Pakistan Journal of Engineering and Technology, PakJET
ISSN (p): 2664-2042, ISSN (e): 2664-2050
Volume: 5, Number: 2, Pages: 93- 99, Year: 2022

Investigating and Auditing Through Facial Recognition Using LBPH Algorithm

Samar Fatima[1*], Rehan Inam Qureshi[1], Muhammad Hassan Khan[3], Ali Akbar Siddique[2], Manahil Athar[1], and Muhammad Hamza[1]

[1]Department of Software Engineering, Sir Syed University of Engineering & Technology, Pakistan.
[2]Department of Telecommunication Engineering, Sir Syed University of Engineering & Technology, Pakistan
[3] School of Computing and Mathematics, Charles Sturt University Sydney, Australia

Corresponding author: Samar Fatima (Samarfatima072@gmail.com)

Received: 5/12/2022, Revised: 01/04/2022, Accepted: 15/05/2022

Abstract- **Technology is constantly advancing, and many individuals submit various video material to social networking websites such as YouTube or Facebook. Since it has become a source of income for many individuals across the world, it is becoming increasingly vital to utilize it in situations when you need to discover a specific person in several video recordings. Another option is to manually go through each video and try to discover the individual segment in order to extract it. Manual searching can take a long time, and it's practically difficult to find a specific video in which a person appears. A person can only focus for 20 to 30 minutes on average to recognize or identify the person in the video, and a video stream may take much longer. Due to the huge quantity of data gathered in the multimedia application, such as videos, a human conducting a video search manually may be difficult to do so properly in such cases. It is critical to automate the procedure in order to eliminate human error and the time it takes to identify the individual in the video footage. Given its popularity and use in applications ranging from our mobile phone games to high-end computers for future forecasting, artificial intelligence may be used to address many difficulties, including this one. In this work, a pre-trained facial recognition classifier known as the Linear Binary Pattern Histogram (LBPH) is utilized to recognize a person in video clips and provide recorded proof of each video in which a person appears, along with the time stamp of his or her visibility. Here, a method is proposed for identifying and tracking down a missing individual utilizing massive video data and Artificial Intelligence without the need for human participation.**

***Index Terms*-- Investigation, Distributed Processing, Face Detection, Local Binary Pattern Histogram (LBPH), Face Recognizer Algorithm.**

I. INTRODUCTION

People all around the world submit video material that surpasses broadband capacity in the age of social media, making it increasingly difficult to identify the desired video clip. Most individuals collect video clips to examine and comment on, however, the required video clip may not have all of the information or images of the person to be searched. Finding persons in video clips has been easier since the civilization and emergence of numerous cultures, as well as technological improvement. These videos may be widely used for a variety of purposes, such as searching for a wanted individual or extracting a clip of a certain individual to document his behavior, etc. By introducing technology into these scenarios, they may become more useful and less time-consuming.

The most common strategy is to use already installed cameras to broadcast continuous video information from remote areas in order to investigate and locate the individual without recording a video in real-time [1], but such a system may prove to be extremely expensive due to the enormous computation power required. Basically, every community in the world has put cameras on almost every site, and everyone carries video capturing equipment in the form of a mobile phone, thus there is little possibility of missing any second of a video stream that may contain critical information [2]. When a person is asked to look for someone in the desired video clips, he will refer to the videos acquired from the cameras deployed remotely everywhere in these times. This information might be especially useful in an inquiry if the individual's face is captured in the tape and the person is then sought in other camera recordings [3]. The issue here is that obtaining the individual's face is far easier than searching for the person in the video footage manually [4]. So, the person's face may be collected from the film many times in a single video stream, or he could be in all of the acquired video footage; in any event, it is critical to properly inspect all clips before documenting their activity. So, the face of the person could be acquired from the footage multiple times in a single video stream, or it is possible that he may appear in all the acquired video footage, in any case, it is crucial to thoroughly examine all clips before documenting this activity. before the crime scene as well and then a police department can identify the perpetrator. Finding someone from a live video stream, on the other hand, is exceedingly difficult and needs hours of waiting. Even then, it is impossible to detect a person; even a brief twitch of an eye might cause a piece of critical

