A Project Based Seminar Report

on

"Road Damage Detection using Region of CNN and YOLO"

Submitted to the

Savitribai Phule Pune University

In partial fulfillment for the award of the Degree of

Bachelor of Engineering

in

Information Technology

by

Vishal Krishan

(T150228559/4359 & TE)

Under the guidance of

Prof. Geeta Patil

Department of Information Technology

Army Institute of Technology

Dighi Hills, Pune-Alandi Road,

Pune 411015.

Semester-VI, Third Year Engineering

2018-2019

CERTIFICATE

This is to certify that the project based seminar report entitled "Road Damage Detection using Region of CNN and YOLO" being submitted by Vishal Krishan (4359 / T150228559 and TE IT) is a record of bonafide work carried out by her under the supervision and guidance of Prof. Geeta Patil in partial fulfillment of the requirement for **TE (Information Technology Engineering) - 2015 Course** of Savitribai Phule Pune University, Pune in the academic year 2018-2019.

Date: 31/03/2019

Place: Pune

Prof. Geeta Patil
Seminar Guide

Dr. Sangeeta Jadhav
Head of the Department

Dr. B. P. Patil
Principal

This Project Based Seminar report has been examined by us as per the Savitribai Phule Pune University, Pune requirements at Army Institute of Technology, Pune-411015 on . . . . . .

Internal Examiner

External Examiner

ACKNOWLEDGMENT

I am highly indebted to my guide Prof. Geeta Patil for her guidance and constant supervision as well as for providing necessary information regarding the seminar report and also for her support in completing the seminar report. I would like to express my special gratitude and thanks to Seminar Coordinator Prof Ashwini Sapkal for giving me such attention and time.

This acknowledgment would be incomplete without expressing my thanks to Dr. Sangeeta Jadhav, Head of the Department (Information Technology) for her support during the work.

I would like to extend my heartfelt gratitude to my Principal, Dr.B. P. Patil who provided a lot of valuable support, mostly being behind the veils of college bureaucracy.

I would also like to express my gratitude towards my parents and friends for their kind co-operation and encouragement which help me in completion of this report. My thanks and appreciations also go to my colleague in developing the seminar report and people who have willingly helped me out with their abilities.

(Vishal Krishan & Signature)

Abstract

Roads are the basic components of the land transportation and the quality of roads commonly decreases because of ageing and deterioration. In order to improve the damage detection, we need to use newest ideas and features. The R-CNN and YOLO methods uses the object detection algorithm which analyses the visual content of an image to recognize instances of a certain damage category, then outputs the category and location of the detected objects. These algorithms help recognizing the conditions of roads by detecting specific types of roads damages in order to use maintenance resources efficiently.

Contents

Contents

List of Figures

Chapter 1

INTRODUCTION TO ROAD DAMAGE DETECTION USING DEEP LEARNING

1.1 Introduction to road damage detection using deep learning

1.1.1 Introduction to Project

Roads make a crucial contribution to economic development and growth and bring important social benefits. They are of vital importance in order to make a nation grow and develop. In addition, providing access to employment, social, health and education services makes a road network crucial in fighting against poverty. Roads open up more areas and stimulate economic and social development. For those reasons, road infrastructure is the most important of all public assets.

Surveys show that adequately maintaining road infrastructure is essential to preserve and enhance those benefits. But a backlog of outstanding maintenance has caused irreversible deterioration of the road network. If insufficient maintenance is carried out, roads can need replacing or major repairs after just a few years. That deterioration spread across a road system very quickly results in soaring costs and a major financial impact on the economy and citizens.

In 2015, there were about five lakh road accidents in India, which killed about

1.5 lakh people and injured about five lakh people. India, as a signatory to the Brasilia declaration, intends to reduce road accidents and traffic fatalities by 50 percent by 2022.

