International Journal of Research in Engineering, Science and Management
Volume 7, Issue 2, February 2024
https://www.ijresm.com | ISSN (Online): 2581-5792

A Survey on Cypher Cam using IoT

Adarsh Hipparagi[1*], Anand Tippanna Ambiger[2], Arun Lachchappa Ambiger[3], Basavaraj Shrishail Halyal[4], Shiddalingesh Hiremath[5]
[1,2,3,4]Student, Basaveshwar Engineering College, Bagalkote, India
[5]Assistant Professor, Basaveshwar Engineering College, Bagalkote, India

***Abstract*: Security becomes one amongst the foremost necessities in our lives now a days, however criminal activities are still at large with criminals unable to be persecuted without eligible proofs of their misdeeds. Cypher camera is one in every of the higher solutions. Surveillance camera functions can be enhanced by adding algorithms that may identify objects. Frame difference is an algorithm to spot objects motion. Background subtraction is one of the methods suitable to further improve frame differences thus increasing its effectiveness and precision, so techniques are required to spot a face which must be quick and sufficiently enough to figure in real time. But there are difficulties within the execution of face identification in low lighting condition. Local Binary Patterns Histogram algorithm is going to be used for identifying face. Face detection from a digital image or video stream is employed often for various purposes but sometimes a system detects an object or area as a face where there's no face in the slightest degree. we use Haar-based algorithm for detecting the faces.**

***Keywords*: Haar based algorithm, LBPH, Frame difference method.**

1. Introduction

Cypher Cam is a python-based application, which is graphical user interface (GUI) that uses icons and menu's in order to manage interaction with the system. The objective is to develop a system that monitors the area in which it is being implemented. It uses machine learning techniques and artificial intelligence to provide a robust and secure monitoring solution. Cypher Cam Works on any operating system which uses camera as hardware with other functionalities like LBPH and Haar-based algorithms which makes us to implement all the features. Cypher Cam ensures real-time threat detection, facial recognition, and efficient video analysis, making it a valuable asset for various applications, from home security to large-scale industrial environments.

2. Literature Survey

Paper [1] describes an application named "Smart Surveillance System". Surveillance cameras are widely used record videos of its surroundings in order to provide security in every place which demands security. If we wish to find the presence of a particular person within the surveillance video, we have to watch the complete video. This is a very tedious task. Sometimes it may even happen that the target person may go unnoticed in the video. To overcome these problems, we have proposed a system named "A Smart Surveillance System", which uses machine learning approach to detect and recognize the target person in the video.

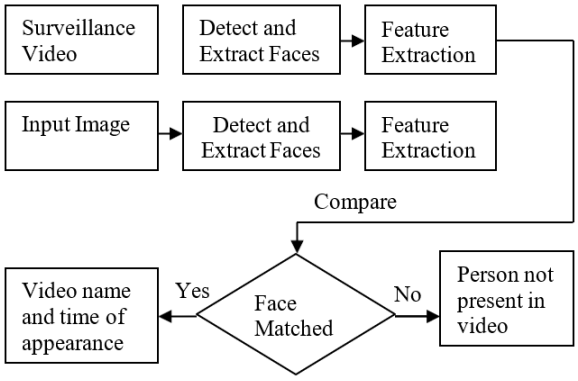

Fig. 1. Working of the software 'Smart Surveillance System'

Paper [2] describes about how the object detection is based on probabilities of bounding boxes and it is a brute force method of scanning every image only once and processing it to get results. The trained datasets are stored as in weights and cfg files. So, we thought, what if when humans are detected, we make the same network load files based on human activity, that is comparing the resulting human found to be rescanned by the same system but with different weights. So, we set out to implement this system with a goal to process raw footage files and detect any actions or activity that is going to occur

Fig. 2. Application to draw individual bounding boxes for the action to be recorded

*Corresponding author: ad2072002@gmail.com

A. Implementing Action Recognition

In the program the same network had to be fed to different configuration files. A recursive method was used for this. In our code we have to separate options for initiating the model. By using a second option we would use the same network with different configuration to process the activity in the function. We then code the program to search for the bounding box in the image. In a single image the various objects are detected like person.

