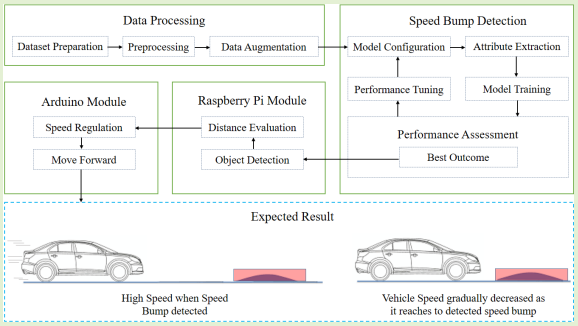
# Deep Learning-Based Speed Bump Detection Model for Intelligent Vehicle System Using Raspberry Pi

Deepak Kumar Dewangan and Satya Prakash Sahu, *Member, IEEE*

***Abstract*—Artificial intelligence in vision based approaches have proven to be effective in various phases of intelligent vehicle system (IVS). An IVS has to intelligently take many critical decisions in heterogeneous environment. Speed bump detection is one such issue in real world due to its varying appearance in dynamic scene. The major issue is the scaling appearance of such objects from far distance and often viewed as small entity. In the proposed article, deep learning and computer vision based speed bump detection model is proposed, which assist and control the driving behavior of an IVS before it reaches to speed bump. The behavior of IVS has been explored and tested by incorporating the proposed method with a real time embedded prototype and found to be more efficient and comparable with state-of-art techniques. The overall performance of the proposed model has been achieved in terms of accuracy, precision and F-Measure as 98.54%, 99.05% and 97.89% respectively in the prepared real time environment.**




***Index Terms*— Artificial intelligence, autonomous vehicle, computer vision, deep learning, speed bump detection, vision sensors.**

## I. Introduction

SPECIALIZED development in automobile sector has directed various organizations and institutions to device remote and autonomous vehicles such as intelligent vehicles, unmanned air vehicles (UAV) and drones, which are just the few examples of new generation in the transportation industry. Among all these examples where human is directly involved, safety is the primary concern especially in case of road transportation, as the chances of hazard cannot be ignored. According to the report of world health organization (WHO) estimated number of road traffic death are very high [1] and causes of these situations are random but primarily includes casual driving and inability to understand the scene [2]. Besides, researchers have started to think on ideas of providing the safety in the form of better scene understanding capability through various sensors. To comprehend the heterogeneous road environment, sensors like radar [3]–[6] and LiDAR [7]–[9] have been equipped in IVS but sometimes restricted to perform due to their fluctuating wavelength, inappropriate data patterns and have high functioning cost. Moreover, deep learning techniques enabled vision based sensors are gaining popularity [10] through which an IVS is capable to understand the scene and can take better decisions in complex situations. However, performance of these sensors rely on the feature representation of intended objects.

In real world situation, it is somewhat difficult to analyze all road entities due to their representation complications. Recently, computer vision based approaches in IVS have become popular for the detection of various objects in a dynamic environment and process them according to feature vector representation. Literature reveals the major phases or modules for the design of IVS or advanced driver assistance system (ADAS) as road detection, lane detection, pedestrian detection, vehicle detection and many more. These pre-known objects have defined and distinct features that a system can observe and identify them at run time. But, some of the real time objects that are mostly missing in context of autonomous driving such as road rail guards and speed bump; which controls the driving behavior of vehicles in real world scenario. Recognizing speed bump from a far distance is significant since it may affect mechanic assembly of vehicle and may lead to hazardous situation which is dangerous for human safety. It is somewhat difficult to understand the different categories of speed bump (marked and unmarked) due to their appearance mostly from the far distance. An IVS is expected to regulate its driving speed when it finds such object and ignore the chances

Manuscript received August 6, 2020; revised September 18, 2020; accepted September 21, 2020. Date of publication September 28, 2020; date of current version January 6, 2021. The associate editor coordinating the review of this article and approving it for publication was Prof. Yongqiang Zhao. *(Corresponding author: Deepak Kumar Dewangan.)*

The authors are with the Department of Information Technology, National Institute of Technology, Raipur (Chhattisgarh) 492010, India (e-mail: dkdewangan.phd2018.it@nitrr.ac.in; spsahu.it@nitrr.ac.in).

Digital Object Identifier 10.1109/JSEN.2020.3027097

of hazard. Understanding the features and detection of speed bump is little explored in existing research studies and had not been associated for the design of IVS.

These observations inspire us to offer a low cost feature descriptor for speed bump detection model. The proposed model is based on vision based approach and has been developed with real time embedded system prototype for IVS. It recognizes marked and unmarked speed bump from a far distance and capable to make decisions accordingly. The proposed prototype utilizes the techniques of neural network to learn features of speed bump and helps to regulate its driving behavior in the presented scenario. For this we have utilized vision cameras which are strategically useful over other stated sensors and have following key features: (1) Compatible in variable color band and ability to process high resolution to observe the scene. (2) Focused range of perspective evaluation.

Diverse applications of ADAS such as road detection, lane detection, vehicle detection etc. are getting significant attention using vision based AI in transport industries. Performance of these systems depend on the scene understanding capability through vehicle's mounted camera. However, existing system consist of expensive hardware, sensors and often requires high computation cost. Also, the true potential of an autonomous vehicle is not revealed about taking decision against speed bump objects. In the adopted approach, speed bump detection through vision camera is facilitated and also ensures the intelligent driving behavior of prototype vehicle against the detected speed bumps. Structure of rest of this article is organized as follows. Related work is provided in section II whereas in section III, adopted strategy along with its working is discussed. The experimental outcome of the proposed model are reported in section IV. Finally, concluding remarks and future directions are mentioned in section V.

## II. Related Works

Various model and feature based mechanism have been utilized to learn significant features of speed bump. Based on the feature processing techniques, studies on scene understanding for IVS has been recognized [11]–[19]. Towards speed bump detection, we have organized various studies into the following categories: 3D point cloud based approaches [20]–[23], sensor based approaches [24]–[31], and vision based approaches [32]–[36]. Finally, we express the workflow of proposed approach in scope of the work.

### A. 3D Point Cloud-Based Approaches

Obtaining significant features of point cloud data consisted the calculation of height variation, intensity of laser points, determining the z point (in 3D plane), road surface roughness representation and data point intensities in two diverse color spaces. These features were processed and examined the height of data points and the change of slope to the road boundaries [20]. J. Byun *et al.* applied markov random field (MRF) to differentiate between drivable and non-drivable road area. The computed feature vector formulation is then approximated to the goal region using principal component analysis (PCA) on the neighborhood covariance matrix [21]. On the other hand, Suzzane *et al.* measured dimension of speed bump type of objects using the carrier frequency which was operated with interferometric approach on 24 GHz. This scheme assured a substantial level of sensitivity and appropriate for related types of application [22]. Surface modelling used the mobile laser scanning (MLS) to yield an even surface and thus the depth representation within reachable and feature-conserving framework got improved [23].