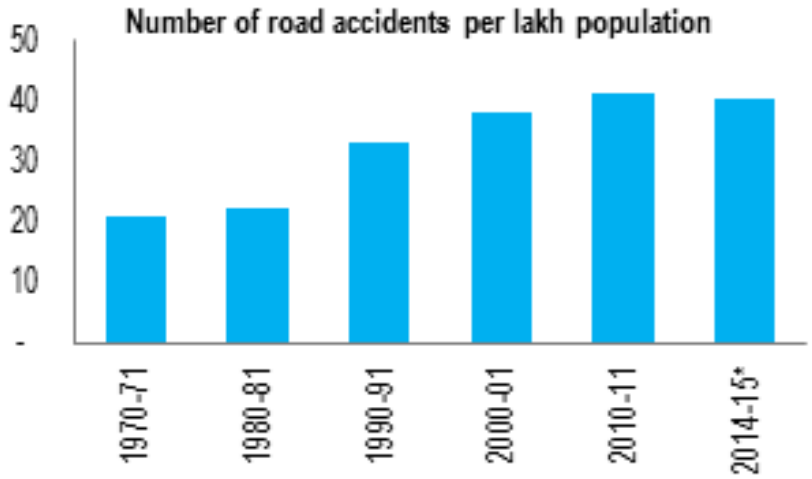

Figure 1.1: Number of accidents per lack population

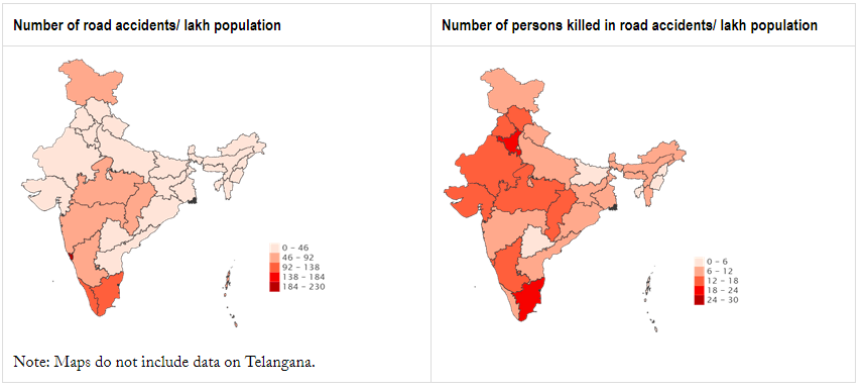

Figure 1.2: Heatmap of India as per accidents

- Potholes has caused 3597 last year , which is much more that any other cause of unnatural deaths.
- They create traffic jab wasting lot of human hours.
- Damage Roads leads to burn more fuel leading to more pollution.
- Jerk due to Potholes leads to several health problems including back issues and pain.

- India import 85 Billion Dollar of fuel i.e 6 Lac Core (6,00000,00,00,000). 10 percent of fuels is wasted due to potholes. We can save lot of this money and increase the rupee value.
- Damaged to imported vehicles caused due to potholes further increase the dollar bill hitting rupee badly

However, due to lack of financial resources, many local governments are not able to conduct sufficient inspections. In fact, some municipalities automate damage determination by using high-performance sensors, but because of their high cost, many municipalities are unable to introduce them. Therefore, there is a need for a method that makes it easy to judge the damage of the road surface at low cost.

1.1.2 Motivation behind project topic

The techniques which are generally used for authentication of user has many loop holes such as time consuming, costly equipment and tools, more man power, high cost of repairment and many more. Our motive is to provide a solution without all the previous loop holes.

1.1.3 Aim and Objective(s) of the work

To build a solution for road damage detection to detect potholes and cracks using Deep Learning. The key to the solution is less equipment cost i.e. using smartphone cameras mounted on the windshields of the cars (mostly cabs), low man power i.e. no need of physical civic body at the site to collect data and high performance i.e. using efficient Deep Learning algorithms.

