Deep on Edge: Opportunistic Road Damage Detection with Official City Vehicles

Makoto Kawano, Takuro Yonezawa and Jin Nakazawa

Graduate School of Media and Governance
Keio University
Email: {makora, takuro, jin}@ht.sfc.keio.ac.jp

***Abstract*—How can we inspect city conditions at low costs? City infrastructures, such as roads, are elements of great importance in urban lives. Roads require constant inspection and repair due to deterioration, but it is expensive to do so with manual labor. Therefore, these works should be done automatically so that the cost of inspecting or repairing becomes cheap. While there are several works to address these road issues, our study focuses on official city vehicles, especially garbage trucks, to detect damaged lane markings (lines) which is the simplest case of road deterioration. Since our proposed system is implemented on an edge computer, it is easy to attach our system to vehicles. In addition, our system utilizes a camera, and since garbage trucks almost run through the entire area of a city every day, we can constantly obtain road images covering wide areas. Our model, which we call Deep on Edge (DoE), is a deep convolutional neural network which detects damaged lines from images. In our experiments, to evaluate our system, we first compared the accuracy of line damage detection of DoE with other baseline methods. Our results show that DoE outperforms previous approaches. Then, we investigate whether our system can detect the line damage on a running car. With this demonstration, we show that our system would be useful in practice.**

Keywords–Smart City; Deep Learning; Edge Computing; Image Recognition;

I. INTRODUCTION

The road is one of the most important infrastructures of a city in planning and development. For instance, people usually use them for going somewhere or for planning land utilization to enrich their livelihoods. However, many roads need repairing since most of them are built in periods of rapid economic growth and have been deteriorating since. Thus, to inspect their condition for road repair, the city administration needs to employ people for constant inspection. Yet manual road inspection is expensive and takes a lot of time; for instance, in order to detect the damage or blur of road markings, people have to check by eye, whose ability has certain limits. In addition, in certain regions such as Japan, public funds for road inspection have been reduced due to current societal conditions. In short, manual road inspection and repair is not enough for sufficient maintenance.

Most previous work has therefore focused on making the cost of road inspection cheaper to increase sustainability. Some works have focused on road flatteness [1]–[3], potholes [4] [5], and cracks [6]. In contrast, we aim to detect the damage or blur of white lines. To our knowledge, only our previous work addresses this problem [7]. Detecting the damage of white lines is difficult to do using smartphone accelerometers such as [1]–[3]. Thus, we use a camera to take pictures/videos [6] [7]. If we use participatory sensing [8] as well as [6] and collect the images, however, the cost issue still remains due to the cost of platform introduction and labor.

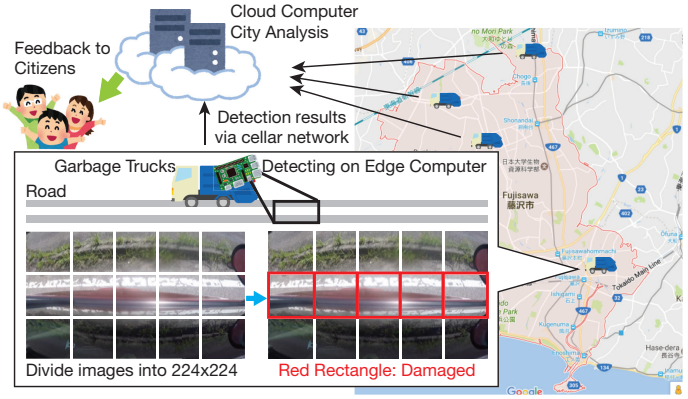

Figure 1. Our system overview. Each city vehicle running in the city detects white line damage. The cloud computer aggregates the results from them and monitors the city.

To tackle this issue, we focus on city vehicles, especially garbage trucks. Since garbage trucks run their services every day and cover a whole area of the city, if the garbage truck equips a camera and takes pictures of roads, we can obtain road images from the whole area. Furthermore, we do not have to pay additional costs for labor or facilities. However, the number of running garbage trucks is so large (e.g., hundreds of trucks) that it is troublesome to storage and manage image data in a centralized way. Simultaneously, if we upload an image every time a camera takes pictures, it would take great communication costs and bandwidth. In summary, our goal is proposing a system that can be attached to garbage trucks and detect white line damage on the spot.

