Smart System for Potholes Detection Using Computer Vision with Transfer Learning

Article · July 2021

3 authors:

Ezekiel Promise Sochima
Rivers State University of Science and Technology

Onate Egerton Taylor
Rivers State University

Dumnamene Sako
Rivers State University of Science and Technology

ISSN No:-2456-2165

Smart System for Potholes Detection Using Computer Vision with Transfer Learning

P. S. Ezekiel[1], O. E. Taylor[2*] & D. J. S. Sako[3]
[1,2,3] Department of Computer Science, Rivers State University, Port Harcourt, Nigeria
*Corresponding Author:

Abstract:- Road potholes are extensively enormous primary disappointments out and about surface. They are brought about by withdrawal and extension of the road surface as water saturates into the ground. To guarantee traffic wellbeing, it is critical and important to regularly investigate and fix road potholes. This paper presents a smart system for potholes detection using computer vision with transfer learning. The system starts by acquiring a pothole images as data, pre-processing the data by creating image annotation and image augmentation on the pothole images. We also created sub directories by creating called images and annotation where we placed our training images in the image folder and the annotated files (which was generated in Xml format) into the annotation folder that we created. We then train our model using transfer learning. By transfer learning, we downloaded a pre-trained yolov3 weight file trained on a coco dataset for object detection. We then set the batch_size to be equal to 4, with a tensorflow Gpu of version ==1.13.1, number of experiments = 200, and train_from_pre_trained_model = pre-trained-yolov3 (the weight file we downloaded). After training, we evaluated and saved the trained model. We had a training accuracy of about 97%. We carried out a real time pothole detection on a live streaming video using opencv library, where we detected multiple potholes images.

Keyword:- *Potholes, Computer Vision, Transfer Learning, Yolov3.*

I. INTRODUCTION

Road networks are key resources, as they support the proficient progression of individuals and merchandise. However, in spite of this basic social and monetary job, numerous nations spent just half or even less of what might be needed for legitimate road upkeep. Keeping up with road networks is frequently the duty of territorial or nearby government elements with spending limitations. Helpless support brings about street deserts, which add to monetary misfortunes, vehicle harm, and fatalities. Early recognition and detailing of such imperfections would permit protection support, which would bring about lower cost, usefulness benefits, and expanded wellbeing. Visual reviews are the most widely recognized technique at present used to evaluate road condition. Their viability is obliged by asset accessibility and the significance of a given street. To be sure, high volume traffic motorways will in general be outwardly reviewed every day and have their vehicle stream observed continually. Low need rustic streets are investigated not exactly once per week, and regularly have undetected harm for longer time [1].

Road potholes are extensively enormous primary disappointments out and about surface. They are brought about by withdrawal and extension of the road surface as water saturates into the ground. To guarantee traffic wellbeing, it is critical and important to regularly investigate and fix road potholes. Presently, potholes are routinely distinguished and revealed by guaranteed investigators and primary designers. This undertaking is nonetheless, tedious and drawn-out. Besides, the discovery results are consistently emotional, on the grounds that they rely altogether upon staff insight. The event of pothole has expanded quickly in uncommon climate like substantial downpour in summer and snowfall, and affects traffic security and street harm. It messes social up like vehicle breakage and mishaps, which are causing social expenses. In this manner, programmed pothole discovery strategies are being read for productive pothole fix and asphalt the executives [2].

Occupied driving, speeding or other driver mistakes are primary driver of mishaps around the world; in any case, awful road conditions are additionally a critical reason. The state of a street ends up being perilous because of number of reasons like flooding, precipitation, harms caused, e.g., by over-burden large vehicles, or poor actual upkeep of the road. Road condition evaluation includes distinguishing and investigating unmistakable sorts of road surface pain, similar to potholes, breaks or surface changes as being upkeep significant highlights. Full scale road highlights are characterized by being of traffic pertinence. For instance, hindrances are likewise traffic-significant highlights; they additionally require identification for driver help. A pothole is an extraordinary instance of street trouble. It very well may be a subjectively molded primary imperfection of a street, and an exact ID of its "line" is commonly inconceivable. They can be ambiguously laid out, yet their greatest profundity can be distinguished all the more unequivocally. Articles like vehicles, people, cyclists, canines or felines are of explicitly characterized shapes (and now recognized by machine learning because of appearance properties) [3].

