Real Time Detection of Speed Hump/ Bump At Night Time

K. Jebastin
Assistant Professor in ECE Department,
PSN College of Engineering and Technology
Tirunelveli,
jebastin@psncet.ac.in

Dhinakaran K
Bangalore, India.
dhina.matlab@gmail.com

Vishnu Thangam
Bangalore India
tvishnu1990@gmail.com

Abstract

Speed breakers or Hump has formed to reduce the accidents, but unfortunately excessive use of speed breakers in urban area / inner cities / metropolitan distracts vehicle drivers. In addition to that, most of time drivers cannot recognize the appearance of unmarked speed breakers and lose control of the vehicle, causing serious accidents and loss of lives. Especially in India speed breakers may not be visible to driver's eye due to fade borders, having no cross lines to indicate, damaged road, less differentiation from road and so on due to poor maintenance, in this article we proposed a new approach to sort out identified problems using image processing techniques. As we seen the high severity is in night time most cases, we planned to developed algorithms to work on night time as driver may avoid / mitigate the critical situation in daytime.

1. Introduction

The speed bump detection system in a vehicle is used to detect speed bumps, humps, speed breaker on the roads and alert the driver or other components in the vehicle to apply appropriate action like brake, steer etc. The driver may miss to notice the speed humps in complex driving conditions especially during night time due to less / no lights. An un-noticed speed-hump on high speed is harmful to passengers especially pregnant women and elderly people travelled in vehicle. It also leads to damage the tires and tubes and even may damages the vehicle also sometime. Usually, speed humps / breakers are built in places like school zone, accident-prone locations, sharp Turing, congested residentialstreets, unmanned level crossings where speed of the vehicle needs to controlled to avoid accidents. The speed humps will have various physical identities such as height, length of ramps, depth etc. However, most of time it is not followed in India. Most of the humps in India are not being constructed and maintained according to the public safety guidelines of Indian Road Congress (IRC) i.e., IRC099, which is resulting in severe discomfort to the driverand even causing loss of direction control which is leading to fatalities.

According to the news published in Times of India on Sep 14, 2015, 14:42 IST, according to the Road Accident Report (2014) published by the road transport and highways ministry, while 4,726 lives were lost in crashes due to humps, 6,672 people died in accidents caused due to potholes and speed breakers.

1.1 Objective

According to Indian roads, different types of Humps are placed in roads to slow down the vehicle speed and sometimes installed in a series of several humps to prevent cars from speeding. Here we have captured few types and decided to detect it all by proposed approach.

Figure 1 a. Hump with markings   b. Hump with partially faded markings

Figure 2 a. Hump with fully faded markings   b. Hump sign board to alert driver

Figure 3 a. Hump in urban area with markings   b. Hump in urban area without markings

Figure 4 a. Hump in urban area with half-length of road b. Hump in urban area with thin span

Reference: Images from Figure (1 – 4) are captured by authors at nighttime. Students / research scholars can use these images for their studies.

1.2 Abbreviations

Abbreviations with expansion

Abbreviation	Expansion
ADAS	Advance Driver assistance System
TP, FP	True Positive, False Positive
TN, FN	True Negative, False Negative

1.3 Definitions

True positive is an outcome where the algorithm correctly predicts the positive class. **True negative** is an outcome where the algorithm correctly predicts the negative class. **False positive** is an outcome where the algorithm incorrectly predicts the positive class. **False negative** is an outcome where the algorithm incorrectly predicts the negative class.**positive class** videos/ images having hump
negative class videos/ images having no hump

2. Literature survey

Speed-Hump Detected using Embedded Sensors. Choi et al. (2012) developed an environment-detection and mapping algorithm for autonomous driving for both high way and off-road real-time environments. Environment detection-and-mapping taking inputs from camera as well as LiDAR for pedestrian crossing detection, lane detection, and speed-bump detection, and obstacle on the road detection. VisLab Embedded Lane Detector (VELD) and a camera are used for lane detection algorithm which provides the lane related information like location, width etc. The obstacle detection algorithm gets data from LiDARs. For the research purpose, apassenger car is designed with 6 LiDARs, 3 cameras, and personal computers. The algorithm mostly focuses on local obstacle map instead of using a global map for more accurate vehicle locating. From the collected information, the risk map and obstacle map is created by the model-based filter, which help the road users to know about the obstacles across the road.

