AggCrack: An Aggregated Attention Model for Robotic Crack Detection in Challenging Airport Runway Environment

Haifeng Li, Jianping Zong, Rui Huang, Zhongcheng Gui, and Dezhen Song

***Abstract*— Crack detection is essential for guaranteeing airport runway structural reliability. An efficient solution we take is to employ a robot equipped with a camera to perform inspection task. However, automatic crack detection for airport runway is challenging, as the runway surface is seriously polluted by fuel stain and aircraft wheel mark, and the cracks need to be detected are usually extremely thin. Thus, we propose a CNN model, AggCrack, to perform the crack detection task. AggCrack adopts an innovative semantic-level attention mechanism on the edges of the targets to focus the model on crucial features, and combines edge information and semantic segmentation for more accurate crack detection. We have implemented the algorithm and have it extensively tested on an airport runway dataset collected by our inspection robot from four different airport runways. Compared with four existing deep learning methods, experimental results show that our algorithm outperforms all counterparts. Specifically, it achieves the Precision, Recall and F1-measure at 84.24%, 70.36% and 76.68%, respectively.**

I. Introduction

Airport runway is among the most fundamental infrastructure to guarantee the safety of aircraft during taking-off and landing. Cracks are common defects occurring on airport runway, which may decrease the stress state of runway pavement and even lead to accidents. Thus, detection of cracks is a mandatory provision according to the regulations on airport pavement management issued by the International Civil Aviation Organization (ICAO). Currently, crack detection for airport runway still relies on manual visual inspection, which is time-consuming, labor-intensive, and error-prone. Therefore, it is necessary to automate the crack detection process. To achieve it, we develop a robot equipped with an on-board RGB camera to perform airport runway inspection, as shown in Fig.1, where the robot moves along a pre-defined grid route to capture runway images. However, automatic crack detection from the captured images is still challenging, as the runway surface is seriously polluted by fuel stain and aircraft wheel mark, and the cracks need to be detected are usually extremely thin, because the thin cracks may be early warning signs of significant failure. Fig.1 presents samples of very noisy images including extremely thin cracks, fuel stains, and textured surface.

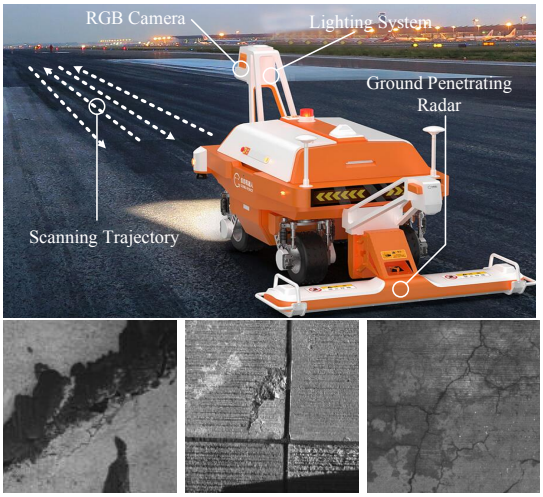

Fig. 1. An illustration to our airport runway inspection robot and the captured runway pavement images with strong background noise.

Traditional crack detection methods exploit the continuous characteristic of cracks to develop local image processing algorithms, such as intensity thresholding, edge detection based techniques, and morphological methods. However, their performance usually depends on the parameter choice with extensive manual tuning, which limits their performance in field applications. In recent years, machine learning, especially deep learning approaches have been proposed and achieved impressive performance for automatic crack detection. However, their performance degrades significantly when dealing with the airport runway scenario with significant background noise.

We observe that cracks have fewer internal features than common targets, but edge features still have relatively strong response to cracks even under the condition with significant noises. This inspires us to find the way to improve the model's attention on edges, which may be disjointed and even contain much noise, for crack detection. As a result, we propose an aggregated attention CNN model, AggCrack, to take the leverage of the attention on edges to detect cracks more accurately and robustly.

We have evaluated our approach on a camera image data set collected from four airport runways using our inspection robot. Comparative results show that our proposed AggCrack

This work was supported in part by National Key Research and Development Project of China under 2019YFB1310601.

