ECSNet: An Accelerated Real-time Image Segmentation CNN Architecture for Pavement Crack Detection

Tianjie Zhang, Donglei Wang, and Yang Lu*

***Abstract*—The ability to perform pixel-wise segmentation on pavement cracks in real-time is paramount in road service condition assessment and maintenance decision-making practices. Recent deep learning detection models are focused on detection accuracy and require a large number of computing sources and long run times. However, highly efficient and accelerated models with acceptable accuracy in real-time pavement crack detection tasks are required but hard to achieve. In this work, we present a customized deep learning model architecture named Efficient Crack Segmentation Neural Network (ECSNet) for accelerated real-time pavement crack detection and segmentation without compromising performance. We introduce some novel parts, including small kernel convolutional layers and parallel max pooling and convolutional operation, into the architecture for crack information quickly extraction and model's parameter reduction. We test latency and accuracy trade-offs of our proposed model using the DeepCrack Dataset. The results demonstrate strong performance in both accuracy and efficiency compared to other state-of-the-art models including DeepLabV3, FCN, LRASPP, Enet, Unet and DeepCrack. It is promising that ECSNet obtains the second place with an F_1 score of (84.45%) and an Intersection over Union (IoU) of 73.08%. Furthermore, our model gains the largest Frames Per Second (FPS) and lowest training time among all the models which is 73.29 and 5011 seconds, respectively. It maintains a good balance between accuracy and efficiency metrics.**

***Index Terms*— Deep learning, image segmentation, pavement crack detection, real-time task**

I. INTRODUCTION

Crack is a common defect of pavement that directly affects pavement service ability and driving safety [1]. Traditionally, the cracks are counted and measured by engineers, but it is time-consuming and labor-intensive [2]. With the development of image-based technologies, pavement crack detection approaches have achieved critical improvements in accuracy and reliability [3]. Crack classification and segmentation are two main focuses of the pavement crack detection. The purpose of classification is to differentiate different crack types while segmentation is a pixel-level extraction of crack from the background [4]. Segmenting cracks from pavement background is a more challenging task than classification and is more attractive to engineers and researchers. During the past decades, a series of crack segmentation methods have been proposed and deep learning-based techniques have become the most interesting and advanced approach to segmenting pavement cracks. This is because deep learning models learn from large-scale data and require little human involvement during training, which can dramatically increase their accuracy and robustness in segmentation tasks [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 higher accuracy. However, it will make the model become slower in training or predicting speed. This is because the increasing layers would dramatically increase the parameters in the model which would consume a lot of computing power during training and prediction. In pavement crack detection practices, 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. Moreover, a higher detection speed is very significant in improving transportation safety [6]. Due to the importance of the real-time inspection job for engineers and decision-makers, an ECSNet is proposed in this work to improve the computational efficiency of the pavement crack segmentation practices in real-time. While increasing the running speed, the proposed network can also keep a trade-off between accuracy and efficiency. An ECS-block which is constructed by a combination of 1×1 2D Convolutional layer (Conv2D), $n \times n$ Conv2D and 1×1 Conv2D is introduced for crack information extraction and model parameter reduction. A parallel max pooling and convolutional operation are also introduced in the network to speed up the downsizing and filtering process. The proposed ECSNet model is

(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).

tested on the DeepCrack dataset in which the image has a resolution of 544×384 pixels. In addition, the presented model is compared with other state-of-the-art models including DeepLabv3 [7], FCN [8], LRASPP [9], Enet [10], Unet [11] and DeepCrack [12]. The result shows that our model performs a good balance between accuracy and efficiency, which means it has the potential to be utilized into real-time pavement crack detection and measurement tasks.

The rest of the paper is organized in the following way: Section II introduces the research works of deep learning in crack segmentation. The details of the proposed ECSNet and the experiment design are described in Section III; Section IV demonstrates and analyzes the experimental results. Section V is the conclusion.