evidence to be missed [5]. Because video cameras are used to cover a large region, they create a large volume of video data for an observer to deal with. Detecting the intended person's clip from such a large number of data might take much longer [6].
Camera operators, for example, are ineffective for applications such as this because they can only contemplate a video for 20 to 30 minutes at a time. After that period of time, their efficiency declines and they lose concentration, which may cause them to overlook important information about the individual they are looking for [7]. But if a break is given to them after every 20, 30 minutes then it will take a lot more time than before. For solving this problem, it is crucial to eliminate human intervention from this process so that the human-related error could be eliminated, and time consumption could be decreased [8].

II. BACKGROUND

Digital investigations have existed since the 1970s. Room for improvement never ceases. Many scholars have gone into tracking the re-occurrence of identical persons in large videos that might be months or days old [9]. It is cumbersome for organizations that are short on time. This paper intends to facilitate better than the existing person detection and system.
Our focus is to identify the person efficiently and recognize them. Also, the result is to provide the user promptly with the time of re-appearance. For instance, the user could be anyone who is assigned to look for a particular event look at the time generated in the report, and view the activity of that specific person for inquiring more into the matter.
To detect a person, we first identify if the appearing body is a person, then recognize the person using face detection and recognition, and then the report generation takes place.

III. METHODOLOGY

The problem of human error in real-time video vigilance investigation could be solved by using artificial intelligence, which is being used in every area for providing solutions. The solutions provided by artificial intelligence have been used in many areas including weather forecast, cybersecurity, healthcare, human resource management, logistics, supply chain, and many more [10][11]. The face and person recognition of artificial intelligence could be used for solving this problem as well [12].
The system we are presenting here in this research paper includes some steps such as detecting a person first and then detecting the face of the person and the block diagram of the proposed model is displayed in FIGURE 1. These two steps help in reducing fraud in this area. It will help the system not to detect a picture as a person in the video. As we are working on video data acquired from the cameras, there could be posters on the walls of the wanted person and the system could also detect the face in the poster as a person. For reducing these kinds of errors, the two techniques are combined. For person and face detection, Haar Cascade Classifier could be used [13] and for face recognition purposes LBPH algorithm could be used [14]. These algorithms work great and provide good efficiency when used together [15].
Then the main feature of the system, face recognition will work on the detected face to determine if the face detected in the video footage at this point is of the wanted person or not. After the face has been recognized as a missing person, then the tracking of the person will start, and it will continue until the algorithm scrutinizes the whole footage and document the time stamp of the detected person. This will tell us the total time the person has been seen in the video in one instance. This will also stop the person and face detection and face recognition algorithms from confusing the same person as two individuals at the same instance.
After all this work has been performed, all the instances when the person to be identified has been seen in the video will be recorded in the file which will be generated after the completion of video processing. The result or output file which will be generated in the end will reveal the timings of the video in which the person has been seen. This will help the police or concerned department in the investigations where video footage is included, and this will also reduce the time of the investigation eventually helping in reaching the culprit.