In order to achieve our goal, we introduce Deep on Edge (DoE), which integrates edge computing and deep neural networks. The overview of our system is depicted in Figure 1. DoE consists of an edge computer (e.g., Raspberry Pi 3) with a camera to be attached to garbage trucks. When DoE detects line damage, the results are reported and sent to cloud computing. Then, we use those reports and understand city condition. We treat the task of line damage detection as a classification problem. We train a convolutional neural network (CNN), a type of deep neural network, on labeled images on a GPU server. At inference time, DoE is loaded on an edge computer and outputs a discrete probability distribution,

 ISBN: 978-1-61208-569-2

assigning each image a likelihood that the white line in the image is damaged. There are some constraints to using DoE on edge computers due to restricted performance, while on the other hand we do not have to consider the number of parameters or inference speed when we use DoE on a high-performance computer. So to use DoE on edge computers, we design the CNN architecture to be as small as possible but to keep the accuracy high. To evaluate DoE, we compared it with baselines on the line damage detection task. As a result, DoE outperforms baselines on this task while reducing the number of parameters. Simultaneously, we visualize DoE activation so that we can understand how it has learned to detect line damage.

The contributions of the paper are summarized as follows:

- Propose the system, called Deep on Edge, which integrates city vehicles, edge computing, and deep convolutional neural networks.
- Pose lane marking (line) damage detection as a classification problem and proposing our model which outperforms other baselines on this problem.
- Discuss the ability of the neural networks through activation visualization to design network architectures appropriate for practical use.

The paper is organized as follows. In the next section, we describe the white line dataset which we use in this paper. In Section III, we present our system which performs line damage detection. Then, we explain in detail the experiments and compare the results with those obtained in a previous research in Section IV. We discuss the experiment result and how DoE learned to detect line damage in Section V. Finally, we introduce related works to compare with our work in Section VI and conclude the paper in Section VII.

II. Summary of Our Lane Marking Dataset

In this paper, for detecting road damage, we focus on the damage or blur of white lines, which we assume is the most common type of lane marking. To collect line images, we attached a normal camera, which can film by 60 fps, on a side of a passenger car so that the camera always films the line. Then, we drove the car within 50km/h for four days from March 30th, 2016 to April 2nd, 2016 in daytime. Note that it was sunny days. While we got videos in which each frame is 1024×768 pixels after filming, we randomly cropped frames into 224×224 pixels. This cropping was done for reducing the training time until the model convergence and allowing the model focus on the line damage. One participant annotated those cropped images with three kind of labels; damaged line, undamaged line, and no line. After the preprocessing described above, we obtained 43000 images of lines. At our experiments, we divide the dataset to 35000:8902. The examples of our dataset are shown in Figure 2 and described in detail in Table I.

III. Deep on Edge System

We pose the task of line damage inspection as a classification problem. For this, we use a dataset of images of lines with three kinds of labels described in the previous section. The input to DoE are image pixels and the target output is a one-hot vector encoding those labels. Given an image, the output of this model is a probability distribution describing the extent of road damage. The advantage of outputting a probability distribution

Figure 2. Dataset samples. Each image is 224×224 pixels by random clipping from video frames, respectively. The images at top row show damaged line, the images at mid row undamaged and at bottom row no line.

TABLE I. OUR DATASET WHICH WE COLLECTED, PREPROCESSED AND ANNOTATED.

Class	Type	Undamaged	Damaged	No line	Total
Binary	Train	10696	14304	–	25000
	Test	3829	5073	–	8902
Trinary	Train	15445	11568	7987	35000
	Test	3932	2957	2013	8902

is that this gives the model the ability to give specific scores to a line image, taking out the necessity of a human expert to give specific ratings.

A. Our Model

In order to detect line damage from images, we adopt a convolutional neural network, which is a special type of feedforward neural network or multi-layer perceptron and works well with two-dimensional images. We design our CNN by referring to the VGG16 architecture [9]. VGG16 is one of the major CNN architectures which was used to win the ILSVR competition in 2014, although it has been outperformed by great advances such as Inception [10] and ResNet [11] [12]. VGG16 only uses convolutional layers with 3×3 kernels and pooling layers with 2×2 kernels. This feature is very significant for DoE since the size of the model is required to become as small as possible to work on edge computers. Given an input image$\mathbf{X}$ of width w, height h and c color channels (usually RGB channels) represented as $\mathbf{X} \in \mathbb{R}^{w \times h \times c}$ at each convolutional layer, it is convolved with d sets of local kernels $\mathbf{W} \in \mathbb{R}^{w \times h \times c \times d}$ and bias $\mathbf{b} \in \mathbb{R}^d$ is added:

$$h = \phi(\mathbf{W} * \mathbf{X} + \mathbf{b}), \tag{1}$$

where $*$ denotes a convolution operation and ϕ is a non-linear function that we use the rectified linear unit (ReLU, $\max\{0, x\}$). Max-pooling, a form of non-linear downsampling, is applied to the output of the convolution. Max-pooling

partitions the input into a set of non-overlapping rectangles by the kernels and outputs the maximum value in each sub-region respectively. This operation is very useful because it reduces the dimensionality of a high-dimensional (high-resolution) output of the convolutional layer and summarizes the activations of neighborhood features so that model becomes robust to local perturbations. Since our input images are filmed from a driven car, the location of lines in the image are not fixed. DoE is built by several alternating convolution layers and max-pooling layers.

In VGG16, the output after some convolution layers and max-pooling layers is flattened for the input of to the following layers, which are fully-connected. If the shape of the output of convolution is $\mathbb{R}^{w \times h \times c}$ and the output dimension of the next fully-connected layer is d, the number of parameters in that FC layer becomes $w * h * c * d$. This is a problem when the size of the input image is large, since the larger the image size is, the larger the number of parameters becomes. To avoid the increase in number of parameters, we use global average pooling [13] instead of flattening. Applying global average pooling allows the number of parameters in the FC layer $c * d$ to be independent of the input image size. At the last layer of DoE, the output is a probability distribution over the possible conditions of the road.

The model of the DoE architecture which we used in our experiment is depicted in Figure 3.

B. Implementation for Practical Use

While DoE is trained with the road images of size 224×224, the size of images from a video camera is much bigger than that. Although our DoE model can take any image resolution, our preliminary experiment showed that DoE cannot detect the line damage accurately at any resolution. In order to tackle this issue and use DoE for practical use, we implement a module that divides the input image into 224×224 sub-regions and reshapes these sub-regions to $X \in \mathbb{R}^{n \times 224 \times 224}$ where n is the number of sub-regions. Even if n is very large, DoE is able to process it all at once. For instance, if the size of an input image is 1280×720, the number of sub-regions becomes $(1280/224) * (720/224) \approx 15$ and the input to model $X \in \mathbb{R}^{15 \times 224 \times 224}$. Although our module crops out the top, bottom, and right sides of the image, this is not a problem because of two reasons: (a) the top and bottom sides of the image usually does not contain the road (b) the road contained on right side is contained in the next input image. Figure 3 also shows this module.

IV. Experiment

In this section, we show two kinds of experiments. First, we compare DoE with baselines which are used in previous works to evaluate DoE. Then, we examine whether DoE is fit for practical use.

A. Accuracy Comparison

In order to evaluate DoE and its architecture, we compare it with previous work [6] [7]. Although Maeda et.al [6] classify the degree of road condition into three types: "smooth (no-damaged)", "need repair" and "not need repair", its actual classifications are difficult to distinguish, as different outputs are produced from visually similar images. To make this problem more interpretable, we simplify this task as binary classification problem: the road which is photographed in given images is whether damaged or not. Therefore, we used the dataset we use consists of images labeled "damaged" and "undamaged" in Table I.

TABLE II. ACCURACY COMPARISON ON THE LINE DAMAGE BINARY CLASSIFICATION TASK. tHE NUMBER OF PARAMETERS IS THE SUM OF WEIGHTS AND BIAS.

Method	Acc.	AUC	Recall	Pre.	F1	Params.
Linear SVM	82.4	0.82	0.87	0.83	0.85	–
Random Forest [7]	84.0	0.83	0.91	0.83	0.87	–
AlexNet [14]	92.5	0.9833	0.92	0.92	0.92	58000K
AlexNet–(d) [6]	92.5	0.9845	0.92	0.92	0.92	1680K
AlexNet–(e) [6]	92.7	0.9859	0.93	0.93	0.93	913K
DoE (ours)	**94.1**	**0.9894**	**0.94**	**0.94**	**0.94**	**18K**

TABLE III. CONFUSION MATRIX OF DOE.