II. Literature Review

Three major kinds of crack detection methods have been presented in recent years. These include image-based methods like edge detection [13] and OTSU [14], machine learning-based methods including K-Means [15] and Support Vector Machine (SVM) [16] and deep learning-based methods which is the most advanced technology and has drawn much attention due to its superior performance in crack segmentation [17].

Several deep learning-based models have been developed in recent years to achieve higher accuracy in pavement crack segmentation. One important way is to introduce deeper layers and larger parameters into the architecture. For example, Sun et al. [18] adopted and enhanced a DeepLabv3 model with an attention module which can assign weights between the high-level and low-level feature maps, and it achieved a higher IoU on the DeepCrack, Crack500 and FMA datasets. Qu et al. [19] proposed a deep learning-based network with hierarchical feature fusion and connected attention architecture to enhance the feature extraction and eventually increased the accuracy of the proposed method (got an F_1 score of 0.86). Chen et al. [20] 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's performance is to increase the amount and diversity of the datasets. Images augmentation methods including traditional image augmentation [21] and Generative Adversarial Networks (GAN) [17, 22, 23] are popularly utilized before training to improve the model's performance. Some researchers merged the visual images with other kind of source data to improve the accuracy of their model. For example, Liu et al. [24] fused the visual image and thermal image together to increase the accuracy of classifying asphalt pavement crack severity. Liu et al. [25] combined the Ground Penetrating Radar B-Scan data and pavement images to improve the accuracy of crack detection.

It is true that either constructing complicated structures or combining multi-source data can improve the accuracy of the model. However, it decreases the model's training and predicting speed [26]. In an on-site pavement crack segmentation task, the model needs to be carried out in a timely fashion on a computationally limited platform. Moreover, the computing resource is not always sufficient enough to train a large-parameter model in the Department of Transportation. Therefore, a model that makes a good trade-off between accuracy and efficiency can be helpful in handling real-time tasks which are very significant to the decision-makers and engineers. On the one hand, a complicated model needs a very long training time and a high computational resource demand. On the other hand, a long-time prediction would hinder real-time work. Thus, there are rising interests and needs in developing small and efficient neural networks for the intelligent transportation system. However, there were only limited studies involving efficient pavement detection methods. For example, Pang et al. [27] 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 in which the image has a resolution of 448×448 pixels. Ronneberger et al. [11] presented an Unet model for biomedical image segmentation, which can use the annotated images more efficiently by using consisting of a contracting path to capture context and a symmetric expanding path to enable precise localization. Liu et al. [28] modified the Unet architecture to make it adopted in crack detection. The modified Unet was tested on a self-collected dataset and got an F_1 score of 0.9. However, the efficiency of the model was not evaluated. 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 results in an IoU of 73.79 and an FPS of 31.3. The Enet proposed by Adam [10] changed the previous encoder-decoder symmetrical structure, reduced the convolution operation in the decoder, and enhanced the processing speed tremendously. It introduced the 1×1 convolutional layer in the structure to reduce the dimensionality and parameters of the model without reducing the accuracy and achieved a higher mean IoU than SegNet [29].

From the above research, it is noteworthy that the convolutional neural network has been widely utilized in pavement crack segmentation tasks. However, the previous work is mainly focusing on introducing deep layers or augmenting the related databases which ignore the inference speed of their model. The efficiency of the model is also significant in on-site applications.

III . METHODOLOGY

A. ECSNet

ECSNet is proposed in this work to accelerate the real-time performance 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.

