Road and Pothole Recognition
A Convolution Neural Networks Application

Thai Nghiem
Department of Electrical and Computer Engineering
Rowan University
December 18 2018
Emails: nghiemt2@students.rowan.edu

***Abstract*—Pothole and cracks detection is critical in the process of maintaining and preserving roads and highways. This paper will discuss Convolution Neural Networks (CNN) and propose its usage in road and pothole recognition. The constructed model achieved a 93% of accuracy (for testing dataset) on detecting just potholes, and 77% of accuracy (also for testing dataset) on differentiating between potholes, cracks, and good road.**

I. Introduction and Motivation

Roads and highways are the necessities of our current transportation model personal vehicles carrying individual to their destinations. Poor road conditions are more than just a public nuisance, as it causes discomfort to drivers, passengers, as well as damage to vehicles and tragic accidents. Roads assessment and repair is a complicated business, which involve interactions between city residents, public works departments (PWDs), and private contractors. A great deal of efforts have been given into identifying and recognizing bad roads and potholes. By automating this process, lots of time and money could be saved and more effort could be focus on fixing and repairing the damaged roads. A solution to this problem would involve mounting a camera on the vehicle, feeding data to a central computer that regularly checked the camera images (while driving) and identified properly potholes, cracks and good roads. The low cost of replacement cameras, and relatively easy scanning process, makes this one of the best option for the automating bad-road identification problem.

In this paper, the first part will give a general understanding of CNNs and their applications in image recognition. The second part of this paper will focus on the use of CNNs for pothole classification process, and demonstrate the feasibility of such an approach.

II. Background

Neural Networks make use of "neurons" to calculate an output from an input. These "neurons" can sometimes be called nodes, and each node is an equation that takes the weighted inputs from previous nodes or from program inputs to evaluate the output. Each weights from the connections between "neurons" need to be trained in order to make a good decision. Convolutional Neural Networks are special type of deep feed forward neural networks that are used mostly for image recognition as they makes use of spatial relation of data. [1]

A. Layers

Since CNNs are deep neural networks, they consist of multiple layers that stack up to produce exceedingly complex equations. There are 4 most common layers in a CNN: convolutional layer, pooling layer, activation layer, and fully-connected layer. All of these layers are discussed in detail below.

1) Convolution Layer: Convolutional layer is the layer that set CNN apart from other deep neural networks architecture. Instead of using simpler addition and multiplication operations on each of the inputs, this layer convolves one or more filters (matrix windows) across the input layer, obtaining information only from locally-spaced inputs. This process can be seen in Figure 1.

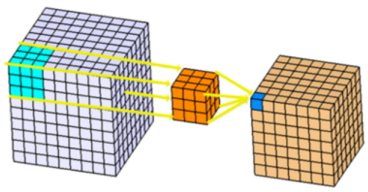

Fig. 1. Convolution Operation [2]

The filter (typically a 5x5x3 matrix) convolution operation generates a feature map for each layer, which gives information on certain characteristics of the image, such as edges and corners. The more convolution layers are used, the more sharpened the features will be. There are a few parameters that are in this layer, which can be used to tune the model, such as the number of filters, filter shape (width, length, and depth), and stride (step size of the filter movement). Stride needs to satisfy the Equation 1 represented below.

$$\frac{N - FilterSize + 2 * ZeroPadding}{Stride + 1} = WholeNumber \tag{1}$$

where N is the size of the input matrix. The formula for calculating the ouput size for any ginven convolution layer is:

$$O = \frac{W - filterSize + 2 * ZeroPadding}{Stride} \tag{2}$$

where, O is the output height/length and W is the input height/length.

2) Pooling layer: In CNNs, pooling layers are commonly used to reduce the computational overhead, as well as to avoid over-fitting. This is because pooling layers reduce the dimensional of the data, meaning there are less weights to train and calculate. There are two types of pooling: average pooling and max pooling. In this application, average pooling is utilized, which operation can be seen in Figure 2.

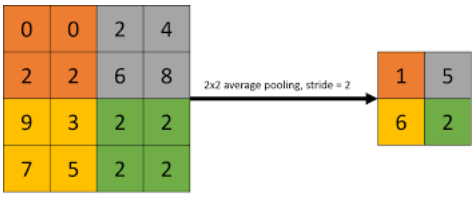

Fig. 2. Average Pooling Operation [3]

One drawback of using pooling layer is that it tends to take away some important information of the input layer. This could be detrimental to the process of extracting features. Therefore, poolings are generally not used in every single layer and can be replaced by Dropouts, which is an effective method of mitigating over-fitting. In the final model, only one pooling layer is used for 2 convolution layers.

