Proceedings of CECNet 2022
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA220571

Comparison of Deep Learning Methods and a Transfer-Learning Semi-Supervised GAN Combined Framework for Pavement Crack Image Identification

Kai-liang LU, Guo-rong LUO, Ming ZHANG, Jin-feng QI and Chun-ying HUANG[1]
College of Automation, Guangzhou Vocational University of Science and Technology, China

Abstract. The pavement crack identification performance of typical models or algorithms of transfer learning (TL), encoder-decoder (ED), and generative adversarial networks (GAN), were evaluated and compared on SDNET2018 and CFD. TL mainly takes advantage of fine-tuning the architecture-optimized backbones pre-trained on large-scale data sets to achieve good classification accuracy. ED-based algorithms can take into account the fact that crack edges, patterns or texture features contribute differently to the identification. Both TL and ED rely on accurate crack ground truth (GT) annotation. GAN is compatible with other neural network architectures, thus can integrate various frameworks (e.g., TL, ED), and algorithms, but the training time is longer. In patch classification, the fine-tuned TL models can be equivalent to or even slightly better than the ED-based algorithms, and the predicting time is faster; In accurate crack location, both ED- and GAN-based algorithms can achieve pixel-level segmentation. It is expected to realize real-time automatic crack identification on a low computational power platform. Furthermore, a weakly supervised learning framework (namely, TL-SSGAN) is proposed, combining TL and semi-supervised GAN. It only needs approximately 10%−20% labeled samples of the total to achieve comparable crack classification performance to or even outperform supervised learning methods, via fine-tuned backbones and utilizing extra unlabeled samples.

Keywords. Image identification, pavement crack, transfer learning, encoder-decoder, generative adversarial network, supervised and semi-supervised learning

1. Introduction

Compared with contact detection techniques, such as non-destructive testing (NDT) [1] or structural health monitoring (SHM) [2], pavement crack identification with visual images via deep learning algorithms [3-6] has the advantages of not being limited by the material of object to be detected, fast speed and low cost. Thus, it has wide application prospects in routine inspection and other preventive detection or monitoring scenarios, where accurate classification and (patch- or pixel-level) segmentation can be obtained to identify the existence, topology, and even size of cracks.

[1] Corresponding author: Chun-ying HUANG, E-mail: huangchunying@gkd.edu.cn

Historically, *hand-crafted feature engineering* algorithms, include edge/morphology algorithms such as Canny, Sobel, HOG, LBP and feature transformation algorithms such as FHT, FFT, Gabor filters. These algorithms do not have to learn from a data set, and most computations are mathematics analytical, lightweight thus fast. However, the shortcomings are the weak generalization ability to various random variable factors, once the scenario or environment changes, the algorithms must be fine-tuned or redesigned, and even fail.

Later on, *machine learning* (ML) algorithms, e.g., CrackIT [7], CrackTree [8], and CrackForest [9] were developed for pavement(concrete) crack image identification. Recently, *deep learning* (DL) has continuously achieved state of the art (SOTA) progress in various missions. DL extracts high-level features automatically from large-scale data sets mainly via non-convex optimization, with stronger generalization ability and higher accuracy, whereas ML is oriented towards lower-level features via convex optimization. However, the mathematical expression of ML is explicit and interpretable [10], [11], whereas that of most DL algorithms are still implicit, known as "Black Box" issue, though great efforts and progress [12-14] have been made to solve the interpretability problem.

We focused on typical DL methods including *transfer learning* (TL) [15], [16], *encoder-decoder* (ED) [3], [4], and *generative adversarial networks* (GAN) [5], [6] for pavement(concrete) crack identification in this study. The main contents and contributions include:

- The fundamental frameworks and characteristics of typical TL, ED, and GAN algorithms are presented. Recent developments of these algorithms on pavement crack identification are summarized. The common architecture, modules, and specific techniques that improve the identification performance are highlighted.
- The patch sample classification performance, full-size image segmentation and detection effect were tested on public pavement crack data sets such as SDNET2018 and CFD. The performance of different neural network models of a certain algorithm, and various algorithms of DL methods, were evaluated and compared within and between categories.
- A *weakly supervised learning framework*, named TL-SSGAN, combining TL and semi-supervised GAN, is proposed, which can maintain comparable crack identification performance to or even outperform the supervised learning algorithms while greatly reducing the number of labeled samples required, through the measures of *(i)* utilizing fine-tuned TL backbones, *(ii)* controlling the ratio of labeled and unlabeled samples, and *(iii)* adding extra unlabeled samples.

