A Comparative Study of Deep Learning Methods for Automating Road Condition Characterization

[1] Zurana Mehrin Ruhi
(20141049)
dept. of Computer Science and Engineering
BRAC Univerisity
Dhaka, Bangladesh
zurana.mehrin.ruhi@g.bracu.ac.bd

[2] Farahatul Aziz Sheetal
(16101083)
dept. of Computer Science and Engineering
BRAC Univerisity
Dhaka, Bangladesh
sheetalfarah@gmail.com

[2] Farisha Hossain Prithu
(16101259)
dept. of Computer Science and Engineering
BRAC Univerisity
Dhaka, Bangladesh
farishahossain408@gmail.com

[1] Hossain Airf
Assistant professor
dept. of Computer Science and Engineering
BRAC Univerisity
Dhaka, Bangladesh
hossain.arif@bracu.ac.bd

***Abstract*—Roads in Bangladesh provide infrastructural facilities to both agricultural as well as industrial sectors of the country. Distressed roads can cause fatal accidents as well as largely decelerate sector progress. This makes swift road inspection and repairs one of the most important aspects of our country's holistic growth. As much as it affects the general public, tackling this is as big a problem for the government as well. Currently, the problem for road repair is a multi-stage problem, which involves getting a complaint from a resident, physical road inspection by some official, identifying the type of damage and then comes the process of actually repairing it. Here, we intend to make this cumbersome process simpler, by automating the problem identification stage. We developed a method leveraging the Machine Learning and Deep Learning capabilities that can potentially detect a damaged road and identify the type of damage viz. pothole and crack. We self-captured data from the roads and streets, thus emulating the data we expect when this method is used in real-life by installing cameras on the city corporation's garbage trucks. We reviewed various models ranging from conventional machine learning to complex deep learning algorithms and ultimately shortlisted three models: CNN, CNN-XGboost, and ResNet. These three models were then optimized for our problem, and then extensive testing was performed to determine the one that outperforms the rest. ResNet-34 emerged as a clear winner, with an accuracy of 87.8 % on the test data. Here, we'll do an in-depth study of the efficacy of these models on our problem statement.**

***Index Terms*—CNN, Residual Network, Machine Learning, XGboost, Road Inspection, Potholes, Cracks.**

I. Introduction

Road maintenance and detection of road surface defects, such as cracks and potholes is a prolonged process, yet a very important part of development. The government needs accurate information to improve road conditions and schedule maintenance at regular intervals. The quality of the road infrastructure is crucial for people who drive. Z. kamal stated in his article [1] that the death toll due to road accidents in the last half of a single year was 2,297 while the number of injured people crossed five thousand. Potholes have been proven to cause catastrophes, especially during rainy seasons, so the drivers need to be cautious. A driver finds it difficult to control the vehicles due to sudden potholes, bumps, and cracks. Moreover, most of the roads in our country are not built maintaining any standard structure. The highways connecting the major divisions of the country are also poorly maintained and in dire need of repair. Most of the drivers often being inexperienced find it harder to drive in these damaged roads causing even fatal head-on collision. Hence, road image analysis is a very important aspect of the analysis of the road condition. Detecting potholes would highly contribute in order to minimize impact, make the ride smooth, and allow the vehicles to issue warnings to the drivers to slow down or avoid them. It would also contribute to updating the city corporation about potholes and cracks and help them to repair the damaged roads at the earliest using the data. Studies and research are still relatively few. Traditional approaches use sensors and expensive types of equipment with high computational costs and complexity of data along with manual inspection which highly demands on specialist's knowledge and experience, which leads to a labor-intensive, time-consuming process. For fast and reliable detection, automatic detection is expected to develop instead of a subjective and slower human inspection and sensor-based approach. Firstly, there was no prior research done in Bangladesh in this field, hence there was no readily available data or survey that would support our objective. The road infrastructure of our country does not follow the standard protocols and the streets in the capital are so crowded that it is harder to find empty roads to collect images without any interference. To tackle these challenges, we conducted a survey from all age group of people who travel on a regular basis to identify the majorly affected roads of Bangladesh. According to our survey analysis, we collected and labeled the data on our own. Instead of making the system real-time, we thought of collecting video data by installing cameras on the city corporation's garbage trucks. This initiative would have a low installation cost and we will be able to collect more data as the trucks traverse around almost every corner of the city. In our study, we applied deep neural networks to extract information about the road and understand the environment surrounding the vehicle. The three models were made compatible with both image and video input. In this paper, we constructed a model capable of detecting potholes and cracks and determine is the road in a good or bad condition based on image processing. The aim of the authors was to formulate the best model for achieving the maximum accuracy and provide an accurate output.