1.1.4 Road Damage Detection Using Region of CNN and YOLO

The seminar topic is road damage detection using RCNN and YOLO. Road damage detection is of importance to road maintenance, which typically requires huge amount of manual efforts. In this study, we leverage deep learning models to analyze the road images to detect road damages efficiently. Specifically, we studied the train and test different deep learning methods to identify efficient models with high accuracy. RCNN is a two-stage detector, it applies object classification and bounding-box regression on proposed

region and can achieve high accuracy in object detection. In this work, we have studied Damage detection algorithm for training and testing.

Research on damage detection of road surfaces using image processing techniques has been actively conducted achieving considerably high detection accuracies. However, many studies only focus on the detection of the presence or absence of damage. However, in a real-world scenario, when the road managers from a governing body needs to repair such damage, they need to know the type of damage clearly to take effective action. In addition, in many of these previous studies, the researchers acquire their own data using different methods. Hence, there is no uniform road damage dataset available openly, leading to the absence of a benchmark for road damage detection. This study makes three contributions to address these issues. First, to the best of our knowledge, for the first time, a large-scale road damage dataset is prepared. This dataset is composed of 9,053 road damage images captured with a smartphone installed on a car, with 15,435 instances of road surface damage included in these road images. These images are captured in a wide variety of weather and illuminance conditions. In each image, the bounding box representing the location of the damage and the type of damage are annotated. Next, we use the state-of-the-art object detection method using convolutional neural networks to train the damage detection model with our dataset, and compare the accuracy and runtime speed on both, a GPU server and a smartphone. Finally, we show that the type of damage can be classified into eight types with high accuracy by applying the proposed object detection method. The road damage dataset, our experimental results, and the developed smartphone application used in this study are made publicly available.

Chapter 2

Introduction to object detection Algorithms

2.1 Object Detection Algorithms

An object Detection algorithm analyzes the visual content of an image to recognize instances of a certain object category, then outputs the category and location of the detected objects. Object detection technology has seen a rapid adoption rate in various and diverse industries. It helps self-driving cars safely navigate through traffic, spots violent behavior in a crowded place, assists sports teams analyze and build scouting reports, ensures proper quality control of parts in manufacturing, among many, many other things.

There are numerous approaches to perform object detection in an image which are followed by the following algorithms

- RCNN
- Fast-RCNN
- Faster-RCNN
- YOLO
- SSD

2.2 Approaches for detecting potholes in Road images

- **Naive Way (Divide and Conquer)** [9] - The simplest approach we can take is to divide the image into four parts:

  1. Upper left-hand side corner
  2. Upper right-hand side corner
  3. Lower left-hand side corner
  4. Lower right-hand corner

  Now the next step is to feed each of these parts into an image classifier. This will give us an output of whether that part of the image has a pothole or not. If yes, mark that patch in the original image.

  But it cannot identify the entire crack from single part hence the prediction will be catastrophic.

Figure 2.1: Naive way Approach

- **Increase the number of divisions** [9]- In order to improve the previous approach, we can exponentially increase the number of patches into the system i.e. increasing the number of bounding boxes around the pothole.

  But with increasing the number of bounding boxes around the same patch will be wastage of computation power since all the computations and bounding boxes yields the same patch.

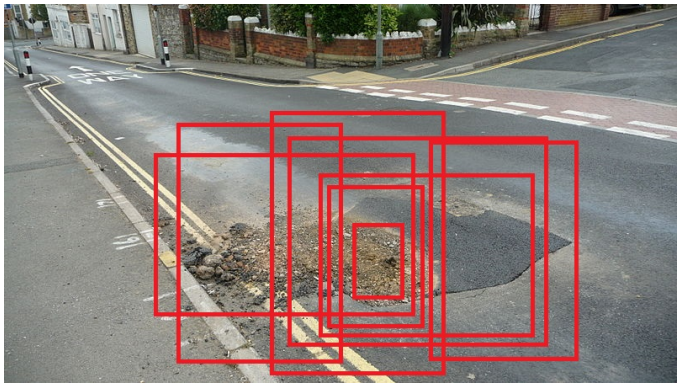

Figure 2.2: Detection based on number of divisions

- **Performing structured divisions** [9]- In order to build our object detection system in a more structured way, we can do the following steps

  1. Divide the image into S*S grid
  2. Define the centroids for each patch
  3. For each centroid, take n different patches of different heights and aspect ratio
  4. Pass all the patches created through the image classifier to get predictions

  Again, taking precise values of S and n is the key to good prediction results. Some model may require large value, and some may need a few.