H. Li, J. Zong and R. Huang are with Civil Aviation University of China, Tianjin, 300300, China. Emails: hfli@cauc.edu.cn, jianpingzong@163.com, and rhuang@cauc.edu.cn.

Z. Gui is with Shanghai Guimu Robot Co. Ltd., Shanghai, 200092, China. Email: guizhongcheng@gm-robot.com.

D. Song is with CSE Department, Texas A&M University, College Station, TX 77843, USA. Email: dzsong@cs.tamu.edu.

outperforms four recent deep learning based crack detection methods. Specifically, our algorithm achieves the Precision, Recall and F1-measure at 84.24%, 70.36% and 76.68%, respectively.

II. Related Works

The image-based automatic crack detection algorithms can be generally divided into two categories: traditional image processing methods, and machine learning based methods.

A. Traditional Image Processing Methods

With perspective of data analysis domain, the traditional image-based crack detection methods can be classified into two types: spatial domain analysis, and frequency domain analysis.

1) Crack detection based on spatial domain analysis: Intensity thresholding methods have been widely used due to simplicity. Intuitively, cracks are darker then surrounding background. Li et al. [1] employ nearby region to generate the gray histogram to distinguish. However, these methods are sensitive to noise, leading to being rarely used alone in practical applications.

Morphological methods leverage the connectivity among crack pixels. Nguyen et al. [2] propose Free-Form Anisotropy, which takes into account brightness and connectivity for crack detection simultaneously. In this category, our group [3] proposes a multi-scale image fusion crack detection method. The common problem to morphological methods is that their performance usually depend upon the parameter choice with manual extensive tuning, which limits their usage in field applications.

Edge detection methods are also applied to detect cracks [4], [5], since cracks and edges have similar characteristics in shape, structure and thickness. However, the main problem is that edge detection methods can only detect a set of disjoint crack fragments and often fail in high-clutter images. How to leverage edge information efficiently for crack detection is an open issue.

2) Crack detection based on frequency domain analysis: Wavelet transform methods have been applied for crack detection. Zhou et al. [6] decompose a whole pavement image into different-frequency sub-bands with wavelet transform. Specifically, pavement cracks are transformed into high-frequency wavelet coefficients while noise is transformed into low-frequency ones. Besides, the details of both cracks and noise in high-frequency sub-bands are preserved. However, it is difficult to apply wavelet transform to detect cracks in poor continuity due to the anisotropic characteristic of wavelets.

B. Machine Learning based Methods

Liu et al. [7] propose Richer Convolutional Features (RCF) to produce high-quality edges efficiently by combining multi-scale and multi-level information of objects. However, these algorithms are still based on the human-selected features, which have weak adaptability and poor robustness in complex environments.

In recent years, Convolutional Neural Networks (CNNs) are widely applied to solve the problem of automatic crack detection. Chen et al. [8] classify the crack patches of the nuclear power plant from video sequences by an eleven-layer CNN, and the Naive Bayes approach is introduced for post-processing the data. However, these methods above only operate at patch-level, which leads to a tremendous amount of detection time. The recent researches of pixel-level semantic segmentation show a couple of great results. Yang et al. [9] use a Fully Convolutional Network (FCN) to detect cracks while quantitatively measuring their spatial characteristics. Yang et al. [10] extend the HED [11] network to the Feature Pyramid and Hierarchical Boosting Network (FPHBN) for pavement crack detection. Lau et al. [12] use a U-Net [13] architecture with pertained ResNet-34 [14] encoder for end-to-end pavement crack detection. Although the deep learning based methods have achieved impressive performance for automatic crack detection, their performance degrades significantly when dealing with the airport runway scenario with strong interference.

III. Algorithm

Considering the challenging airport runway environment, a potential solution to deal with the pavement pollution is to raise the model's attention on edges, which are more prominent than internal features of cracks. Thus, we propose a CNN model, AggCrack, to process the collected runway images, as shown in Fig.2. First, the image goes through a feature extractor to generate pyramid features in five different scales. Feature maps of each scale are reused to preserve details in the expansive paths. After the feature extractor, two parallel branches of expansion module are implemented. Among them, one is set to focus the model on edges, and the other is set to generate initial crack semantics prediction maps. Finally, a fusion module is implemented to achieve more accurate crack detection results by combining crack semantic information and edge features. In this section, we will introduce each module of AggCrack in detail.