### B. Sensor-Based Approaches

Speed-breaker early warning system (SWAS) was offered on a given time series of 3 axis accelerometer data to identify speed bump and hump type of objects. Moreover, it utilized amplitude vector representation to eliminate the overhead associated with fine-tuning of accelerometer [24]. In another approach, data elements of acceleration sensor and location in global positioning system (GPS) module were combined and processed to localize the speed bump object. Devices were calibrated and investigated on variable road surfaces by an application called "Advanced Road Traveller" [25]. Similarly, scheme of sensor's signals on speed bump were detected using magnetometer, accelerometer, and GPS. Location and time label of the extracted data were then sent to the server that employed k-means clustering to process the sensor's data [26]. In another study of sensors data, Manuel *et al.* engaged smartphone accelerometer to recognize several road anomalies. It consisted of sliding window method that pre-process the sensors data, then obtained the statistical features, and applied support vector machine (SVM) to classify those anomalies [27]. In another approach, Jose *et al.* deployed genetic algorithm (GA) and operated over accelerometer, gyro features, and GPS sensors data to spot speed bumps on the road [28]. To analyze road conditions; Luis *et al.* employed machine learning techniques and characterize the accelerometer data by feature pool as Bag of Words (BoW) [29]. Some of the studies involved smartphone based applications where inbuilt sensors were used to observe the fluctuation patterns of vehicle's driving through variable road surfaces like speed humps, bumps, uneven roads, cracks, potholes, and smooth road. To counter imminent frontal impacts as an early warning system, smartphone based accelerometer sensors were processed [30], [31].

### C. Vision-Based Approaches

Traditional practices of median and Gaussian filtering, connected component and binary image conversion had been applied to recognize speed bump, mostly to those road areas where speed bump built with small height. However, finding such bump areas are still need to be verified under poor road structure and low illumination conditions [32]. Hua *et al.* applied speed up robust features (SURF) with BoW and employed K-means clustering to construct a visual dictionary to detect vehicle shakes. Further, obtained feature maps were then applied to classify speed bump and non-speed bump using SVM [33]. Jeong *et al.* delivered a method to deal with pitch angle valuation between camera and the road surface, noisy range measurements, and interference from obstacles. These stated issues were handled by incorporating free space valuation, calculation of digital elevation map (DEM) and

road surface profile approximation [34]. The time-sliced image sequences involved small fraction of data and very less redundancy observed in consecutive frames. Using this approach, recognition of speed bump used time-sliced image generation, analysis of vertical motion, and further processed for road bump localization [35]. Moreover, Varma *et al.* has utilized deep learning approach on android phone to detect marked and unmarked speed bumps [36]. Similarly, Sandeep *et al.* has also applied deep learning techniques to detect the speed bump object on the road using a pre-trained ResNet-50 network and YOLO algorithm [37].

*Scope of the Work:*

In existing approaches, detection of speed bumps rely mostly on smartphone sensor due to their operational simplicity. But these sensor-based approaches are not very proficient in challenging road surface environment. Likewise, sensors like LiDAR and radar may not portray the actual scene effectively, producing a malformed pattern at specific points which are comparatively higher than the actual estimation thus may lead to false detection in given scenario. Various image processing practices identifies speed bump to a firm level, but have limited effectiveness under shadows or low illumination. Despite these, vision based approaches may be fused up with deep learning practices that may assure the precise detection of speed bump while extracting significant features; which are little explored in most of the existing state-of-the-art techniques. Marked speed bump have similar feature representation as zebra crossing from a perspective view. Distinguishing between these requires robust feature representation that may benefit an autonomous vehicle to modify its current driving behavior and make decisions accordingly. To overcome the issues discussed, a computer vision based convolutional neural network (CNN) model is proposed to learn the significant features of speed bump objects and thus improve the detection performance of speed bump. Further, it not only detects the speed bump with high accuracy but also provides an early warning to the developed prototype vehicle to avoid any kind of mishap associated with speed bump and suitable to take logical driving decisions in the presented scenario. The main contribution of the proposed work is précised as follows:

(1) A new computer vision and CNN based model has been designed for the accurate detection of speed bump to overcome the limitations of various sensors.

(2) Optimal hardware resources are identified to deal with speed bump detection and their operative compatibility towards the early warning system and decision making capability for the proposed prototype vehicles.

(3) Proposed approach uses on-board camera unit and computing device (Raspberry Pi device mounted on vehicle prototype) to validate the concept of speed bump detection in prepared real time scenario.

## III. Proposed Method

Initially, the proposed approach consist of determining the learning capability of proposed model for speed bump objects and then speed regulation of the vehicle prototype against the detected speed bump is followed. In this direction, following stages are taken into account:

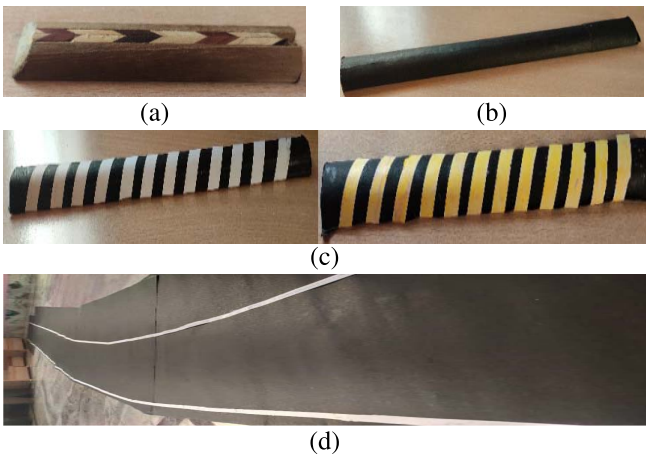



Fig. 1. (a) Wooden material used for Speed bump. (b) Unmarked speed bump. (c) Marked speed bump (white and yellow). (d) Prepared road / lane environment.

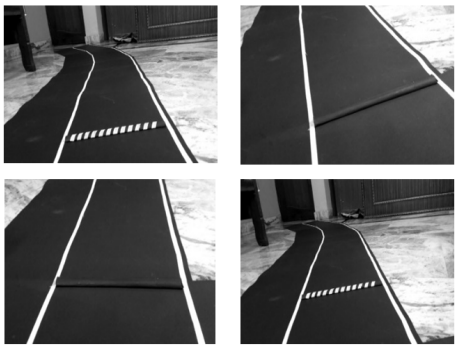


Fig. 2. Sample image grid of prepared dataset.