2. Deep Learning Methods

2.1. Transfer Learning

Overall Framework and Procedure of TL. TL learns the basic reusable features via CNN backbone models, which are typically SOTA models that have been architecture-optimized and pre-trained on large-scale generic datasets (e.g., ImageNet). Then, the weight parameters of the upper and/or output layers are fine-tuned on a specific data set

(e.g., pavement crack dataset). TL is suitable for small datasets, easy to adjust, with good generalization performance, and provides fast training. The overall procedure and detailed data flow and steps of TL can refer to Fig. 8 of [16].

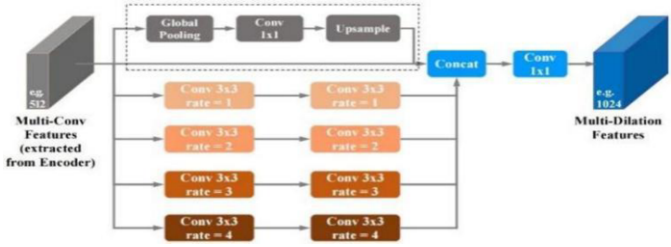
Common CNN Backbones. The evolution of common CNN backbone networks and the top-1/top-5 accuracy performance on ImageNet of typical backbone models for TL were illustrated. [2]

2.2. Encoder-Decoder

Motivation. The ED framework can compensate for the limitation of CNN [17] / FCN [18] algorithms for identifying complex crack topology (e.g., alligator cracks), that is, *(i)* pavement cracks have various morphology and topology. However, the CNN/FCN filters use specific kernels (3×3, 7×7, etc.), which limits the receptive field range and the robustness of crack identification. *(ii)* The fact that crack edges, patterns or texture features contribute differently to the identification has not been taken into account.

Architecture and Mechanism of Typical ED models. ED-based FPCNet [3] is one of the models with excellent accuracy and speed. It includes two sub-modules: a *multi-dilation* (MD) module and *SE-Upsampling* (SEU) module. Another model, U-HDN [4], similar to FPCNet [3], integrates MD module and *hierarchical feature* (HF) learning module based on U-net [19]. These two models are composed of similar or common sub-modules, such as the MD module (Fig. 1) based on the dilated convolution kernel operation, the SEU module (Fig. 2), and the U-net likewise main architecture (Fig. 3 and Fig. 4). Dilated convolution [20] enlarges the kernel's *context* window size, instead of using a sub-sampling operation or a larger filter with many more parameters.

As shown in Fig. 1, the MD module concatenates six branches, i.e., four dilated convolutions (double operations per branch) with rates of {1, 2, 3, 4} / {1, 2, 4, 8} / {2, 4, 8, 16}, which can be set according to the statistics of crack width, a global pooling layer, and the original crack multiple-convolution (MC) features. After concatenation, a 1×1 convolution is performed to obtain the crack MD features, which represent *contextual features ranging from pixel-level to global-level*, thus can detect cracks with different widths and topology.



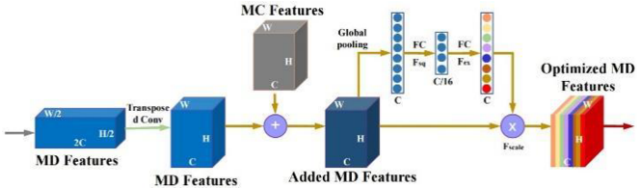
Remarks: The MC features and global pooling branches in the dashed rectangular box can also be integrated into the ED main architecture such as the design in U-HDN [4].

Figure 1. Multi-dilation module [3], [4].