Lion et al. (2018) proposed a cost-effective method of speed bumps detection and height estimation by using theMicrosoft Kinect Sensor. the edge of the speed bump is being detected by Canny edge detection. The system was not testedfor real-time videos. Both Kinetic sensor and camera inputs are used to detect speed bump.

A. Speed-Hump Detected using a smartphone

Chellaswamy et al. (2018) gave special importance to safety of the driver from the advanced driver assistance (ADAS) system. They used 3 axis accelerometers built within an Android-based smartphone to record and analyses the different environmental conditions which affect the health of the driver and the automobile. In real-time, the overall awareness of safety has increased for drivers by analyzing and alerting about the complexity of road. GPS tracking and GSM module was developed. GPS coordinates system, road condition maps are created through Google Earth. multiple-axis classification method is introduced for a better road anomaly detection system, which increase the bump and pothole classification accuracy.

Roma et. al. (2014) present a system to monitor road and traffic conditions. Nericell is system to collect a data by piggybacking with smartphones that people take with them. They mostly concentrate on the sensing component like accelerometer, microphone, GSM radio, and GPS sensors in smartphone map the speed-bumps, potholes, honking, and braking. They also tried to address many challenges like the arbitrary orientation of smartphone, honk detection and localization in an energy-efficient manner. The electiveness of the sensing functions using Nericell is experimented on Bangalore roads and produced significant results.

3. Solution approach

In this article we propose a navel approach to detect Hump during night time with help of image processing and considered below criteria to be satisfied

- Should run on Embedded platform
- Should perform on Real time / on the fly
- Should solve identified scenarios that mentioned in section 1.1

With above points, it is very clear that using artificial networks like deep learning may not be a good choice as it takes more time to execute in embedded hardware where memory and power is limited. Decided to use image processing gradient-based approach with minimal usage of machine learning, detailed steps have been captured in below overall architecture diagram,

Proposed framework is to detect speed hump based on Estimating gradient features with help of Sobel Edge detectors, in general, hump or speed brakers will be little higher than the road surface. Even though there are no markings or any indications, still it can be highlighted when heavy light fall at night time, there should be a small shadow behind the hump when car head lamp hit on it, this is one of the major assumptions here to solve this problem

Figure 5. Sample Input Image

The first step of this solution would be frames / images conversion from input videos, Real time videos / offline videos are used as input sources. Alternate frames will be processed for faster execution. Figure 5 shows the input frame extracted from offline or real time videos

3.1 ROI selection

Since it is running on embedded platform, optimization needs to be considered as much as possible, we have decided to perform Region of Interest to suppress the region which is away from vehicle trajectory. It may lead to cut down almost 60 percent of area in an original image, Figure 6 shows the area which is in vehicle trajectory and removed the area where further processing is not required. ROI will be calculated by detecting dynamic vanishing point

Below points are involved in detecting Vanishing Point

- Line extraction
- Line classification: Vertical and Horizontal
- Vanishing Point Detection through clustering the line according to its orientation

Note: as our subject matter is not ROI instead of Hump detection, have not covered above ROI much here.

Figure 6. Region of Interest (ROI)

3.2 Sobel edge detection

Core of the algorithm in our proposed approach is to detect gradient features of current frames to get highlighted part of the road surface. We are using Sobel edge detector in both Horz and Verz direction.

The Sobel operator performs a 2-D spatial gradient measurement on an image and so it highlights regions of high spatial frequency that correspond to edges.it will find the approximate absolute gradient magnitude at each point in an input grayscale image.

When using Sobel Edge Detection, the image will be processed in the X (Horz) and Y (Verz) directions separately first, and then combined both together to form a gradient image which represents the sum of the X and Y edges of the image.

As we are not dealing anything on color component of image so better covert RGB scale to a Grayscale image as a initial step, then can apply Sobel edge detector kernel in both direction as mentioned above, kernel is a 3 x 3 matrix consisting of differently (or symmetrically) weighted indexes with zero weighted at center. This will represent the Sobel filter that we will be implementing for an edge detection.

When we need to scan across the X direction of an image, have to use the following X Direction Kernel to scan for large changes in the gradient. Similarly, for Y direction of an image.

A Sobel edge detection operator consists of a pair of convolution kernels as shown in the succeeding text. The second kernel is simply a rotation of the first (Table 1).