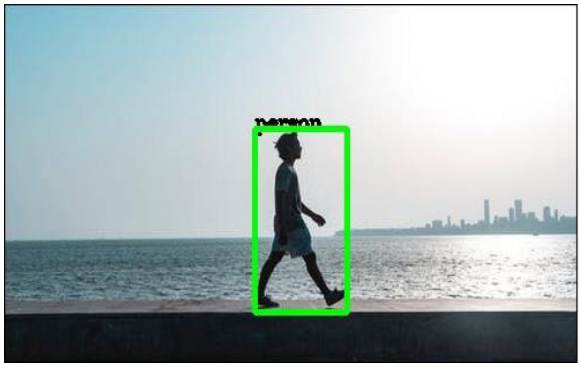

Fig. 3. Person being Recognized by Yolo [3], [4]

In Paper [3], there are four components in the technique proposed. First, is to generate Haar-like features that are digital image features used in object recognition. Second, is the introduction of a new image representation called the Integral Image which allows the features used by the detector to be computed very quickly. The third component is a simple and efficient classifier which is built using AdaBoost learning algorithm to select a small number of critical visual features from a very large set of potential features. The fourth component is a method for combining classifiers in a cascade which allows background regions of the image to be quickly discarded while spending more computation on promising face-like regions.

Methodology: Every face detection algorithm has different operation and procedures to detect a face. This section explains the components of both algorithms and step by step towards its function. First, is the face detection using Haar-like features, and second, is the face detection using Local Binary Pattern features.

Haar-like Features:

The features that Viola and Jones used are based on Haar wavelets. Haar wavelets are single wavelength square waves (one high interval and one low interval). In two dimensions, a square wave is a pair of adjacent rectangles, one light and the other is dark. This method is quite similar to binary method that separates the obvious different.

The presence of a Haar-like feature is determined by subtracting the average dark-region pixel value from the average light-region pixel value. If the difference is above the threshold (set during learning), that feature is said to be present. Examples of Haar-like features are shown in Figure 4.

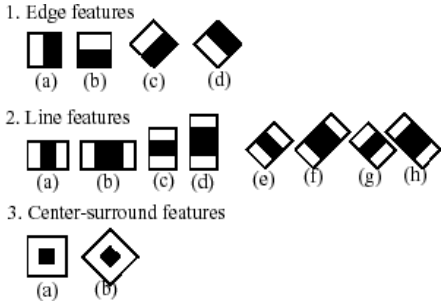

Fig. 4. Haar-like features

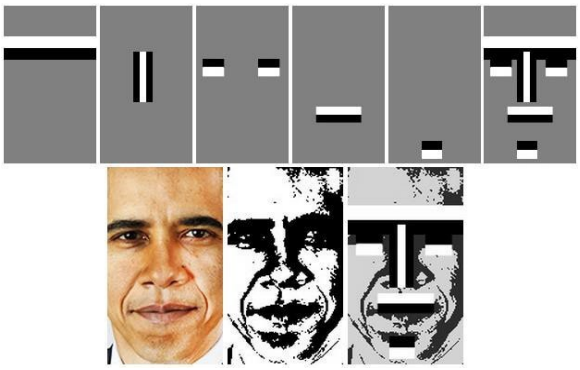

Fig. 5. Haar-like features application

Local Binary Patterns:

Local Binary Patterns (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighbourhood of each pixel with the value of the centre pixel and considers the result as a binary number. Due to its discriminative power and computational simplicity, LBP texture operator has become a popular approach to various applications. The LBP texture method has provided excellent results in various applications. Perhaps the most important property of the LBP operator in real-world applications is its robustness to monotonic gray-scale changes caused, for example, by illumination variations. Another important property is its computational simplicity, which makes it possible to analyse images in challenging real-time settings. The LBP texture analysis operator, introduced by Ojala et al, is defined as a grayscale invariant texture measure, derived from a general definition of texture in a local neighbourhood. The original LBP operator forms labels for the image pixels by thresholding the 3 x 3 neighbourhood of each pixel with the centre value and considering the result as a binary number. The histogram of these 28 = 256 different labels can then be used as a texture descriptor

