DWIT COLLEGE

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

Institute of Science and Technology

OBJECT DETECTION AND TRACKING USING HISTOGRAM OF ORIENTED GRADIENTS (HOG) BASED KALMAN FILTER

A PROJECT REPORT

Submitted to

Department of Computer Science and Information Technology

DWIT College

In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology

Submitted by

Ravi Adhikari

August, 2017

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

SUPERVISOR'S RECOMENDATION

I hereby recommend that this project prepared under my supervision by RAVI ADHIKARI entitled **"OBJECT DETECTION AND TRACKING USING HOG BASED KALMAN FILTER"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

…………………………………………

Sanjeeb Prasad Panday (Ph.D.)

Assistant Professor

IOE, Tribhuvan University

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

LETTER OF APPROVAL

This is to certify that this project prepared by RAVI ADHIKARI entitled **"OBJECT DETECTION AND TRACKING USING HOG BASED KALMAN FILTER"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

…………………………………………. Sanjeeb Prasad Panday (Ph.D.) [Supervisor] Assistant Professor IOE, Tribhuvan University	…………………………………………. Hitesh Karki Chief Academic Officer DWIT College
…………………………………………. Dhiraj Kedar Pandey [External Examiner] IOST, Tribhuvan University	…………………………………………. Dr. Sunil Chaudhary [Internal Examiner] Lecturer, DWIT College

ACKNOWLEDGEMENT

I would like to thank Sanjeeb Prasad Panday (Ph.D.), Assistant Professor, IOE, TU and Mr. Sarbin Sayami, Lecturer, DWIT for supervising this project and giving assistance when faced with problems. Without their guidance and persistent help, this project would not have been possible.

I would like to thank Mr. Hitesh Karki, Chief Academic Officer of DWIT College for guiding me on various aspect of this project.

At last but not the least my sincere thanks go to my parents and members of my family, who have always supported me and to all of my friends who directly or indirectly helped me to complete this project report.

Ravi Adhikari

TU Exam Roll No: 3038/070

STUDENT'S DECLARATION

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

... ...

Ravi Adhikari

Date: August, 2017

ABSTRACT

Object Detection and Tracking detects the people available in a video frame and tracks them in subsequent frames. In this project, there are two major parts: Detection and Tracking. The detection of human shape in a video frame was detected after Histogram of Oriented Gradients (HOG) features were calculated and was classified with using Linear Support Vector Machine (SVM). The testing was done in MIT and INRIA pedestrian dataset where the MIT dataset had 509 training and 200 test images of pedestrians in city scenes. The INRIA dataset had 1805 photos with 64x128 resolution cropped from a large set of personal photos. For each image, the histogram of Oriented Gradients was calculated and the classification was done. The detection accuracy of the application was 85% based on the MIT and INRIA Person Detection Dataset. The detected human was tracked using Kalman Filter.

Keywords**:** Object Detection, Object Tracking, Histogram of Oriented Gradients, Kalman Filter

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CAMSHIFT:	Continuous Adaptive Mean Shift
DWIT:	Deerwalk Institute of Technology
HOG:	Histogram of Oriented Gradients
IOE:	Institute of Engineering
INRIA:	Institut National de Recherche en Informatique et en Automatique
KLT:	Kanade–Lucas–Tomasi
KG:	Kalman Gain
MIT:	Massachusetts Institute of Technology
R-CNN:	Region -based Convolutional Neural Network
RPN:	Region Proposed Network
SVM:	Support Vector Machine

CHAPTER 1: INTRODUCTION

Detecting objects in videos has always been popular project in the field of computer vision. One of the important reasons is its wide variety of applications, such as video surveillance, robotics, and intelligent transportation systems. However, detecting humans in video streams is a difficult task because of the various appearances caused by different clothing, pose and illumination. Many human detection methods have been developed but most of these methods are focus on finding powerful features or classifiers to obtain high detection rate. For applications such as real-time human detection for robotics and automotive safety, both efficiency and accuracy are important issues that should be considered carefully. In this paper, the problem is tackled by computing histograms of oriented gradient (HOG) for detection of object in videos and Kalman filtering is used to track detected objects. The aim of the proposed system is to achieve accurate human detection, while maintains efficient for applications that require fast human detection and tracking.

1.1 Background

The rapid growth of Artificial Intelligence and Robotics has greatly developed the field of computer vision. The increasing use of video sensors, with Pan-Tilt and Zoom capabilities or mounted on moving platforms in surveillance applications, have increased researcher's attention on processing arbitrary video streams. The processing of a video stream for characterizing events of interest relies on the detection, in each frame, of the objects involved, and the temporal integration of this frame based information to model simple and complex behaviors. This high-level description of a video stream relies on accurate detection and tracking of the moving objects, and on the relationship of their trajectories to the scene. [1]

1.2 Problem Statement

Object Detection and tracking highly relies on the features extracted from a video frame. In each frame, a complex set of features need to be processed, extracted, computed and classified. In cases such as video streams generated from a moving aircraft - there is high noise and clutter. For real time systems, such as tracking vehicles in case of automatic car, the results need to be quickly computed and fed into the main system. Having high set of features makes the system more accurate and robust. However, it costs processing time and high computational resources. The proposed system uses a newer method of detecting objects i.e. Histogram of Oriented Gradients, where the video frames quickly generate the gradients which can then be classified for detection using a previously trained model.

1.3 Objectives

1.3.1 General objective

To extract Histogram of Oriented Gradients (HOG) in video frames and use a Linear Support Vector Machine (SVM) to classify the extracted HOG features in real time and initialize a tracker for the object using Kalman Filter.

1.3.2 Specific objective

1) To extract HOG features of individual video frames
2) To train and classify HOG for detecting Human Object using Linear SVM.
3) To track the detected Human Object and predict its position in the next frame using Kalman Filter.

1.4 Scope

Object Detection and Tracking can be used by researchers and individual users to detect and track objects in video streams. The application is explorable and highly applicable in the field of Robotics, Self-Driving Vehicles, Security Cameras, etc.

1.5 Limitation

a) The training of the classifier takes a large image dataset input and hence it can be slower to train.
b) The detection of object is susceptible to difficult lighting conditions.

1.6 Outline of Document

This section describes how the report is organized.

a) Preliminary Section: This contains Title Page, Supervisor's Recommendation, Letter of Approval, Acknowledgement, Student's Declaration, Abstract, Table of Contents, List of Figures, List of Tables and List of Abbreviations
b) Introduction Section: This section deals with the background of the project, the Problem Statement, Objectives, Scope and Limitation. Requirement and Feasibility Analysis Section: This section contains the Literature review of the project, Requirements of the project and Feasibility Analysis performed
c) System Design Section: This section deals with the Methodology used for the solution of the problem and Algorithm used to solve the problem.
d) Implementation and Testing Section: This section deals with the Implementation details, Testing and Results obtained.
e) Maintenance and Support Plan Section: This section discusses the Maintenance plan that will be performed and the support that application will be provided.
f) Conclusion and Recommendation Section: This section deals with the conclusion of the project and recommendation for solving the limitations that occurred during the development of the project.

The first chapter dealt with the background, problem statement, objectives and scope of the project. The next section will discuss the literature review and requirement analysis.

CHAPTER 2: REQUIREMENT AND FEASIBILITY ANALYSIS

2.1 Literature Review

2.1.1 Background Subtraction for Object Detection

Background subtraction is a commonly used technique for motion segmentation in static scenes [2]. It attempts to detect moving regions by subtracting the current image pixel-by-pixel from a reference background image. The pixels where the difference is above a threshold are classified as foreground. The creation of the background image is known as background modeling (e.g. by averaging images over time in an initialization period). After creating a foreground pixel map, some morphological post processing operations such as erosion, dilation and closing are performed to reduce the effects of noise and enhance the detected regions. The reference background is updated with new images over time to adapt to dynamic scene changes

2.1.2 Temporal Differencing for Object Detection

In temporal differencing, moving regions are detected by taking pixel-by-pixel difference of consecutive frames (two or three) in a video sequence. Temporal differencing is the most common method for moving object detection in scenarios where the camera is moving. Unlike static camera segmentation, where the background is comparably stable, the background is changing along time for moving camera; therefore, it is not appropriate to build a background model in advance. Instead, the moving object is detected by taking the difference of consecutive image frames t-1 and t. [3]

2.1.3 Mean-Shift Approach for Object Tracking

Mean-shift is an approach [4] to feature space analysis. This is an iterative approach which shifts a data point to the average of data points in its neighborhood similar to clustering. It has found its application in visual tracking [5] and probability density estimation. Mean Shift tracking uses fixed color distribution. In some applications, color distribution can change, e.g., due to rotation in depth. Continuous Adaptive Mean Shift (CAMSHIFT) [6]. CAMSHIFT can handle dynamically changing color distribution by adapting the search window size and computing color distribution in a search window.