At present, the primary techniques for recognizing potholes actually depend on open announcing through hotlines or sites, for instance, the potholes detailing site in

ISSN No:-2456-2165

Ohio [4]. Nonetheless, this revealing typically needs exact data of the dimensional and area of potholes. Additionally, this data is normally obsolete too. This paper presents a smart system that will detect and alert the users of potholes in Nigeria.

II. RELATED WORKS

Varona et.al. (2020) proposed a deep learning technique to automatically recognize the various types of road facet, and to recognize potholes from destabilizations delivered by hindrances or driver activities in the group detecting based application setting. Specifically, they investigate and apply distinctive deep learning models: convolutional neural network, LSTM algorithm, and repository figuring models. Their experiment was conducted with true data, and the outcomes showed a promising exactness in tackling the two problems. The accuracies are 98% for convolutional neural network, and 78% for Last Short-Term Memory [5].

Anaiss et.al. (2019) proposed a virtual roadway network auditor (VRNI), which persistently screens roadway conditions and gives choice help to directors and specialists. The virtual street network monitor proposes a novel roadway damage recognition strategy dependent on two versatile one-class support vector machine models, which were applied on the vertical and sidelong speed increase information. They assessed this technique on information from a genuine arrangement on school transports in New South Wales, Australia. Their trial results show that their proposed strategy reliably distinguishes 97.5% of the road vandalization of 4% bogus caution rate that identify with considerate inconsistencies, such development joints [6].

Dhiman and Klette (2019) created two methods dependent on sound system vision examination of roadway conditions in front of the vehicle. They additionally planned two models for deep learning-based pothole recognition. They completed a trial assessment of the two proposed deep learning models as far as accuracy and review, and they acquired the accompanying outcomes: 96.2and 92.3 for accuracy and review of the main model and 65.2 and 53.2 for the second model [3].

Fan et.al. (2019) present a powerful pothole recognition framework. To recognize vandalize and intact roadway regions, they changed a thick difference map. To accomplish more noteworthy divergence change productivity, they used brilliant area search and dynamic programming to gauge the change boundaries. For the extraction of potential intact roadway regions from the changed divergence map, they received Otsu's thresholding strategy. Their test results show that the correct recognition accuracy of their proposed framework is around 98.7% and the general pixel-level precision is roughly 99.6% [7].

Fox et.al. (2017) create a publicly supported framework to distinguish and confine potholes in multi-path conditions utilizing accelerometer information from implanted vehicle sensors. Their proposed publicly supported framework decreases the necessary organization data transmission by deciding the grade of the road and bank point data in every vehicle to channel speed increase parts that don't compare to pothole conditions. They assess their framework on reenacted and true information, break down tradeoffs in the quantity of vehicles and the measure of data transmission needed for precise location, and contrast the outcomes with the more straightforward single path identification situation. The looked at result shows that as the path keeps on expanding, the precise pace of location diminishes [8].

Bansal et.al. (2020) proposes a procedure DeepBus for ongoing ID of surface anomalies on roadways utilizing Internet of Things (IoT). The proposed procedure utilizes IoT sensors to identify potholes progressively. The area of these potholes would be accessible on a midway facilitated map. They looked at the exhibition of different AI models (Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, Random Forest and Ensemble Voting) in view of various boundaries (Accuracy, F-score, Precision and Recall) and distinguished that Random Forest is the best model for pothole identification with 86.8% exactness [9].

Kang and Choi (2017) utilized a 2D LiDAR and Camera in identifying potholes. To further develop the pothole discovery accuracy, the mix of heterogeneous sensor framework was utilized. They likewise utilized Image-based pothole recognition technique to work on the precision of pothole location. The exhibition of their proposed procedure depends on blunder rate. Their test result shows that their proposed strategy have a low mistake pace of about 0.02% [10].

