**Vehicle and Pedestrian Detection using Haar Cascade**

**A Project Report**

Submitted to the Faculty of Engineering of

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA, KAKINADA**

In partial fulfillment of the requirements for the award of the Degree of

# BACHELOR OF TECHNOLOGY

In

# COMPUTER SCIENCE AND ENGINEERING

By

**Ch. Keerthi Sai D. Prathyusha**

**(18481A0545) (18481A0557)**

**Ch. Lakshmi Pallavi B. Harish Chandra**

**(18481A0549) (18481A0525)**

Under the guidance of

**Dr. T. Srinivasa Rao, M.Tech , Ph.D.**

Associate Professor of CSE Department

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING SESHADRIRAO GUDLAVALLERU ENGINEERING COLLEGE

**(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)**

## SESHADRIRAO KNOWLEDGE VILLAGE GUDLAVALLERU – 521356

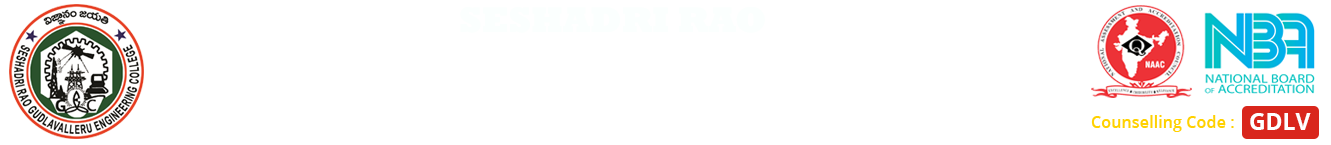
**ANDHRA PRADESH 2021-2022**

**SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE**

**(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)**

**SESHADRI RAO KNOWLEDGE VILLAGE, GUDLAVALLERU**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

**This is to certify that the project report entitled “Vehicle and Pedestrian Detection using Haar Cascade” is a bonafide record of work carried out by Ch. Keerthi Sai (18481A0545), D. Prathyusha (18481A0557), Ch. Lakshmi Pallavi (18481A0549), B. Harish Chandra (18481A0525) under the guidance and super vision in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of Jawaharlal Nehru Technological University Kakinada, Kakinada during the academic year 2021-22.**

**Project Guide Head of the Department**

**(Dr. T. Srinivasa Rao) (Dr. M. BABU RAO)**

**External Examiner**

**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragements crown all the efforts with success.

We would like to express our deep sense of gratitude and sincere thanks to **Dr. T. Srinivasa Rao,** Associate Professor, Department of Computer Science and Engineering, Department of Computer Science and Engineering for his constant guidance, supervision and motivation in completing the project work.

We feel elated to express our floral gratitude and sincere thanks to **Dr. M. Babu Rao**, Head of the Department, Computer Science and Engineering for his encouragements all the way during analysis of the project. His annotations, insinuations and criticisms are the key behind the successful completion of the project work.

We would like to take this opportunity to thank our beloved principal **Dr. G.V.S.N.R.V Prasad** for providing a great support for us in completing our project and giving us the opportunity for doing project.

Our Special thanks to the faculty of our department and programmers of our computer lab. Finally, we thank our family members, non-teaching staff and our friends, who had directly or indirectly helped and supported us in completing our project in time.

**Team members**

**Ch. Keerthi Sai (18481A0545)**

**D. Prathyusha (18481A0557)**

**Ch. Lakshmi Pallavi (18481A0549)**

**B. Harish Chandra (18481A0525)**

# INDEX

**TITLE PAGENO**

# LIST OF ABBREVIATIONS i

LIST OF FIGURES ii

[ABSTRACT iv](#_TOC_250016)

[CHAPTER 1: INTRODUCTION 1-2](#_TOC_250015)

* 1. [INTRODUCTION 1](#_TOC_250014)
  2. [OBJECTIVES OF THE PROJECT 2](#_TOC_250013)
  3. [PROBLEM STATEMENT 2](#_TOC_250012)

[CHAPTER 2: LITERATURE SURVEY 3 -4](#_TOC_250011)

[CHAPTER 3: PROPOSED SYSTEM 5-16](#_TOC_250010)

* 1. [METHODOLOGY 5](#_TOC_250009)
  2. ALGORITHM 6
  3. [IMPLEMENTATION](#_TOC_250008)  10

CHAPTER 4: RESULTS & ANALYSIS 17-21

[4.1 RUNNING THE APPLICATION](#_TOC_250003) 17

CHAPTER 5: CONCLUSION & FUTURE SCOPE 21

* 1. [CONCLUSION 21](#_TOC_250002)
  2. [FUTURE SCOPE 21](#_TOC_250001)

[REFERENCES 22](#_TOC_250000)

LIST OF POs AND PEOs 23

Mapping of Project outcomes with graduated POs and PEOs 25

|  |  |
| --- | --- |
| **Abbreviation** | **Explanation** |
| PCA | Principal Component Analysis |
| LBP | Linear Binary Patterns |
| HoG | Histogram of Object Gradients |

