EFFICIENT EYE : SMART SECURITY WITH OPENCV & HAAR CASCADE FOR STORAGE OPTIMIZATION & SURVEILLANCE

1st VIKAS KUMAR MAURYA.
MCA
SHARDA UNIVERSITY

2nd KUNDAN KUMAR JHA
MCA
SHARDA UNIVERSITY

3rd ARYAN SHRIVASTAVA
MCA
SHARDA UNIVERSITY

Vikas707maurya@gmail.com Kundankavyanshu1234@gmail.com
2022007642.aryan@pg.sharda.ac.in
*Corresponding Author – **Dr. Yogesh Singh Rathore**
Professor, Computer Science & Applications (CSA)
Greater Noida, UP, India
yogesh.rathore@sharda.ac.in

ABSTRACT

These days, banks and corporate offices prioritise security. The majority of people and businesses use closed-circuit television, or CCTV, cameras to keep an eye on what goes on in banks and offices after hours and without the presence of security personnel, particularly on non-working days. The footage that surveillance system cameras record is important for both crime prevention and investigation in many places. Traditional video surveillance is less effective since the suggested solution not only records but also triggers alarms by evaluating things. Because the system doesn't need a database, the camera is more effective and lightweight, allowing it to recognise human objects quickly. The goal of this smart camera is to deter theft by alerting the owner and setting off an alarm. With the use of the OpenCV Library and the Haar cascade algorithm, Camera offers a dependable monitoring system that can identify suspicious object of any kind, including people or animals. The goal of this project is to deliver strong securities with minimal space requirements while maximising efficiency and speed. Tracing the video is also effective. due to the fact that it only records activities that has been identified as involving human presence.

Key words: OpenCV, SNS, Storage, Alert, Haar Cascade, Smart Security Camera, Object Detection.

I. INTRODUCTION

The increased concern for public safety in recent years has led to a rise in the deployment of security cameras for surveillance. Nevertheless, the real-time detection and response to any security threats is limited by the nature of standard surveillance cameras. In an effort to get around these restrictions, researchers are looking at using computer vision algorithms to create smart security cameras that have the ability to recognise and react to security threats automatically.

This study presents an original approach for building a smart security camera using OpenCV and Haar cascade[1].

The object recognition technique known as Haar cascade is based on machine learning and uses Haar-like properties to identify things in pictures. OpenCV is an open-source toolkit for computer vision that provides tools for processing images, extracting features, and object detection[2].

The suggested system is made up of a camera module that records video, a computer system that analyses the video in real time, and a user interface that shows the analysis' findings. To identify and detect various objects and events in the video footage, the system makes use of openCV and Haar cascade. Identifying possible security risks, such intruders, is the system's primary goal. In order to accomplish this, the system uses the Haar cascade Algorithm to analyse the movie and identify any human objects in the footage.

In order to limit false positives and achieve high detection accuracy, the suggested system makes use of a multi-stage cascade of classifiers. There are several steps in the cascade, and each one has a collection of ineffective classifiers. Haar-like features taken from both positive and negative training samples are used to train the weak classifiers. A number of tests were carried out utilising video footage shot in various settings in order to assess the effectiveness of the suggested approach. The experiments' outcomes demonstrate that the suggested approach achieves good detection accuracy and low false positive rates.

With great precision, the system was able to identify and detect a variety of items and events, such as suspicious parcels, cars, and human faces. The key feature of this camera is its smart mode, which employs the Haar cascade algorithm [3,4] to identify human activity. The device also uses the AdaBoost algorithm [5] to improve this algorithm below the hooks. We are raising this camera's security level by taking these actions. Using the AWS SNS system, we alert the owner when a human item is discovered [6]. This platform is utilised to guarantee data availability and mobility while also enhancing security.

II. LITERATURE REVIEW

A approach for recognising objects in photos utilising a cascade of basic characteristics and boosting was presented by Viola, Paul, and Michael J. Jones [4]. They also show how effective this method is for real-time face detection. Using a set of easily calculated basic features, or "Haar-like features," the method entails training a classifier on a set of positive and negative samples. Next, to enable effective processing and rejection of nonobject portions in the image, the classifier is arranged into a cascade of steps, each comprising a subset of the characteristics. Ultimately, AdaBoost—a machine learning technique that turns a collection of weak classifiers into a strong classifier—is used to enhance the classifier.