II. Literature review

Pothole detection is demonstrated in many types of research outside our country using sensors and a few conventional algorithms. However, the roads of our country being very distinguishable from the roads abroad, the conventional approaches are not very useful. A few pieces of research are discussed here to gain perspective. Mednis et al. [2] proposed an automated system to detect potholes in real-time with little or no human interaction which will be based on Smartphones with Accelerometers and is implemented on the Android Operating System. The author focused on accelerometer data processing for pothole which is implemented on Android so that the system will be portable with less hardware complexity. The system running on a smart-phone should be able to detect while driving in different vehicles. Their proposed algorithm identifies the types of potholes along with using a bit advanced algorithm (Z-DIFF), which detected the fast changes in vertical acceleration data. Both the algorithm needs to know the Z-axis position. As the device is not using complex hardware resources standard deviation of vertical axis acceleration was used which was implemented in algorithm STDEV. After using the event detection algorithms on collected data, statistical analysis was made in terms of previously marked ground truths which were performed using EGNOS. 92% of all ground truth items were true positive 77% of all detected events were true hits. The test drive sessions detected irregularities 83-90% of potholes clusters. We can see 7% were not detected by any of the algorithms and 8% of the gapes escaped from the algorithms. Although they had an accuracy of 90% detection, the system is not enriched with self-calibration functionality. Moreover, using only smartphones limits the collection of data as it cannot handle huge amounts of data and it does not have such high computational power. Madli et al.[3] The paper proposes a solution to track potholes and bumps of roads within a low cost which gives timely alert to drivers about the preseumps. The author proposed a system that will identify potholes as well as bumps using ultrasonic sensors. Capturing all the factors the system will alert the drivers so that they can take precautionary steps to avoid accidents. Peripheral Interface Control (PIC 16F887A) processes all the sensor inputs and alerts the drivers. Moreover, the HC-SR04 ultrasonic sensor is used which captures the distance using the reflection of sound waves. Using all these components the systems divided it's architecture into 3 parts: microcontroller module, server module and a mobile application module. Their proposed model did not employ any machine learning or image processing approach to solve the problem thus they were not able to show any percentage of accuracy level. Moreover, the information stored on a mobile server database will not be effective for a huge amount of data and if there is any network error the system will fail to alert the drivers in real-time. Kawasaki et al. [4] The study approaches a system of shadow reduction to detect cracks with higher accuracy based on the percolation theory. Taking images from roads as input and then propose an algorithm for crack detection using a diagnostic analysis system so that cracks can be detected accurately and fast. The defected road images are collected through stereo camera and areas are captured by a stereo machine and U-V disparity. A pothole is detected by the inertial measurement unit (IMU) where a hole is identified by the data obtained from laser range finder (LRF) and then GPS mapping is done. The input images contain shadows of trees and other objects and due to camera conditions, a photo was taken time and weather also affect the image. All these reasons lead to the misdetection of cracks. If no noise is considered then the detection rate is 47.9% which is very low then the threshold value is changed and the problem is solved but the author said that still, noise appears which is again a problem. Ajit et al.[5] undertook a study to create a survey of Indian roads, to suggest the tactic to detect lanes, potholes and road sign, and their classification and to suggest automated driver guidance mechanisms. During this case, Color Segmentation and Shape Modeling with Thin Spline Transformation (TPS) is used with the nearest neighbor classifier for road sign detection and Classification. Jin at el. [6] proposes a histogram-based texture measure to extract the features of an image and identify potholes using a nonlinear support vector machine (SVM). The author of this paper mainly focuses on identifying potholes and cracks and distinguishing between them using partial differential equations (PDE) models. This method uses a nonlinear support vector machine to identify whether the area is a pothole or not. The model will not be effective for some complicated cases such as if the pothole is covered with mud or dust then the depression value becomes relatively flat with no grainy section. So, in case of the defect classes, the training model cannot correctly identify them and thus the number of training samples should increase for correct reorganization.

