Vision Based Road Hump and Speed Breaker Detection

Prof. Varunakshi Bhojane[1], Romali Surve[2], Krunal Rane[3]

[1] *Professor, Dept of Computer Engineering, Pillai College of Engineering, New Panvel, Maharashtra, India*
[2,3]*B.E. Computer Engineering Student, Pillai College of Engineering, New Panvel, Maharashtra, India*

---***---

Abstract - *Speed Breaker detection and tracking are widely applied to intelligent image and video surveillance. The goal is to detect Speed Breaker in a scene, localize them in the image and, if possible, infer their exact articulations. This research addresses the deep learning–based techniques. By using the You Only Look Once version 2 (YOLOv2) detector and deep convolutional neural network (CNN) we detect the objects obstructing while driving the vehicle. Poor road conditions like cracks, potholes, open sewer caps, and extreme road conditions can cause inconvenience to passengers, damage to vehicles, and accidents. Detecting those obstacles has become relevant due to the rise of the autonomous vehicle. Although previous studies used various sensors and applied different image processing techniques, performance is still significantly lacking, especially when compared to the tremendous leaps in performance with computer vision and deep learning. Hence, the object detection technique (YOLO) will be used. We do not make a selection or technique best method and optimal because the best technique depends on the needs, concerns and existing environment. The goal of object detection is to detect all instances of objects from a known class, such as people or faces in an image. Typically, only a small number of instances of the object are present in the image, but there is a very large number of possible locations and scales at which they can occur.*

Key Words: Speed Breaker detection, Computer vision, Machine Learning, Deep Convolutional, Neural Network

1. INTRODUCTION

The quality of road is crucial to people who drive. In some areas, drivers need to be cautious because damaged speed breaker and road humps have been proven to cause catastrophes, especially during the rainy season. Detecting humps would allow vehicles to issue warnings so drivers can slow down and avoid them (or the vehicle itself can adjust settings to avoid them), minimize the impact, and make the ride smooth.

Speed breaker detection is about sensing the road ahead of an autonomous vehicle. Nonetheless, studies and research on road-surface damage are still few. Several of them use traditional methods, with sensors and expensive equipment to label images in a classification task, but not to detect dam- age coordinates. Recently, object detection using end-to-end deep learning has been reported to outperform traditional methods. Costly sensors, battery life, computation power, and the complexity of data integration have been reduced by simply relying on imagery input to detect objects. In this study, we train and evaluate object detection with You Only Look Once version 2 (YOLOv2) that has a state-of-the-art convolutional neural network (CNN) at its core.[1]

Literature survey describes various techniques used in the work. Identifying the current literature in related domain problem and identifying the techniques that have been developed and present the various advantages and limitation of these methods used extensively in literature.

A) Detection of Potholes Using a Deep Convolutional Neural Network - Although previous studies used various sensors and applied different image processing techniques, performance is still significantly lacking, especially when compared to the tremendous leaps in performance with computer vision and deep learning. This research addresses this issue with deep learning–based techniques. We applied the You Only Look Once version 2 (YOLOv2) detector and pro- pose a deep convolutional neural network (CNN) based on YOLOv2 with a different architecture. Despite a limited amount of learning data and the challenging nature of pothole images, our proposed architecture is able to obtain a significant increase in performance over YOLOv2 (from 60.14 to 82.43 average precision).[1]

B) Real Time Detection of Speed Hump/Bump and Distance Estimation with Deep Learning using GPU and ZED Stereo Camera - Most of the humps in India are n o t being constructed and maintained according to the public safety guidelines of Indian Road Congress (IRC) i.e., IRC099, which is resulting in damage to the vehicles, severe discomfort to the driver and even causing loss of direction control which is leading to fatalities. Very few methods were discussed in literature for un-marked speed hump/bump detection. We propose a method that detects and informs the driver about the upcoming un-marked and marked speed hump/bump in real time using deep learning techniques and gives the distance the vehicle is away from it using stereo- vision approaches.[2]

2. PROPOSED WORK

Controlling the computer mouse using the eyes movement requires a fast and effective algorithm, that's brought us to decrease the running time of the tool to the minimum by di- viding the operation into few steps and using a tracking algorithm in order to avoid unnecessary calculations

a System Architecture

The system architecture is given in Figure 1. Each block is described in this Section.

