e-ISSN: 2582-5208
International Research Journal of Modernization in EngineeringTechnology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:05/May-2023 Impact Factor- 7.868 www.irjmets.com

AUTOMATIC SMART SECURITY CAMERA

Mohd. Jawed Khan[*1], Adarsh Kumar[*2], Ajay Kumar[*3], Ankit Prajapati[*4], Arpan Kumari[*5]

[*1]Assistant Professor, Department Of Computer Science & Engineering, Greater Noida Institute Of Technology (Engineering Institute), Greater Noida, Uttar Pradesh, India.

[*2,3,4,5]B.Tech Student, Department Of Computer Science & Engineering, Greater Noida Institute Of Technology (Engineering Institute), Greater Noida, Uttar Pradesh, India.

DOI : https://www.doi.org/10.56726/IRJMETS38693

ABSTRACT

Nowadays security becomes the major aspect in corporate offices and banks. Most of the individuals and organizations use Closed Circuit Television (CCTV) cameras to monitor activities in the offices and banks when it is closed and when there are no any security guard available especially during off days. In various cities, the footage captured by surveillance system cameras are significant for prevention and investigation of crime. Because the system proposed not only records but it also trigger alerts by analysing objects, traditional video surveillance is less effective. The system requires no database, this makes the camera more efficient and light weight making it to detect the human object in very less time. This smart camera purpose is to stop the robbery by sending the alarm to the owner and making him/her alert. This camera provides a reliable monitoring mechanism that can detect any kind of suspicious object like human or animal through OpenCV Library using Haar cascade algorithm. The purpose of this project is to provide high level security with less storage capacity requirement with maximum efficiency with speed.

Tracing the video is also efficient. Because it captures only the detected activity which involves human presence.

Keywords: Object Detection, Smart Security Camera, Haar Cascade, Storage, Alert, Opencv, SNS.

I. INTRODUCTION

The use of security cameras for surveillance purposes has become increasingly prevalent in recent years due to the growing concern for public safety. However, traditional surveillance cameras have limitations in their ability to detect and respond to potential security threats in real-time. To overcome these limitations, researchers have been exploring the use of computer vision algorithms to develop smart security cameras that can automatically detect and respond to security threats.

This research paper presents a novel approach for developing a smart security camera using Haar cascade and OpenCV [1]. Haar cascade is a machine learning-based approach for object detection that uses Haar-like features to classify objects in images. OpenCV is an open-source computer vision library that provides tools for image processing, feature extraction, and object detection.

The proposed system consists of a camera module that captures video footage, a computer system that processes the video footage in real-time, and a user interface that displays the results of the analysis. The system uses Haar cascade and openCV to detect and recognize different objects and events in the video footage.

The main objective of the system is to detect potential security threats such as intruders. The system achieves this by analyzing the video and detecting the human object in the video footage using Haar cascade Algorithm [2].

The proposed system utilizes a multi-stage cascade of classifiers to achieve high detection accuracy and reduce false positives. The cascade consists of multiple stages, each of which contains a set of weak classifiers. The weak classifiers are trained on Haar-like features extracted from positive and negative training samples.

To evaluate the performance of the proposed system, a series of experiments were conducted using video footage captured in different environments. The results of the experiments show that the proposed system achieves high detection accuracy with low false positive rates. The system was able to detect and recognize different objects and events with high accuracy, including human faces, vehicles, and suspicious packages.

Smart mode is main feature of this camera which uses Haar cascade algorithm [3,4] to detect the human and to enhance this algorithm under the hooks the system uses the feature of AdaBoost algorithm [5] too. By doing these we are increasing the security level of this camera. When human object gets detected we are sending the

e-ISSN: 2582-5208
International Research Journal of Modernization in EngineeringTechnology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:05/May-2023 Impact Factor- 7.868 www.irjmets.com

alarm to the owner using AWS SNS system [6]. This system is used to increase the security of the data as well as to provide portability and availability of data.