Li et.al. (2018) propose a sound system vision framework for pothole identification. The framework contains two USB cameras taking photograph at the same time. They utilized the acquired boundaries gotten from the camera in adjustment with checkerboard to figure the divergence map. They projected 2-dimensional picture focuses to 3-dimensional world focuses utilizing the dissimilarity map. All focuses underneath the street surface model was recognized as pothole district. The exploratory outcome shows that powerful identification of potholes in various street and light conditions [11].

Tsai and Chatterjee (2018) presents a pothole recognition strategy utilizing 3D asphalt information and a watershed technique. They complete tests utilizing the 3D information gathered on tenth Street, Atlanta, Georgia and 6 mi of street on U.S. 80, Savannah, Georgia, has shown a 94.97% exactness, 90.80% accuracy, and 98.75% review. It has been exhibited that the proposed strategy is promising for pothole location and can give a dependable technique to pothole identification, particularly when 3D asphalt information have been gathered for break discovery and as of now accessible [12].

III. METHODOLOGY

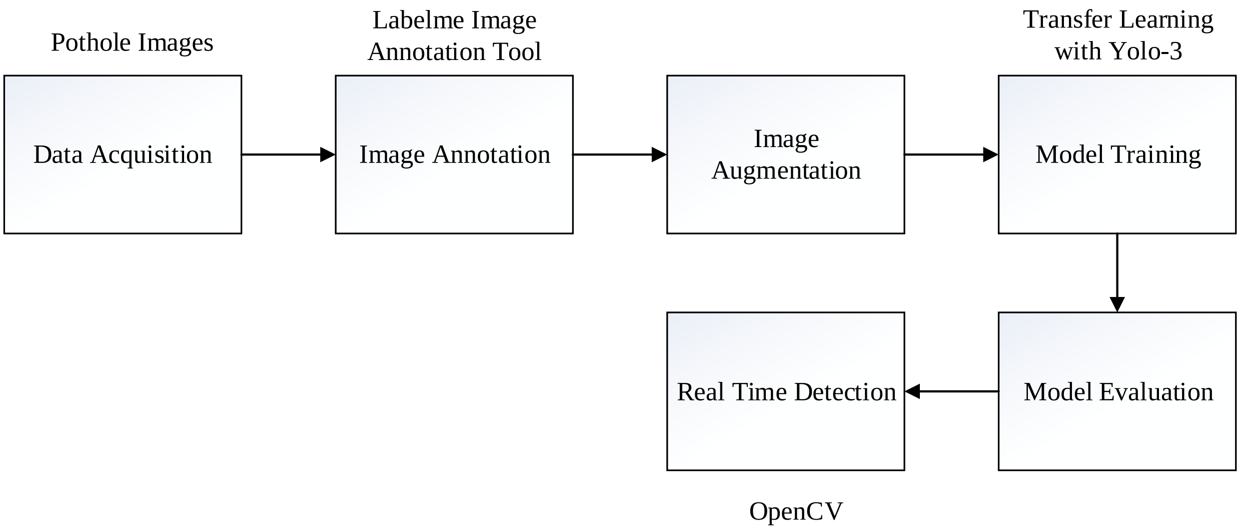

Figure 1 Architecture of the Proposed System

Data Acquisition: The dataset consist of 1200 images of road with potholes. These images was collected online using google search engine to search for potholes images in Port Harcourt City, Rivers State Nigeria. The acquired dataset will be categorized into two which is a training data and a testing data.

Image Annotation: Image annotation is the task of annotating an image with labels. The labels are used to give information to a computer vision model about the given images that is to be used as an input data. For annotation of our training and testing images, we will be using Labelme annotation tool in annotating our images. The labelme will be used in segmenting our images (training and testing), drawing bounding boxes and generating coordinates for each of the training image.

Image Augmentation: image augmentation is used in expanding our dataset. We also used image augmentation parameters for rotation, scaling, cropping, flipping, translation, affine transformation zooming and sharing of images. This was used as a result of generating more images and changing the position of the images so as to obtain a better recognition of our trained model.

$x = f_x . \frac{x}{z} + x_c, \; y = f_y . \frac{y}{z} + y_c,$ ... Image Translation

Where x and y are the image pixel, fx and fy are the focal lengths in x and y coordinate direction, and (xc, yc) is the principal point in the image plane.