3) Activation layer: Similar to the pooling layer, activation layer can also be used to avoid over-fitting of the model. However, more importantly, activation layer is responsible for increasing the nonlinear properties of the decision function and of the overall network. Two common types of activation are used in this application are ReLU, which stands for Rectified Linear Unit, and softmax. ReLU operation can be seen in Figure 3 below.

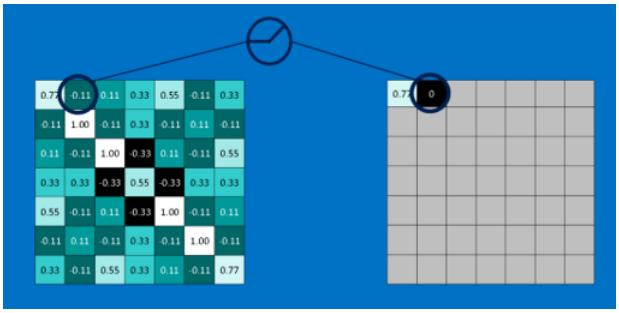

Fig. 3. Rectified Linear Unit Operation [4]

The output size of a ReLU layer is identical to its input size. As it can be seen from the picture above, this activation layer takes the output of a certain layer and set any negative values to zero. Basically, the equation governing the ReLu activation layer can be seen as follow.

$$f_x = max(0, x) \tag{3}$$

4) Fully Connected layer: The feature maps created by the above layers are then analyzed using a set of fully-connected layers. Before these maps are passed into the final set of layers, all of data are flatten into vectors. And just like any other deep neural networks' hidden layers, the fully-connected layer calculates the category that the original image falls into. This process can be seen in Figure 4.

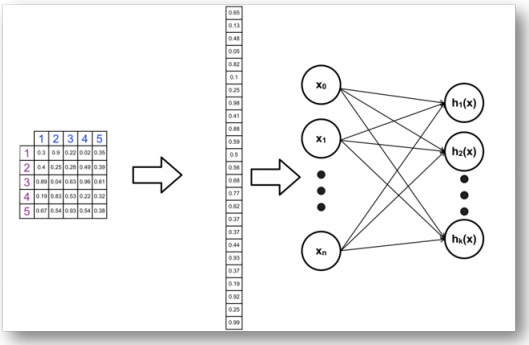

Fig. 4. Fully-connected Layer [5]

B. Applications of CNNs

As mentioned above, CNNs are mostly used for Image Recognition, even if the image is rotated or translated. This neural networks architecture can also be used for Speech Recognition, or natural language processing, and Recommender Systems (rating & preferences). Other than the above, CNNs are used in determining whether a defendant is a flight risk based on criminal history or predicting earthquakes based on seismic data [6].

C. Reason for choosing a CNN

From the above brief description about CNNs architecture and the theory behind each layer, it can be observed that CNNs solve the most two critical problem for image recognition: the "curse of dimensionality" and the "spatial relation of data". For the pothole detection problem, a relatively large dataset was provided, which consists of about 6000 RGB images (1280x720x3). This mean that there is a total of 165,888,000 possible features that a neural networks have to process in a restricted amount of time. Also, as regular ANNs do not account for spatial relation of data, which means that input pixels that are far apart are treated same way as pixels that are next to each other, a great deal of important information will be lost when detecting potholes and cracks. In short, the CNN is chosen for its use of spatial relation in data and its efficiency in dealing with large number of data.

III. Methodology

A. Problem Statement

For the purpose of automating the process of identifying and recognizing bad roads and potholes, a problem statement is defined by the author of this application note: "Differentiate between a road that is in a good condition versus a road that

has pothole(s) and/or cracks using a 1280x720 RGB image". An example of the image can be seen in Figure 5.

Fig. 5. Example of an image to be classified

From this definition, there are two types of problems that will be addressed in this report: the 'pothole detection/recognition' problem, and the 'good road, cracks and pothole classification' problem.