II. LITERATURE REVIEW

Viola, Paul, and Michael J. Jones. [4] described a method for detecting objects in images using a cascade of simple features and boosting, and demonstrates its effectiveness in detecting faces in real-time applications. The method involves training a classifier on a set of positive and negative examples using a series of simple features (Haar-like features) that can be computed efficiently. The classifier is then organized into a cascade of stages, each consisting of a subset of the features, which allows for efficient processing and rejection of non-object regions in the image. Finally, the classifier is boosted using AdaBoost, a machine learning algorithm that combines weak classifiers into a strong classifier.

The Haar Cascade algorithm has since become a popular method for object detection in computer vision, and has been applied to various tasks such as pedestrian detection, vehicle detection, and even detecting defects in industrial products.

Yoav Freund and Robert E. Schapire. [5] introduced the AdaBoost algorithm, which is a popular and widely used algorithm for ensemble learning and classification. It has been applied to a wide range of tasks, including face detection, object recognition, and spam filtering.

Several studies have been conducted on the use of OpenCV and Haar cascades for smart security cameras. For example, in [7], the authors proposed a smart security camera system that uses OpenCV and Haar cascades for detecting and tracking moving objects. The system can differentiate between humans and animals and can send alerts to the user in real-time.

Li, S., Li, Z., & Wang, J. [8] proposed a smart security camera system that uses OpenCV and Haar cascades for detecting and tracking suspicious activities. The system can detect and track multiple objects simultaneously and can send alerts to the user when an activity is detected.

Zhang, Z., Li, B., & Peng, C. [9] proposed a smart security camera system that uses OpenCV and Haar cascades for detecting and tracking people in real-time. The system can distinguish between different types of people, such as adults and children, and can send alerts to the user when a person is detected.

III. METHODOLOGY

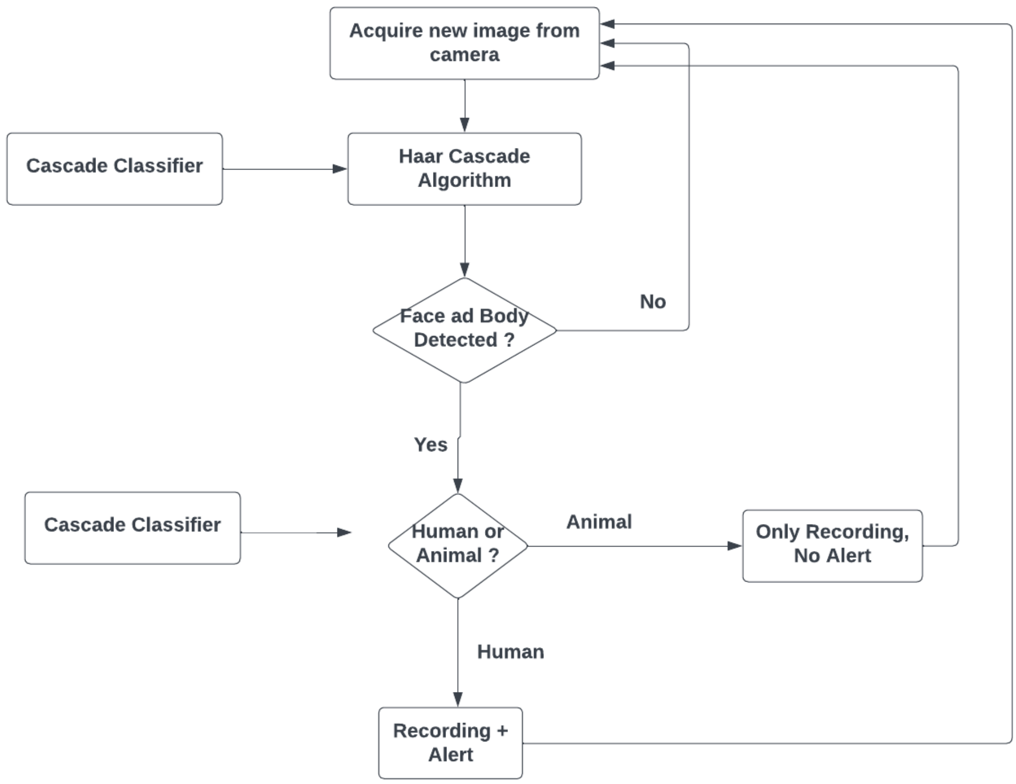

Fig 1. Figure showing the working flow of the system

IRJMETS

e-ISSN: 2582-5208

International Research Journal of Modernization in EngineeringTechnology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:05/Issue:05/May-2023 Impact Factor- 7.868 www.irjmets.com