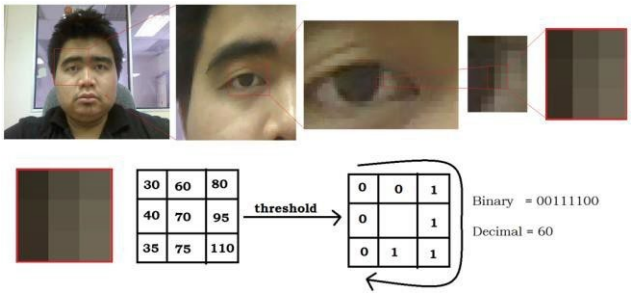

Fig. 6. The basic LBP operation

Table 1

Algorithm	Dataset	Detected Faces	Hit Rate	Detection Speed (ms)
Haar	Color FERET	976/1127	86.6%	235.4117
	MIT	1670/2000	83.5%	255.5048
	Taarlab	647/759	85.2%	231.5865
Overall		**3293/3886**	**84.7%**	**241**
LBP	Color FERET	1004/1127	89%	95.44924
	MIT	1779/2000	89%	101.8864
	Taarlab	674/759	88.8%	104.9335
Overall		3457/3886	89%	101

Paper [4], Frame differences is a method that is commonly used to detect an object through its motions. Therefore, the main objective of this paper is to propose a motion detection method that could be used with a live cam input from a video camera. The method is used to improve the effectiveness of security cameras like those used on Smarthome and CCTVs. The method that will be used are frame differences with the addition of background subtraction. These methods were chosen due to their high precision by comparing the amounts of pixels on each frame. The result of this research is hoped to be useful as a further reference on readers on the topic of motion detection

Methodology: Frame difference method is used to detect every motion that an object make that was captured by the camera. The frame difference algorithm takes every pixel within 2 frames to be compared sequentially and adds their differences on that block. This difference then was intended to be shown as a "motion" that resulted from a moving object that was caught by the camera. The equation shows the differential of pixel values, with Δn as the differential value on the nth frame and in as the pixel intensity on the nth frame.

$$\Delta_n = |I_n - I_{n-1}|$$

After the value of Δn is obtained, the motion of the object can be calculated by comparing the value of Δn with a threshold that has been stated. The value of the threshold is usually within 15% of the range that was used as the observed pixel intensity. So, if the range consists within 0-255 then, the threshold that will be used is rounded up to 40 [15]. Motion depicted as (Mn) can then be calculated by doing the operation per pixel with the equation.

$$Mn = \begin{cases} In, \Delta n \geq T \\ 0, \Delta n \leq T \end{cases}$$

Background Subtraction:

Background subtraction method models the background thoroughly from pixels by using different filters such as running average/approximate median filter or temporal median filter. The main objective of this method is to make a model of the background to be a reference on foreground detection. If the foreground on the t^{th} frame is shown as Ft, and pixel intensity and background values is depicted as It and Bt, then the value of foreground can be calculated using the equation.

$$Ft = \begin{cases} 1, & |It - Bt| > T \\ 0, & else \end{cases}$$

Paper [5], The authors successfully developed a reliable and scalable face recognition. Ready-to-use transmission system. Poor performance in the early stages of operations (especially the high false alarm rate) has led to research on more complex solutions. Artificial intelligence concepts such as neural networks and expert systems have been integrated into more computer-oriented systems.

Methodology: The Structural Similarity Index (SSIM) metric extracts 3 key features from an image: Luminance, Contrast and Structure. The comparison between the two images is performed on the basis of these 3 features.

