Nighttime Vehicle Detection Using Computer Vision

Sushruth Badri, Sri Soumya Somu, K. Vamsi Meghana and V. Aparna

Abstract Many accidents occur during the night due to the improper visibility of the road ahead. One of the main reasons is of the discomfort posed by the high beam light of the oncoming vehicle, glares our eyes while driving. This discomfort might result in a lapse of concentration thereby resulting in an accident. Our primary aim is to automatically detect the head light using tracking and segmenting the frames extracted from the video signals that are fed by a camera and automatically switch the lighting condition of our vehicle from low beam to high beam or vice versa to avoid discomfort to the driver of the oncoming vehicle. We use MATLAB to simulate the results of our algorithm. In MATLAB, we mainly use computer vision and image processing to make necessary alterations to the input to get the necessary output.

Keywords Segmentation · Tracking · Blob analysis · Headlight detection
Computer vision

1 Introduction

90% of a driver's reaction depends on his visibility of the oncoming road condition and vehicles, and visibility of the same is severely limited during the night, making it dangerous to drive at night. Currently while driving the car headlight of the vehicle is switched on throughout the night at full power. The number of vehicles on road is increasing drastically making it even more difficult to drive at night. During the night, most of the accidents occur due to dazzling of the headlight. The headlight beam of the oncoming vehicle directly affects the driver's eyes and results in blurred vision or glare. It takes around 3–8 s for the driver to come back to the normal vision. This nighttime vehicle detection system can switch the headlights from high beam to low beam or vice versa without the intervention of the

S. Badri · S. S. Somu · K. Vamsi Meghana · V. Aparna (✉)
Manipal Institute of Technology, Manipal, India
e-mail: aparna.sreejith@manipal.edu

H. S. Saini et al. (eds.), *Innovations in Electronics and Communication Engineering*,
Lecture Notes in Networks and Systems 33,
https://doi.org/10.1007/978-981-10-8204-7_17

driver by processing some road and light conditions fed by the camera. Our main objective is to reduce the brightness on the eyes of the oncoming observer while making sure that the user will not be caused discomfort. It is important for the auto dipper to differentiate between headlight and taillight and then dim the light accordingly. Our primary goal is also to run the system of switching from low beam to high beam and vice versa without any manual effort. Low beams ensure less illumination whereas high beams provide clear visibility of the road ahead when vehicles are absent. Safety is always important while travelling at nighttime. A black and white camera is also sufficient to detect the cars and identify whether it is a headlight or a taillight depending on the brightness (Headlights provides high illumination and taillights provide low illumination). Segmentation, tracking and detection are the three main steps of our project.

The proposed system can be implemented in MATLAB. Then it can further be implemented in hardware.

2 Methodology

Our proposed method consists of the following four steps:

Pre-processing:

At first in MATLAB, the video sample is imported using VideoFileReader. The number of training frames is set and the input is taken from the camera and demarcated into corresponding frames. Then each frame is cropped to get only the area of interest. The frame is cropped and then evaluated to eliminate the street lights since these lights might also be detected as oncoming vehicular lights in later steps. Hence, we crop and remove the above part which contains street lights. Each input frame from the camera undergoes this process and all the further steps are done on these cropped frames.

Segmentation:

Segmentation is used to make the identification of the headlight in a much easier way. Image segmentation is the process of dividing the image into different parts which can be further processed separately.

1. Background Subtraction

Background subtraction method is used to partition the frames to foreground (white) and background (black). The frames with RGB values are converted into greyscale images which will greatly simplify the further processing of the input frames. Greyscaling discards the color values of an image and simplifies the process of differentiating the headlights from the background greatly compared to its RGB counterpart. In MATLAB, the Foreground Detector System object draws similarities between a video frame and a background model, and determines which areas of the frame lie in the foreground and which ones lie in the background. It then masks the foreground to show only the background of the present frame. After grey-scaling the images will have to go through the filtering process. This process filters out any noise in the image. The next step is to remove other unwanted objects and noises. We can do that using clean foreground detector. We then remove the noises from the foreground to obtain only the headlights, street lights and other significant and few unwanted lights.

2. Identifying Round Objects

Using thresholding, we convert the RGB frame into a greyscale image. Then prepare for boundary tracing. By using morphological functions, we remove all objects containing fewer than 20 pixels. This will eliminate any pertaining noise in the frame giving us a clearer frame to process further. For the next step we are required to compute the area enclosed by different light blobs to eliminate the unwanted blobs. Incomplete blobs result in faulty area computations. Hence, we fill any holes and use regionprops to estimate the area bounded by each of the boundaries that are traced. Option 'noholes' speed up the processing by stopping bwboundaries from searching for inner contours. We then compute the circularity of the blobs using the following formula.

$$\text{circularity} = (\text{Perimeter}.^{\wedge}2)./(4 * \text{pi} * \text{area})$$

After finding the circularity of each blob, we eliminate the ones which have a circularity of less than 0.94. This will help any irregular or non-circular reflections on the roads. By this process, we finally identify the round shape objects and remove the vehicle headlight reflections identified on the road. We now have only the required lights that we need to track to decide whether to dip the light or not.