B. Data Analysis & Augmentation

The data-set that was obtained for solving the above problem statement was provided by the NJDOT Clinic, which is lead by Dr. Bouaynaya and Dr. Farzan Kazemi. The original description of the data-set provided can be seen as follows:

dataset1-01: Only good road, no other attributes
dataset1-02: All bad road (cracks and potholes)
dataset1-03: Only cracks (good road and bad road)
dataset1-04: Only potholes (good road and bad road)

From the above description, it was decided that for the 'pothole detection/recognition' problem, only 'dataset1-01' and 'dataset1-04' can be used as 'good roads' class and 'pothole roads' class, respectively. The first data-set has a total of 4000 RGB images, and the second data-set has a total of 2000 RGB images, each with a height of 720 pixels and a width of 1280 pixels. Since these data-sets are unbalanced, and 6000 images for a classification problem is relatively small, data augmentation and image preprocessing need to be applied to create more data. Two processing technique are used for this application: random rotation and blurring. Also, all the images are resized to 128 px height and 128 px width for faster computing time. The processed images can be seen in Figure 6 below.

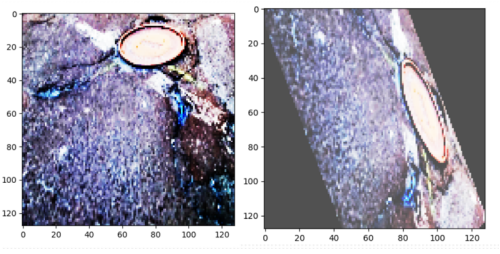

Fig. 6. Example of an image that was applied image preprocessing

For the 'good road, cracks and pothole classification' problem, the same 2 data-sets are used and is added with 'dataset1-03' for the third class: 'cracks'. The same image processing methods were used on these data-sets.

As a result, each difference classes mentioned above (good road, potholes, cracks) has a total of 10,000 images. Hence, for the 'pothole detection/recognition' problem, 20,000 total images were used, and for the 'good road, cracks and pothole classification' problem, 30,000 images were used.

C. Standards & Constraints

1) Constraints: Some constraints were set on this application research due to the limits of available resources. First of all, the time it takes to train all the CNN models that are designed in this project must fit into the schedule decided by the instructor, which is about 3 weeks. Second of all, the size and complexity of the data and the model were limited by the available system memory, CPU power and GPU power. Lastly, a constraint is also put on the data-set, as there are only 6,000 images, which is relatively small for a classification problem.

2) Standards: Four industry standards was followed by this project. First, Training and Testing Data divided into 70% and 30% of all data. That means that for the 'pothole detection/recognition' problem, 14,000 images were used as Training Data and 6,000 as Testing Data. For the 'good road, cracks and pothole classification' problem, 21,000 images were as Training Data and 9,000 as Testing Data. Second, Early Stopping was utilized to mitigate over-fitting of the model. Third, Keras wrapper for TensorFlow was used to create the CNNs. Several methods are used across the projects are: fit, evaluate, Sequential, etc. Last, all the data is taken from the same sensor and same orientation.

D. Models for Pothole Recognition

Several CNN models was created and experimented throughout this research application. Most of the these models is constructed of convolutional, average pooling, ReLu, and fully-connected layers. Only two most notable models will be presented. The first model is a 'naive' one, while the second model is the model that was improved based on previous naive models.

1) First Model: The architecture of the first model can be seen in Figure 7.

The first model consists of 3 convolutional layers, each with a sub-sample (average pooling), and 2 hidden fully-connected layers, with the utilization of dropouts and 'relu'. Furthermore, the first model add one more step to the data augmentation: converting from RGB images to gray-scale images. The result can be seen in Figure 8.

Results of the first model after training will be discussed in Section IV.