A. Feature Extractor

We analyze the prevalent semantic segmentation methods such as FCN, DeepLabV3 and U-Net both in theory and experiment, and find that U-Net has advantages in crack detection. Compared with the other networks, U-Net adopts a more conservative convolution strategy, but it is conducive to extracting detailed features from thin cracks. Moreover, U-Net is designed for retinal vessels segmentation, which have similar characteristics to cracks in shape and structure. Thus, we select U-Net as our backbone to extract features for crack detection.

Inspired by similar VGG [15] network, we build the feature extractor for following crack semantic segmentation

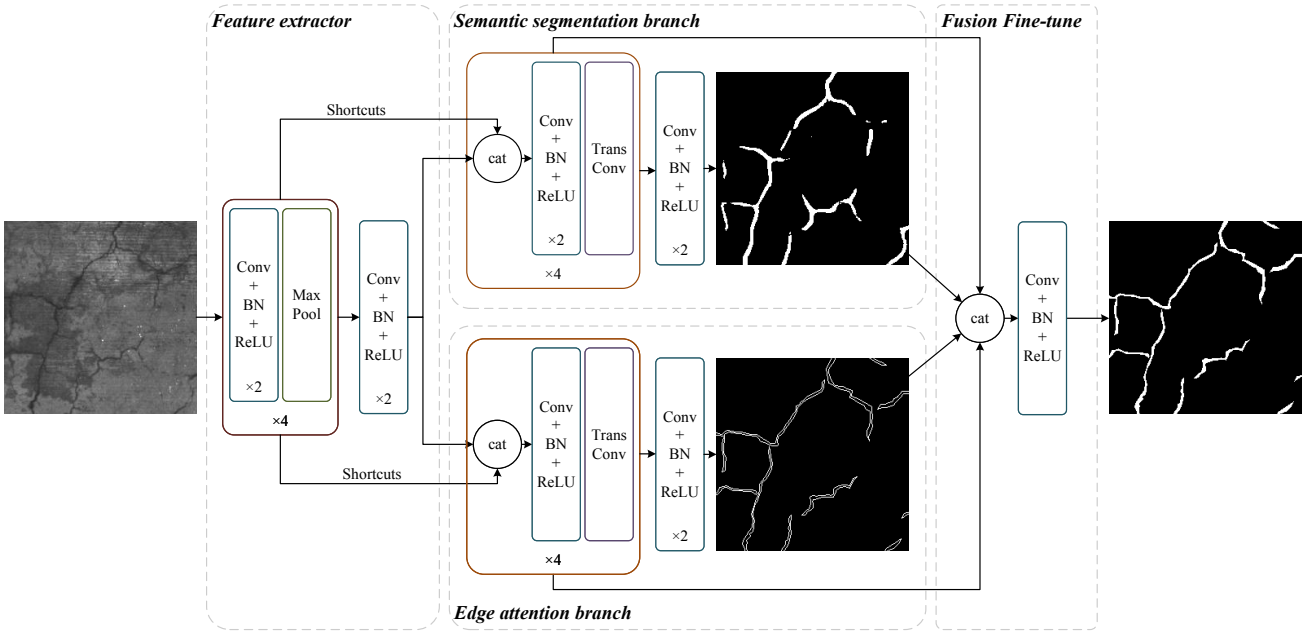

Fig. 2. Architecture of the proposed AggCrack model.

branch and edge attention branch. The feature extractor consists of 4 modules, and generates 5 feature maps in different scales. Each module of the feature extractor consists of 2 convolution layers and 1 maxpooling layer. In addition, batch-normalization and ReLU activation function are applied after each convolution layer, and they dramatically relief the case that large areas of noises overwhelm cracks which have much fewer pixels. Before maxpooling, the feature maps are output for shortcuts. After each module, the feature maps downsample in half scale, but double the channels for richer semantic features. Besides, the feature extractor with 4 downsampling layers ensures enough receptive field. The feature maps obtained by the extractor are fed into following two branches, respectively.

