ECSNet: A Real-time Image Segmentation Algorithm for Pavement Crack Detection

Tianjie Zhang, Donglei Wang, and Yang Lu\*

[[1]](#footnote-1)

***Abstract*—The ability to perform pixel-wise segmentation on pavement crack in real-time is of paramount importance in road quality assurance and quality control. Recent deep learning detection models are focused on the detection accuracy and require a large number of computing sources and long run times. In this work, we present an efficient deep learning model called Efficiency Crack Segmentation Neural Network (ECSNet) for the real-time pavement crack detection and segmentation task. An ECS-block which is constructed by a combination of 1×1 Conv2D, 3×3 Conv2D and 1×1 Conv2D is introduced for crack information extraction and model parameter reduction. The parallel max pooling and convolutional operation are introduced to speed up the downsizing and filtering process. We present experiments on latency and accuracy tradeoffs on the DeepCrack Dataset. Our proposed ECSNet shows a strong performance both in accuracy and efficiency compared to other state-of-the-art models including DeepLabV3, FCN, LRASPP, Enet and DeepCrack. The results show that ECSNet obtains the second place on the F1 score (F1) and Intersection over Union (IoU). It reaches the largest FPS and lowest training time among all the models. It promises a trade-off between the accuracy and efficiency metrics.**

***Index Terms*—** **Pavement crack detection; real-time task; image segmentation; deep learning**

# I. INTRODUCTION

C

rack is a common defect of pavement which directly affects pavement service ability and driving safety [1]. Traditionally, the cracks are counting and measuring by engineers, but it is time-consuming and labor-intensive [2]. With the development of the image-based technologies, pavement crack detection approaches have achieved critical improvements in accuracy and reliability [3]. Deep learning becomes the most interesting and advanced method to detect pavement cracks as it learns from large-scale data and requires little human involvement during training [4]. Crack classification and segmentation are two main focuses on the pavement cracks detection. Segmentation is a pixel-level extraction of crack from the background which is a more complex task than classification [5].

The general trend of improving deep learning-based segmentation models is to include deeper layers and make more complicated structures in order to achieve high accuracy. For example, Sun et al. [6] adopted and enhanced a DeepLabV3 model by attention module which can assign weights between the high-level and low-level feature maps, and it achieved a higher Intersection over Union (IoU) on the DeepCrack, Crack500 and FMA datasets. Qu et al. [7] proposed a deep learning-based network with hierarchical feature fusion and connected attention architecture to enhance the feature extraction and eventually increase the accuracy of the proposed method (got a F1 score of 0.86). Chen et al. [8] proposed an encoder-decoder structural model based on a modified SegNet and got a mean Pixel Accuracy (mPA) of 83% which was higher than FCN-8s and MRCNN on a self-collected dataset.

Another trend to improve the model performance is to increase the amount and diversity of the datasets. Images augmentation methods including traditional image augmentation [9] and Generative Adversarial Networks (GAN) [10-12] were popularly utilized before training to improve the model’s performance. Some researches merged the visual images with other kind source data to improve the accuracy of their model. For example, Liu et al. [13] fused the visual image and thermal image together to increase the accuracy of classifying asphalt pavement crack severity. Liu et al. [14] combined the Ground Penetrating Radar B-Scan data and pavement images to improve the accuracy of crack detection.

However, it is not necessarily making deep learning-based segmentation models more efficient with respect to size and speed by constructing complicated structures or combine multi-source data [15]. In the pavement crack detection, the segmentation tasks need to be carried out in a timely fashion on a computationally limited platform. A high-accuracy but computing-costly model is hard to perform well in a real-life mobile application.

A model that makes a good trade-off between the accuracy and efficiency can be helpful to handle the real-time tasks which is very significant to the decision-makers and engineers. Thus, there are rising interests and needs in developing small and efficient neural networks for the intelligent transportation system. However, there were only limited researches involving in the efficient pavement detection methods. Pang et al. [16] proposed a Deep Crack Segmentation Network (DcsNet) by incorporating two feature extraction branches to achieve the balance between speed and accuracy. It reached an IoU of 58.5 and an FPS of 67.5 on Crack500. Wang et al. [1] proposed a lightweight road crack segmentation model based on a bilateral segmentation network. The model was trained and tested on Crack500. It got an IoU of 73.79 and FPS of 31.3. The Enet proposed by Adam [17] changed the previous encoder-decoder symmetrical structure, reduced the convolution operation in the decoder, and enhanced the processing speed tremendously. It introduced the convolutional layer in the structure to reduce the dimensionality and parameters of the model without reducing the accuracy and achieved a higher mean Intersection over Union (mIoU) than SegNet [18].