Fig 1 shows the working flow of the system. Following are the steps that the system goes through :

1) Acquire new mage from camera : The image has been captured through the video recorded by the camera. This image is treated as data for filtering the human body. The image and video capturing is done using OpenCV [2] library of Python.

2) Haar Cascade Algorithm : The Haar Cascade algorithm is a popular method for object detection in images and videos. It uses a set of trained classifiers to detect specific features of an object, such as edges, lines, and corners, and then uses these features to recognize the object in the image or video. A classifier is trained with number of images in a machine learning model for object detection and such classifier is used for detecting an object in an image.

Haar-like features are simple rectangular patterns that are used to describe the properties of an object. These features are computed by subtracting the sum of the pixel values in a white rectangle from the sum of the pixel values in a black rectangle. The resulting value is then normalized by the area of the rectangle. These features can be computed at different scales and positions in an image to detect objects of varying sizes and orientations.

The cascade classifier is a series of simple classifiers that are trained on positive and negative examples of the object of interest. The classifier evaluates each Haar-like feature and assigns a weight to each feature based on its ability to distinguish between positive and negative examples. The weights are then combined to give a final score for the object. If the score exceeds a threshold, the object is detected.

The training process for the cascade classifier involves several stages, with each stage consisting of multiple weak classifiers. The weak classifiers are trained on a subset of the features and the negative examples. The features with the highest weights are then selected for the next stage, and the process is repeated until the desired detection accuracy is achieved.

The Haar Cascade algorithm involves several steps, including:

a) XML Files for Mapping: The XML files used in the Haar Cascade algorithm contain the trained classifiers for each stage of the cascade. These files are generated during the training phase and contain information about the features used by each classifier, as well as the threshold values used to make decisions. During the detection phase, the XML files are loaded into the algorithm and used to detect the object in the image.Used the default xml file for mapping the image with our object. These XML files includes trained models for various human body parts detection.

b) Feature Extraction: In this step, the algorithm extracts features from the images using the Haar wavelet. The Haar wavelet is a mathematical function that can detect edges, lines, and corners in an image.

The Haar wavelet is a simple and efficient wavelet transform that uses a step function to identify discontinuities in signals or images. It is a piecewise constant function that has a value of +1 in one half of its support and -1 in the other half, and it is zero elsewhere. The Haar wavelet is used in image compression, noise reduction, and feature extraction, among other applications.

The Haar wavelet transform is performed by dividing the input signal or image into non-overlapping blocks and applying the Haar wavelet function to each block. The resulting wavelet coefficients represent the different frequency components of the signal or image. The wavelet coefficients can then be quantized and compressed to reduce the size of the data while preserving important features of the original signal or image.

Overall, the Haar wavelet is a powerful mathematical tool that is widely used in signal and image processing, as well as in other fields such as finance and geology.

c) Training: The algorithm uses the extracted features to train a set of classifiers using machine learning techniques. The classifiers are trained to distinguish between positive and negative images. A machine learning algorithm AdaBoost is used to train a classifier on the extracted features. The classifier is trained to distinguish between positive and negative samples.