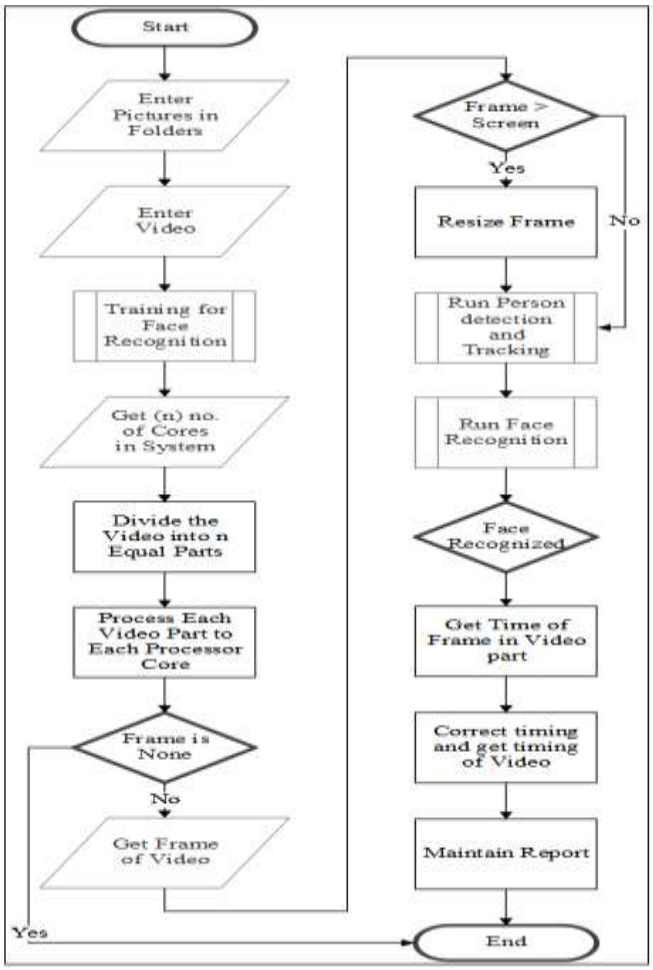

FIGURE 1. System Block diagram

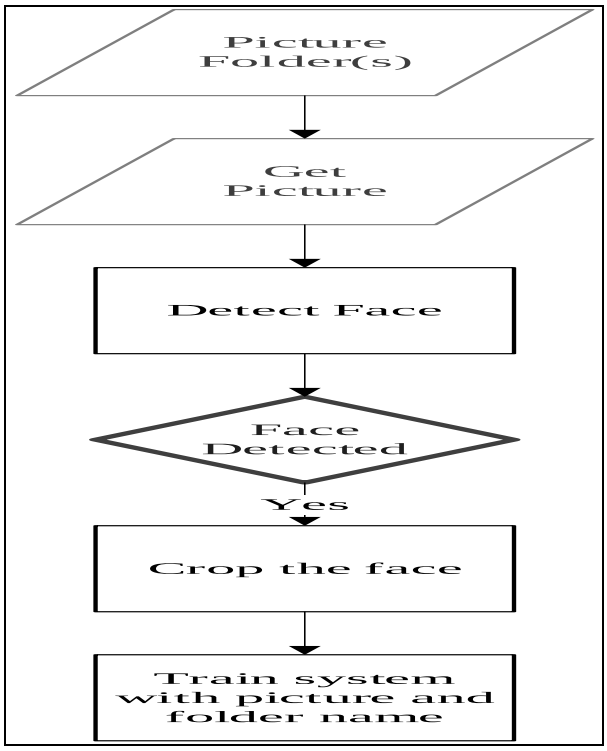

FIGURE 2. Subsystem (Training for Face Recognition) Flow

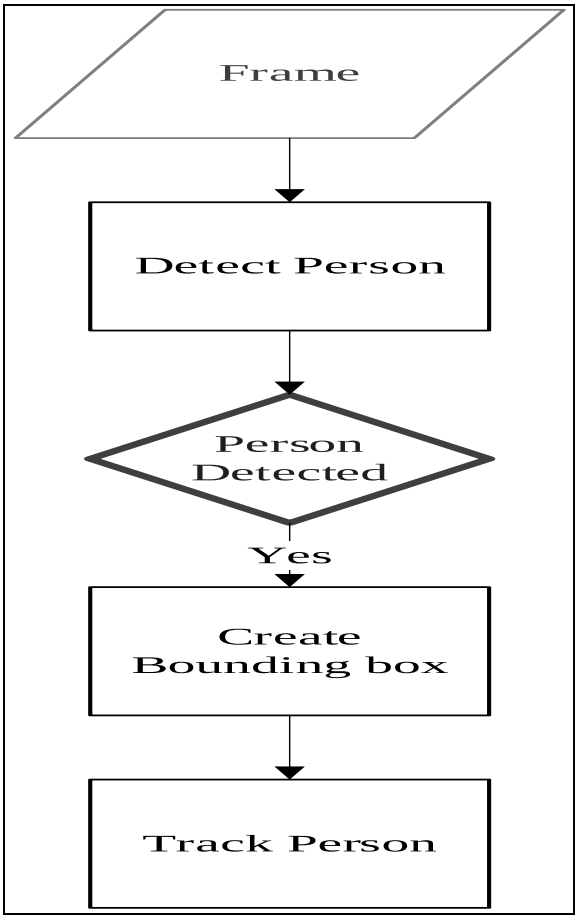

FIGURE 3. Subsystem (Run Person Detection and Tracking) Flow

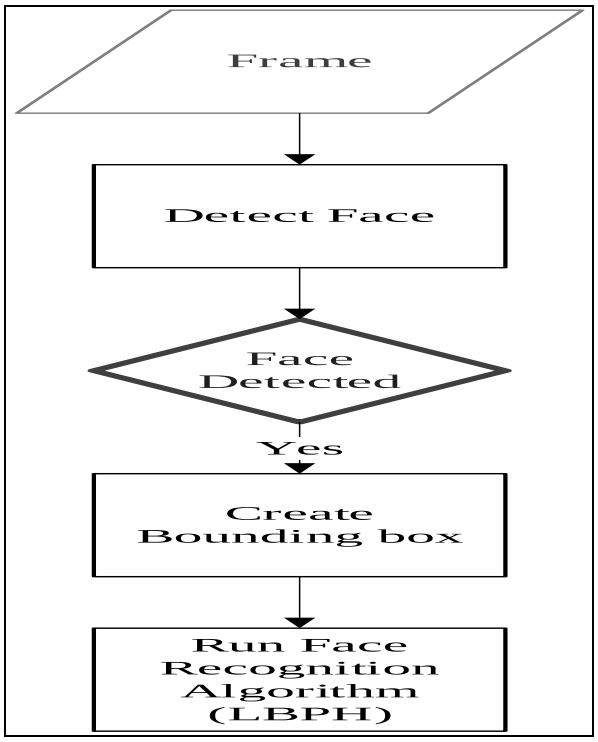

FIGURE 4. Subsystem (Run Face Recognition) Flow

A. Haar CASCADE CLASSIFIER FOR FACE DETECTION

Haar cascade is a machine learning-based object detection algorithm. In this algorithm, a lot of positive and negative images are used for training purposes.

1) Positive Images: The images accommodate the object that the classifier is supposed to identify.
2) Negative Images: The images which do not accommodate the object that the classifier is supposed to identify.

Haar Cascade classifier could be used to detect any kind of object given that you have the XML file of the object you want to detect. The XML file for the respective object could also be created by oneself. In our system, the Haar Cascade classifier is used to detect the human face [16].

B. HOG DESCRIPTOR AND SVM FOR PEDESTRIAN DETECTION

HOG is a feature descriptor used in computer vision and image processing used for object detection. It is a machine learning algorithm which most popularly used for pedestrian detection due to its smooth edges [17]. We have used it in combination with the SVM algorithm to achieve high accuracy in pedestrian detection.
There are a total of four parameters in HOG Descriptor. With image being the only required parameter [18]. All other three parameters win Stride, padding, and a scale factor are an option.

C. LOCAL BINARY PATTERN HISTOGRAM (LBPH) ALGORITHM FOR FACE RECOGNITION

LBP (Local Binary Pattern) was described in 1994 and it is a very efficient texture operator which works by labeling pixels of the image by thresholding neighboring pixels and considering

resultants in binary format [19]. Additionally, it was determined that when combined with HOG (Histogram of Oriented Gradient), the performance of the algorithm improves significantly [20].

There are four parameters used in the algorithm. Radius, neighbors, grid X, and grid Y. Where the radius represents the radius around one pixel and is used to build circular LBP and is usually equal to 1, Neighbors is the number of sample points used to build circular LBP, Grid X is the number of cells in the X-axis, and Grid Y is several cells in the Y-axis, all three of these parameters are mostly set to 8.