Model Training: The model will be trained using a pre-trained yolo-3 coco weight model that was trained on a coco dataset for different object detection. We will be optimizing the parameters of the pre-trained model weights to be a batch_size of 4, 200 number of experiments, and the number of object names to be one (1). This process of training a new model from a pre-trained model weights is called transfer learning.

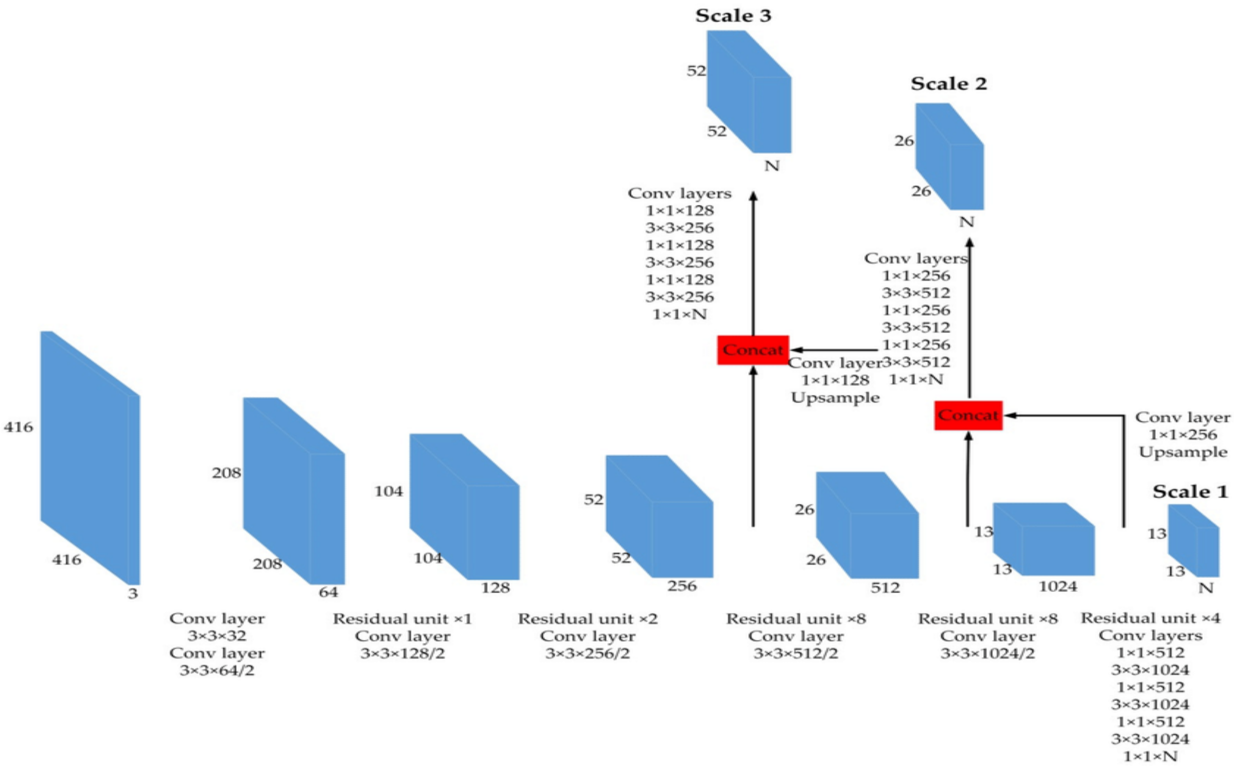

Figure 2 Architecture of Yolo v3

```
image = readImage()
NoOfCells = 7
NoOfClasses = 4
threshold = 0.7
step = height(image)/NoOfCells
prediction_class_array = new_array(size(NoOfCells,NoOfCells,NoOfClasses))
predictions_bounding_box_array = new_array(size(NoOfCells,NoOfCells,NoOfCells,NoOfCells))
final_predictions = []
for (i<0; i<NoOfCells; i=i+1):
        for (j<0; j<NoOfCells;j=j+1):
                cell = image(i:i+step,j:j+step)
                prediction_class_array[i,j] = class_predictor(cell)
                predictions_bounding_box_array[i,j] = bounding_box_predictor(cell)
                best_bounding_box =  [0 if predictions_bounding_box_array[i,j,0, 4] > predictions_bounding_box_array[i,j,1, 4] else 1]
                predicted_class = index_of_max_value(prediction_class_array[i,j])
                if predictions_bounding_box_array[i,j,best_bounding_box, 4] * max_value(prediction_class_array[i,j]) > threshold:
                        final_predictions.append([predictions_bounding_box_array[i,j,best_bounding_box, 0:4], predicted_class])
print final_predictions
```

Figure 3: Pseudo Code of Yolo v3 Algorithm

Model Evaluation: This involves the testing and evaluation the general performance of the model based on accuracy, precision, cross-validation. We will evaluate the performance of our trained model based on accuracy, and cross validation.

$$Precision = \frac{TP}{TP + TP}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP is True Positive Result, TN is the True Negative Result; FP is the False Positive Result, and FN is the False Negative Result.

Real Time Detection: We will carry out a real time detection of potholes on Port Harcourt City in Rivers State using Open CV in loading a live video from a web cam, we will show the web cam to major roads in Port Harcourt in detecting multiple potholes as possible.

IV. RESULT AND DISCUSSION

This system proposes a real time system for potholes detection using computer vision. The system starts by acquiring images of potholes in Port Harcourt City Rivers State. The dataset comprises of 1200 images of potholes. These images was pre-processed using image annotation tool and image augmentation. The image annotation tool used in this work is that of the labelme annotation tool. We carried out the annotation of each of images we acquired. We performed segmentation of the images we will be using as data by drawing a bounding box of each of the images. We also label each of the images, so as to make it known to our proposed model for easy identification of potholes. We also perform image augmentation so as to rotate, zoom, translate, and transform the