Pothole Classification Using CNNs

Aditya Gera
Department of Mechanical Engineering
Stanford University
adigera@stanford.edu

1 Introduction

Potholes are one of the extreme classes of road damage which can not only cause misalignment of the vehicles from their intended path but also damage vehicle structure and components. Both of these can lead to accidents. The damage caused is also dependent on the speed of the vehicle. Therefore it becomes necessary to avoid them which calls for their detection. With the onset of autonomous vehicles employment in passenger travel, the accurate detection is important for taking evasive measures to prevent potential accidents and reduced vehicle life by preventing subjection to unecessary high stresses. It can also be conducive to road maintenance body in cities.

Use of convolutional neural networks (CNNs) for image classification tasks is quite a popular practice due to their highly efficient results. This study focuses on using convolutional neural networks to come up with a robust model to detect potholes and suggest some unprecedented applications. The borrowed dataset used consists of images taken in South Africa.

2 Related work

Existing studies use various image processing techniques [6], machine learning using accelerometers [3], laser imaging [7], SVMs [2] etc. CNNs have also been used to detect potholes for both detection and classification (on different dataset), or with different architecture like location-aware CNNs [1].

3 Dataset and Features

The dataset used in this study is taken from [5][6]. The downloaded dataset was in 14 different folders containing high resolution images of size 3680 x 2760 with around 1958 images with potholes and around 9289 images without potholes. The images are captured from a camera mounted inside of the car behind the wind-shield during daytime. The images were merged and 2000 (1600 negative and 400 positive) randomly chosen images are used for validation test set and rest were used for training.

3.1 Data Augmentation

The data was augmented and normalized. The data was resized to size 400 x 300, maintaining the aspect ratio, before feeding it to the neural network. The mean and standard deviation of original images were computed for normalization. Random cropping, resizing and horizontal flipping was done to augment the data during training. An example of augmented images is shown in figure 3

4 Methods & Models

Convolutional networks are good at pattern recognition and identifying the complex features in an image, and thus, to tackle this problem, CNNs for 2D images has been used. Due to the limited time of project and with

CS230: Deep Learning, Winter 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

Figure 1: Raw image with Pothole (ResNet18)

Figure 2: Raw image without Pothole (ResNet18)

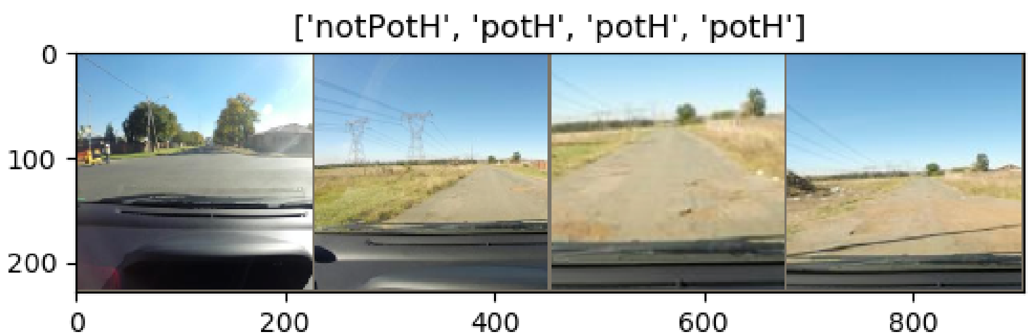

Figure 3: An random set of augmented images

an aim to focus more on experimenting with different techniques in tuning, transfer learning was used on PyTorch.
The experimentation was run on two pre-trained models, ResNet18 and GoogLeNet. For both the networks, the last layer was replaced by a fully connected linear layer with 2 neurons. The output from first neuron was taken to correspond to images without potholes. For loss, cross entropy loss function was used, with and without weights for unbalanced dataset. For optimization, experiments were ran with SGD, SGD with momentum, Adam on ResNet18 and for GoogLeNet, SGD with momentum and Adam were used.

