Image Based Street Pothole Detection

Subhanjani Mallampati, Sweety Sojrani, Divya Sidhabathuni, Archana Yadawa
San Jose State University
1 Washington Sq, San Jose, CA, 95192, USA
archana.yadawa@sjsu.edu
divya.sidhabathuni@sjsu.edu
sweety.sojrani@sjsu.edu
subhanjani.mallampati@sjsu.edu

Abstract

The proposed solution aims to use machine learning and artificial intelligence algorithms to create an accurate and efficient pothole detection system. Having the ability to detect for potholes on the roads will reduce the country's annual spending cost to repair these potholes. Existing solutions are not as effective nor efficient at detecting potholes on roads. Both machine and deep learning methodologies and algorithms have been incorporated to create an effective and accurate pothole detection system.

1. Introduction

Potholes wreak havoc on vehicles on a daily basis. In America, it costs $3 billion annually to repair damage caused by potholes. The force of impact a car's tires has while exiting a pothole can cause severe damage. Currently, there are a few systems available that provide pothole detection. For example, some Land Rover vehicles have a built-in pothole detection system. This connected car technology allows a vehicle to collect data about the location and severity of the potholes. It can then send warnings to the driver indicating to slow down or be more cautious. Another example is the all-new Focus pothole detection system. This system adjusts the vehicle's suspension responses to provide the smoothest ride quality.

The proposed solution aims to use machine learning and artificial intelligence algorithms to create an accurate and efficient pothole detection system. Having the ability to detect for potholes on the roads will reduce the country's annual spending cost to repair these potholes. This report will explain our proposed solution in further detail. First, related work will be discussed using Google Scholar and ResearchGate papers. Next, training and testing data will be presented. Then, machine learning and AI algorithms will be explained. Lastly, references will be credited.

1.1 Related Work

In Danti et al. (2012), a system incorporating image processing algorithms was used as the main detection and classification mechanism. The study aimed to detect a variety of road related objects such as lanes, road signs and potholes. The paper states that a pothole is normally characterized by a distinctive black color in the road, and that this can be the predominant characteristic used to detect a pothole. To successfully segment the pothole region, a black and white threshold was applied to the image that would highlight the pothole area. The method does not rely on the use of any machine learning techniques to detect a pothole. The conclusion of the study indicated that the algorithm did not deliver desirable results and segmented many undesirable areas within an image and that a more effective filtering method would need to be used to improve the accuracy of the approach

Another method for the detection of potholes proposed segmenting a road based on its defect and non-defect regions (Koch & Brilakis, 2011a). Similar to the previous approaches, it utilized image thresholding and accomplished this by using a histogram shape-based thresholding algorithm. A pothole is detected in a defect region by the use of morphological thinning and elliptic regression. Morphological thinning is an algorithm that is used on binary images to separate foreground pixels from the background

pixels (Kaehler & Bradski, 2008). The paper assumes that a pothole is elliptically shaped and therefore implements an elliptic regression algorithm which aims to determine how well the collection of data points (pixels) fit an elliptical shape. All of the images processed by the algorithm were obtained by placing a high-speed camera on a robot. The placement of the camera simulates being placed on the back of a moving vehicle with the camera mounted in such a manner that it would be tilted towards the road surface. Images were processed in two stages. The first stage operated in real-time and identified frames with defect regions. The second stage would pass only the frames that had defects to the more complex pothole detection algorithm. The system parameters were obtained by using a training set of 50 pothole images. It was deployed on 70 images and was found to have a precision of 81.6%. The images that were used in the paper were mostly acquired via Google Images. These images were taken by different people with different cameras at a variety of angles to the road surface. Also, some of the photos were taken up close to the potholes whilst others were not. Once a pothole is detected, it can also be tracked so that the same pothole in subsequent frames is only counted as a single pothole occurrence. This is useful when it's necessary to accurately determine the number of potholes detected. Koch & Brilakis (2011a) continued with further research that led to the work in Koch & Brilakis (2011b) where a method for tracking the potholes between frames was discussed. This paper disregards this aspect as the focus is warning a driver of potholes, so that it is only necessary to classify whether a section of road has potholes or not.