2.1.4 KLT for Object Tracking

The Kanade–Lucas–Tomasi (KLT) feature tracker is basically a feature extraction approach. It is based on the early work of Lucas and Kanade on an iterative image registration technique [7] that makes use of spatial intensity gradients to guide the search towards the best match. The method was developed fully by Tomasi and Kanade [8].

2.2 Requirement Analysis

Functional requirement	Non-functional requirement
Detect Object in the Video Frame	Calculate the HOG features from the current Frame and Classify the frame detection windows
Track the Object in subsequent frames	Initialize a tracker for the current position of the Object and record it
Predict the next position of the Object	Show a symbol in the frame where the object is predicted to occur

Table 2.1 Functional and Non-Functional Requirements

Table 1 describes the functionality of Object Detection and Tracking which includes the selection of a video file from the user directory, reading individual frames from the video, locating the previously trained object, and predicting the next position of the object. The application provides a detected window, and the expected position of the object as output.

2.3 Feasibility Analysis

2.3.1 Technical feasibility

Object Detection and Tracking is a desktop application. It uses Tkinter to show interactive controls to the user and uses python on its backend. The application is designed to be standalone and does not require internet connectivity.

The application as developed over python is cross platform and can run on any operating system. All of the tools and technology required by the application are easily available on the Internet for free of cost. Hence, the application was determined to be technically feasible.

2.3.2 Operational feasibility

The application has simple controls and no extra knowledge is required to run the application. It has easy to use design and the project dependencies are easy to meet. Hence, the Object Detection and Tracking application was determined operationally feasible.

2.3.3 Schedule Feasibility

The figure 2.1 shows the schedule feasibility of the project. The project was done over the period of 15 weeks. The research was done from the first week till fifth week. The familiarity of languages, Implementation of Algorithms was provided 8 and 12 weeks respectively. The documentation was done slowly over the entire course of the project.

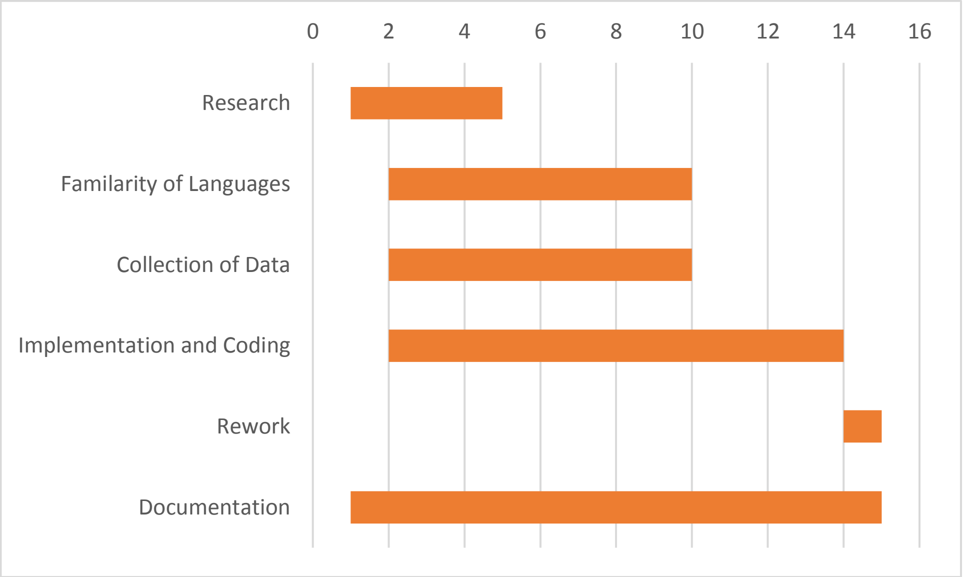


Figure 2.1 Gantt Chart for the Project

This chapter dealt with the previous work done and different approaches done in the field of detection and tracking. This section also presented requirements for the project and Feasibility Analysis was discussed. The next chapter will discuss about the System design i.e. Methodology used and Algorithms implemented.

CHAPTER 3: SYSTEM DESIGN

3.1 Methodology

Object Detection has been already implemented using HOG based features. Motion tracking for objects has been implemented using Kalman Filter which uses foreground detection algorithm. The Object Detection and Tracking application uses the combination of these two implementations. Object is Detected in the video frame which is then used as input for the tracking purpose.

3.1.1 Data collection

Data used in the application for training of the HOG based features is used from MIT and INRIA Pedestrian Dataset. The detection of positive sliding window requires the comparison of current HOG features of the window with the previously trained model.

The image from the MIT Pedestrian Dataset was in ppm format whereas the INRIA Pedestrian Dataset was in lossy png format.

Positive sample image on Figure 3.1 shows a man standing in an open space.

Figure 3.1 Positive Sample Image from the INRIA Dataset

The negative sample image on Figure 3.2 shows an empty street with a vehicle parked at the right side of the road.

Figure 3.2 Negative Sample Image from the INRIA Dataset

3.1.2 Data selection

In order to train the system, the positive images require to be small and with minimum background available. In order to train the negative images, large images with maximum background available are required. The images were filtered from the two datasets and were selected for training the model.

3.1.3 Data preprocessing

The image datasets from the MIT and INRIA datasets are normalized on its gamma and color. The positive samples are cropped to 64x128 for reducing larger background. The individual frames are resized to 640x480 resolution.

The gamma normalization of the input image was proposed by Dalal and Triggs [9]. Adding this computational overhead improved slight performance by 84% to 85%. So, this computational overhead is removed in the implementation.

3.2 Algorithm

Two major algorithms were used were:

3.2.1 Histogram of Oriented Gradients

Figure 3.3 shows the block diagram of HOG computation and classification algorithm.

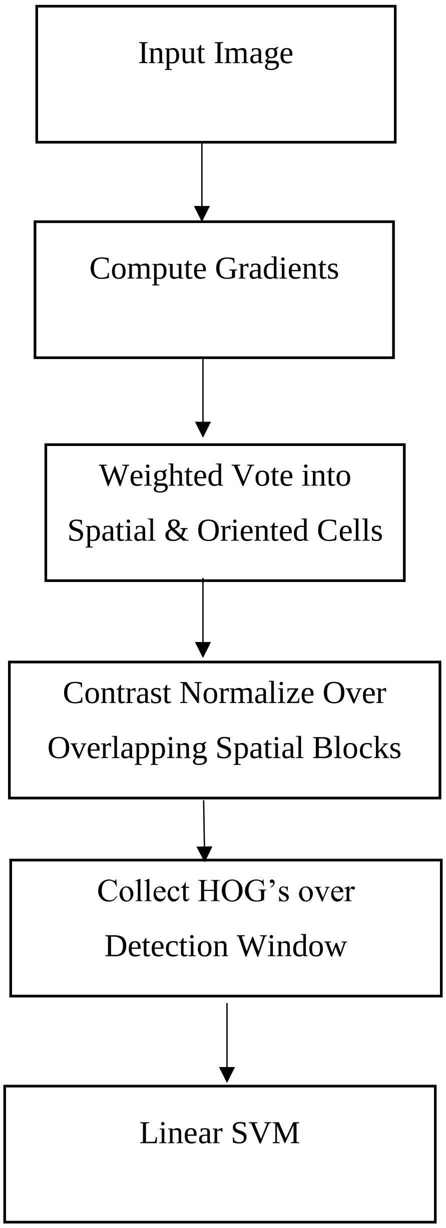


Figure 3.3 HOG Feature Extraction Algorithm

Gradient Computation equation is shown below:

$$g = \sqrt{{g_x}^2 + {g_y}^2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3.1)}$$

$$\theta = arctan\frac{g_x}{g_y} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3.2)}$$

