclidean distance will be: d (X, Y)=—X-Y—.

Finally, we need to repeat these processes until finding the convergence point.

The flowchart below shows how k-means clustering works:

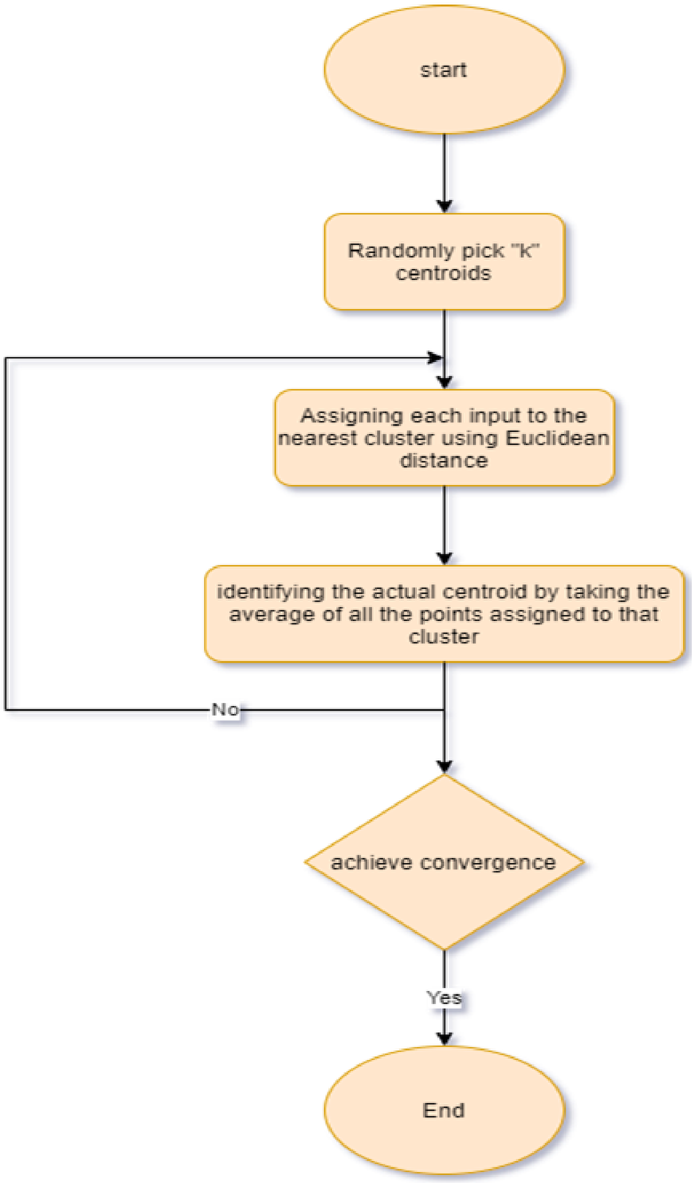

start
Randomly pick "k" centroids
Assigning each input to the nearest cluster using Euclidean distance
identifying the actual centroid by taking the average of all the points assigned to that cluster
No
achieve convergence
Yes
End

Figure 2.1: K-means algorithm workflow

2.2.2 Best First Search Algorithm

Best First Search algorithm is an informed search algorithm and traversal technique which uses both priority queue and heuristic function to find the most promising node[1]. The algorithm uses two lists for tracking the traversal and searching the graph space. They are 1) OPEN, 2) CLOSED. An 'OPEN' list keeps track of the current nodes available for traversal and 'CLOSED' list keeps track of the nodes that are already traversed. This algorithm traverses the shortest path first in the queue. The time complexity of the algorithm is O(n*logn).This algorithm can switch between BFS and DFS and takes the advantage of both. It is more efficient than DFS. But it has higher chances of getting stuck in a loop..

First of all, two empty lists are needed to be created; OPEN and CLOSED. Then we need to start from the initial node and put it in the ordered 'OPEN' list[2]. If the 'OPEN' list is empty, then exit the loop and return "False". Then we will select the first node in the OPEN list and move it to the CLOSED list. If N is a goal node, we will move the node to the closed list and exit the loop returning "True" otherwise we will expand node N to generate the next nodes and add all those to the OPEN list. Finally, we will reorder the nodes in the OPEN list in ascending order according to an evaluation function f(n).

2.2.3 OpenCV

OpenCV stands for Open Source Computer Vision Library. It is a machine learning software library and open-source computer vision. Intel developed it in the year 2000. This library mainly aims at problem solving time computer vision. In other words, it is a library which is used for Image Processing[5].

OpenCV is used as an image processing library in many computer vision real-time applications. It is mainly used to do all the operations related to Images like analyzing the data from the Camera of an embedded system or your computer or anything that captures images [1]. It is written in C++ and supported by various programming languages such as C++, Java, Python and can work in tandem with NumPy, Matplotlib and SciPy. Approved by more than one thousand contributors on GitHub, the computer vision library keeps strengthening for uncomplicated and effortless image processing.

OpenCV consists of more than 2500 algorithms and is used to build computer vision and machine learning applications. Some applications of OpenCV include read and write Images, detection of faces and their features, text recognition in images, developing augmented reality apps.
There are thousands of functions available in OpenCV. It mainly focuses on image processing, analysis, video capture, along with face detection and object identification. Added to that, the applications for OpenCV also cover areas such as segmentation and recognition, 2D and 3D feature toolkits, motion tracking, gesture recognition, image stitching, high dynamic range (HDR) imaging, augmented reality, and so on [14].

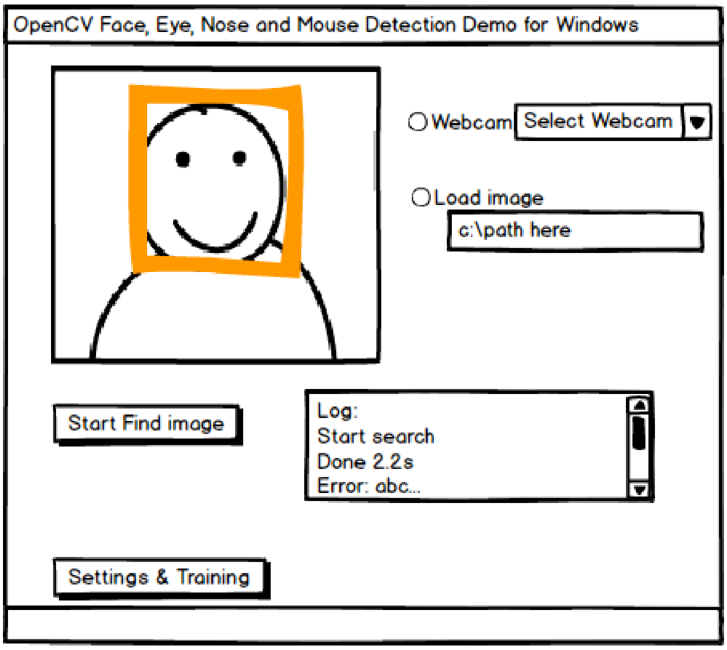
Figure 2.2: Features of OpenCV

Moreover, to support some of the previous application areas, a module with statistical machine learning functions is included. In OpenCV, images are transformed into multi-dimensional arrays, which dramatically clarify their manipulation. For instance, a grayscale image can be interpreted as a 2D array with pixels fluctuating from 0 to 255. Colored images are slightly more troublesome because it deals with 3D arrays where each pixel is rendered in three different color channels. By splitting the original image into its blue, green, and red components helps to grasp how the color layered structure works [13].