### A. Primary Setup

To validate the proposed vehicle's intelligent behavior against the idea of visual task, varieties of speed bump object along with real time alike road / lane environment is prepared. Black paper sheets and wooden materials (or any similar item) is utilized to shape the road environment and speed bump object respectively. Marked and unmarked categories of speed bump are considered in the task. The length of these objects are subjective and depends upon the prepared road environment. We have set the distance between lane markings approximately to 30 cm and the same dimension is used for speed bump length, whereas its width is set to around 2.2 cm and the considered height is.7 cm. This is worth noting that, the width dimension is not much extended because larger width may represent speed hump not the bump object. Assembled visuals of the setup in indoor environment is represented in figure 1.

### B. Dataset

In order to learn significant features of speed bump objects, proposed model in the prepared setup rely on the speed bump images. Therefore, we have captured 575 images for marked, unmarked (white and yellow) categories of speed bump from various viewpoints through smart camera and raspberry camera module. Samples captured in the environment can be visualized through figure 2.

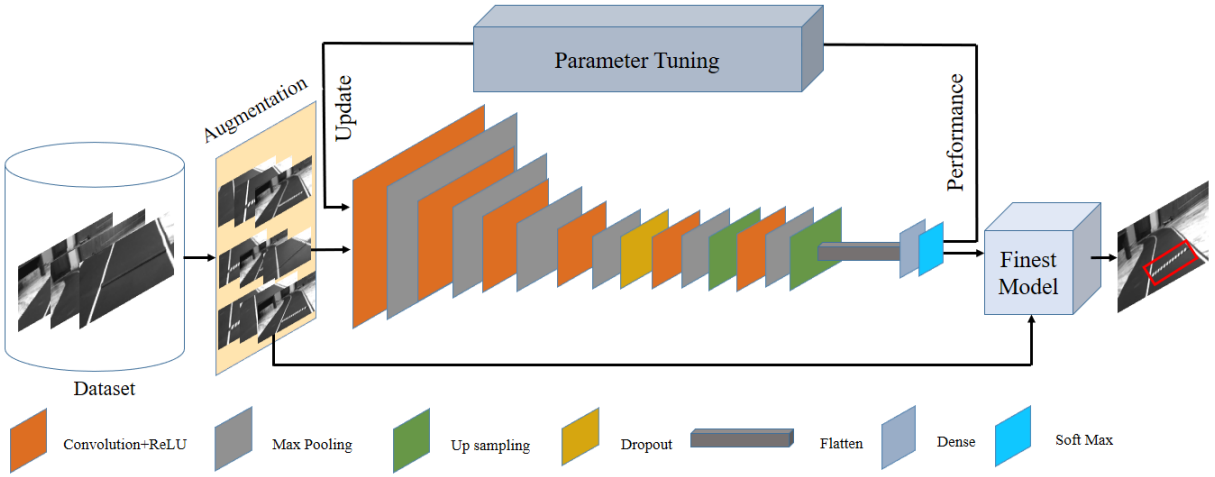



Fig. 3. Proposed speed bump detection using covolutional neural network (CNN) architecture.

### C. Data Preprocessing and Augmentation

Smoothening and denoising operations [38], [39] has been applied to all images in the prepared dataset as a preprocessing step. This process supports the model to obtain more intrinsic characteristics of the speed bumps without compromising the significant features. However, reliable and truthful detection of CNN based approaches often involves operating a large amount of data. Due to this, we applied data augmentation technique through warping, flipping, rotation, noise, blurring and thus increasing the size of the dataset to 3450 samples.

### D. Architecture

Offered architecture is developed for accurate and fast speed bump detection that makes it optimally compatible for the proposed AI enabled vehicle prototype application. It takes input frame as an input and provides speed bump objects enclosed in a bounding box. Initially, a set of filters having size of 7*7, 5*5 and 3*3 has been investigated; out of which 5*5 and 3*3 is found suitable to obtain the robust features for speed bump objects. Procedure in this approach mainly involves extraction of more inherent features to deal with multi scale speed bump objects. The architecture is characterized and structured with various layers of convolution, pooling and fully connected layers as shown in figure 3. The feature vectors for speed bump proposal regions are transferred to the pooling layer to extract the defined size feature vectors. This defined sized vectors are then transmitted to the fully-connected layer to assist classification and localization of speed object proposal regions. Padding of similar dimension is used to process spatial magnitude, which provides equal opportunity for all the pixels in the boundary region to participate in learning the features from the detected object in the frame. Similarly, the filter with stride of unit movement is considered in the feature map. At each stride location, k number of object proposal regions are instantaneously processed and recorded. Summarized details of operational layers and other significant functions are described in table I. Similarly, the structure of proposed architecture is unrolled and represented in table II where each layer is characterized with its configuration details as represented in figure 3.

TABLE I
OPERATIONAL LAYERS

| Layers | Operational Expressions |
|---|---|
| Convolutional | $C(x,y) = (I * k)[x * y] \sum_{a}^{max} \sum_{b}^{max} k[a,b] I[x-a, y-b]$ |
| Pooling | $Pooling_{max}(C) = max_t P_r$ |
| Dense | $F(Weight * X + Bias)$ |
| Optimizer | $p_t = \beta_1 p_{t-1} + 1 - \beta_1 g_t; q_t = \beta_2 q_{t-1} + 1 - \beta_2 g_t^2$ |
| Learning Rate | $w_{i+1} = w_i + (\alpha * \delta * input)$ |

$C$ = convolutional operation, $I$ = input image, $k$ = kernel, $x$ *and* $y$ = feature vector indexing of row and column, $P$ = activation score from attained max pooling strategy, $X$ = input vector containing number of neurons of preceding layer, $p$ *and* $q$ = moving averages, $g$ = mini batch gradient, $\beta$ = hyper parameter, $\alpha$ = rate of learning, $\delta$ = target output.

TABLE II
CONFIGURATION DETAILS OF LAYERED ARCHITECTURE

| Layers | Operational Layer | Output Shape | Filter Size | Weights |
|---|---|---|---|---|
| 0 | Input | 64, 64, 3 | - | - |
| 1 | Convolution 2D | 64, 64, 200 | 5, 5 | 15200 |
| 2 | Activation | 64, 64, 200 | - | 0 |
| 3 | Max Pooling | 32, 32, 200 | 2, 2 | 0 |
| 4 | Convolution 2D | 32. 32. 200 | 5, 5 | 1000200 |
| 5 | Activation | 32, 32, 200 | - | 0 |
| 6 | Max Pooling | 16, 16, 200 | 2, 2 | 0 |
| 7 | Convolution 2D | 16, 16, 150 | 5, 5 | 750150 |
| 8 | Activation | 16, 16, 150 | - | 0 |
| 9 | Max Pooling | 8, 8, 150 | 2, 2 | 0 |
| 10 | Convolution 2D | 8, 8, 150 | 3, 3 | 202650 |
| 11 | Activation | 8, 8, 150 | - | 0 |
| 12 | Max Pooling | 4, 4, 150 | 2, 2 | 0 |
| 13 | Dropout | 4, 4, 150 | - | 0 |
| 14 | Convolution 2D | 4, 4, 100 | 3, 3 | 135100 |
| 15 | Activation | 4, 4, 100 | - | 0 |
| 16 | Max Pooling | 2, 2, 100 | 2, 2 | 0 |
| 17 | Up Sampling | 4, 4, 100 | - | 0 |
| 18 | Convolution 2D | 4, 4, 128 | 3, 3 | 115328 |
| 19 | Activation | 4, 4, 128 | - | 0 |
| 20 | Max Pooling | 2, 2, 128 | 2, 2 | 0 |
| 21 | Up Sampling | 4, 4, 128 | - | 0 |
| 22 | Flatten | 2048 | - | 0 |
| 23 | Dense | 2 | - | 4098 |
| 24 | Activation | 2 | - | 0 |