The visualization of HOG features is shown below in figure 3.4

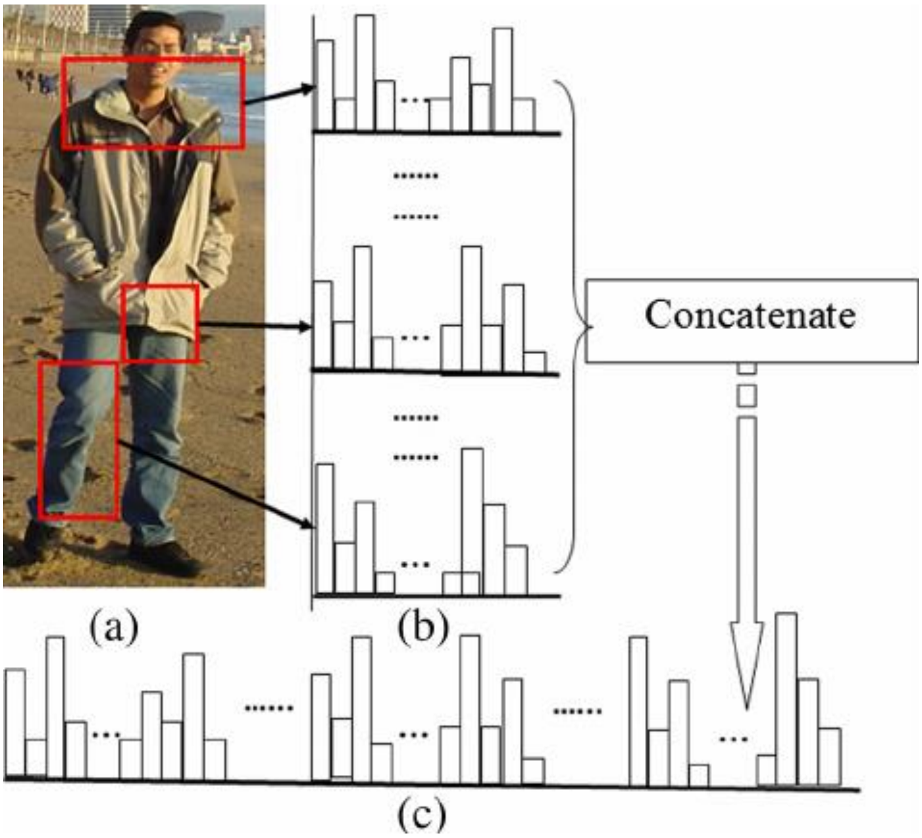


Figure 3.4 Visualization of HOG features in Images.

(a) Original Image (b) Block Histogram (c) Merging Histograms to create multi-dimensional Feature vector

The figure 3.5 below shows the computed gradients using the Sobel method.

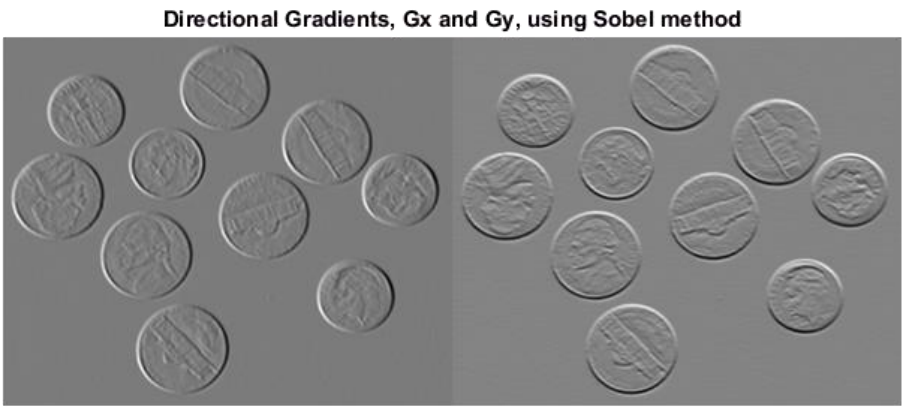


Figure 3.5 Calculation of Directional Gradients using Sobel Method

The figure 3.6 below illustrates the weighted vote that is carried out in the calculation of Histogram of Gradients (HOG) features.

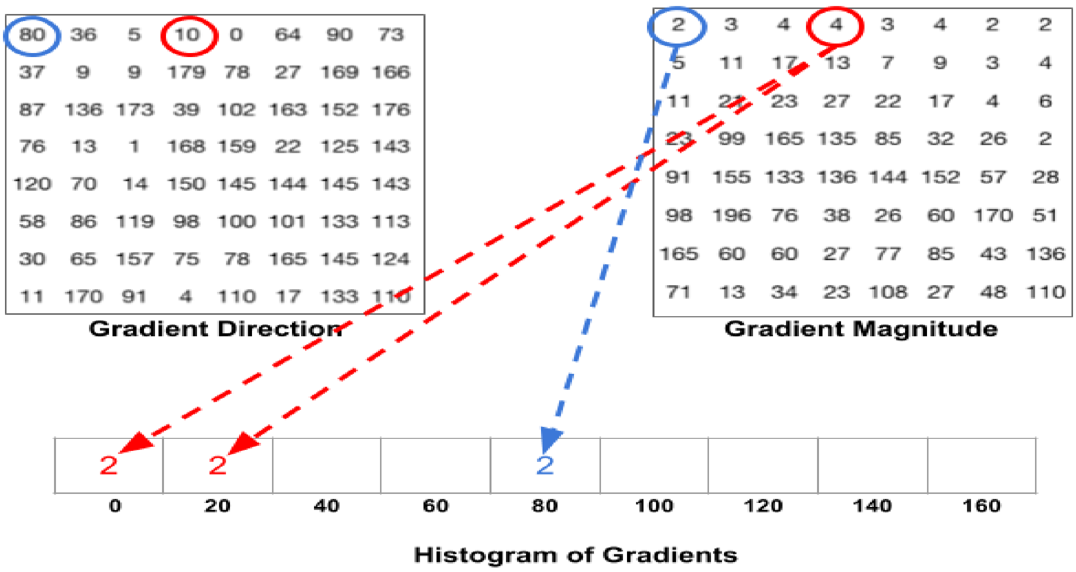


Figure 3.6 Weighted Vote in HOG

The figure 3.7 and 3.8 below show the computed HOG feature after the gradient voting for a positive and a negative image respectively.

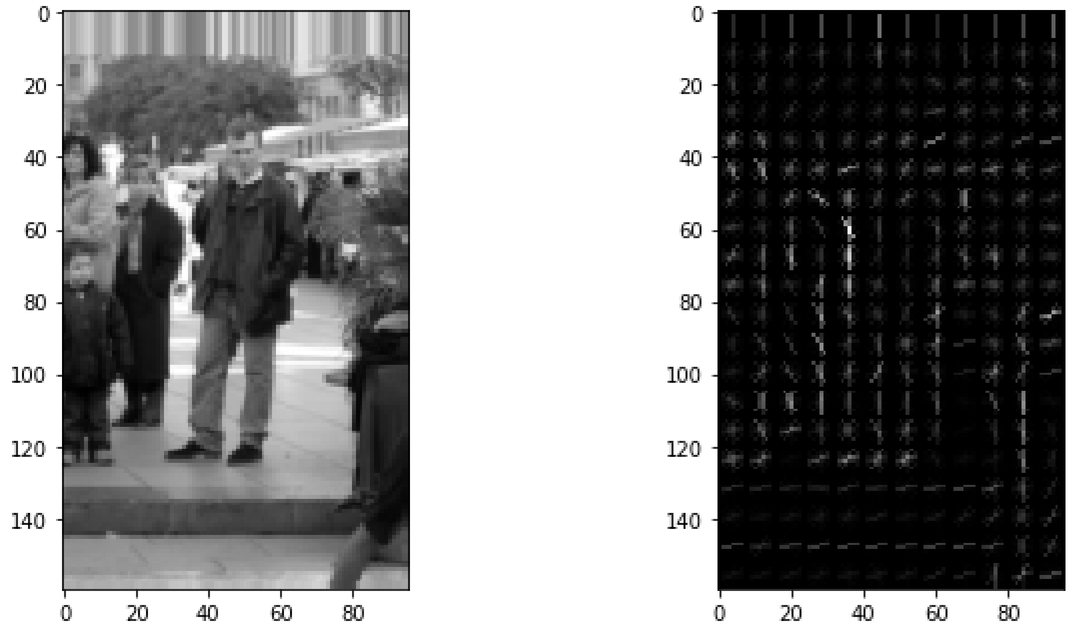


Figure 3.7 HOG Based Feature of Positive Image

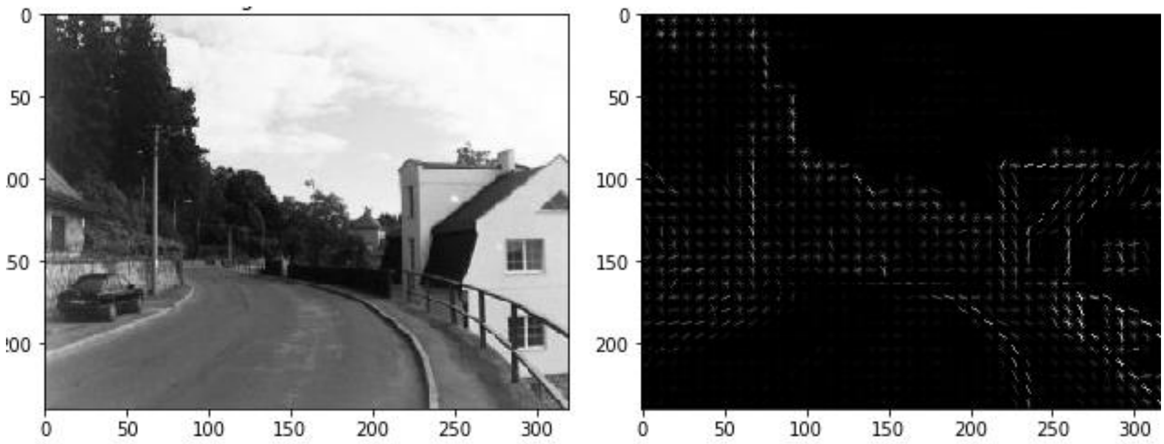


Figure 3.8 HOG Based Feature of Negative Image

3.2.2 Support Vector Machine

A Support Vector Machine (SVM) is a classification and a regression tool that can find the best separable plane between two or more data groups. The main objective of the Support Vector Machine is to find the best splitting boundary between data. In their basic form, SVM constructs the hyperplane in input space that correctly separates the example data into two classes. This hyperplane can be used to make the prediction of class for unseen data. The hyperplane exists for the linearly separable data. Consider two points $\mathbf{X_P}$ and $\mathbf{X_N}$ belonging to each convex hull. Make them as close as possible without allowing them to leave their respective convex hulls. The median hyperplane of these two points is the maximum margin separating hyperplane.