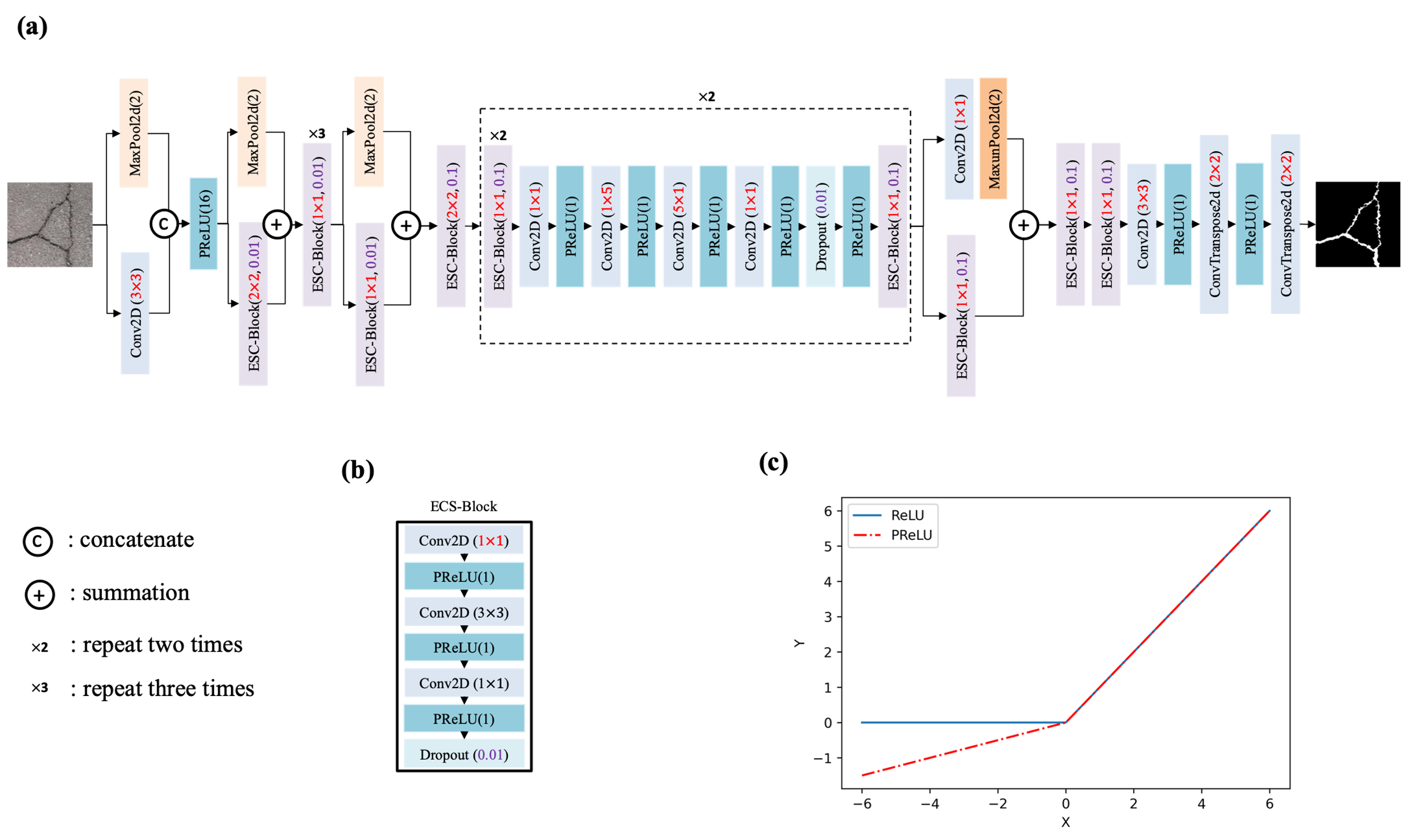
Due to the importance of the real-time job for engineer and decision maker, an Efficient Crack Segmentation Neural Network (ECSNet) is proposed in this work to improve the computing efficiency of the pavement crack segmentation tasks. While increasing the run speed, the network also keeps a trade-off between accuracy and efficiency. An ECS-block which is constructed by a combination of Conv2D, Conv2D and Conv2D is introduced for crack information extraction and model parameter reduction. The parallel max pooling and convolutional operation are also introduced in the network to speed up the downsizing and filtering process. The proposed ECSNet is tested on the DeepCrack dataset and is compared with other state-of-the-art models including DeepLabV3 [19], FCN [20], LRASPP [21], Enet [17] and DeepCrack [22]. The result shows that our model performs a good trade-off between the accuracy and efficiency, which means it can be utilized into the real-time pavement crack detection and measurement tasks.

The paper is organized in the following way: The details of the proposed ECSNet and the experiment design are described in Section II; Section III demonstrates and analyzes the experimental results. Section IV is the conclusion.

# II. Methodology

## A. ECSNet

Efficiency Crack Segmentation Neural Network (ECSNet) is proposed in this work to improve the efficiency of the pavement crack detection tasks without decreasing the accuracy of the segmentation result. The architecture of the proposed ECSNet is shown in Figure 1.



**Figure 1.** The architecture of the proposed ECSNet and its components: (a) The architecture of the proposed ECSNet; (b) The structure of an ECS-Block which is consisting of three convolutional layers: 1×1 Conv2D, 3×3 Conv2D and 1×1 Conv2D. The PReLU is used as an activation function in the ECS-Block; (c) A comparison between ReLU and PReLU.

The proposed ECSNet contains 58 convolutional layers in total. Some techniques used in the network to avoid redundant and accelerate training speed are described below:

(1) The distribution of each layer’s inputs changes during training due to the changing parameters of the previous layer. It slows down the training procedure as the layer would require parameter initialization. Therefore, each convolution layer in ECSNet is followed by a Batch Normalization layer [23] to standardize the parameters. The operation in Batch Normalization layer is shown in Equation (1).

(1)

Where x is the input, y is the output, E[x] is the mean value of x, Var[x] is the variance of x, is a value added to the denominator for numerical stability, and are learnable parameters during the training.

By applying the Batch Normalization, the model can standardize the parameters efficiently and speed up the training procedure.

(2) Instead of using large kernels directly, we use small kernel convolutional layers like kernel convolutional layer (Conv2D) in the architecture to downsize the parameters first. We introduce the ECS-Block, a combination of Conv2D, Conv2D and Conv2D, which can be regarded as decomposing a large-kernel convolutional layer into a series of smaller operations. It can obviously reduce the number of parameters and make the process less redundant compared to using large convolutional operations.