Number of Parameters 18171		Prediction			Recall
		Undamaged	Damaged	No line	
Ground Truth	Undamaged	2795	162	0	0.945
	Damaged	196	3734	2	0.950
	No line	0	1	2012	1.00
Precision		0.945	0.950	0.999	0.980

As baselines, we adopt two kinds of methods. The first method tests classic machine learning algorithms: a support vector machine classifier (SVM) that can achieve good performance at binary classification, and a random forest which can detect the line damage [7] as well as our work. The second method is a deep neural network. We choose the AlexNet which is proposed in [14] and won the ILSVR competition in 2012, and has been used quite actively since [6]. Further, since the aim of our study is road detection on an edge computer, the smaller model is desirable and we also examine the alternative models that are proposed in [6]: AlexNet–(d) and AlexNet–(e).

Before training DoE, we initialize the weights of DoE with random values and use the Adam [15] stochastic gradient descent algorithm with a learning rate of 0.0005, a momentum of 0.9 and a batch size of 32. Meanwhile, those of baseline networks use respective values of 0.0001, 0.9 and 100. We then trained models with early stopping, which is a training procedure that stops training if the error on the validation set stops decreasing.

Table II shows the experiment result. DoE outperformed baseline methods, even though the number of parameters is quite less than others. This result shows that deep architectures do not necessarily have good performance in computer vision tasks, even if it has been reported as good architecture. In short, it is necessary to tweak model architectures for specific tasks.

B. Practice Investigation

For practical use, we examine whether DoE is able to detect line damage from an actual image from a camera. For this, we retrain DoE from a binary classification problem to a 3-class classification problem; "undamaged (no-damaged)", "damaged", "no line". When we use DoE that solves a binary classification problem, it may cause false detection when there is no line. The result confusion matrix of 3-class classification

 ISBN: 978-1-61208-569-2

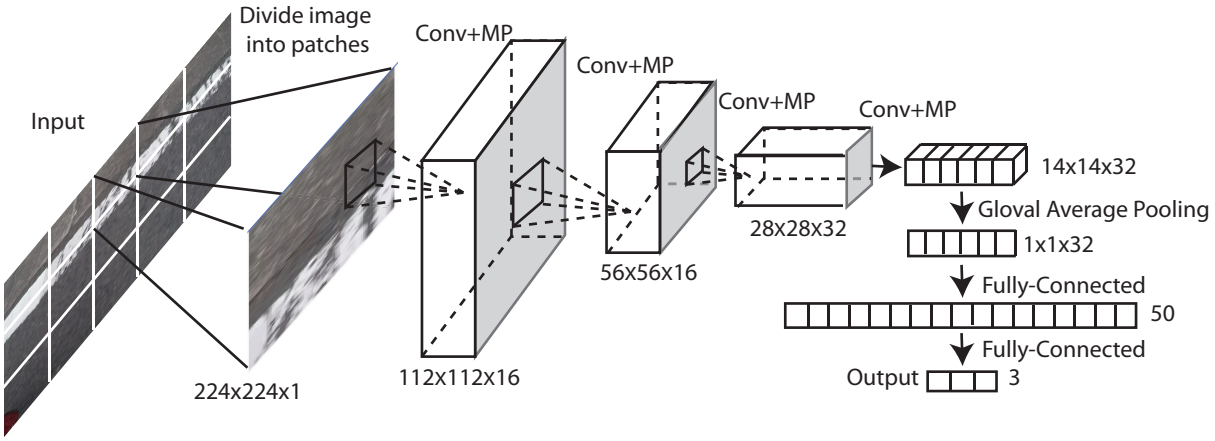

Figure 3. Our model on DoE architecture. "Conv + MP" denotes that convolutional operation with 3×3 kernels with strides 1 and max pooling operation with 2×2 kernels with strides 2. After four convolution and pooling, applying global average pooling [13].

is shown in Table III and actual detection in Figure 4. As a result, DoE can classify the road condition with 98% accuracy. Furthermore, in the case of Figure 4(a) (b), DoE classifies the patches perfectly. Note that in (b), at the location of the yellow font, the left upper patch is classified as "undamaged" correctly, while patches right side hand of it is classfifed as "damaged". On the other hand, DoE misclassifies "undamaged" patches as "damaged". This might happen if the line is dirty or something is on the line (e.g. the shoe is photographed in Figure 4(d)).