The SEU module, as shown in Fig. 2, performs up-sampling operations in the decoder to continuously restore the resolution of the MD feature to the original size of input image, so that pixel-level crack identification can be realized. Through the *squeezing* and *excitation* learning, the SEU module adaptively assigns different weights

[2] https://github.com/mikelu-shanghai/Typical-CNN-Model-Evolution

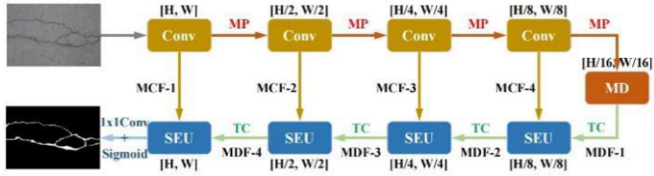
to different crack features such as edges, patterns, and textures. The inputs of the SEU module are the MD features and MC features, and the output is the optimized MD features after weighted fusion.



The detailed implementation remarks: (a) The SEU module first restored the resolution of MD features via transposed convolution. Then, MC features were added to MD features to fuse the associated crack information concerning crack edge, pattern, and texture, etc. (b) Subsequently, the SE operation was applied to the added MD features to learn the weights of the different features. Global pooling was first performed to obtain global information on C channels. After *squeezing* F_{sq} and *excitation* F_{ex} (two fully connected layers), the weight of each feature for its channel was obtained. (c) Finally, each feature in the added MD features was multiplied by its corresponding weight F_{scale}.

Figure 2. SE-Upsampling module [3].

The overall architectures of FPCNet [3] and U-HDN [4] are shown in Fig. 3 and Fig. 4 respectively. FPCNet [3] embedded the above MD and SEU modules into a common semantic segmentation network. U-HDN [4] integrated the MD module (bounded by the red dotted rectangular box in Fig. 4) and the HF learning module (in the yellow dotted box) based on the modified U-net architecture (in the blue dotted box), and zero filling was adopted during the up- and down-convolution paths.



Implementation details: (a) four Convs (two 3×3 and ReLUs) + max pooling were used as the encoder to extract features. Next, the MD module was employed to obtain the information on multiple context sizes. Subsequently, four SEU modules were used as the decoder. (b) *H, W* indicate the original image size. The red, green, and blue arrows indicate max pooling, transposed convolution, and 1×1 convolution + sigmoid, respectively. MCF denotes the MC features extracted in the encoder and MDF denotes the MD features.

Figure 3. Overall architecture of FPCNet [3].

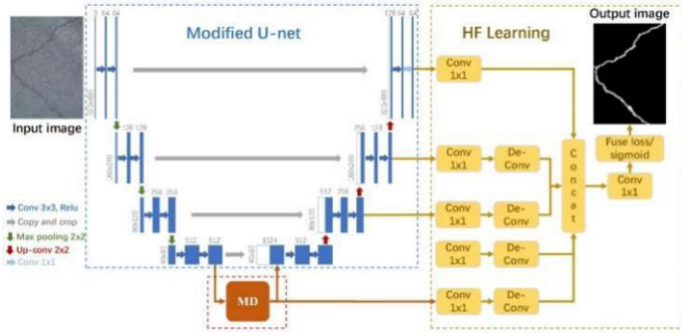


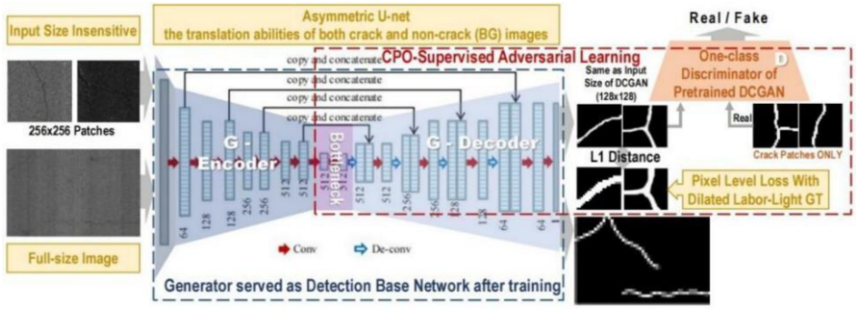
Figure 4. Overall architecture of U-HDN [4].

In addition, Li *et al.* [21], inspired by DenseNet, fused a densely connected convolution module and a deep supervision module to extract detailed crack features. Yu *et al.* [22] introduced CCapFPN, based on a capsule feature pyramid network (FPN) architecture. In summary, the aforementioned ED models [3], [4], [21], [22] contribute to common ideas, similar architectures, and fundamental modules.