training images so as to make it easy for the model to detect potholes despite the position of the image. After the processes we then divided our dataset into training set and testing set. We allocate 80% of the images to our training folder and 20% of the remaining data to our test folder. We also created two sub folders called images and annotation for both the training and testing folder. The images folder contains the pothole images while the annotation folder contains an xml file for each of the images that we are to use for both training and testing of our model. The xml file in the annotation folder contains the coordinate (x, y) of each of the images for both training and testing data. This coordinates is to enable us to draw a bounding box of various potholes images. After these processes, we then train our model using transfer learning. By transfer learning, we downloaded a pre-trained yolov3 weight file trained on a coco dataset for object detection. We then set the batch_size to be equal to 4, with a tensorflow Gpu of version ==1.13.1, number of experiments = 200, and train_from_pre_trained_model = pre-trained-yolov3 (the weight file we downloaded). Figure 4 shows the loss values obtained at the first seven training steps. Figure 5 shows the evaluation of our model performance. Figure 7 shows the matrix evaluation of our trained model while figure 8, and 9 shows real time detection of potholes images that is carried out using opencv, and figure 10 shows the models evaluation in terms of accuracy, where model2 has the highest accuracy of about 97%.

```
Using TensorFlow backend.
Generating anchor boxes for training images and annotation...
Average IOU for 9 anchors: 0.78
Anchor Boxes generated.
Detection configuration saved in  potholes/json/detection_config.json
Training on:    ['potholes']
Training with Batch Size:  4
Number of Experiments:  200

Epoch 1/200
480/480 [==============================] - 395s 823ms/step - loss: 36.9000 - yolo_layer_1_loss: 3.2970 - yolo_layer_2_loss: 9.4923 - yolo_layer_3_loss: 24.1107 - val_loss: 15.6321 - val_yol
Epoch 2/200
480/480 [==============================] - 293s 610ms/step - loss: 11.9330 - yolo_layer_1_loss: 1.3968 - yolo_layer_2_loss: 4.2894 - yolo_layer_3_loss: 6.2468 - val_loss: 7.9868 - val_yolo_
Epoch 3/200
480/480 [==============================] - 293s 610ms/step - loss: 7.1228 - yolo_layer_1_loss: 1.0583 - yolo_layer_2_loss: 2.2863 - yolo_layer_3_loss: 3.7782 - val_loss: 6.4964 - val_yolo_l
Epoch 4/200
480/480 [==============================] - 297s 618ms/step - loss: 5.5802 - yolo_layer_1_loss: 0.9742 - yolo_layer_2_loss: 1.8916 - yolo_layer_3_loss: 2.7144 - val_loss: 6.4275 - val_yolo_l
Epoch 5/200
480/480 [==============================] - 295s 615ms/step - loss: 4.8717 - yolo_layer_1_loss: 0.7568 - yolo_layer_2_loss: 1.6641 - yolo_layer_3_loss: 2.4508 - val_loss: 6.3723 - val_yolo_l
Epoch 6/200
480/480 [==============================] - 300s 624ms/step - loss: 4.7989 - yolo_layer_1_loss: 0.8708 - yolo_layer_2_loss: 1.6683 - yolo_layer_3_loss: 2.2598 - val_loss: 5.8672 - val_yolo_l
Epoch 7/200
```