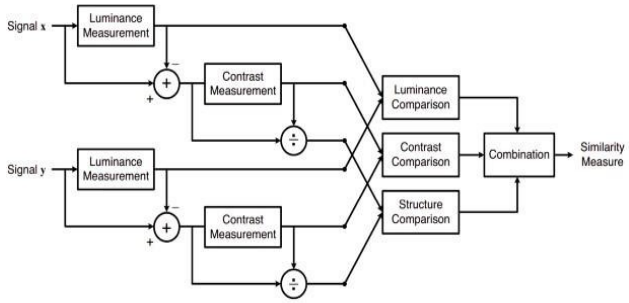

Fig. 7. SSIM algorithm block diagram

LBPH (Local Binary Pattern Histogram):

The LBPH uses 4 parameters: Radius, neighbors grid x and grid y. The first calculation step of LBPH is to create an intermediate image that better describes the original image by extracting facial features. To this end, the algorithm uses a sliding window concept based on radius and neighboring parameters. Now that we have the image generated in the previous step, we can use the parameters Grid X and Grid Y to divide the image into multiple grids, as shown in the following figure: predict what you want to meet, apply the same steps to the mark The histogram is compared with the trained model. This is how the function works.

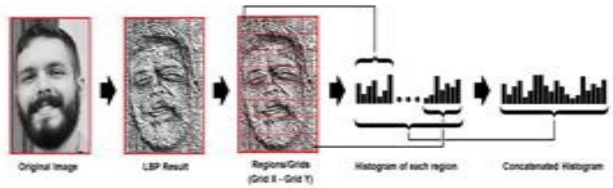

Fig. 8. LBPH model

Paper [6], In this paper the authors worked on abnormal behavior detection took a supervised learning approach. Diverse contributions have been made in the development of behavior recognizers for smart building surveillance applications. In automatic roaders, human surveillance, the

vehicle or human activities and behaviors are detected and recognized for monitoring and warning purposes, for detecting human behavior.

Methodology:

Capturing video:

OpenCV is an open-source python library that contains various functions for image and video operations. With OpenCV, we can capture a video from the camera. cv2.VideoCapture () method is defined to get a video capture object for camera. Create an infinite loop and use the read () method to read the frames using above created object. Cv2.imshow () method is used to show the frames in the video. Loop will break when the user clicks a specific key.

Motion detection:

In the proposed work, Motion detection is performed by using OpenCV and Pandas library. Captured videos are treated as a stack of pictures called frames. Different frames are compared to the static frame which has no movements. We compared two images by comparing the intensity value of each pixel. Firstly, we convert a color image into a grayscale image, then a gray-scale image is converted to GuassianBlur so that change can be easily found. After that difference between the static background and the current frame is found out. If we found to change between them is greater than 30 it will show white color. Then contour of the moving object.

Feature extraction:

Feature extraction is the process of identifying the important features of the data. It reduces an initial set of raw data to more manageable groups for processing. So here, we will start with reading colored images, using the imread () method. Using the shape function, the shape of the image is found out. Suppose the shape for the image is 375*500. So the number of features will be 187500. If you want to change the shape of the image that is also can be done by using reshape function from NumPy where we specify the dimension of the image. For this scenario, the image has a dimension (375, 500, and 3). This three represent the RGB value as well as the number of channels. Now we will use the previous method to create the features. The total number of features will be for this case 375*500*3 = 562500. This colored image has a 3D matrix of dimension (375*500 * 3) where 375 denotes the height, 500 stands for the width and 3 is the number of channels. To get the average pixel values for the image, we will use a for a loop. Now we will make a new matrix that will have the same height and width but only 1 channel. To convert the matrix into a 1D array we will use the Numpy library. CT is found out.

Classifier:

An important step for surveillance activity recognition is to detect, localize, and track each individual throughout the video stream. This task is not feasible with object detectors that are trained on general categories of data. For this purpose, we fine-tuned a lightweight CNN model for human detection with new data and enabled it to work in a changing surveillance environment. It is superior to state- of-the-art methods, its effectiveness is verified from experiment. This architecture makes our system able to achieve LSTM-level accuracy while being more efficient than the LSTM.

Paper [7], In this paper SSD and MobileNets based algorithms are implemented for detection and tracking in python environment. Object detection involves detecting region of interest of object from given class of image. Different methods are – Frame differencing, Optical flow, Background subtraction. This is a method of detecting and locating an object which is in motion with the help of a camera. Detection and tracking algorithms are described by extracting the features of image and video for security applications. Features are extracted using CNN and deep learning. Classifiers are used for image classification and counting. YOLO based algorithm with GMM model by using the concepts of deep learning will give good accuracy for feature extraction and classification. Section II describes SSD and MobileNets algorithm, section III explains method of implementation, and section IV describes simulation results and analysis.