2.3. Generative Adversarial Networks

Overview and Taxonomy. GAN makes the generated samples obey the distribution of real data via two networks (discriminator and generator) adversarial training. GAN has the advantages of *(i)* accurate estimation of the density function, *(ii)* efficient generation of required samples, *(iii)* elimination of deterministic bias, and *(iv)* good compatibility with various neural network architectures, algorithms, and techniques. Thus, GAN has received extensive attention and many GAN variants were derived [23-25]. The evolution diagram and main characteristics of some common GANs were shown in Fig. 7 of [3]. It can be used for hard sample mining in semi-supervised/unsupervised learning.

Typical GAN algorithms for Crack Identification. ConnCrack [5] and CrackGAN [6] are representative GANs for highly accurate crack identification. However, the training and prediction times were longer in the earlier version. The improved CrackGAN [6], based on DCGAN and with encoder-decoder as the generator, proposed crack-patch-only (CPO) supervised adversarial learning and the asymmetric U-Net architecture to perform end-to-end training with partially accurate ground truth (GT) (i.e.1-pixel curve manual labeling) labeled in a labor-light manner. The overall architecture of Crack-GAN [6] is illustrated in Fig. 5.



Remarks: (a) D is a One-class DCGAN discriminator that is pre-trained only by crack patches, which will enable the network to always generate crack images. It is a key module to conquer the "All-Black" issue. (b) Pixel-level loss is used to make sure the crack pattern generated is just the same as the input patches, which is achieved by optimizing the L_1 distance from Dilated Labor-light GT, e.g., 1 pixel dilated 3 times. The total loss is thus L_1 distance + adversarial loss generated by the DCGAN. Transfer learning is employed to train the prototype of the encoding network and transfer the knowledge from DCGAN to provide generative adversarial loss for end-to-end training. (c) An asymmetric U-net network is introduced to balance the crack and non-crack/background samples that are severely imbalanced. d) After the training, the generator G will act as the detection network for new samples. e) The whole network can handle arbitrary full-size images, since it's a fully convolutional network (FCN). For more implementation details, refer to [6].

Figure 5. Overall architecture of CrackGAN [6].

CrackGAN [6] was developed to handle the practical problem named "All-Black" issue, i.e., the network converged to the state that the whole crack image was regarded

[3] https://arxiv.org/ftp/arxiv/papers/2112/2112.10390.pdf

as the background, which is caused by (i) the data imbalance of crack and background/uncrack samples; and (ii) blurred boundaries of tiny long cracks that per-pixel accurate labeling is difficult or infeasible. CrackGAN [6] can significantly reduce the workload of GT labeling and achieve excellent performance when dealing with full-size images for pixel-level crack segmentation. The computational efficiency is also greatly improved (predicting a 4096×2048 image takes approximately 1.6 s on an NVIDIA 1080Ti GPU).

3. Evaluation and Comparison of the Crack Identification Algorithms

3.1. Public Crack Datasets [4]

SDNET2018. It is a *patch-level annotated* dataset for training and benchmark test of crack-identification algorithms. There are 230 photographs, including 54 bridge decks, 72 walls, and 104 sidewalks. Each photo was cropped into 256×256 patches. In total, it contains 56, 092 sample images. The crack sizes vary from 0.06 *mm* to 25 *mm*. Random variable factors (environmental, background, interference, etc.) in the images were listed [5], and many positive (crack) and negative (uncrack) samples are difficult to be recognized by human eyes.

Crack Forest Dataset (CFD). It is a *pixel-level annotated* pavement crack dataset that reflects the general situation of Beijing urban pavements. It is one of the benchmark baseline datasets. In total, 118 photos were collected and the samples contain noise or interference factors such as lane lines, shadows, and oil stains.

3.2. Crack Patch Classification Evaluation

The patch classification results of the TL algorithms were compared with FCN [18] (FCN is a one-stage pixel-level semantic segmentation algorithm that does not require window sliding), and the ED-based FPCNet [3], as listed in Table 1.

Table 1. Patch classification test performance comparison.