Figure 2.3: Image Division in S * S grid cells

Figure 2.4: Object division in S * S grid cells

2.3 Region of Convolutional Neural network

The problem with Convolutional Neural Network approach is that it mainly focuses for image classiďŹcation. This means one has to select a huge no of regions to classify the image and this could computationally blow-up the processing.

To bypass the problem of selecting a huge number of regions, Ross Girshick et al [11]. proposed a method where we use selective search to extract just 2000 regions from the image called as region proposals. Thus, now, instead of trying to classify a huge number of regions, you can just work with these 2000 regions. These region proposals are generated using the selective search algorithm which is written below.

Selective Search:

Step 1: It first takes an image as input.

Step 2: Then, it generates initial sub-segmentation i.e. candidate regions from the image

Step 3: The technique then combines the similar regions to form a larger region, based on color similarity, texture similarity, size similarity, and shape compatibility using Greedy Algorithm.

Step 4: Finally, these regions then produce the final object locations (Region of Interest)

Following is the succint summary of the steps followed in RCNN to detect objects

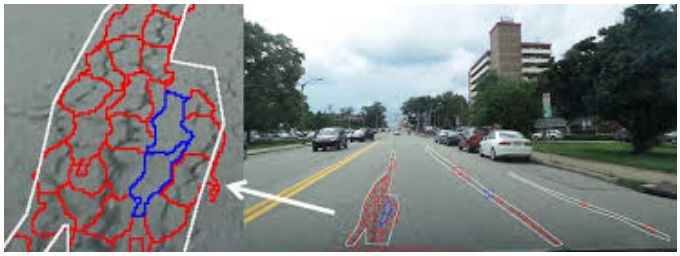

Figure 2.5: Selective Search in Road Damages

Step 1: We first take a pre-trained convolutional neural network.

Step 2: Then, this model is retrained. We train the last layer of the network based on the number of classes that need to be detected.

Step 3: The third step is to get the Region of Interest for each image. We then reshape all these regions so that they can match the CNN input size.

Step 4: After getting the regions, we train SVM to classify objects and background. For each class, we train one binary SVM.

Step 5: Finally, we train a linear regression model to generate tighter bounding boxes for each identified object in the image.

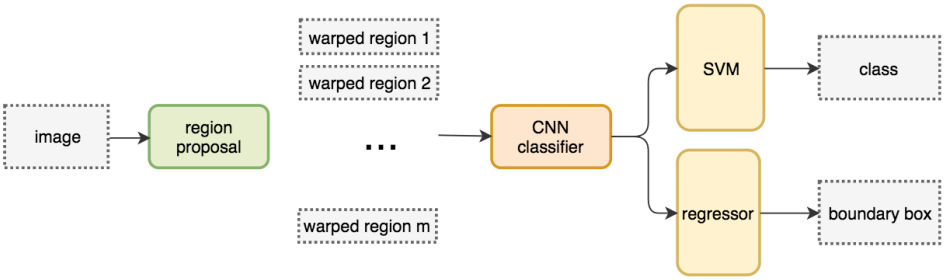

Figure 2.6: Diagramatic representation of pothole detection in RCNN

Though RCNNâs mean Average Precision (mAP) has an improvement over OverFeat architecture but it also has many drawbacks which leads to its limited use in Object detection.

2.4 Problems with RCNN

So far, weâve seen how RCNN can be helpful for object detection. But this technique comes with its own limitations.