The points $\mathbf{X_P}$ and $\mathbf{X_N}$ can be parametrized as

$$X_P = \Sigma i\epsilon p\ \alpha_i\ x_i \qquad \Sigma i\epsilon p\ \alpha_i\ x_i = 1 \qquad \alpha_i \geq 0 \ldots\ldots\ldots\ldots (3.3)$$

$$X_N = \Sigma i\epsilon N\ \alpha_j\ x_j \qquad \Sigma i\epsilon N\ \alpha_j\ x_j > 1 \qquad \alpha_j \geq 0$$

where sets P and N respectively contain the indices of the positive and negative examples. The optimal hyperplane is then obtained by solving

$$min_\alpha\ \parallel XP - XN \parallel^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.4)$$

under the constraints of the parametrization (3.3). The separating hyperplane is then represented by the following linear discriminant function:

$$\hat{y}(x) = (X_P - X_N)\ x + (X_N X_N - X_P X_P)/2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.5)$$

Since XP and XN are represented as linear combination of the training patterns, both the optimization criterion (3.4) and the discriminant function (3.5) can be expressed using dot products between patterns.

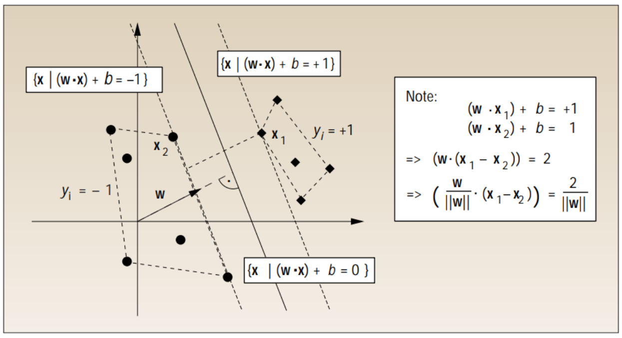


Figure 3.9 Support Vector Machine

3.2.3 Kalman Filter

The figure 3.9 shows the workflow of the Kalman Filter. The equations below show the computations to be performed. The Kalman Filter predicts the new position of the target object shown below by the Update Estimate block.

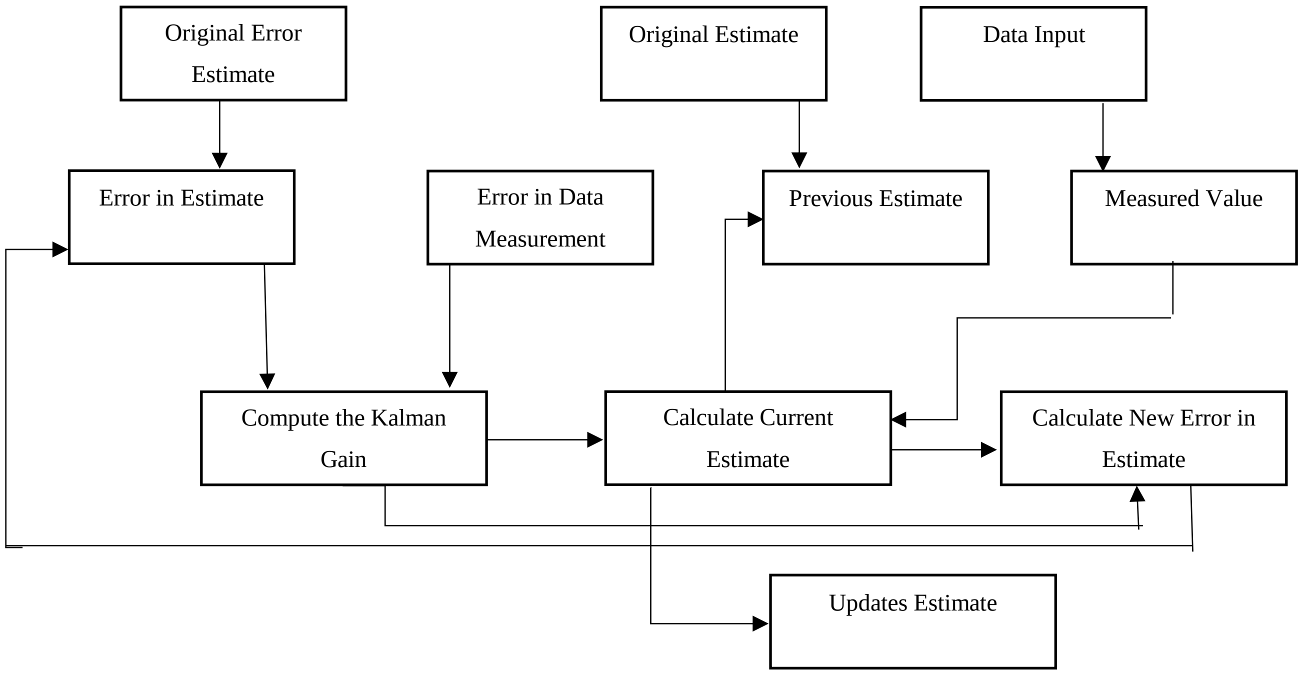


Figure 3.10 Kalman Filter Workflow

The equation below is the state representation of the Kalman Tracked object

$\boldsymbol{x(k+1) = F(k)x(k) + G(k)u(k) + v(k)}$ ………… …………..……………… (3.6)

Where, $\boldsymbol{x(k)}$ is the n_x dimensional state vector

$\boldsymbol{u(k)}$ is the n_u dimensional known input vector

$\boldsymbol{v(k)}$ is the unknown zero mean process noise with covariance

$\boldsymbol{E[v(k)v(k)'] = Q(k)}$ ……………………………………………………………… (3.7)

The measurement equation is $\boldsymbol{z(k) = (k)(k) + (k)}$ where k=1, … ……………….…. (3.8)

$\boldsymbol{w(k)}$ is the unknown zero mean white measurement noise with covariance

$\boldsymbol{E[w(k)w(k)'] = R(k)}$ ……………………………………………………….…... (3.9)