Since then, the Haar Cascade algorithm has gained popularity as a technique for object recognition in computer vision. It has been used for a variety of tasks, including the detection of vehicles, pedestrians, and even flaws in industrial items.

The AdaBoost algorithm was developed by Yoav Freund and Robert E. Schapire [5] and is a well-liked and frequently applied algorithm for group learning and classification. Numerous applications have made use of it, such as spam filtering, object identification, and face detection. The use of cascades of Haar and OpenCV in intelligent safety cameras systems has been the subject of numerous studies. For instance, in [7], the authors suggested a smart security camera system that tracks and detects moving objects using Haar cascades and OpenCV. In addition to having the ability to distinguish between people and animals, the system can notify the user in real time. A smart security camera system that tracks and detects suspicious activity using OpenCV and Haar cascades was proposed by Li, S., Li, Z., and Wang, J. [8]. Multiple items can be tracked and recognised at once by the system, which can also notify the user when an action is discovered.

In order to detect and track people in real-time, Zhang, Li, and Peng [9] devised a smart security camera system that makes use of OpenCV and Haar cascades. When a person is discovered, the system can notify the user by sending differentiating messages based on the type of person—for example, adult vs youngster.

III. METHODOLOGY

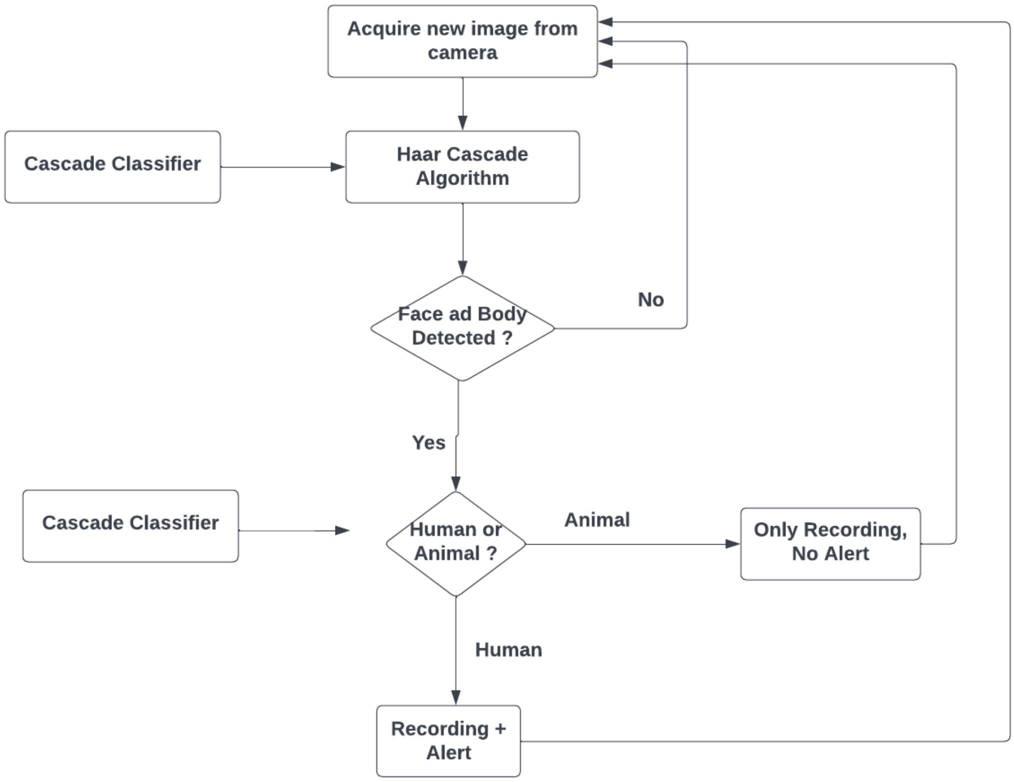

Fig 1. System's working flow

1). Obtain new image from camera: The image was obtained from the video that the camera shot. The information in this image is used to filter the human body. The Python OpenCV [2] package is used for the capture of images and videos.

2). Algorithm of Haar Cascade : The Haar Cascade algorithm is a well-liked technique for object recognition in pictures and movies. It detects particular aspects of an item, including edges, lines, and corners, using a collection of trained classifiers, and then utilises these features to identify the object in the picture or video. A machine learning model trains a classifier to recognise objects in a large number of photos. This classifier is then used to identify objects in images.