Some simple techniques are used to shape our images in our required format. Using OpenCV and support of inbuilt functions in OpenCV, it can perform the implementations by just writing a few codes. Using OpenCV, an image can be read by using the function cv2.imread ().For loading a colored image, cv2.imread color function is needed. Here any transparency of the image will be neglected. Cv2.imread grayscale function loads the image in grayscale form. Cv2.imread unchanged loads images in the alpha channel. cv2.imshow () function can be used to display an image in a window and the window will automatically fit the image size. cv2.waitKey () is a keyboard binding function. The function waits for stated milliseconds for any keyboard event. cv2.destroyAllWindows () directly destroys all the windows that were created. Also, cv2.imwrite () function is used to store an image. In the working directory, it will save the image in PNG format.

Moreover, in OpenCV, images can be changed between different color spaces. It can record a colored object in a video. Different geometric transformations to images like rotation, translation can be applied here. OpenCV can disciple images to binary images through global thresholding, Adaptive thresholding, Otsu's binarization, etc. The images can be blurred, filtered with custom kernels, find image gradients, edges, etc. How to use them for image blending, image pyramids and all about contours can be done by Opencv. All about histograms in OpenCV can be learned by it. An object in an image using Template Matching, detect lines and circles in an image can be searched through it[6].

In the field of real-time human and computer interaction and mobile robotics, OpenCV is anticipated in making computer vision accessible between programmers and users. Hence, the library comes with authorized code and hand-tuned assembly language in binary form that is optimized for Intel processors, with the purpose for the users to both learn from the library and make use of its achievements.



c:\path here

```
Log:
Start search
Done 2.2s
Error: abc...
```

Figure 2.3: OpenCV demo

Chapter 3

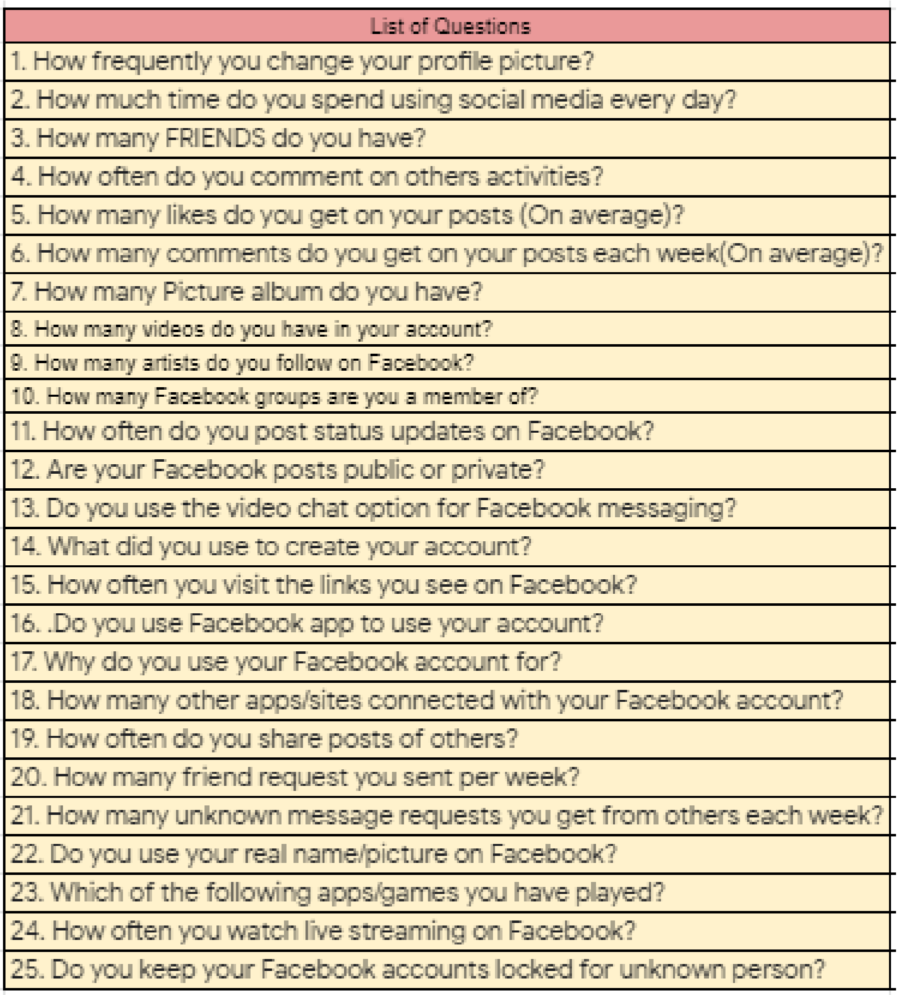
Proposed model

3.1 Dataset description

While social networking sites allow for networking across the region and improved business which is creating a positive force in the world today, some disadvantages come along as well. When a large chunk of personal, intimate, and secret information is posted on social networking sites, it simply leads to unfavorable fallout. Identity theft is the biggest concern that takes place over social networking. The act of stealing someone's personal information to use for personal gain and profit is called identity theft. If an individual tends to post a generous amount of private data on the internet such as their name, address, street no, phone no and birth date, etc, the straightforward passageway to this data makes identity theft pretty smooth. Not only does stealing an identity ruin the victim's name, but additionally skeptically alter their credit score which sometimes can take money from the victim. Therefore, an individual should take precautions to protect their identity in virtual life, the way they would do in real life.

That is why, to prevent identity theft, all the information about Facebook profiles, features, and privacy policies should be gathered. A dataset that contains many important questions was created by us. A survey form consisting of 25 questions helped us in collecting data from people. In our research work, we have released a dataset of Facebook profile information form and posted it throughout our social media and Facebook groups for research purposes, which we found quite valuable. As a result, in 3 to 4 hours we got almost 200 responses. After that, we waited for 3-4 weeks and got a total of 505 responses which were very useful for us.

Added that, the details which people provided in our survey form were completely protected considering their privacy. After data processing, all identifying information was encoded immediately. Names and identification numbers, email IDs were maintained on a protected local server which was available only for our teammates. After the last influx of data is processed the collected information will be destroyed instantly. The set of questions about privacy information, games they play, spending time on Facebook or software they use, and other labels will be released only after a considerable delay to ensure that individual's identities remain anonymous. To access any part of the dataset, destined users must need approval from the teammates.



List of Questions
1. How frequently you change your profile picture?
2. How much time do you spend using social media every day?
3. How many FRIENDS do you have?
4. How often do you comment on others activities?
5. How many likes do you get on your posts (On average)?
6. How many comments do you get on your posts each week(On average)?
7. How many Picture album do you have?
8. How many videos do you have in your account?
9. How many artists do you follow on Facebook?
10. How many Facebook groups are you a member of?
11. How often do you post status updates on Facebook?
12. Are your Facebook posts public or private?
13. Do you use the video chat option for Facebook messaging?
14. What did you use to create your account?
15. How often you visit the links you see on Facebook?
16. .Do you use Facebook app to use your account?
17. Why do you use your Facebook account for?
18. How many other apps/sites connected with your Facebook account?
19. How often do you share posts of others?
20. How many friend request you sent per week?
21. How many unknown message requests you get from others each week?
22. Do you use your real name/picture on Facebook?
23. Which of the following apps/games you have played?
24. How often you watch live streaming on Facebook?
25. Do you keep your Facebook accounts locked for unknown person?