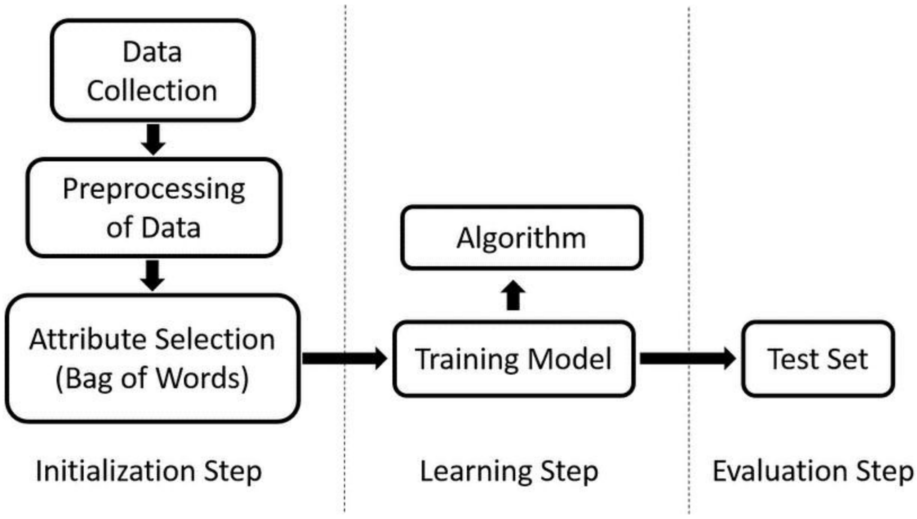

Fig. 1: Proposed System Architecture.

b Initialization Description

Image acquisition: Different types of image acquisition devices such as digital camera, CCD line-scan camera, area scan camera and scanner are used for capturing the fabric images.

Image labeling: Labeled data is a group of samples that have been tagged with one or more labels. After obtaining a labeled dataset, machine learning models can be applied to the data so that the new unlabeled data can be presented to the model and a likely label can be guessed or predicted for that piece of unlabeled data.

Installing packages:

A] Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large scale data processing) that aims to simplify package management and deployment.

B] Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK or Theano. It was developed with a focus on enabling fast experimentation.

C] Training: Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then, the training, or learning begins.

There are two approaches to training - supervised and un- supervised. **Supervised** training involves a mechanism of providing the network with the desired output either by manually "grading" the network's performance or by providing the desired outputs with the inputs. **Unsupervised** training is where the network has to make sense of the inputs without outside help.[1]

3. RESULTS AND ANALYSIS

The implementation detail is given in this section.

a YOLO Object Detection

YOLO takes an approach different from other networks that use a region proposal or a sliding window; instead, it reframes object detection as a single regression problem. YOLO looks at the input image just once, and divides it into a grid of S x S cells. Each grid cell predicts B bounding boxes, a confidence score representing the intersection over union (IOU) with the ground truth bounding box, and the probability that the predicted bounding box contains some objects:

$$Confidence = Pr(object) * IOUTruthPred$$

IOUTruthPred denotes intersection over union between the predicted box and ground truth. Each cell also predicts C conditional class probabilities, Pr(object). Both confidence score and class prediction will output one final score telling us the probability that this bounding box contains a specific type of object.