1. It still takes a huge amount of time to train the network as you would have to classify 2000 region proposals per image. Suppose we have N images, then the number of CNN features will be N*2,000
2. The entire process of object detection using RCNN has three models:
   - CNN for feature extraction
   - Linear SVM classifier for identifying objects
   - Regression model for tightening the bounding boxes.

   All these processes combine to make RCNN very slow. It takes around 40-50 seconds to make predictions for each new image, which essentially makes the model cumbersome and practically impossible to build when faced with a gigantic dataset.
3. The selective search algorithm is a fixed algorithm. Therefore, no learning is happening at that stage. This could lead to the generation of bad candidate region proposals.

These problems have been again solved by Ross Girshick et al. in his new Fast R-CNN algorithm.

Chapter 3

You Only Look Once

3.1 Intorduction to YOLO

RCNN object detection uses regions to localize the object within the image. The network does not look at the complete image, instead it looks for the parts of the image which have high probabilities of containing the object. This becomes the fallback for the algorithms like RCNN and a new algorithm rises which instead of looking at the parts of image one at a time, looks the complete image only once and perform its operation.

You Only Look Once is an object detection algorithm much different from the region-based algorithms seen above. It basically merges the two steps of the RCNN algorithm into one step by developing a neural network which internally divides the image into regions and predicts categories and probabilities for each region. Thus, applying the prediction once makes YOLO achieve a signiďŹcant speedup compared to R-CNN-based algorithms; hence can be used for real-time prediction.

To solve the road damage type detection problem, we consider a road damage as a unique object to be detected. In particular, each of the different road damage types is treated as a distinguishable object. Then, we use YOLOv2 (that has a state-of-the-art convolutional neural network at its core.) Object detection algorithm which will be trained on the road damage dataset to learn the visual patterns of each road damage type.

3.2 Dataset settings for YOLOv2

One of the best ways to collect the dataset is through smartphone cameras mounted on the windshield of the cars. In order to divide the road damages among types, the distribution of the damage classes should be well balanced. Hence, we used the Python Augmentor library to generate synthesized images for training images that contain such classes.

While augmentation, we carefully selected the image processing techniques (e.g., brightening, gray-scale) provided by the Augmentor tool to assure that the road damage scenes were not affected. Another technique that we considered in processing the training dataset is cropping. We cropped every image to create a smaller size of the original image while making sure that the cropped image contains the annotated regions.

Such a dataset enables the training model to focus on learning the features of regions of interest properly by discarding irrelevant scenes (e.g., sky view). Consequently, we had three training datasets: original (D), augmented (Da), and cropped (Dc).[1]

Following are the steps performed in YOLO Object detection.

Step 1: It first takes an image an as input

Step 2: Then, it divides the image into S*S grid cells

Step 3: 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 potholes. Confidence = Pr(object) * IOU IOU denotes intersection over union between the predicted box and ground truth. Each cell also predicts C conditional class probabilities, Pr(object). Here object can be treated as a Pothole.

Step 4: 4. Both confidence score and class prediction will output one final score telling us the probability that this bounding box contains a specific type of pothole.

To create an object detector, Darknet53 model can be used for the YOLO framework. The model will be trained using the road damage classes. To experiment our solution thoroughly, we will be generating different versions of the trained models using each dataset (D, Da, and Dc) by varying three parameters: number of iterations for model training (T), minimum conďŹdence threshold (C), and non-maximum suppression (NMS).

For every test image, YOLO generates a set of predicted boxes and each box is tagged with a prediction conďŹdence score and a predicted label. To avoid reporting the boxes tagged with very low prediction conďŹdence score, we discarded the boxes whose conďŹdence scores were below C. Furthermore, increasing the value of NMS (default value 0.45) increases the number of overlapped predicted boxes; hence increasing the chances of correctly predicting a ground-truth box.

Figure 3.1: Pothole detection in YOLOv2

3.2.1 Methodology

There were two different model architectures used in conducting the training. The motivation behind two models is to improve the performance of the trained model within YOLOv2 by changing some parameters [2].