Figure 3.1: List of Questions

3.1.1 Data preprocessing

As we mentioned, we kept collecting responses for a month. During this time, we tried to gather ideas for two clustering algorithms and some ideas on how to preprocess the dataset to implement it. When we received 505 responses, we then stopped collecting responses, and we started to work on the responses. Our initial idea was to assign every possible answer to all 25 questions in numerical values to run our dataset. Then, we opened a sheet and started to save all the possible answers from the form to assign values for the responses from the sheet. Moreover, in some questions, we kept blank options so that anyone can provide the answers by using their own words. While we checked the responses, we found many responses where people shared their views on those questions differently.

Along with the provided possible options, we also took the individual responses and assigned them numerical values. Here, we also got many responses where the answer and views of the people for most of the questions were the same. In this case, we assigned those answers with a unique numerical value as their views for the particular questions were similar.

For example, we have one question where we have asked why people used Facebook accounts. We got around 42 answers from the people where we only provided three options and kept blank options. In these responses, there were some answers where some answers were the same as mentioned earlier. For example, one said "meeting new people," one said "keeping in touch with friends," and one "networking." Here, the meaning of all three responses is the same, so we assigned these answers with one numerical value, which is 1. Also, there are some questions where we kept only two or three options to answer.

For instance, we can say there is a question where we have asked whether they keep their Facebook accounts locked for unknown people or not. We provided three options which are respectively "Yes," "No," and "Don't want to share." In this question, 249 people responded "No," 218 people responded "Yes," and 38 people responded, "Don't want to share." We also assigned unique numerical values for these three answers and kept notes for ourselves to find out the meaning of the given numerical values if needed. This is how we preprocessed the data so that we can use it for implementation.

3.2 Model Description

3.2.1 Dataset detection

Following the preprocessing of the data, our next task is to run the dataset in an algorithm to determine the ratio of false and true data from it. However, the data we've gathered is genuine, and we'll need to come up with a suitable algorithm for it. Upon researching, we read about some clustering algorithms that classify the data points into a number of regions such that data points in the same region are more likely to other data points in the same region and dissimilar to the data points in other regions. Since our purpose is to cluster the dataset into one group and gather all the data in one region to compare those data with some additional test data; our objective is to find out whether our system can distinguish fake or real under the aegis of deep learning.

Looking at the classification of clustering algorithms, we found an algorithm named K means algorithm which we found relatable to our research. It's an iterative algorithm that separates the dataset into the appropriate clusters, with related data types grouped according to the specifications. The value of K usually determines the number of clusters, but there can be more if necessary. However, we only need one cluster to store the data in one category for our purposes.

As previously mentioned, K denotes that the algorithm has a centroid point, and the data is processed based on the point, which is modified for each data measurement. The calculation is done based on the euclidean distance between two points and data stored simultaneously for every data. If we were to partition the data into two clusters, we can take two centroids for each cluster at first. Then for clustering, we can compare the distance between each centroid and the new data points to evaluate which case has the shortest distance between the centroid points and the new data points. The data is clustered under the same centroid point for the smaller one, and the centroid points are modified correctly by measuring the mean of the two data sets.

Our purpose is to cluster the whole dataset in one group as we have these data taken are real and we wanted to compare these data with some test data to analyze the accuracy of the detection process. We came up with an idea to automate this method using the k-means algorithm so that our algorithm can distinguish which data is true and which data is false. As previously mentioned, our dataset contains 25 questions, to which 505 responses have been given. We used 505 responses as our training data, with the first data points serving as our initial centroid stage. Then, we measured the euclidean distance for each piece of data and saved the value in an array. We have also modified the centroid points for each data point contained in the cluster simultaneously. The aim of storing the distance between each piece of data in an array is to determine the maximum distance between them.If we find the maximal distance after storing all of the data in a cluster, we will use it as a threshold. We will also provide the most recent centroids for each data point so centroid points for each data point keep updated.

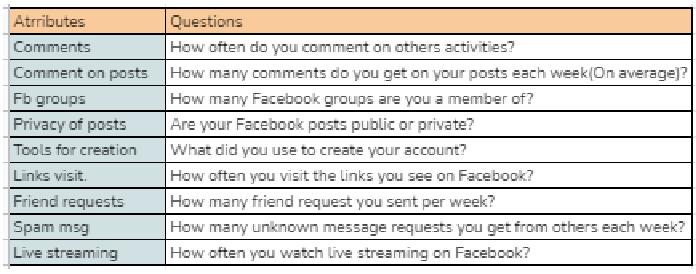
We checked this with data to see if the data printed were true or false after we had the most recent centroid and maximal distance for our training dataset. We used the data points as new centroid values and measured the difference between them and the most recent centroid from the test data. If we discover that the distance is greater than the threshold, we should determine that the data is fake; but, if the new distance is smaller than the threshold, we can decide that the data is genuine and stored in the cluster.

Since our attributes in the dataset are 25. Initially we have implemented the algorithm with all the attributes and compared it with the new data to identify whether the new data is real or not. Since we used 25 attributes, the sophistication of the algorithm and the program's runtime are both relatively high. This has also had an impact on the quality of the results received.

We used Weka, an open source platform that assists tools for data preprocessing and execution of many Machine Learning algorithms, to reduce the complexity and achieve a more effective performance. We needed to reduce the number of attributes, so we used Weka to search the modified attributes. There are three simple algorithms for selecting attributes in Weka. Among the three algorithms, there are two algorithms which we found helpful to minimize the attributes for our dataset which are as follows, best first algorithm and greedy stepwise algorithm. We initially tested the program with both attributes to see how it performed on our dataset.

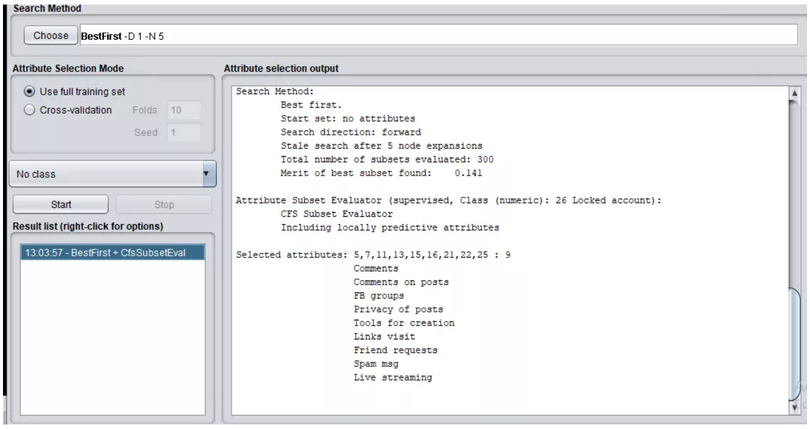
Upon running, Best first algorithm provided us nine attributes and depth wise algorithm provided us six attributes. We found one thing as common, the six attributes for greedy step wise, were also provided by the best first search algorithm. As for best first search algorithms, we got three more attributes and the six attributes from greedy step wise, so we chose the best first search algorithm to implement on our dataset for more efficiency.

The following are the attributes (questions) that we discovered while running the dataset using the best first search algorithm, as well as the detailed questions:



Attributes	Questions
Comments	How often do you comment on others activities?
Comment on posts	How many comments do you get on your posts each week(On average)?
Fb groups	How many Facebook groups are you a member of?
Privacy of posts	Are your Facebook posts public or private?
Tools for creation	What did you use to create your account?
Links visit.	How often you visit the links you see on Facebook?
Friend requests	How many friend request you sent per week?
Spam msg	How many unknown message requests you get from others each week?
Live streaming	How often you watch live streaming on Facebook?

