The 2019 World Congress on
Advances in Nano, Bio, Robotics and Energy (ANBRE19)
Jeju Island, Korea, September 17 - 21, 2019

Deep learning-based crack detection using knowledge distillation

Donghwa Lee

Dept. of Computer & Communication Engineering, Daegu University,
Gyeongsan-si, Gyeongsangbuk-do 38453, Korea
leedonghwa@daegu.ac.kr

ABSTRACT

In this paper, a deep learning-based crack detection algorithm using knowledge distillation technique is presented. Recently, deep learning algorithm applying knowledge distillation technique is widely used in various embedded environments. This technique was applied to crack detection and the deep learning structure was lightened. For experiments, we select the YOLO algorithm that performs well among various deep learning algorithms. Through crack detection experiments, it is confirmed that the computation speed is fast as the crack detection performance of this algorithm is maintained.

1. INTRODUCTION

Recently, deep learning algorithm based on CNN (Convolutional Neural Networks) shows excellent performance in various object recognition areas (Krizhevsky 2012, LeCun 2015). In particular, you only look once(YOLO) algorithm shows very good detection performance (Redmon 2016). However, due to computational speed limitations, implementing CNN-based deep learning algorithms is still a difficult problem in low-power systems such as embedded.
Recently, various attempts have been made to reduce the computational cost of convolutional layers while maintaining the performance of deep learning algorithms (Hinton 2014, Romero 2015). These methods, called knowledge distillation (KD), attempt to reduce computational complexity by eliminating overlapping parts of deep learning structures.

In this paper, we apply deep learning algorithm in the crack detection domain and study the method to reduce the computation speed. As a deep learning algorithm, we modified the YOLO algorithm as an image classification problem. In addition, we implemented hints training method by separating the structure of YOLO algorithm and partially trained the same deep learning structure. Finally, we compared the performance of the original structure with the reduced YOLO structure.

2. PROPOSED KNOWLEDGE DISTILLATION METHOD

The structure of the YOLO algorithm for face detection is shown in Fig. 1. This structure consists of several layers of convolutional layers, and finally outputs classification results to determine the presence of cracks. The input image consists of 16 images of 104 × 104 × 3 and inputs 416 × 416 × 3 data by algorithm. The detection result is the presence or absence of crack, and the final detection stage determines the presence or absence of crack in 16 images. The YOLO algorithm is faster than other deep learning algorithms, but still difficult to apply in low-power environments such as embedded systems.

	layer	filters	size	input		output
Part 0	0 conv	32	3 x 3 / 1	416 x 416 x 3	->	416 x 416 x 32
	1 max		2 x 2 / 2	416 x 416 x 32	->	208 x 208 x 32
	2 conv	64	3 x 3 / 1	208 x 208 x 32	->	208 x 208 x 64
	3 max		2 x 2 / 2	208 x 208 x 64	->	104 x 104 x 64
	4 conv	128	3 x 3 / 1	104 x 104 x 64	->	104 x 104 x 128
	5 conv	64	1 x 1 / 1	104 x 104 x 128	->	104 x 104 x 64
	6 conv	128	3 x 3 / 1	104 x 104 x 64	->	104 x 104 x 128
	7 max		2 x 2 / 2	104 x 104 x 128	->	52 x 52 x 128
	8 conv	256	3 x 3 / 1	52 x 52 x 128	->	52 x 52 x 256
	9 conv	128	1 x 1 / 1	52 x 52 x 256	->	52 x 52 x 128
	10 conv	256	3 x 3 / 1	52 x 52 x 128	->	52 x 52 x 256
	11 max		2 x 2 / 2	52 x 52 x 256	->	26 x 26 x 256
	12 conv	512	3 x 3 / 1	26 x 26 x 256	->	26 x 26 x 512
	13 conv	256	1 x 1 / 1	26 x 26 x 512	->	26 x 26 x 256
Part 1	14 conv	512	3 x 3 / 1	26 x 26 x 256	->	26 x 26 x 512
	15 conv	256	1 x 1 / 1	26 x 26 x 512	->	26 x 26 x 256
	16 conv	512	3 x 3 / 1	26 x 26 x 256	->	26 x 26 x 512
	17 max		2 x 2 / 2	26 x 26 x 512	->	13 x 13 x 512
	18 conv	1024	3 x 3 / 1	13 x 13 x 512	->	13 x 13 x1024
	19 conv	512	1 x 1 / 1	13 x 13 x1024	->	13 x 13 x 512
	20 conv	1024	3 x 3 / 1	13 x 13 x 512	->	13 x 13 x1024
	21 conv	512	1 x 1 / 1	13 x 13 x1024	->	13 x 13 x 512
	22 conv	1024	3 x 3 / 1	13 x 13 x 512	->	13 x 13 x1024
	23 conv	1024	3 x 3 / 1	13 x 13 x1024	->	13 x 13 x1024
	24 conv	1024	3 x 3 / 1	13 x 13 x1024	->	13 x 13 x1024
	25 conv	1024	3 x 3 / 1	13 x 13 x1024	->	13 x 13 x1024
	26 conv	30	1 x 1 / 1	13 x 13 x1024	->	13 x 13 x 30
	27 detection					

Fig. 1 YOLO algorithm structure for crack detection. The structure is divided into two parts for hints training.