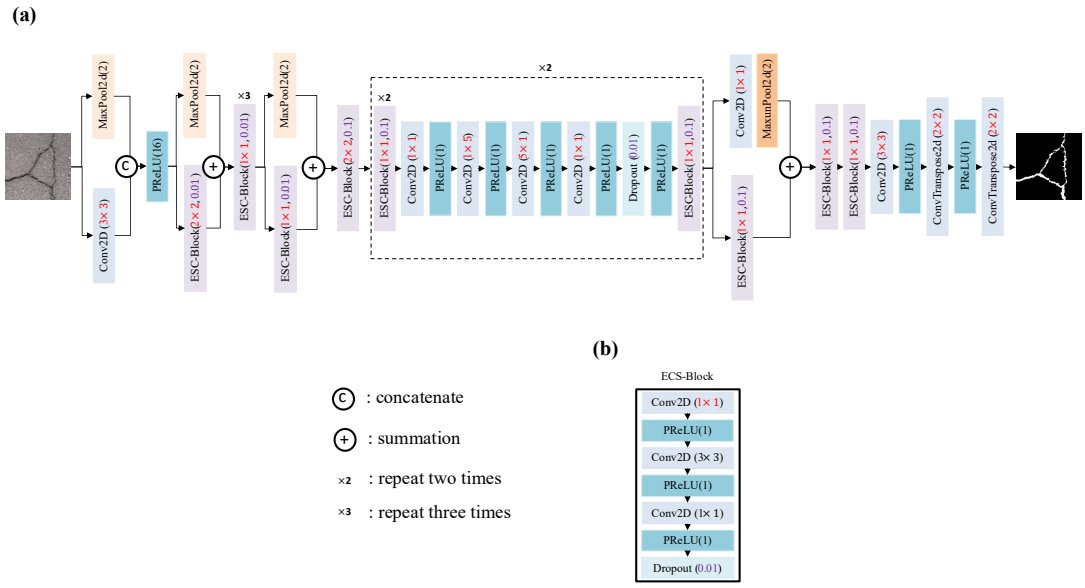

Figure 1. The architecture of the proposed ECSNet and its components: (a) The architecture of the proposed ECSNet; (b) The structure of an ECS-Block which is consisting of three convolutional layers: 1×1 Conv2D, 3×3 Conv2D and 1×1 Conv2D. The PReLU is used as an activation function in the ECS-Block.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

The proposed ECSNet contains 58 convolutional layers in total. Our acceleration strategy is to tailor the neural network architecture for pavement crack inspections by customizing the training procedures. A series of techniques 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 convolutional layer in ECSNet is followed by a Batch Normalization layer [30] to standardize the parameters. The operation in the Batch Normalization layer is shown in Equation (1).

$$y = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} \times \gamma + \beta, (1)$$

Where x is the input, y is the output, ϵ is a value added to the denominator for numerical stability ($\epsilon = e^{-5}$ in this case), γ and β are learnable parameters during the training. E[x] and Var[x] are the mean and variance of x, respectively.

By applying 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 1×1 Conv2D in the architecture to downsize the parameters first. We introduce the ECS-Block, a combination of 1×1 Conv2D, 3×3 Conv2D and 1×1 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).

$$P = (H \times W \times D + B) \times K, (2)$$

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), and K is the number of filters in this layer.

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

$$R_1 = \frac{1\times1\times D\times K+(3\times3\times K)\times K+(1\times1\times K)\times D}{(3\times3\times D)\times D}, (3)$$

Where R_1 stands for the ratio of parameters from ECS-Block and a 3×3 Conv2D. D is 64 and K is 16 in this case. Therefore, based on the Equation (3), R_1 equaling 8.5 means that the computational cost of a standard convolutional operation is 8.5 times higher than the proposed ECS-Block.

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 ECS-Block. The first parameter represents 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 $n \times n$ convolutional layer can be decomposed into two smaller ones: one with a $1 \times n$ kernel and one with a $n \times 1$ kernel. This idea has been worked on and proved in Adam's work [10]. The cost of the operation is lowered but the variety of operations increases 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 $\frac{n}{2}$ times lower than a standard $n \times n$ convolutional layer as shown in Equation (4)

$$R_2 = \frac{(n\times n\times D)\times D}{(1\times n\times D)\times D+(n\times 1\times D)\times D}, (4)$$

Where R_2 stands for the ratio of parameters contained in a $n \times n$ convolutional layer and a combination of $1 \times n$ Conv2D and $n \times 1$ 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 3×3 filter to downsize the input first, rather than using a 1×1 filter directly on the input image to sharply lower its size. After that, a convolutional layer with a 2×2 kernel size is used to downsize the pavement images again. A kernel size of 2×2 makes sure that we can take the full input into consideration from a 3×3 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 Parametric Rectified Linear Unit (PReLU) [31] is used as the nonlinear layer in the network. The equation of PReLU is shown below.

$$PReLU(x) = \begin{cases} x, x \geq 0 \\ Ax, x < 0 \end{cases}, (5)$$

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.