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Figure Name** | **Page No.** |
| 3.1.1 | Methodology | 5 |
| 3.2.1 | Haar Features | 7 |
| 3.2.2 | Integral Images Calculation | 7 |
| 3.2.3 | Combining Weak Classifiers | 8 |
| 3.2.4 | Cascading Classifiers | 9 |
| 3.3.1 | Frames of a Sample Video | 11 |
| 3.3.2 | Image after subtracting Background | 11 |
| 3.3.3 | Flow Chart to create XML File | 12 |
| 3.3.4 | Positive Images of Pedestrians | 12 |
| 3.3.5 | Positive Images of Vehicles | 12 |
| 3.3.6 | Negative images of Vehicles and Pedestrians | 13 |
| 3.3.7 | Loading The images into GUI | 13 |
| 3.3.8 | Setting the number of stages | 14 |

|  |  |  |
| --- | --- | --- |
| 3.3.9 | Setting the size for same amount of images | 14 |
| 3.3.10 | XML File Visualization | 15 |
| 3.3.11 | Implementing Pedestrian Detection | 15 |
| 3.3.12 | Implementing Bike Detection | 16 |
| 3.3.13 | Implementing car Detection | 16 |
| 3.3.14 | Implementing Bus Detection | 16 |
| 4.1 | Running the File | 16 |
| 4.2 | Detecting Pedestrians | 17 |
| 4.3 | Detecting Bus | 19 |
| 4.4 | Detecting Cars | 19 |
| 4.5 | Detecting Two-Wheelers | 20 |

# ABSTRACT

Computer Vision Plays a vital role in traffic management and surveillance systems and has been an active research area in the past years. In systems like these, the detection of vehicles and also classification of the vehicle plays a major role. The datasets are traffic videos of urban environment taken from various cities around the world which were used to train the classifier hence generating a robust classifier. The proposed approach is computationally less expensive with faster processing speed. The experiments on-road prove it to be a robust and real time algorithm which is highly competitive with the existing architecture.

This System has the following Haar Cascades:

* Haar Cascade for Car detection
* Haar Cascade for Bus Detection
* Haar Cascade for Two-Wheeler Detection
* Haar Cascade for Pedestrian Detection

# CHAPTER -1 INTRODUCTION

## INTRODUCTION

Vision Recognition by Computer is an important part of traffic management and surveillance systems, and it's been a hot topic for research in recent years. Vehicle detection and classification play an important role in systems like this.

The datasets consist of traffic footage from several cities across the world that were used to train the classifier, resulting in a robust classifier. The proposed method is less expensive in terms of computation and has a faster processing speed.

Experiments on the road show that it is a reliable and real-time algorithm that is competitive with existing architecture.

*“A baby learns to crawl, walk and then run. We are in the crawling stage when it comes to applying machine learning.” ~****Dave Waters***

Machine learning is a tool for turning information into knowledge. In the past 50 years, there has been an explosion of data. This mass of data is useless unless we analyse it and find the patterns hidden within. Machine learning techniques are used to automatically find the valuable underlying patterns within complex data that we would otherwise struggle to discover.

The Haar cascade is a machine learning object detection approach that uses features given by Paul Viola and Michael Jones in their work to identify things in an image or video.

Obtaining data (pictures) from video, applying pre-processed images, categorising images into different categories, and training these images using the cascade algorithm are all part of the training-classifier stage.

In addition, the applying-classifier procedures include gathering images from video, recognising safety objects, generating a safety score, and providing feedback based on that score. With the help of computers, industrial organisations will be able to identify and control the safety of a working environment automatically.

## OBJECTIVES OF THE PROJECT

Vehicle and Pedestrian Detection can be helpful in various aspects. Usually, they are mostly used in

* + - Traffic Management.
    - Surveillance Systems.

## PROBLEM STATEMENT

* Amount of training time and Operational Cost are high.
* Identifying the Pedestrians and Vehicles in the Images/Videos is little difficult.

# CHAPTER- 2 LITERATURE SURVEY

Many templates and objects matching methods with exceptionally high accuracy existed before Haar cascade's development and use, such as the scale-invariant feature transform, speed up robust feature, and orientated fast and rotated binary robust independent elementary features.

These algorithms are highly efficient, but their extensive processing durations prevent them from being used for real-time detection. The Haar-cascade algorithm, on the other hand, is an ML-based approach in which a cascade function is trained from a large number of positive and negative images. It is then used to recognise items in other photos.

Haar feature selection, integral image creation, AdaBoost training, and cascading classifiers are the four stages of the algorithm. The course classifier comprises of an assortment of stages, where each stage is a gathering of powerless students. The feeble students are straightforward classifiers called choice stumps. Each stage is prepared utilizing a procedure called supporting.

Helping gives the capacity to prepare an exceptionally precise classifier by taking a weighted normal of the choices made by the powerless students. Each phase of the classifier marks the locale characterized by the current area of the sliding window as one or the other positive or negative. A positive demonstrates that an item was found and a negative show that no articles were found. In the event that the mark is negative, the arrangement of this area is finished, and the identifier slides the window to the following area. Assuming that the mark is positive, the classifier passes the district to the following stage. The identifier reports an article found at the current window area when the last stage orders the locale as a positive.