Figure 3.2: Selected questions



Search Method

BestFirst -D 1 -N 5

Attribute Selection Mode

● Use full training set
○ Cross-validation Folds 10
Seed 1

No class

Result list (right-click for options)

13:03:57 - BestFirst + CfsSubsetEval

Attribute selection output

```
Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 300
        Merit of best subset found:    0.141

Attribute Subset Evaluator (supervised, Class (numeric): 26 Locked account):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 5,7,11,13,15,16,21,22,25 : 9
                     Comments
                     Comments on posts
                     FB groups
                     Privacy of posts
                     Tools for creation
                     Links visit
                     Friend requests
                     Spam msg
                     Live streaming
```

Figure 3.3: Selected attributes after implementing weka

After that, we took the nine attributes and created a new dataset to run on our algorithm. We also added a start index and end index on our program to count also any specific index and check with our algorithm. It would help us to check with limited attributes and find out the centroids and euclidean distance respectively.

3.2.2 Image processing and OTP

One of the latest issues of the automation industry is Computer vision.Computer vision is the sector of computer science that focuses on empowering machineries to determine and operate objects in images and videos the exact way that humans do.Facial recognition,robotics, self-driving cars all depend upon computer vision to function. The central and fundamental element of computer vision is face recognition. Face recognition is the ability of a system or software to recognize and acknowledge what an image represents and appears for.

Before moving to our task which is interconnected to images, it is important and compulsory to form the image to be suitable as input data for the next procedure. In this project, our main attention is on image processing, specifically face detection. Face detection is a part of image processing that manipulates machine learning to discover and recognize faces in images. It is one of the most important fields of image processing in this technological era.

Now-a-days, many useful libraries and projects have been generated and these can assistusdecode image processing problems using machine learning.

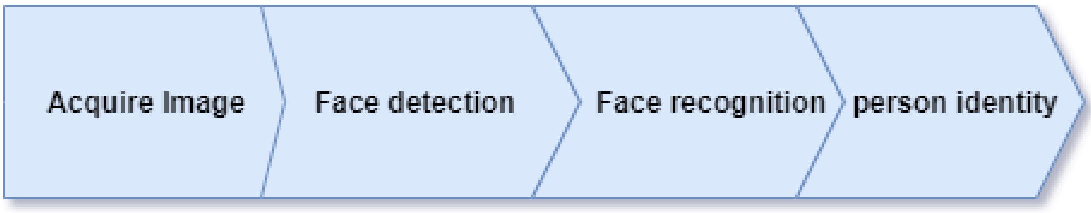


Figure 3.4: Steps of face recognition system applications

In this project, we have a list of libraries which we have used to detect and recognize human faces in the image. i) OpenCV ii) cv2 iii) dlib iv) face recognition v)numpy and vi) OS
First of all, when a user inputs an image, that image is loaded from the specified file with the help of cv2.imread() method. However, if the system cannot read (because of missing file, improper permissions, unsupported or invalid format) the image then this method will return an empty matrix then we will check the image contains an object such as birds, airplanes, table, horse, chair, cow, dining table, bus, motorbike, dog, sheep, sofa etc. or includes human face. This detection will be done by face detection model which includes a pre-trained Caffe deep learning model offered by OpenCV to detect faces. This model discovers and confines faces in an image.

After detecting the image, if it is about an object, the user will be denied further access because the user must put the image of human faces. Afterwards, if the image is of a human face, the server will try to find a match in the dataset of images. This

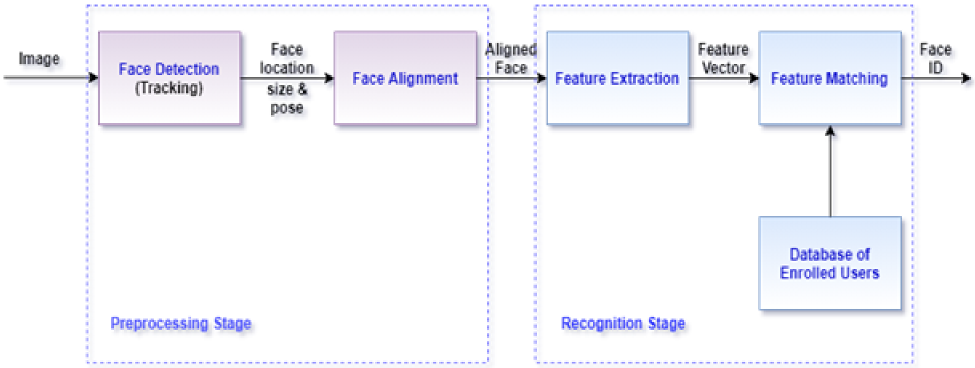
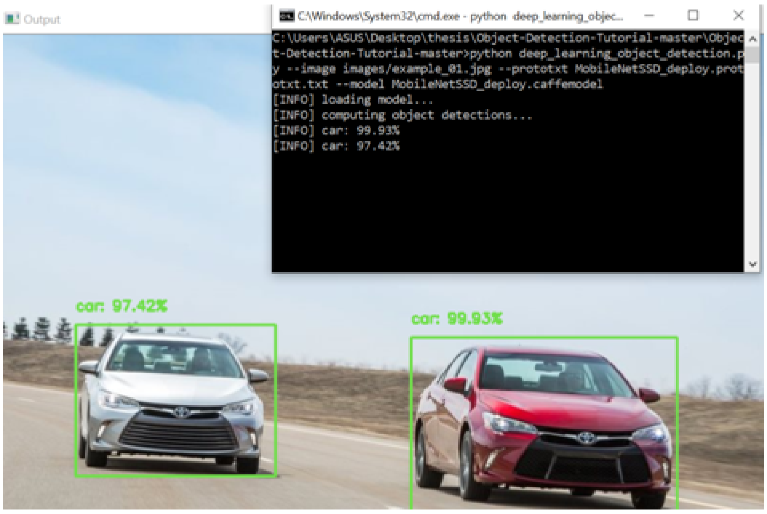


Figure 3.5: Face Recognition processing flow)

operation will be executed by a face recognition library which is the latest trend in Machine Learning techniques. By using a face recognition library from Python or the command line, faces can easily be recognized and operated.It comes up with a simple face recognition command line tool. It finds absolutely all the faces that are shown in a picture and can easily locate and outline of each person's eyes, nose, mouth and chin and recognize who pops up in each photo.

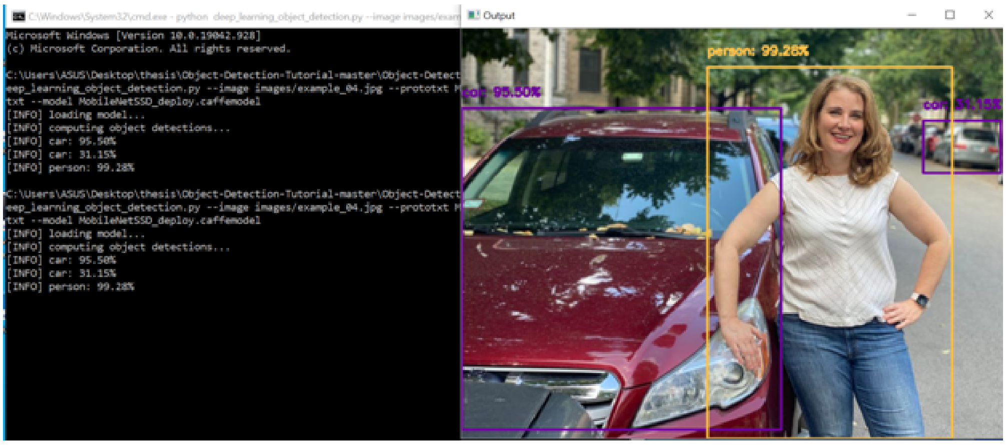


```
C:\Users\ASUS\Desktop\thesis\Object-Detection-Tutorial-master\Objec
t-Detection-Tutorial-master>python deep_learning_object_detection.p
y --image images/example_01.jpg --prototxt MobileNetSSD_deploy.prot
otxt.txt --model MobileNetSSD_deploy.caffemodel
[INFO] loading model...
[INFO] computing object detections...
[INFO] car: 99.93%
[INFO] car: 97.42%
```

Figure 3.6: Object detection

Additionally, Dlib, a powerful library, will be operating, having a large assumption in image processing which is similar to OpenCV. Besides this, there is a module named face recognition which is provided by Dlib.

