Object Tracking and Unusual Change Detection using Raspberry Pi

SRF Student: Bharath Y. P.

B. M. S. College of Engineering, Bangalore – 560019

Mentor: Prof. Sankar K. Pal

Indian Statistical Institute, Kolkata-700108

Object Tracking and Unusual Change Detection using Raspberry Pi

SRF Student: Bharath Y. P.

B. M. S. College of Engineering, Bangalore – 560019

Mentor : Prof. Sankar K. Pal

Indian Statistical Institute, Kolkata-700108

Abstract

Object tracking is the technique used in Video processing to keep track of the detections in the image which can continuously change its location over time. This technique helps us in identifying the moving object by assigning it a unique id, thereby differentiating itself from the other moving objects detected in frames of the video. Once the task of tracking the object is achieved, it is easy for us to store all the object's trajectory, their velocity, their acceleration. In order to smoothen its trajectory path, the noise in the measurement of the object's detected location can be taken into consideration and predict an even better trajectory path. Kalman filter is used to achieve this task of considering the noise in measurement and predict an even more accurate measurement of the object's location and thereby smoothening the trajectory's path. Then we came up with a new application, namely, unusual change detection in the frame, it can be defined by detecting when any of the moving objects in the scene has unusual movement in its path, velocity or acceleration. By continuously monitoring these values, a given frame can be marked unusual by observing the values of these parameters of the objects and check if they have crossed a threshold value. This technique is useful in analysing surveillance video wherein a user need not go through entire video to check for any unusual behaviour in the moving objects, but rather, look at only the frames which were marked by this technique. This can be implemented in real time by continuously processing the images obtained from a static camera. All the algorithms and techniques mentioned have been implemented using PYTHON programming language, Scipy (the library used for scientific computing and technical computing) and OPEN-CV (open source computer vision library). The code is being tested on several datasets like surveillance videos and optimization of the algorithm is being done simultaneously to improve the tracking algorithm. The Final objective is to implement this in real time using a laptop's camera or using a Raspberry Pi.

Introduction

Computer vision tasks include methods for acquiring, processing, analysing and understanding digital images and extraction of high-dimensional data from the real world to produce meaningful information out of it. Object tracking is an important task within the field of computer vision.

This task consists of three main steps [1]: a) detection of interesting points such as the center of the moving object, b) tracking of such points from frame to frame and c) analysis of the tracks to recognize their behavior. Even though these steps seem to appear simple and implementable, tracking of objects can be a difficult task due to the complex motion of objects, occlusions, noise in images etc. These problems can be tackled to some extent by pre-processing the input images, by using state of the art detection models and by using better algorithms which can establish an accurate one to one correspondence of points detected across frames.

Object detection is one of the tasks in the field of computer vision. The current state of the art model [2] which is used to achieve this task, not only detects and classify the objects in an image but also localizes them in the image. One of the detection algorithms which is used to detect objects in the image is by using a Background Subtraction technique in which the background is modeled using a mixture of Gaussians. [3]. Moving objects are then separated from the background by performing morphological operations on the masked image obtained from background subtraction algorithm in which the background and the foreground are separated in the form of a binary image which has well-defined contours of the objects and noises removed. Contour approximation algorithm [4] is then used to obtain the centers of the objects along with their bounding box attributes (height and width). These centers can be treated as points unique to the object for which point tracking is done.

With the knowledge of location of the detected objects, set of detections in a given frame can be associated with the new set of detections in the subsequent frame based on the cost function (Euclidean distance between the detected points as cost, Histogram matching of the object enclosed by the bounding boxes, etc.) by using an assignment algorithm. One such algorithm is called Hungarian Assignment algorithm [5] (also known as Munkres Algorithm [6]). The detected objects are then tracked from frame to frame with these above techniques.

The trajectory can then be made more accurate with the help of Kalman filters [7]. With the available trajectory information of the point, velocity of the detected point is measured in terms of pixels/frame (analogous to meters/second) by calculating the pixel displacement of the detected point in successive frames, and similarly, acceleration of the point is calculated by calculating the difference in the point's velocity in successive frames.