III. Implementation

A. Dataset

There was no open-source data for our use case and specifically nothing that would fit the road structure of Bangladesh. So the dataset was collected manually covering variations such as: the size of potholes, width and length of cracks, brightness and others while taking all corner cases into account. The first step of pre-processing was to extract frames from the videos that were recorded at 30fps. We extracted 2 frames per second to reduce redundancy in our training data by removing similar images, thus preventing our deep learning network model from overfitting.

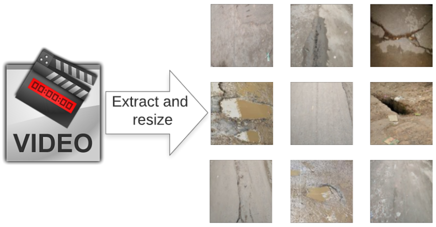

Fig. 1: Processed Images

All images were then processed using PIL (Python image library) and resized to 128x128, thus minimizing the information loss from the image. Since we will be training our models in the RGB space, we also converted single-channel also known as grayscale images into 3 channeled (RGB) images by placing the same channel for all the three channels for the resultant RGB image. In order to further diversify our data synthetically, we have also applied augmentation techniques. Then we labeled all images with relevant tags (Potholes, cracks and good roads) manually by making observations and judgments.

B. Convolution Neural Network

CNN (Convolution Neural Network) model includes several layers like the input layer, convolution layers, pooling layers, non-linear activation layer and others. There are many standard architectures proposed that help solving complex classification problems. One of those is a VGG16 architecture that inspires our model[7]. The input layer is fed 128x128 resized RGB images and 1x1 sized paddings are added. These two things are done to avoid overhead calculations while at the same time retain more information. Zero paddings add zero pixels (black frame) around the images that allow the kernel to cover more space. The network comprises of two primary modules:

The feature extraction module has 4 convolutional layer blocks with Rectified Linear Unit (ReLU)[8] as the activation function. We used a 3x3 kernel-size filter as it was the appropriate for our input data size. As for the number of filters, they are set in the power of 2. We tried out different combinations to find out the suitable numbers and ultimately set it as 64 and 128 accordingly. Given an input vector of dxd, we have set padding size p =1, filter size k =3, and stride s=1, the output size of each convolution layer is determined as:

$$S = \frac{(d - k + 2p)}{s} + 1 \tag{1}$$

$$S = d \tag{2}$$

The values of the feature map are then calculated using the following formula where m and n denote the rows and column of the resultant matrix:

$$Conv[m,n] = (d * k)[m,n] = \sum_i \sum_j k[i,j]d[m-i,n-j] \tag{3}$$

After passing this map through ReLU that transforms all the negative inputs to zero, maxpooling is applied to reduce overfitting and avoid slow convergence on validation data. The output size of this layer is given by S', Where pool size is 2, stride s =2 and padding is 1. So our formula becomes:

$$S' = \frac{(d - P) + 2p}{s} + 1 \tag{4}$$

$$S' = \frac{d}{2} + 1 \tag{5}$$

The classification module begins with flattening the final maxpooling layer into an one-dimensional array of length 10368, that data is then passed onto the fully connected layers. These Dense (fully connected) layers are stacked together with the last layer using Softmax activation function. Here, within the final layer, there are three labels: Potholes, Cracks and Good roads. Softmax function is used in this layer specifically to get probabilistic values for each label to classify the input images into the above-mentioned classes.

Preliminary training was performed on a smaller dataset with just 1315 images and later, more data was captured that increased the dataset to 2762 images. The validation accuracy improved by 22.84% in doing so. The comparison between two datasets and is given below:

TABLE I: CNN training results

	Old Dataset (1315)	New Dataset (2762)
Number of epochs	100	100
Training accuracy	91.6%	97.28%
Validation accuracy	63.89%	78.48%

The training process comprises of two parts: Forward pass and backward pass. During the forward pass, the training input data is fed to the network and predictions (label probabilities for each class) are computed. Using these predictions and the ground truths, we use categorical cross-entropy cost function to compute the model's loss. The loss function is defined as:

$$J(W) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y_i}) + (1 - y_i) \log(1 - \hat{y_i})] \tag{6}$$

Then, we start the backward propagation with a learning rate of 0.001, which is continuously optimized during the run using Adam optimizer. Adam, being an adaptive learning rate method, incorporates momentum[9] and moving averages($\hat{m}_t$,$\hat{v}_t$) while updating weights denoted with w_t using the following function:

$$w_t = w_t - 1\eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{7}$$

Below is the list of hyperparameters that have been optimized to achieve the highest performance.