Total Parameters: 2,222,726

Training process involves adaptive moment estimation (Adam) to identify the behavior of the model [40] and learning rate of 0.001 for 32 batch sizes are set and used. The smaller batch size (32) has been chosen in order to provide the


Fig. 4. Distance measurement setup.

regularizing behavior and avoiding the generalization faults. For the given input frame, network produces prediction of the intended speed bump object enclosed within the bounding box. First of all, the operational filters and pooling layers decreases the input dimensions. Then, the features are obtained with the combination of convolutional and other mentioned layers to make predictions for detection of speed bumps by a classifier. For avoiding the underfitting and overfitting issues, the dropout function has been used for regulating the associated weights by dropping the nodes based on some criteria. Then, the updated weights are spread across all the remaining nodes and assist to gradually shrink and optimize the model.

Assuming that the feature maps ($FM_i$) are the transitional outcome during training of proposed model. As each feature map parameters may have distinct spatial positions against image of $Height \times Width$ dimension, then the similar position will be shared with represented $FM_i$ elements. The elements of the $FM_i$ at the same pixel position have identical receptive region in the image. Hence, $FM_i$ units of different layers have receptive regions of variable sizes on the images and they are utilized to notice objects of different size. Therefore, high-level features processed at the last layers are raised with low-level statistics that helps the model to learn significant features for detecting different size objects. This information is utilized to anticipate the category from a precise receptive region and neutralize the region box nearby the intended multiscale objects in the image.

### E. Object to Camera Distance Calculation

Once the object is detected that is enclosed in a bounding box, the very next approach for an IVS is to measure the distance of this object in real world. Calculating this distance is necessary aspect of IVS; as it has to change its driving behavior according to the detected speed bump. Work in [41] and [42] involves distance measurement for detected vehicle object in road environment. The performance under standard condition was steady but requires high computation. To overcome this, simple logic is employed to determine the actual distance of object to the camera. It is obvious that whenever an object moves away from the camera its size is reduced and increased when it moves closer to the object. Using this general perception, the bounding box that holds speed bump object assist to determine the distance from the camera and can be visualized in figure 4. The left and right boundary markings are considered as Point 1 and Point 2. The difference of Point 2 and Point 1 is easily obtainable and measured in pixels. The difference between these two points varies with the scale of object and depends on the vehicle movement towards the speed bump object. It allows to derive the linear equation for calculated difference between Point 2, Point 1 and the tentative distance D1 and D2. For example, the D1 for a random position on track of speed bump to the vehicle position is taken as 60 cm, and let the difference between Point 2 and Point 1 is 70 pixels. Similarly, another distance D2 is set to 90 cm for speed bump to the vehicles fix position and pixel difference between similar points is 45 pixels. Using these parameters we solve equations of y=mx+c for 60=70m+c and 90=45m+c.

Once the parameters of equation is determined, then the value of m and c will be utilized to calculate the distance between object and vehicle on every movement (in centimeter) towards the speed bump. The moment when vehicle comes closer to object, its size gets increased and is computed with dynamic change in pixel positions (Point 2 and Point 1). This helps to represent the actual distance which is closely related to the focal length of camera in real time scenario.

TABLE III
EVALUATION MEASURES AND THEIR EXPRESSIONS

| Evaluation Measures | Operational Expressions |
|---|---|
| Accuracy | *TP+TN/(TP+TN+FP+FN)* |
| Sensitivity | *TP/(TP+FN)* |
| Specificity | *TN/(TN+FP)* |
| Precision | *TP/(TP+FP)* |
| F-Measure | *2*TP/(2*TP+FN+FP)* |
| FPR | *1-(TN/(TN+FP))* |

$TP$ = true positive, $TN$ = true negative, $FP$ = false positive, $FN$ = false negative, FPR = false postive rate.

## IV. EXPERIMENTAL RESULTS AND DISCUSSION

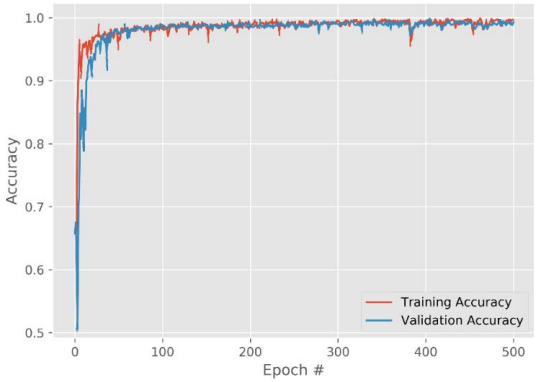
The proposed model for speed bump detection discussed in preceding section have been implemented and examined in real time environment. Key parameter such as Adam optimizer, learning rates, batch size and categorical cross entropy loss are well tuned to make the model more operative. Moreover, 500 iterations are discovered as an optimal stopping criteria for safe convergence during model training. Model design and training is executed under Intel (R) Xeon (R) W-2175 CPU @ 2.50 GHz CPU, 128 GB RAM having 64 GB of shared and 8 GB of dedicated GPU capability. Similarly, real time execution consists of scene processing through camera module Raspberry Pi 3+ and Arduino for motor / wheel operation of vehicle prototype. Major substantial phases are discussed as follows:

### A. Evaluation Measures

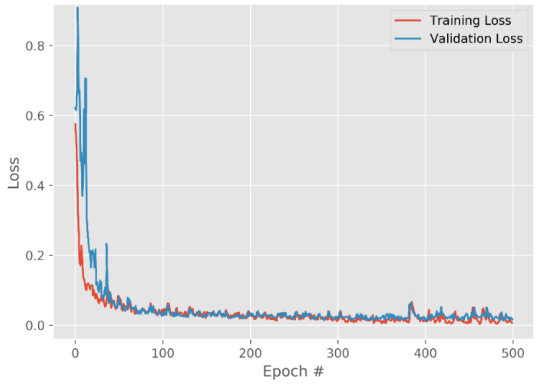
Performance of speed bump detection is examined through standard evaluation metrics, built upon confusion metrics featured by its true positive (TP), true negative (TN), false positive (FP) and false negative (FN) elements. Table III represents brief description of observed measures.