1) Training of Algorithm: There are two steps in applying the LBPH algorithm, training, and testing. In the training step, we give the labeled dataset to the algorithm. The dataset contains images of the people with the respected ID associated with them which could be the number or name of that person. This information will be needed for the algorithm to give you output based on the input results.
2) Testing of Algorithm: The second step is applying the algorithm. Here, we are applying the algorithm to the video footage. The algorithm will run after the face has been detected and will start doing the calculations i.e., finding the most related image in the dataset by comparing the histograms and returning the result (the related person ID) based on the minimum histogram value.

D. CENTROID TRACKING ALGORITHM FOR PERSON TRACKING

The centroid tracking algorithm is a tracking algorithm used to track objects. Here we will use this algorithm to track the people in the video footage. This will stop the system from confusing the same person as two individuals.

The centroid tracking algorithm work by taking the bounding box coordinates of the person and calculating the centroid value. The unique ID is assigned to each object i.e., different IDs will be assigned to different people and the different occurrence of the same person as well. The person is tracked by the method of calculating the Euclidean distance between the people. Such as if there are three people in the video footage at the same time, then the distance between the people in the last frame will be calculated with the distance of the person in the current frame to see if the person is the same. And if the answer is yes i.e., if a minimum distance value between two objects is found then the same ID will be allocated to the object in the current frame marking them as the same person. The X and Y coordinates are then updated and if any new object is detected then a new ID will be allocated to the new object and their centroid values will be calculated in the same way. And if any object has moved out of frame, then it will be deregistered from the system.

E. EFFICIENCY AND SPEED

The efficiency and speed of the system also matter. As this system is made by keeping the criminal investigation in mind so if it takes 20 hours to process 18 hours of video then it could be more effective but not efficient. We use some techniques to increase the speed of the system.

1) Resizing Frames: The video that the user gives as input to the system could of any quality i.e., could be low-quality video footage of 240p, and could also be a high-quality video of 1080p. So, the system needed to be made to support all qualities of videos. That is why we tested our system in different qualities of videos. And while testing we note that the videos having better quality are generating bigger frames than the screen. The frames of the video are not fully visible on the screen. One thing to note here is that the screen quality and aspect ratio also matter. As the video is generating frames in full size regardless of the size of the screen. And as the size of video frames is bigger, it is taking more time to process each frame which is obvious as larger frames and good quality means more pixels which means more computational cost and processing time. So, for solving these problems we resized the frame by taking the aspect ratio of the screen and frame and finding the difference between them in percent. Then finding the lowest value between the percentages of width and height and reducing the frame by the lowest percentage.
2) Speeding Up Code: We have used multiple methods which take less operational cost and time. As we are made the system in a python programming language.

   a) Less use of for loop: We note that the for loop takes more time as it is dynamic in python.
   b) Using List comprehension: Also, list comprehension is faster than the append method. So, we use list comprehensions wherever possible especially for creating bounding boxes.
   c) Using more local variables: The global variables or class level variables takes more time than local variables, so we use fewer global and class-level variables, especially inside loops. We convert all the class level variables to locals before passing them to loops so that it will take less time.
   d) Using f strings: While researching we find out that f string is the fastest way of formatting strings as compared to other methods. So, we use f string everywhere in the code for string formatting. The benchmarks of different string formatting methods are as follows.
   e) Using 1 for infinity loop: For infinite while loop using True takes more time than using 1 as True is a Boolean operator and 1 is an integer so we use 1 instead of True for infinite loops.
   f) Using multiple assignments: Assigning the values to multiple related variables at a time is more efficient than assigning the value to one variable at a time. Such as forgetting the values of x, y, width, and height of the person for creating a bounding box using multiple assignments will be more efficient.

3) Multiprocessing: Resizing Frame and Speeding up the code make the processing faster but not faster than the video time.

We intended to make the system process the video faster than the total time of the video. Which was not achieved by the above methods, especially for videos of high quality. So, for that purpose, we use multiprocessing (distributed processing or computing). We use distributed processing for distributing the workload on multiple cores of the processor. For this, we need a multicore system and nowadays almost all system has more than one core. So, keeping this thing in mind we make use of all cores by diving the video footage into n equal parts, where n is several cores of the processor. Then each part is sent to each core for processing and after the processing is done the result will be generated by each core concerning the timing of the video part. The timing of the wanted person occurrence will be corrected concerning the timing of the video and the final report will be generated.