AdaBoost is a machine learning algorithm used for binary classification problems. It works by iteratively training weak classifiers on weighted versions of the training data, with each weak classifier focusing on the samples that the previous weak classifiers got wrong. The final prediction is made by combining the predictions of all the weak classifiers with their associated weights.

d) Cascading: Haar cascades consist of a series of classifiers that are trained to detect different features of an object. The cascading procedure works by dividing the image into smaller sub-regions and applying the classifiers to these sub-regions in a sequence. The trained classifiers are organized into a cascade, with each stage consisting of a set of classifiers. The classifier is used to create a cascade of classifiers, where each stage of the cascade consists of multiple weak classifiers. The output of each stage is used as input to the next stage, with the aim of reducing the false positive rate while maintaining a high detection rate. The cascade is designed to quickly reject negative images, reducing the number of false positives.

By using the cascading procedure, the algorithm can quickly discard regions of the image that are unlikely to contain the object of interest, thus reducing the overall computational time required for object detection.

The cascade classifier uses a technique called the cascade filter, which consists of the following steps:

(i) Image Preprocessing: The input image is first converted to grayscale, which reduces the computation required to process the image. Then, the image is resized to a fixed size for consistency across all images.

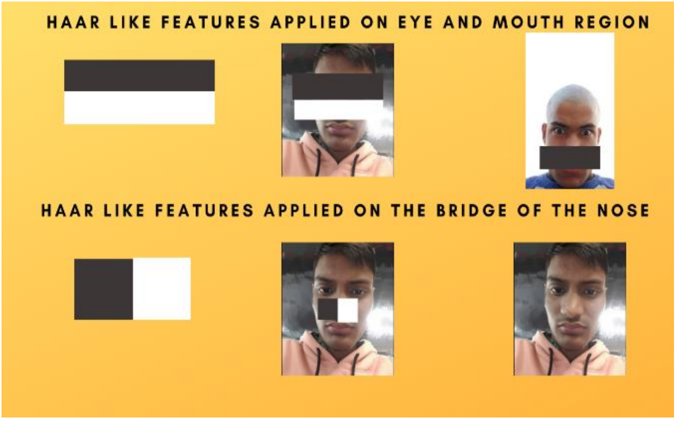

Fig 2. The figure shows the haar feature detecting eyes, nose and mouth of the human object.

The Fig 2. shows the haar feature detecting eyes, nose and mouth of the human object.

(ii) Haar Feature Selection: The next step involves selecting the best Haar-like features that can distinguish the object of interest from the background. Haar features are rectangular areas that have different intensities or color values, and they are computed by subtracting the sum of pixels in a white rectangle from the sum of pixels in a black rectangle.

0.4	0.7	0.9	0.7	0.4	0.5	1.0	0.3
0.3	1.0	0.5	0.8	0.7	0.4	0.1	0.4
0.9	0.4	0.1	0.2	0.5	0.8	0.2	0.9
0.3	0.6	0.8	1.0	0.3	0.7	0.5	0.3
0.2	0.9	0.1	0.5	0.1	0.4	0.8	0.8
0.5	0.1	0.3	0.7	0.9	0.6	1.0	0.2
0.8	0.4	1.0	0.2	0.7	0.3	0.1	0.4
0.4	0.9	0.6	0.6	0.2	1.0	0.5	0.9

Fig 3.1

0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

Fig 3.2

Fig 3. Sum Of The Dark Pixels /Number Of Dark Pixels – Sum Of The Light Pixels /Number Of The Light Pixels

$$(0.7 + 0.4 + 0.1 + 0.5 + 0.8 + 0.2 + 0.3 + 0.7 + 0.5 + 0.1 + 0.4 + 0.8 + 0.9 + 0.6 + 1.0 + 0.7 + 0.3 + 0.1)/18$$

$$(0.1 + 0.5 + 0.8 + 0.4 + 0.1 + 0.2 + 0.6 + 0.8 + 1.0 + 0.9 + 0.1 + 0.5 + 0.1 + 0.3 + 0.7 + 0.4 + 1.0 + 0.2)/18$$

$$0.51 – 0.53 = - 0.02$$

The rectangle represented by Fig 3.1 (on the left) is a depiction of an sample image with pixel values in the range of 0.0 to 1.0. The rectangle represented by Fig 3.2 (on the right) is a Haar kernel with all the dark pixels on the right and all the light pixels on the left. The formula to calculate the Haar calculation is the difference between the average of the pixel values in the darker area and the average of the pixel values at the lighter area. If the difference came out is close to 1, then there is an edge that is detected by the haar feature.