Table 1 Sobel edge detection

-1	0	1
-2	0	2
-1	0	1

X- Direction Kernal

-1	-2	-1
0	0	0
1	2	1

Y- Direction Kernal

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, with one kernel for each perpendicular orientation. The kernels can be applied separately to surface measurement, to produce separate calculations of the gradient component in each orientation.

$$G_x = S_x * Z$$
$$G_y = S_y * Z$$

These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of the gradient. Figure 7 shows Gradient result of ROI

$$G=\sqrt{Gx^2 + Gy^2}$$

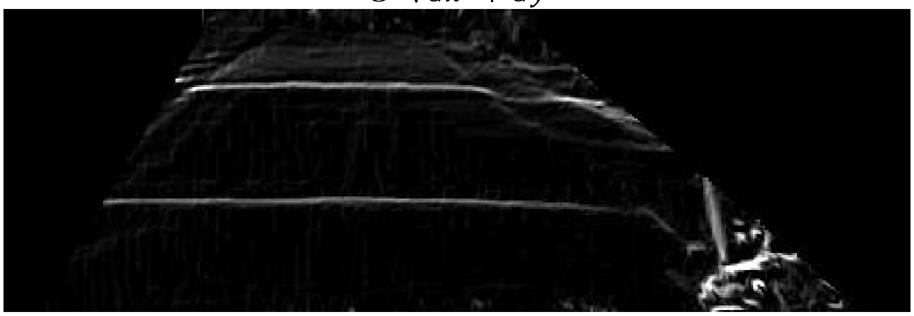

Figure 7. Gradient of ROI Image

3.3 Adaptive Thresholding

Once we received a gradient image after Sobel edge detection, to segment the candidate region where Hump may present binarization / Image thresholding is needs to be performed. Figure 9 shows the result of Adaptive Threshold.
Image thresholding used to segment a digital image based on a certain characteristic of the pixels (like intensity value). Approach needs a binary representation of the image to suppress the part that are not relevant to interest of object (the area may not be part of hump), each pixel will be classified into two categories, such as “Black” or “White”. This might be a common task in every image processing application, and few computer vision applications.
Steps involved

1) Compute histogram and probabilities of each intensity level.
2) Set up initial m_i and μ_i. Step through all possible thresholds.
3) Step through all possible thresholds t=1… maximum intensity
   3.1) Update m_I (weight) and μ_i (mean).
   3.2) Compute $\sigma_b^2(t)$(variance)

4) Desired threshold corresponds to the maximum $\sigma_b^2(t)$.

5) You can compute two maxima (and two corresponding thresholds) $\sigma_{b1}^2(t)$ is the greater max and is the greater or $\sigma_{b2}^2(t)$ equal maximum.

6) Desired threshold = $\frac{threshold1+threshold2}{2}$

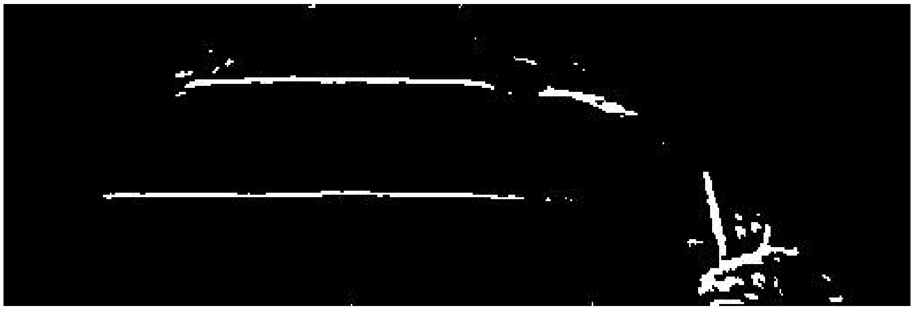

Figure 8. Adaptive Thresholded Image

3.4 Hump features extraction

This is the process of extracting the features from Hump candidates that produced from thresholder image in general, Hump will have unique features like specific span of length, width, perimeter, position that candidate of hump occurs in image, no of segmented pixels etc., Figure 9 shows the result after eliminating the pixels which is not related to Speed Hump