V. Discussion

A. Activation Visualization

In general, while it is said that CNNs are useful for image recognition, it is difficult to understand what the network learn. For instance, Figure 5 is the visualization of the kernel of first convolution layer of DoE before training and after. To tackle this problem, we visualize the activation of each kernel of DoE when the input damaged road image comes as shown in Figure 6. From the visualization, we can see that the model activates the part of damaged line like noisy dots, while there are only a few activated on the undamaged line. Remarkably, at the fifth layer, the activation of each unit in each image is mostly opposite. This result shows that DoE correctly learns the damage of line.

B. Input Image Generation

Furthermore, to understand the model in detail, we generate the image that DoE is likely to classify to damaged and undamaged. This method is inspired by [16]. The output of DoE is through a sigmoid function which has the asymptote $y = 0$ and $y = 1$ and the nature:

$$\lim_{x \to \infty} sigmoid(x) = 1 \tag{2}$$

$$\lim_{x \to -\infty} sigmoid(x) = 0. \tag{3}$$

Therefore, sigmoid is likely to output nearly 1 when it receives a large input and vice versa. Utilizing this nature, we can observe the output of DoE changes with fixed model parameters when we change the input. Beginning with a randomly initialized image, we use gradient ascent:

$$x \leftarrow x + \eta \frac{\partial a_i(x)}{\partial x} \tag{4}$$

to change an input image. Note that x denotes generated image input and η denotes learning rate. Furthermore, $a_i(x)$ represents the output of the ith layer and we use the last layer$i = 7$. Then, we maximize and minimize the output of $a_7(x)$ by Eq. (4). To emphasize the features which model learned, we applied Lp norm regularization:

$$||x||_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{\frac{1}{p}} \tag{5}$$

and total variation:

$$V(x) = \sum_{i,j} \sqrt{|x_{i+1,j} - x_{i,j}|^2 + |x_{i,j+1} - x_{i,j}|^2}, \tag{6}$$

which smoothed the images. The results are depicted in Figure 7. The image in the left of Figure 7 is classified by DoE as "undamaged", and the right image is classified as "damaged". There are much more white parts in the "undamaged" image than the "damaged" image. This shows that DoE recognizes line damage. In summary, from these visualizations, we found that DoE has learned to extract patterns and differentiate "damage" and "no damage" from the dataset, without any clues except from given labels.

VI. Related Works

Smart City. There are numerous of works that tackle city/urban problems from a point of smart city view. Zheng et.al [17] have contributed to urban energy issue and city planning by estimating the location of gas stations from the trajectories of taxis. Simultaneously, other works analyze urban livelihood from a geographic aspect [18] and detect where crime occurs [19]. These works are very important for citizens and the administration of cities to make their livelihood much better. Our work is an example of smart city work which makes transportation in cities more comfortable.

Road Inspection. From the point of the road inspection, there are a lot of points to inspect roads. One of those points is flatteness. To detect the flatteness, the use of accelerometer devices or the smartphones accelerometers is the straightforward