Figure 4: Training processes with loss values

```
Model File:  Potholes/models/detection_model-ex-07--loss-4.42.h5
Using IoU :  0.5
Using Object Threshold :  0.3
Using Non-Maximum Suppression :  0.5
Potholes: 0.9231
mAP: 0.9231
===============================
Model File:  potholes/models/detection_model-ex-10--loss-3.95.h5
Using IoU :  0.5
Using Object Threshold :  0.3
Using Non-Maximum Suppression :  0.5
Potholes: 0.9725
mAP: 0.9725
===============================
Model File:  potholes/models/detection_model-ex-05--loss-5.26.h5
Using IoU :  0.5
Using Object Threshold :  0.3
Using Non-Maximum Suppression :  0.5
Potholes: 0.9204
mAP: 0.9204
===============================
Model File:  Potholes/models/detection_model-ex-03--loss-6.44.h5
Using IoU :  0.5
Using Object Threshold :  0.3
Using Non-Maximum Suppression :  0.5
Potholes: 0.8120
mAP: 0.8120
===============================
Model File:  potholes/models/detection_model-ex-18--loss-2.96.h5
Using IoU :  0.5
Using Object Threshold :  0.3
Using Non-Maximum Suppression :  0.5
potholes: 0.9431
mAP: 0.9431
===============================
Model File:  potholes/models/detection_model-ex-17--loss-3.10.h5
Using IoU :  0.5
Using Object Threshold :  0.3
Using Non-Maximum Suppression :  0.5
Potholes: 0.9404
mAP: 0.9404
===============================
```

Figure 6: Evaluation of trained model

ISSN No:-2456-2165


```
[{
    'average_precision': {'Potholes': 0.9231334437735249},
    'map': 0.9231334437735249,
    'model_file': 'hololens/models/detection_model-ex-07--loss-4.42.h5',
    'using_iou': 0.5,
    'using_non_maximum_suppression': 0.5,
    'using_object_threshold': 0.3
},
{
    'average_precision': {'Potholes': 0.9725334437735249},
    'map': 0.97251334437735249,
    'model_file': 'hololens/models/detection_model-ex-10--loss-3.95.h5',
    'using_iou': 0.5,
    'using_non_maximum_suppression': 0.5,
    'using_object_threshold': 0.3
},
{
    'average_precision': {'potholes': 0.92041334437735249},
    'map': 0.92041334437735249,
    'model_file': 'hololens/models/detection_model-ex-05--loss-5.26.h5',
    'using_iou': 0.5,
    'using_non_maximum_suppression': 0.5,
    'using_object_threshold': 0.3
},
{
    'average_precision': {'potholes': 0.81201334437735249},
    'map': 0.81201334437735249,
    'model_file': 'hololens/models/detection_model-ex-03--loss-6.44.h5',
    'using_iou': 0.5,
    'using_non_maximum_suppression': 0.5,
    'using_object_threshold': 0.3
},
{
    'average_precision': {'potholes': 0.94311334437735249},
    'map': 0.94311334437735249,
    'model_file': 'hololens/models/detection_model-ex-18--loss-2.96.h5',
    'using_iou': 0.5,
    'using_non_maximum_suppression': 0.5,
    'using_object_threshold': 0.3
},

]
```