Thereafter, the user will be granted access depending on whether the image matches any other image in the database. If matched then an OTP will be sent to the person's email address that the image matched with to authenticate the user .A one-time secret key (OTP) consequently produced numeric or alphanumeric series of characters that verifies the client for a solitary exchange or login session.The OTP include restricts a few kinds of wholesale fraud by not permitting any caught client data/picture to be utilized a subsequent time.We will be using on-demand OTPs which are not reusable and will expire after being used. For instance let's assume



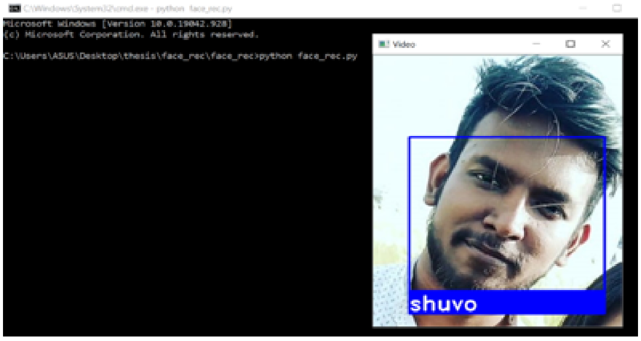
C:\Windows\System32\cmd.exe - python deep_learning_object_detection.py --image images/exam
Output
person: 99.28%
car: 95.50%
car: 31.15%

```
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Desktop\thesis\Object-Detection-Tutorial-master\Object-Detect
eep_learning_object_detection.py --image images/example_04.jpg --prototxt M
txt --model MobileNetSSD_deploy.caffemodel
[INFO] loading model...
[INFO] computing object detections...
[INFO] car: 95.50%
[INFO] car: 31.15%
[INFO] person: 99.28%

C:\Users\ASUS\Desktop\thesis\Object-Detection-Tutorial-master\Object-Detect
eep_learning_object_detection.py --image images/example_04.jpg --prototxt M
txt --model MobileNetSSD_deploy.caffemodel
[INFO] loading model...
[INFO] computing object detections...
[INFO] car: 95.50%
[INFO] car: 31.15%
[INFO] person: 99.28%
```

Figure 3.7: Detecting Human face



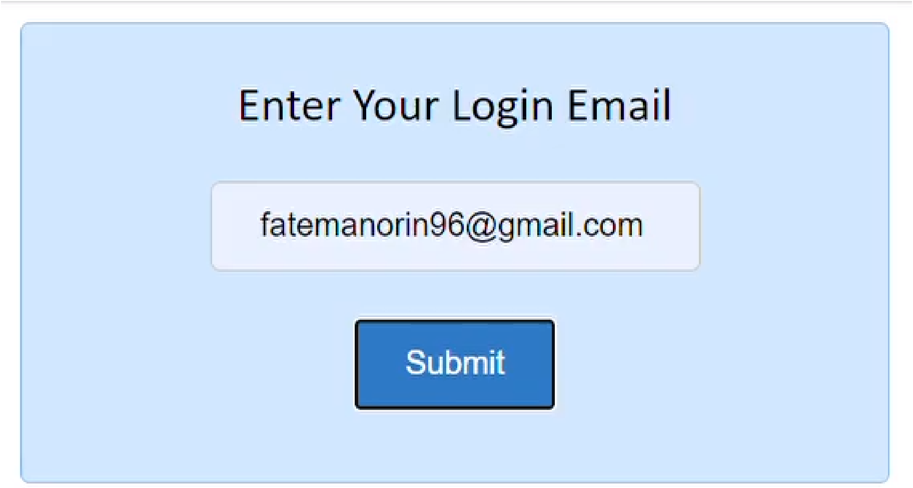
```
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Desktop\thesis\face_rec\face_rec>python face_rec.py
```

Figure 3.8: Recognizing face using OpenCV

the image matches with the image of person X. An OTP will be sent to that person's registered Email address. There can be a scenario where he/she wants to have a new account. It will solely depend on that person.According to their will they will be able to keep continuing opening the account.

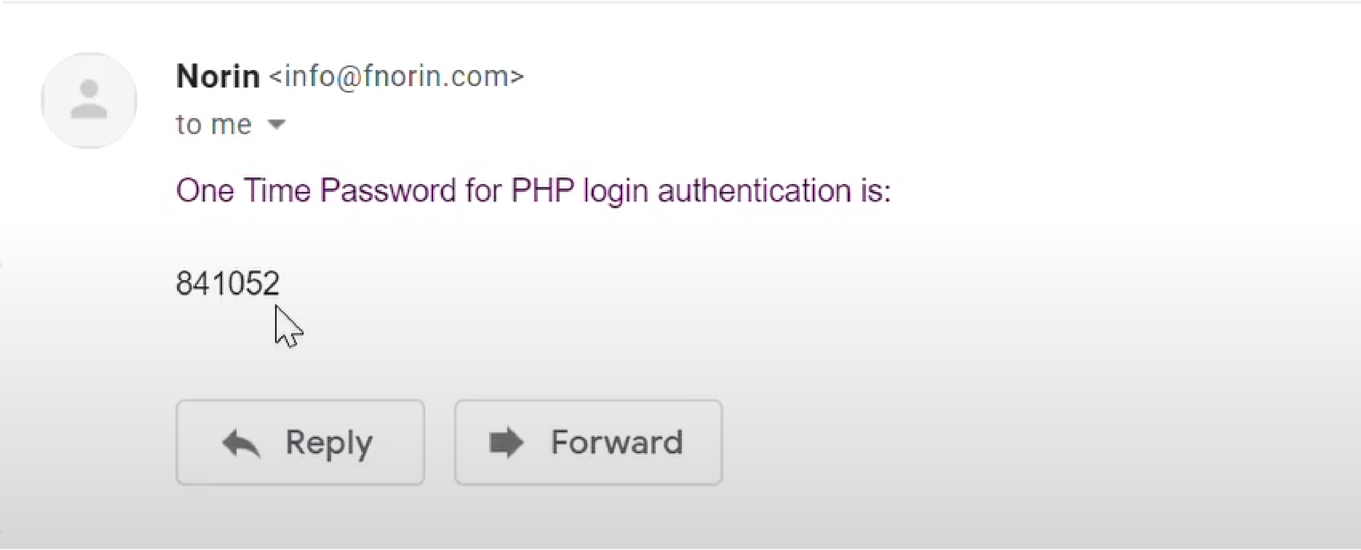
The client recovers the OTP and adds it into the brief to confirm its character and acquire access. Be that as it may, if some intruder is attempting to open a record with another person's picture effectively in our information base, he will be halted abruptly. As without the OTP which is simply accessible to the genuine holder of the record, he won't continue.



