International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Enabling Safe Driving through AI - Based Speedbump and Pothole Detection for Indian Roads

Shreyas Sridhar

Abstract: ***The objective of this study is to use Computer Vision and Artificial Intelligence techniques for detecting Potholes and Speed bumps on Indian roads to improve driver safety Supervised machine learning AI technique was utilized by using the Convolutional Neural Network (CNN) to create an image detection model which divided the Indian road images into 3 categories – normal, Speed Bump and Pothole. Our code was able to accurately classify the images into one of the three categories. We trained a Convolutional Neural Network using over 800 unique Indian road images on Speed bumps and Potholes. We got an accuracy of 91% using this model, indicating that a reasonable sized CNN model can very accurately detect these Indian road conditions. We compare the complexity of this model versus a model that is not trained with India road images, to showcase the added complexity of building a model for Indian conditions.***

Keywords: Safe Driving, Safety, Pothole, Artificial Intelligence

1. Introduction

Two of the most common road threats that reduce driver safety in India are potholes and unmarked speed bumps. Safety risk is especially severe for drivers of two wheelers like bikes, scooters and bicycles. The total number of road accidents in India in 2015 was 1, 48, 707. Of that 11, 084 were deaths due to speedbumps. In 2014, the fatality was marginally lower with 11, 008deaths. On an average speed bumps cause around 10000 deaths a year, making India rank one of the highest in road accidents. Many of the speedbumps are unmarked and/or poorly designed making it very difficult for the drivers to spot these ahead of time and slow down as needed. A similar road hazard which takes a lot of lives and a constant source of frustration, especially for two - wheeler drivers, is potholes. These come in various shapes and sizes and are sometimes hard to detect for the naked eye when traveling at high speeds causing over 9000 deaths and 23000 injuries in the past 3 years.

Overall accidents caused due to potholes and speed bumps result in considerable economic losses to individuals, their families and to the nation as a whole.

This paper aims to explore an AI - based approach that can be used to detect speed bumps and potholes, specific to Indian road conditions, to prevent road accidents and save many lives. Several speed bump detection techniques exist that are Cloud based [2] or sensor based [3] or even AI techniques based [4]. Cloud based techniques are not viable for Indian conditions where cloud connectivity might not always be available or for economic reasons because the solution ought to be cheap. Sensor based techniques have the drawbacks as listed in [4]. AI based techniques for Speed Bump detection that exist today are not reliable for Indian road conditions because they are not trained with Indian road specific data. This research endeavors to fix this gap and highlight the deltas in the model between those build for Indian road conditions vs. rest.

2. Approach

Main goal of research was to create a CNN model that could accurately identify road obstacles like speed bumps and potholes tailored to Indian road conditions.

First, we need to create a model with high accuracy for inference. An image classification model was created using a convolutional neural network and it was trained with 1000 images across normal, Speed Bump and Potholed Indian roads. These images were collected between images on Kaggle. com, photos taken in - person on the roads of Bengaluru city and general Google search., Second, we provided a set of test images where the model would infer whether it was a normal road, or a road with a speed bump or road with a pothole.