2) Second Model: As mentioned above, model 2 is an improvement in performance of the first model based on the observation that model 1 tends to over-fit. The second model is constructed of 2 convolution layer (down from 3 layers), with only 1 sub-sampler, and 1 hidden fully-connected layer (down from 2 layers). In addition to dropouts and 'relu', model 2 also

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 31, 31, 16)	1040
activation_1 (Activation)	(None, 31, 31, 16)	0
conv2d_2 (Conv2D)	(None, 16, 16, 32)	12832
activation_2 (Activation)	(None, 16, 16, 32)	0
conv2d_3 (Conv2D)	(None, 8, 8, 64)	51264
global_average_pooling2d_1 (	(None, 64)	0
dropout_1 (Dropout)	(None, 64)	0
activation_3 (Activation)	(None, 64)	0
dense_1 (Dense)	(None, 1024)	66560
dropout_2 (Dropout)	(None, 1024)	0
activation_4 (Activation)	(None, 1024)	0
dense_2 (Dense)	(None, 512)	524800
dropout_3 (Dropout)	(None, 512)	0
activation_5 (Activation)	(None, 512)	0
dense_3 (Dense)	(None, 2)	1026
activation_6 (Activation)	(None, 2)	0

Total params: 657,522
Trainable params: 657,522
Non-trainable params: 0

Fig. 7. Architecture of the First model

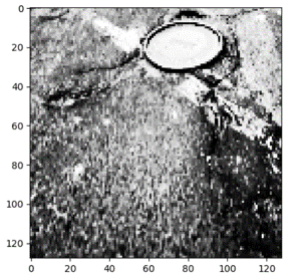

Fig. 8. Data input of the First model

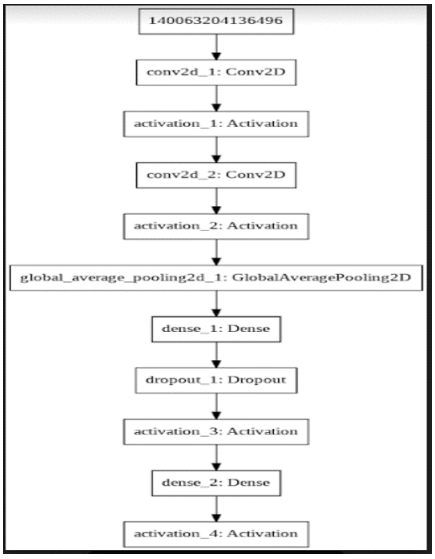

Fig. 9. Architecture of the Second model

Fig. 10. Data input of the classification model

utilized Early Stopping to prevent over-fitting. The architecture of the first model can be seen in Figure 9.

In the figure above, the first square box represents the input layer, and 'global-average-pooling2d' layer acts as a flatten layer. Model 2 does not convert RGB images to gray-scale to maintain the information given by color channels. Hence, the inputs of this model is Figure 6 above. Training results and activation layers are discussed in Section IV.

E. Models for Road, Cracks, and Pothole Classification

This model of CNN uses the same architecture as model 2 presented above. That is, it also has 2 convolution layers and 1 hidden layer. However, this model solves a different problem: classify between 3 classes (good road, cracks, and potholes). Hence, this model is added a third class named 'cracks'. An example image that belongs to 'cracks' class is shown in Figure 10 below.

Results and activation layers is shown in the next section.

IV. Results and Discussion

A. Results

For the first model that solve the 'pothole recognition' problem, the accuracy achieved by passing testing data through the network is 84.4%. However, when training the network using the the training data-set, the accuracy comes out to be roughly 96%. This represents over-fitting. As it can been seen in Figure 7 of the model 1 summary, the total parameters that this model has to train is 657,522. This is a relatively large number of parameters and this would cause the model a lot of time for the model to be trained and lead the model to over-fit.

Therefore, an improvement was made to this model by reducing the numbers of layers, which effectively reduced the number of parameters. The number of parameters in the second model is only 33,842. The accuracy of the second model for solving the 'pothole recognition' problem is 92.6% on the testing data set, and 95% on the training data-set. This is a significant improvement and is the final result of this research proposal. An effort to visualize the activation layers of the model is made and represented below, since it is beneficial to the understanding of the model. In order to do this, a test image (figure 11) is passed through the trained model.

This is the picture of a pothole, and the results of the 2 convolution layers are shown in Figure 12 and 13.

As it can be seen, each out layers give us a different characteristic of the image. The different in color highlights us about the features of the pothole, and the further layer goes down, the closer each feature is presented.

It took the CPU and GPU in the Dell Inspiron 7535 40 minutes to train this network, and the best model was trained for 121 epochs with the stop condition set on value loss (with a patience of 15). Misclassification of this second model is also looked at closely and is discussed in Section V below.