Enter Your Login Email

fatemanorin96@gmail.com

Figure 3.9: Login interface



Norin <info@fnorin.com>

to me

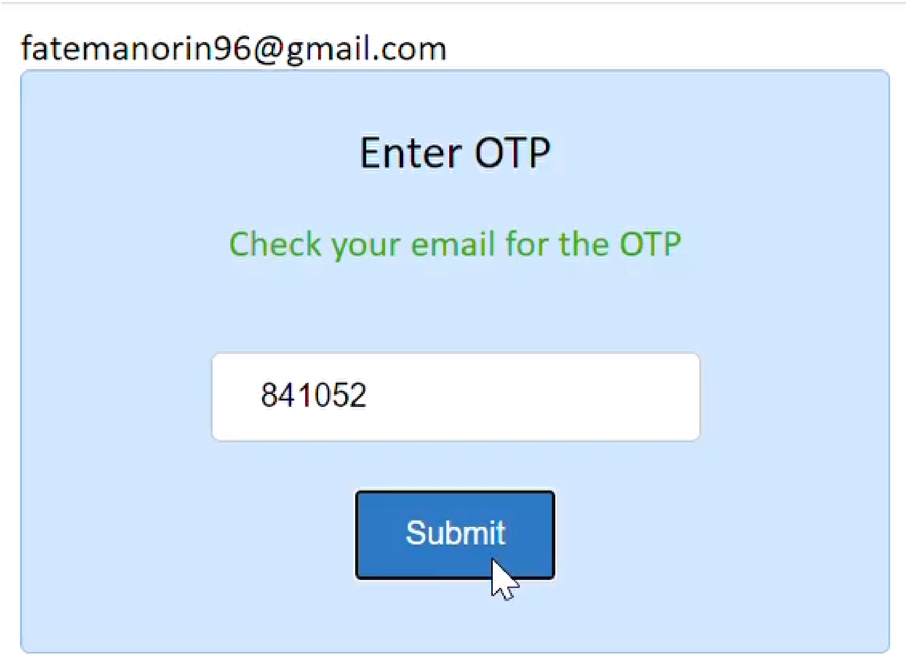
One Time Password for PHP login authentication is:

841052

Figure 3.10: One time password

On the other hand, if the image does not match with the images from dataset, access will be approved without any further complication. Numerous individuals as of now have their email open on their PC or their cell phones promptly close by, so getting to email won't be an issue. Email's ubiquity means that one-time passwords are convenient to use. This defensive mechanism is a crying need for personal security. If an adversary is trying to open an account using someone else's identity, he/she will no longer be successful. But if somebody wants to have multiple accounts, this process will maintain their privacy and allow them to do so.

To sum up, many exciting and reasonably robust applications have been created in face detection. As most advanced algorithms can also apply to other various domains, it has a more extensive and more helpful impression than recognizing faces in pictures alone. Future investigation will focus on improving recognition exactness, web based preparation of such locators, and novel applications.



fatemanorin96@gmail.com

Enter OTP

Check your email for the OTP

841052

Figure 3.11: Input OTP from email

Chapter 4

Experimentation

4.1 Algorithm experimentation

For this paper, we have implemented the k-means algorithm to train our dataset to detect the fake and real data from our dataset. We have taken the value of centroid K=1 as we were not required to make more than one cluster as our purpose is to gather our training data in one cluster so that we can compare the data with our test dataset. Then, the first data was assigned as the centroid point. Next, we have calculated the euclidean distance for each data point respectively. As the distances were calculated, the centroid point was also updated accordingly. When the distance was measured between two data points, the centroid point got updated by the average of the two data points. Moreover, the distances were stored in an array as we had to print out the maximum distances after the checks were done. After we got the maximum distances and the final centroid point for each attribute, we then opted to check the data with the test dataset to find out whether our algorithm was being able to detect fake accounts or not. From the test data, we have taken all data consecutively and measured the euclidean distance between the latest centroid data found from training data and these data. Finally, after comparing, if the distance was more than the maximum distance, it printed as Fake and if the distance was less than the maximum distance, the output was Real.

4.2 Workflow of Image processing

For this paper. We have also implemented image processing and One time password to prevent users from creating fake accounts.

First of all, the user will input an image. If the image is of any human face then the system will detect that image using opencv and dlib libraries. Afterwards, the face will be cropped using a numpy library. Lastly, the face will be recognized using face recognition library. If the face gets matched with any other picture from our dataset, the user will not have any access or else he/she will have further access.