Cascade classifier training requires a set of positive samples and a set of negative images. Haar-like features are attributes extracted from images used in pattern recognition. Their names are derived from their similarities to Haar wavelets. First, the pixel values inside the black area are added together; subsequently, the values in the white area are added. Next, the total value of the white area is subtracted from the total value of the black area. This result is used to categorize image sub-regions. The application of this algorithm varies from face detection to other object recognition applications. During the Haar-cascade algorithm process, the AdaBoost learning algorithm was also applied to boost the performance of the training process. AdaBoost required a large number of examples that had a strong effect on the generalization of the training error. It combined weak classifiers into strong ones using its specific Equations.

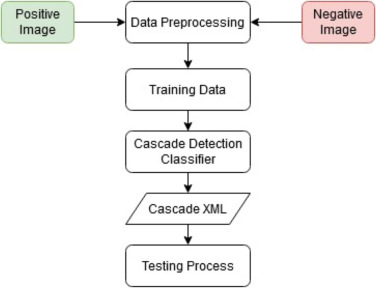
Before the introduction of the Haar-cascade algorithm in 2001, many object recognition applications have been created. Devi et al. used an additional principal component analysis (PCA) to reduce

the complexity of face images, decrease data size, and remove noise. Subsequently, Navaz et al. combined PCA with neural networks for face recognition and sex determination. These previous algorithms demonstrated some disadvantages such as a low percentage of classification and high mean square error. Meanwhile, with the advantages of quick detection and high efficiency, the Haar cascade was applied in many studies. This concept was applied in real-time video with a high accuracy rate and fast speed. However, this implementation depended on the video quality (light, angle, no obstacles). Additionally, Cuimei et al. improved the Haar cascade by combining three different classifiers (colour HSV, histogram matching, and eyes/mouth detection). In addition to this algorithm, a few others can be applied to real-time tracking topics such as linear binary patterns (LBPs) or a histogram of object gradients (HOG). Cruz et al. and Guennouni et al. compared these three algorithms together in their project of detecting objects using UAVs. The results indicated that the Haar-like cascade performed better than LBP in accuracy rate and better than HOG in speed.

Moreover, there are many researches on using deep learning and applying it in detecting cloths and non-hardhat-use for fashion and surveillance videos. However, these previous deep-learning algorithms were applied mostly for fashion with HOG (switch is slower than Haar cascade), not for safety management control. Therefore, in this work, we used the Haar cascade to train classifiers with fast speed and high accuracy. With these advantages of the Haar cascade algorithm, our system to train and detect safety objects in real time as well as calculate a safety score will be a valuable contribution to human working safety.

# CHAPTER- 3 PROPOSED SYSTEM

## 3.1 Methodology:

****

**Fig 3.1.1: Methodology**

In our project, we collected information i.e., the positive images (i.e., objects) around 7000 images and negative images (i.e., non-objects) around 4000 images to train the classifier, similar to other machine learning models.

By calculating all the parameters present in the given dataset, The output is displayed as the identification of objects and pedestrians.

Before Haar cascade’s invention and application, many templates and objects matching algorithms with extremely high accuracy existed, such as the scale-invariant feature transform, speed up robust feature, and oriented fast and rotated binary robust independent elementary features. These algorithms exhibit a high efficiency but cannot be applied to real-time detection owing to their long processing times.

Meanwhile, the Haar-cascade algorithm is an ML-based approach where a cascade function is trained from numerous positive and negative images. It is subsequently used to detect objects in other images or videos.

## Algorithm:

### Haar Cascade Classifier: -

Object Detection using Haar feature-based cascade classifiers is an effective method proposed by Paul Viola and Michael Jones in the 2001 paper, "Rapid Object Detection using a Boosted Cascade of Simple Features".

It is a machine learning based approach in which a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

The algorithm can be explained in four stages:

Calculating Haar Features

Creating Integral Images

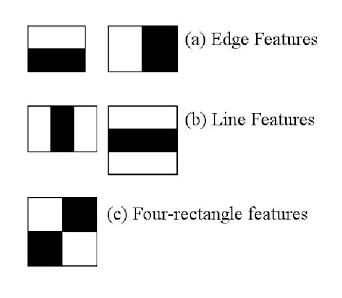
Using Adaboost

Implementing Cascading Classifiers

It’s important to remember that this algorithm requires a lot of **positive images** and **negative images** of non-faces to train the classifier, similar to other machine learning models.

## Calculating Haar Features

The first step is to collect the Haar features. A **Haar feature** is essentially calculations that are performed on adjacent rectangular regions at a specific location in a detection window. The calculation involves summing the pixel intensities in each region and calculating the differences between the sums. Here are some examples of Haar features below.