To calculate the sum of all the image pixels lying in the darker area and the lighter area of a Haar feature, you will first need to define the Haar feature that you are interested in. Haar features are rectangular areas of an image, with some areas being darker than others. And then find out their difference. Now, if the image has an edge separating dark pixels on the right and light pixels on the left, then the haar value will be closer to 1. That means, if the haar value is closer to 1, we can say that there is an edge detected. In the above example, since the haar value is far away from 1, there is no edge detected.

(iii) Integral Image: The integral image is used to speed up the computation of Haar features. It is a two-dimensional array where each element represents the sum of all the pixels to the left of it and above it in the original image. Calculation of haar features takes O(n * n) time. Using Integral image, this complexity is reduced to O(1) so that the mapping is time efficient.

0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	74

1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

Fig4. Integral Image Input Image

Fig 4. The figure Fig 4 shows formation of the integral image. In the integral image, a pixel is the sum of all the pixels on the left of it and all the pixels above it in original (input) image.

(iv) Cascade Classification: Finally, the cascade classifier applies the trained weak classifiers in a cascading manner. Each stage of the cascade classifier consists of multiple fragile classifiers, and if an image fails to pass any stage, it is rejected as a negative sample. If an image passes all stages, it is classified as a positive sample.

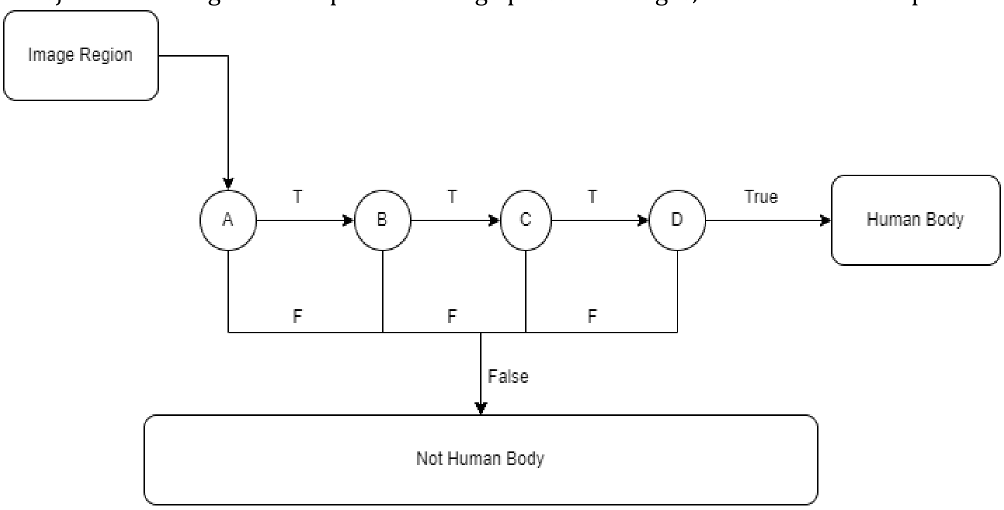

Fig 5. The figure above shows Cascade Filter. If an image passes all stages, it is classified as a positive sample and if an image fails to pass any stage, it is rejected as a negative sample.

e) Detection: The final step is to use the cascade to detect the object in the image. The algorithm scans the image at different scales and positions, using the classifiers to detect the object. If the object is detected, its location is returned.