Haar-like features are straightforward rectangular patterns that characterise an object's characteristics. The computation of these features involves deducting the total of the pixel values within a white rectangle from the total of the pixel values within a black rectangle.
Afterwards, the area of the rectangle is used to normalise the value. Various items with different sizes and orientations can be detected in a picture by computing these attributes at multiple scales and positions. Simple classifiers that have been trained on both positive and negative instances of the object of interest make up the cascade classifier. Based on its capacity to discern between positive and negative samples, the classifier rates each Haar-like feature and gives it a weight. To get the object's ultimate score, the weights are then added together. The object is identified if the score is higher than a predetermined level. The cascade classifier is trained in numerous phases, each of which consists of a number of weak classifiers. A portion of the characteristics and the negative instances are used to train the weak classifiers. The procedure is then repeated until the required detection accuracy is attained, with the attributes with the highest weights being chosen for the following step.
There are multiple steps in the Haar Cascade algorithm, which include:
XML Files for Mapping: Each stage of the cascade's learned classifiers are contained in XML files that are used by the Haar Cascade technique. These files, which are produced during the training process, include details on the characteristics that every classifier uses and the threshold values that are applied for making judgements. The item in the image is detected in the detection phase by using the XML files that have been fed into the algorithm.mapped the picture to our object using the built-in XML file. These XML files comprise trained models for the detection of different human body parts.
Feature Extraction: Using the Haar wavelet, the algorithm takes features out of the pictures in this stage. An image's edges, lines, and corners can be identified using a mathematical function called the Haar wavelet.
In order to detect discontinuities in signals or images, the wavelet of Haar is a straightforward and effective transformation of wavelet that employs a function. Its support has a value of +1 in half of it and -1 in the other, it is a piecewise constant function with zero in the other half. A few of the uses for the Haar wavelet are feature extraction, noise reduction, and picture compression.
The signal’s input or images is divided into blocks that do not overlap for the wavelet of Haar transform, and every block is then given the Haar wavelets function. The various component frequencies of the signal or picture are represented by the resulting wavelet coefficients. After that, the wavelet coefficients can be compressed and quantized to lower the data size while keeping crucial aspects
All things considered, the wavelet of Haar is a potent math instrument with extensive applications Among other fields, signal and image processing disciplines like economics and geology.
a). Training: Using machine learning techniques, the algorithm trains a collection of classifiers using the feature of extraction. The ability to discern between negative and positive images is ingrained in the classifiers. The collected characteristics are utilised to train a classifier using the AdaBoost machine learning algorithm. To differentiate between negative and positive samples, the classifier is trained.

A ML approach called AdaBoost is applied to binary classification issues. Weak classifiers are iteratively trained Utilising weighted renditions of the training set, every weak classifier concentrating on examples that the preceding weak classifiers misclassified. All of the weak classifiers' predictions are combined with the corresponding weights to get the final prediction.

b). Cascading: A set of classifiers that have been trained to recognise various aspects of an object make up a Haar cascade. The way the cascading technique operates is by first segmenting dividing it into more manageable sub-regions, then sequentially apply the classifiers to these small-regions. The trained classifiers are arranged in a cascade, with a set of classifiers at each stage. A cascade of classifiers is formed using the classifier; several weak classifiers make up each stage of the cascade. The goal is to lower the false positive rate while keeping the detection rate high by using each stage's output as the input for the subsequent stage. The cascade's rapid rejection of negative images lowers the quantity of false positives.

The technique may quickly eliminate areas of the image that are unlikely to contain the object of interest by employing the cascade procedure, which cuts down on the overall calculation time needed for object detection.

The cascade classifier employs the cascade filter approach, which entails the subsequent actions:

i). Image Preprocessing : In order to minimise the amount of computation needed to the image processing , the input image is to first transformed to grayscale. For across all photos consistency, the image is then scaled to a predetermined size.