Figure 7: Matrix evaluation in terms of precision

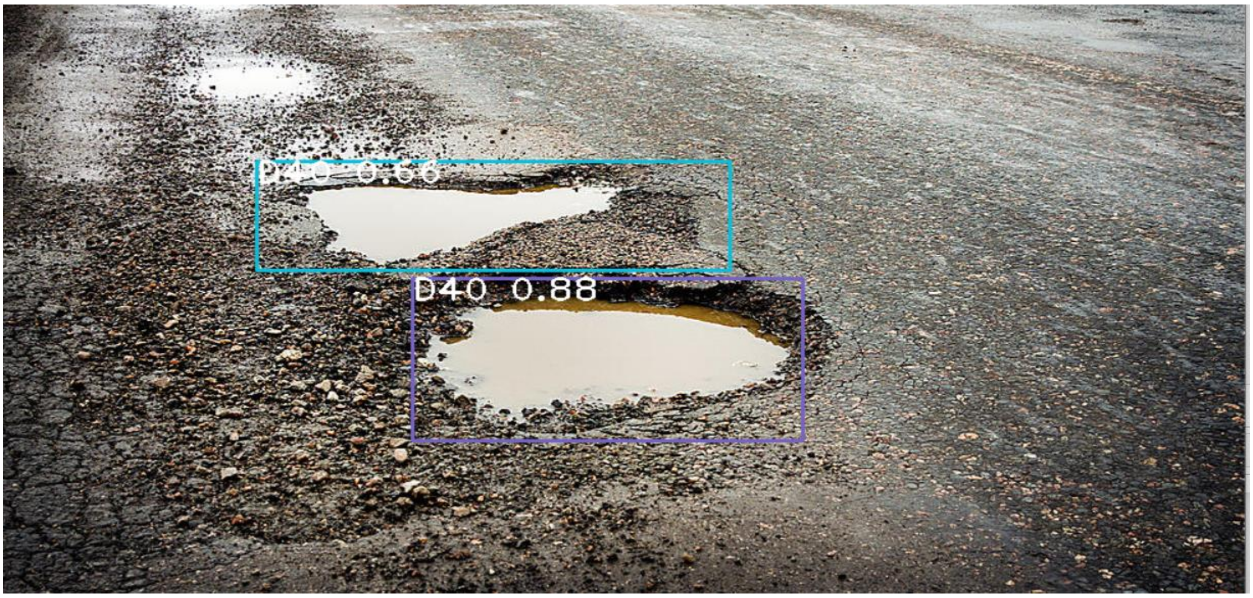

Figure 8: Detected images of Potholes

ISSN No:-2456-2165

Our trained model detected more than one pothole in images with confidence level of 66 %and 88%.

Figure 9: Detected Images of Potholes

Here, our model shows 97% confidence level while detecting potholes.

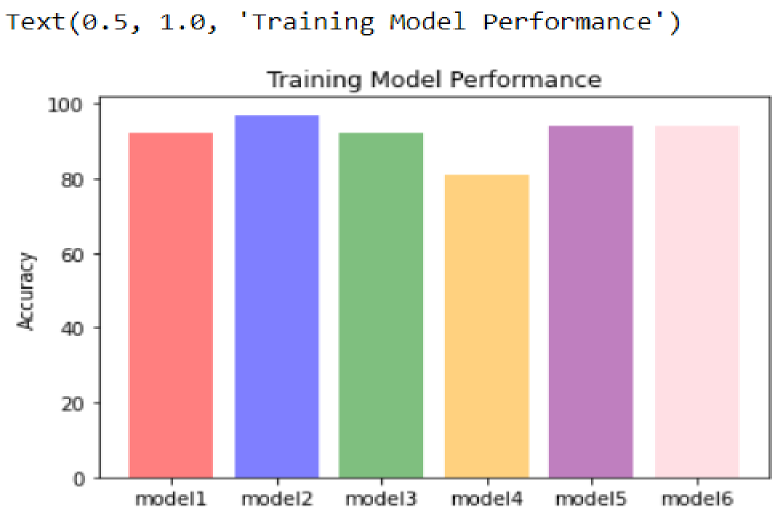

Figure 10: Bar chat Representation of Training Model

This shows the model performance in terms of accuracy during training. We 6 models just to find the model with the best accuracy result. The bar chat above shows that the model2 has the highest training accuracy result ahead other models.