Object Detection and Tracking Algorithms:

Single Shot Detector (SSD) algorithm:

SSD is a popular object detection algorithm that was developed in Google Inc. It is based on the VGG-16 architecture. Hence SSD is simple and easier to implement.

A set of default boxes is made to pass over several feature maps in a convolutional manner. If an object detected is one among the object classifiers during prediction, then a score is generated. The object shape is adjusted to match the localization box. For each box, shape offsets and confidence level are predicted. During training, default boxes are matched to the ground truth boxes. The fully connected layers are discarded by SSD architecture. The model loss is computed as a weighted sum of confidence loss and localization loss. Measure of the deviation of the predicted box from the ground truth box is localization loss. Confidence is a measure of in which manner confidence the system is that a predicted object is the actual object. Elimination of feature resampling and encapsulation of all computation in a single network by SSD makes it simple to train with MobileNets. Compared to YOLO, SSD is faster and a method it performs explicit region proposals and pooling (including Faster R-CNN).

MobileNets algorithm:

MobileNets uses depthwise separable convolutions that helps in building deep neural networks. The MobileNets model is more appropriate for portable and embedded vision-based applications where there is absence of process control. The main objective of MobileNets is to optimize the latency while building small neural nets at the same time. It concentrates just on size without much focus on speed. MobileNets are constructed from depthwise separable convolutions. In the normal convolution, the input feature map is fragmented into multiple feature maps after the convolution. The number of parameters is reduced significantly by this model through the use of depth wise separable convolutions, when compared to that done by the network with normal convolutions having the same depth in the networks. The reduction of parameters results in the formation of light weight neural network.

Methodology:

Object detection:

Frame differencing Frames are captured from camera at

regular intervals of time. Difference is estimated from the consecutive frames. Optical Flow This technique estimates and calculates the optical flow field with algorithm used for optical flow. A local mean algorithm is used then to enhance it. To filter noise a self-adaptive algorithm takes place. It contains a wide adaptation to the number and size of the objects and helpful in avoiding time consuming and complicated preprocessing methods.

Object tracking:

It is done in video sequences like security cameras and CCTV surveillance feed; the objective is to track the path followed, speed of an object. The rate of real time detection can be increased by employing object tracking and running classification in few frames captured in a fixed interval of time. Object detection can run on a slow frame rate looking for objects to lock onto and once those objects are detected and locked, then object tracking, can run in faster frame speed.

Paper [8], This paper aims to elaborate the various techniques in video surveillance, automated video analysis and insight generation. These techniques were used to build the Software System for Automated Surveillance for Academic Institution's Campus premises. In this paper they have discussed data collecting, storing and analysis technique for CCTV Camera Surveillance. They found that for feature extraction and tuning system to work hand in hand with deep learning model haar cascade was useful. By using Image Mosaicing technique images could be stitched and camera position limitation were removed. Thus, among many methods of collecting camera input they found IP based camera technique on distributed network is useful and CNN model was useful for detail analysis.

CNN (Convolutional Neural Network):

The convolutional neural network is the most effective and efficient way to classify images, CNNs can be trained on a large-scale database and then its learnings can be enhanced and used in other task with less amount of training data. The working of CNNs is inspired by the human brain, CNNs try to mimic the working of human brain using small units called perceptron which are analogous to neurons in human brain. The perceptron can accept input and produce an output, input is to the perceptron is associated with a weight and these weights can be changed.

In CNNs the initial task is to extract features from the images. For this task the CNNs use filters. images are passed through these multiple filters creating new images. The filters extract small features of the image and combines small features to detect large features in further layers. This process may reduce the resolution of the images. The CNN contains various layers namely input layer, hidden layer and output layer. The input layer accepts the input image and the output layer gives us the required output vector. Training a CNN can take several days or weeks depending on the amount of training data available. There are publicly available pre-trained models that are trained by the research teams.