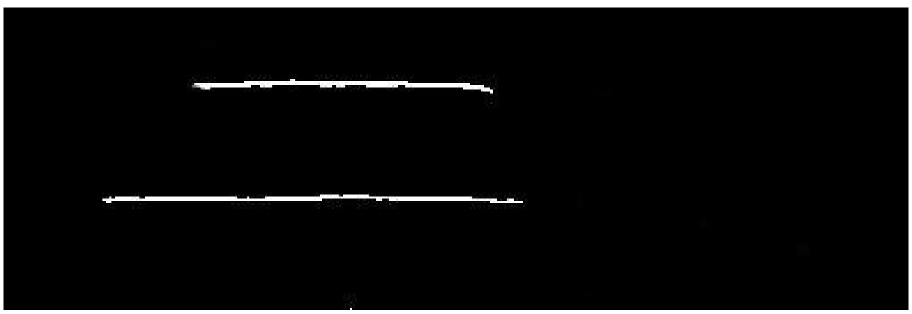

Figure 9. Perfect Hump candidate's image

3.4.1 Decision trees

To classify the Hump candidate either it is a Hump or not with generated features a decision tree classifier has been used. A Decision tree classifier is a recursive partition of the instance space. Decision Tree algorithm is one of the supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm able to solve regression as well as classification problems too. The objective of Decision Tree is to create a training model for predicting the class or value of the target variable by learning simple decision rules inferred from training data.

In Decision Trees, for predicting a class label for a given record we will start from the root of the tree. Then we compare the values of the root attribute with the record's attribute. In this way of comparison, we follow the branch corresponding to that value and jump to the next node.

One of common problem in decision tree algorithm is to select the root node out of n attributes, randomly selecting any node to be the root cannot solve the issue, may leads to poor accuracy. to solve this attribute selection problem, researchers worked and come up with suggestion using some criterion like information gain, gini index, etc

The information gain in the decision tree will be defined as the amount of information improved in the nodes before splitting them for making further decisions. Information gain can be calculated from below formula

$$\text{Information Gain} = 1 - \text{Entropy}$$

Formula for Entropy,

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log \mathrm{P}(x_i)$$

X – Random variable
Xi – possible outcomes
P(Xi) - probability of possible outcomes.

Here, leftmost (x) coordinate, start of the bounding box in the horizontal direction, The topmost (y) coordinate, start of the bounding box in the vertical direction. The horizontal size (Length) of the bounding box. The vertical size (Height) of the bounding box, The total area of the connected component. These are the attributes or features we used here to classify the candidate as Hump or not by using Decision Trees, Figure 10 shows the algorithm flow for the steps involving in both training and testing