Here, unusual change is defined when there is a sudden change in the movement of the object. Sudden change is observed when the acceleration of the object crosses a predefined threshold value.

Raspberry Pi is a modern minicomputer which is capable of handling tasks related to computer vision with the help of a camera module. Using the camera module, problems pertaining to real time can be implemented.

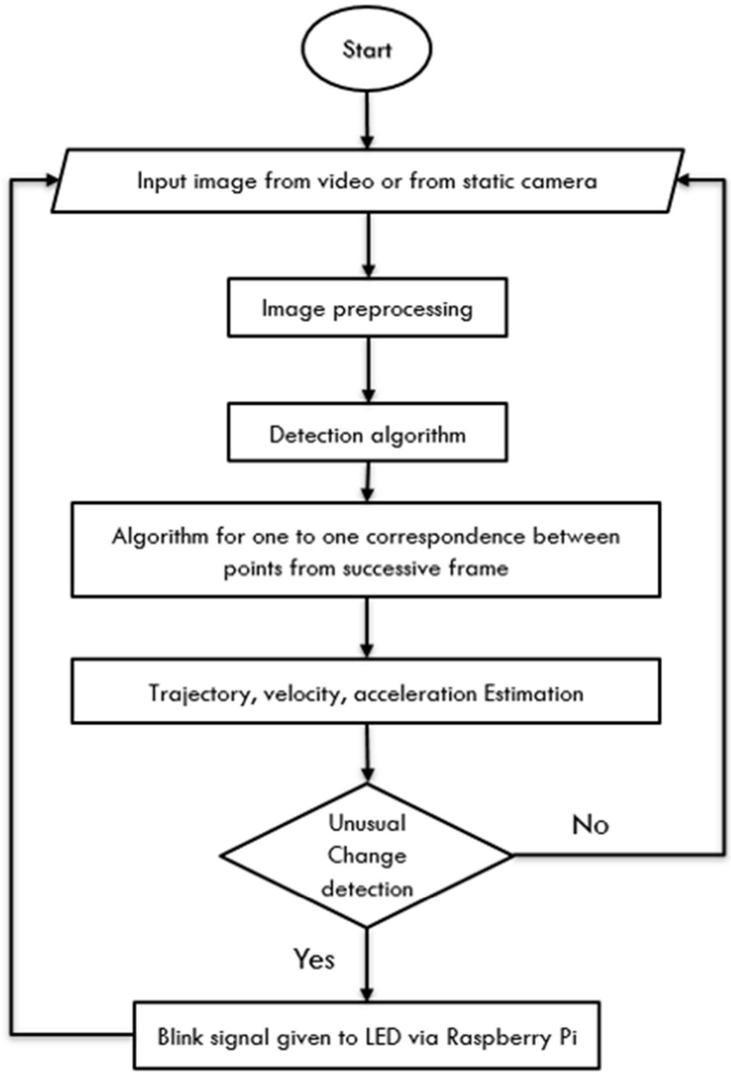

Figure 1: Flow chart of the system

Objective

Object tracking and unusual change detection in a given sequence of frames obtained from the video or from Pi camera module by using Raspberry Pi.

Methodologies

1. Background Subtraction:

It is a technique used in image processing and computer vision to separate the foreground objects from the background. We have used Mixture of Gaussians method by modeling each pixel by a mixture of Gaussians and uses an online approximation to update the background model. With the background model, a given pixel can be classified as foreground or background based on the probability value given by the Gaussian model for that pixel.

"cv2.bgsegm.createBackgroundSubtractorMOG()" function from OPEN-CV library is used to model each pixel of the image obtained from the video. The output of this function is a binary image which represents foreground pixels as white and background pixels as black.

Fig. 2(a) Input image with moving objects

Fig. 2(b) Binary mask image with foreground and background separated

2. Image processing:

Image processing techniques like image blurring, morphological transformations are done to remove the noise and to obtain a well-defined contour

The Noise from the binary image is removed by convolving the image with a 3x3 averaging filter (blurring) followed by erosion operation. Dilation operation is then performed on the blurred image so as to group foreground pixels of the same object and form a well-defined contour as much as possible in the masked binary image.