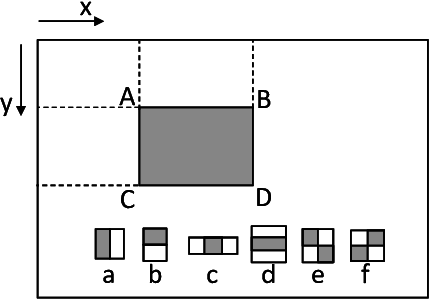


**Fig 3.2.1: Haar Features**

These features can be difficult to determine for a large image. This is where **integral images** come into action because the number of operations is reduced using the integral image.

## Creating Integral Images

Without going into too much of the mathematics behind it (check out the paper if you’re interested in that), integral images essentially speed up the calculation of these Haar features. Instead of computing at every pixel, it instead creates sub-rectangles and creates array references for each of those sub-rectangles. These are then used to compute the Haar features.



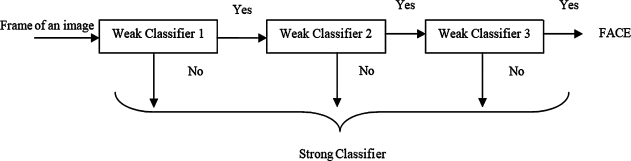
**Fig 3.2.2: Integral Images Calculation**

It’s important to note that nearly all of the Haar features will be **irrelevant** when doing object detection, because the only features that are important are those of the object. However, how do we determine the best features that represent an object from the hundreds of thousands of Haar features? This is where **Adaboost** comes into action.

## Adaboost Training

Adaboost essentially chooses the best features and trains the classifiers to use them. It uses a combination of **“weak classifiers”** to create a **“strong classifier”** that the algorithm can use to detect objects.

Weak learners are created by moving a window over the input image, and computing Haar features for each subsection of the image. This difference is compared to a learned threshold that separates non-objects from objects. Because these are “weak classifiers,” a large number of Haar features is needed for accuracy to form a strong classifier



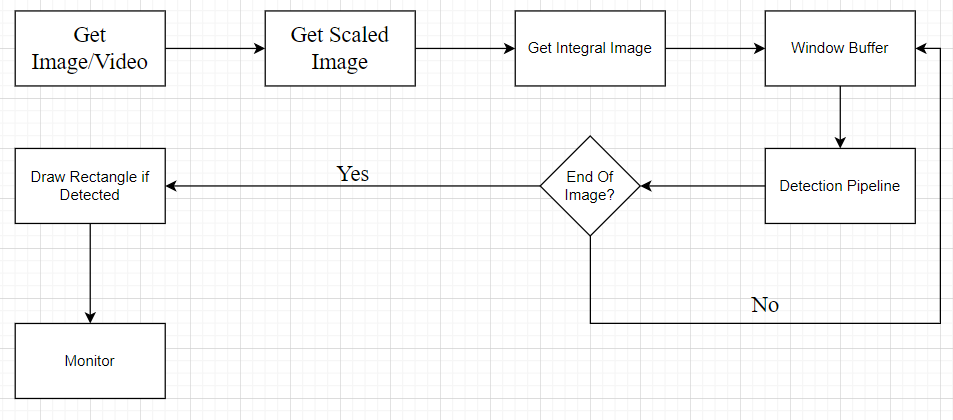
**Fig 3.2.3: Combining Weak Classifiers**

The last step combines these weak learners into a strong learner using **cascading classifiers.**

## Implementing Cascading Classifiers

The cascade classifier is made up of a series of stages, where each stage is a collection of weak learners. Weak learners are trained using boosting, which allows for a highly accurate classifier from the mean prediction of all weak learners.

Based on this prediction, the classifier either decides to indicate an object was found (positive) or move on to the next region (negative). Stages are designed to reject negative samples as fast as possible, because a majority of the windows do not contain anything of interest.



**Fig 3.2.4: Cascading Classifiers**

It’s important to maximize a **low false negative rate**, because classifying an object as a non-object will severely impair your object detection algorithm.

**Pseudocode:**

Let Ftarget is target overall false positive rate.

Let P is a set of positive examples.

Let N is a set of negative examples.

Let F0=1, D0=1 and i=0 where F0 is overall false positive rate at layer 0, D0 is acceptable detection rate at layer 0 and i is the current layer.

While Fi> Ftarget :

Increase Layer by 1

Use P and N to train with AdaBoost to make xml file.

Check the result of new classifier for Fi and D0

Decrease the threshold for new classifier to adjust detection rate r>=d\* Fi-1

N=empty

If Fi> Ftarget , use the current classifier and false detection to set N.

## 3.3 IMPLEMENTATION:

**Haar Cascade Using Python:**

The data collected consists of about overall of 7000 images including the vehicles, pedestrians and the background images.

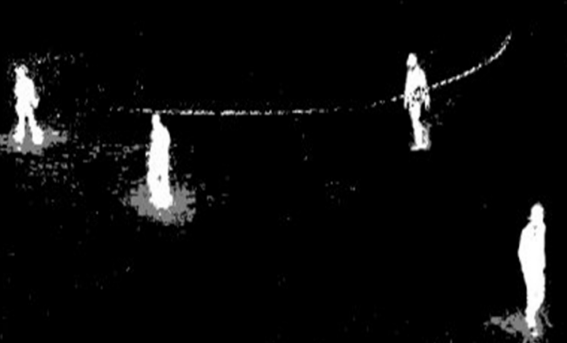
These images are trained and tested on the videos for required output. The detection phase of the proposed system is that, mainly a window of the target size is moved over the input image, and for each subsection of the image the Haar-like feature is calculated. This difference is then compared to a learned threshold that separates non-objects from objects. The biggest advantage of this method is the calculation speed.