Algorithm/Model [a]	Accuracy	Precision	Recall	F_1-score	Predicting Time @ GTX1080Ti [b]
TL-MobileNetV2	0.936 8	0.947 7	0.973 2	0.960 3	8.1 ms/patch
TL-InceptionV3	0.941 0	0.952 4	0.980 6	0.966 3	16.1 ms/patch
TL-Resnet152V2	0.956 1	0.961 3	0.981 7	0.971 4	50.2 ms/patch
Original FCN [18]	0.965 8	0.972 9	0.945 6	0.959 0	19.8 ms/patch
ED-FPCNet [3]	0.970 7	0.974 8	0.963 9	0.969 3	67.9 ms/patch

[a] TL algorithms were tested on SDNET2018, whereas the FCN and ED algorithms were tested on CFD (statistical results at the pixel level), which is relatively more recognizable than SDNet2018. The results are the average of 10 runnings'. [b] The predicting time includes image loading, pre-processing, and inference time.

It can be seen from Table 1 that: *(i)* the testing accuracy of TL algorithms on SDNET2018/CFD has exceeded the ImageNet baseline of the backbones. *(ii)* The performance of fine-tuned TL algorithms is close to or even slightly better than (e.g., on recall & F_1-score) that of the FCN and ED algorithms, which is attributed to the

[4] Typical public pavement(concrete) crack datasets, their sample features, and download sources were collected and listed in [16].

[5] https://arxiv.org/ftp/arxiv/papers/2112/2112.10390.pdf

architecture-optimized backbones pre-trained on large-scale datasets. Thus, the TL's predicting time is much lower than the 67.9 *ms* of FPCNet [3].

The TL algorithms are trained and tested on patch samples cropped from the original images. Owing to the limitation of the minimum size of the patches, they have a poor perform on full-size image segmentation [26], so are generally used for classification. The crack segmentation relies mainly on the ED and GAN algorithms.

On the other hand, the imbalance of crack and uncrack/background samples (SDNET2018 is approximately 1:5.6) may be compensated by the class weight or class-balanced loss function. However, owing to the ambiguous boundary of the topological complex cracks and thin cracks (e.g., pavement crack images collected by high-speed vehicle cameras, and defect images of the industrial products on an automatic assembly line), accurately per-pixel annotation is labor-intensive and even impossible, thus crack GT semi-accurate annotation (i.e., 1-pixel curve labeling, with 2 pixels labeling error) is usually adapted in practice, which will lead to bias in the evaluation via accuracy, precision, recall, or F_1-score, and even worse, may cause to encounter the "All-Black" issue. Hence, the HD-score is advised as a quantitative evaluation indicator [6], [27] for crack pixel-level segmentation, with good discrimination when crack GT semi-accurate labeling [28].

3.3. Crack Segmentation Performance

A patch-level RCNN based method proposed in [26], cannot accurately locate the crack location owing to the limitation of the patch size. The prediction of full-size images required the usage of a sliding window, which increased prediction time a lot. It took approximately 10.2*s* for a single CFD sample. FCN-VGG [15] is a pixel-level identification algorithm that implements end-to-end training relying on accurate GT annotation at each pixel. When there is a deviation in the GT, thin cracks will not be detected. DeepCrack [29] achieves good performance with multi-scale hierarchical fusion, with HD-score of 94 and prediction time of 2.4*s*, but it also relies on the accurate GT.

As mentioned above (detailed in Fig. 5), by introducing CPO supervision and asymmetric U-net, CrackGAN [6] proposed a GAN architecture with a single-class discriminator. It successfully avoided the "All-Black" problem caused by the inherent imbalance of positive and negative samples, and significantly improved the ability of accurate crack segmentation, with HD-score as high as 96, especially good at the identification of thin cracks. And it only takes approximately 1.6*s* to predict a 4096×2048 image on an NVIDIA GTX1080Ti GPU.