3.3 System Design

3.3.1 Class Diagram

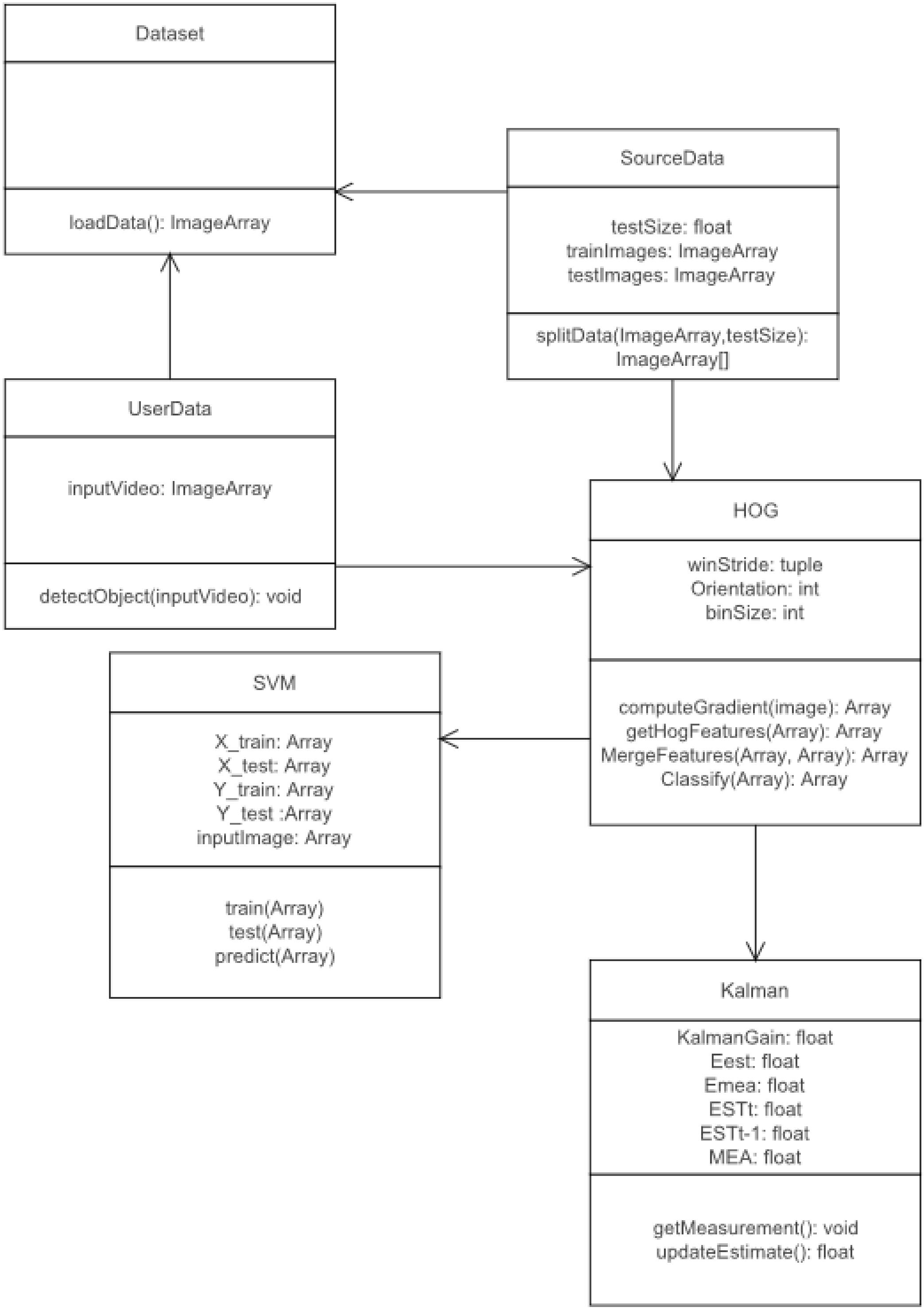


Figure 3.11 Class Diagram

Figure 3.10 shows the class diagram for the project. The Dataset class is inherited by UserData and SourceData. The splitData method in SourceData splits the dataset into training and testing samples based on the test size. The loadData method loads the data into the memory and creates an array for the computation. The ComputeGradients in the HOG class calculates the gradients for the image dataset. Then, features are calculated using the getHOGFeatures method. The features are merged using MergeFeatures method. The final array is fed to the SVM classifier. The SVM classifier classifies the training and test data using the train method. The classifier can detect new Objects using the predict method. The detected objects are passed into the Kalman Filter where the new possible position of the object is predicted.

3.3.2 State Diagram

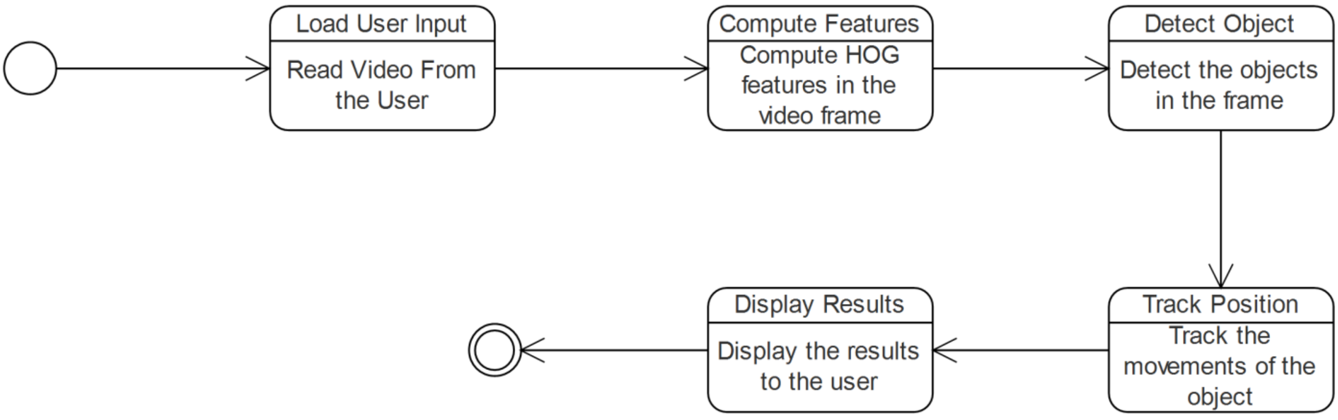


Figure 3.12 State Diagram

The figure 3.11 shows the state diagram of the project. The major flow of control and data into the application is shown in figure 3.11.

This chapter dealt with the methodology and algorithms used in the project. The next section will deal about the implementation methods and results that were obtained.

CHAPTER 4: IMPLEMENATION AND TESTING

4.1 Implementation

Users can run the application by simply running the desktop interface. The available options on the interface is loading an image for detection and loading a video for object detection, and motion tracking.

When a user clicks on the load image button the interface, the user then needs to select the path of the image file. The program then reads the selected image and computes its Histogram of Gradients (HOG) features. This feature is useful for the classification step to let the application know whether the selected image contains any objects that is recognizable by the application.

When the user clicks on load video button, the user needs to select a mp4 video. The individual frames on the video is processed and the HOG features are calculated. The sliding window checks on these HOG features and compares these cells with the previously trained model. When the object is detected, a boundary is created showing the detected objects. In case of overlapping, if the overlapping threshold is above 0.65, the two boundary boxes are merged and a single boundary box is shown. Based on the detected region, a Kalman Filter is initialized. The Kalman filter tracks the object and predicts its motion in the video. The tracking continues on the object which is added for the case of occlusion.

4.1.1 Tools used

CASE tools:

a. Edraw Max

Desktop Interface:

a. Tkinter

Background:

a. Python
b. OpenCV

c. Sklearn

This section describes the tools and technologies used in the Object Detection and Tracking application. The application has a desktop interface which is created using Tkinter. On the background, the code runs on Python programming language and uses the computer vision libraries.

All the algorithms required for the application are written in Python programming language. There are three major algorithms used in the application. Histogram of Oriented Gradients(HOG) is used to extract features from the video frame, Linear SVM is used to classify the features in the video frame and Kalman Filter is used to track the position of the detected object in motion. The OpenCV library was used for all the major algorithms in the project.

4.2 Results

4.2.1 Input Image

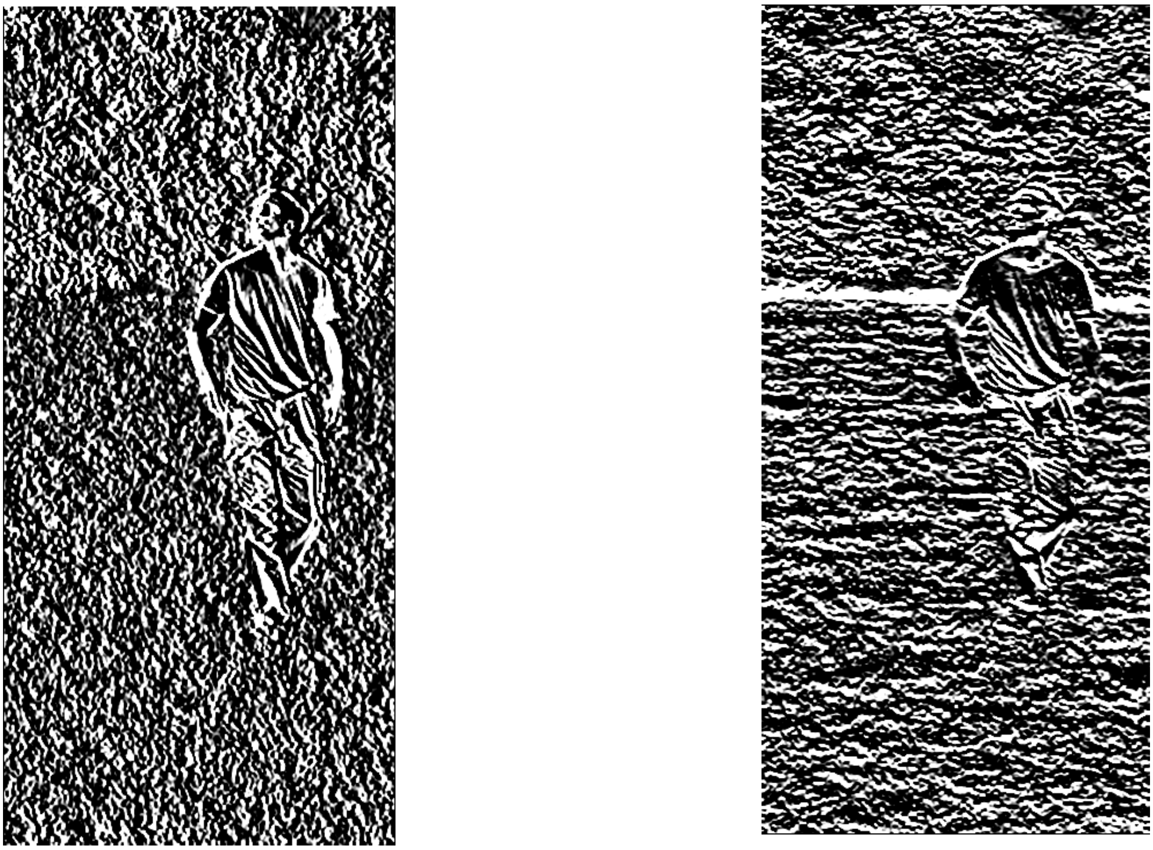
The figure 4.1 (a) and (b) shows the sample image taken from the INRIA dataset where the individual computation steps will be visualized later in this section.