A classifier is trained with number of images in a machine learning model for object detection and such classifier is used for detecting an object in an image.

e-ISSN: 2582-5208
International Research Journal of Modernization in EngineeringTechnology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:05/May-2023 Impact Factor- 7.868 www.irjmets.com

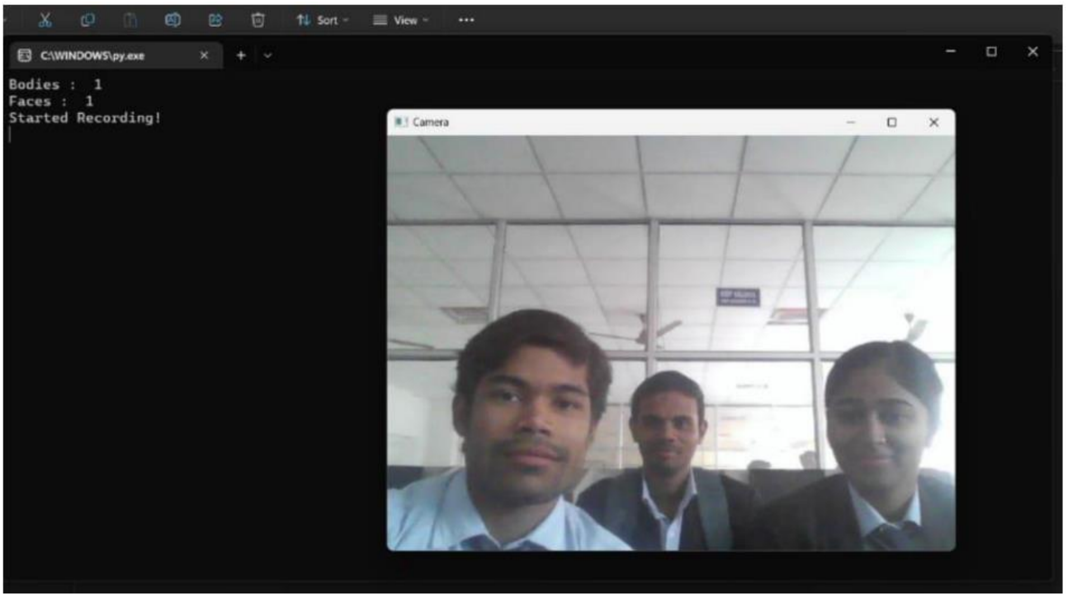

Fig 6.

The figure Fig 6. shows the working of the system. If a human object is detected, the system automatically starts recording and if the human object is not detected, if either stops recording (if the recording is started) or it doesn't starts any recording.

3) Recording Process : Whenever the suspicious object is detected using HaarCascade Algorithm, the recording function of the camera gets started automatically through OpenCV Libraray . By this function, storage requirement is subsequently reduced to very low because we are recording only when object is detected.

4) Security Alert System : When the human object gets detected, an alert is send to the owner. Here, SNS features of AWS has been used to send the message safely and securily to the owner. This procedure is running in the loop which makes it able to detect the object all the time without any time gap.

AWS offers several social networking service (SNS) features that allow users to build, scale, and operate applications that send notifications to various endpoints or devices. Here are some of the key features of AWS SNS:

a) Topics: AWS SNS allows users to create topics, which are logical access points that can receive and distribute messages. Each topic can have multiple subscribers, and messages sent to a topic are delivered to all its subscribers.

b) Subscriptions: Users can subscribe to a topic to receive notifications through various channels, such as email, SMS, HTTP, Lambda, or mobile push notifications.

c) Mobile push notifications: AWS SNS provides support for mobile push notifications for iOS, Android, Fire OS, and Windows devices. Developers can send push notifications to devices using the SNS mobile push API.

d) Fanout: AWS SNS supports fanout, which is the ability to send a single message to multiple endpoints or subscribers at once. This helps ensure that messages are delivered reliably and efficiently.

e) Filtering: Users can use AWS SNS to filter messages based on attributes such as message contents, sender identity, or subscription attributes. This allows developers to send targeted notifications to specific subscribers based on their interests or preferences.

f) Encryption: AWS SNS provides encryption features to ensure the security of messages sent between endpoints. Users can choose to encrypt messages with their own keys or use AWS-managed encryption keys.