TABLE I
STING FORMATTING

String Formatting Methods	Time
f-string	1.0
Concatenate string	1.33
Join a sequence of strings	1.73
% Formatting operator	2.40
Format () Method	3.62
Template (Method)	11.48

IV. RESULTS

The system could be used for criminal identification through recognition and the combination of algorithms used here could help solve many problems. Such the combination of Haar Cascade face detection and LBPH for recognition generate highly accurate results as compared to when the LBPH is used with any other algorithm for face detection. This could be because both are working on face features for recognition and detection. HOG is used here for person detection.

The use of a centroid tracking algorithm in the system will help the system from confusing the same person as two individuals. As all the people are being tracked in the footage so this will tell the system that it is the same occurrence of the same person. Then when the next occurrence occurs it will be saved in the report. The report will be generated at the end containing all the occurrences of the wanted person in the video footage. FIGURE 5, 6, and 7 represent the percentage cases of each instance in terms of True Positive, True Negative, False Positive, and False Negative. For determining how much efficiency has been achieved in the video processing by implementing multiprocessing same video has been processed by the system in different quality. And as the system is using multiprocessing so to be affected less by the background processes the process has been repeated 5 times for each video quality. The average results on the video processing time in seconds by using multiprocessing in the system of cores i5 processer having 8 logical cores are as follows. The total time of the video being tested is 151 seconds. TABLE 2 represents a scenario where it is displayed that multiprocessing of the videos will perform much faster.

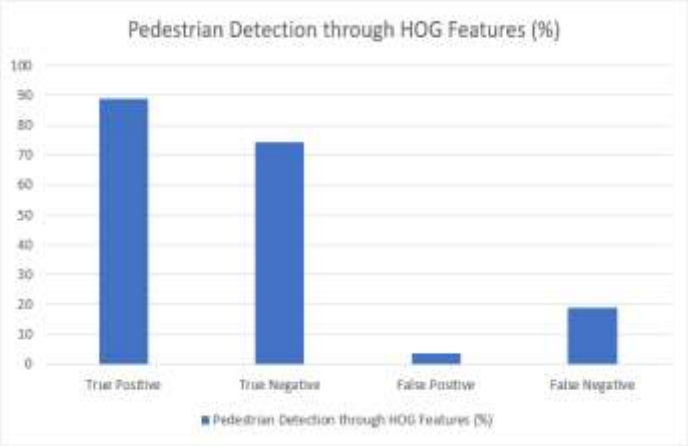

FIGURE 5. Results of pedestrian detection through HOG algorithm (%)

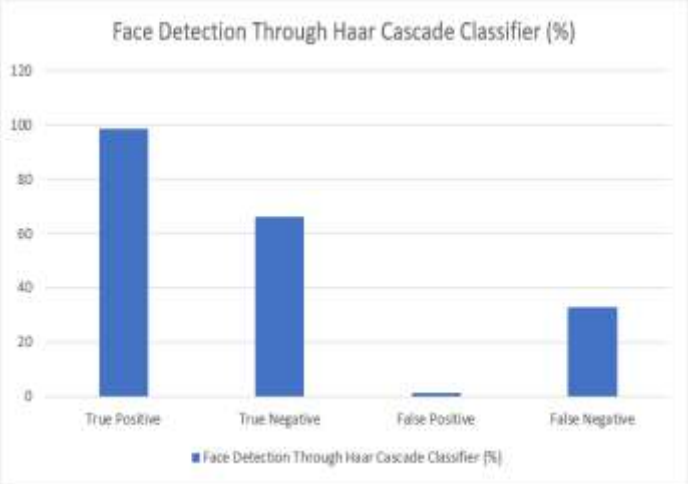

FIGURE 6. Results of Face Detection through Haar Cascade algorithm (%)