The parameters in a convolutional layer can be calculated through Equation (2)

(2)

Where P stands for parameters, H is the height of the filter, W is the width of the filter, D is the number of filters in the previous layer, B is the bias (B= 0 in this work), K is the number of filters in this layer.

Thus, the ratio of the total parameter included in the ECS-Block and a standard Conv2D can be calculated using Equation (3).

(3)

Where stands for the ratio of parameters from ECS-Block and a Conv2D. D is 64 and K is 16 in this work. Therefore, based on the Equation (3), the computational cost from the proposed ECS-Block is 8.5 times less than a standard convolution.

In addition, a dropout layer is applied at the end of the ECS-Block to regularize the result from the convolutional layer to avoid overfitting problems.

There are two parameters for the BCS-Block. The first parameter stands for the kernel size of the first convolutional layer in the block. The second parameter stands for the p value in the Dropout layer.

(3) The ECSNet does not use a normal encoder-decoder symmetrical structure. In order to speed up the prediction, the convolutional operation is reduced in the decoder part which enhances the processing speed tremendously.

(4) Each  convolutional layers can be decomposed into two smaller ones: one with a kernel and one with a kernel. This idea has been worked and proved in Adam’s work [17]. The cost of the operation is lowered but the variety of operations increased which can increase the ability of network to learn more various details about the road images.

The computational cost can also be calculated through Equation (2). The utilized method can reduce the parameters about times lower than a standard convolutional layers as shown in Equation (4)

(4)

Where stands for the ratio of parameters contained in a convolutional layer and a combination of Conv2D and Conv2D.

(5) For a real-time detection job, it is necessary to down sample the input image quickly after inputting the training image. However, an aggressive dimensionality reduction in operation can hinder the spatial information transferring. Thus, we use a convolutional layer with a filter to downsize the input firstly, rather than using a filter directly on the input image to sharply lower its size. After that, a convolutional layer with a kernel size is used to downsize the pavement images again. A kernel size of makes sure that we can take the full input as consideration from a filter and decrease the information loss during the training.

(6) It is common to use a max pooling layer right after a convolutional layer to downsize the image. However, it would lower the computational efficiency. In this work, we perform the max pooling operation in parallel with a convolutional layer, and concatenate them together. It can merge the feature maps generated by pooling and convolution operations which will also speed up the inference time without losing information.

(7) The PReLU is used as the nonlinear layer in the network. The equation of PReLU is shown below.

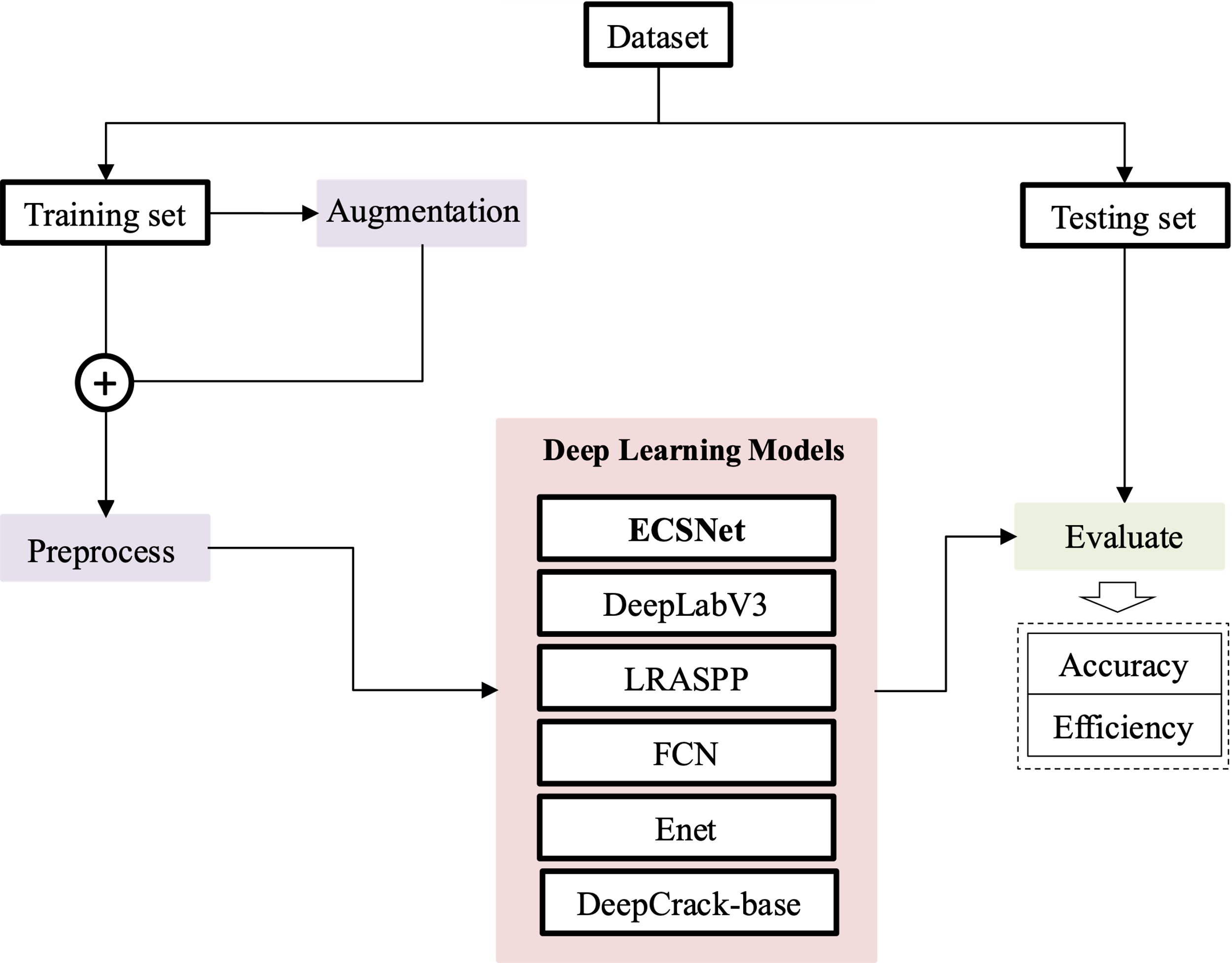
(5)

