DOI: 10.21917/ijivp.2026.0540

EFFICIENT PAVEMENT CRACK DETECTION FOR REAL-TIME ROAD MAINTENANCE USING DEEP LEARNING MODELS

G. Prema, S. Arivazhagan, R. Shriram and R. Sri Venkadesh
Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, India

Abstract

Pavement crack detection needs to be done to identify and assess cracks in road surfaces. Detecting the crack and its measurement by manual methods is extremely time-consuming and requires a lot of manpower. This process is crucial for maintaining road safety and infrastructure integrity. By detecting cracks early, authorities can prioritize repairs and prevent further damage, ultimately extending the lifespan of roads and reducing maintenance costs. Additionally, crack detection helps improve driving conditions and safety for motorists by enabling timely repairs to be made. Overall, pavement crack detection plays a vital role in ensuring the durability, safety, and efficiency of road networks. Some factors, such as non-uniform intensity, complexity, and irregular patterns of cracks, complicate the process, and the accuracy of the results may be affected. The aim of this study is to develop a practical crack segmentation method for real-time maintenance. In this paper, two models are proposed based on U-Net architecture and feature pyramidal network (FPN) architecture. To verify the superiority and generalizability of the proposed method, two publicly available CRACK500 and CFD datasets are used. Metrics such as AIU (Average Intersection over Union) and ODS (Overall Dice Similarity) measure are used to evaluate the performance. These metrics indicate that the proposed method effectively segments cracks in pavement images, demonstrating its potential for use in real-world applications.

Keywords:
Deep Learning, U-Net, FPN, Crack Segmentation

1. INTRODUCTION

Cracks that occur in the pavement are one of the common threats to the road and highway safety. Crack detection is the essential measure to be taken for the maintenance of the pavement without any damage. Automatic crack detection is preferred over manual crack detection as it does not require any manpower or expertise. After conducting a lot of surveys on various crack detection algorithms, it was concluded that deep learning is the most widely used approach because of its excellent, robust feature representation capability. In an earlier period, Liu et al. [11] and Kaseko et al. [10] proposed a methodology based on thresholding approaches based on the assumption that crack regions are thicker than other regions. After a thorough analysis of the different crack detection algorithms, we understand that deep learning has been extensively used in recent years due to its strong and robust feature representation capabilities. It incorporates techniques like skip connections, dilated convolutions, and encoder-decoder architecture to improve segmentation accuracy by handling various object shapes and sizes. The pixel-wise classification methodology is efficient these days. With the inspiration of deep learning techniques, Long et al. [9] proposed an approach by extending traditional CNN for classifications to produce dense pixel-wise predictions for semantic segmentation tasks, allowing for end-to-end learning and inference on images of arbitrary sizes. This work demonstrates the effectiveness of a fully connected network for semantic segmentation. In [8], Liu et al. proposed a methodology to improve edge detection mechanisms through the combination of a deep convolutional network with a dense network and a backward refinement path module. The first part comprises a dense network, which focuses on forward path feature extraction. The second part of the backward refinement path module, which fuses feature maps with intermediate feature maps along the forward propagation path. This aims to integrate low-level detailed features with high-level abstract features. It achieves a 0.5 increase in the performance of average intersection over union (AIU) over the previously proposed methodology. In [15], Qingsong Song et al. focused on crack segmentation using two stage framework with improved U-Net based on self-supervised contrastive learning. This will improve accuracy by the way of integrating attention mechanisms and residual structure into the architecture. This method effectively utilizes self-supervised contrastive learning to reduce the reliance on large volumes of labeled data making it more efficient compared to traditional supervised learning methods. Despite using fewer labeled samples for fine-tuning and pre-training compared to other methods, the proposed framework achieves comparable segmentation performance, showcasing its efficiency and effectiveness. In [12], Fan et al. presented a novel approach for automatic crack detection on road pavements using a modified U-net architecture called U-HDN. The U-HDN method incorporates a multi-dilation module and a hierarchical feature learning module to enhance crack detection accuracy. This method can extract and fuse different context sizes and levels of feature maps, enhancing its ability to detect cracks in various conditions. This study outlines future directions for optimizing computational efficiency, exploring video streaming detection, addressing different types of cracks, and integrating various segmentation algorithms for a comprehensive crack detection system.

Yang et al. [4] use holistically nested edge detection (HED), an edge detection method, as the backbone architecture to overcome the problems in the thresholding approach, such as failures in extraction in the presence of airtacts and grayscale fluctuations. Moreover, he introduces a feature pyramid module for enriching context information in lower-level layers with the help of higher-level layer feature maps by using top-down architecture. But it works slightly differently in low-light conditions. By way of overcoming the disadvantages of airtacts in [4] and [1] they introduced the methodology of feature pyramidal networks with hierarchical boosting networks at the end of the structure. This innovative network integrates context information into low-level features in a feature pyramidal way. Finally, it balances the contribution of both easy and hard samples through sample reweighting in a hierarchical manner. It somehow achieves better performance in low-illumination conditions. In DAUNET, they overcome the failures in low illumination and shadow conditions by introducing the fusing features at different scales, embodying dilated convolution modules, and also by selecting the appropriate loss function. This work emphasises the use of U-net architecture and achieves a dramatic change in metric for the GAPs 384 dataset.