2. Project Team and Roles

We are team 10 and our team consists of 4 members. Each member's contribution has been listed below.

Member 1: Subhanjani Mallampati

- Labeled both train and test images
- Trained images using CNN
- Participated in all reports and presentations

Member 2: Sweety Sojrani

- Labeled both train and test images
- Trained images using SSD
- Participated in all reports and presentations

Member 3: Divya Sidhabathuni

- Labeled both train and test images
- Trained images using YOLO
- Participated in all reports and presentations

Member 4: Archana Yadawa

- Labeled both train and test images
- Trained images using Faster R-CNN
- Participated in all reports and presentations

3. Data Preparation

We have labeled the data using the labelImg application. There are a total of 2500 images. Out of the dataset of 2500 images, 80 percent of it is the training data while the remaining 20 percent will be used for test data. 2500 images have been split into two categories positive and negative. There are an equal number of positive (with potholes) and negative (without potholes) images in order for the training data to have an unbiased binary classification of data. These 2000 images will be used to train our model. The remaining 500 have been used for testing our model. The images are from the point of view of a dashboard camera in a car. Mounted on the windscreen of a driving car, this view accurately reflects the commercial applications and the San Jose smart city initiative. To label positive images, we have set a common schema and labeled potholes. For negative images we selected the entire image and labeled it as no pothole.

Training Example:

Figure 3.1 Training Image with Labeled Pothole

Test Example:

Figure 3.2 Test Image

3.1 Data Preparation

We performed Data preparation steps for training set of 2000 images. The images did not have many noise elements for better pothole detection. This was performed on the dashboard images and the images with only pothole were also used for training.

Below are the steps that were performed for Data preparation:

1. Generate dataset using LabelImg which converts JPEG image file to XML with pothole labeled
2. Convert the XML file to CSV records which has image details as shown in the snippet of the csv file in below Figure 3.1.1.

train_labels

filename	width	height	class	xmin	ymin	xmax	ymax
1_G0029202.JPG	3680	2760	pothole	1944	1378	2060	1457
1_G0029284.JPG	3680	2760	pothole	1075	1403	1202	1445
1_G0029284.JPG	3680	2760	pothole	2706	1292	2763	1314
1_G0029431.JPG	3680	2760	pothole	2171	1481	2376	1534
1_G0029509.JPG	3680	2760	pothole	1878	1331	1947	1361
1_G0029509.JPG	3680	2760	pothole	1739	1468	1783	1509
1_G0029717.JPG	3680	2760	pothole	1839	1251	1893	1265

Figure 3.1.1 CSV File

3. Convert this csv file to train.record which will be used by the model as input.

3.2 Data Validation

The models will be tested on 500 images to derive the loss and accuracy of the model for the test images. The loss and accuracy will be compared to finally chose the result model. The models are explained in detail in the next section.

4. Machine Learning Models

4.1 YOLO

YOLO is an object detection algorithm that is popular for detecting objects in images. It is unique as it only as the look at the image one time which increases speed and efficiency. This algorithm works by splitting images into an X by X grid. Each one of these grids will then take m bounding boxes which enclose an object respectively. These bounding boxes are YOLO will output a confidence score which determines how certain the predicted bounding box actually encloses some object. Based on this score, we can determine whether or not how accurate the algorithm was in detecting a pothole or whether the image is not a pothole. YOLO is faster than many algorithms, which is a big factor in detecting the potholes in a huge dataset. An issue with YOLO is that it struggles with detecting small objects due to the spatial restrictions of the algorithm. The major reason why YOLO is a popular algorithm to use is because of its speed as well as its ability to be used in real time.