Overall, AWS SNS provides a powerful and flexible platform for sending notifications across a wide range of channels and devices, making it an essential tool for building modern applications.

e-ISSN: 2582-5208
International Research Journal of Modernization in EngineeringTechnology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:05/May-2023 Impact Factor- 7.868 www.irjmets.com

How the system is comparatively efficient with accurate?

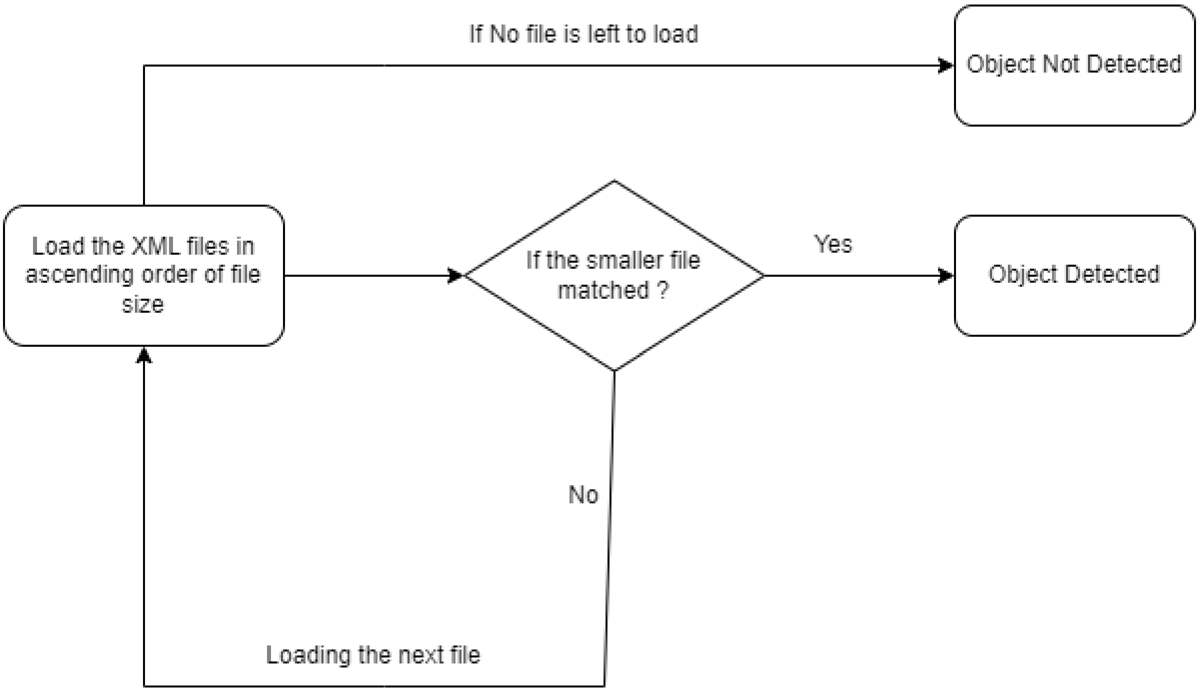

Fig 7. The figure shows how the system is comparatively more efficient and accurate.

The Fig 7 shows how the system is comparatively more efficient and accurate. The system will load the XML files in ascending order of the file size. The smaller files is processed first. If the smaller file is matched and it detected any human body part, the system will not go to match the other XML files. If a file is not matched, then only the system goes to match the other XML files greater in size. At last, if no file is left to process, this means that the human object is not detected.

The algorithm should not crash in mid at any cost and the algorithm should faster as well as accurate . Because the speed and high requirement of algorithm can slow down the detection system and this can be a glitch in any security system. Instead of this there should be high accuracy and it is very important so that not a single object can pass without getting detected at any cost. So to do this we are using the Haar cascade object detection where we are not using the default object detection[10] method of Haar cascade instead we are adding other object part detection too so that nothing can escape without getting detected. But when we do this then we require a lot of xml files which will increase the time because to load all the xml files it takes lot of time. To cover those things we are arranging the detection system based on their xml file size.