The architecture of a CNN completely depends on the domain area, the architecture can be created by domain expert. The CNN can contain multiple hidden layers but increase in hidden layer increases its complexity. In this proposed system we will use CNN for image classification of students. This system will be trained on student dataset containing at least 400 images per student. The CNN classifier model will be trained and saved on a different much powerful machine and later will be imported into the attendance system module for classification, this will allow us to run the program on a much power efficient or commodity hardware.

While the system is taking continuous real time video input from the CCTV cameras, we have to detect human faces, we could have used CNN in this stage but CNN would have been resource intensive and might not be able to detect multiple faces in single frame. To resolve this issue, we will use the Haar Cascade Classifier as it is more efficient and effective for object detection and face detection.

3. Conclusion

In conclusion, the survey of the above-mentioned papers provides valuable insights into the diverse methodologies and techniques employed in the field of surveillance systems and object detection. These papers collectively contribute to the advancement of security solutions, addressing various aspects such as face detection, motion detection, object recognition, and behavior analysis.

Through the implementation of machine learning algorithms, including Haar-like Features, Local Binary Patterns (LBP), and Convolutional Neural Networks (CNNs), researchers have demonstrated innovative approaches to automate surveillance tasks and enhance the efficiency of security systems. Additionally, advancements in object detection algorithms such as Single Shot Detector (SSD) and MobileNets offer real-time tracking and identification capabilities, catering to the evolving demands of security applications.

Furthermore, the integration of sophisticated metrics like the Structural Similarity Index (SSIM) for face recognition and feature extraction underscores the importance of robust and scalable solutions in surveillance environments. These advancements not only improve the accuracy and reliability of surveillance systems but also contribute to the scalability and adaptability required for deployment in diverse settings, from academic institutions to industrial facilities.

Overall, the survey highlights the multidisciplinary nature of surveillance system research, encompassing elements of computer vision, machine learning, and signal processing. By leveraging these technologies and methodologies, researchers continue to push the boundaries of what is achievable in terms of security, monitoring, and threat detection, paving the way for safer and more secure environments in the future.

References

[1] B. W. Balkhande, Deepak Dhadve, Pranalee Shirsat, Mayuri Waghmare, "A Smart Survillience System," in International Journal of Recent Technology and Engineering, vol. 9, no. 1, May 2020.

[2] Rishabh Paunikar, Shubham Thakare, Utkarsh Anuse, B. W. Balkhande, "Smart Surveillance System," International Journal of Engineering Applied Sciences and Technology, vol. 4, no. 12, pp. 494-496, April 2020.

[3] Kushsairy Kadir, Mohd Khairi Kamaruddin. Haidawati Nasir, Sairul I. Safie, Zulkifli Abdul Kadir Bakti, "A Comparative Study between LBP and Haar-like features for Face Detection Using OpenCV."

[4] A. M. Husein, Calvin, David Halim, Raymond Leo, William, "Motion detect application with frame difference method on a surveillance camera," Indonesia MECNIT 2018 IOP Conf. Series: Journal of Physics: Conf. Series 1230, 2019.

[5] G. Chandan, A. Jain, H. Jain and Mohana, "Real Time Object Detection and Tracking Using Deep Learning and OpenCV," 2018 International Conference on Inventive Research in Computing Applications, Coimbatore, India, 2018, pp. 1305-1308

[6] Neha Kardile, Rutuja Deshmukh, Vaibhav Kalhapure, Devidas Jaybhay, "Intelligent Video Surveillance System using Deep Learning," International Research Journal of Engineering and Technology, vol. 9, no. 5, May 2022.

[7] G. Chandan, A. Jain, H. Jain and Mohana, "Real Time Object Detection and Tracking Using Deep Learning and OpenCV," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2018, pp. 1305-1308.

[8] Ishan Kokadwar, Anurag Kulkarni, Sayali Khare, Vaibhav Limbhore, Swati Chandurkar, "Camera based Smart Surveillance System-Literature Survey," in International Research Journal of Engineering and Technology, vol. 7, no. 6, June 2020.