The reason to use PReLU rather than ReLU is that the crack in pavement is the relative dark part in the background. The PReLU can give a learnable weight to the small values while they are totally ignored in the ReLU function, as shown in Figure 1(c). Moreover, PReLU can improve the proposed model fitting by increasing very little computational cost and overfitting risk. 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.

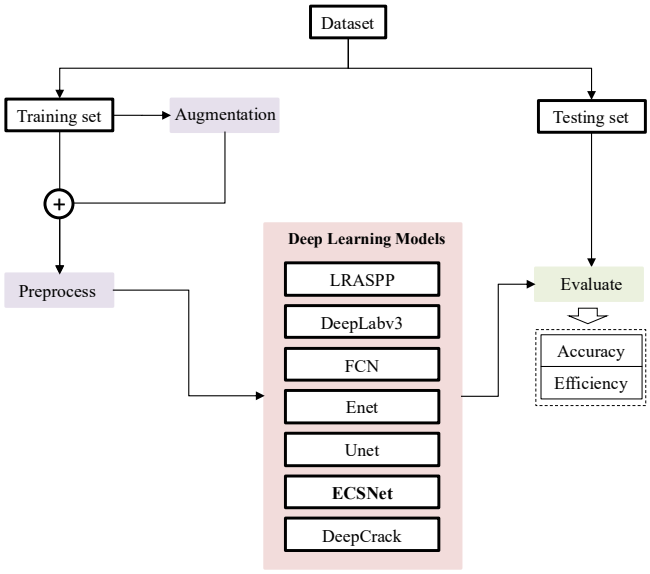

Figure 2. The overall procedure to evaluate the performance of the deep learning neural networks. Models including

LRASPP, DeepLabv3, FCN, Enet, Unet, ECSNet and DeepCrack are tested and compared in accuracy and efficiency.

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 on 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 highly related to the amount and diversity of datasets it used. The training and testing set will be processed by changing the color to gray before inputting the images into the model. The input channels of all the models are set to one, so that the gray images can be used as input to train and test the model. In order to reduce the computing source, all the images and their ground truths are resized to 256×256 pixels automatically during the training procedure.

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

(1) DeepLabv3: We adopt the ResNet-50 to the original DeepLabV3 structure by applying convolution to extract dense features.

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

(3) FCN: We adopt an FCN network with a Resnet50 [32] backbone.

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

(5) Unet: We include the popular Unet architecture, which is widely recognized for its effectiveness in semantic segmentation tasks. Unet consists of an encoder-decoder structure with skip connections that allow for the fusion of low-level and high-level features, facilitating the precise localization of cracks in pavement images. The encoder part captures contextual information, while the decoder part reconstructs the segmented output.

(6) DeepCrack: This model was developed by 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) [33] with a momentum of 0.9 as an optimizer to update the network. A learning rate of 1e-4 is used during the training, and a batch size is set to 16 according to other research work and the limitation of the GPUs. BCEWithLogitsLoss [34] 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 [4]. Totally 300 epochs are trained on each network. This is because the loss of all the models keeps stable after 300 epochs running. 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. The mean value and standard deviation would be calculated to represent the average performance and robustness of the model. By doing this, the results would be more 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, and 64 GB RAM.

C. 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 accuracy and efficiency metrics.

For the accuracy part, F_1 and IoU are utilized to evaluate the semantic segmentation results.