(a) (b)

Figure 4.1 Input Images (a) Positive Image (b) Negative Image

4.2.2 Gradient Computation

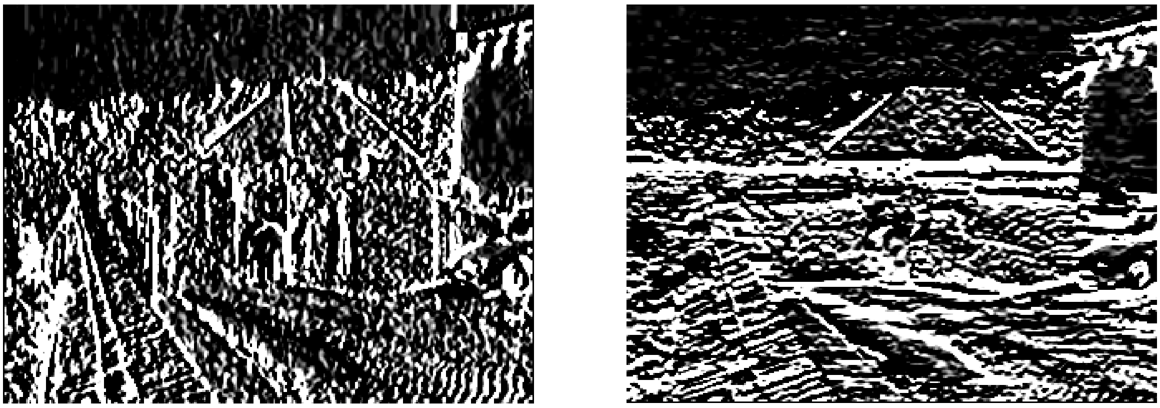
An image gradient is a directional change in the intensity or color in an image. The gradient computed for the positive and negative sample is shown below.



(a) *(b)*

Figure 4.2 Computed Gradients of Positive Image

(a) Computed Gradient over Y (b) Computed Gradient over X



(a) *(b)*

Figure 4.3 Computed Gradients of Negative Image. (a) Over X (b) Over Y

Figure 4.2 shows the Computed gradients for positive image using X in (a) and using Y in (b). The figure 4.3 shows the computed gradients for negative image using X in (a) and using Y in (b).

4.2.3 Histogram Computation over 3 channels

The histogram features are calculated after weighted vote into spatial and oriented cells. The figure 4.4 shows the histogram features for positive image that has been stacked into a single feature vector.

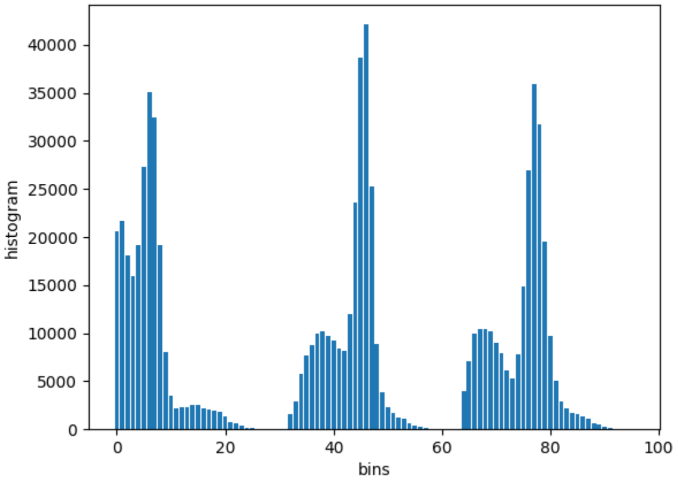


Figure 4.4 Histogram Features for Positive Image using bin size 32

The figure 4.5 shows the histogram features for negative image that has been stacked into a single feature vector.

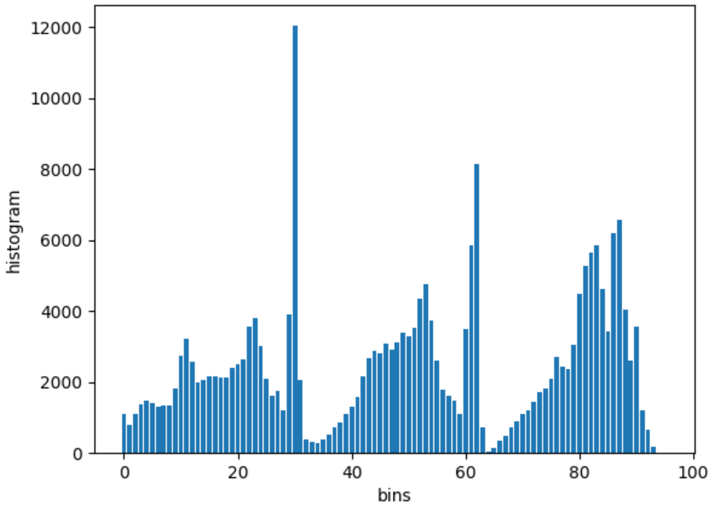
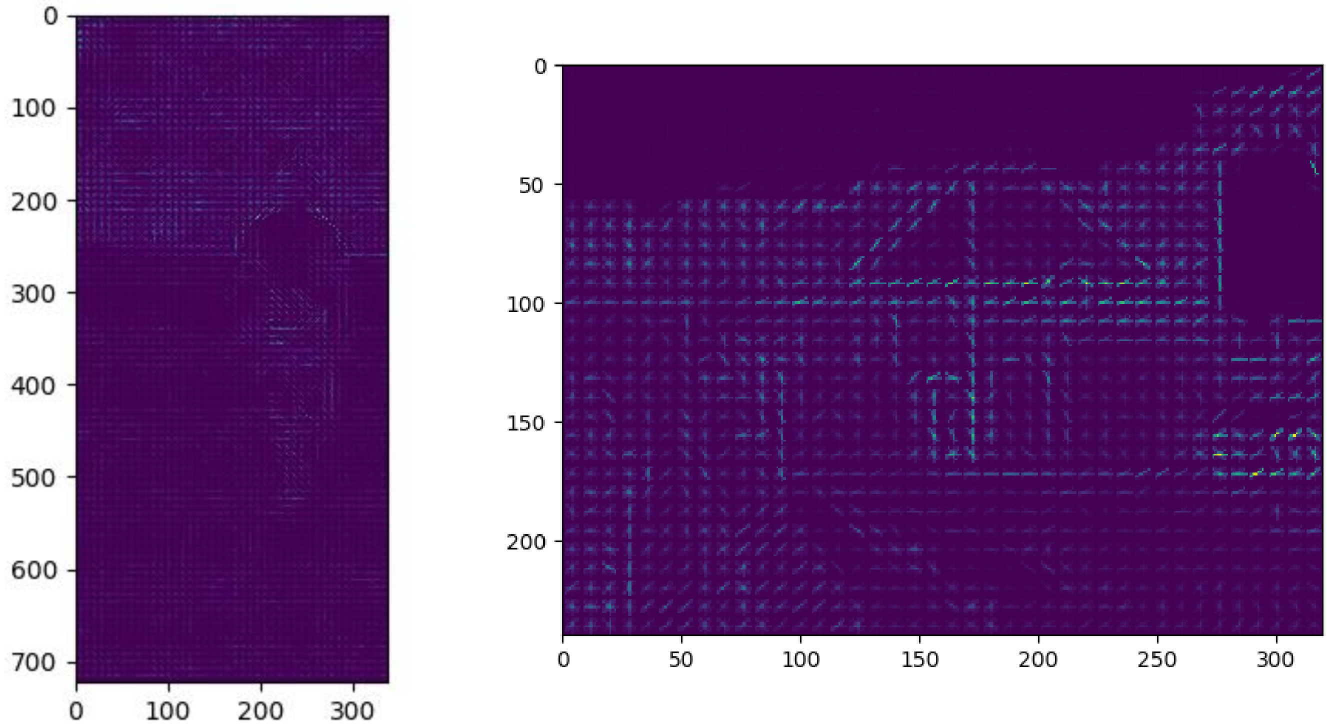


Figure 4.5 Histogram Features for Negative Image using bin size 32

4.2.4 HOG Calculation

The visualization of the final HOG image is shown below in figure 4.6.