Parameter considered for the system to work

1. **Haar cascade classifier :** The system uses AdaBoost Algorithm, which is considered to be the best for image detection in less time with the help of Integral Image.
2. **Brightness and Background :** The algorithm is designed to work best with images that have good contrast between the object being detected and the background. If the object is too bright or too dark compared to the background, it may be difficult for the algorithm to detect it reliably.
3. **Xml trained file :** The system uses the best trained XML file for mapping.

IV. RESULT

The research has been performed considering above parameters. We have taken 20 human object in different ways, out of which 18 are detected by the system. Other two which are not detected, are wrapped in clothes such that their body parts were not visible. In this way, we concluded that the system accuracy is 90%.Also, the system can detect human object only when there is proper contrast between the human object and the background. The camera must differentiate between the human object and the background to detect the images and work efficiently.

The system uses trained XML files for every parts of the human body which gives the system to detect images with 90% accuracy.

The system maps XML files according to their increasing order of file size. By doing this, the system becomes more efficient and fast.

V. CONCLUSION

In conclusion, the use of Haar cascades and OpenCV in a smart security camera system can be an effective solution for real-time object detection and tracking in security applications. However, the performance of the system depends on the accuracy of the cascade classifier, the quality of the video stream, and the computational resources available for processing.

VI. REFERENCES

[1] Atif, M., Khand, Z.H., Khan, S., Akhtar, F. and Rajput, A., 2021. Storage Optimization using Adaptive Thresholding Motion Detection. Engineering, Technology & Applied Science Research, 11(2), pp.6869-6872.

[2] G, Chandan & Jain, Ayush & Jain, Harsh & Mohana, Mohana. (2018). Real Time Object Detection and Tracking Using Deep Learning and OpenCV. 1305-1308.

[3] Padilla, Rafael & Filho, Cicero & Costa, Marly. (2012). Evaluation of Haar Cascade Classifiers for Face Detection.

[4] Viola, Paul, and Michael J. Jones. "Rapid object detection using a boosted cascade of simple features." Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. I-511. IEEE, 2001.

[5] Yoav Freund and Robert E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting." Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119-139, 1997.

[6] Amazon Simple Notification Service: The Missing Piece of the Cloud Puzzle, by M. David Allen and Joe Walkenhorst. This paper discusses how SNS can be used as a messaging system for cloud-based applications.

[7] Zhou, Y., & Liu, Y. (2018). Design and implementation of intelligent monitoring system based on OpenCV. Journal of Physics: Conference Series, 1013(1), 012080.

[8] Li, S., Li, Z., & Wang, J. (2019). A Smart Camera System for Detecting and Tracking Suspicious Activities in Real Time. IEEE Access, 7, 118634-118645.

[9] Zhang, Z., Li, B., & Peng, C. (2019). Research on smart security monitoring system based on OpenCV. Journal of Physics: Conference Series, 1282(4), 042065

[10] Hasan, Ramadan & Sallow, Amira. (2021). Face Detection and Recognition Using OpenCV. Journal of Soft Computing and Data Mining. 2. 10.30880/jscdm.2021.02.02.008.

[11] Dhaya, R., 2020. CCTV Surveillance for Unprecedented Violence and Traffic Monitoring. Journal of Innovative Image Processing (JIIP), 2(01), pp.25-34.

[12] Amit, Yali & Felzenszwalb, Pedro & Girshick, Ross. (2020). Object Detection.

[13] Sreenu, G., and MA Saleem Durai. "Intelligent video surveillance: a review through deep learning techniques for crowd analysis." Journal of Big Data 6.1 (2019): 1-27.

[14] Kakadiya, Rutvik, et al. "Ai based automatic robbery/theft detection using smart surveillance in banks." 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, 2019.

[15] Sedky, M.H., Moniri, M. and Chibelushi, C.C., 2005, September. Classification of smart video surveillance systems for commercial applications. In IEEE Conference on Advanced Video and Signal Based Surveillance, 2005. (pp. 638-643). IEEE.

[16] J. Fagertun, 2005. Face Recognition. Master Thesis, Technical University of Denmark (DTU).