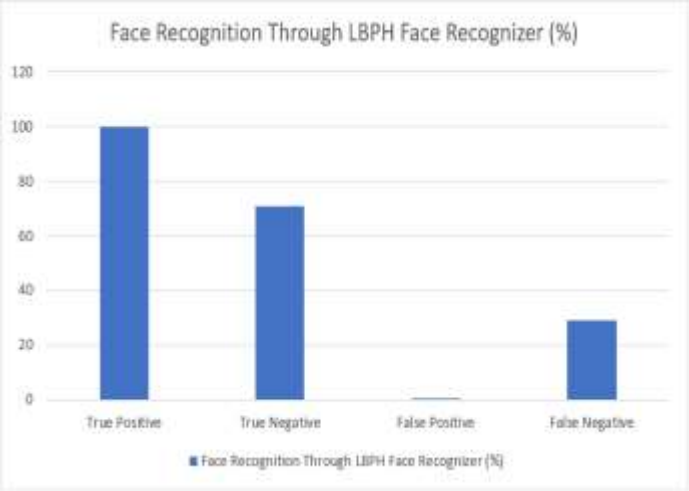

FIGURE 7. Results of Face Recognition through LBPH Face Recognizer (%)

TABLE II
MULTIPROCESSING

Without Multiprocessing	With Multiprocessing
162 sec	41.06 sec
209.8 sec	58.6 sec
269.2 sec	88.4 sec
387.2 sec	279.6 sec
506 sec	306.4 sec

The results are of 240, 360, 480, 720, and 1080p qualities of video respectively. Here it can be observed that the processing improvement decreases with the increase in quality, and it is because now it has more data to process with each frame. And there is always some energy loss in the system, intensive task causes more loss so as larger frames. But keeping in mind that video footages are rarely 720 or 1080p the improvement that could be achieved through the system is at least three times. This system will automate the process and as the process will be automated then there will be no human intervention hence no human error. And it will help in making investigation faster.

V. CONCLUSION

The algorithms used in the system discussed above provide good accuracy among the other algorithms used for the same purpose. And the combination of LBPH and Haar Cascade makes the accuracy even higher. Centroid tracking helps reduce the computational cost by stopping the rerunning of the whole process on each frame. And the system will automate the acquired video footage scrutinization work of the investigation unit. There could be more work done in this area for speeding the processing of the system such as distributed processing could be used for speeding the whole process which will divide, the video into multiple chunks, and the process will be done the different cores of the processor, which will improve the system speed. Making the process even faster and hence speeding the overall investigation process.

ACKNOWLEDGMENT

We would like to thank our Software engineering department for guiding us and providing resources to complete our task on time.

REFERENCES

[1] Siddiqui, Aalia, and Sarah Wayland. "Lost from the conversation: Missing people, and the role of Police media in shaping community awareness." *The Police Journal*, 2021: 0032258X20984502.

[2] Heyns, Andries M. "Optimisation of surveillance camera site locations and viewing angles using a novel multi-attribute, multi-objective genetic algorithm: A Day/night anti-poaching application." *Computers, Environment and Urban Systems*, vol. 88, pp. 101638, 2021.

[3] Terrill, William, and Laura Zimmerman. "Police Use of Force Escalation and De-escalation: The Use of Systematic Social Observation with Video Footage." *Police Quarterly*, 2021: 10986111211049145.

[4] Vennam, Preethi, Pramod TC, Thippeswamy BM, Yong-Guk Kim, and Pavan Kumar BN. "Attacks and Preventive Measures on Video Surveillance Systems: A Review." *Applied Sciences* vol. 11, no. 12, pp. 5571, 2021.

[5] Kostka, Genia, Léa Steinacker, and Miriam Meckel. "Between security and convenience: Facial recognition technology in the eyes of citizens in China, Germany, the United Kingdom, and the United States." *Public Understanding of Science* vol. 30, no. 6, pp. 671-690, 2021.