Where A is a learnable parameter, the initial value of A is 0.25. x means the number of layers to input into the activation function.

As shown in Figure 1(c), the PReLU can give a learnable weight to the small values while they are totally ignored in the ReLU function. The reason to use PReLU rather than ReLU is that the crack in pavement is the relative dark part in the background. Thus, using a PReLU can keep the crack information more accurate when filtering the images.

## B. Overall procedure

The overall procedure to evaluate the performance of the ECSNet and comparison with other popular deep learning-based segmentation models are shown in Figure 2.



**Figure 2.** The overall procedure to evaluate the performance of the deep learning neural networks.

A public dataset named DeepCrack is utilized to evaluate the accuracy and efficiency of the deep learning models. The DeepCrack dataset is an open-source dataset published in GitHub (https://github.com/yhlleo/DeepCrack). This dataset consists of 537 RGB crack images with manually annotated segmentations. The image has a resolution of 544 × 384 pixels. In this work, the dataset is randomly divided into training data and testing data at a ratio of 8:2. Then, the images in the training dataset are augmented by data-augmentation methods include random crop, flip and rotate. We perform the same data augmentation operation on the image and the ground truth of the image. These augmented images will be totally new inputs to the neural network. By doing this, the deep learning-based model’s performance would be improved because deep learning algorithm is deep related on the amount and diversity of datasets it used. After augmentation, the training set is processed by changing the color to gray. In order to reduce the computing source, all the images and their ground truths are resized to pixels before input into the deep learning models.

Five popular models are also tested on this dataset and compared with our proposed ECSNet. Models includes DeepLabV3, LRASPP, Enet, DeepCrack. The details of these models are shown below.

(1) DeepLabV3: We adapt the ResNet-50 to the original DeepLabV3 structure by applying convolution to extract dense features.

(2) LRASPP: We adapt a Lite R-ASPP Network model with a MobileNetV3-Large backbone.

(3) FCN: We adopt FCN network with a Resnet50 [24] backbone.

(4) Enet: A deep neural network architecture for real-time semantic segmentation proposed by Adam. All the hyperparameters are set as default.

(5) DeepCrack: This model was developed by the Zou who publish the DeepCrack Dataset. The model was originally designed for the DeepCrack Dataset and all the hyperparameters are set as default.

All models are trained using Root Mean Squared Propagation (RMSProp) [25] with a momentum 0.9 as optimizer to update the network. A learning rate of 1e-4 and a batch size of 16 are utilized during the training. BCEWithLogitsLoss [26] is used as the loss function, which combines a sigmoid layer and the Binary Cross Entropy in one single class. This loss is more numerically stable than using a plain Sigmoid followed by a BCELoss [5]. Totally 300 epochs are trained on each network. The pixels whose output probability value is lower than 0.5 are classified as crack, and others are identified as background. Each model is trained and tested three times to make the results statistically meaningful.

The data augmentation methods and CNN models are all implemented in Python and computed under the following machine speculations: Windows 10, Intel(R) Core (TM) i9-10900X CPU, NVIDIA RTX A4000 with 16 GB memory, 64GB RAM.

## D. Evaluation Metrics

The goal of a real-time oriented deep learning model is making a good trade-off between accuracy and efficiency. Thus, in this paper, we have to evaluate and compare both accuracy metrics and efficiency metrics.

For the accuracy part, F1 score (F1) and Intersection over Union (IoU) are utilized to evaluate the semantic segmentation results.

F1 is defined based on the harmonic average of Precision and Recall as shown in Equation (6). The precision and recall can be calculated using Equation (7) and (8).

(6)

(7)

(8)

Where TP denotes True Positive, FP is False Positive, FN is False Negative, P is precision, R is recall and F1 denotes F1 score. F1 score is a more reasonable metric than Precision or Recall according to Equation (6).

The IoU measures the overlap between the prediction result and the ground truth. It is used to measure how much the predicted areas overlaps with the ground truth. IoU is calculated according to Equation (7).

(7)

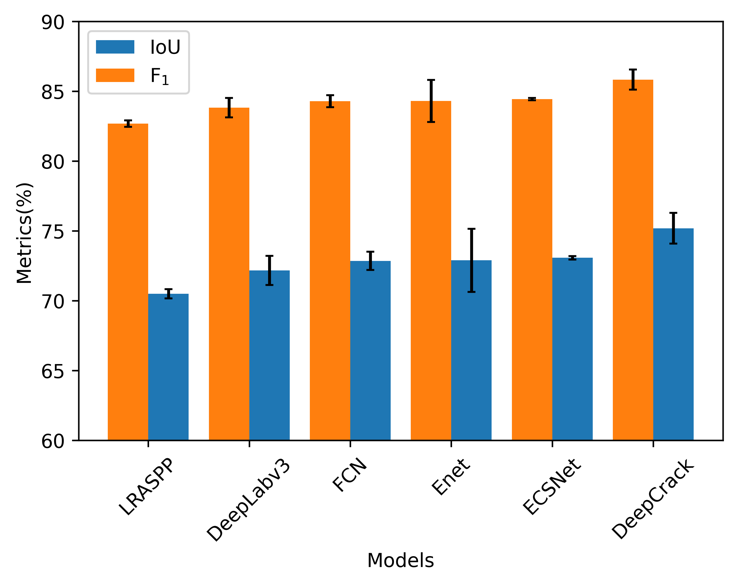
Where represents the number of pixels belonging to class i but predicted as class j. The IoU represents how much pixels are correctly predicted and it is a common metric in the image segmentation tasks.