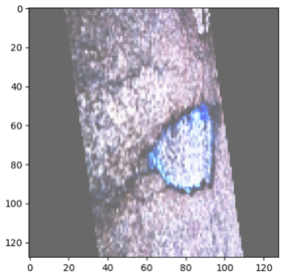

Fig. 11. Second model test image

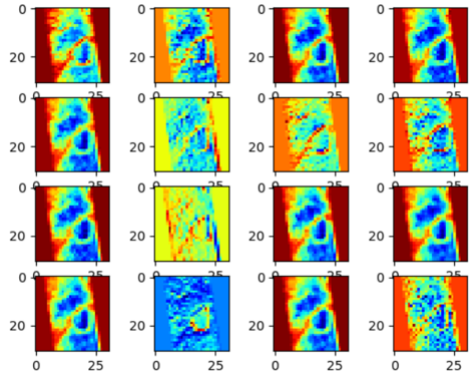

Fig. 12. Output of the first convolution layer in the second model

For the 'Road, Cracks, and Pothole Classification' problem, the accuracy comes out to be 77% on the testing data-set. This low in accuracy is due to one limitation in the given data-set, which will also be addressed in the next section.

B. Misclassification & Why?

For the 'pothole recognition' problem, in the testing data-set, there are a total of 444 miss classified images, in which 190 is for the first class, and 254 is for the second class. This is a slightly unbalanced result, and it favors the first class, which is the 'good road' class. The reason for such unbalance is because there are as double the original images for the first class as for the second class, so more 'good roads' are correctly identified. To understand even more about how these images are misclassified, an collections of missclassifed images are presented below in Figure 14 and Figure 15.

From the 2 figures, it can be clearly seen why the model get confused between the 2 classes. Figure 14 shows the images of the 'good road' class that was classified as 'pothole road'. While some of the images are indeed showing roads in a good condition, a significant amount of these images show roads are in bad condition. One obvious one is the image on the bottom left corner of figure 14. It can be seen that there is indeed a pothole in the image. Also, several images in figure 14 shows roads that have potholes being covered, or roads with lots of cracks in them. This confusion in the way the original data is labeled has definitely lead to a confusion in the CNN model.

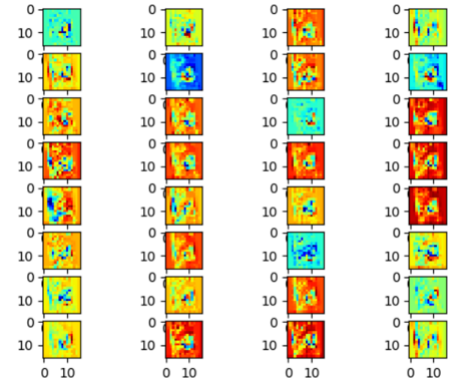

Fig. 13. Output of the second convolution layer in the second model

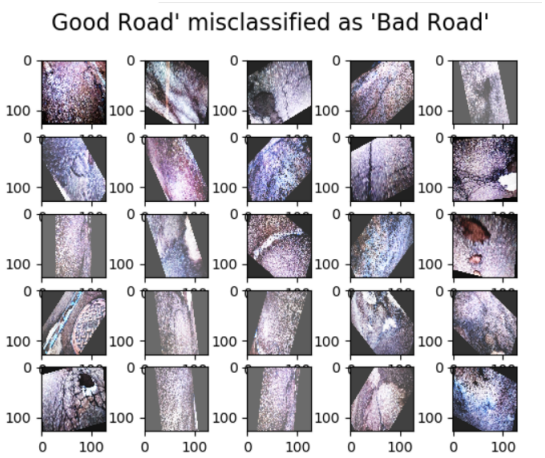

Fig. 14. 'Good road' images that are classified as 'pothole road'

On the other hand, figure 15 shows the images of the 'pothole road' class that was classified as 'good road'. With our naked eyes, it can be seen that many of these images are indeed images showing a road in a very good condition (no cracks or pothole). The reason actually lies in the data augmentation methods. By rotating the images to get more data, some information that lies in the corner of the images are lost. Some of the information are insignificant and can be ignored, but some actually shows part of a pothole. One such example can be seen in Figure 16.

Therefore, by cutting these corners out to create more data, we have accidentally tricked neural networks to misclassify.

The same types of problem are also resides int the 'Road, Cracks, and Pothole Classification' solution, but worst since some images have both cracks and pothole in them but only 1 class was labeled. One example of such confusion is in Figure 17.