[6] Hussain, Tanveer, Khan Muhammad, Weiping Ding, Jaime Lloret, Sung Wook Baik, and Victor Hugo C. de Albuquerque. "A comprehensive survey of multi-view video summarization." *Pattern Recognition* vol. 109, pp. 107567, 2021.

[7] Mabrouk, Amira Ben, and Ezzeddine Zagrouba. "Abnormal behavior recognition for intelligent video surveillance systems: A review." *Expert Systems with Applications* vol. 91, pp. 480-491, 2018.

[8] Chokkadi, Sukhada, and Abhir Bhandary. "A Study on various state of the art of the Art Face Recognition System using Deep Learning Techniques." *arXiv preprint arXiv:1911.08426*, 2019.

[9] Girmay, Siem, Faniel Samsom, and Asad Masood Khattak. "AI based Login System using Facial Recognition." In *2021 5th Cyber Security in Networking Conference (CSNet)*, IEEE, 2021, pp. 107-109

[10] Li, Jian-hua. "Cyber security meets artificial intelligence: a survey." *Frontiers of Information Technology & Electronic Engineering* vol. 19, no. 12, pp. 1462-1474, 2018.

[11] Prabhu, Tanniru, and C. Dhanamjayulu. "Automated Home Security using Facial Recognition with IoT Remote Monitoring." In *2021 Innovations in Power and Advanced Computing Technologies (i-PACT)*, IEEE, 2021, pp. 1-7.

[12] Mohammed, Uzayruddin Siddiqui, Vinod Kumar Shukla, Robin Sharma, and Amit Verma. "Next Step to the Future of Restaurants Through Artificial Intelligence and Facial Recognition." In *Cyber Intelligence and Information Retrieval*, pp. 89-101. Springer, Singapore, 2022.

[13] Rahmad, C., R. A. Asmara, D. R. H. Putra, I. Dharma, H. Darmono, and I. Muhiqqin. "Comparison of Viola-Jones Haar Cascade classifier and histogram of oriented gradients (HOG) for face detection." In *IOP conference series: materials science and engineering*, vol. 732, no. 1, p. 012038. IOP Publishing, 2020.

[14] Deeba, Farah, Aftab Ahmed, Hira Memon, Fayaz Ali Dharejo, and Abddul Ghaffar. "LBPH-based enhanced real-time face recognition." *Int. J. Adv. Comput. Sci. Appl* vol. 10, no. 5, pp. 274-280, 2019.

[15] Jagtap, A. M., Vrushabh Kangale, Kushal Unune, and Prathmesh Gosavi. "A Study of LBPH, Eigenface, Fisherface and Haar-like features for Face recognition using OpenCV." In *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, IEEE, 2019, pp. 219-224.

[16] Rahmad, C., R. A. Asmara, D. R. H. Putra, I. Dharma, H. Darmono, and I. Muhiqqin. "Comparison of Viola-Jones Haar Cascade classifier and histogram of oriented gradients (HOG) for face detection." In *IOP conference series: materials science and engineering*, vol. 732, no. 1, p. 012038. IOP Publishing, 2020.

[17] Firuzi, Keyvan, Mehdi Vakilian, B. Toan Phung, and Trevor R. Blackburn. "Partial discharges pattern recognition of transformer defect

model by LBP & HOG features." *IEEE Transactions on Power Delivery* vol. 34, no. 2, pp. 542-550, 2018.

[18] Bakheet, Samy, and Ayoub Al-Hamadi. "A framework for instantaneous driver drowsiness detection based on improved HOG features and naïve Bayesian classification." *Brain Sciences* vol. 11, no. 2, pp. 240, 2021.

[19] Upadhyay, Ashutosh, and S. Vijayalakshmi. "Efficient Half Face Detection System Based on Linear Binary Pattern." *Journal of Computational and Theoretical Nanoscience* vol. 17, no. 5, pp. 2342-2348, 2020.

[20] A. M. Jagtap, V. Kangale, K. Ununa, & P. Gosavi, "A Study of LBPH, Eigenface, Fisherface and Haar-like features for Face recognition using OpenCV", 2019 International Conference on Intelligent Sustainable Systems (ICISS), , 2019, pp. 219-224.