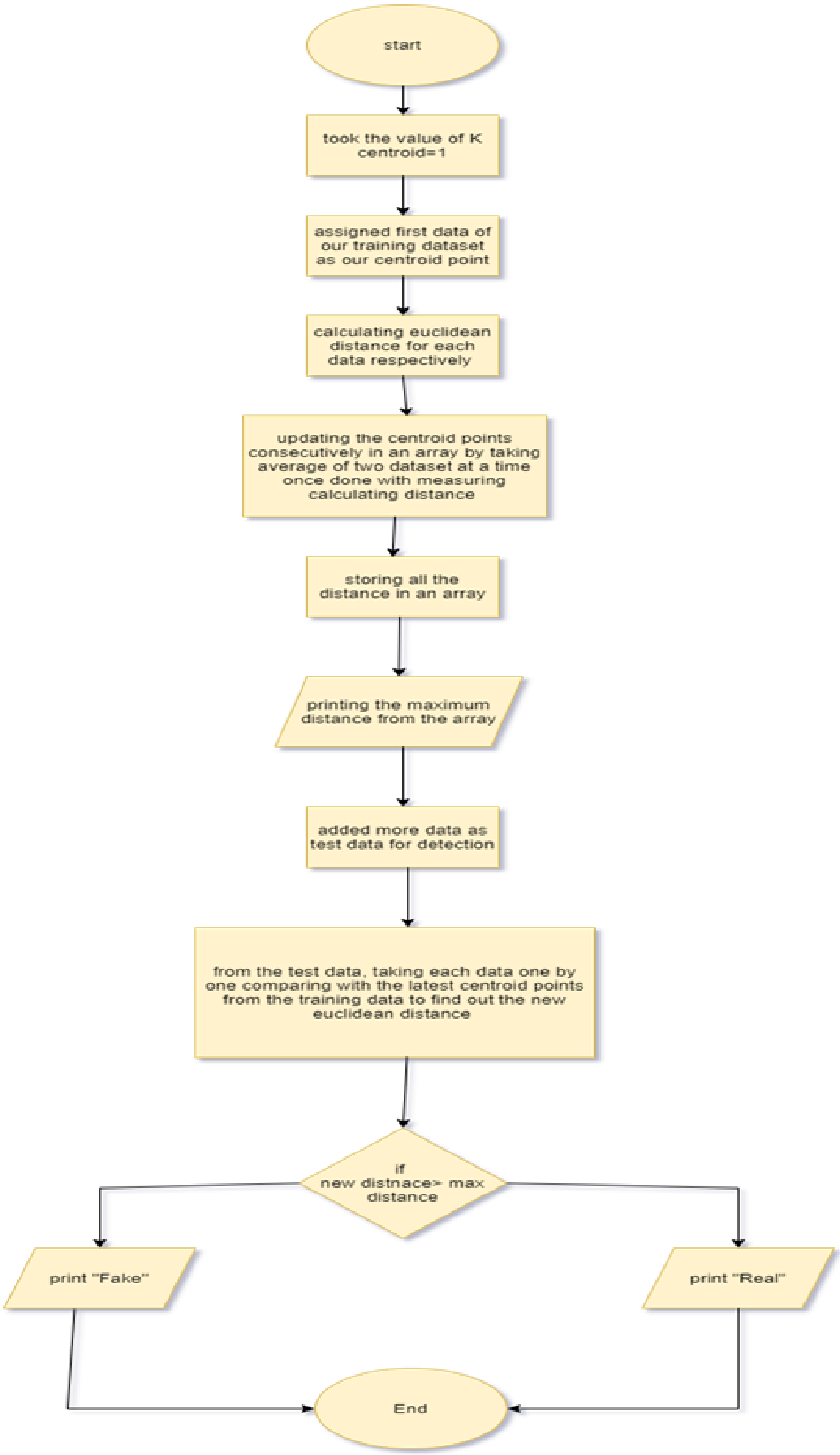


Figure 4.1: Workflow of fake account detection

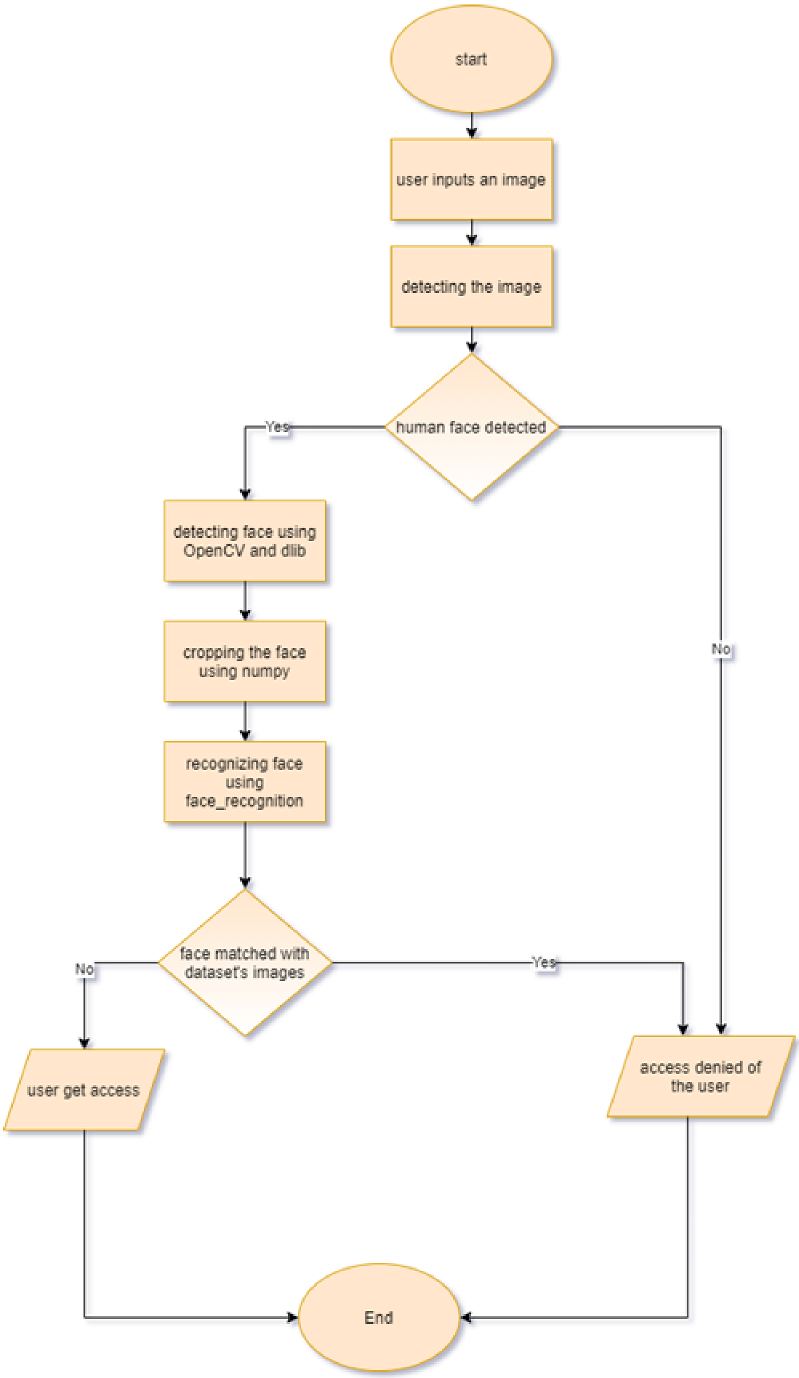

start
user inputs an image
detecting the image
human face detected
Yes
No
detecting face using OpenCV and dlib
cropping the face using numpy
recognizing face using face_recognition
face matched with dataset's images
No
Yes
user get access
access denied of the user
End

Figure 4.2: Workflow of image processing

4.3 Workflow of implementation of OTP

If the image gets matched with any image from the dataset then an OTP will be sent to the email account of the person with whom the image matched. So, if the person wants to use that OTP then the user will have further access or else the access will be denied.
A flowchart below of OTP (One Time Password) is added below to show how OTP works in our system-

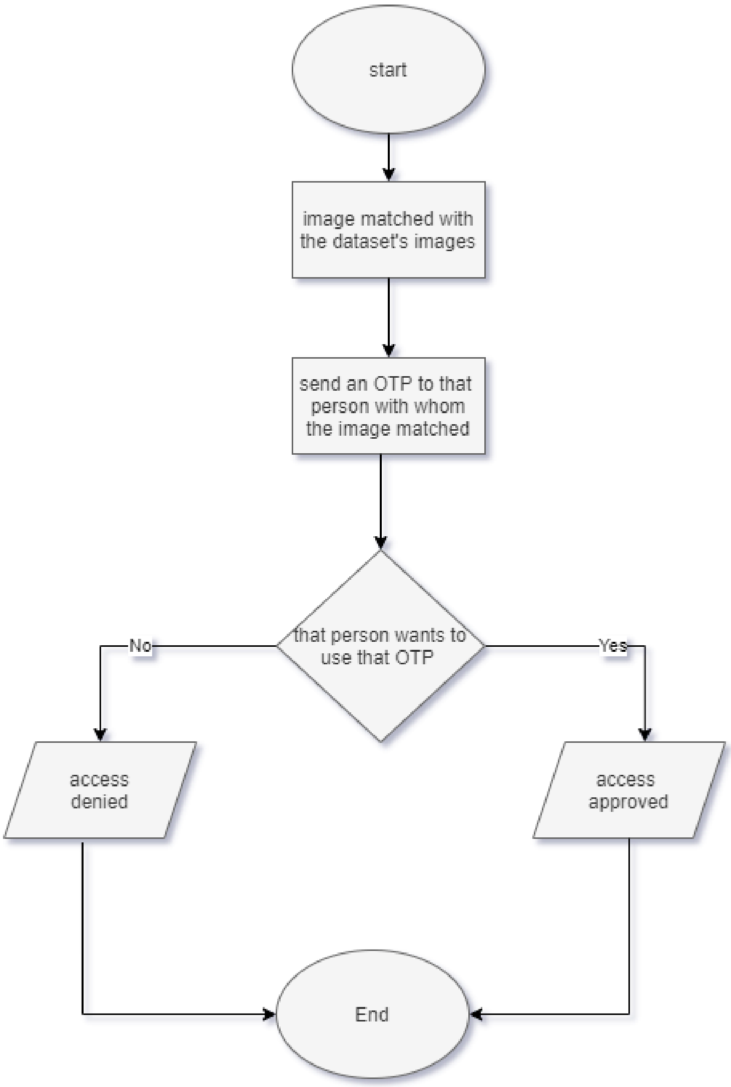

start
image matched with the dataset's images
send an OTP to that person with whom the image matched
that person wants to use that OTP
No
Yes
access denied
access approved
End

Figure 4.3: Workflow of OTP

Chapter 5

Result Analysis

5.1 Analysis of Dataset detection

Once we have completed the whole process, we have tried to determine how accurately our algorithm works. Since there is no available dataset, we have created a dataset for this research. Also, comparative analysis is not possible for us to do as there is no work on this dataset.