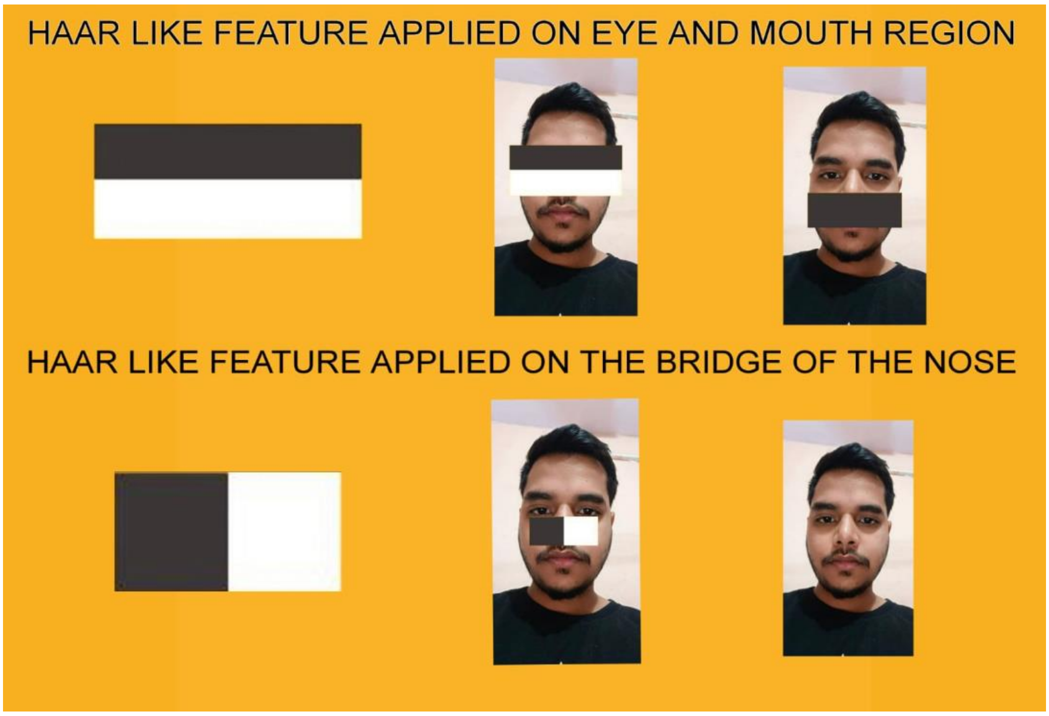

Fig 2. The illustration displays the haar characteristic, which may identify a human object's mouth, nose, and eyes.

ii). Selection of Haar Feature: To separate the object of interest from the background, the next step is to choose the best Haar-like features. To calculate Haar features, subtract the total of the pixels in a white rectangle from the sum of the pixels in a black rectangle. Haar features are areas of rectangular with varying intensities or colour values.

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 Pixel's /Number Of Dark Pixel's – Sum Of The Light Pixels /Number Of The Light Pixels

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

0.51 – 0.53 = - 0.02

A sample image with pixel values ranging from 0.0 to 1.0 is contained in the rectangle shown by Fig. 3.1 (right). Since all of the bright pixels are on the left and all of the dark pixels are on the right, the rectangle in Figure 3.2 (on the right) is a Haar kernel. The difference between the average of the pixel values in the lighter and darker areas is the formula used to do the Haar computation. If the difference approaches one, the haar feature will indicate an edge.

To obtain the sum of all the image pixels that lie within the lighter and darker regions of a certain Haar feature, you must first select the feature that interests you. Haar features are rectangular areas with different levels of darkness inside a picture. and then determine what makes them unique. In the event that the image contains an edge dividing the left and right light and dark pixels, respectively, the value of haar will now be closer to 1. As a result, we can declare that an edge has been detected if the haar value approaches 1. It is impossible to identify an edge in the previously given example since the haar value is so far from 1.

iii). Image Integral: Haar feature calculation is accelerated by using the integral image. The array is two-dimensional, with each entry representing the total of all the pixels in the original image to its left and above. It takes O(n * n) time to calculate haar features. Image Integral reduces this complexity to O(1), making the time of mapping-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

Fig4. Image Integral

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

Input Image

Fig 4. The creation of the integral image is depicted in Figure 4. A pixel in an integral image is the total of all the pixels in the original (input) image to its left and all the pixels to its top.

iv). Cascade Classification: Lastly, a cascading application of the learned weak classifiers is made via the cascade classifier. Multiple fragile classifiers make up each level of the cascade classifier, and an image is rejected as a negative sample if it fails at any of the stages. An image is categorised as a positive sample if it passes each stage.