B. Edge Attention Branch

Edge features have relatively strong response when the cracks are seriously polluted by noises, and they are more prominent than internal features of cracks. In the process of image annotation, it is important to pay more attention to accurately identify the edges of targets, especially for the crack-like thin objects. However, the final annotation fills the target regions, which makes the edge information become weak for the model training.

In order to solve this problem, we build an edge attention branch to preserve the edge information for the following crack detection. It consists of 4 expansive modules to reconstruct the edges of target, and each module of the branch consists of 2 convolution layers and 1 transposed convolution layer. Besides, each expansive module takes feature map from shortcut as the supplement. Then, the branch employs sigmoid function to output a probability map.

The labels for edge attention are converted from the trajectories of manual annotations, and the trajectories are drawn in binary images.

The final probability map and the preceding feature maps are fed into the fusion module for more accurate crack detection.

C. Semantic Segmentation Branch

By observing the collected images, we notice that some cracks have similar characteristic with fuel stain and scratches in shape and texture, but the cracks are still demarcated by edges. Thus, the images are processed by semantic segmentation branch in advance, to output a preliminary segmentation result. Then, the feature maps from edge attention branch are fused into the segmentation result to complete crack detection.

In general, the architecture of expansive path in semantic segmentation branch is similar to the edge attention branch. The difference is that the segmentation branch is supervised by segmentation labels. In practice, we find that even human annotations have inevitable deviations. To accommodate the deviations, the labels are processed for 2 pixels tolerance.

D. Fusion Fine-tune Module

Considering that the single probability map given by edge attention branch are not rich enough to establish the relevance between edges and segmentation, the adjacent preceding feature maps are also fused by concatenating them. Thus, the fusion module takes the feature maps from the edge attention branch and the preceding preliminary segmentation branch, to generate the final detection results. In the AggCrack, we adopt a compact fusion fine-tune module for the reason

that segmentation regions are close to the crack edges. The fusion fine-tune module adopted in AggCrack consists of two consecutive convolution layers, and it is assigned to fill the edges and balance segmentation result, to connect the scattered cracks and confirm boundary.

E. Loss Function

The AggCrack outputs one-channel probability maps in each branch and the fusion module, and these probability maps have the same size of input image. Also, the total loss is summed by the three of them.

Value 1 indicates cracks or edges. Relatively, background is presented as value 0. With previously defined outputs, improved binary cross-entropy loss is used for calculating the distance between output and label, respectively. Considering the proportion of crack and edge pixels to background pixels is extremely small, and they are difficult to be identified, Focal Loss [16] is implemented to solve the unbalance between them. Essentially, the pixel-wise loss is formulated as

$$FL(y_i, \hat{y_i}) = -\begin{cases} \alpha_t(1-\hat{y_i})^\gamma \log \hat{y_i} & y_i = 1 \\ (1-\alpha_t)\hat{y_i}^\gamma \log(1-\hat{y_i}) & y_i = 0 \end{cases} \quad (1)$$

where i means the corresponding i-th pixel of the label and the output. $\hat{y_i}$ denotes the probability of being a crack pixel, y_i correspondingly denotes the label of input image. And both y_i and $\hat{y_i}$ range in $[0, 1]$, where 0 denotes background and 1 denotes target. And $(1-\hat{y_i})^\gamma$ and $\hat{y_i}^\gamma$ give the loss a motive that makes the model learn to classify difficult targets. α_t denotes the weights for balancing background and targets, it alleviates the extreme imbalance caused by the gamma term.

IV. Experiments

We have evaluated the AggCrack on a representative dataset collected by our inspection robot from civil aviation airports. To validate the superiority of the proposed algorithm, we have compared it with four state-of-the-art methods. Furthermore, ablation studies have been performed to evaluate the effectiveness of the different strategies.

A. Robotic Crack Detection System

We designed a robot equipped with various sensors to perform surface inspection task and underground defects detection. And it captures images with a Genie Nano M1920 Mono camera fixed on the inspection robot, as shown in Fig.1. The camera is installed downward, keeping the optical axis of the sensor to be perpendicular to the airport runway, and its resolution is set to 1800×900. Note that since we have to perform inspection during the night when the airport is closed, a LED area array lighting system is used on our robot.