To find out the percentage, we have added some fabricated data in our dataset to check on our algorithm to measure the accuracy of our research.

As we know, for binary classification, accuracy can be calculated in terms of positive and negative data.

$$\text{Accuracy } = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{5.1}$$

Here TP = True positive which means the real data we have in our dataset. If we compared this data within themselves. Then the result should have come as real and this is called true positive.

TN= True Negative which indicates basically to the real data but it is printed as fake data. We have checked with some data which was the real data but when we compared it with our training dataset, it reflected as fake data.

FP= False Positive which identifies the fake data as real. It means there were some fake data which was identified as real data.

FN= False Negative which is mainly fake data. The fake data system indicated as fake data is known as false negative.

Here. to find out the accuracy, we have added a few more data as test data so that the above mentioned equation can calculate the accuracy. We assigned 480 data as our training dataset. Then, we fabricated 25 data as True positive and 36 data as True Negative from the test data. After that, we made 10 False Positive data and 10 False Negative data.

If we calculated these data in the equation, we found the accuracy is 75.30 percent ((25+36)/(25+36+10+10)).

As the size of our dataset is not bigger, the accuracy we found is lesser. However, if we have more data, the accuracy will be increased. For larger dataset, the accuracy will rise and might be some more than the mentioned one.

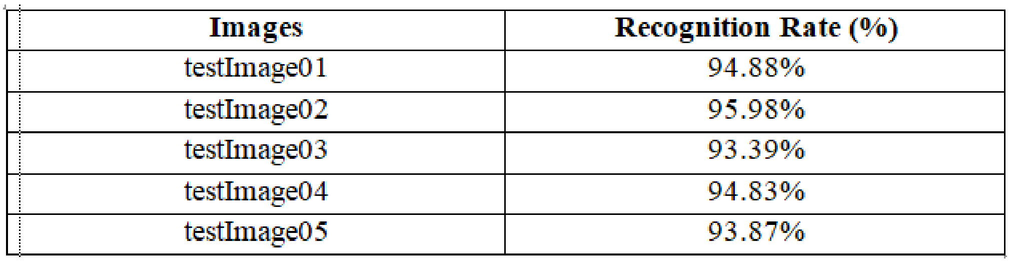
Lastly, we have the accuracy of 75.30 percent of our work. The larger dataset might increase the accuracy as we have more data to check with our algorithm and we will be able to find out how efficiently our system can detect fake and real data.

We would also like to do time to time evaluation by collecting the data of existing accounts. Our plan is to check and run the data in our algorithm so that we can eliminate the fake profiles from the existing accounts.

5.2 Analysis of Image processing

Moreover, to upgrade the execution of human face detection, we are planning to improve many things such as color processing, edge detection, etc can be added.

Here is the table below of Human Face detection rate-



Images	Recognition Rate (%)
testImage01	94.88%
testImage02	95.98%
testImage03	93.39%
testImage04	94.83%
testImage05	93.87%

Figure 5.1: Human Face detection rate

Human Face detection is based on five cases and the average time of this detection is 2-4 seconds. This procedure has been executed in the Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, 2712 Mhz, 2 Core(s), 4 Logical Processor(s).

In the future, our goal is to acquire more accurate and precise result by using advanced methodologies and libraries as face detection.

Chapter 6

Conclusion

In order to detect fake accounts, a work plan has been planned and developed for our proposed solution. At the outset, we have gathered and preprocessed data in order to prepare our dataset. Then we choose the best algorithm to distinguish between true and false accounts. To identify fake accounts, we must cluster the relevant data. We choose K-means clustering for implementation because it is more accurate than the other algorithms for our proposed solution. As a result, we will put the algorithm into action and evaluate the results and accuracy rate. Upon running this algorithm in our dataset, we found the accuracy is 75.30 percent. If we have a larger dataset, our accuracy will be increased.

Then, we will run the image processing by gathering images and distinguishing between true and false profiles. Finally, after matching the user's data with the database, a one-time password would be submitted to them for authentication. By doing so, we will be able to identify false accounts and avoid identity fraud. We are hoping that our model will help to reduce the number of fake accounts and the vast amount of trouble that can be caused by these accounts.

Our future plan is to add the image processing and OTP together in a website. As our motive is to stop the user from creating fake accounts and also we wanted that none of the genuine users are affected. There is a chance that genuine users might want to create more than one account. To make this happen, OTP is generated and if any user has already an account, wants to create one more account, firstly the images of the user will be checked and then an OTP will be sent to the user's first created account's email address for verification. By doing this, no fake accounts can not be created.

In the future, we are also hoping to add more features to help us detect the image. In addition to that, we will also send the OTP to the phone number of the user so that it will be easier for him to obtain the password with more ease. Finally, we will gradually apply our model to various other social media such as Linkedin, Instagram, and Twitter.

Bibliography

[1] H. Berliner, "The b* tree search algorithm: A best-first proof procedure," in *Readings in Artificial Intelligence*, Elsevier, 1981, pp. 79–87.

[2] R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of a," *Journal of the ACM (JACM)*, vol. 32, no. 3, pp. 505–536, 1985.

[3] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[4] M. Yedla, S. R. Pathakota, and T. Srinivasa, "Enhancing k-means clustering algorithm with improved initial center," *International Journal of computer science and information technologies*, vol. 1, no. 2, pp. 121–125, 2010.

[5] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to opencv," in *2012 proceedings of the 35th international convention MIPRO*, IEEE, 2012, pp. 1725–1730.

[6] A. Mordvintsev and K. Abid, "Opencv-python tutorials documentation," *Obtenido de https://media. readthedocs. org/pdf/opencv-python-tutroals/latest/opencv-python-tutroals. pdf*, 2014.

[7] A. Gupta and R. Kaushal, "Towards detecting fake user accounts in facebook," in *2017 ISEA Asia Security and Privacy (ISEASP)*, IEEE, 2017, pp. 1–6.

[8] Y.-C. Chen and S. F. Wu, "Fakebuster: A robust fake account detection by activity analysis," in *2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, IEEE, 2018, pp. 108–110.

[9] D. M. J. Garbade and J. Michael, "Understanding k-means clustering in machine learning, 2018," *URL https://towardsdatascience. com/understanding-k-means-clustering-in-machine-learning-686c67336aal*, 2018.

[10] M. Mohammadrezaei, M. E. Shiri, and A. M. Rahmani, "Identifying fake accounts on social networks based on graph analysis and classification algorithms," *Security and Communication Networks*, vol. 2018, 2018.

[11] F. C. Akyon and M. E. Kalfaoglu, "Instagram fake and automated account detection," in *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, 2019, pp. 1–7.

[12] M. Smruthi and N. Harini, "A hybrid scheme for detecting fake accounts in facebook," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 7, no. 5S3, 2019.

[13] A. Hachcham, "Exploring image processing techniques — opencv," 2020.

[14] F. Llorens, F. J. Mora, M. Pujol, R. Rizo, and C. Villagrá, "Working with opencv and intel image processing libraries. processing image data tools.,"