### B. Performance of Speed Bump Detection Model

During the training phase of speed bump detection, the behavior of model in the current state at each step is determined and evaluated. Similarly, to confirm the optimal performance; validation criteria is also considered during training and both parameters (training and validation) are represented by learning curves. From a computational viewpoint, performance in terms of numerical stability is needed without compromising the detection accuracy at the higher stages.




(a)




(b)

Fig. 5. Qualitative performance graph of proposed model during training. (a) Accuracy. (b) Loss.

For this, activation function softmax is utilized to approximate a complex and random function during forward propagation through $\hat{S}_q = e^{t_q}/\sum_{x=1}^{m} e^{t_p}$. The $\hat{S}_q$ represents the $q^{th}$ point of $\hat{S}_q$ and $t_q$ is the $q^{th}$ point of linear variable S. In this approach, if some elements of $t_p$ are large then the optimum value for loss is not obtained. To deal with this situation we applied subtraction and normalization and the chances of underflow is also reduced by combining this function with categorical loss.

Also, we observe the change to the learning rate (LR) is not linear and is dependent on the batch size. Too large or small values of LR may degrade the model performance and shows its inefficiency on learning the significant features. Therefore, after so many trials, the empirical value of LR taken as 0.001 and decay of 0.1 that are found suitable and optimum for the mentioned scenario and achieving the maximum accuracy. With the close observation it is found that the training accuracy congregate to its maximum accuracy in the earlier stages of training (refer figure 5 (a)). Similarly, validation accuracy also converges to its extreme point while moving together with training accuracy. It is worth mentioning that the gap between training and validation accuracy is almost reduced and indicates the absence of overfitting and under fitting problem. Moreover, the perceived fluctuations are almost steady and shows the stable performance of model. The optimization process (Adam) is implemented to determine the behavior of model towards training error. It is much expected that the designed model will have least error during training.

TABLE IV
QUANTITATIVE SCORE OBTAINED FOR STANDARD EVALUATION MEASURES

| Evaluation Measures | Score |
|---|---|
| Accuracy | 98.54% |
| Sensitivity | 96.75% |
| Specificity | 99.50% |
| Precision | 99.05% |
| F-Measure | 97.89% |
| FPR | 0.49 |

The training and validation loss can be visualized from figure 5 (b) that indicates model loss reaches to approximately zero. However, initial loss occurred at very high points (>0.8) but it gradually decreased and converged in an optimal way. Also, the oscillations of error for both training and validation are not that much inconsistent as they were at their initial stages.

Apart from the qualitative evaluation represented in figure 5, we also express the quantitative analysis as discussed in section 4.2. From the mentioned metrics, the computed accuracy of 98.54% is achieved. However, in case where class distribution is not uniform we also computed F-Measure which is well suited for imbalanced class distribution. But, both accuracy and F-Measure confirms the steady behavior of model. Similarly, precision score nearly equal to 1 describes that the model relevancy towards its detected data. These numeric score confirms the model true potential and verified the qualitative performance in the same direction. Obtained scores for other metrics are given in table IV.

### C. Vehicle Prototype Configuration

Run-time implementation of speed bump detection requires embedded vehicle prototype and for this Raspberry Pi 3B+ model is used. The utilized resource has the computing capability of 1.4GHz 64-bit quad-core ARM Cortex-A53 CPU and 1GB LPDDR2 SDRAM module, sufficient for the proposed model operation. One monocular camera unit of 8 megapixel is attached with the module and able to capture static images of $3280 \times 2464$ pixel and supports 1080p30 video processing. A distinct module of Arduino Uno microcontroller (for wheel / motor processes) has also been used with Raspberry Pi module using serial communication. Pi module captures the frames, identifies the speed bump object within a bounding box, calculates the distance from vehicle to the object and sends the processed information to Arduino device to modify its driving behavior according to the scene. Developed prototype vehicle can be visualized in figure 6.

### D. Intelligent Behavior of Vehicle Using Raspberry Pi and Arduino Communication

Apart from the proposed approach, we have incorporated the vehicle's driving mechanism towards defined lane portion. It uses standard image processing concepts to identify the lane marking and lane center position; so that each time it deviates from the lane center (LC) position, its mechanism tries to keep it to the center of the lane. Against LC position, Raspberry sends 6 different signals (0-lane center, 1-little deviated to the right lane, 2- extremely deviated to right lane,

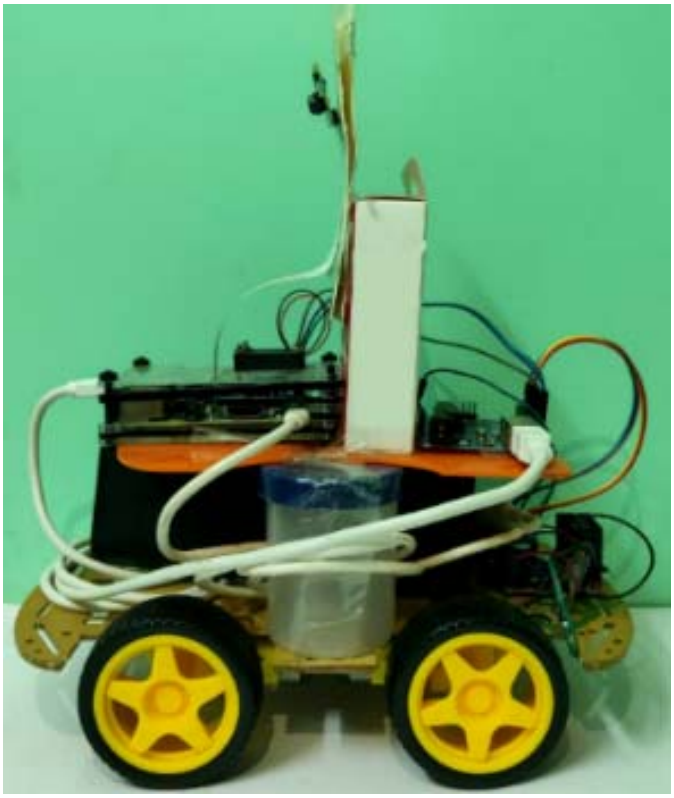


Fig. 6. Prototype Vehicle.

3- little deviated to left lane, 4- extremely deviated to left lane and 5- when there is difficulty to find lane due to low illumination) to Arduino and it chooses the appropriate driving decision.