V. CONCLUSION AND FUTURE WORK

This paper presents a smart system for potholes detection using computer vision with transfer learning. The system starts by acquiring a pothole images as data, pre-processing the data by creating image annotation and image augmentation on the pothole images. We also created sub directories by creating called images and annotation where we placed our training images in the image folder and the annotated files (which was generated in Xml format) into the annotation folder that we created. We then train our model using transfer learning. By transfer learning, we downloaded a pre-trained yolov3 weight file trained on a coco dataset for object detection. We then set the batch_size to be equal to 4, with a tensorflow Gpu of version ==1.13.1, number of experiments = 200, and train_from_pre_trained_model = pre-trained-yolov3 (the weight file we downloaded). After training, we evaluated and saved the trained model. We carried out a real time pothole detection on a live streaming video using opencv library, where we detected multiple potholes images. This work can further be extended by creating an android app that will detect potholes and use google map in saving the location of the detected potholes to database.

REFRENCES

[1]. J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, H. Balakrishnan , "The pothole patrol: using a mobile sensor network for road surface monitoring", In: proceedings of the 6th International Conference on Mobile systems, applications, and services, ACM, 29–39, 2008
[2]. C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," Advanced Engineering Informatics, vol. 29, no. 2, pp. 196–210, 2015.
[3]. A. Dhiman, R. Klette, "Pothole Detection Using Computer Vision and Learning", Ieee Transactions On Intelligent Transportation Systems, 21(8), 3536-3550, 2019.
[4]. R. Madli, S. Hebbar, P. Pattar, and V. Golla, "Automatic Detection and Notification of Potholes and Humps on Roads to Aid Drivers," IEEE Sensors Journal, vol. 15, no. 8, pp. 4313–4318, 2015.
[5]. B. Varona, A. Monteserin, A. Teyseyre "A deep learning approach to automatic road surface monitoring and pothole detection", Personal and Ubiquitous Computing, 24(4), 519-534, 2020.
[6]. A. Anaissi, N. L. Dang Khoa, T. Rakotoarivelo., · M. M. Alamdari, · Y. Wang, "Smart pothole detection system using vehicle-mounted sensors and machine learning", Journal of Civil Structural Health Monitoring, 9(1), 91-102, 2019.
[7]. R. Fan, U. Ozgunalp, B. Hosking, M. Liu, I. Pitas, "Pothole Detection Based on Disparity Transformation and Road Surface Modeling", Ieee Transactions On Image Processing, vol. 29, 897-908, 2019.
[8]. A. Fox, B.V.K. Vijaya Kumar, J. Chen, F. Bai, "Multi-lane Pothole Detection from Crowdsourced Undersampled Vehicle Sensor Data", IEEE Transactions on Mobile Computing 16(12), 3417-3430, 2017.
[9]. K. Bansal, K. Mittal, G. Ahuja, A. Singh, S. S. Gill, "DeepBus: Machine learning based real time pothole detection system for smart transportation using IoT", Internet Technology Letters 3 (3), vol.156, 1-6, 2020
[10]. B. Kang, S. Choi, "Pothole Detection System Using 2D LiDAR and Camera", 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), 744-746, 2017.
[11]. Y. Li, C. Papachristous, D. Weyer, "Road Pothole Detection System Based on Stereo Vision", Naecon 2018-IEEE National Aerospace and Electronics Conference, 292-297, 2018.
[12]. Y. Tsai, A. Chatterjee, "Pothole Detection and Classification Using 3D Technology and Watershed Method", Jouranl of Computing in Civil. Engineering, 2018, 32(2), 04017078, 2018