Tracking:

Detected lights in the frame are tracked using Blob Analysis technique indicating a green box around the lights detected. In MATLAB, we use vision. Blob Analysis returns a blob analysis object, H, with each stated property set to the stated value. We use BoundingBoxOutputPort to find coordinates of bounding boxes. We set this property to true. This gives us coordinates of the bounding boxes. We then set the minimum and maximum blob area to 50 and 500 or may vary according to the surrounding conditions. The step of blob analysis computes and returns the statistics of input image. We then insert a rectangular green box to track the vehicle headlights. In this way, we take every frame and try to detect the vehicle headlights. Even if one light is detected, the light can be switched from high beam to low beam. When no oncoming lights are detected, the lights will be in high beam. In this way, we can successfully detect the lights and provide a better way of switching the lights automatically to avoid any discomfort experienced by the driver in switching the lights and the oncoming vehicle driver. This is a simple yet very effective way for the above purpose as can be seen in the results of simulations that have been shown in the result analysis section.

3 Result Analysis

Segmented Result:

1. *Foreground* Segmentation:

   The lights are shown in white and the background in black (Fig. 1).

2. Clean Foreground Result:

   The noise is eliminated from the segmented frame (Fig. 2).

3. Detecting Round Objects:

   The boundaries are added to round objects and circularity is calculated. The objects with circularity less than 0.94 are neglected (Fig. 3).

Tracking:

The detected lights are tracked as follows (Fig. 4).

Fig. 1 Foreground segmentation result

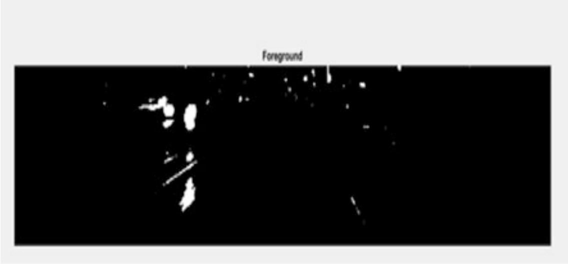


Fig. 2 Clean foreground result

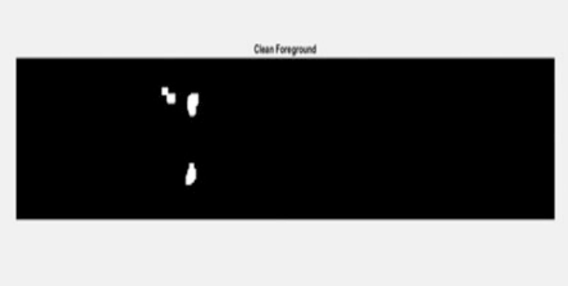
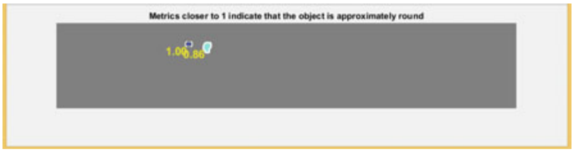


Fig. 3 Detecting round objects


Metrics closer to 1 indicate that the object is approximately round
1.00
0.86

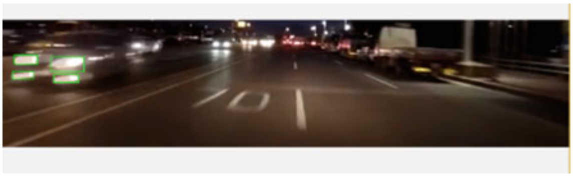


Fig. 4 Lights are tracked and green boxes are inserted around them

4 Future Scope

We presented a software for detecting and tracking the oncoming vehicles for driving safely during the night. The system performance is satisfactory for real-time conditions. One of the many advantages of this system is that it is a simple algorithm and works well in real-time conditions. The classification mistakes between vehicles, road signs and other features of the frame are very low and can be improved using distinctive invariant features and by increasing the size of the testing and training data. The results are encouraging and several improvements will be included to the current implementation in the future. Further improvements can be done in the classification phase to increase the accuracy of the classification of various features of the frame. To attain a more consistent tracking of the oncoming headlights, some parameters such as the blob area and shape of the blob can be experimented. Hardware implementation can also be sought after, owing to the projects simple hardware demands.

References

1. Jakub Sochor B (2014) Fully automated real-time vehicles detection and tracking with lanes analysis
2. Alcantarilla PF, Bergasa LM, Jiménez P, Parra I, Llorca DF (2011) Automatic lightbeam controller for driver assistance. Mach Vis Appl. https://doi.org/10.1007/s00138-011-0327-y
3. Jurić D On-road night-time vehicle light detection and tracking methods overview
4. Nemade B (2016) Automatic traffic surveillance using video tracking. Procedia Comput Sci 79:402–409