4. Overall approach has clearly explained in below diagram

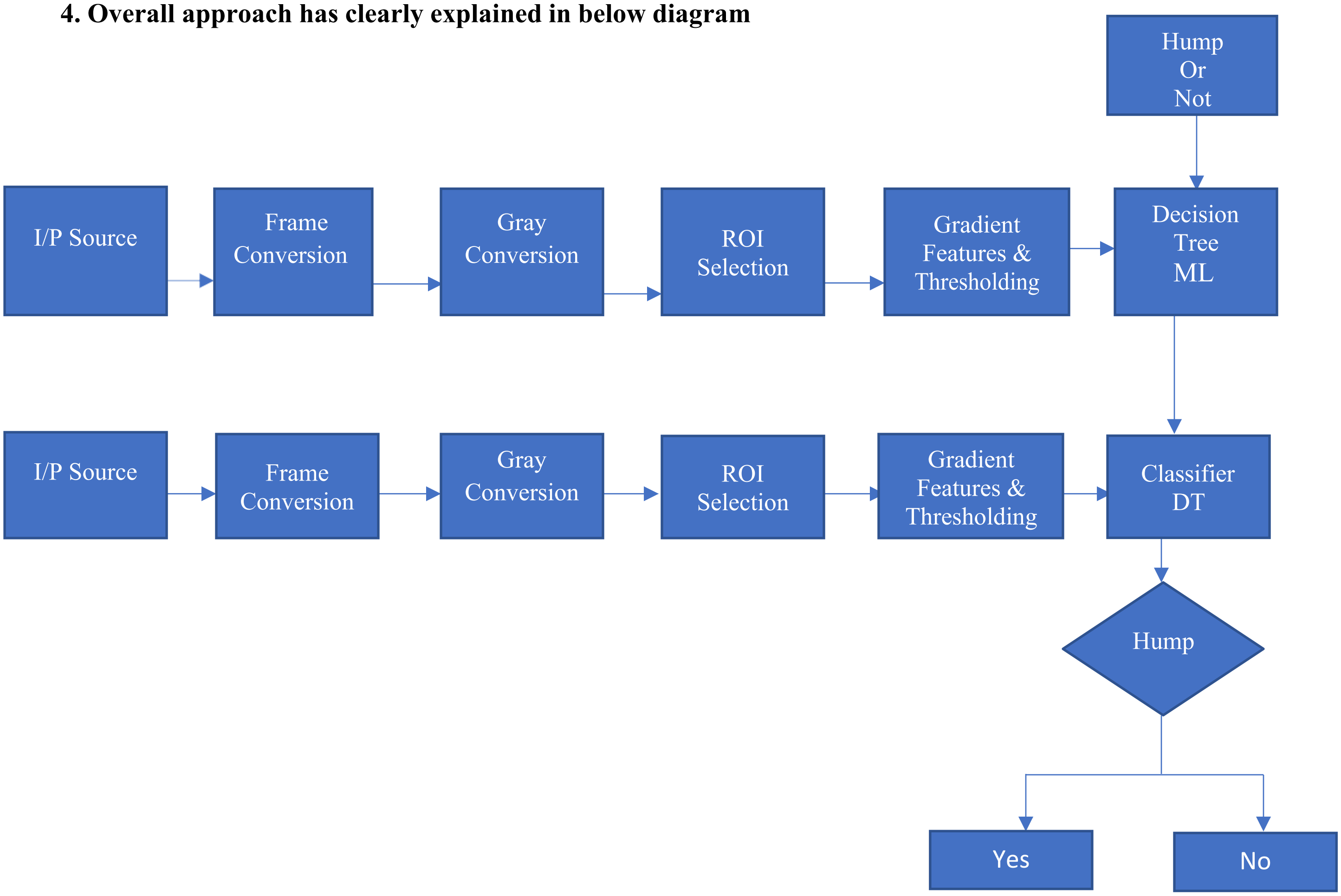

Figure 10. Block diagram of Hump detection

4.1 Proposed Hump detection – Algorithm Flow

Training

Step1: Frame separation from Input videos

Step 2: Gray conversion

Step 3: vanishing point detection

Step 4: Region of interest selection

Step 5: Sobel edge detection on ROI

Step 6: Gradient features extraction

Step 7: Adaptive thresholding Binarization

Step 8: Hump properties features extraction

Step 9: Training with Decision Tree Machine Leaning algorithm

Prediction

Step1: follow the same steps from 1 to 8

Step 2: Testing with Decision Tree Machine Leaning algorithm

5.Dataset

There is no public dataset available for speed hump/bump. We have captured around 100 videos with minimumof 5 mins of each video by creating the set up in our own vehicle, and volume of data comes around 30 GB. The datasetconsists of with and without marking bumps, humps, rumble strips, shaded bumps due to trees/buildings, bumps with faded paints. Dataset covers all scenarios mentioned in section 1.1. as we planned to scale up this solution to a worthfulproduct in market, very much keen on taking of real time videos with scenario that we face in our daily life. we capturedvideos in and around KTC Nagar and Palayamkottai area in Tirunelveli district 627000, Tamil Nadu, India, from swift car by mountingthe camera on rear view mirror

6. Sample results

Figure 11 a. Input frame b. Frame with Hump detection

Figure 12 b. results of proposed Approach, red color clearly shows the ROI wherein algorithm supposed tobe performed and green line highlighted the Hump, detected by this approach

7. Results and Discussion

We evaluate our proposed Hump detection method based on an image dataset with volume of 30 GB sized videos. Datacaptured in front camera mounted on vehicle. Algorithm is developed on Intel Core i5 processor with 8GB RAM. For better analysis, approach has been tested for 2 different datasets wrt to TP (Table 2) and FP (Table 3), TP videos will have Hump and expected metric will be TP and FN likewise FP and TN are expected for FP videos which does not have Hump,