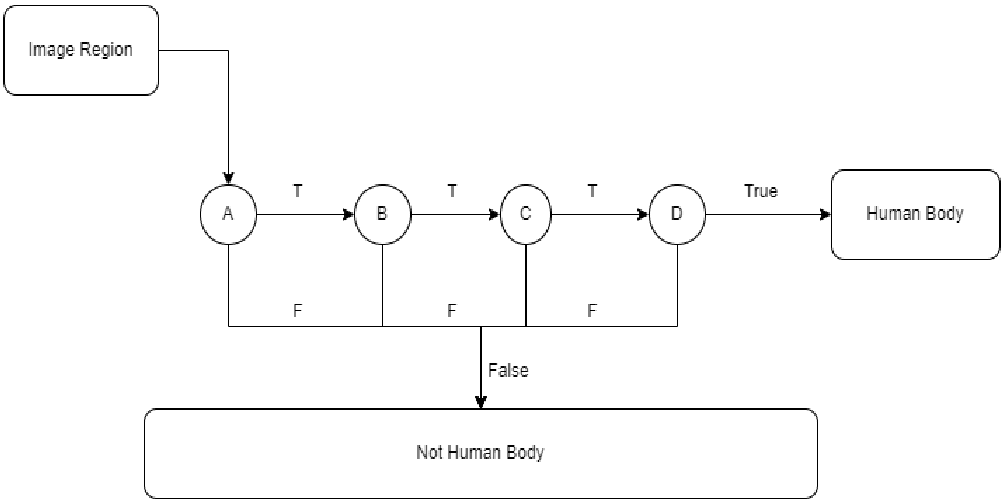

Fig 5. The Cascade Filter is seen in the above figure. An image is categorised as a positive sample if it successfully completes every step; if it fails at any point, it is discarded as a negative sample.

v). Detections: Using the cascade to find the object in the picture is the last stage. In order to identify the object, the algorithm uses the classifiers to scan the image at various scales and locations. The object's location is returned if it is found.
A machine learning model uses a classifier to detect objects in photos. The classifier is taught using a large number of images.

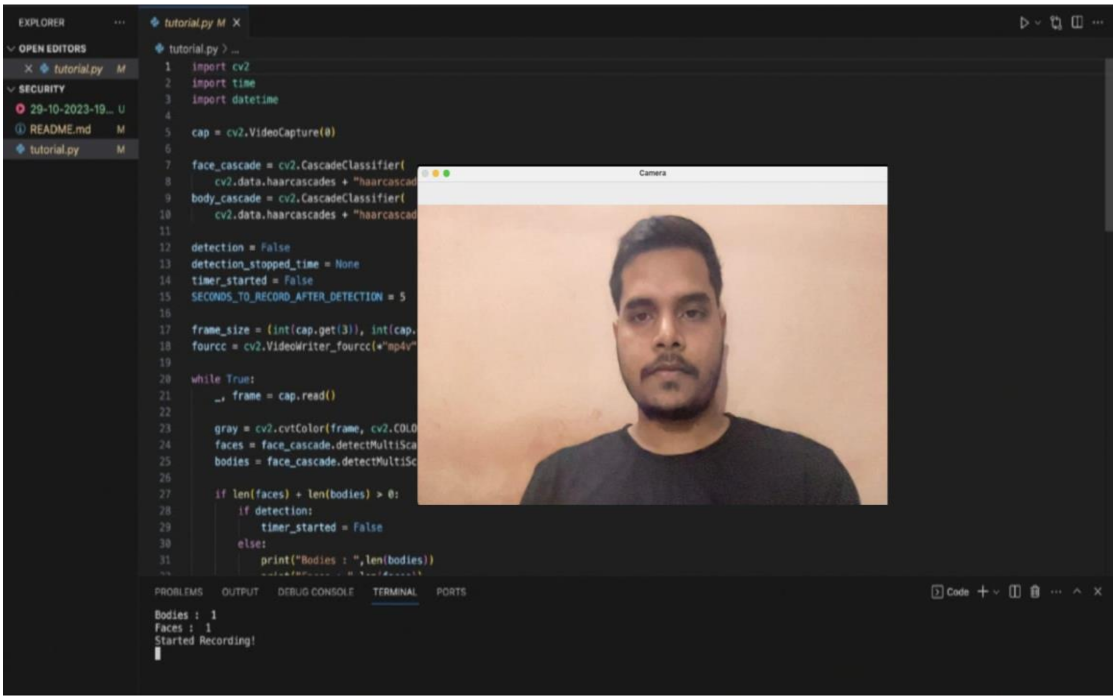

Fig 6.