In speed bump detection, the distance measured using the concept discussed in section E of III is applied here to examine the intelligent behavior of vehicle. Once the distance is measured from speed bump object, Raspberry Pi module immediately sends this information to the Arduino microcontroller. This signal corresponds to the range between lower and higher threshold condition that indicates the safe distance from a vehicle to the speed bump. Arduino module upon receiving the data (signal), modifies its pulse width modulation (PWM) by changing its duty cycle from 100% to 25% for both left and right motors together. Therefore, it reduces the motor/wheel speed before it reaches to the speed bump object and increases its default speed when it cross over the speed bump by adjusting the duty cycle of motor to the normal condition. This task is accomplished for both marked and unmarked speed bump categories and the results are shown in the figure 7 (a-d). Slowing down the motor speed when it detects the speed bump is the early warning signal made by Pi board through the serial communication between Pi and Arduino board. By this mechanism, vehicle's decision making capability is identified figure 8 (a-b). Baud rate at the receiver end is kept to 9600 baud/second. However, the motor speed cannot be immediately reduced to some specific speed due to the inertia of motors. Keeping this fact, we start reducing the vehicles speed with additional distance measurement so that before it reaches to the speed bump, slowing down of motor speed is synchronized.

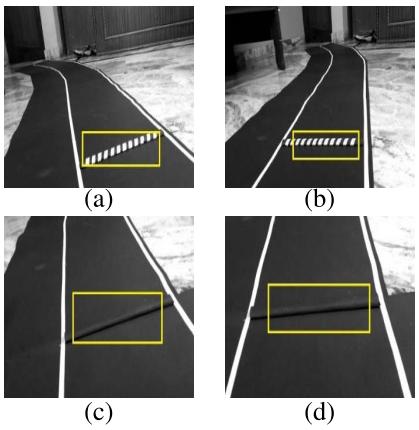



Fig. 7. Speed Bump detection result (a-b) Marked, (c-d) Unmarked.

TABLE V
COMPARISON WITH STATE-OF-ART METHODS

| Method | Accuracy Score |
|---|---|
| Varma et al. [36] | 97.44% |
| W. Devpriya et al. [32] | 90.00% |
| Jose et al. [28] | 97.14% |
| Proposed Model | 98.54% |

## V. EVALUATION OF PROPOSED MODEL IN REAL TIME AND COMPARISON WITH STATE-OF-ART METHODS

The proposed setup has been developed and is similar to real time scenario with the following concerns like: a) the functional prototype modifies its driving decision when it detects speed bump object, b) marked (white and yellow lines) and unmarked categories of speed bumps are considered, and c) the regulation of vehicle's speed is controlled through microcontroller and embedded vision sensors.

Despite of the speed bump road environment, the proposed model achieves higher detection accuracy for marked and unmarked categories of speed bump. Considering the optimal design concepts of proposed model, significant features are learned and validated through standard evaluation measures. From the existing studies, it has been observed that the domain of vision-based speed bump detection for IVS is not much investigated and its intelligent behavior about decision making is little explored. In this direction, Varma *et al.* utilized SSD-Mobilenet and stereo camera is incorporated to process the distance headed for speed bump [36]. Bounding boxes were then fused up for the detected speed bump and reached accuracy of 93.83% for unmarked and 97.44% for marked speed bump. Similarly, standard image processing were used and average accuracy of 85% is attained [32]. Jose *et al.* applied acclerometric approach along with genetic algorithm to discover the characteristics of skewness and driving range of the gyro's x-axis; skewness of the gyro's y-axis; and kurtosis of the accelerometer's y-axis information [28]. The acquired accuracy using mentioned feature composition was reported as 97.14%. Comparing the performance of our proposed model with existing state-of-art methods, we find the proposed model outperforms all the approaches and listed in table V.

From the achieved results, it has been observed that the proposed model is capable of extracting significant features

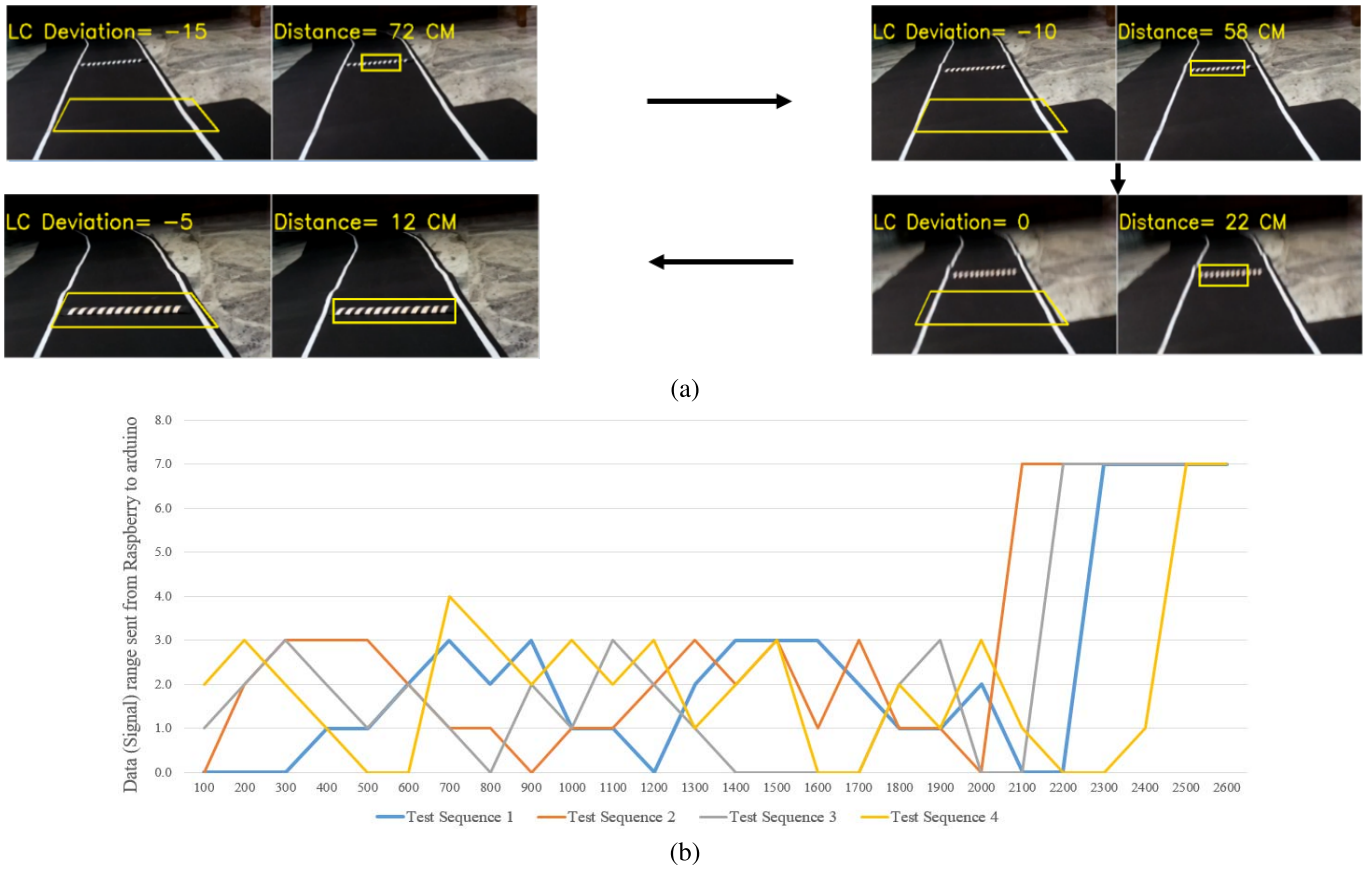



Fig. 8. Run time performance of vehicle prototype (Combination of vision camera, Raspberry Pi and Arduino microcontroller). (a) Lane center calculation is determined to keep vehicle between the lane lines, while calculating the distance to speed bump is carried out to change the driving behaviour accordingly. (b) Result of test sequences and shows range (0.0 to 6.0) for lane keeping mechanism while range (7.0) indicates that distance to speed bump is very close and hence vehicle's speed is needed to be reduced.

from a dynamic road environment and make accurate detection of speed bumps. In addition, our model assist the prototype vehicle to modify its driving behavior according to the detected speed bump which shows the compatibility of model and hardware interface and can be opted in real-time applications.