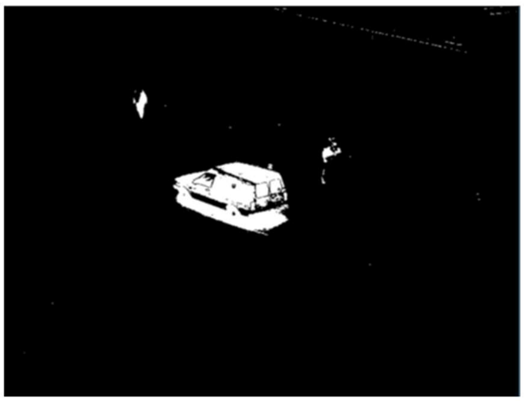

Fig. 3(a). Before processing

Fig. 3(b). After processing

3. Contour approximation algorithm:

"cv2.findContours()" function is used to get the center location or center point (x,y) of the contour (objects) and bounding box(width, height) of all the detected contours. The location of the contour's center and the bounding box of the contour simply describes the position and size of the detected object in the frame obtained from the video or camera.

Fig. 4(a). Binary image with well-defined contour

Fig. 4(b). Coordinates and the bounding boxes which were obtained from contour approximation algorithm

4. Attributes:

The detected objects are then identified with a unique id along with attributes like location (x, y), velocity, acceleration, height and width (h, w), trajectory (an array containing the object's previous detected locations in the past frames).

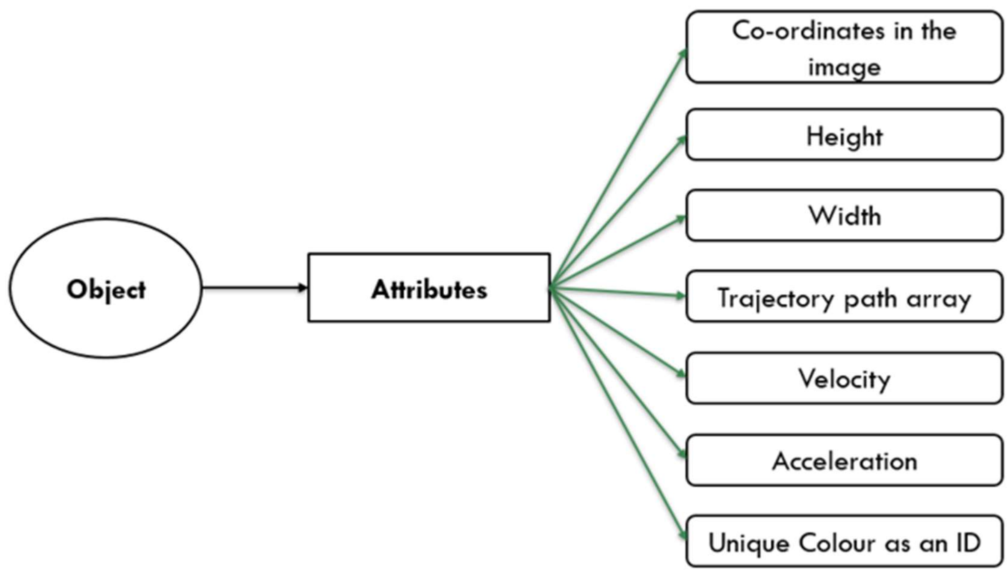

Fig. 5. Object and its attributes.

5. Assignment Process:

The object's location is updated by comparing its position with every detection in the new frame by associating it with the one which yields the least Euclidean distance. This problem is easy if the number of objects detected is only one. It becomes complicated as the number of detected objects increases. This is solved using the Hungarian algorithm (also known as Munkres algorithm) by evaluating the cost matrix. The cost matrix is defined as the matrix containing the Euclidean distance values between the detected objects at frame t-1 and the detected objects at frame t. All the objects which got paired up with new detections were updated (attribute's values like position and velocity are recalculated) and the remaining set of detections which were not paired up with any the old detections are treated as new objects. These steps are followed for every iteration of the frame.