b YOLOv2 Architecture

There were two different model architectures used in con- ducting our training. The first model is based on the Darknet architecture of YOLOv2. It contains 31 layers in which 23 are convolutional layers with a batch normalization layer be- fore leaky rectified linear unit (ReLu) activation, and a max- pool layer at the 1st, 3rd, 7th, 11th, and 17th layers. In order to train our own dataset, we needed to reinitialize the final convolutional layer so it outputs a tensor with a, 13 x 13 x 30 shapewhere, 30 = 5 bounding boxes x (4 coordinates + 1 confidence value + 1 class probability)

c Anchor Box Model

In object detection techniques, normally each of the grid cells can detect only one object. The problem arises when there is more than one object in each cell, so we can handle this situation using the idea of anchor boxes. Instead of predicting a one-dimensional 5 + num of class, it instead predicts (5 + num of class) 1 Route to layer 16th 2 Route to layer 27th and 24th num of anchor boxes. Each anchor box is designed for detecting objects of different aspect ratios and sizes.

TABLE 1: Five Anchor Box Sets Generated By K-Means Clustering On A Pothole Training Set

Anchor Box Set	**Width**	**Height**
Set 1	1.834849	0.697362
Set 2	3.690766	2.024326
Set 3	6.013811	6.493899
Set 4	7.706391	3.544638
Set 5	10.78841	8.534208

d How YOLO works

You can take a classifier like VGGNet or Inception and turn it into an object detector by sliding a small window across the image. Using a sliding window gives several hundred or thousand predictions for that image, but you only keep the ones the classifier is the most certain about.

This approach works but it's obviously going to be very slow, since you need to run the classifier many times.

YOLO takes a completely different approach. It's not a traditional classifier that is repurposed to be an object detector. YOLO actually looks at the image just once (hence its name: You Only Look Once) but in a clever way.

YOLO divides up the image into a grid of 13 by 13 cells:

Fig. 2: Output Image of Speed Breaker.

4. CONCLUSIONS

The study of different domain techniques is presented. The different techniques such as Image acquisition, Image La- belling, Installing Packages and Training of YOLO is explained. The comparative study of various techniques used in different papers is presented in this report. The hybrid approach is proposed with some modification in the classifiers used in previous papers in order to get more efficient and accurate result. Some extra features are to extracted in the paper and the system also provides information about the type of Road Hump and Speed Breaker .The applications of this domain is also identified and presented.

ACKNOWLEDGEMENT

We would like to express our specials thanks and gratitude to Principal Dr. Sandeep Joshi, H.O.D. Prof. Sharvari Gov ilkar, Project Guide Prof. Varunaskshi Bhojane and Project Coordinator, Prof. Rupali Nikhare. Who gave us the opportunity to do this project as a part of Project in the topic "Vision Based Road Hump and Speed Breaker Detetction" Which helped us in applying the knowledge that we have ac- quired during the semester and learn new concepts.

REFERENCES

[1] https://machinethink.net/blog/object-detection-with- yolo/

[2] Gunjan Chugh, Divya Bansal and Sanjeev Sofat, **"Road Condition Detection Using Smartphone Sensor: A Survey"**, International Journal of Electronic and Elec- trical Engineering, Vol. 7, No. 6, 2014.

[3] Ajit Danti, Jyoti Y. Kulkarni, and P. S. Hiremath, **"An Image Processing Approach to Detect Lanes, Pot- holes and recognize road Signs in Indian Roads"**, International Journal of Modeling and Optimization, Vol. 2, No. 6, December 2012.

[4] Abdullah Alfarrarjeh, Dweep Trivedi, Seon Ho Kim, Cyrus Shahabi **"A Deep Learning Approach for Road Damage Detection from Smartphone Images"**, Integrated Media Systems Center, University of Southern California, Los Angeles, CA 90089, USA, 2018.

[5] V S K P Varmaa, Adarsh Sa, K I Ramachandranb, Binoy B Naira **"Real Time Detection of Speed Hump/Bump and Distance Estimation with Deep Learning using GPU and ZED Stereo Camera"** , 8th International Conference on Advances in Computing and Communication (ICACC-2018) , pp. 988 -997, 2018.