Figure 6 illustrates how the system functions. The system will begin recording automatically if it detects a human item; otherwise, it will either cease recording (if it has already begun) or begin no recording at all.

3) Recording Process : The camera's recording feature is initiated automatically by OpenCV Libraray as soon as a suspicious object is identified using the HaarCascade Algorithm. Because we only record when an object is recognised thanks to this function, the storage demand is subsequently quite low.

4) Security Alert System : The owner receives a notification as soon as the human object is found. The message was sent to the owner securely and safely by utilising AWS's SNS functionality. The fact that this process is looping allows it to continuously and without pause detect objects.

Building, scaling, and managing applications that distribute alerts to multiple endpoints or devices is made possible by the social networking service (SNS) functionalities that AWS provides. Among AWS SNS's salient characteristics are the following:

Topics: Using AWS SNS, customers can establish topics—logical access points with message distribution and reception capabilities. All subscribers of a topic get messages sent to it, and each topic may have more than one subscriber.

a) **Subscriptions:** In order to get notifications via email, SMS, HTTP, Lambda, or mobile push notifications, users can subscribe to a topic.

b) **Mobile push notifications:** Devices running Windows, iOS, Android, and Fire OS can receive mobile push alerts over AWS SNS. With the SNS mobile push API, developers may leverage push alerts to devices.

c) **Fanout:** Fanout—the capacity to transmit a single message to several endpoints or subscribers simultaneously—is supported by AWS SNS. This makes it possible for communications to be delivered effectively and dependably.

d) Filtering: Sender identity, subscription attributes, message contents, and other attributes can all be used by users to filter messages using AWS SNS. This enables developers to notify particular users in a tailored manner according to their interests or preferences.

e) Encryption: To guarantee the security of messages delivered between endpoints, AWS SNS offers encryption options. Users have the option to utilise AWS-managed encryption keys or encrypt messages using their own keys. All things considered, AWS SNS offers a strong and adaptable sending's platform notifications across a variety of and devices channels, making it a crucial instrument for developing contemporary applications.

How the system is comparatively efficient with accurate?

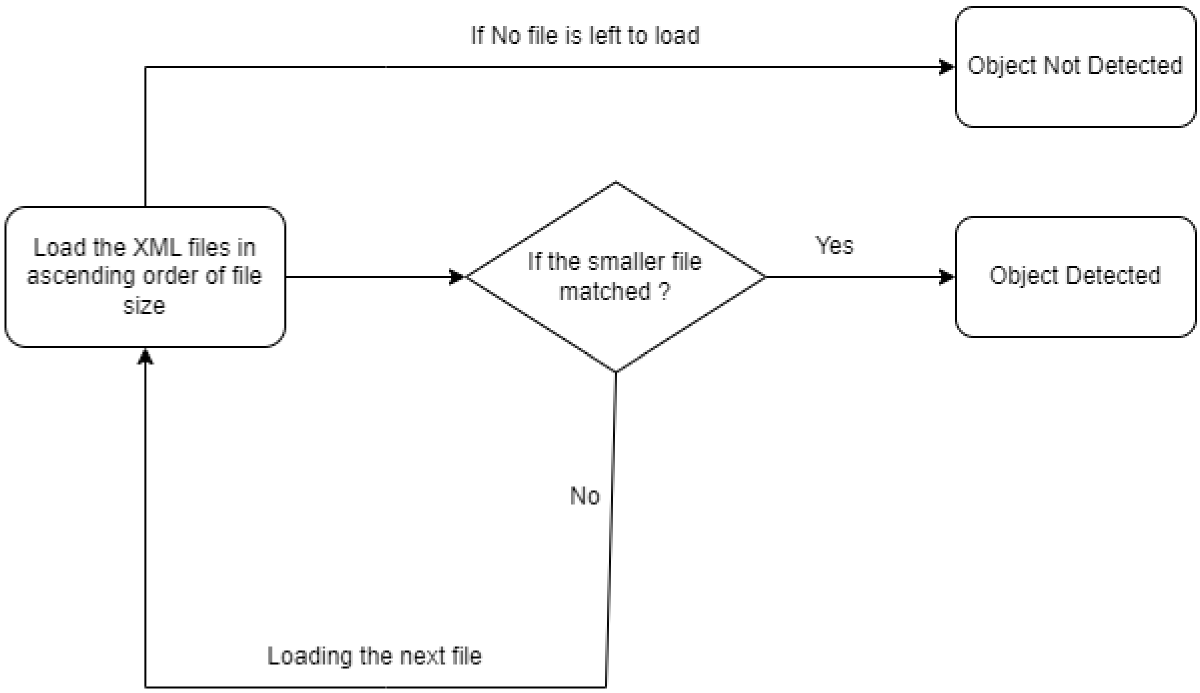

Fig 7. The system's relative efficiency and accuracy are demonstrated in the figure.