For the efficiency part, the training time of each model is measured to show how much computational resources the model needed in training. The frames per second (FPS) in the testing procedure are used to evaluate the efficiency of the models in predicting. The FPS can be an index for the real-time jobs. A higher FPS indicates the model can process more images in a unit time, which means it will have a better performance in real-time tasks. The number of parameters is also calculated from each model as it is an important factor in a mobile device. A lightweight model can be easier to be deployed in a smart phone.

# III. Results

## A. Accuracy evaluation

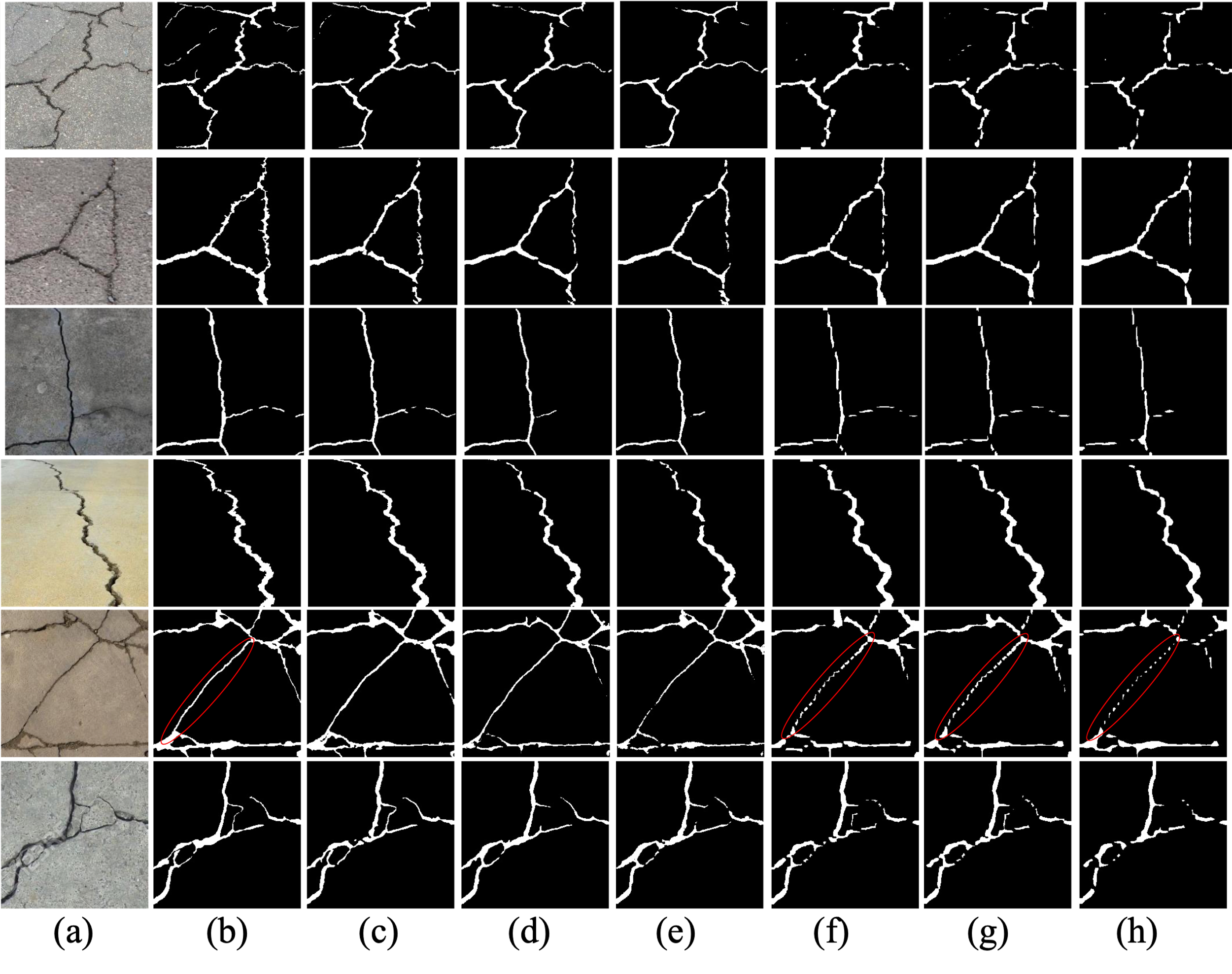
Figure 3. shows the F1 score and IoU values from the models tested in this work, including DeepLabV3, LRASPP, FCN, Enet, DeepCrack and ECSNet.



**Figure 3.** The F1 score and IoU of all the testing models.

It shows that the DeepCrack gets the highest F1 score (85.84%) and IoU values (75.19%) among all the models. It is noteworthy to find that the ECSNet gets the second place in both F1 score (84.45%) and IoU value (73.08%). The variance of the testing results shows that ECSNet gets a lowest variance among all the approaches. In other words, the performance of ECSNet is more stable than other methods.