4.1 Models

4.1.1 ResNet18

ResNet18 is a state-of-the-art convolutional network architecture with 18 convolutional layers used for wide variety of applications. The ResNet18 architecture also features skip connections, which adds the output of the lth layer $a^{[l]}$, to the linear output of the l + 2th layer, $z^{[l+2]}$, for certain l. This connection allows the network to learn the identity function more simply, improving network performance. This is because we may choose to "exclude" any hidden layers by learning the identity function. [4]

4.1.2 GoogLeNet

GoogLeNet is another state-of-the-art convolutional neural network architecture, containing 22 trainable layers and 5 max-pooling layers. The network contains Inception modules which make up the majority of the layers. Inception modules essentially allow multiple convolutions of the same input image with different sized kernels. The output volumes of those convolutions are concatentated and fed into the next layer. In this network, we convolve the image with 5 5, 1 1, max-pooling followed by 1 1, and 3 3 filters with stride 2. [4]

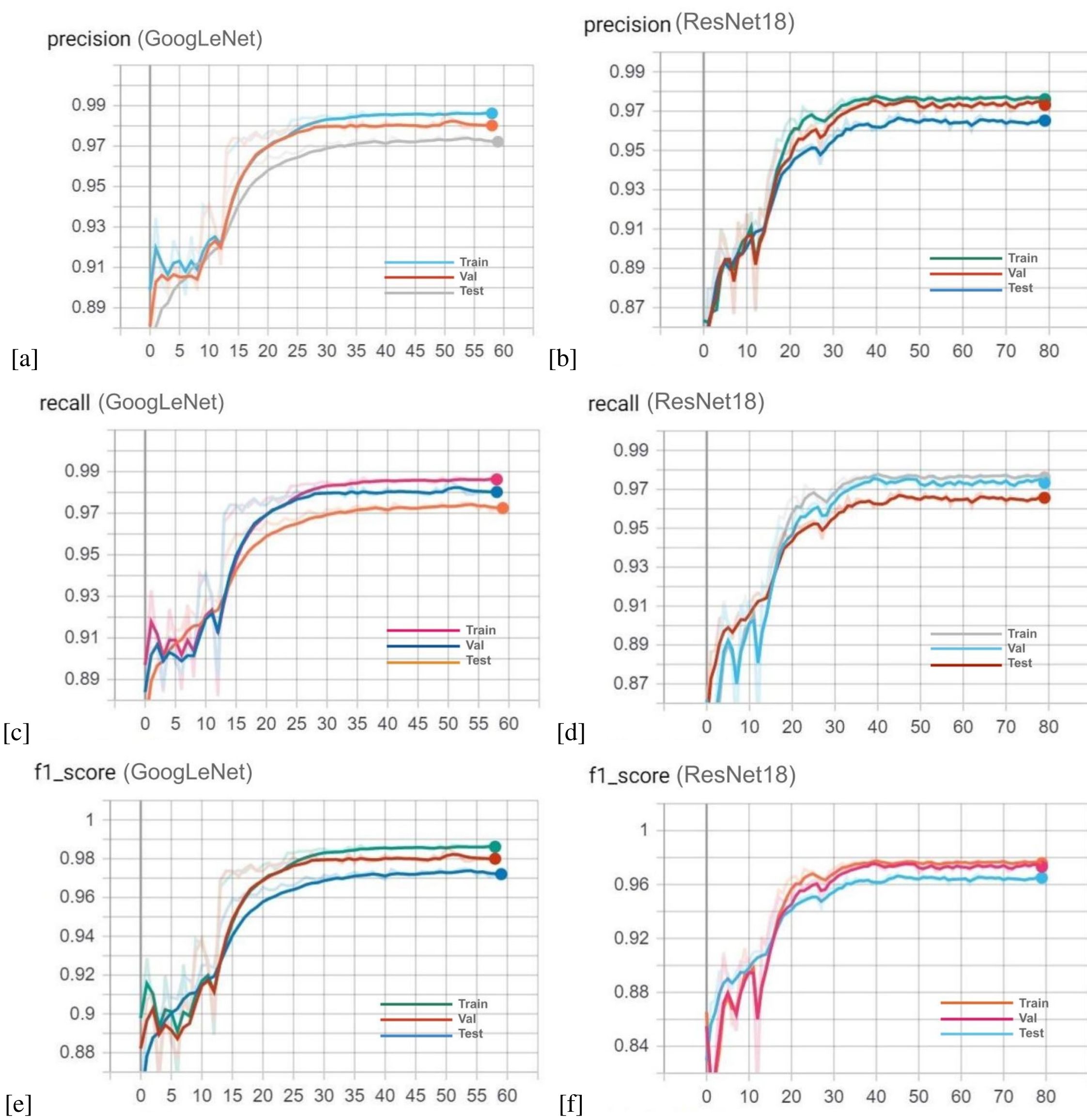

Figure 4: [a] GoogLeNet Precision curve [b] ResNet18 Precision curve [c] GoogLeNet Recall curve [d] ResNet18 Recall curve [e] GoogLeNet f1 score curve [f] ResNet18 f1 score curve