- Architecture F1 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 before leaky rectified linear unit (ReLu) activation, and a maxpool layer at the 1st, 3rd, 7th, 11th, and 17th layers. In order to train the dataset, there is a need to reinitialize the final convolutional layer, so it outputs a tensor with a 13 x 13 x 30 shape, where 30 = 5 bounding boxes x (4 coordinates + 1 confidence value + 1 class probability).

Layer	Type	Filters	Size/Pad/Stride	Output
0	Convolutional	32	3 x 3 / 1 / 1	416 x 416
1	Maxpool	-	2 x 2 / 0 / 2	208 x 208
2	Convolutional	64	3 x 3 / 1 / 1	208 x 208
3	Maxpool	-	2 x 2 / 0 / 2	104 x 104
4	Convolutional	128	3 x 3 / 1 / 1	104 x 104
5	Convolutional	64	1 x 1 / 0 / 1	104 x 104
6	Convolutional	128	3 x 3 / 1 / 1	104 x 104
7	Maxpool	-	2 x 2 / 0 / 2	52 x 52
8	Convolutional	256	3 x 3 / 1 / 1	52 x 52
9	Convolutional	128	1 x 1 / 0 / 1	52 x 52
10	Convolutional	256	3 x 3 / 1 / 1	52 x 52
11	Maxpool	-	2 x 2 / 0 / 2	26 x 26
12	Convolutional	512	3 x 3 / 1 / 1	26 x 26
13	Convolutional	256	1 x 1 / 0 / 1	26 x 26
14	Convolutional	512	3 x 3 / 1 / 1	26 x 26
15	Convolutional	256	1 x 1 / 0 / 1	26 x 26
16	Convolutional	512	3 x 3 / 1 / 1	26 x 26
17	Maxpool	-	2 x 2 / 0 / 2	13 x 13
18	Convolutional	1024	3 x 3 / 1 / 1	13 x 13
19	Convolutional	512	1 x 1 / 0 / 1	13 x 13
20	Convolutional	1024	3 x 3 / 1 / 1	13 x 13
21	Convolutional	512	1 x 1 / 0 / 1	13 x 13
22	Convolutional	1024	3 x 3 / 1 / 1	13 x 13
23	Convolutional	1024	3 x 3 / 1 / 1	13 x 13
24	Convolutional	1024	3 x 3 / 1 / 1	13 x 13
25	Route [16][1]	512	-	26 x 26
26	Convolutional	64	1 x 1 / 0 / 1	26 x 26
27	Reorganize	256	2 x 2 / 0 / 2	13 x 13
28	Route [27] [24][2]	1280	-	13 x 13
29	Convolutional	1024	3 x 3 / 1 / 1	13 x 13
30	Convolutional	30	1 x 1 / 0 / 1	13 x 13

Figure 3.2: Architecture F1 in YOLOv2

- Architecture F2 F2 is a modified version of the F1 architecture in an attempt to reduce the computational costs and model size of the neural network. F1 requires roughly 48 million parameters. On the other hand, F2 needs only 18 million parameters with just 27 layers. The final result shows that the average precision and recall score of the F2 architecture is better than the F1 architecture while being smaller in size and faster in computation speed.

  Few modifications require to build the F2 architecture from F1.

Step 1: After calculations, each 23rd and 24th layer consumes more than

nine million parameters. Layer 29 alone requires more than 11 million parameters. Removing these three convolutional layers can remove about 30 million parameters.

Step 2: Make F2âs 23rd layer have a filter size of 2048 by modifying F1âs 26th convolutional layer from 64 filters to 256 filters.

Step 3: The reorganized 25th layer has a depth of 1024 by reorganizing the 24th layer from 26 x 26 x 256 to 13 x 13 x 1024.

Step 4: Route layer 26 and route layer 25 (13 x 13 x 1024) with the 22nd convolutional layer (13 x 13 x 1024) output 13 x 13 x 2048.

Step 5: F2-Anchor is created by modifying the width and height of F1âs anchor boxes.