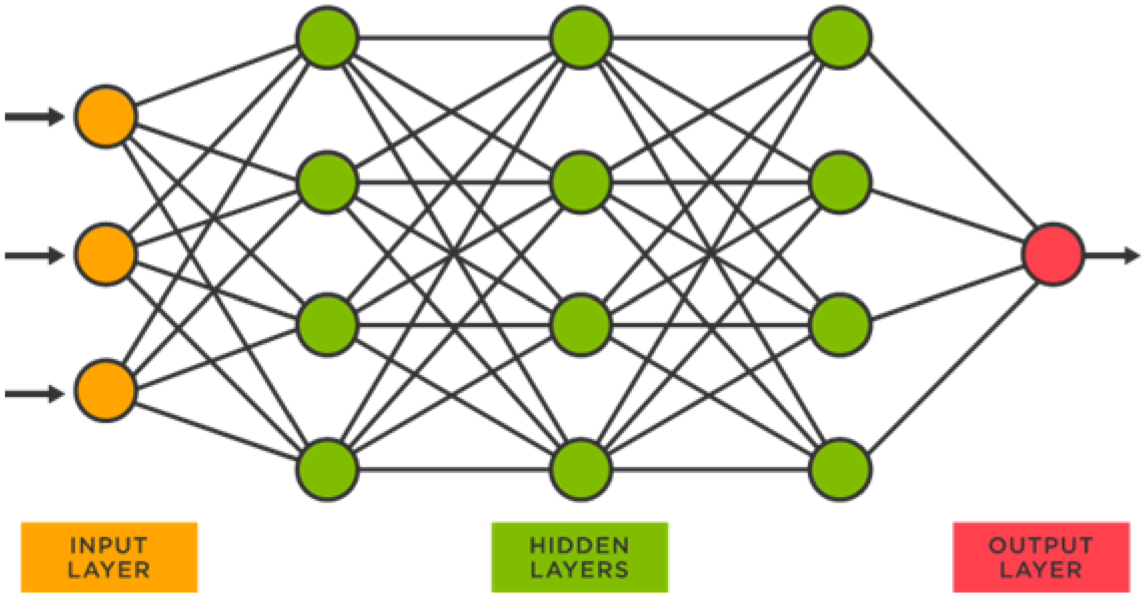
3. Concepts used

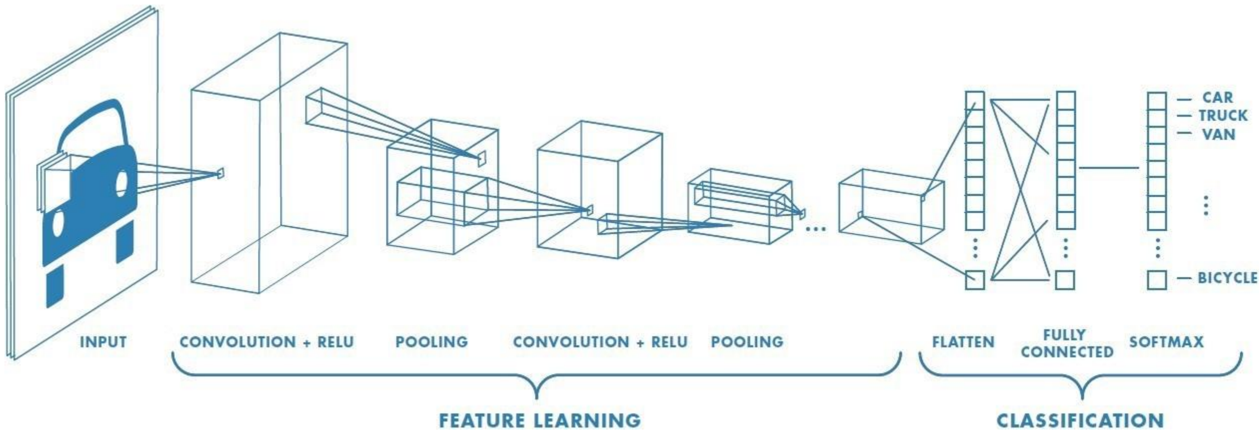
1) Supervised learning

Supervised learning is a machine learning trick in which a model is trained by providing labeled data. The model is trained using training datasets until it can distinguish between the data in different classes. Once the model has been trained, it can be used to classify and label unlabeled data which are known as test sets.

2) CNN (convolutional neural networks)

Convolutional neural network is a type of classification technique. The input image given to CNN is called the Input layer, which is typically an image of size NxN. Machine automatically infers relationships from the given image and this is achieved through what is called as hidden layers in the CNN mode. These hidden layers extract interesting "features" from the input image which it can then use to infer the overall image, in gross terms. The operation of the hidden layers replicates the function of a human brain with each layer behaving like a neuron in the human brain. Because of this functionality, hidden layers are thus called the Feature learning layer (during training) or Feature extraction layer (during actual inference on a real - life image). Each hidden layer in itself is two sub layers – the first layer performing the function of convolution where the input image matrix is "convoluted" into a smaller matrix with a "kernel" matrix (which represent a particular feature) that represents one of the features that need to be extracted from the input image. To 2nd sub layer is the "pooling" layer which either uses the max or average of a group of matrix values, to reduce the size of the overall matrix that need to be processed further. This is required so that the computational requirements for image reduction are reduce. After a certain number of iterations of Convolution + Pooling, the resultant matrix is flattened out (using a matrix operation) and a certain number of dense layers added. Eventually another math function like SoftMax is applied over the output from the previous layer to get the final image classification. Classification is mathematically represented as a value from 0 to 1 for each given feature. These are pictorially shown below:





Model creation, training and testing

The data set was created by me and all the 1000 images were either found on Google, Kaggle. com or were personally taken on the roads of Bengaluru. I took care to take images of a variety of unmarked speed bumps on the roads of Bengaluru, along with good ones. Overall image set contains images of speed bumps, potholes and normal roads.