F_1 is defined based on the harmonic average of Precision and Recall as shown in Equation (6). The precision and recall can be calculated using Equations (7) and (8).

$$F_1 = \frac{2PR}{P+R}, \quad (6)$$

$$P = \frac{TP}{TP+FP}, \quad (7)$$

$$R = \frac{TP}{TP+FN}, \quad (8)$$

Where TP denotes True Positive, FP is False Positive, FN is False Negative, P is precision, R is recall and F_1 is F_1 score. F_1 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 overlap with the ground truth. IoU is calculated according to Equation (9).

$$IoU = \sum_{i,j}^{k} \frac{p_{ii}}{p_{ij}+p_{ji}-p_{ii}}, \quad (9)$$

Where p_{ij} represents the number of pixels belonging to class i but predicted as class j. The IoU represents how many 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 needs in training. The Frames Per Second (FPS) in the testing procedure is used to evaluate the efficiency of the models in predicting. The FPS can be an index for real-time jobs. A higher FPS indicates the model can process more images in a unit of 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.

IV. Results

A. Accuracy evaluation

Figure 3. shows the F_1 score and IoU values from the models tested in this work, including DeepLabv3, LRASPP, FCN, Enet, Unet, DeepCrack and ECSNet.

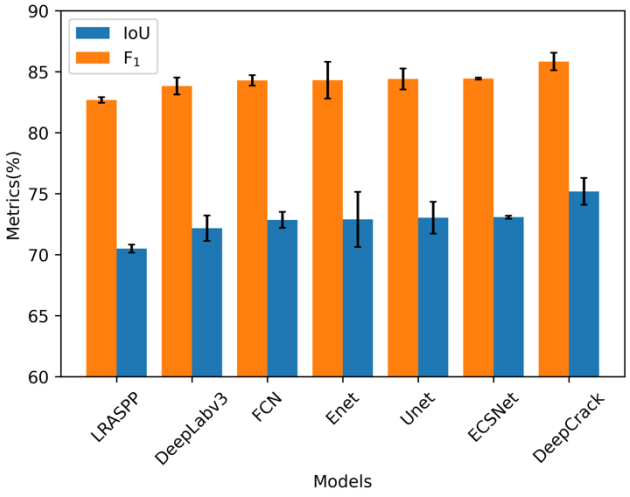

Figure 3. The F_1 score and IoU of all the tested models. The models are organized in ascending order based on the IoU values achieved on the DeepCrack dataset. The blue bar stands for the IoU and the orange bar stands for the F_1 score of each model. The black error bar is the standard deviation.

It shows that the DeepCrack gets the highest F_1 score (85.84%) and IoU values (75.19%) among all the models. It is noteworthy that the ECSNet gains the second place in both F_1 score (84.45%) and IoU value (73.08%). The error bar of the testing results shows the standard deviation of each model. It is interesting that ECSNet gets the lowest standard deviation among all the approaches. In other words, the performance of ECSNet is more stable and robust than other methods.

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

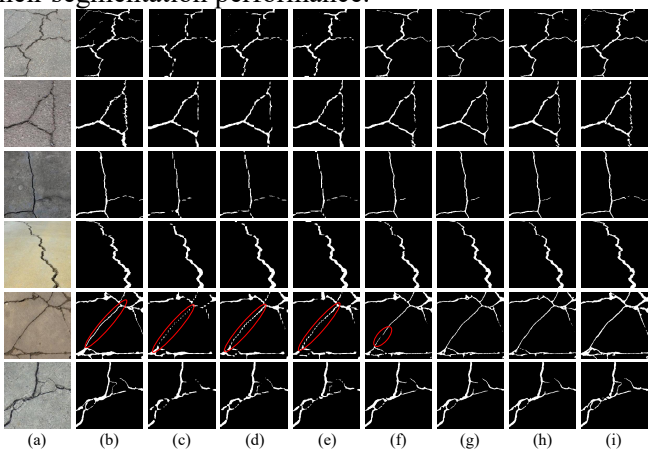

Figure 4. The visualization of detection results of different deep learning methods on the DeepCrack Dataset: (a) original image; (b) ground truth; (c) LRASPP; (d) DeepLabv3; (e) FCN; (f) Enet; (g) Unet; (h) ECSNet; (i) DeepCrack. The red circle shows models including FCN, DeepLabv3 and LRASPP are intermittent in prediction compared to the ground truth.