(a) *(b)*

Figure 4.6 HOG Features for (a) Positive Image (b) Negative Image

4.2.5 Classification

(a) *(b)*

Figure 4.7 Classification of the input images as true output(a) and false output (b)

4.3 Testing

The system was tested using the MIT and INRIA dataset. MIT dataset had 200 test images of people in the city scenes. INRIA dataset had 1805 photos which were cropped from personal photos. These photos were provided in 64x128 resolution.

4.3.1 Testing for Sample Images

Figure 4.8 shows the true input image given to the classifier and figure 4.9 shows the true detection

Figure 4.8 True Input Image to the classifier

Figure 4.9 True Detection of the postive Input Image

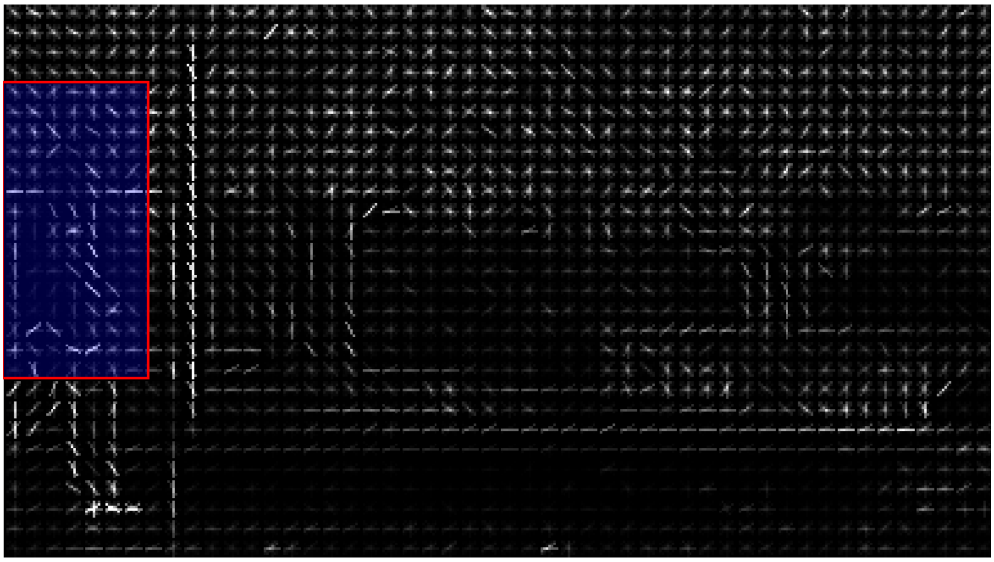


Figure 4.10 HOG Features shown for Positive Input Image

Figure 4.11 Negative Input Image to the classifier

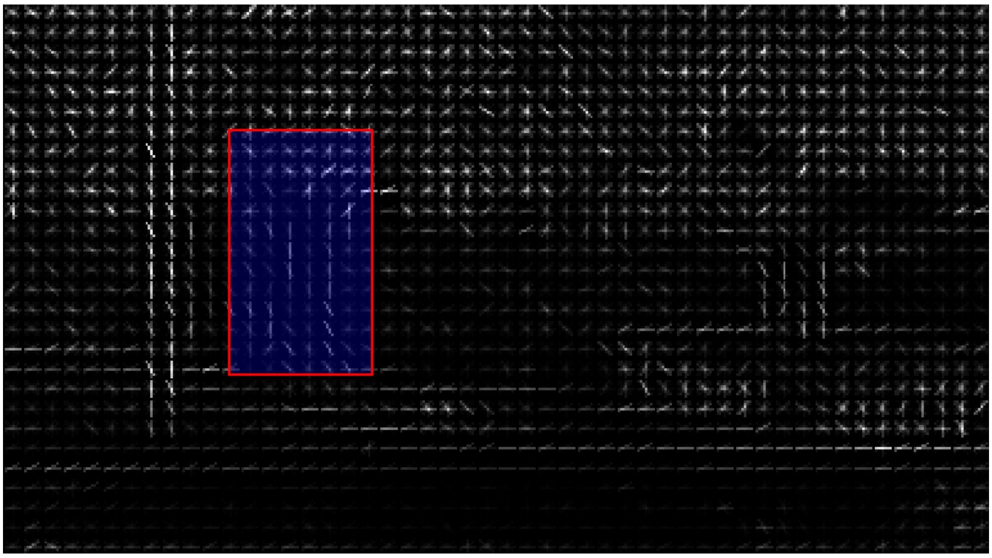


Figure 4.12 Detection of Positive Features in Negative Image

Figure 4.11 shows the negative input image provided to the classifier. Figure 4.12 shows the detection window which classifies the input image as positive i.e. the highlighted window contains a human object. Figure 4.13 shows that the false input image contains a human object.

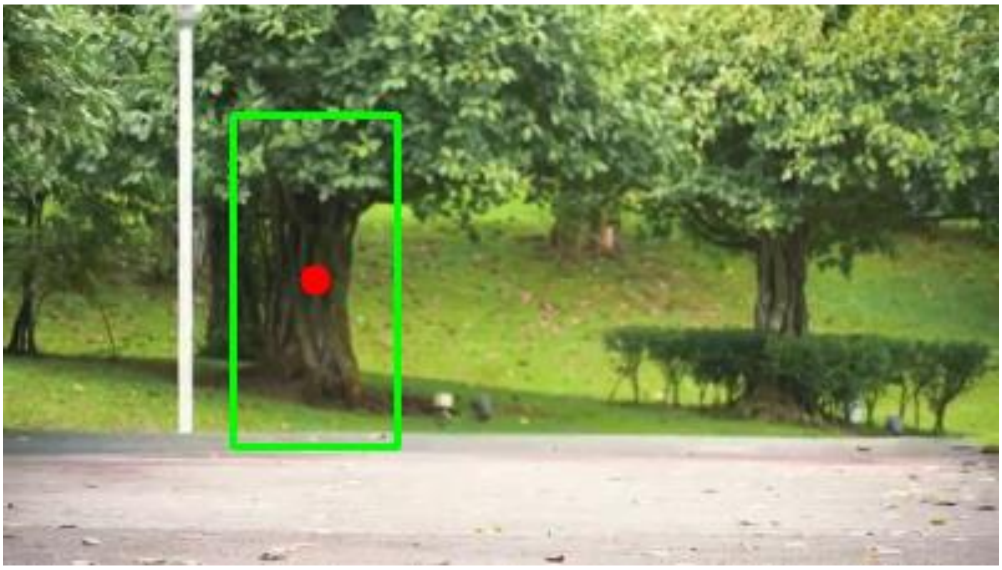


Figure 4.13 Detection of False Input Image as Positive

This section dealt about the Testing and Implementation part. The next section will discuss about the Maintenance and Support Plan for the project.

CHAPTER 5: MAINTENANCE AND SUPPORT PLAN

5.1 Maintenance Plan

Object Detection and Tracking application will implement corrective maintenance for resolving different bugs that may occur when the project is live. Perfective maintenance will be implemented for increasing the efficiency of the application by optimizing various implementation methods. Preventive maintenance will be implemented to make sure the application is free from bugs and copes up with up to date technologies.

5.2 Support Plan

Object Detection and Tracking will be presented to the Computer Vision community so that the community can further explore and enhance the application. The application can be used by various organization for security assurance, artificial intelligence and Robotics.

This section dealt about the Maintenance and Support Plan for the project. The next section will deal about the Conclusions and Recommendations for the project.

CHAPTER 6: CONCLUSION AND RECOMMENDATION

6.1 Conclusion

Object Detection and Tracking application was successfully implemented using HOG based Kalman Filter. The detection accuracy of the application was 85% based on the MIT and INRIA Person Detection Dataset. The testing was done in MIT and INRIA pedestrian dataset where the MIT dataset had 509 training and 200 test images of pedestrians in city scenes. The INRIA dataset had 1805 photos with 64x128 resolution cropped from a large set of personal photos. Over a sequence of video frames, the detection of an object is not consistent i.e. it is lost between frames and again detected on latter frames. This detection is tracked by a Kalman filter and an estimate of possible location is provided during the missed detection in frames.