4. A Weakly Supervised Learning Framework: TL-SSGAN

CrackGAN [6] utilized the semi-accurate labor-light annotation (i.e., 1-pixel curve manual labeling) to significantly reduce labeling difficulty and workload. On the other hand, *semi-supervised* GAN (SSGAN) can also significantly reduce the need for labeled samples from the perspective of semi-supervision [30], [31]. A weakly supervised learning framework (i.e., TL-SSGAN) that integrates TL and SSGAN, is proposed for classification, as shown in Fig. 6. TL backbone models can be pre-trained or fine-tuned. The ratio of labeled to unlabeled samples can be used as a variable parameter, and extra

unlabeled samples may also be added. The output of the TL-SSGAN is whether the test samples are generated or real, and whether a crack or not.

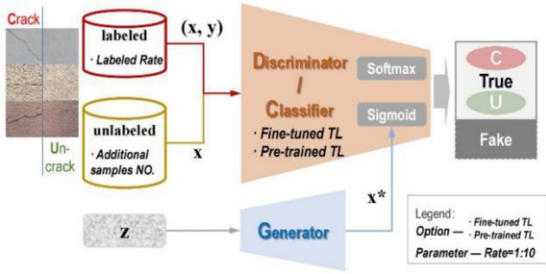


Figure 6. Pipeline of semi-supervised GAN embedded TL models as encoder (Discriminator/Classifier).

The evaluation tests were performed on SDNET2018, and results seen in Table 2.

Table 2. Test performance of TL-SSGAN on SDNET2018.

TL-SSGAN Framework with Different Models and Data			Accuracy	Precision	Recall	F_1-score
Row	TL Backbone, fine-tuned or not	Labeled Rate + Extra Unlabeled Samples				
1	Resnet152V2, pre-trained	1:30	0.888 6	0.926 7	0.945 3	0.935 9
2	Resnet152V2, pre-trained	1:20	0.889 2	0.931 4	0.947 8	0.939 5
3	Resnet152V2, pre-trained	1:10	0.905 9	0.939 6	0.955 3	0.947 4
4	Resnet152V2, fine-tuned	1:30	0.901 6	0.929 3	0.956 7	0.942 8
5	Resnet152V2, fine-tuned	1:20	0.906 5	0.930 0	0.958 9	0.944 2
6	Resnet152V2, fine-tuned	1:10	0.924 5	0.941 9	0.974 9	0.958 1
7	InceptionV3, fine-tuned	1:10	0.917 0	0.945 5	0.961 0	0.953 2
8	MobileNetV2, fine-tuned	1:10	0.910 6	0.936 2	0.959 0	0.947 5
9	Resnet152V2, fine-tuned	1:10 + Extra 10k samples	0.931 4	0.946 9	0.972 1	0.959 3
10	Resnet152V2, fine-tuned	1:10 + Extra 20k samples	0.938 8	0.957 0	0.972 3	0.964 9
11	Resnet152V2, fine-tuned	1:5	0.946 0	0.972 1	0.970 1	0.971 1
12	Resnet152V2, fine-tuned	1:5 + Extra 10k samples	0.953 2	0.973 0	0.978 6	0.975 8

(1) In general, models of TL-SSGAN achieved good patch classification performance on SDNET2018. The best accuracy, precision, recall, and F_1-score were 0.953 2, 0.973 0, 0.978 6, and 0.975 8 respectively when utilizing 1/5 labeled samples of the total plus extra 10k unlabeled samples via Resnet152V2, which can outperform the supervised TL and ED algorithms (refer to Table 1 & Table 2).

(2) Both the backbone model and fine-tuning mechanism in the TL framework contributed significantly to the improvement of accuracy. For instance, all other factors and parameters being the same, Resnet152V2-based algorithm is 1.5%, 0.6%, 1.7% and 1.1% better than MobileNetV2-based, respectively, on metrics of accuracy, precision, recall, and F_1-score (i.e., Row 6 *vs.* 8 in Table 2); while the results of the fine-tuned algorithm increase by approximately 1.5−2.1%, −0.2−0.3%, 1.2−2.1%, 0.5−1.1% respectively, compared to the pre-trained algorithm, when both with Resnet152V2 as the backbone (i.e., Row 4-6 *vs.* Row 1-3 respectively).