TABLE II: CNN hyperparameters details

Parameter	Description	Value
K	Number of the kernels	64,128
f	Filter Size	(3,3)(3,3)
s	Stride size	Default
P	Pool size	(1,1)(1,1)
J(W)	Loss function	Categorical Cross-entropy
ReLU(x)	Activation function	ReLU
Softmax(x)	Activation function	Softmax
η	Learning Rate	0.001
b	Batch size	128

C. ResNet34

ResNet successfully deals with the vanishing gradient problem, by using skip connections to pass the information from past layers to the deeper layers. As per the basic building block of residula networks[10], we followed the below equation for our model:

$$Y = \mathcal{F}(x, \{W_i\}) + x \tag{8}$$

Where $\mathcal{F}(x, \{W_i\})$ is the residual mapping that we have learned. Generally, a residual function $\mathcal{F}(x)$ is defined as following, denoting the underlying mapping as $\mathcal{H}(x)$:

$$\mathcal{F}(x) = \mathcal{H}(x) - x \tag{9}$$

The addition operation between $\mathcal{F}$ and x, the inputs is performed by both shortcut and element-wise connection. Finally, a linear projection W_s has been included in the equation to ensure the dimensions are equal:

$$Y = \mathcal{F}(x, \{W_i\}) + W_s x \tag{10}$$

A brief representation of our model is given below:

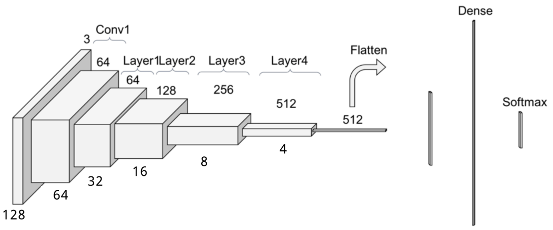

Fig. 2: ResNet34 Architecture of our model

In every block, convolution is followed by batch normalization [11] and ReLU activation, except the last convolution of the block. After these 4 layers are computed, two average pooling[12] layer operation is done where the kernel-size and stride are automatically adapted according to needs. Finally, a flattened feature map is passed through the linear layers using softmax function for the desired outcome.

To achieve the transfer learning, we first loaded the parameters of a ResNet model trained over the ImageNet[13] dataset. Fully connected layer of this pre-trained model was removed and two fully connected layers had been added followed by adaptive pooling layers. To reduce overfitting, we also added two dropout [14] layers that randomly drops information units in order to prevent the network to memorize a certain pattern and adapt accordingly. The last layer was accompanied by softmax activation function in order to classify the inputs.

D. CNN-XGboost

XGBoost being a member of the gradient boosting family [15] accords the same principals that have been followed in our previous models. We have consolidated extreme gradient boosting as a classifier along with a convolutional neural network extracting the features from our dataset. After training our respective CNN model, we extracted a feature vector from the penultimate dense layer with 4096 features. Similar feature vector from across our training data acts as a new training dataset that has to be fed into the XGboost classifier. Denoting the inputs extracted from this feature map as x_f we derived the output as :

$$\hat{y_i} = \sum_j W_j X_f \tag{11}$$

Another representation of the output is:

$$\hat{y_i} = \sum_{t=1}^{t} f_t(X_f) \tag{12}$$

Where $f_t(X_f)$ represents the function performing necessary calculation related to output scores of leaf nodes.

As Extreme gradient boosting algorithm cannot be optimized [16] using conventional methods, we have used an objective function to compute the loss during training:

$$L_x = \sum_{i=1}^{n} l(y_i, \hat{y_i} + f_t(X_f)) + \Omega(f_t) \tag{13}$$

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{14}$$

Here, γ is the loss reduction value that controls the split of a node and T is the number of leaves in the tree.