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, ECSNet and Unet are quite consistent with the ground truth compared to other models. It is noteworthy that the segmentation results from DeepCrack are the closest to the ground truth.

B. Efficiency evaluation

The model parameters, average training time, memory usage and average inference time are important indexes to indicate the efficiency of the model. Table 1 shows the parameters of each model.

TABLE 1. The complexity of different methods, including model parameters, average training time, memory footprint and average inference time.

Models	Params (M)	Training Time (S)	Memory Footprint (MB)	Inference Time (MS)
LRASPP	3.22	3036.8	24.74	16.52
DeepLabv3	39.63	16244.4	307.62	22.64
FCN	32.95	12291.8	256.01	20.37
Enet	0.36	6494.3	3.047	17.85
Unet	13.39	14560.6	103.45	15.61
ECSNet	0.41	5011.2	2.52	13.66

DeepCrack	30.90	108967.7	244.57	33.93

Figure 5 shows the comparison of training time, parameters and FPS among different models. The black bar is the calculated standard deviation from three experiments.

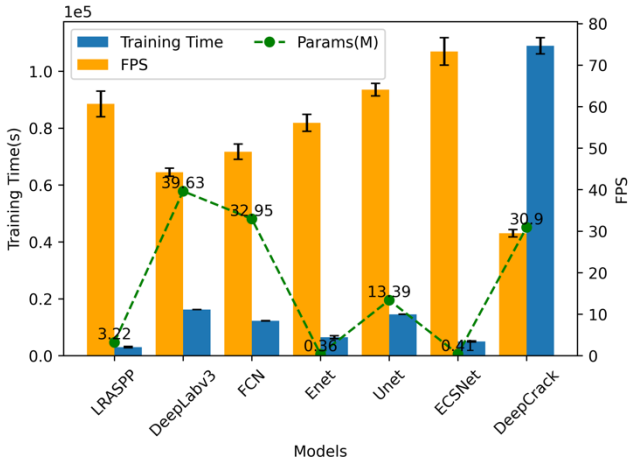

Figure 5. The efficiency-related performance of each model. The blue bar stands for the training time used by each model. The yellow bar is the FPS of each model performed on the DeepCrack dataset. The green points are the parameters contained in the models. The black error bar is the standard deviation.

It is noteworthy that the training time each model spend is quite consistent with the FPS. In other words, a model's structure 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 speed in pavement crack segmentation. It is about 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 Unet, 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 of 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, training time and FPS as shown in Figure 6-8. Each model is trained and tested three times on the dataset to make the results meaningful in statistics. Therefore, there are three points for each model. Figure 6 shows the relationship between the IoU and parameters in the model.

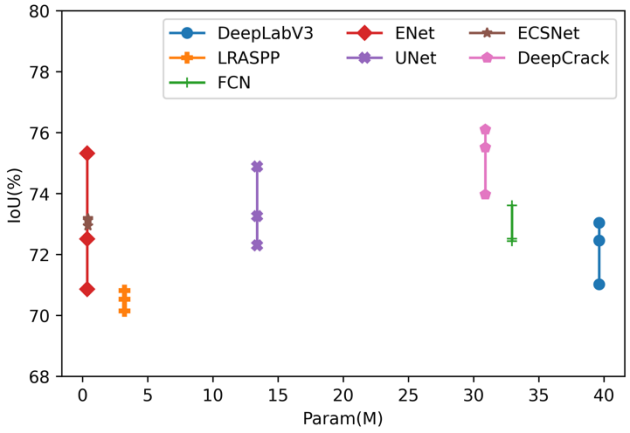

Figure 6. The relationship between IoU and the number of parameters in each model. Each point means a different test and there are three tests for each model.