Some visualization of detection results of these deep learning-based models is shown in Figure 4 to show their segmentation performance.

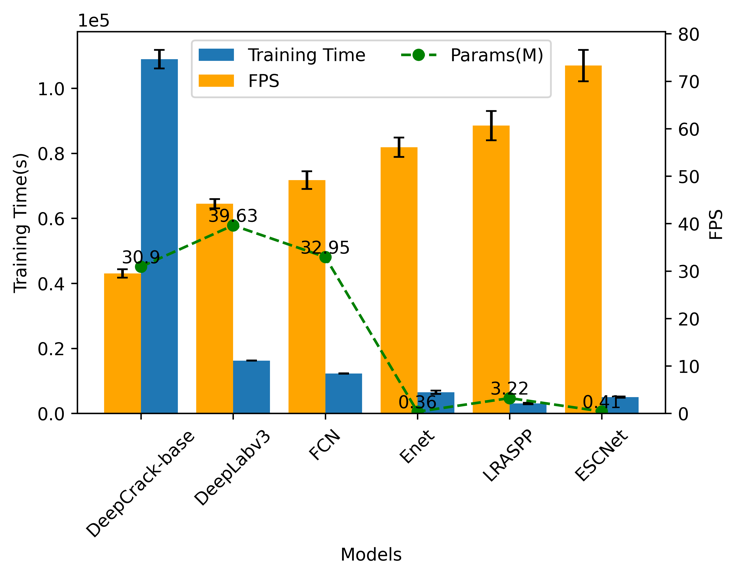


**Figure 4.** The visualization of detection results of compared methods on the DeepCrack Dataset: (a) original image; (b) ground truth; (c) DeepCrack; (d) ECSNet; (e) Enet; (f) FCN; (g) DeepLabV3; (h) LRASPP.

As we can see from Figure 4., the results from FCN, DeepLabV3 and LRASPP are intermittent compared to the ground truth especially in the red-circle part. The segmented images of DeepCrack and ECSNet are quite consistent with the ground truth compared to other models. It shows that the segmentation results from DeepCrack are the closest to the ground truth.

## B. Efficiency evaluation

Figure 5 shows the required training time for each model in the training process and the Frames per Second (FPS) when using the trained model to predict.



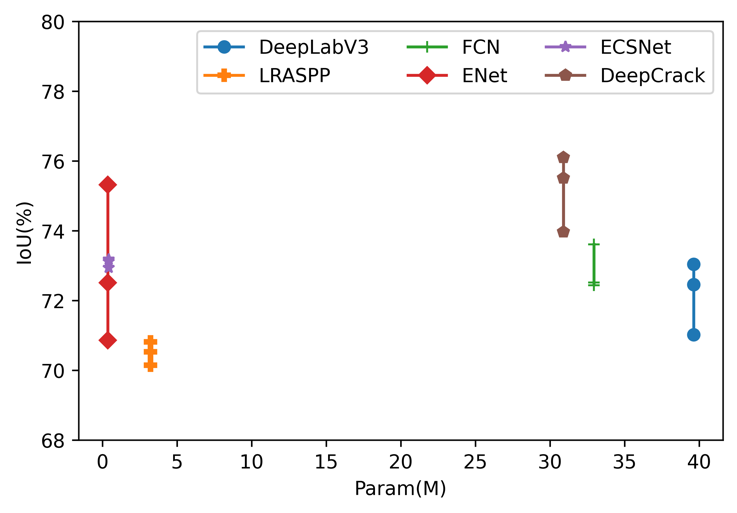
**Figure 5.** The training time, FPS and parameters of each model.

It is noteworthy that the training time each model spend is quite consistent with the FPS. In other words, a model’s complexity determines its efficiency, no matter in training or predicting. The DeepCrack consumes the highest time to train its parameters and also has the lowest FPS. The proposed ECSNet has the highest FPS among all the models which shows its high efficiency in pavement crack segmentation. It is 2.5 times faster than the DeepCrack model.

The parameters contained in each model are calculated and shown in Figure 5. There is no obvious linear relationship between the size of parameters and the efficiency. However, it is obvious that the lightweight models including Enet, LRASPP and ECSNet, need an averagely lower training time than the complex deep learning-based models (DeepCrack, DeepLabV3 and FCN). Moreover, the lightweight model has a higher FPS in predicting.

## C. Relationship between the accuracy and efficiency

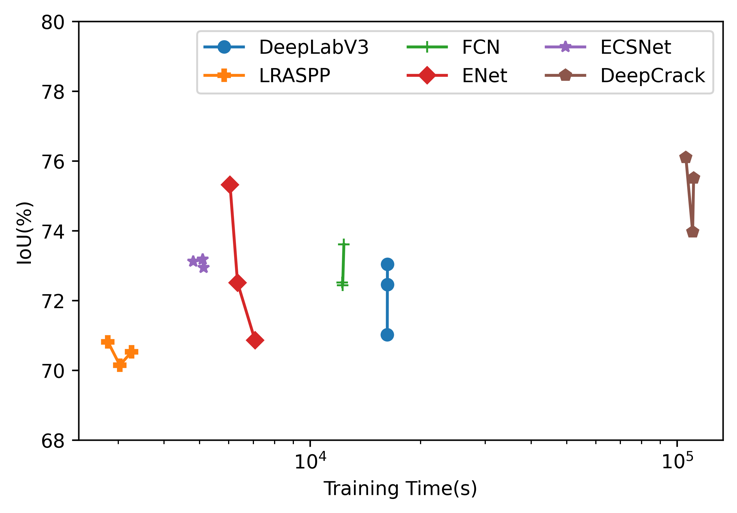
In order to give an inspect view to the performance of each method, the accuracy and efficiency are considered together to show the model’s capacity of making a good balance between them. The IoU is used as the main index to represent the model’s performance in accuracy. The IoU is compared with model’s parameters, FPS and training time as shown in Figure 6-8. Each model is trained and tested three times on the dataset to make the results meaningful in statistic. Therefore, there are three points for each model. Figure 6 shows the relationship between the IoU and parameters in the model.