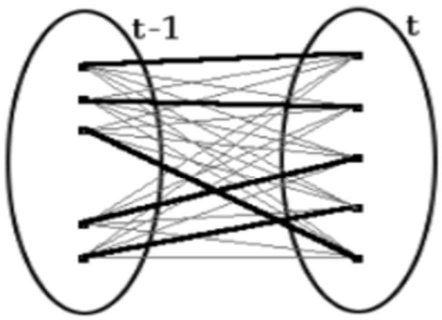

Fig. 6(a). Establishing one to one correspondence for the detected points in subsequent frames

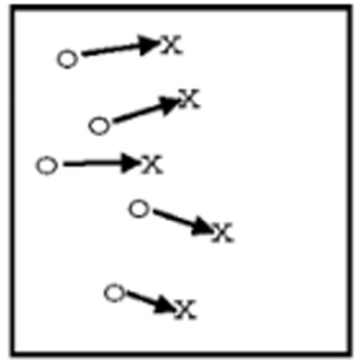

Fig. 6(b). Updating every object's attribute to the new detections.

6. Trajectory:

The detected objects have an attribute called trajectory. The trajectory attribute here is an array containing the previous locations at which the object was detected in the past frames. This trajectory array is used to calculate the velocity of the object and the acceleration of the object at different instances.

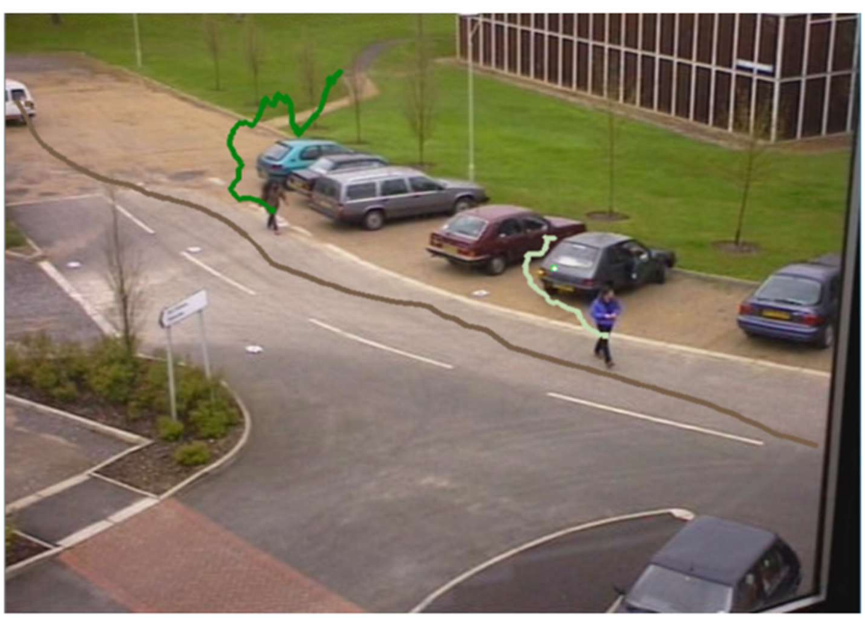

Fig. 7. Plot of trajectory of the moving objects with colour being represented as unique id.

7. Kalman filter:

Kalman filter, also known as linear quadratic estimation, is an algorithm that uses a series of observed measurements over time which may contain noise in the measurements and produces a new value for those observed variables which tend to be more accurate than the those based on the single measurement alone. The detected object's trajectory which contains a series of observed locations(x,y) is fed to this algorithm, which in turn produces a more accurate and smoothened trajectory of the object.