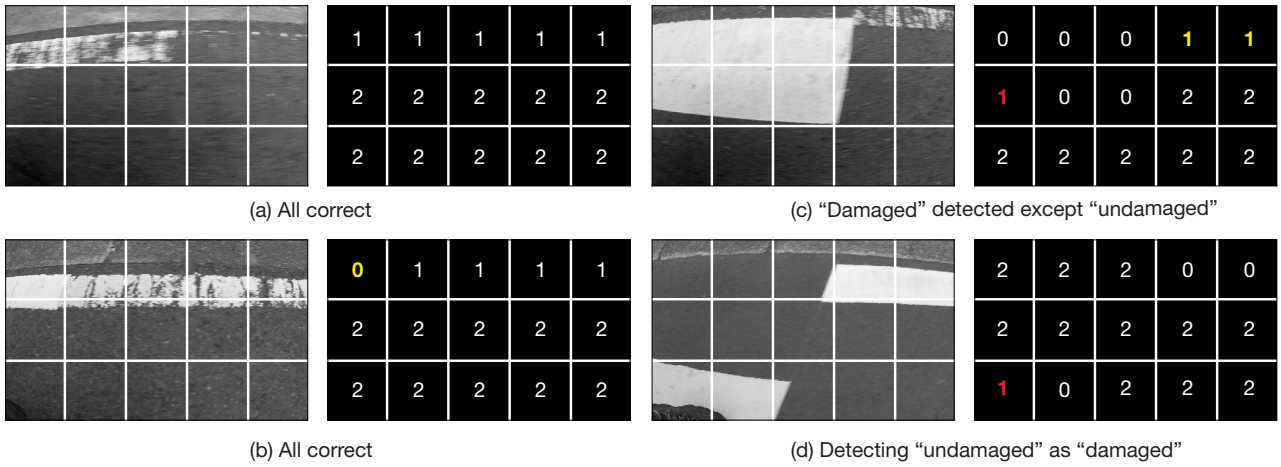

Figure 4. The result of the line damage detection with actual images. Each number denotes the classes, respectively; 0: undamaged, 1: damaged, 2: no line. (a) (b) DoE classifies all patches correctly. (c) Although DoE mistakes classifying “undamaged” as “no line” (at red fonts), it correctly detects damage at yellow fonts.

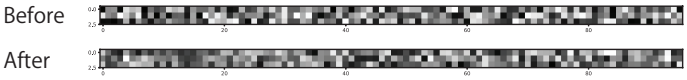

Figure 5. The visualization of kernels of first convolution layer of DoE before training and after.

approach [1]–[3]. Our goal of this study is line damage/blur detection, which is difficult to detect with accelerometers since the value of accelerometers do not change with respect to line damage. The other point of roads inspection is cracking. Concurrent to our method, Maeda et.al [6] used deep CNNs to detect road damage from images which are uploaded by citizens. Although they succeeded in detecting road damage on the application of smartphones, the model size is still too large to work on edge computer because of insufficient RAM. Furthermore, they relied on people to give image data, which is called *participatory sensing* [8] and depends on the motivation of participants. While there are numerous works to invent the incentive to make people more likely to participate [20] [21], the cost to offer the platform for participatory sensing still remains a problem. In contrast, we propose a system which collects images by using city vehicles, so that constant image data comes in daily.

The Aspect of Deep Learning. In order to run DoE on edge computers, a small model size is preferred. In the aspect of model compression, there is a lot of approaches [22] [23] and those are in progress. While our system divides the images into 224×224 patches in order to let DoE focus on the line, a model with an attention mechanism can be introduced to find the place where it should focus on in the image [24] [25]. Furthermore, while DoE classifies whole images, semantic segmentation [26] [27] can perform pixel-wise classification. In summary, we designed DoE to be simple, but there are a lot of improvements which can be made using new architectures.

VII. Conclusion

We presented DoE, a system that detects line damage via deep convolutional neural network working on an edge computer. Regarding the problem as a classification task, DoE produces a probability distribution over possible road conditions. This allows it to express its uncertainty about the damage of road. While previous work focused on detecting road flatteness, potholes or cracks, DoE is able to detect the damage or blur of lines. Our experiments show that DoE outperforms other methods for road damage detection, even though it has less parameters than other models. We further investigated how DoE learns to detect line damage by visualizing their activations over certain kinds of images, and generating images which the system is more likely to estimate as having damage. Furthermore, we show that it is practical to attach DoE to official city vehicles.

Acknowledgment

This work was supported by National Institute of Information and Communications Technology. This work was supported by RIKEN, Japan.

References

[1] B. Zhao, T. Nagayama, N. Makihata, M. Toyoda, M. Takahashi, and M. Ieiri, “Iri estimation by the frequency domain analysis of vehicle dynamic responses and its large-scale application,” in Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services. ACM, 2016, pp. 41–46.

[2] J. Takahashi, Y. Kobana, Y. Tobe, and G. Lopez, “Classification of steps on road surface using acceleration signals,” in proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services on 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2015, pp. 229–234.

[3] T. Nagayama, A. Miyajima, S. Kimura, Y. Shimada, and Y. Fujino, “Road condition evaluation using the vibration response of ordinary vehicles and synchronously recorded movies,” Proceedings of the SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring, 2013, p. 86923A.

 ISBN: 978-1-61208-569-2

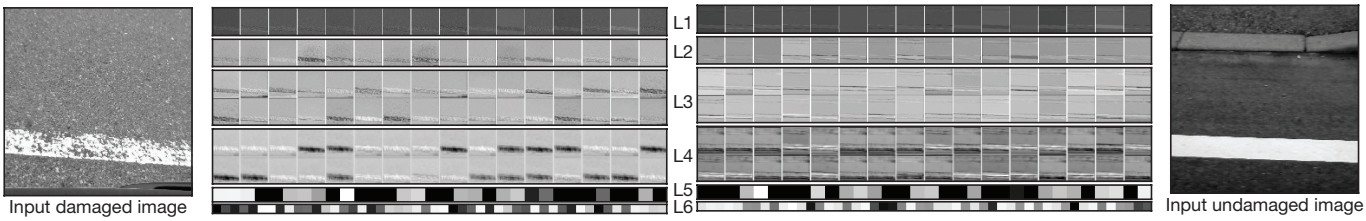

Figure 6. The visualization of the activation of each layer when the damaged line and undamaged line input images come. Remarkably at the fifth layer, each activation is active oppositely.