The list of manually adjusted hyperparameters for optimizing the model is given below:

TABLE III: XGboost hyperparameters details

Parameter	Description	Value
η	Learning rate	0.1
L_x	Loss function	Objective(multi:softmax)
NBR	Number of trees	120
MD	Maximum depth of tree	7
T	Number of leaf nodes	128
Softmax(x)	Activation function	Softmax

IV. Results and Analysis

We had 553 images in our test set that we fed to all three models and the results are discussed here. As we have a multi-class problem, so we had to modify the binary approach of calculating Precision and Recall score using the below equations:

$$Precision = \frac{1}{3}\sum_{c=1}^{3}\frac{C_c}{T_c} \tag{15}$$

$$Recall = \frac{1}{3}\sum_{c=1}^{3}\frac{C_c}{A_c} \tag{16}$$

Where C_c denotes correctly predicted images of class ”c”, T_c denotes total predicted images of class ”c” and A_c denotes number of actual images of class ”c”.

Below is the confusion matrix of our ResNet34 model that achieved the highest precision and recall scores:

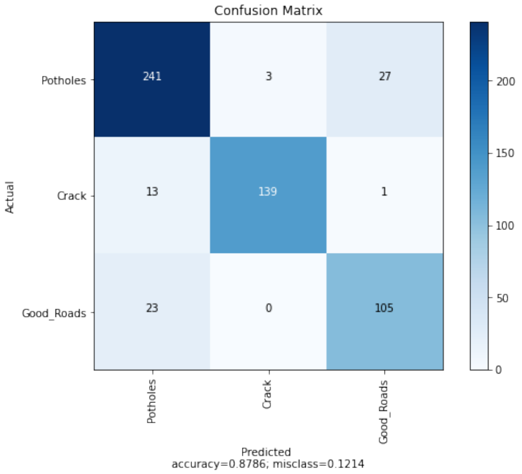

Fig. 3: Confusion matrix of ResNet34 model

The training accuracy, precision and recall scores of all three models are graphically compared below:

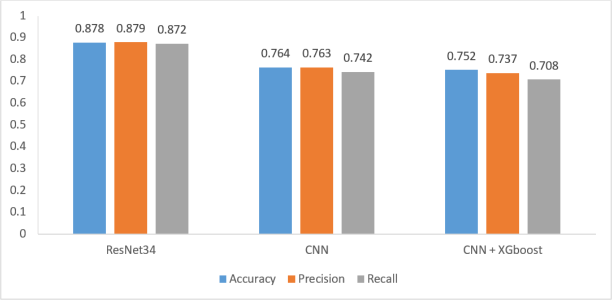

Fig. 4: Accuracy, Precision and Recall of three models

The highest accuracy of 87.86% was achieved by ResNet34 model, while the other two models, lagged behind with an accuracy of 76.49% (CNN) and 75.23% (XGboost). To further analyze the best model found, we use the algorithm proposed by Penghui Wang [17] and Amila Akagic [18], further referred to as model 1 and model 2 respectively. These two models are chosen, as they are the most recent works that target a similar problem statement and give better results when compared with other previous models. As per the accuracy, precision and recall given in figure 5.4, it is pretty evident that our model can detect potholes and cracks quite robustly even in varying conditions of climate, light and road type. On the other hand, both [17] and [18] try to deal only with the problem of pothole detection exclusively. So, to draw a more uniform comparison between these models with the proposed model, we just use the numbers for class potholes. As computed from the accuracy matrix shown in figure 5.2, the accuracy for our proposed model for class Potholes is 88.92%, while for model 1 and 2 it is claimed to be 86.7% and 82%. These results can be better visualized in the bar chart shown below.