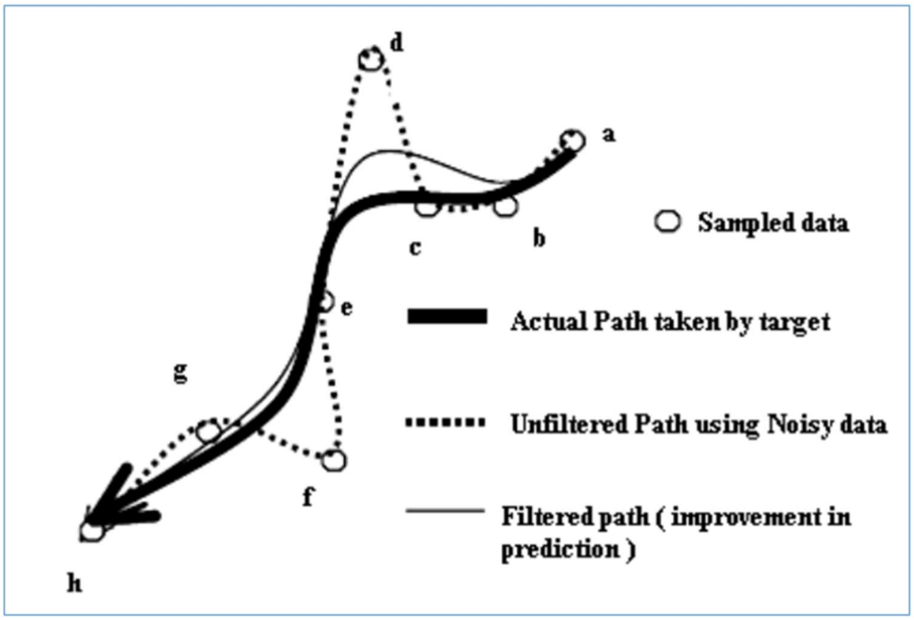

Fig. 8. Application on Kalman filter

8. A trajectory of the object without using the Kalman Filter:

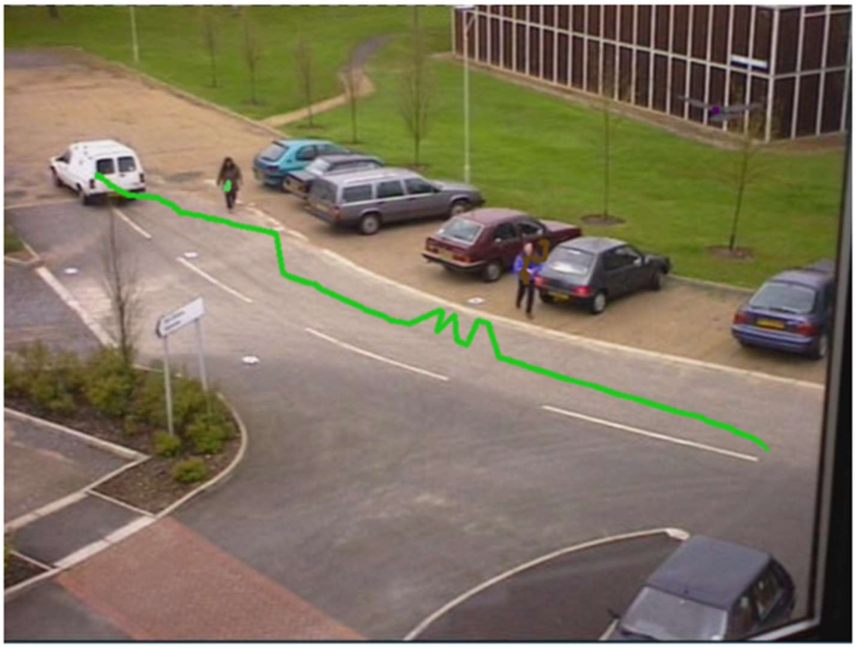

Fig. 9(a). Plot of trajectory without using Kalman filter.