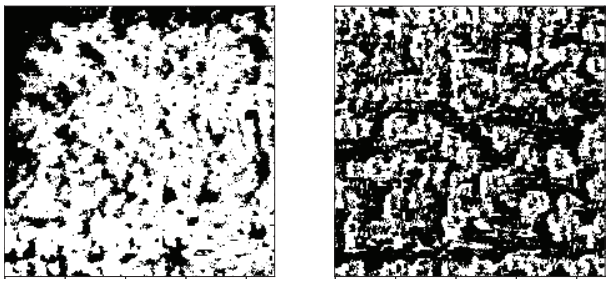

Figure 7. The examples that are generated to let DoE more likely to classify as "undamaged", and vice versa. There are more the white parts in "undamaged" image than "damaged" one.

[4] X. Yu and E. Salari, "Pavement pothole detection and severity measurement using laser imaging," in Electro/Information Technology (EIT), 2011 IEEE International Conference on. IEEE, 2011, pp. 1–5.

[5] K. Azhar, F. Murtaza, M. H. Yousaf, and H. A. Habib, "Computer vision based detection and localization of potholes in asphalt pavement images," in Electrical and Computer Engineering (CCECE), 2016 IEEE Canadian Conference on. IEEE, 2016, pp. 1–5.

[6] H. Maeda, Y. Sekimoto, and T. Seto, "Lightweight road manager: smartphone-based automatic determination of road damage status by deep neural network," in Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems. ACM, 2016, pp. 37–45.

[7] T. Kawasaki, M. Kawano, T. Iwamoto, M. Matsumoto, T. Yonezawa, J. Nakazawa, and H. Tokuda, "Damage detector: The damage automatic detection of compartment lines using a public vehicle and a camera," EAI MOBIQUITOUS2016IWWSS2016, 11 2016, pp. ppNA–ppNA.

[8] D. Estrin, K. M. Chandy, R. M. Young, L. Smarr, A. Odlyzko, D. Clark, V. Reding, T. Ishida, S. Sharma, V. G. Cerf et al., "Participatory sensing: applications and architecture [internet predictions]," IEEE Internet Computing, vol. 14, no. 1, 2010, pp. 12–42.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[12] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," arXiv preprint arXiv:1610.02915, 2016.

[13] M. Lin, Q. Chen, and S. Yan, "Network in network," arXiv preprint arXiv:1312.4400, 2013.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

[15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.

[16] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," Google Research Blog. Retrieved June, vol. 20, 2015, p. 14.

[17] F. Zhang, N. J. Yuan, D. Wilkie, Y. Zheng, and X. Xie, "Sensing the pulse of urban refueling behavior: A perspective from taxi mobility," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 6, no. 3, 2015, p. 37.

[18] A. Venerandi, G. Quattrone, and L. Capra, "Guns of brixton: which london neighborhoods host gang activity?" in Proceedings of the Second International Conference on IoT in Urban Space. ACM, 2016, pp. 22–28.

[19] A. Mashhadi, S. Bhattacharya, and F. Kawsar, "Understanding the impact of geographical context on subjective well-being of urban citizens," in Proceedings of the Second International Conference on IoT in Urban Space. ACM, 2016, pp. 29–35.

[20] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in Infocom, 2013 proceedings ieee. IEEE, 2013, pp. 1402–1410.

[21] T. Luo, H.-P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in INFOCOM, 2014 Proceedings IEEE. IEEE, 2014, pp. 127–135.

[22] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in Advances in Neural Information Processing Systems, 2015, pp. 3123–3131.

[23] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," arXiv preprint arXiv:1602.02830, 2016.

[24] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in International Conference on Machine Learning, 2015, pp. 2048–2057.

[25] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," arXiv preprint arXiv:1502.04623, 2015.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[27] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

 ISBN: 978-1-61208-569-2