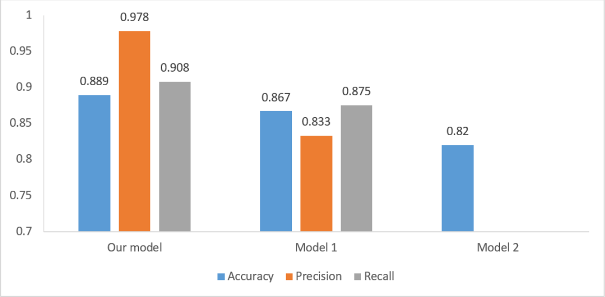

Fig. 5: Comparison with existing methods

Our model is able to differentiate between both pothole and cracks, along with good roads with an overall accuracy of 87.8%. Though the CNN model provided a 97.28% training accuracy, it failed to achieve a higher validation accuracy, thus clearly showing that it faces a bad variance problem. The XG-boost model, which used CNN as a feature extractor module also performed poorly on validation data with a precision score of 16.28% lower than the ResNet34 model. Thus, we conclude that the CNN feature extractor network, when trained from scratch, was getting overfitted due to limited training data available with us. That is where the prowess of ResNet34 model, trained using transfer learning approach, makes it the most viable and robust one for our problem statement.

V. Conclusion

This paper proposes an architecture based on the Residual Neural Network algorithms to detect the condition of roads. The model differentiates between good and bad roads and further detects potholes and cracks in order to speed up the process of road reconstruction and maintenance. After extensive shortlisting, we evaluated three different algorithms: Convolution neural network, Residual Network (34 layer) and Extreme Gradient Boosting algorithm to solve this problem. Moreover, the algorithms were assessed on our own collected dataset. All these models' performance were comparatively close, but the Residual Network (ResNet34) model gives the highest test accuracy around 87.8% for the given dataset. Although, currently we are only focusing on classifying potholes and cracks of roads, there are other factors which affect the roads such as type of surface, surface conditions, road spillage, shoulder drop-off and so on. In the future, we will work towards including these features as well, in our study. The features that would be focused are width and dimensions of the roads, surface conditions and shoulder drop-off as per our survey analysis. Additionally, we intend to detect roads covered with mud in the rainy seasons where detecting these features will be difficult, as mud can cause slippage of vehicles. Most importantly, our future plan is to execute and test our project in real world situation. As our topic is a broad and nearly unexplored one, we see huge scope of research to improve our project. We wish to further work on this project for the safety of our people as well as for the country.

Acknowledgement

Firstly we would like to express our sincere gratitude to our advisor Hossain Arif sir and our co-advisor Md. Saiful Islam sir for their continuous support and guidance. his kind support and advice in our work. He helped us whenever we needed help.

Finally to our parents without their throughout support it may not be possible. With their kind support and prayer, we are now on the verge of our graduation.

References

[1] Z. Kamal. (2017, 07) At a glance: Bangladesh road accident fatalities on the rise. [Online]. Available: https://www.thedailystar.net/world/south-asia/bangladesh/road-accident-in-bangladesh-2017-statistical-data-essay-at-a-glance-1427245

[2] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using android smartphones with accelerometers," 06 2011, pp. 1 – 6.

[3] R. Madli, S. Hebbar, P. Pattar, and P. GV, "Automatic detection and notification of potholes and humps on roads to aid drivers," *IEEE Sensors Journal*, vol. 15, pp. 1–1, 08 2015.

[4] Y. Kawasaki, K. Matsushima, and Z. Zhong, "Image-based pavement crack detection by percolation theory," 10 2017, pp. 1–6.

[5] A. Danti, J. Kulkarni, and P. 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, pp. 658–662, 01 2012.

[6] J. Lin and Y. Liu, "Potholes detection based on svm in the pavement distress image," *2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, pp. 544–547, 2010.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[8] A. F. Agarap, "Deep learning using rectified linear units (relu)," 03 2018.

[9] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," *30th International Conference on Machine Learning, ICML 2013*, pp. 1139–1147, 01 2013.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 06 2016, pp. 770–778.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 02 2015.

[12] B. Mcfee, J. Salamon, and J. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE Transactions on Audio Speech and Language Processing*, vol. 26, 08 2018.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 06 2014.

[15] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics*, vol. 7, p. 21, 12 2013.

[16] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," 08 2016, pp. 785–794.

[17] P. Wang, Y. Hu, Y. Dai, and M. Tian, "Asphalt pavement pothole detection and segmentation based on wavelet energy field," 2017.

[18] A. Akagic, E. Buza, and S. Omanovic, "Pothole detection: An efficient vision based method using rgb color space image segmentation," 05 2017.