5 Experimentation

Total 63 experiments were conducted, out of which 14 were stopped in between due to insufficient GPU memory or too much noise after certain number of epochs. The models were trained on three AWS EC2 p2.xlarge instances. The first experiment was run on pretrained ResNet18 using original image size, SGD optimizer, batch size 64, with dataset's calculated mean and standard devaition. From there on, to decrease epoch time and for better results, the images were resized to 400 x 300, ImageNet's normalization parameters were used. The various hyper-parameters were varied like momentum (0.88 to 0.93), Adam's β 1 (0.89 to 0.92), learning rate (0.001, 0.0007), decay in learning rate (after 5,7,10,13,15 steps), and weight decay (0.001, 0.0005, 0) individually and then simultaneously to improve the results on both models. At the end, the weights were added to the cross entropy loss function for neutralizing the imbalance in data.

The code can be found here

6 Results & Evaluation

The best trained model's weighted metrics (precsion, recall, and f1-score) for both ResNet18 and GoogLeNet are given in figure 4 for comparison.

The table below shows the maximum values of above graphs

Model	Phase	Precision	Recall	F1 Score
ResNet18	Train	97.60 %	97.61 %	97.59 %
	Validation	97.32 %	97.33 %	97.32 %
	Test	96.46 %	96.57 %	96.51 %
GoogLeNet	Train	98.67 %	98.67 %	98.67 %
	Validation	98.13 %	98.04 %	98.08 %
	Test	97.32 %	97.35 %	97.34 %

6.1 Model Comparison

From the table, we can see that GoogLeNet performed better than ResNet18 by 1% across all phases and metrics. Without much tuning, the performance of GoogLeNet was better on the data set used in this study than the un-tuned ResNet18. This can be related to the inception architecture of the GoogLeNet.

6.2 Model Analysis and development

The data set used was a highly imbalanced data set in terms of positive and negative cases. 60 experiments out of the 63 were done without accounting weights in the cross entropy loss function. After reaching the saturating weighted precision and recall of 98%, the confusion matrix in figure-6 revealed the model's performance to be more accurate on negative examples. Thus, in last 3 experiments, three different set of weights ([0.2, 0.8], [0.4, 0.6], and based on sample size) to penalize more on positive examples. The resultant confusion matrix post these experiments revealed an improvement in detection of positive examples but at the cost of increased mis-classification of negative examples which varied metrics by a very small margin. Interestingly, the number of mis-classified examples increased from 19 to 21 which resulted in drop in weighted precision by 0.7 %. Also, some of the positive examples in the data set are quite difficult examples even for humans either due to the overlapping of tree shadow and potholes, sun glare or relatively very small size of potholes.

7 Future Work

For this particular data set, the future work would be experimenting with more architectures, or even building one from scratch as the images are correlated due to the way of capturing. A smaller and a more focused architecture could be good solution. Another aspect is to collect more varied data, which could be done by collaborating with a government body of a city installed with CCTV cameras. This could be further used to aid the road maintenance bodies by detecting potholes.