The parameters contained in a model are an index to show its proper to be deployed in a mobile device. In other words, a lightweight model is more reasonable for mobile usage. Figure 6 shows that the Enet and ECSNet have nearly the same parameter amount (0.36 M 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 standard deviation 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.

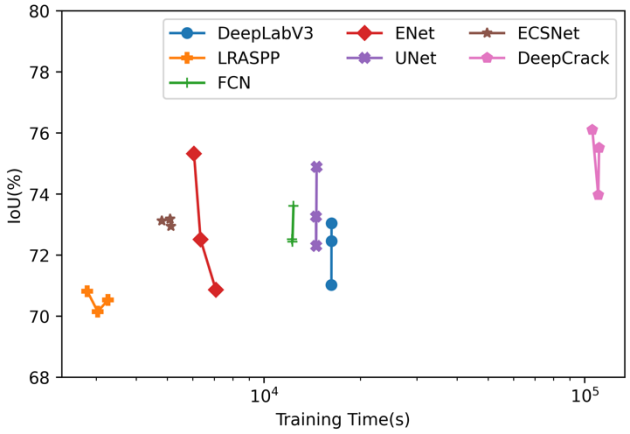

Figure 7. The relationship between IoU and Training Time of

each model. Each point means a different test and there are three tests for 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 second place in the average IoU value. Compared with other segmentation algorithms, our proposal approach achieves a trade-off between the amount of training time and accuracy. The LRASPP uses the lowest time (about 2800 seconds) in the 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.

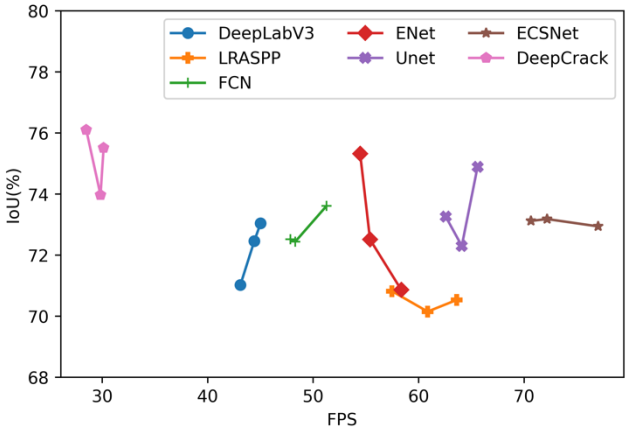

Figure 8. The relationship between IoU and FPS in each model. Each point means a different test and there are three tests for each model.

As we can see from Figure 8, the proposed ECSNet gets an absolutely higher FPS (73.29) than any 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. It only losses 2.8% of IoU compared to the model DeepCrack. However, the FPS of ECSNet is 2.5 times higher than DeepCrack (29.5). In other words, our designed network makes a trade-off between the model's accuracy metrics and real-time requirements.

V. Conclusion

In this work, an ECSNet is proposed for real-time pavement crack detection tasks. This model is designed as a lightweight real-time semantic segmentation model with high computational efficiency. An ECS-block consisting of 1×1 Conv2D, $n \times n$ Conv2D and 1×1 Conv2D is introduced to network as an alternative of a normal $n \times n$ Conv2D to reduce the model parameter and improve the computing speed. It shows the parameters contained in an ECS-block are 8.5 times lower than a normal $n \times n$ Conv2D. A parallel max pooling and convolutional operation are also introduced in the network to replace the sequence of convolution layer and max pooling layer, which speed up the downsizing and filtering process.

The proposed ECSNet is compared with other popular models, including DeepLabv3, LRASPP, FCN, Enet, Unet, and DeepCrack. According to the overall performance on the DeepCrack Dataset, the ECSNet obtains an F_1 score of 84.45% and an IoU of 73.08%, which outperforms all the popular models except the model DeepCrack. The segmentation result from ECSNet is clearer and more continuous than DeepLabv3, LRASPP and FCN. It is also demonstrated that the ECSNet has the highest FPS (73.29) and lowest training time (5011 seconds) among all trained models. Comparing the IoU with efficiency metrics, it is worth noticing that the proposed ECSNet keeps a good balance between the accuracy and efficiency metrics. It only losses 2.8% of IoU to gain a 2.5 times higher FPS compared to DeepCrack model, which shows that ECSNet is a well-designed real-time image segmentation algorithm for the pavement crack detection task.