(3) In the aspect of data usage, when the ratio of labeled/total sample number is 1:10, compared to 1:30 (Resnet152V2 as the backbone), the metric results increase by approximately 1.9−2.5%, 1.4%, 1.1−1.9%, 1.2−1.6% (i.e., Row 3 *vs.* 1 and Row 6 *vs.* 4). By adding extra unlabeled samples up to 2 times, an increase of approximately 0.7% can be achieved (i.e., Row 10 *vs.* 6). Therefore, the weakly supervised mechanism of the

proposed TL-SSGAN framework not only reduces the dependence on labeled samples but also improves the classification performance by using incremental unlabeled samples.

5. Conclusion

In summary, the patch classification accuracy performance of the fine-tuned TL algorithms on public crack datasets (e.g., CFD, SDNET2018), is comparable to or slightly better than that of ED algorithms; and the predicting time cost is less (approximately $8.1 - 50.2$ *ms/patch*). In accurate crack location, both the ED and GAN can achieve pixel-level segmentation, furthermore, CrackGAN can achieve high-precision segmentation under labor-light partially accurate GT annotation (e.g., 1-pixel curve manual labeling). In terms of detection efficiency, CrackGAN only takes approximately 1.6 *s* to predict a 4096×2048 image on an NVIDIA GTX1080Ti GPU. It is expected to realize real-time crack detection on a low computational power platform.

We proposed a weakly supervised learning framework TL-SSGAN combining TL and semi-supervised GAN, through the measures of *(i)* utilizing fine-tuned TL backbones, *(ii)* controlling the ratio of labeled and unlabeled samples, and *(iii)* adding extra unlabeled samples, which can maintain comparable crack classification performance to or even outperform the supervised learning while greatly reducing the number of labeled samples (approximately 1/10 ~ 1/5 of the total) needed.

To conclude, the combination of various deep learning frameworks, such as TL, ED and GAN/SSGAN, can integrate the advantages of each mechanism and improve the performance of the overall architecture. Although pavement crack is taken as a case study, the algorithms discussed here can be easily modified for crack identification in other engineering structures.

Acknowledgements

This work was supported by NSFC (No.51405289) and Research Project of Education Department of Guangdong Province (No.2022KTSCX192).

References

[1] K.L. Lu, W.G. Zhang, Y. Zhang, H. Huang, Y.S. Chen, W.Y. Li and C. Wang, "Crack analysis of multi-plate intersection welded structure in port machinery using finite element stress calculation and acoustic emission testing," *International Journal of Hybrid Information Technology*, vol. 7, no. 5, pp. 323–340, 2014.

[2] A. Martone, M. Zarrelli, M. Giordano, and J. M. L´opez-Higuera, *Structural health monitoring in buildings, bridges and civil engineering*. New Jersey: Photonics for Safety and Security, World Scientific, 2013.

[3] W.J. Liu, Y.C. Huang, Y. Li, and Q. Chen, "FPCNet: Fast Pavement Crack Detection Network Based on Encoder-Decoder Architecture," *arXiv:1907.02248*, 2019.

[4] Z. Fan, C. Li, Y. Chen, J.H. Wei, L. Giuseppe, X.P. Chen and D. M. Paola, "Automatic crack detection on road pavements using encoder-decoder architecture," *Materials*, vol. 13, no. 13, pp. 1–18, 2020.

[5] Q.P. Mei and M. G¨ul, "A cost effective solution for pavement crack inspection using cameras and deep neural networks," *Construction and Building Materials*, vol. 256, p. 119397, 2020.

[6] K.G. Zhang, Y.T. Zhang, and H.D. Cheng, "CrackGAN: Pavement Crack Detection Using Partially Accurate Ground Truths Based on Generative Adversarial Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 1306–1319, 2021.