Fig. 2 shows how to apply hints training to YOLO algorithm. Once you have reduced the structure of the convolution layer, train each part separately to achieve the KD effect. After the YOLO structure was divided into two parts. Perform hints training for each part using each intermediate input and output as it is. In the training of each part, we train each part by applying linear function instead of ReLU function at the end.

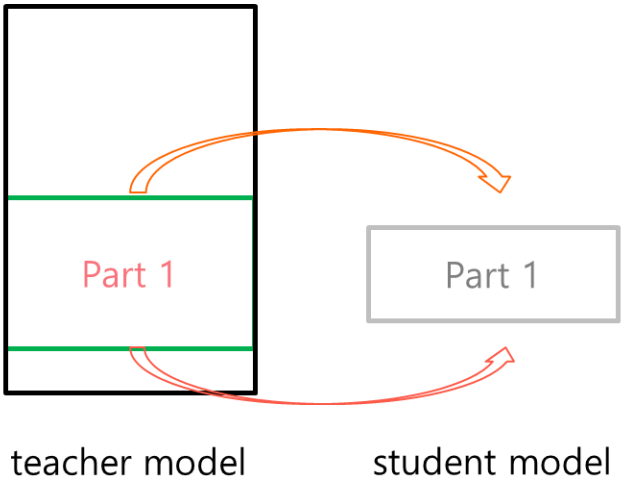

Fig. 2 Hints training on the CNN structure

3. EXPERIMENTAL RESULTS

In crack detection experiments, the crack data of the directly collected structures were used. Experiment with crack and no crack data. Figure 3 shows some examples of crack data sets. Of the datasets, I used 320 images for training and 64 images for testing. By modifying the last layer of the YOLO model, you are set up to learn whether to crack 16 images at a time. For the original YOLO model with hint training, the structure was divided into two parts, hint training was performed for each part, and finally fine tuning. The tiny YOLO model is trained through fine tuning in the final stage, using the weights learned from the VOC model. For each of the three YOLO models, 20,000 lessons are repeated to fine tune.

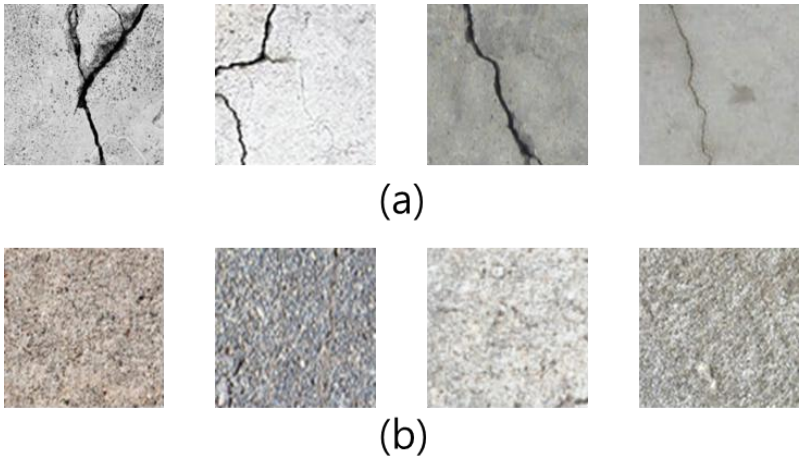

Fig. 3 Computational meshes for Gyeongju station

The crack detection results are shown in Table 1. Hint training is well performed and outperforms the tiny YOLO model. It shows that the CNN structure is reduced and the KD effect of the proposed structure can be obtained by hints training.

Table 1 Crack detection results

YOLO structure	Precision
Conventional YOLO	85.9 %
Hints trained YOLO	79.7 %
Tiny-YOLO	70.3 %

4. CONCLUSIONS

In this paper, we implemented deep learning-based crack detection algorithm and obtained KD effect through hints training. In addition, we performed experiments for crack detection based on YOLO algorithm and confirmed that the performance of KD-applied structures does not decrease significantly. In this study, we confirmed the applicability of deep learning algorithms to crack detection even for low power systems such as embedded systems.

REFERENCES

Hinton, G., Vinyals, O., Dean, J. (2014), “Distilling the Knowledge in a Neural Network”, *Proc. NIPS 2014 Deep Learning Workshop*.

Krizhevsky, A., Sutskever, I. and Hinton, G. (2012), “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, 1097-1105.

LeCun, Y., Bengio, Y. and Hinton, G. (2015), “Deep learning,” *Nature*, 521.7553, 436-444.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016), “You only look once: Unified, real-time object detection,” *Proc. CVPR 2016*.

Romero, A., Ballas, N., Kahou, S., E., Chassang, A., Gatta, C., Bengio, Y. (2015), “FitNets: Hints for Thin Deep Nets,” *Proc. ICRL 2015*.