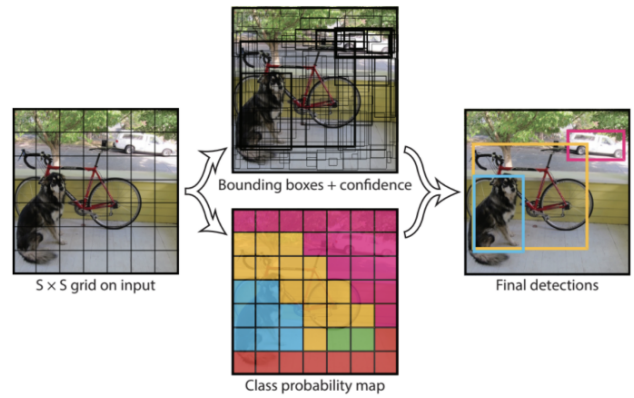

Figure 4.1.1 YOLO Architecture

The above image depicts the architecture of YOLO. It goes through the process of finding the bounding boxes as well as the confidence scores.

Finally, after those steps the image can have a bounding box with the object being detected.

4.2 CNN

CNN is deep learning algorithm which takes in a input image, assigns importance to various aspects/objects in the image and is able to differentiate one from the other. There is very few pre-processing steps needed for CNN as opposed to other machine learning classification algorithms. The convolutional network is able to successfully capture the spatial and temporal dependencies in an image by applying relevant filters.

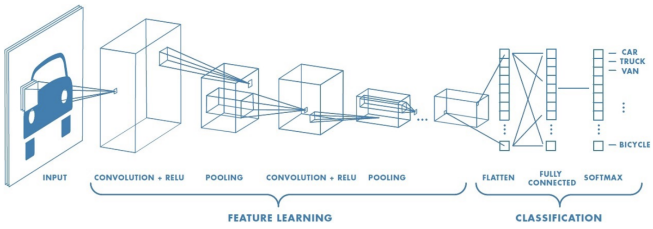

Figure 4.2.1 CNN Architecture

The above Figure 4.2.1 shows the process of the CNN algorithm. The first step is to provide an image. Then, parameters are chosen, padding is added, and filters are applied to that image. Next, convolution is performed on the image. Pooling is also performed in order to reduce the numbers of parameters. Additional convolutional layers can be added if needed. Then, the next step is to flatten the output and feed it to the fully connected layer. The last step is to output the class.

CNN has a few advantages and disadvantages. We chose to use CNN because it uses feature learning to find the most relevant pattern for classification. The CNN algorithm focuses only on relevant features of the input image requiring fewer parameters to use in the convolution. In the end, the result of using CNN reduces the amount of operations needed and increases efficiency at the same time.

4.3 SSD

SSD which stands for single shot detector is another algorithm used in object detection. This algorithm works by taking in images and splitting an image into default boxes. Each box has an intersection over union value(IOU). There is also a threshold value that is selected. Now, if the default box has an IOU value within the threshold then this means that an object has been detected successfully. The image below shows the architecture of how SSD works and how it goes through the process of detecting an object. After it goes through the convolution layer, the last step of detecting the object is done.

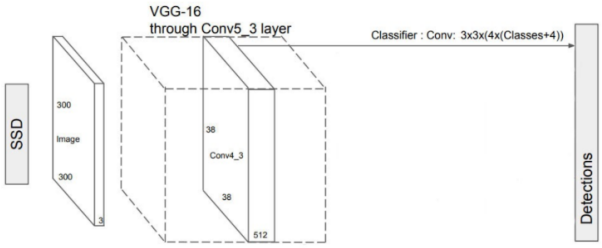

Figure 4.3.1 SSD Architecture

SSD is an efficient algorithm in the sense that it outputs the result with high accuracy but has a downside of being a bit slower than other algorithms such as YOLO.