Due to the use of integral images that quickly and efficiently generates the sum of values in a rectangular subset of a grid, henceforth a Haar-like feature of any size can be calculated in constant time (approximately 60 microprocessor instructions for a 2-rectangle feature). The image shows a single frame of a sample video. The image contains pedestrians as objects. The second one is a snapshot of the output after using MOG2 Background Subtraction. The Gray section under a pedestrian represents the shadow of the person.



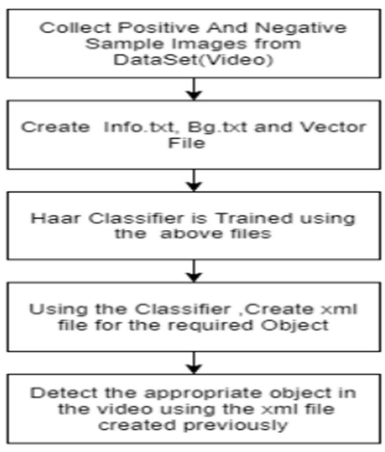
**Fig 3.3.1: Frame of a Sample Video.**

As a pre-processing step to improve the performance, Background Subtraction Using MOG2 was performed with the help of OpenCV library functions. The results for MOG2 subtraction for pedestrian detection is shown in Second Image:



**Fig 3.3.2: Image after subtracting Background**

Next, we create the XML files of the various objects (target object) to be detected i.e., one XML file each for bus, car, two-wheeler and pedestrians. The steps involved in creating the XML files is shown below.



**Fig 3.3.3: Flow Chart to create XML File**

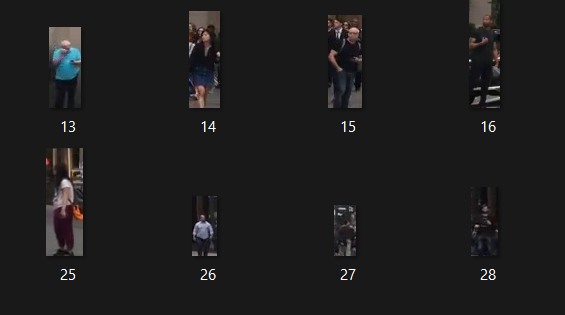
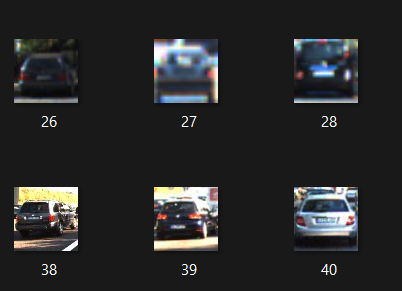
**Building includes the following steps:**

**Data Collection: Collection of Positive and Negative Image Set.**

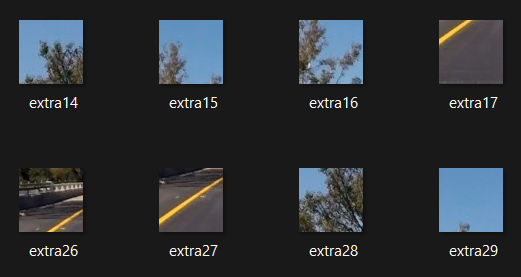
Collect the images of the pedestrians and the objects as a dataset from Kaggle.

Classify the images into positive and negative dataset.

The positive images consist of the objects and the pedestrian images (represented as p) and the Negative images consists of the background images (represented as n).

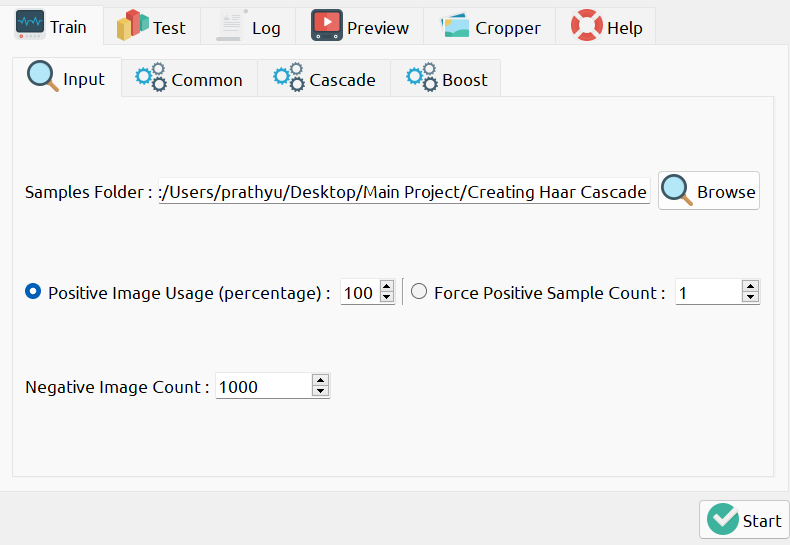
**Fig 3.3.4: Positive Images of Pedestrians Fig 3.3.5: Positive Images of Vehicles**