9. The trajectory of the object using the Kalman filter

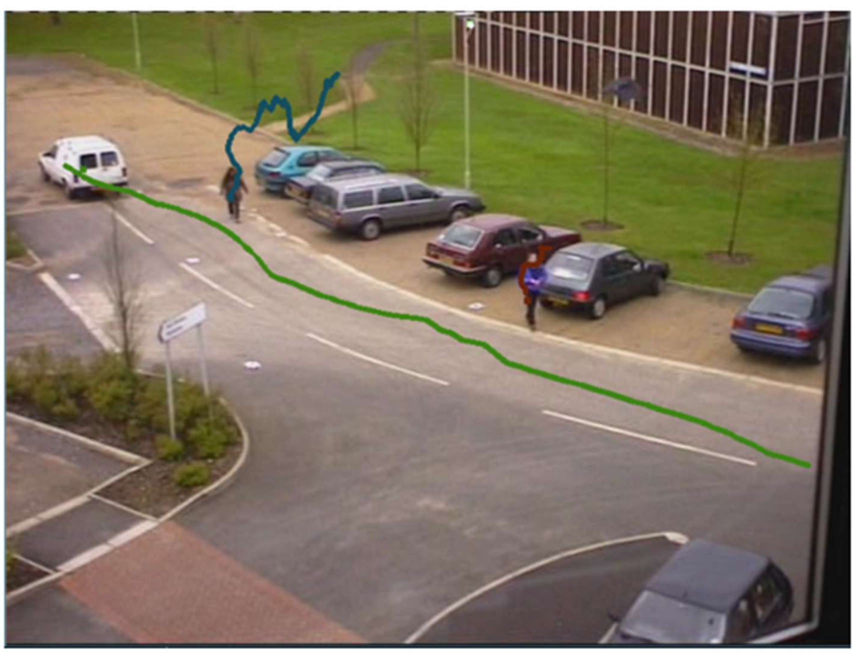

Fig. 9(b). Plot of trajectory using Kalman filter

10. Path, Velocity, Acceleration of the object:

With the trajectory (previously detected locations) of the object being known, the velocity of the object is obtained by subtracting the current location from the previous location. Similarly, the acceleration of the object is obtained by subtracting the current velocity with its previous velocity. The values of these attributes are updated after every iteration of the frame.

Fig. 10. Velocity and Acceleration of the objects being tracked.

11. Unusual change detection:

Unusual change in the frame is defined when there is a sudden change in the movement of the object. By continuously monitoring the acceleration attribute, the given object is marked unusual by observing the values for these attributes of the objects and check if they have crossed the predefined threshold value. The predefined threshold value is set accordingly based on the positioning of the camera. Here, the threshold value set for the acceleration parameter and is equal to 0.5.

Fig. 11(a). No unusual change since the object's acceleration is below 0.5 (threshold)

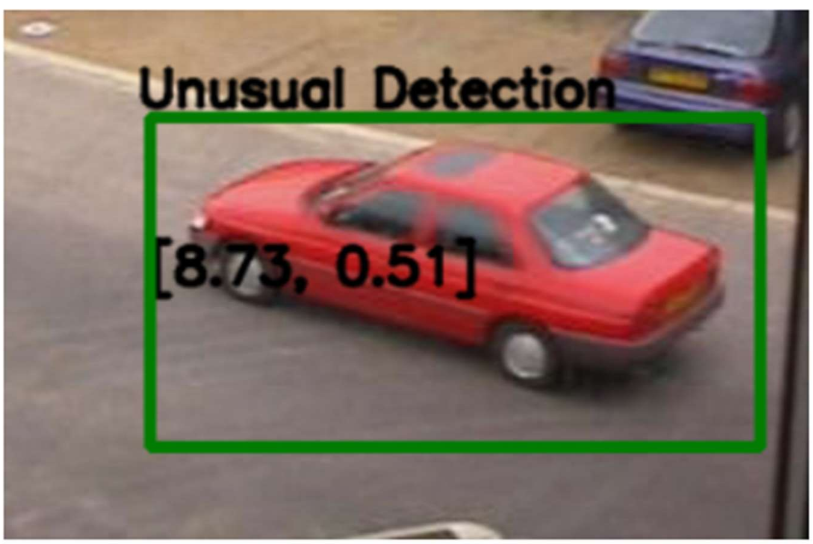

Fig. 11(b). Unusual change detected since the object's acceleration is greater than 0.5 (threshold)

11. Implementation in Raspberry Pi.

The Raspberry Pi has a Camera module connected to it which is capable of supplying the real-time images. GPIO pin 7 in Raspberry Pi is used as an OUTPUT PIN to which the anode of the Led is connected and the cathode is shorted with the GROUND PIN of Raspberry Pi.

The code runs on python which uses OpenCV Library for image processing.