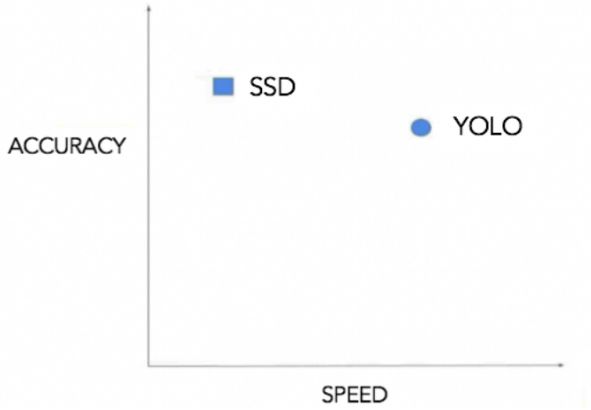

Figure 4.3.2 Comparison of SSD and YOLO by Accuracy and Speed

As seen in the above image, SSD is slower than YOLO but has a higher level of accuracy. The primary reasons why SSD is a good choice of algorithm to use in an object detection project is that speeds up the whole detection process by eliminating the need for a region proposal network. It also generates a good accuracy while using lower resolution images.

4.4 Faster R-CNN

Similar to CNN, R-CNN is also used for classification models. There are two main

networks in R-CNN. One is RPN, which is used to generate region proposals. Another is a network that uses the region proposals to detect objects. We chose to use R-CNN because is uses RPN to generate fixed set regions and anchor boxes for object detection. Faster R-CNN has faster speed.

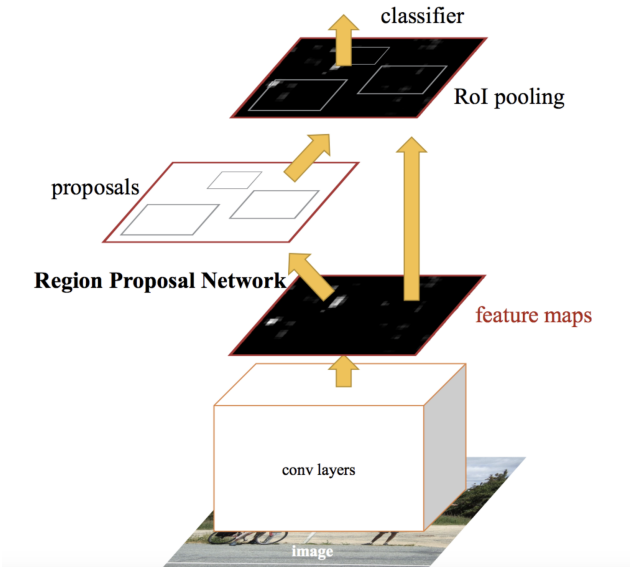

Figure 4.4.1 Faster R-CNN Architecture

The above Figure 4.4.1 shows the steps in Faster R-CNN. First, the images go through convolution layers and feature maps are extracted. Then, a sliding window is used in RPN for each location anchor boxes are used for generating region proposals. After this, for each location anchor boxes are used for generating region proposals. The fourth step is to use the CLS layer to find objects. Finally, the reg layer is used to get coordinates.

4.5 Model Selection and Justification

We implemented the machine learning and Deep Learning models of YOLO, SSD, Faster R-CNN and found that SSD performed well with time and accuracy for detecting the pothole as compared to rest of the models. Below is the final model architecture that is created based on our experiments.