In short, the three problems - unbalanced, miss-labeled, and data loss in augmentation process - are the reason why none of the model achieve higher than 95% accuracy.

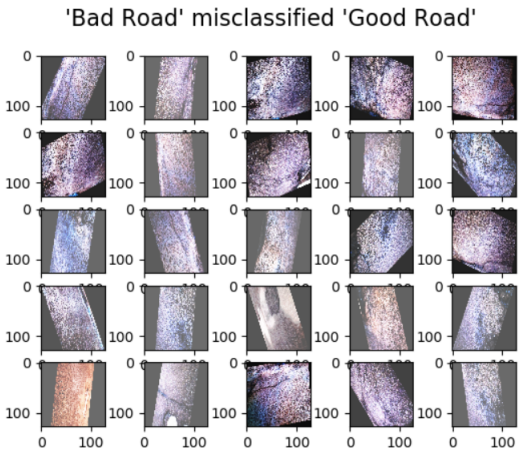

Fig. 15. 'Pothole road' images that are classified as 'good road'

Fig. 16. Image of a pothole lies in the top left corner

C. Possible Ventures for Improvement

With a 92.6% accuracy for the testing data-set, model 2 is a strong candidate for solving the automating problem. Nonetheless, in order to improve the performance, a much better effort need to go into pre-processing the data.

For the 'miss-labeling' problem, one solution is to completely disregard images that have both potholes and cracks, or images that are in the corners. However, a large amount of original data will be lost.

For the 'data lost' problem, one solution is to better the data augmentation process by adding one extra step of zooming out on the images after rotating, so that the information in the corners won't be lost.

More notably, one possible solution that could solve both of the above problem is to have a different deep neural networks architecture that could identify and classify multiple classes in the same picture. Mask R-CNN [7] is a good architecture that can satisfy the stated requirement. Figure 17 shows how the Mask R-CNN correctly classify multiple objects in one single picture.

And lastly, to solve the unbalance problem, more data need to collected and add to the 'pothole' and 'cracks' class.

V. Conclusions

This project confirms the idea that CNNs can perform very well when used for image classification. The final model was able to achieve 92.6% of accuracy on detecting potholes on the road from a 1280x720 RGB image, without over-fitting the training data-set. Three problems to the network were identified and the solutions to these problems are fully addressed. The findings in this project can help automate the process of identifying and recognizing bad roads and potholes and they can be applied in future projects using deep neural networks.

Fig. 17. Image showing both pothole and cracks, but was only labeled as 'cracks'

Fig. 18. Mask R-CNN correctly classify multiple objects in one single picture

VI. Acknowledgement

The author would like to give thanks to Dr. Robi Polikar, Prof. Muhammad Umer, and Dr. Bouaynaya from Rowan University. Without their inputs and guidance, the project would not be possible.

VII. Appendix

Trained models and python code can be found at https://github.com/nghiemthai1/Pothole-Detection-using-Machine-Learning

References

[1] Convolutional Neural Network. Wikipedia, Wikimedia Foundation, 17 Dec. 2018, en.wikipedia.org/wiki/Convolutional-neural-network.

[2] [ML 20] Convolution Neural Network Part 3. SEPTENI TECHNOLOGY Developer's Blog, labs.septeni-technology.jp/technote/ml-20-convolution-neural-network-part-3/.

[3] 10154029720402034. The Deep Learning(.Ai) Dictionary Towards Data Science. Towards Data Science, Towards Data Science, 6 Apr. 2018, towardsdatascience.com/the-deep-learning-ai-dictionary-ade421df39e4.

[4] Brandon Rohrer, brohrer.github.io/how-convolutional-neural-networks-work.html.

[5] Ung Dung Convolutional Neural Network Trong Bi Ton Phn Loi nh. Viblo, $viblo.asia/p/ung-dung-convolutional-neural-network-trong-bai-toan-phan-loai-anh-4dbZNg8ylYM$.

[6] B. Rouet-Leduc, C. Hulbert, N. Lubbers, K. Barros, C. Humphreys and P. Johnson, "Machine Learning Predicts Laboratory Earthquakes", Geophysical Research Letters, vol. 44, no. 18, pp. 9276-9282, 2017

[7] Matterport. (2018, October 21). Matterport/Mask-RCNN. Retrieved from https://github.com/matterport/Mask-RCNN