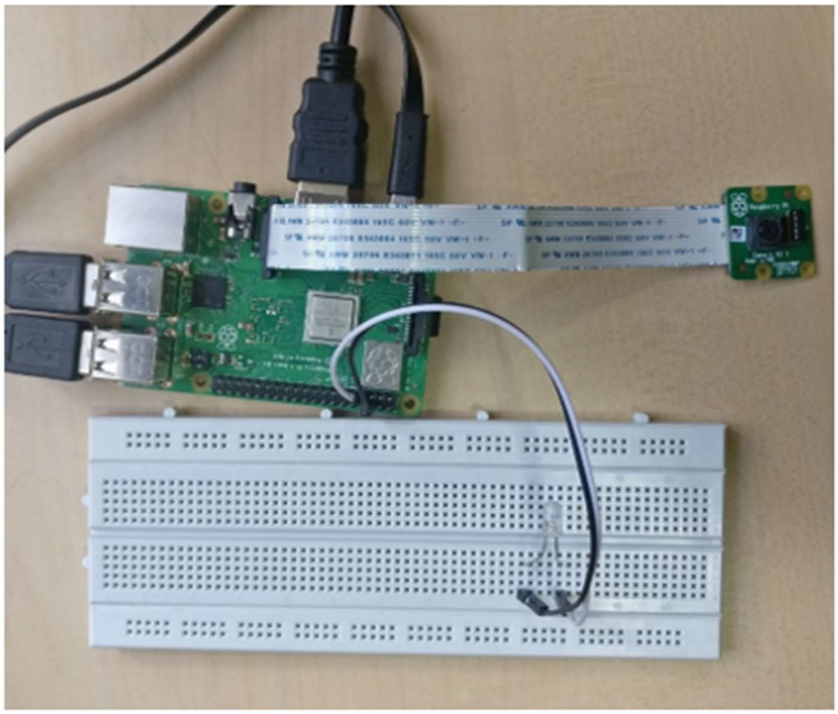

Fig. 12. Raspberry Pi setup

Results and Discussion:

The moving objects were detected and localized in the frame using background subtraction and contour approximation algorithms. The trajectory of the object is estimated and made more accurate by using the Kalman filter. The velocity and acceleration of the objects detected in the frame are measured and updated for each iteration of the frame.

The code to implement these algorithms and techniques were written in Python programming language by leveraging the predefined functions present in OpenCV library, Scipy

Unusual activity was observed when the values of the attributes crossed the predefined threshold. This was implemented in real time using Raspberry Pi wherein the blink of the LED signified an unusual activity taking place.

Future Scope:

Object tracking has several applications in the field of Robotics, Surveillance systems, Traffic control, Gesture recognition, etc. Tracking the objects in the scene becomes more difficult as the conditions (the number of objects, occlusion, lighting, background, etc.) changes. Extensive research is being done on this to improve the accurateness of the algorithm in predicting the trajectory of interesting points. Deep learning techniques have a lot of potential to solve tasks related to computer vision since it was used to solve many problems in Pattern recognition with better results than the state of the art algorithms which didn't use Deep Learning.

Raspberry Pi device is extensively used nowadays at places where size and portability are of concern. It can run programs of several languages (c++, python, etc.). The solutions to the computer vision problems can be implemented on Raspberry Pi and use it to solve real-world problems such as traffic management, vehicle over speed detection, surveillance systems, etc.

Acknowledgement

I would like to thank Indian Academy of Sciences for providing me an opportunity at Indian Statistical Institute, Kolkata. I thank my mentor Prof. Sankar K. Pal for his guidance and for the lessons learned from him.

I also thank Dr. Debarti Bhunia Chakraborty for her guidance and support throughout this internship.

References

[1] A. Yilmaz, O. Javed, M. Shah, “Object tracking: A survey”, *ACM Computing Surveys*, Vol. 38, Issue 4, Article No. 13, Dec 2016.

[2] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *CoRR* abs/1506.02640 (2015)

[3] P. Kaewtrakulpong, R. Bowden, “An Improved Adaptive Background Mixture Model for Real Time Tracking with Shadow Detection”, in *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems*, 2001.

[4] M. W. Ren, J. Y. Yang, and H. Sun, "Tracing boundary contours in a binary Image[j]" *Image and Vision Computing*, vol. 20, pp125-131, 2002

[5] KUHN, H. W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2* (1955), 83–97

[6] MUNKRES, J. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics 5*, 1 (March 1957), 32–38.

[7] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35-45,1960.