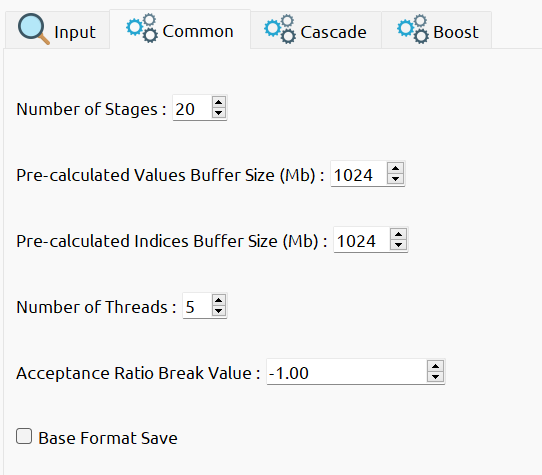
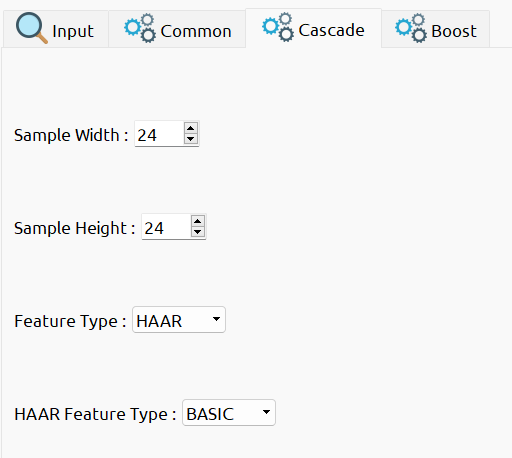
**Fig 3.3.6: Negative Images of Vehicles and Pedestrians**

**Data Pre-processing.**

Now train the Images using Cascade Classifier GUI by setting the number of stages including height and width.

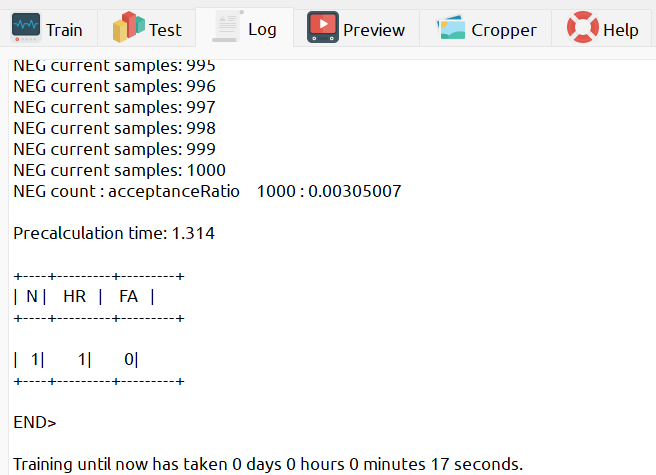


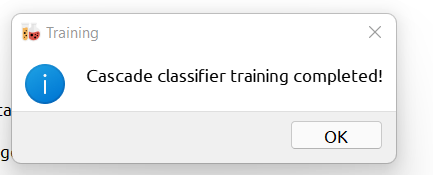
**Fig 3.3.7: Loading The images into GUI**

**Fig 3.3.8: Setting the number of Stages Fig 3.3.9: Setting the size for same amount of images.**

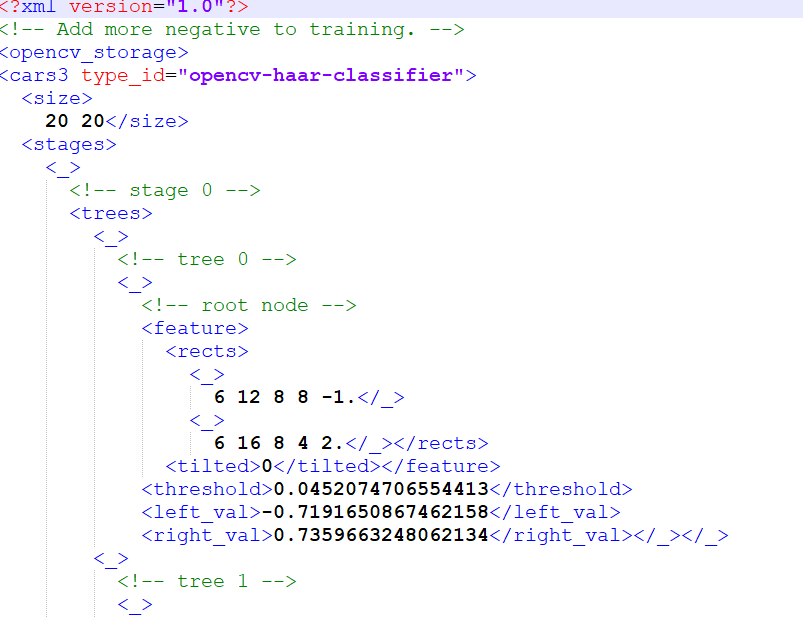
**Training the images:**

****

****

The Training images result as an XML file which is used as a classifier.