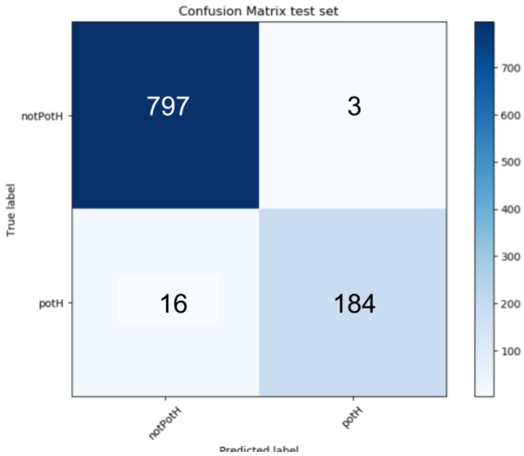

Figure 5: Confusion matrix for Non-weighted Loss

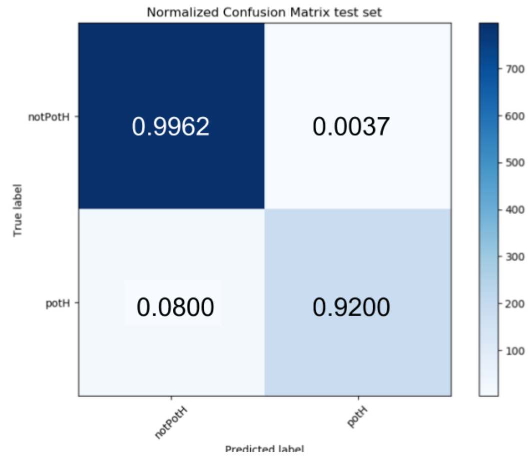

Figure 6: Normalized confusion matrix for Non-weighted Loss

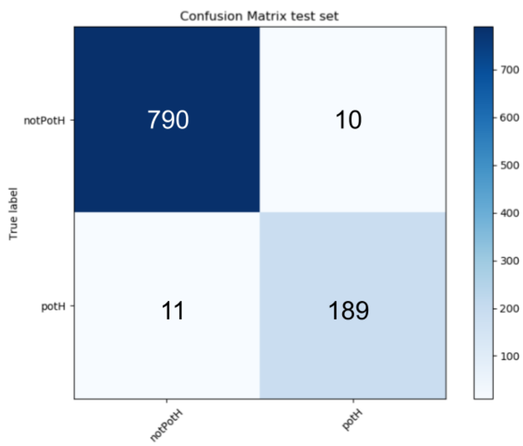

Figure 7: Confusion matrix for Weighted Loss

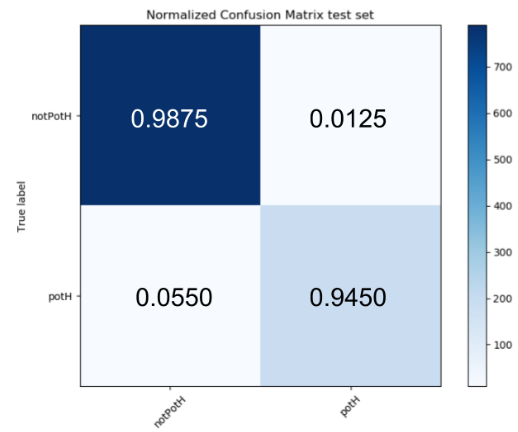

Figure 8: Normalized confusion matrix for Weighted Loss

References

[1] Qinlong Gu Hanshen Chen Minghai Yao. “Pothole detection using location-aware convolutional neural networks”. In: *International Journal of Machine Learning and Cybernetics* (2020). DOI: `10.1007/s13042-020-01078-7`.

[2] Nhat-Duc Hoang. “An Artificial Intelligence Method for Asphalt Pavement Pothole Detection Using Least Squares Support Vector Machine and Neural Network with Steerable Filter-Based Feature Extraction”. In: (2018).

[3] Kihoon Baek Hyunwoo Song and Yungcheol Byun. “Pothole Detection using Machine Learning”. In: *Advanced Science and Technology Letters* (2018). DOI: `10.14257/astl.2018.150.35`.

[4] Eric Zhou Leon Tran Shawn Zhang. “CNN Transfer Learning for Visual Guitar Chord Classification”. In: *NIPS, 2016, Barcelona* (2016).

[5] MJ Booysen S. Nienaber and RS Kroon. “A Comparison of Low-Cost Monocular Vision Techniques for Pothole Distance Estimation”. In: (Dec. 2015). DOI: `10.1109/SSCI.2015.69`.

[6] MJ Booysen S. Nienaber and RS Kroon. “DETECTING POTHOLES USING SIMPLE IMAGE PROCESSING TECHNIQUES AND REAL-WORLD FOOTAGE”. In: *South African Transport Conference* ().

[7] X. Yu and E. Salari. “Pavement pothole detection and severity measurement using laser imaging”. In: *2011 IEEE INTERNATIONAL CONFERENCE ON ELECTRO/INFORMATION TECHNOLOGY* (2011).