Layer	Type	Filters	Size/Pad/Stride	Output
0	Convolutional	32	3 x 3 / 1 / 1	416 x 416
1	Maxpool	-	2 x 2 / 0 / 2	208 x 208
2	Convolutional	64	3 x 3 / 1 / 1	208 x 208
3	Maxpool	-	2 x 2 / 0 / 2	104 x 104
4	Convolutional	128	3 x 3 / 1 / 1	104 x 104
5	Convolutional	64	1 x 1 / 0 / 1	104 x 104
6	Convolutional	128	3 x 3 / 1 / 1	104 x 104
7	Maxpool	-	2 x 2 / 0 / 2	52 x 52
8	Convolutional	256	3 x 3 / 1 / 1	52 x 52
9	Convolutional	128	1 x 1 / 0 / 1	52 x 52
10	Convolutional	256	3 x 3 / 1 / 1	52 x 52
11	Maxpool	-	2 x 2 / 0 / 2	26 x 26
12	Convolutional	512	3 x 3 / 1 / 1	26 x 26
13	Convolutional	256	1 x 1 / 0 / 1	26 x 26
14	Convolutional	512	3 x 3 / 1 / 1	26 x 26
15	Convolutional	256	1 x 1 / 0 / 1	26 x 26
16	Convolutional	512	3 x 3 / 1 / 1	26 x 26
17	Maxpool	-	2 x 2 / 0 / 2	13 x 13
18	Convolutional	1024	3 x 3 / 1 / 1	13 x 13
19	Convolutional	512	1 x 1 / 0 / 1	13 x 13
20	Convolutional	1024	3 x 3 / 1 / 1	13 x 13
21	Convolutional	512	1 x 1 / 0 / 1	13 x 13
22	Convolutional	1024	3 x 3 / 1 / 1	13 x 13
23	Route [16][3]	512		26 x 26
24	Convolutional	256	1 x 1 / 0 / 1	26 x 26
25	Reorganize	1024	2 x 2 / 0 / 2	13 x 13
26	Route [25] [22][4]	2048		13 x 13
27	Convolutional	30	1 x 1 / 0 / 1	13 x 13

Figure 3.3: Architecture F2 in YOLOv2

3.3 Denser Grid Model

As we have studied in Performing structured divisions, the values of S and n should be high enough to maximize efficiency, so in YOLOv2 the network operates at a network resolution of 416 x 416, and after its convolutional layers down sample the images by a factor of 32, the grid cell (output feature map) is 13 x 13 [see Fig. 3.4]. Unlike the desired square input resolution of the inspired model (F1), the dataset comes with various resolutions. After the calculations, the input is decided to use an 832 x 672 resolution because it is the average resolution of the dataset, which will produce a 26 x 21 grid

after down sampling [see Fig. 3.5]. The Den-F2-Anchor model is developed, which is the combination of the F2 architecture with the denser grid model and the anchor box model.