B. Dataset

The dataset, named as APD, is collected by our inspection robot from four different airport runways. During inspection, the robot navigates within a predefined region along a grid route to collect images, as shown in Fig.1. When scanning, the robot transfers the camera images to the nearby data analysis center in a van using 4G/5G connection. Then the collected data will be automatically analyzed off-line.

We select 2000 typical images with cracks for APD dataset from more than a hundred thousand images initially captured. The resolution of images is 1800×900 pixels. The ground truth of cracks are labeled manually by two different experts individually. Cracks in APD dataset are generally extremely thin, some are even only one-pixel width. Corresponding to the real world, the cracks in APD dataset are even less than 1mm wide, and they are distributed from 0.5mm to 2cm. Due to the serious pollution by fuel stain and aircraft wheel mark, this dataset is quite challenging.

APD dataset has been randomly divided into training and test sets with a ratio of 7:3. So, there are 1400 images for training AggCrack and other comparison models, 600 images for test. With the dataset ready, we now discuss the implementation details next.

C. Implementation Details

We have implemented our AggCrack algorithm with the deep learning framework PyTorch. The convolution layers use 3×3 kernels and pad 1 pixel. The weights of all the convolution layers are initiated by kaiming uniform [17]. The maxpooling layers use 2×2 kernels and stride 2 to reduce the size of feature maps. The transposed convolution layers are set as 2×2 kernels and stride 2 correspondingly. Adam is implemented to optimize the models. ReLU is the first choice for the non-linear activation. Furthermore, LeakyReLU is tested in the following ablation studies.

All the experiments are run on a GeForce RTX 3090 GPU. Considering the GPU RAM limitation, the images are mirror padded to a multiple of 512 in width and height. Then, each image is cut into several sub-image with resolution of 512×512 pixels, which will be fed into the AggCrack model. In the practice of training, the images are augmented with rotating in 90, 180, 270 degrees and flipping in horizon and vertical. The initial learning rate is set to be 0.001. The model is trained from scratch, and if there is no performance improvement, the training will stop after 10 epochs. Usually, the model is trained for about 20 epochs. In the inference stage, the probability maps are affirmed into 0 and 1 with the threshold being 0.5.

D. Compared Methods and Evaluation Metrics

To evaluate the performance of the AggCrack, we compare it with four state-of-the-art deep learning based crack detection algorithms, including:

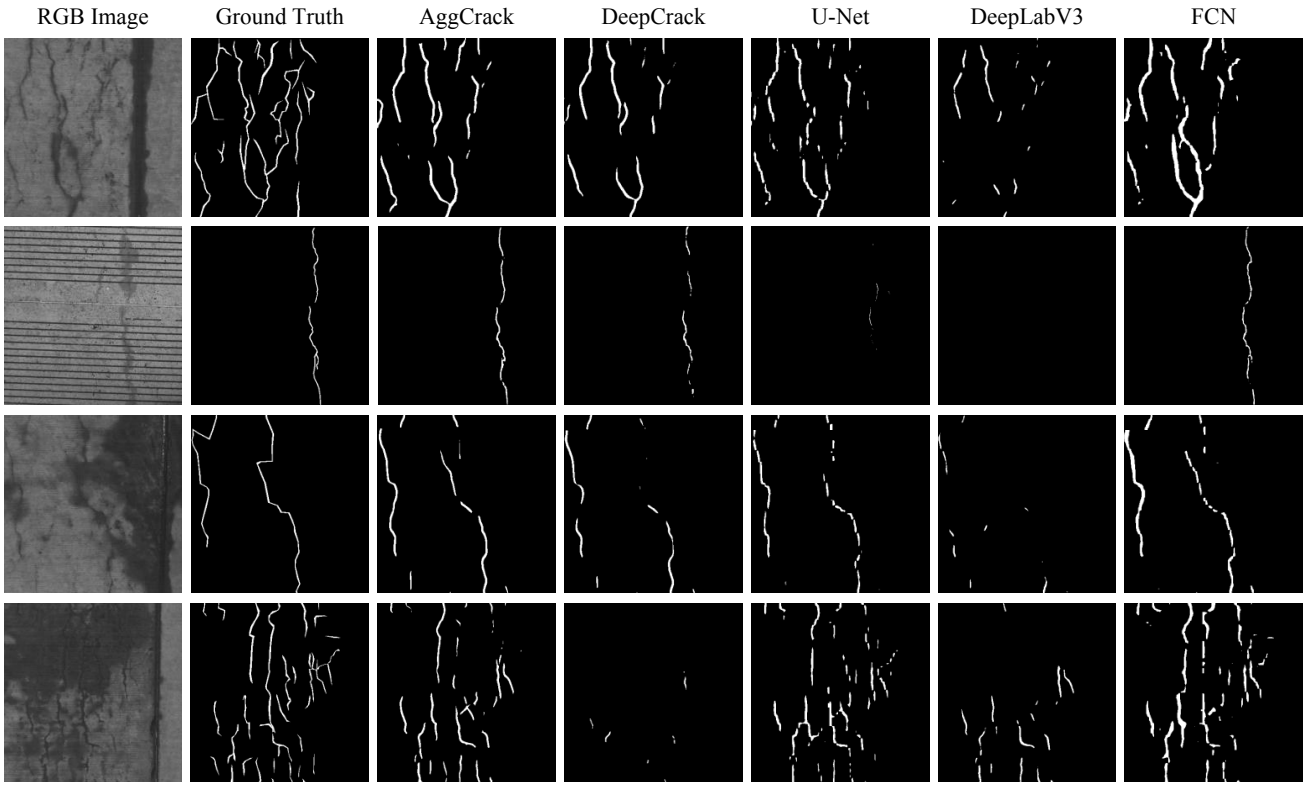

Fig. 3. Examples results of different algorithms on APD dataset.

- U-Net [13]. U-Net performs skip connections for lower feature fusion, which gives a result with fine boundary on retinal vessel segmentation.
- FCN [18]. Fully Convolutional Networks for Semantic Segmentation (FCN) implements multi-scale feature map upsampling for accurate segmentation.
- DeepLabV3 [19]. DeepLabV3 employs atrous convolution in cascade to boost the performance of segmentation.
- DeepCrack [20]. DeepCrack employs deep supervision from lower feature maps to higher feature maps, and gives an accurate crack segmentation.

To evaluate the performances of them quantitatively, three common metrics, including $Precision$, $Recall$ and $F1$-$measure$, are employed. Considering acquiring an exactly the same prediction map as the ground truth is difficult for a image of resolution of 1800×900, we allow two pixels deviation as the tolerance for the metrics.

E. Results on APD dataset

We test the AggCrack with its counterparts listed above on APD dataset. And the results are summarized in Tab.I, where the results of all compared methods are obtained using their respective open source codes. Fig.3 presents representative images and their detection results where we can find the images in APD dataset are quite challenging. From the experiment results, the AggCrack outperforms all counterparts in overall performance. Although DeepCrack achieves the best performance on precision benefiting from its supervision by multi-scale feature maps, its recall is quite low, which indicates the probability of missed detection is high. For airport runway inspection task, it is obvious that the higher recall value is desired, because any missed cracks are potential risks for aircraft safety. Our AggCrack model achieves 4.74% lower than DeepCrack on precision, but 14.56% higher on recall. Compared with the FCN model, our network takes two convolution layers in each expansive module, which results in better precision, as the expansive module has the process of adjusting the reconstruction. In practice of training DeepLabV3, we find that the atrous convolution is not well fitted in the crack detection task. One possible reason is that the offset of atrous convolution may not fall into the target regions correctly due to the thinness of cracks. In terms of computing consumption, AggCrack runs in 2.67 FPS on one RTX 2080Ti, which satisfy the needs of the robot inspection task.

F. Ablation Studies

We conduct extensive ablation experiments to analyze the effectiveness of our design choice in AggCrack. Tab.II shows the effects of different functions on the AggCrack model, where 'w/o' indicates that the corresponding function is removed from the model, and 'w/' means that the function is adopted to replace the original operation.

TABLE I

CRACK DETECTION RESULTS ON APD DATASET.