As mentioned earlier, CNN and the concept of supervised machine learning was utilized. We created two different models, and the training data was split into 3 classes which consists of labeled images of a particular situation. The first models labeled classes included: 'pothole' and 'normal road'.

And the second programs labeled classes were: 'Speed Bumps', and 'normal road'.

Images used were of dimension TBDxTBD.

The models consisted of a 3 layer convolutional neural network which outputs 3d feature maps. Our first two layers are conv2d layers, these are convolutional layers that will deal with our input images, which are seen as two dimensional matrices.

32 and 32 are the number of nodes in each layer. After this the flattening function is used which converts the 3d feature maps into 1d feature vectors. Finally a Dense layer of 128 units was added and passed through the sigmoid activation function.

4. Results and Further Research

We were able to get a very high accuracy from our databases. From our speedbump database we were able to identify the obstacle with an accuracy of 94.378% on our testing database, a final accuracy of around 94.38% was observed from our training database and an accuracy of 88.71% was observed from our validation database. Whereas our pothole database was able to identify the

Paper ID: SR22522233403 DOI: 10.21275/SR22522233403

obstacle with an accuracy of 93.782% on our testing database, a final accuracy of 93.298% was observed from our training database and an accuracy of 86.51% was observed from our validation database.

Here are a few images which have been accurately identified and classified.

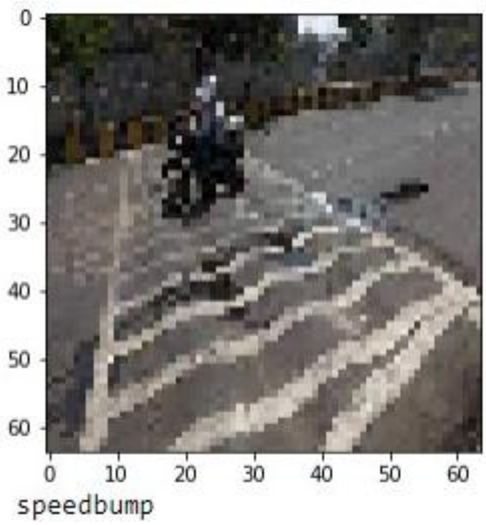


Figure 4: Speedbump on an Indian road

Figure 5: A regular Indian road (marketplace)

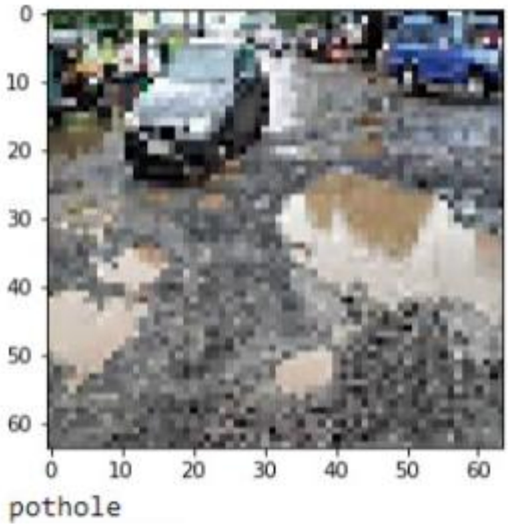


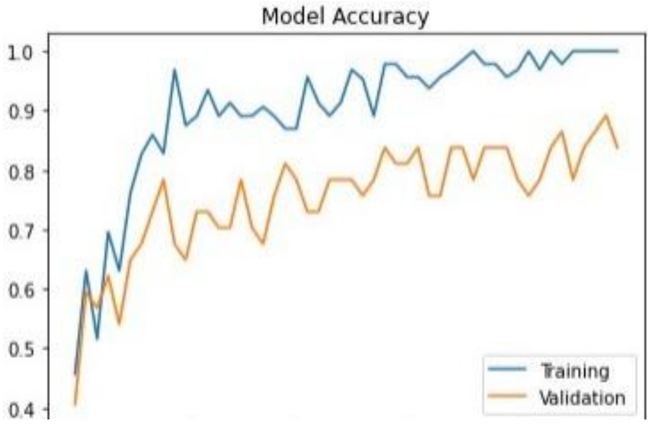
Figure 6: Indian road with many potholes

We did more research by comparing the model complexity required for Indian roads vs. western roads. During the investigation we figured out that the number of layers required to accurately detect speed bumps and potholes on Indian roads is significantly higher.