The classifier File which contains XML File

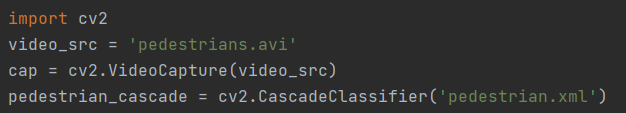


**Fig 3.3.10: XML File Visualization**

**Testing on Videos:**

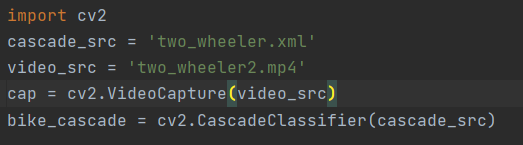
The different kinds of XML Files are imported accordingly including the testing videos.

**For Pedestrian Detection:**



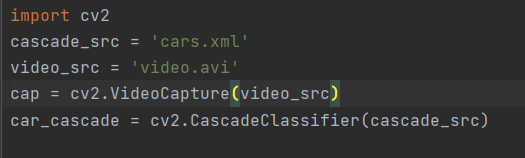
**Fig 3.3.11: Implementing Pedestrian Detection**

**For Bike Detection:**



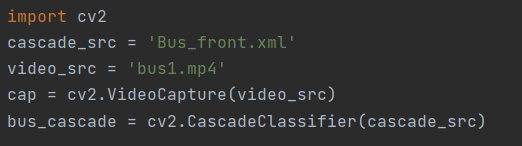
**Fig 3.3.12: Implementing Bike Detection**

**For Car Detection:**



**Fig 3.3.13: Implementing Car Detection**

**For Bus Detection:**

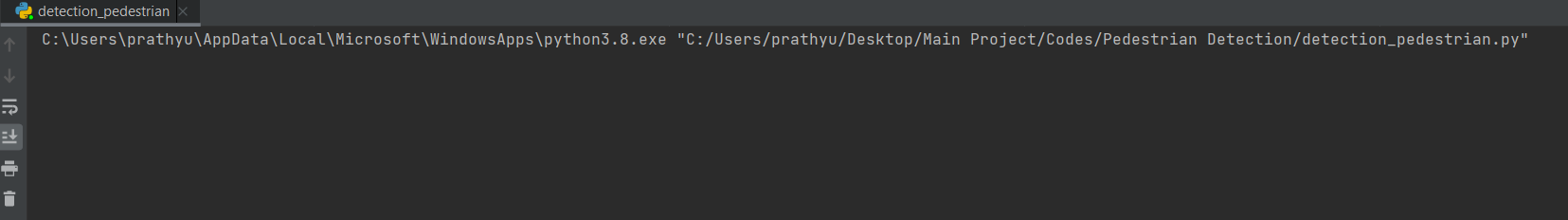


**Fig 3.3.14: Implementing Bus Detection**

## CHAPTER- 4 RESULT AND ANALYSIS

**Running The Application:**

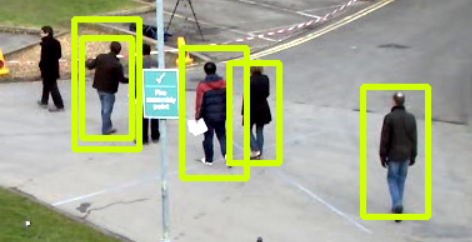
* Run the application by running the `app.py` file. By default, It displays the output.

****

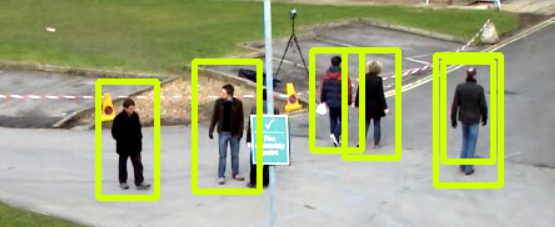
**Fig 4.1: Running the File**

**Outputs:**

**Detecting Pedestrians:**

****  ****

****

****

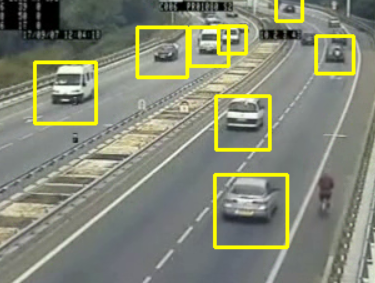
**Fig 4.2: Detecting Pedestrians**

**Detecting Bus:**

**** ****

**Fig 4.3: Detecting Bus**

**Detecting Cars:**

** **

**Fig 4.4: Detecting Cars**

**Detecting Two-Wheelers:**

****

****

**Fig 4.5: Detecting Two-Wheelers**

## CHAPTER- 5 CONCLUSION AND FUTURE SCOPE

## CONCLUSION

We present a simple yet successful strategy for speeding up the usual walker and vehicle identification procedure without losing accuracy in this study. The main addition is that we use low-resolution sketch for coarse detection and employ the quick Haar or Adaboost algorithms. Because the coarse detection process rejects the majority of detection windows, the computational labour of the fine detection phase is significantly decreased. We'll keep researching the HOG attribute in the future with the objective of making it uncomplicated to draw out. Furthermore, we want to leverage depth facts to detect pedestrian and vehicle. Using detailed particulars supplied by other sensors, we can acknowledge ROIs more quickly and squarely.