Table 4 shows the overall accuracy. Achieved table 2

7.1 TP Results

Table 2. TP videos with accuracy metrics

Video type	Duration (Mins)	TP (%)	FN (%)
Hump videos	100	90	10

7.2 FP Results

Table 3. FP videos with accuracy metrics

Video type	Duration (Mins)	FP (%)	TN (%)
Clean videos	100	2	98

7.3 Confusion Matrix

Table 4. videos with accuracy metrics

Samples Type	Predicted No	Predicted Yes	No of samples
Actual No (clean)	TN: 95	FP: 5	100
Actual Yes (Hump)	FN: 2	TP: 98	100

8. Future improvements

- Current algorithm may not perfectly fit for Day time, as this paper only focus on night time,upcoming version may solve this.
- There should be improvisation on performances
- Going forward approach may able to classify potholes also along with Hump

9. Conclusion

The idea of this paper is to detect hump to prevent or mitigate the collusion and vehicle damage by hitting onit. The proposed method has resolved the identified problem and been proven by shown metrics for various dataset with all environmental scenarios like heavy rain, foggy, mist etc., proposed approach uses Sobel edgedetection. Since this paper uses very basic image processing features such as gradient and physical propertiesof hump. we can use this solution in any embedded platform without much effort.

References

Tentative Guidelines on the Provision of Speed Breakers for Control of Vehicular Speeds on Minor Roads. *Indian Roads Congress* Jan 1988. (Accessed on April 2, 2018).

Choi, J., Lee, J., Kim, D., Soprani, G., Cerri., P., Broggi, A., & Yi, K., Environment-detection and mapping algorithm for autonomous driving in rural or offroad environment, *IEEE Transactions on Intelligent Transportation Systems*, 13(2), 974-982,2012.

Yun, H. S., Kim, T. H., & Park, T. H., Speed-Bump Detection for Autonomous Vehicles by Lidar and Camera. *Journal of Electrical Engineering & Technology*, 14(5), 2155-2162,2019.

K. M. Lion., K. H. Kwong., and W. K. Lai., Smart speed bump detection and estimation with kinect,*4th International Conference on Control, Automation and Robotics (ICCAR),* Auckland, pp. 465- 469,2018.

chellaswamy C., Anusuya T., Famitha H., Amirthavarshini S B., IoT Based Humps and Pothole Detection on Roads and Information Sharing‖, *International Conference on Computation of Power, Energy, Information and Communication*, 2018.

Roma K Goregaonkar., and Snehal Bhosale., Assistance to Driver and Monitoring the Accidents on Road by using Three Axis Accelerometer and GPS System, *International Journal of Electronics Communication and Computer Engineering*,5(4) 260- 264, 2014.

Mohan, P., Padmanabhan, V. N., & Ramjee, R., Nericell. *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems - SenSys* '08. doi:10.1145/1460412-1460444, 2008.

Gunawan, F. E. Yanfi, B. Soewito.,A vibratory-based method for road damage classification, *International Seminar on Intelligent Tech-nology and Its Applications (ISITIA)*,1-4,2015.

Kiran, M. V. K. Vimalkumar K., Vinodhini, R. E.. Archanaa., R . An Early Detection-Warning System to Identify Speed Breakers and Bumpy Roads using Sensors in Smartphones., *International Journal of Electrical and Computer Engineering*,7(3):1377-1384,2017.

Pooja, P. R., Hariharan, B. An early warning system for traffic and road safety hazards using collaborative crowd sourcing,*International Conference on Communication and Signal Processing (ICCSP)* , 1203-1206,2017.

Biographies

Jebastin k has completed his Master degree in Engineering in Joe Suresh Engg college, Tirunelveli, Tamil nadu, did his Bachelor of Engineering in National college of Engg, Tirunelveli, Tamil nadu, working as Asst Professor in PSN College of Engineering and Technology, melathediyoor, Tirunelveli, Tamil nadu for more than 10 Years.

Dinakaran k has completed his Bachelor of Engineering in National college of Engg, Tirunelveli, Tamil nadu, he worked on image processing applications such as Object detection, pedestrian detection, Optical character recognition for automotive especially ADAS, Aerospace, Medical, Product inspection domain.

Vishnu has completed Bachelor of Engineering in P.A College of Engg, Pollachi, Tamilnadu. He has experience in real-time Image processing, embedded, ROS and Deep learning product development. He has worked on several deep learning models like COVID-19 face mask compliance, social distancing calculation, etc.,