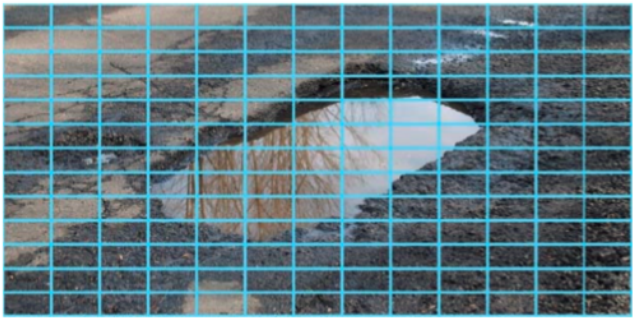

Figure 3.4: 13 * 13 Grid cell of YOLOv2

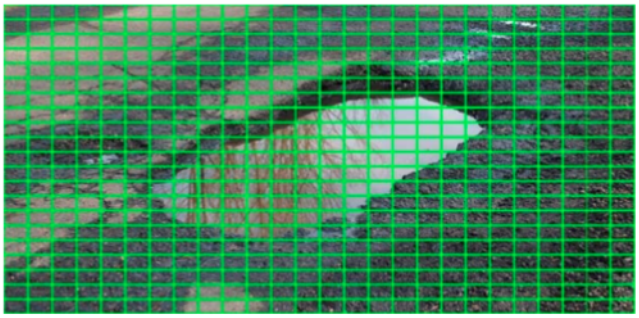

Figure 3.5: 26 * 21 Grid cell of DenF2Anchor

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 this situation is handled using the idea of anchor boxes. Instead of predicting a one dimensional 5 + num-of-class, it instead predicts (5 + num-of-class) Ä num-of-anchorboxes. Each anchor box is designed for detecting objects of different aspect ratios and sizes. For example, box 1 can detect objects that are wide but short, whereas box 2 detects objects with a rectangular shape.

Figure 3.6: Anchor anomaly in YOLO

3.4 Results

We evaluated the performance of the networks using precision and recall scores with the following formulas:

$$AveragePrecision = \frac{\sum_i TP/(TP+FP)}{n} \tag{3.1}$$

$$Recall = \frac{\sum_i TP/(TP+FN)}{n} \tag{3.2}$$

TP, FP, and FN denote true positive, false positive, and false negative, respectively, with n representing the total number of testing images. A good example of training can be performed on a GeForce GTX 1080 with 10 GB RAM and the use of Adam optimizer because of its tendency for expedient convergence. Training of F1 model is started with a low learning rate so as to quickly reduce loss. After 100 training epochs, the learning rate can be increased to bring finer granularity. F2 Model is also trained similar to F1 model.

The testing results of the research on South Korea roads with the same models is shared below.

Model	Average Precision	Recall	Parameters	Frames Per Second
F1 (YOLOv2)	60.14	65.61	48 million	23
F2-Anchor	67.74	74.93	18 million	**32**
Den-F2-Anchor	**82.43**	**83.72**	**18 million**	21

Figure 3.7: Results in different models of YOLOv2

Chapter 4

CONCLUSION

This seminar is undertaken to explore two important object detection algorithms briefly in order to detect road damages efficiently. RCNN and YOLO both have its merits and demerits over one another. Example RCNN takes more time due to multiple steps in its object detection and YOLO combines the multiple steps of RCNN into single one to become faster, and on the other hand YOLO finds difficulty in detecting smaller objects, suppose the initial small cracks on road goes undetected by the YOLO and it becomes the problem in later stages. These arenât the only two object detection algorithms present in the space, there are other algorithms as well like Fast-RCNN, Faster-RCNN, RetinaNet and YOLOv3 which have their own methodology and performance. In future new Radical concepts are arriving each year which are moving us forward in the better approach of the Object Detection.

Bibliography

[1] 2018 IEEE International Conference on Big Data A Deep Learning Approach for Road Damage Detection from Smartphone Images

[2] Journal of Universal Computer Science, vol 24, no. 9 (2018) Detection of Potholes Using a Deep Convolutional Neural Network

[3] Robotics Institute, School of Computer Science Carnegie Mellon University Vision for Road Inspection

[4] 2018 IEEE International Conference on Big Data Road Damage Detection And ClassiďŹcation In Smartphone Captured Images Using Mask R-CNN

[5] Darknet: Open Source Neural Networks in C http://pjreddie.com/darknet/

[6] Journal of Civil Structural Health Monitoring 2018 Crack Detection in Prestressed concrete structures by measuring their natural frequencies

[7] Towards Data Science https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

[8] Augmentor 0.2.3, https://augmentor.readthedocs.io/en/master/

[9] Analytics Vidhya-Basic Object Detection https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/

[10] Analytics Vidhya-Basic Object Detection on python https://www.analyticsvidhya.com/blog/2018/06/understanding-building-object-detection-model-python/

[11] 2014 IEEE CVPR Rich feature hierarchies for accurate object detection and semantic segmentation