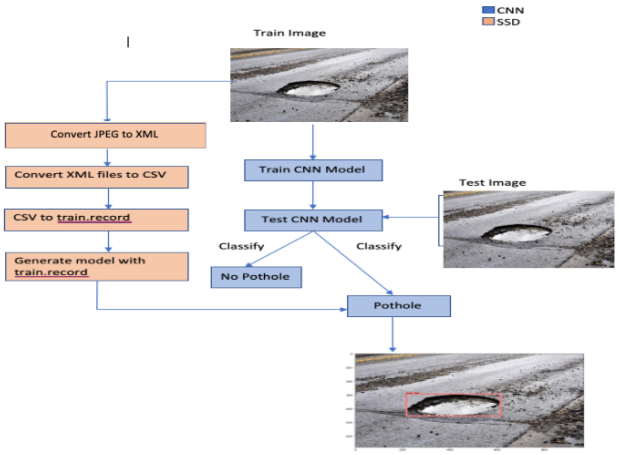

Figure 4.5.1 Model Architecture

The model is created as combination of CNN and SSD. Because SSD works best for object detection when image is positive. It gives an accuracy of 78% for a negative image (no pothole) whereas CNN model gives an accuracy of 93% with negative and positive images. So, It is a good idea to first test the image with CNN to classify the image as positive and negative which is more than 93% accuracy and then. The positive classified image is tested with trained SSD model to detect the pothole and result a bounding box on the pothole with detection accuracy.

5. Results with Case Study

There were multiple changes that helped improved results. The first change that we made was using a bigger data set. Next, we used images that had prominent potholes. Lastly, combining CNN and SSD proved to improve accuracy.

5.1 YOLO

Figure 5.1.1 YOLO Pothole Detection Test Image

As shown in the Figure 5.1.1 above, YOLO does not accurately detect the entire pothole dimensions. YOLO produced our lowest results, compared to the other models. The reason that YOLO did not perform well is because the model is known to have trouble detecting small objects in an image.

5.2 CNN

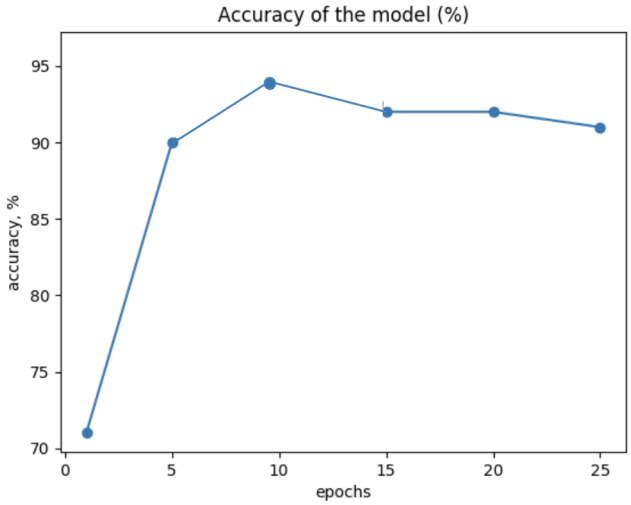

Figure 5.2.1 CNN Accuracy vs Epochs

The Figure 5.2.1 above shows the accuracy for CNN at each epoch. As the image shows, the accuracy started to decrease after 10 epochs, which means that after the 10th epoch, the model kept showing the same results but, stopped producing better results. The classifications for CNN were either positive (pothole detected) or negative (no pothole detected).

5.3 SSD

From the plot of loss Figure 5.3.1, we can see that the model has comparable performance on both train and validation datasets (labeled test). If these parallel plots start to depart consistently, it might be a sign to stop training at an earlier epoch.