Although ECSNet showcases promising efficiency in crack segmentation, there are a few notable limitations that require attention. Firstly, further model optimization is needed to reduce computational requirements and enhance overall efficiency, especially when dealing with larger datasets or real-time deployment scenarios. Secondly, the presence of data biases can impact the model's performance, necessitating the need for careful consideration of dataset quality and diversity during training. Addressing these limitations can be achieved by incorporating a multi-scale module to capture cracks of varying sizes and exploring transfer learning techniques to improve the generalization and robustness of ECSNet. These future directions have the potential to enhance the model's efficiency, mitigate data biases, and enable wider applicability of ECSNet in practical crack segmentation tasks.

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, X. Yang, S. Lau, X. Wang, S. Luo, V. C. S. Lee, and L. Ding, "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, pp. 04022023, 2022.

[4] 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.

[5] 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.

[6] H. Zhang, Y. Song, Y. Chen, H. Zhong, L. Liu, Y. Wang, T. Akilan, and Q. J. Wu, "MRSDI-CNN: Multi-model rail surface defect inspection system based on convolutional neural networks," *IEEE Transactions on Intelligent Transportation Systems,* vol. 23, no. 8, pp. 11162-11177, 2021.

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

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation." pp. 3431-3440.

[9] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, and V. Vasudevan, "Searching for mobilenetv3." pp. 1314-1324.

[10] 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.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation." pp. 234-241.

[12] 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.

[13] H. Han, H. Deng, Q. Dong, X. Gu, T. Zhang, and Y. Wang, "An advanced Otsu method integrated with edge detection and decision tree for crack detection in highway transportation infrastructure," *Advances in Materials Science and Engineering,* vol. 2021, 2021.

[14] Y. Quan, J. Sun, Y. Zhang, and H. Zhang, "The method of the road surface crack detection by the improved Otsu threshold." pp. 1615-1620.

[15] C. Fang, L. Zhe, and Y. Li, "Images crack detection technology based on improved K-means algorithm," *journal of multimedia,* vol. 9, no. 6, pp. 822, 2014.

[16] Y. Sari, P. B. Prakoso, and A. R. Baskara, "Road crack detection using support vector machine (SVM) and OTSU algorithm." pp. 349-354.

[17] 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, pp. 119397, 2020.

[18] 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.

[19] 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.

[20] T. Chen, Z. Cai, X. Zhao, C. Chen, X. Liang, T. Zou, and P. Wang, "Pavement crack detection and recognition using the architecture of segNet," *Journal of Industrial Information Integration,* vol. 18, pp. 100144, 2020.

[21] Z. Xu, H. Guan, J. Kang, X. Lei, L. Ma, Y. Yu, Y. Chen, and J. Li, "Pavement crack detection from CCD images with a locally enhanced transformer network," *International Journal of Applied Earth Observation and Geoinformation,* vol. 110, pp. 102825, 2022.

[22] Y. Hou, S. Liu, D. Cao, B. Peng, Z. Liu, W. Sun, and N. Chen, "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.

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

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

[25] 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, pp. 104698, 2023.

[26] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[27] 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.
[28] Z. Liu, Y. Cao, Y. Wang, and W. Wang, "Computer vision-based concrete crack detection using U-net fully convolutional networks," *Automation in Construction,* vol. 104, pp. 129-139, 2019.
[29] 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.
[30] S. Ioffe, and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift." pp. 448-456.
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." pp. 1026-1034.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition." pp. 770-778.
[33] T. Kurbiel, and S. Khaleghian, "Training of deep neural networks based on distance measures using RMSProp," *arXiv preprint arXiv:1708.01911*, 2017.
[34] 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, pp. 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.

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.

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.