6.2 Recommendation

The application was trained over still frames and used in detecting video streams. Training the application over video streams can highly improve the detection rate of the application. Newly proposed algorithm for detection uses Region Proposal Network (RPN) combined with Fast Region -Based Convolutional Neural Networks (R-CNNs) which can provide better accuracy and faster results [10].

This section dealt about the Conclusion and Recommendations for the project. The next section contains References and Appendix.

References

[1] I. Cohen and G. Medioni, "Detecting and Tracking Moving Objects for Video Surveillance," *IEEE Proc. Computer Vision and Pattern Recognition*, 1999.

[2] A. M. Mclvor, "Background Subtraction Techniques," *Proc. of Image and Vision Computing*, 2000.

[3] S. Shaikh, K. Saeed and N. Chaki, Moving Object Detection Using Background Subtraction, Springer, 2014.

[4] Y. Cheng, "Mean shift, mode seeking, and clustering," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1998, pp. 790-799.

[5] D. Comaniciu, V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *IEEE Proc. on Computer Vision and Pattern Recognition*, 2000, pp. 673-678.

[6] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, 1998.

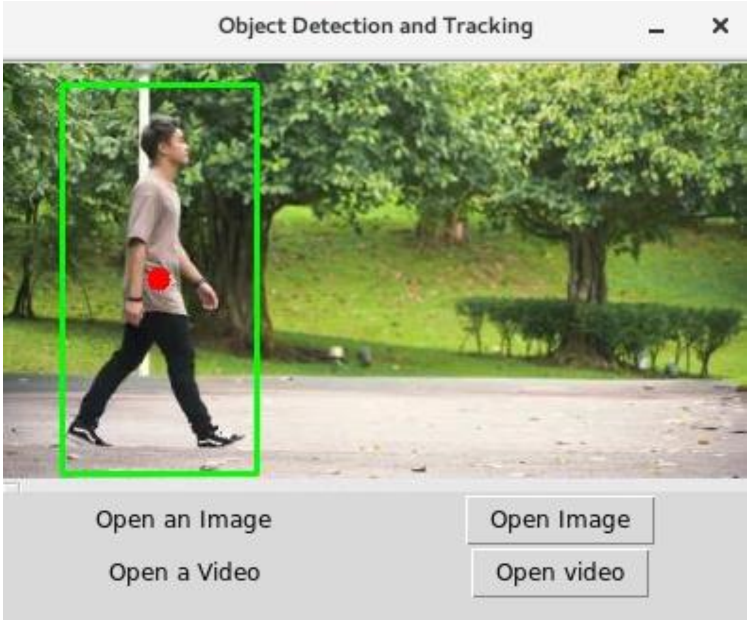
[7] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Applicaiton to Stereo Vision," in *International Joint Conference on Artificial Intelligence*, 1981, pp. 674-679.

[8] T. Carlo and K. Takeo, "Detection and Tracking of Point Features. Carnegie Mellon University Technical Report," *CMU-CS-91-132,* 1991.

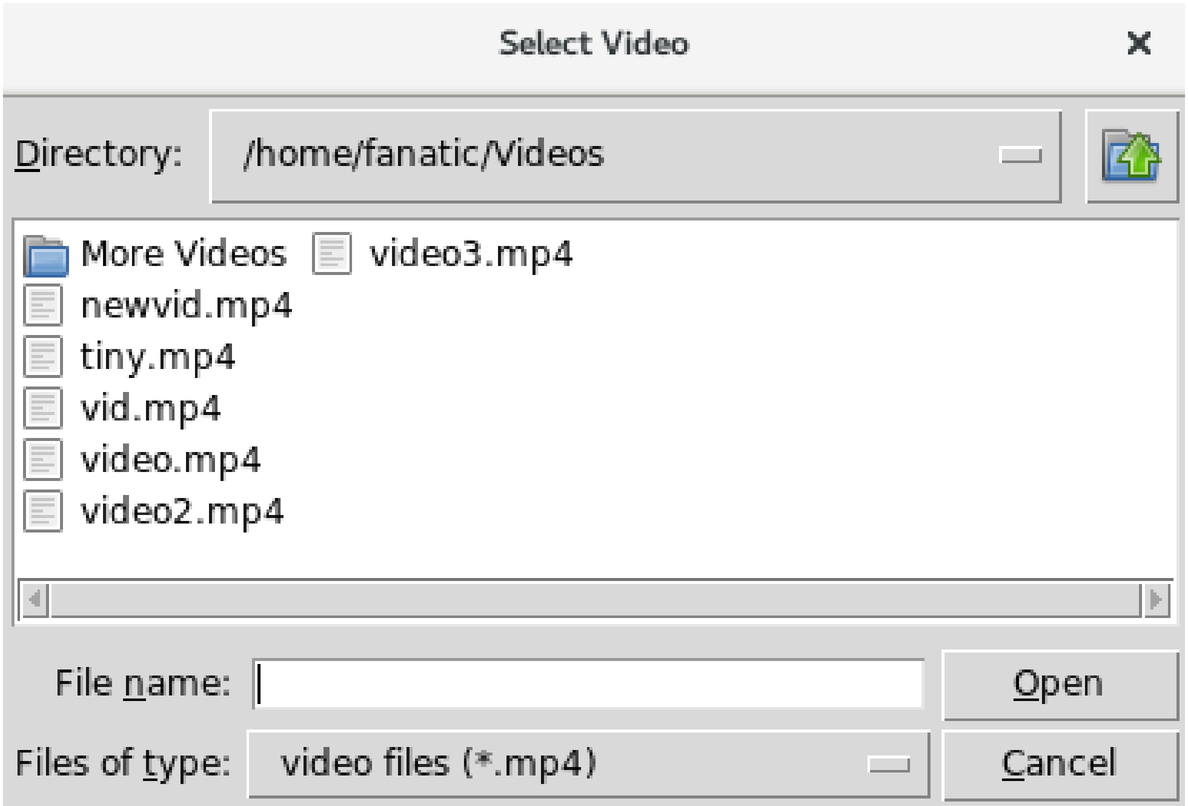
[9] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for HumanDetection," *Computer Vision and Pattern Recognition*, 2005.

[10] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

Appendix

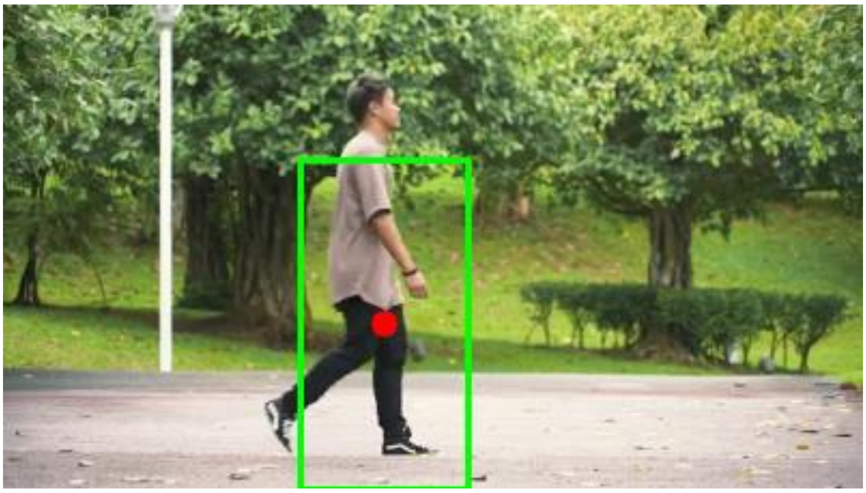


Appendix I Initial Interface for the User

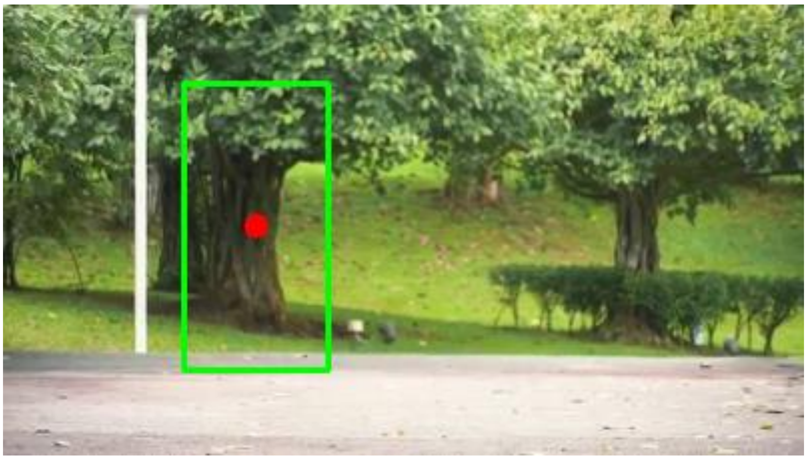


Appendix II User is asked to select the video file for detection and tracking

Appendix III No Detection



Appendix IV True Detection



Appendix V False Detection