**Figure 6.** The relationship between IoU and number of parameters in each model.

The parameters contained in a model is an index to show its proper to be deployed in a mobile device. In other words, a light weight model is more reasonable for a mobile usage. Figure 6 shows that the Enet and ECSNet has the nearly same parameter amount (0.36M for Enet and 0.41 M for ECSNet), which is much smaller than other models. The average IoU value of Enet and ECSNet is 72.86% and 73.08%, respectively. The accuracy of ECSNet is a little higher than Enet. Furthermore, the variance of IoU from Enet (2.26%) is much higher than ECSNet (0.12%). It means the performance from Enet is not stable enough like ECSNet.

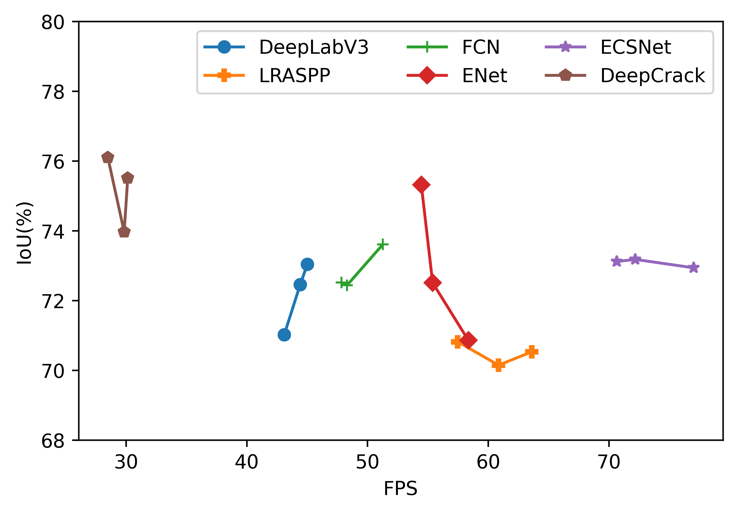
Figure 7 shows the relationship between the model’s training time and IoU.



**Figure 7.** The relationship between IoU and Training Time of each model.

The training time shows the model’s consumption on the computational source in order to train its parameters. It is an important factor as the model is always waiting to be updated with the increasing dataset. This is because updating the parameter can help improve the performance. The results show that the ECSNet only consumes about 5000 seconds to reach a second place in the average IoU value. Compared with other segmentation algorithms, our proposal approach achieves a trade-off between the amounts of training time and accuracy. The LRASPP uses the lowest time (about 2800 seconds) in training procedure. However, the IoU metric is quite low (70.5%).

Figure 8 shows the relationship between IoU and FPS in each deep learning method.



**Figure 8.** The relationship between IoU and FPS in each model.

As we can see from Figure 8, the proposed ECSNet get an absolutely higher FPS (73.3) than other models. The proposed method shows great potential in real-time applications as the FPS is an important index for pavement crack real-time detecting. However, it worth noticed that the IoU of ECSNet doesn’t drop when the network pays more attention to improving its performance on the information compression and detection speed. In other words, our designed network makes a trade-off between the model’s accuracy metrics and real-time requirements.

# IV. Conclusion

In this work, an Efficiency Crack Segmentation Neural Network (ECSNet) is proposed for real-time pavement crack detection tasks. The proposed network adopts some technics to downsize the model structure without losing features. An ESC-Block is introduced replacing the large-scale filters in the deep learning structure in order to reduce the parameters. This model is designed as a lightweight real-time semantic segmentation model with high computational efficiency.

According to the overall performance on the DeepCrack Dataset, the ECSNet outperforms all the popular models except DeepCrack model in F1 score and IoU. The segmentation result from ECSNet is clearer and more continues than DeepLabV3, LRASPP and FCN. It is also demonstrated that the ECSNet has the highest FPS and lowest training time among all trained models. Comparing the IoU with efficiency metrics, we find that the proposed ECSNet keeps a trade-off between the accuracy and efficiency metrics. It means ECSNet is a well-designed real-time image segmentation algorithm for the pavement crack detection task.

References

[1] W. Wang and C. Su, "Deep learning-based real-time crack segmentation for pavement images," *KSCE Journal of Civil Engineering,* vol. 25, no. 12, pp. 4495-4506, 2021.

[2] J. Liu *et al.*, "Automated pavement crack detection and segmentation based on two‐step convolutional neural network," *Computer‐Aided Civil and Infrastructure Engineering,* vol. 35, no. 11, pp. 1291-1305, 2020.

[3] T. Wen, H. Lang, S. Ding, J. J. Lu, and Y. Xing, "PCDNet: Seed Operation–Based Deep Learning Model for Pavement Crack Detection on 3D Asphalt Surface," *Journal of Transportation Engineering, Part B: Pavements,* vol. 148, no. 2, p. 04022023, 2022.

[4] S. Luo, J. Yao, J. Hu, Y. Wang, and S. Chen, "Using Deep Learning-Based Defect Detection and 3D Quantitative Assessment for Steel Deck Pavement Maintenance," *IEEE Transactions on Intelligent Transportation Systems,* 2022.