## VI. Conclusion

In an autonomous driving scenario, a real time embedded system prototype has been proposed, which not only detects the speed bump using vision camera but also utilizes the best of its learning through CNN and applies the intelligence to regulate its speed when such objects occurs. Proposed approach contains two module: first module deals with dataset preparation, training and detection of speed bump; whereas second module shows the operational performance over the detected objects through its logical driving behavior. Test results have revealed promising outcome and that too with optimal power depletion. Quantitative examination also validates the detection accuracy of 98.54%, precision of 99.05% and 97.89% for F -measure. In future direction, capability of prototype is planned to be tested on various cognitive techniques of optimization. Finally, the adopted strategy and implementation is not restricted only to speed bump detection task, but can also be adopted for other intelligent vehicle applications where the detection of various challenging objects are required.

## References

[1] (2019). *Road Traffic Deaths, Global Health Observatory Data Repository by World Health Organization*. Accessed: Jul. 15, 2020. [Online]. Available: http://apps.who.int/gho/data/node.main.A997

[2] J. J. Rolison, S. Regev, S. Moutari, and A. Feeney, "What are the factors that contribute to road accidents? An assessment of law enforcement views, ordinary drivers' opinions, and road accident records," *Accident Anal. Prevention*, vol. 115, pp. 11–24, Jun. 2018, doi: 10.1016/j.aap.2018.02.025.

[3] D.-H. Kim, "Lane detection method with impulse radio ultra-wideband radar and metal lane reflectors," *Sensors*, vol. 20, no. 1, p. 324, Jan. 2020, doi: 10.3390/s20010324.

[4] D. Clarke, D. Andre, and F. Zhang, "Synthetic aperture radar for lane boundary detection in driver assistance systems," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Sep. 2016, pp. 238–243, doi: 10.1109/MFI.2016.7849495.

[5] D. Felguera-Martín, J. T. González-Partida, P. Almorox-González, and M. Burgos-García, "Vehicular traffic surveillance and road lane detection using radar interferometry," *IEEE Trans. Veh. Technol.*, vol. 61, no. 3, pp. 959–970, Mar. 2012, doi: 10.1109/TVT.2012.2186323.

[6] C. Adam, R. Schubert, N. Mattern, and G. Wanielik, "Probabilistic road estimation and lane association using radar detections," in *Proc. 14th Int. Conf. Inf. Fusion*, no. 1, 2011, pp. 937–944.

[7] J. Jung and S. H. Bae, "Real-time road lane detection in urban areas using LiDAR data," *Electronics*, vol. 7, no. 11, pp. 1–14, 2018, doi: 10.3390/electronics7110276.

[8] F. Xu, L. Chen, J. Lou, and M. Ren, "A real-time road detection method based on reorganized LiDAR data," *PLoS ONE*, vol. 14, no. 4, pp. 1–17, 2019, doi: 10.1371/journal.pone.0215159.

[9] Y. Wang and Y. Tsai, "A lane detection method based on 3D-LiDAR," in *Proc. 16th Intell. Transp. Syst. Asia Pacific Forum Fukuoka, Automot. Res. Test. Center, ITS AP Fukuoka*. Fukuoka, Japan: Fukuoka International Congress Center Fukuoka, May 2018, pp. 1–10.

[10] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, "A survey of deep learning applications to autonomous vehicle control," *IEEE Trans. Intell. Transp. Syst.*, early access, Jan. 7, 2020, doi: 10.1109/tits.2019.2962338.

[11] A. Nurhadiyatna and S. Loncaric, "Multistage shallow pyramid parsing for road scene understanding based on semantic segmentation," in *Proc. 11th Int. Symp. Image Signal Process. Anal. (ISPA)*, Sep. 2019, pp. 198–203, doi: 10.1109/ISPA.2019.8868554.

[12] B. Baheti, S. Gajre, and S. Talbar, "Semantic scene understanding in unstructured environment with deep convolutional neural network," in *Proc. IEEE Region Conf. (TENCON)*, Oct. 2019, pp. 790–795, doi: 10.1109/TENCON.2019.8929376.

[13] Y. Peng, W. Han, and Y. Ou, "Semantic segmentation model for road scene based on encoder-decoder structure," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 1927–1932.

[14] L. Vlacic, M. Parent, and F. Harashima, *Intelligent Vehicle Technologies*. Amsterdam, The Netherlands: Elsevier, 2001.

[15] Y. Gu, Q. Wang, and S. Kamijo, "Intelligent driving data recorder in smartphone using deep neural network-based speedometer and scene understanding," *IEEE Sensors J.*, vol. 19, no. 1, pp. 287–296, Jan. 2019, doi: 10.1109/JSEN.2018.2874665.

[16] C. Fan, X. Ye, and W. Gu, "KRUS: A knowledge-based road scene understanding system," in *Proc. 14th Int. Conf. Pattern Recognit.*, 2002, pp. 731–733, doi: 10.1109/icpr.1998.711249.

[17] S. K. Venkateshkumar, M. Sridhar, and P. Ott, "Latent hierarchical part based models for road scene understanding," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 115–123, doi: 10.1109/ICCVW.2015.25.

[18] W.-J. Won, T. H. Kim, and S. Kwon, "Multi-task deep learning design and training tool for unified visual driving scene understanding," in *Proc. 19th Int. Conf. Control, Automat. Syst. (ICCAS)*, Oct. 2019, pp. 356–360.

[19] Y. Sun, H. Lu, and Z. Zhang, "RvGIST: A holistic road feature for real-time road-scene understanding," in *Proc. 14th ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput.*, Jul. 2013, pp. 649–655, doi: 10.1109/SNPD.2013.86.

[20] A.-V. Vo, L. Truong-Hong, and D. F. Laefer, "Aerial laser scanning and imagery data fusion for road detection in city scale," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 4177–4180, doi: 10.1109/IGARSS.2015.7326746.