[7] O. Henrique and C. P. Lobato, “Automatic road crack detection and characterization,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 155–168, 2013.
[8] Q. Zou, Y. Cao, Q.Q. Li, Q.Z. Mao, and S. Wang, “CrackTree: Automatic crack detection from pavement images,” *Pattern Recognition Letters*, vol. 33, no. 3, pp. 227–238, 2012.
[9] Y. Shi, L.M. Cui, Z.Q. Qi, et al., “Automatic road crack detection using random structured forests,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3434–3445, 2016.
[10] J. Wright and Y. Ma, *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge University Press, 2021.
[11] C. Giraud, *Introduction to High-Dimensional Statistics*, 2nd ed. Chapman and Hall, CRC, 2021.
[12] R. Ge, H. Lee, and J.F. Lu, “Estimating normalizing constants for log-concave distributions: Algorithms and lower bounds,” in *STOC 2020: the 52nd Annual ACM Symposium on Theory of Computing*, 2020, pp. 1–46.
[13] Y.D. Yu, K.H.R Chan, C. You, C.B. Song, and Y. Ma, “Learning diverse and discriminative representations via the principle of maximal coding rate reduction,” in *NeurIPS 2020: the 34th Conference on Neural Information Processing Systems*, 2020, pp. 1–28.
[14] K.H.R. Chan, Y.D. Yu, C. You, H.Z. Qi, J. Wright, and Y. Ma, “Deep networks from the principle of rate reduction,” *arXiv:2010.14765*, 2020.
[15] X.C. Yang, H. Li, Y.T. Yu, X.C. Luo, T. Huang, and X. Yang, “Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1090–1109, 2018.
[16] K.L. Lu, “Advances in deep learning methods for pavement surface crack detection and identification with visible light visual images,” *Computer Enginerring & Science,* vol.44, No.4, pp. 674-685, 2022.
[17] Y. Fei, C.P.(Kelvin) Wang, A. Zhang, C. Chen, J.Q. Li, Y. Liu, G.W. Yang, and B.X. Li, “Pixel-Level Cracking Detection on 3D Asphalt Pavement Images through Deep-Learning- Based CrackNet-V,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 273–284, 2020.
[18] E. J. Long and T. D. Shelhamer, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.
[19] O. Ronneberger, P. Fischer and T., Brox, “U-net: convolutional networks for biomedical image segmentation,” in *Proceedings of the Lecture Notes in Computer Science*, 2015, pp. 234–241.
[20] F. Yu and V. Koltun, “Dilated residual networks,” in *CVPR*, 2017, pp. 472–480.
[21] H.F. Li, J.P. Zong, J.J. Nie, Z.L. Wu, and H.Y. Han, “Pavement Crack Detection Algorithm Based on Densely Connected and Deeply Supervised Network,” *IEEE Access*, vol. 9, pp. 11 835–11 842, 2021.
[22] Y.T. Yu, H.Y. Guan, D.L. Li, Y.J. Zhang, S.H. Jin, and C.H. Yu, “CCapFPN: A Context-Augmented Capsule Feature Pyramid Network for Pavement Crack Detection,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.
[23] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Deep learning for visual understanding: Part 2 generative adversarial networks,” *IEEE Signal Processing Magazine*, no. 1, pp. 53–65, 2018.
[24] Y. Hong, U. Hwang and S. Yoon, “How generative adversarial nets and its variants work: An overview of GAN,” *ACM Computing Surveys*, vol. 52, no. 1, pp. 1–43, 2019.
[25] Z.W. Wang, Q. She and T.E. Ward, “Generative adversarial networks in computer vision: A survey and taxonomy,” *ACM Computing Surveys*, vol. 54, no. 1, pp. 1–41, 2021.
[26] Y.J. Cha, W. Choi and O. B¨uy¨uk¨ozt¨urk, “Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.
[27] Y.C.(James) Tsai and A. Chatterjee, “Comprehensive, Quantitative Crack Detection Algorithm Performance Evaluation System,” *Journal of Computing in Civil Engineering*, vol. 31, no. 5, p. 04017047, 2017.
[28] Y. Inoue and H. Nagayoshi, “Crack detection as a weakly-supervised problem: Towards achieving less annotation-intensive crack detectors,” in *ICPR*, 2020, pp. 1–10.
[29] Q. Zou, Z. Zhang, Q.Q. Li, X.B. Qi, Q. Wang, and S. Wang, “DeepCrack: Learning hierarchical convolutional features for crack detection,” *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1498–1512, 2019.
[30] J. T. Springenberg, “Unsupervised and semi-supervised learning with categorical generative adversarial networks,” in *International Conference on Learning Representations (ICLR)*, 2016, pp. 1–20.
[31] Z.H. Dai, Z.L. Yang, F. Yang, W.W. Cohen and R. Salakhutdinov, “Good semi-supervised learning that requires a bad GAN,” in *NIPS*, 2017, pp. 6511–6521.