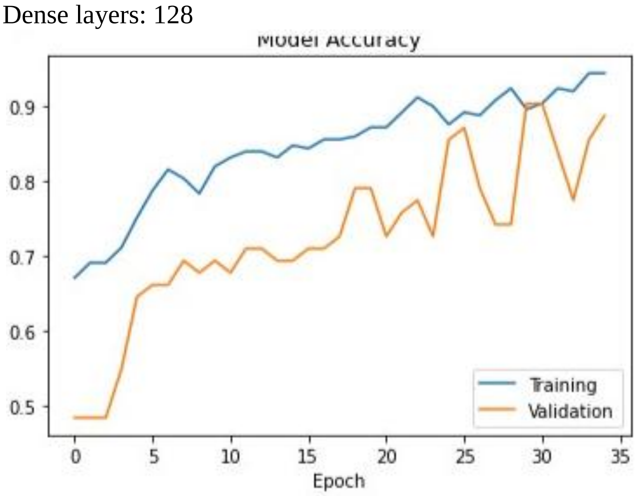
For this we first ran the model with just western road related pictures (both speed bump and normal) two different values of dense layers– 128 and 30. Both of them show 90+% accuracy in training and 80+% accuracy in validation, indicating that the additional dense layers provide very little accuracy improvement.

Results are shown below

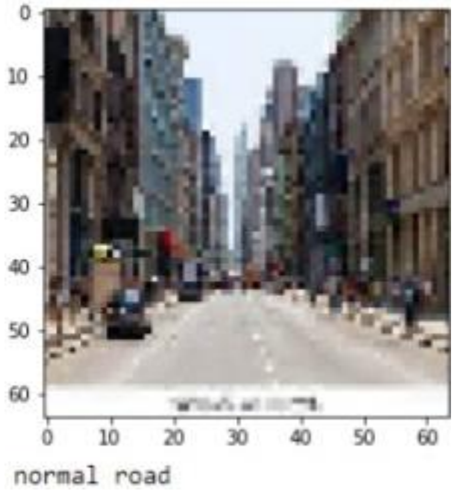
Dense layers: 30

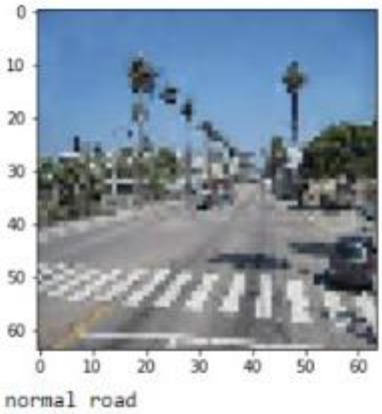


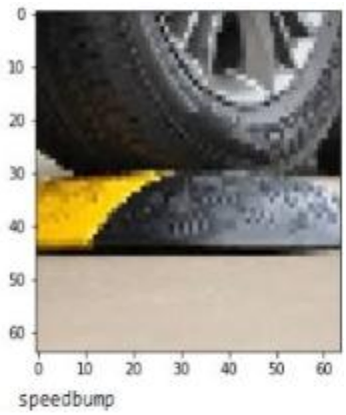
Dense layers: 128



Below are a few results from American roads with actual pictures:







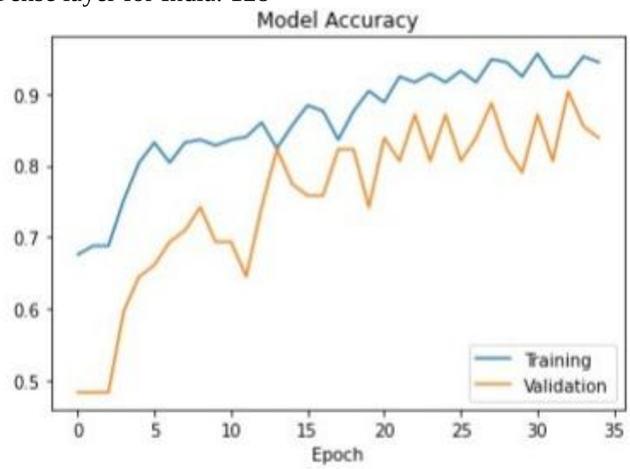
Secondly, we created two programs for Indian roads (both normal and speed bumps), with labeled images of speed bumps and normal streets – one with a dense layer of 128 and another with 30. For Indian roads the accuracy varied tremendously for dense layer count of 30 the average was below 70%.