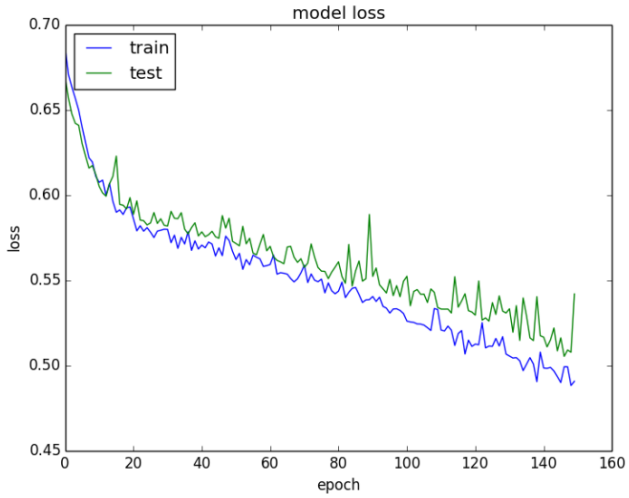

Figure 5.3.1 SSD Loss vs Epoch

From the plot of accuracy Figure 5.3.2 we can see that the model could probably be trained a little more as the trend for accuracy on both datasets is still rising for the last few epochs. We can also see that the model has not yet over-learned the training dataset, showing comparable skill on both datasets.

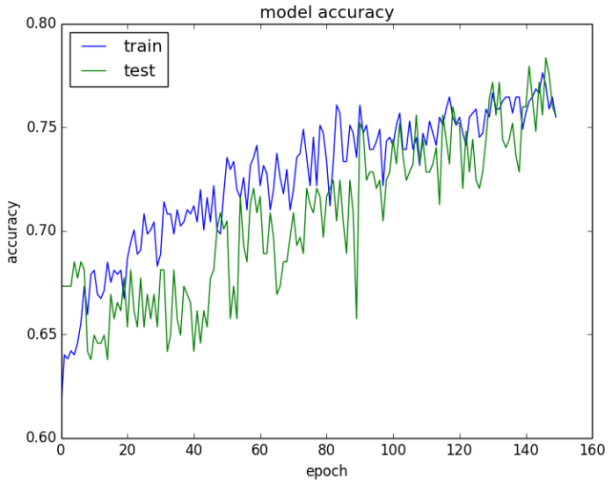

Figure 5.3.2 SSD Accuracy vs Epoch

5.4 Faster R-CNN

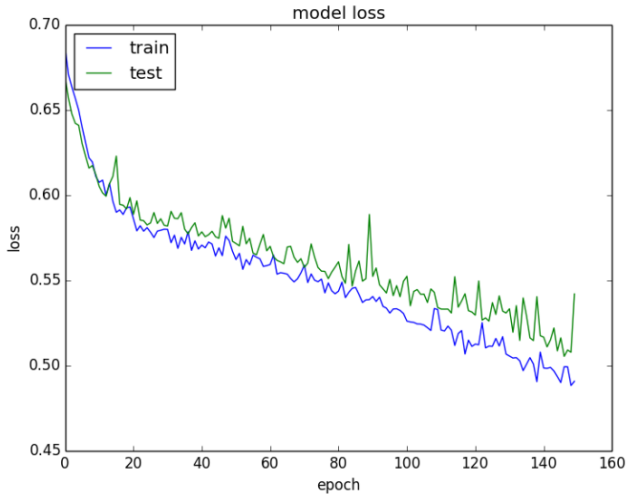

Figure 5.4.1 Faster R-CNN Loss vs Epoch

6. Comparison of Results

The final results of each of the models is as follows:

- YOLO: 63%
- SSD: 82%
- CNN: 93%
- Faster R-CNN: 74%

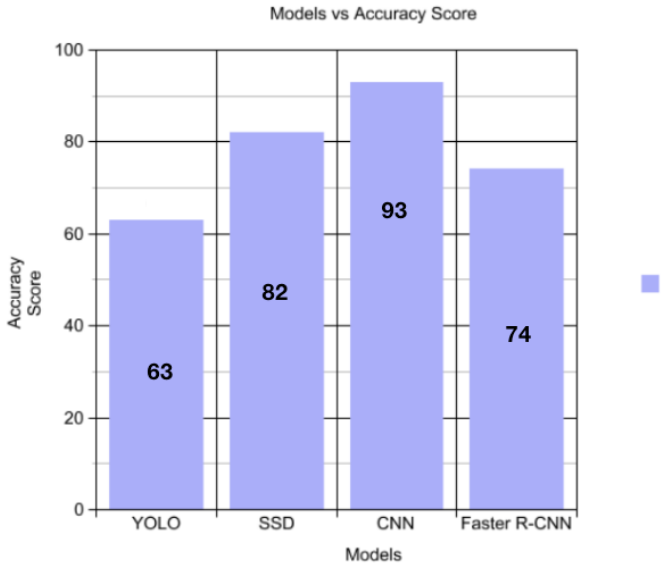

Figure 6.1 Accuracy Score for All Models

Higher resolution plotted in Figure 6.2 below improves object detection for small objects significantly while also helping large objects. When decreasing resolution by a factor of two in both dimensions, accuracy is lowered by 15.88% on average but the inference time is also reduced by a factor of 27.4% on average. This result graph proved that high resolution images will result in higher mAP(Mean Average Precision)

Figure 6.2 Overall mAP vs GPU Time

7. Experience and Lessons Learned

This pothole detection project gave us numerous areas to brainstorm, learn, adapt and grow as developers. One critical aspect of growth happened as a team project. We learned how to deal with disagreements, misconceptions and use version control while maximizing comfort to collaborate and effectively share code. We had a couple of designs and suggestions regarding the technologies we want to use. But we traced pros and cons for every framework and decided on what is best for the team, keeping in mind the skills that other team members already know. Another critical aspect of development happened in research and analytical part as individuals. While moving forward in the project, we spent a lot of time to research and download appropriate pothole pictures that would be used as the training set. A few times we had to work through the training data we had to make sure we can teach our algorithm appropriately. This critical analysis was very hectic but also very rewarding in the end. The final aspect of growth is growing as a machine learning algorithmic/developer. The various machine learning algorithms and libraries we explored gave us deeper insight into the future possibility and become a more skilled member of the tech group of the Bay Area. We measured pros and cons of multiple functionalities that would give us the same result. For example, we used TensorFlow which is one of the biggest assets to machine learning in the industry. We applied

SSD, CNN, YOLO, Faster R-CNN algorithms which gave us a preview of industry level technologies. This project helped us gain insight to industry practices and improved our intellect in machine learning. In today's time, machine learning is one of the most crucial and fastest growing developer communities. Being a part of such a community opens lots of doors for us.

7.1 Future Work

Machine Learning technology will allow a vehicle to gather data about the location and severity of potholes by means of pothole detection using the implemented model. It can be extended to include broken drains and manhole covers. This data will be used to train the model and the output will be notification to the driver allowing the driver to slow down. This requires pothole detection over a distance, so the model should detect small pothole amidst the noise objects in the image.

1. For detecting small pothole, SSD which we implemented gives best accuracy as compared to other models. The combined model implemented should give faster output because negative images will be classified with CNN fast with greater accuracy and then SSD will be detecting potholes. SSD works best on small potholes as compared to the other models.

8. References

[1] S NIENABER, M.J. (THINUS) BOOYSEN, RS KROON 2015 "Detecting Potholes Using Simple Image Processing Techniques and Real-World Footage"

[2] VIJAYA BASHKAR, GOWRI MANOHAR 2016 "Surface Pothole Depth Estimation Using Stereo mode of Image Processing"

[3] J. Karuppuswamy, V. Selvaraj, M.M. Ganesh, E.L. Hall, Detection and avoidance of simulated potholes in autonomous vehicle navigation in an unstructured

[4] C. Koch, I. Brilakis, Pothole detection in asphalt pavement image, Advanced Engineering Informatics, Vol. 25(3), pp. 507- 515, 2011.

[5] Mustaffar M, Ling TC, Puan OC. "Automated Pavement Imaging Program (APIP) For Pavement Cracks Classification and Quantification – A Photogrammetric Approach", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B4. Beijing, 2008.

[6] Hannes Schulz and Sven Behnke. Object-class segmentation using deep convolutional neural networks. In Proceedings of the DAGM Workshop on New Challenges in Neural Computation, 2011.