Figure 7 illustrates the system's relative improvements in accuracy and efficiency. The systems will Aquire the XML file in descending order of file size. The smallers file are handled initially. The machine won't proceed to match the other XML files if the smaller file matches and it finds body part of the Human. The systems will only matches the larger XML files if a file cannot be matched. Finally, if there are no files remaining to process, this indicates that the human item was not picked up.

At all costs, the algorithm should be fast and accurate, and it shouldn't crash in the middle. Because of this, any security system may experience a malfunction when the detection system becomes slower due to the high requirements and speed of the algorithm. Instead of this, there should be very high precision because it is crucial that no object pass without being identified at all costs. In order to ensure that nothing escapes undetected, we are employing the Haar cascade object detection technique, which differs from the default object detection approach [10]. Additionally, we are adding additional object part detection to the cascade.

However, doing this means that we'll need a lot of XML files, which will lengthen the duration because loading them all takes time. In order to address those issues, we have set up the detection mechanism according to the size of the XML files.

Parameters considered for the system to work

1. **Haar cascade classifier :** System makes use of the AdaBoost Algorithm, which is thought to be the greatest for quickly detecting images when combined with Integral Image.
2. **Brightness and Background :** Images with strong contrast between the object being recognised and the backdrop are ideal for the algorithm's optimal performance. It could be challenging for the algorithm to identify the object accurately if it is more dark and bright in relation in the backgrounds.
3. **Xml trained file :** For mapping, the system makes use of the Most-trained XML file.

IV RESULTS

The aforementioned parameters have been taken into consideration when conducting the research. Eighteen of the twenty human objects that we have seized in various ways have been identified by the system. The other two, who are undetected, have their bodily parts covered with clothing. We came to the conclusion that the system's accuracy is 90% in this fashion.Additionally, only when there is appropriate contrast between the human object and the background can the algorithm identify a human object. For the camera to identify images and function well, it needs to be able to distinguish between the human object and the background.

The system arranges XML files in ascending order of file size. For each part of the human body, the system uses training XML files, which enables the system to recognise images. As a result, the system becomes quicker and more effective.

IV. CONCLUSION

Conclusively, Security applications utilising real-time item tracking and identification may be effectively achieved by integrating OpenCV and Haar cascades into a smart security camera system. However, the quality of the video stream, the processing power of the computer, and the accuracy of the cascade classifier all affect how well the system performs.

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.68696872.

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

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

[6] M. David Allen , Joe Walkenhorst, Amazon Simple Notification Service: The Missing Piece of the Cloud Puzzle. The usage of SNS as a message system for cloud-based applications is covered in this study.

[7] Liu, Y., and Zhou, Y. (2018). OpenCV-based intelligent monitoring system design and implementation. Physics Conference Series Journal, 1013(1), 012080.

[8] Wang, J., Li, Z., and Li, S. (2019). Real-time tracking and detection of suspicious activities using a smart camera system. IEEE Access (2007) 7 (118634–118645).

[9] Li, B., Peng, C., and Zhang, Z. (2019). study on an OpenCV-based intelligent security monitoring system. Conference Series, Journal of Physics, 1282(4), 042065

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

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

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

[13] Sreenu, G., and Saleem Durai, MA. "Intelligent video surveillance: a review through deep learning techniques for crowd analysis." Published in 2016. 6.1 (2019): 1–27 in Journal of Big Data.

[14] Rutvik, Kakadiya, and colleagues. "Ai based automatic robbery/theft detection using smart surveillance in banks." 2019 will see the third International Conference on Aerospace, Electronics, and Communication Technology (ICECA). 2019 IEEE.

[15] Chibelushi, C.C., Moniri, M., and Sedky, M.H. (2005). List of intelligent video security systems for business use. Published in 2005 by IEEE Conference on Advanced Video and Signal Based Surveillance, pp. 638–643. In IEEE.

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