This does point to the fact that the number of details of Speedbumps being more complicated in India, requires the model to be more complicated. So models created for western conditions cannot be directly applied to Indian roads. But even a dense layer of 128 is implementable reasonably in modern IC's.

We reached a conclusion based on the results, that the number of neural layers does not necessarily mean that the accuracy would be more, adding more layers helps you extract more features but as discussed above the number of features on American roads are comparatively less to Indian roads, so the number of neural layers does not completely matter.

Below figures show the accuracy.

Dense layer for India: 128



Dense layers: 30

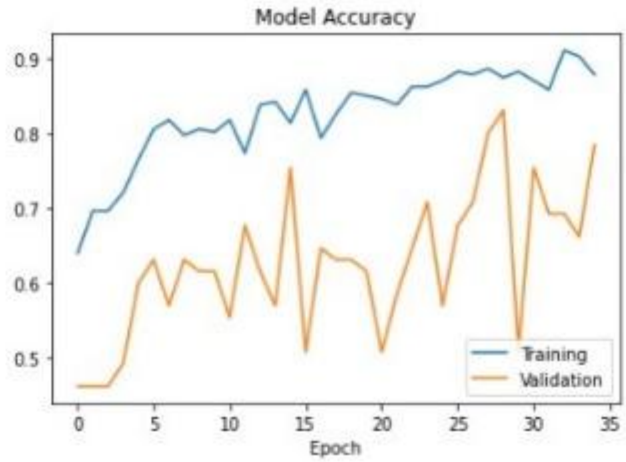
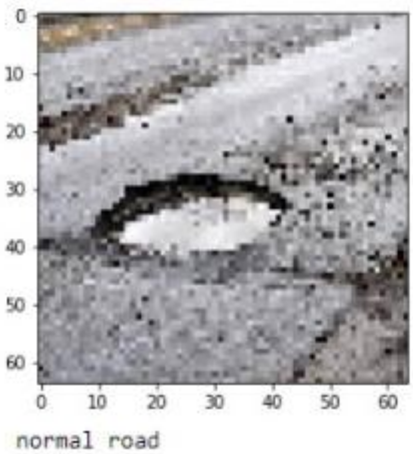


Figure 7: shows a pothole as a normal road when the number of layers are reduced



5. Further Exploration

From the results above it is evident that an AI based model that is efficient with a high degree of accuracy can be built for Indian road conditions. This model can be tested with drivers in India, mainly two - wheeler drivers as they are at most risk. With the rise of online rides like Ola, Uber and other transportation apps, risk of road accidents in India have increased because. Ola and Uber drivers generally drive around half the day or more which could lead them to get tired and hence leads to not paying as much attention to the roads. Therefore our detection system could help prevent accidents in this critical business sector.

It is also known that this model is solely trained for the daytime as all the images in the database are taken during the daytime. However, many accidents occur during the night time where the visibility is limited, to prevent this issue a different model can be trained to identify these said obstacles by creating and using a dataset with only thermal images from the night, and it can provide accurate results.

References

[1] Sumit Saha (2018), *A Comprehensive Guide to Convolutional Neural Networks — the ELI5* https: //towardsdatascience. com/a - comprehensive - guide - to - convolutional - neural - networks - the - e li5 - way - 3bd2b1164a53way
[2] Om Shankar Kiradoo, Kavish Vyas (2019), *Improper speed humps: Causing casualties, revealing* arbitrariness. https: //counterview. org/2019/12/07/improper - speed - humps - causing - casualties - reve aling - arbitrariness/
[3] Devjyot Ghoshal (2017), *Speed breakers on Indian roads kill more than 10, 000 people every year*. https: //qz. com/india/908870/speed - breakers - on - indian - roads - kill - more - than - 10000 - people – every year/
[4] Deepak Kumar Dewangan, Satya Prakash Sahu. *Deep Learning - Based Speed Bump Detection Model for Intelligent Vehicle System Using Raspberry pi.* https: //ieeexplore. ieee. org/document/9206567
[5] Jitendra Sharma (2021). *Building a Convolutional Neural Network Using TensorFlow – Keras* https: //www.analyticsvidhya. com/blog/2021/06/building - a - convolutional - neural - network - using - tensorflow - keras/