[21] J. Byun, B.-S. Seo, and J. Lee, "Toward accurate road detection in challenging environments using 3D point clouds," *ETRI J.*, vol. 37, no. 3, pp. 606–616, Jun. 2015, doi: 10.4218/etrij.15.0113.1131.

[22] S. Melo *et al.*, "24 GHz interferometric radar for road hump detections in front of a vehicle," in *Proc. Int. Radar Symp.*, Jun. 2018, pp. 1–9, doi: 10.23919/IRS.2018.8448029.

[23] D. Craciun, J.-E. Deschaud, and F. Goulette, "Automatic ground surface reconstruction from mobile laser systems for driving simulation engines," *Simulation*, vol. 93, no. 3, pp. 201–211, Mar. 2017, doi: 10.1177/0037549716683022.

[24] M. Jain, A. Singh, S. Bali, and S. Kaul, "Speed-breaker early warning system," in *Proc. 6th USENIX/ACM Workshop Netw. Syst. Dev. Reg.*, 2012, pp. 1–6.

[25] P. V. P. Tonde, A. Jadhav, S. Shinde, A. Dhoka, and S. Bablade, "Road quality and ghats complexity analysis using Android sensors," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 3, pp. 101–104, Mar. 2015, doi: 10.17148/ijarcce.2015.4324.

[26] D. V. Mahajan and T. Dange, "Estimation of road roughness condition by using sensors in smartphones," *Int. J. Comput. Eng. Technol.*, vol. 6, no. 7, pp. 41–49, Jul. 2015.

[27] M. R. Carlos, M. E. Aragon, L. C. Gonzalez, H. J. Escalante, and F. Martinez, "Evaluation of detection approaches for road anomalies based on accelerometer readings—Addressing who's who," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 10, pp. 3334–3343, Oct. 2018, doi: 10.1109/TITS.2017.2773084.

[28] J. M. Celaya-Padilla *et al.*, "Speed bump detection using accelerometric features: A genetic algorithm approach," *Sensors*, vol. 18, no. 2, pp. 1–13, 2018, doi: 10.3390/s18020443.

[29] L. C. Gonzalez, R. Moreno, H. J. Escalante, F. Martinez, and M. R. Carlos, "Learning roadway surface disruption patterns using the bag of words representation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 2916–2928, Nov. 2017, doi: 10.1109/TITS.2017.2662483.

[30] Y. A. Daraghmi and M. Daadoo, "Intelligent smartphone based system for detecting speed bumps and reducing car speed," in *Proc. MATEC Web Conf.*, vol. 77, 2016, p. 3, doi: 10.1051/matecconf/20167709006.

[31] F. Seraj, B. J. Van Der Zwaag, A. Dilo, T. Luarasi, and P. Havinga, "RoADS: A road pavement monitoring system for anomaly detection using smart phones," in *Proc. Int. Workshop Mach. Learn. Urban Sensor Data*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 9546, 2016, pp. 128–146, doi: 10.1007/978-3-319-29009-6_7.

[32] W. Devapriya, C. N. K. Babu, and T. Srihari, "Real time speed bump detection using Gaussian filtering and connected component approach," *Circuits Syst.*, vol. 7, no. 9, pp. 2168–2175, 2016, doi: 10.4236/cs.2016.79188.

[33] H.-T. Chen, C.-Y. Lai, and C.-A. Shih, "Toward community sensing of road anomalies using monocular vision," *IEEE Sensors J.*, vol. 16, no. 8, pp. 2380–2388, Apr. 2016, doi: 10.1109/JSEN.2016.2517194.

[34] J.-K. Lee and K.-J. Yoon, "Temporally consistent road surface profile estimation using stereo vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1618–1628, May 2018, doi: 10.1109/TITS.2018.2794342.

[35] H.-T. Chen, C.-Y. Lai, C.-C. Hsu, S.-Y. Lee, B.-S.-P. Lin, and C.-P. Ho, "Vision-based road bump detection using a front-mounted car camcorder," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 4537–4542, doi: 10.1109/ICPR.2014.776.

[36] V. S. K. P. Varma, S. Adarsh, K. I. Ramachandran, and B. B. Nair, "Real time detection of speed hump/bump and distance estimation with deep learning using GPU and ZED stereo camera," *Procedia Comput. Sci.*, vol. 143, pp. 988–997, Jan. 2018, doi: 10.1016/j.procs.2018.10.335.

[37] S. Shah and C. Deshmukh, "Pothole and bump detection using convolution neural networks," in *Proc. IEEE Transp. Electrific. Conf. (ITEC-India)*, Dec. 2019, pp. 1–4, doi: 10.1109/itec-india48457.2019.itecindia2019-186.

[38] R. R. Nair, E. David, and S. Rajagopal, "A robust anisotropic diffusion filter with low arithmetic complexity for images," *EURASIP J. Image Video Process.*, vol. 2019, no. 1, Dec. 2019, Art. no. 48, doi: 10.1186/s13640-019-0444-5.

[39] K. Naveed, B. Shaukat, S. Ehsan, K. D. Mcdonald-Maier, and N. U. Rehman, "Multiscale image denoising using goodness-of-fit test based on EDF statistics," *PLoS ONE*, vol. 14, no. 5, pp. 1–25, 2019, doi: 10.1371/journal.pone.0216197.

[40] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, pp. 1–14, *arXiv:1609.04747*. [Online]. Available: http://arxiv.org/abs/1609.04747

[41] L. Huang, T. Zhe, J. Wu, Q. Wu, C. Pei, and D. Chen, "Robust inter-vehicle distance estimation method based on monocular vision," *IEEE Access*, vol. 7, pp. 46059–46070, 2019, doi: 10.1109/ACCESS.2019.2907984.

[42] J. Danial, D. Feldman, and A. Hutterer, "Position estimation of moving objects: Practical provable approximation," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1985–1992, Apr. 2019, doi: 10.1109/LRA.2019.2899430.

**Deepak Kumar Dewangan** received the B.E. degree in computer science and engineering from Pt. R. S. U., Raipur, India, in 2005, and the M.Tech. degree in computer science and engineering from CSVTU Bhilai, Bhilai, India, in 2012. He is currently pursuing the Ph.D. degree in information technology with the National Institute of Technology Raipur, Raipur. He has been associated with various institutions and have more than ten years of experience in academics. His research interests include development of artificial intelligence-based real-time applications, computer vision, hardware software interface, and robotics.

**Satya Prakash Sahu** (Member, IEEE) received the B.E. and M.Tech. degrees in computer science and engineering from Rajiv Gandhi Technical University, Bhopal, India, and the Ph.D. degree in information technology from the National Institute of Technology Raipur (NIT Raipur), India. He is an Assistant Professor with the Department of Information Technology, NIT Raipur. His research interests include artificial intelligence, machine learning, image processing, medical imaging, and soft computing. He has authored more than 30 research papers in national and international conferences and journals.