## FUTURE SCOPE

Design and simulate complicated video sequences, then evaluate them using the same tracking technique. In the hypothetical case, occlusion is employed for an item of the same hue for the moving objects, or else larger occlusion with a longer occlusion period is used. Increasing the number of objects aids in determining the effectiveness and functionality of the tracking method.

Weight parameters must be provided for each pixel's distinct intensity levels. In a picture, if an intensity value is designated as foreground based on the current frame, there is a lower possibility that the foreground also has comparable pixel coordinates, therefore the BG weightage for the pixel is set to the minimal value rather than the starting value. Adding weightage less than the beginning value has the benefit of deleting the old pixel value with the least likelihood rather than the evolving scene.

# REFERENCES

[1] D. M. Gavrila, J. Giebel, and S. Munder, “Vision-based pedestrian detection: the protector system,” in Proc. of the IEEE Intelligent Vehicles Symposium, Parma, Italy, 2004.

[2] Save-U. [Online]. Available: <http://www.save-u.org/>

[4] D. Geronimo, A. M. Lopez, and A. D. Sappa, “Survey of Pedestrian Detection for Advanced Driver Assistance Systems,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011.

[5] D. M. Gavrila and V. Philomin, “Real-time object detection for “smart” vehicles,” In Proc. of the ICCV, pp. 87–93, Kerkyra, Greece, 1999.

[6] C. Papageorgiou and T. Poggio, “A Trainable System for Object Detection,” Int’l J. Computer Vision, vol. 38, no. 1, pp. 15-33, 2000.

[7] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 886-893, 2005.

[8] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng, “Fast Human Detection Using a Cascade of Histrograms of Oriented Gradients,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 1491-1498, 2006.

[9] G. Xu, X. Wu, L. Liu, and Z. Wu, “Real-time pedestrian detection based on edge factor and Histogram of Oriented Gradient,” in Proc. IEEE Conf. Information and Automation, pp. 384-389, 2011

[10] R. Lienhart, and J. Maydt, “An extended set of Haar-like features for rapid object detection,” ICIP02, pp. 900–903, 2002.

**SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE**

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)

Seshadri Rao Knowledge Village, Gudlavalleru

**Department of Computer Science and Engineering**

# Program Outcomes (POs)

## Engineering Graduates will be able to:

* 1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
  2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
  3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
  4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions., component, or software to meet the desired needs.
  5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
  6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
  7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
  8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
  9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
  10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
  11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
  12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

# Program Specific Outcomes (PSOs)

PSO1: Design, develop, test and maintain reliable software systems and intelligent systems. PSO2: Design and develop web sites, web apps and mobile apps.

## PROJECT PROFORMA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classification of**  **Project** | **Application** | **Product** | **Research** | **Review** |
| √ |  |  |  |

**Note: Tick Appropriate category**

|  |  |
| --- | --- |
| **Project Outcomes** | |
| Course Outcome (CO1) | Identify and analyze the problem statement using prior technical knowledge in the domain of interest. |
| Course Outcome (CO2) | Design and develop engineering solutions to complex problems by employing systematic approach. |
| Course Outcome (CO3) | Examine ethical, environmental, legal and security issues during project implementation. |
| Course Outcome (CO4) | Prepare and present technical reports by utilizing different visualization tools and evaluation metrics. |

## Mapping Table

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CS1537: MAIN PROJECT** | | | | | | | | | | | | | | | |
| **Course Outcomes** | **Program Outcomes and Program Specific Outcome** | | | | | | | | | | | | | | |
| **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** |  | **PSO 1** | **PSO 2** |
| CO1 | 3 | 3 | 1 |  |  |  |  | 2 | 2 | 2 |  |  |  | 1 | 1 |
| CO2 | 3 | 3 | 3 | 3 | 3 |  |  | 2 | 2 | 2 |  | 1 |  | 3 | 3 |
| CO3 | 2 | 2 | 3 | 2 | 2 | 3 | 3 | 3 | 2 | 2 | 2 |  |  | 3 |  |
| CO4 | 2 |  | 1 |  | 3 |  |  |  | 3 | 3 | 2 | 2 |  | 2 | 2 |

**Note: Map each project outcomes with POs and PSOs with either 1 or 2 or 3 based on level of mapping as follows:**

1-Slightly (Low) mapped 2-Moderately (Medium) mapped 3-Substantially (High) mapped