[5] T. Zhang, D. Wang, A. Mullins, and Y. Lu, "Integrated APC-GAN and AttuNet Framework for Automated Pavement Crack Pixel-Level Segmentation: A New Solution to Small Training Datasets," *IEEE Transactions on Intelligent Transportation Systems,* 2023.

[6] X. Sun, Y. Xie, L. Jiang, Y. Cao, and B. Liu, "Dma-net: Deeplab with multi-scale attention for pavement crack segmentation," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 10, pp. 18392-18403, 2022.

[7] Z. Qu, C.-Y. Wang, S.-Y. Wang, and F.-R. Ju, "A method of hierarchical feature fusion and connected attention architecture for pavement crack detection," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 9, pp. 16038-16047, 2022.

[8] T. Chen *et al.*, "Pavement crack detection and recognition using the architecture of segNet," *Journal of Industrial Information Integration,* vol. 18, p. 100144, 2020.

[9] Z. Xu *et al.*, "Pavement crack detection from CCD images with a locally enhanced transformer network," *International Journal of Applied Earth Observation and Geoinformation,* vol. 110, p. 102825, 2022.

[10] Y. Hou *et al.*, "A deep learning method for pavement crack identification based on limited field images," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 11, pp. 22156-22165, 2022.

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

[12] B. Xu and C. Liu, "Pavement crack detection algorithm based on generative adversarial network and convolutional neural network under small samples," *Measurement,* vol. 196, p. 111219, 2022.

[13] F. Liu, J. Liu, and L. Wang, "Deep learning and infrared thermography for asphalt pavement crack severity classification," *Automation in Construction,* vol. 140, p. 104383, 2022.

[14] Z. Liu, X. Gu, J. Chen, D. Wang, Y. Chen, and L. Wang, "Automatic recognition of pavement cracks from combined GPR B-scan and C-scan images using multiscale feature fusion deep neural networks," *Automation in Construction,* vol. 146, p. 104698, 2023.

[15] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861,* 2017.

[16] J. Pang, H. Zhang, H. Zhao, and L. Li, "DcsNet: a real-time deep network for crack segmentation," *Signal, Image and Video Processing,* pp. 1-9, 2022.

[17] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147,* 2016.

[18] 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,* vol. 39, no. 12, pp. 2481-2495, 2017.

[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] 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.

[21] A. Howard *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314-1324.

[22] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "DeepCrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing,* vol. 338, pp. 139-153, 2019.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015: pmlr, pp. 448-456.

[24] 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.

[25] T. Kurbiel and S. Khaleghian, "Training of deep neural networks based on distance measures using RMSProp," *arXiv preprint arXiv:1708.01911,* 2017.

[26] E. Salcedo, M. Jaber, and J. R. Carrión, "A Novel Road Maintenance Prioritisation System Based on Computer Vision and Crowdsourced Reporting," *Journal of Sensor and Actuator Networks,* vol. 11, no. 1, p. 15, 2022.

**Tianjie Zhang** received the M.S degree in transportation engineering from Southeast University, China, in 2019. He is currently a Ph.D. candidate in the Department of Computer Science, Boise State University. His research mainly focuses on developing ML/AI algorithms in the transportation fields such as pavement damage inspection, physics-informed neural networks modelling in materials.

A person with long hair

Description automatically generated with low confidence**Donglei Wang** is currently pursuing a Master of Science in the Department of Civil Engineering at Boise State University. Her research interests mainly focus on independent networked community-level resilience assessment, natural hazard vulnerability analysis, and quantitative risk assessment for spatially distributed civil infrastructure networks.

A person wearing glasses and a suit

Description automatically generated with medium confidence**Yang Lu** Yang Lu was born in Nanjing, China, on March 9, 1979. He obtained his M.S. in Civil Engineering from Tsinghua University and Ph.D. in Transportation Infrastructure Engineering from Virginia Tech.

He is currently an Associate Professor of Civil Engineering at Boise State University. Prior to joining Boise State, he was an ARRA Fellow Research Associate at the National Institute of Standards and Technology (NIST), where he developed a micromechanics-based mechanistic approach to predicting the infrastructure materials performance. His major field of expertise is sustainable infrastructure and materials.

Dr. Lu’s research integrates multimodal characterization and multiscale modeling techniques to understand the properties and performance of novel transportation infrastructure materials under various service conditions. His representative work includes deep learning-enabled structural health monitoring, chemo-mechanical degradation accelerated by climate change, and virtual microstructure platform-enabled heterogeneous materials design. He is the recipient of the prestigious NIST outstanding associate award, CAES visiting faculty award, and US Office of Naval Research faculty research award. He has published 40+ peer reviewed journal papers and 20+ peer reviewed conference proceedings. Dr. Lu is affiliated with several professional organizations, including ASCE, ACI, ICE, and TRB, where he serves on 9 technical committees and as an active reviewer for 20+ journals.

1. (Corresponding author: Yang Lu).

   Tianjie Zhang is with the Department of Computer Science, Boise State University, Boise, ID, 83725 USA (e-mail: tjzhang@u.boisestate.edu).

   Donglei Wang is with the Department of Civil Engineering, Boise State University, Boise, ID, 83725 USA (e-mail: dongleiwang@u.boisestate.edu).

   Yang Lu is with the Department of Civil Engineering, Boise State University, Boise, ID, 83725 USA (e-mail: yangfranklu@boisestate.edu). [↑](#footnote-ref-1)