Method	Precision	Recall	F1-measure
FCN	76.85%	69.48%	72.98%
DeepLabV3	83.58%	61.51%	70.87%
U-Net	83.28%	67.60%	74.62%
DeepCrack	**88.98%**	55.80%	68.59%
AggCrack	84.24%	**70.36%**	**76.68%**

TABLE II

ABLATION STUDIES ON MODULES FROM AGGCRACK.

Method	Precision	Recall	F1-measure
AggCrack	84.24%	**70.36%**	**76.68%**
w/o fusion fine-tune	87.98%	63.94%	74.06%
w/ stride conv	87.46%	62.86%	73.15%
w/ LeakyReLU	85.92%	63.32%	72.91%
w/ stride conv and LeakyReLU	**92.15%**	56.61%	70.14%

First, we remove the fusion fine-tune module to find out if the module can directly use the feature maps generated from edge attention branch. Tab.II shows that the model achieves higher precision. A reasonable explanation is that the model makes an intersection over the segmentation and edge rather than takes leverage of edges for better segmentation. Second, considering that the maxpooling may lead to the loss of some insignificant features of cracks, we adopt the stride convolution to learn a better self-adaption pooling. The result indicates that it makes the model tend to extract significant features rather than get sufficient features. Then, we notice that the ordinary ReLU activation function leads to more dying neurons, which causes gradient instability in the training stage. As shown in Tab.II, the LeakyReLU has no significant improvement on the AggCrack model, and even decrease the recall, but accelerate training by $10\times$ faster, which converges in 2 epochs. Relatively, AggCrack usually converges at 20 epochs without LeakyReLU. Finally, we have implemented both stride convolution and LeakyReLU on AggCrack, and the precision comes to 92.15% unprecedentedly, but the cost is the significant decrease of recall.

V. CONCLUSION AND FUTURE WORK

We proposed an aggregated attention CNN model, AggCrack, to detect cracks automatically for the robotic inspection of airport runway. Our AggCrack novelly proposes the semantic-level attention mechanism to leverage the edge information to combine with crack semantic information to robustly find cracks even in the presence of significant noise level. We extensively tested our algorithm with real airport runway data collected from four different airport runways. The comparative results demonstrated that the AggCrack can effectively detect the airport runway cracks and had outperformed the state-of-the-art techniques.

In the future, we plan to fuse the 2D camera images with 3D laser ranger finder inputs to further improve the detection performance.

REFERENCES

[1] Q. Li and X. Liu, "Novel approach to pavement image segmentation based on neighboring difference histogram method," in *2008 Congress on Image and Signal Processing*, vol. 2, pp. 792–796, IEEE, 2008.

[2] M. Avila, S. Begot, F. Duculty, and T. S. Nguyen, "2d image based road pavement crack detection by calculating minimal paths and dynamic programming," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 783–787, IEEE, 2014.

[3] H. Li, D. Song, Y. Liu, and B. Li, "Automatic pavement crack detection by multi-scale image fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2025–2036, 2018.

[4] R. S. Lim, H. M. La, Z. Shan, and W. Sheng, "Developing a crack inspection robot for bridge maintenance," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 6288–6293, IEEE, 2011.

[5] R. S. Lim, H. M. La, and W. Sheng, "A robotic crack inspection and mapping system for bridge deck maintenance," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 367–378, 2014.

[6] J. Zhou, P. S. Huang, and F.-P. Chiang, "Wavelet-based pavement distress detection and evaluation," *Optical Engineering*, vol. 45, no. 2, p. 027007, 2006.

[7] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3000–3009, 2017.

[8] F.-C. Chen and M. R. Jahanshahi, "Nb-cnn: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392–4400, 2017.

[9] X. Yang, H. Li, Y. Yu, X. 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.

[10] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature pyramid and hierarchical boosting network for pavement crack detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1525–1535, 2019.

[11] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403, 2015.

[12] S. L. Lau, E. K. Chong, X. Yang, and X. Wang, "Automated pavement crack segmentation using u-net-based convolutional neural network," *IEEE Access*, vol. 8, pp. 114892–114899, 2020.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[14] 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*, pp. 770–778, 2016.

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

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[18] 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*, pp. 3431–3440, 2015.

[19] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[20] Q. Zou, Z. Zhang, Q. Li, X. 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, 2018.