This paper proposes encoder-decoder architecture for crack segmentation, with key contributions summarized as follows:

- Utilization of a U-net based model and a feature pyramidal model, showcasing improved performance over existing crack detection methods.
- Implementation of simple augmentation techniques and the application of various loss functions to enhance performance, particularly in low-light conditions.
- Demonstrating improved results without increasing the epoch count, instead utilizing Otsu thresholding for better outcomes.
- Evaluation of results using standard measures by comparing predicted and ground truth images.

The paper is organized as follows: Section 2 outlines the datasets used and their augmentation process. Section 3 details the proposed methodology, presenting two different architectures. Section 4 discusses the optimizers and loss functions employed. Section 5 showcases the experimental results and compares them with state-of-the-art detection methods.

2. DATASETS USED

2.1 CRACK500

The CRACK500 dataset [1], is a widely used benchmark dataset captured by smartphones at Temple university in the field of pavement crack detection. It consists of 500 images of size 2550x1440, with 250 original images and 250 corresponding ground truth images. These images are further resized to 256x256. Six different augmentations, including horizontal and vertical flips, contrast level changes, saturation level adjustments, brightness level modifications, and hue alterations, are applied which is depicted in Table.1. The total number of images in the CRACK500 dataset, excluding the ground truth images, is 1200 for training and 50 for testing. The Fig.1 illustrates a few samples from the dataset.

2.2 CFD DATASET

The Crack Forest dataset (CFD), sourced from Kaggle and introduced by Shi et al. [5], consists of 118 images sized at 320x480 pixels. After augmentation 600 images are used for training and 18 images for testing.

Table.1. Augmentation results

Datasets	Total images (Excluding Ground truth)	Training Images	Testing Images
CRACK500	250	(5*200)+200=1200	50
CFD	118	(5*100)+100=600	18

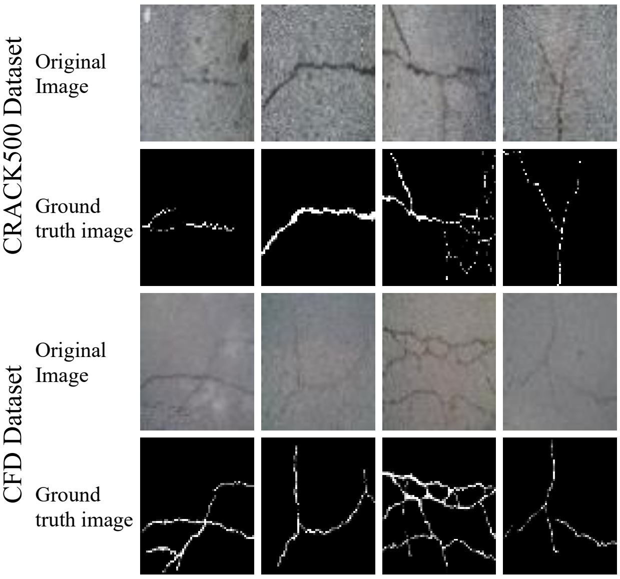


Fig.1. Original and Corresponding Ground Truth Images from CRACK500 and CFD Datasets

3. PROPOSED METHODOLOGY

Our methodology employs two key architectures: U-Net and Feature Pyramidal Network (FPN).

3.1 U-NET

U-Net architecture includes two parts: Encoder and Decoder The encoder in our model preprocesses input images to a standard size of 256x256 and consists of five stages, each containing convolutional layers followed by max pooling. These layers extract features and reduce spatial dimensions. The decoder, consisting of transposed convolutions and upsampling, reconstructs the original image size, aiding in fine-grained feature extraction. We utilize pre-trained ImageNet weights for transfer learning in the encoder, specifically using ResNet-18 or VGG-19, with modifications to remove classification layers. ResNet-18 is favored for its faster execution. ReLU activation is applied at each stage. A sample representation of the encoder is shown in Fig.2. The decoder in our model follows a standard U-Net architecture, with skip connections linking encoder and decoder to preserve global context and fine-grained features. The schematic representation of Encoder-decoder U-Net architecture is depicted in Fig.3. Training and Non-Trainable parameters are detailed in Table.2.

Table.2. Trainable and Non-Trainable parameters

CNN	Total Parameters	Non-Trainable Parameters	Trainable Parameters
ResNet18_U-Net	14340570	14330644	9926
VGG19_U-Net	29061969	29057937	4032
ResNet18_FPN	13815370	13805124	10246
VGG19_FPN	22882241	22879937	2304

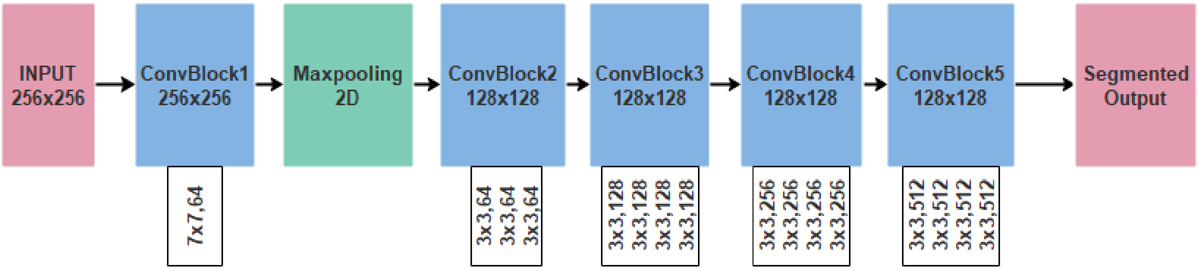


Fig.2. Resnet18 Encoder representation

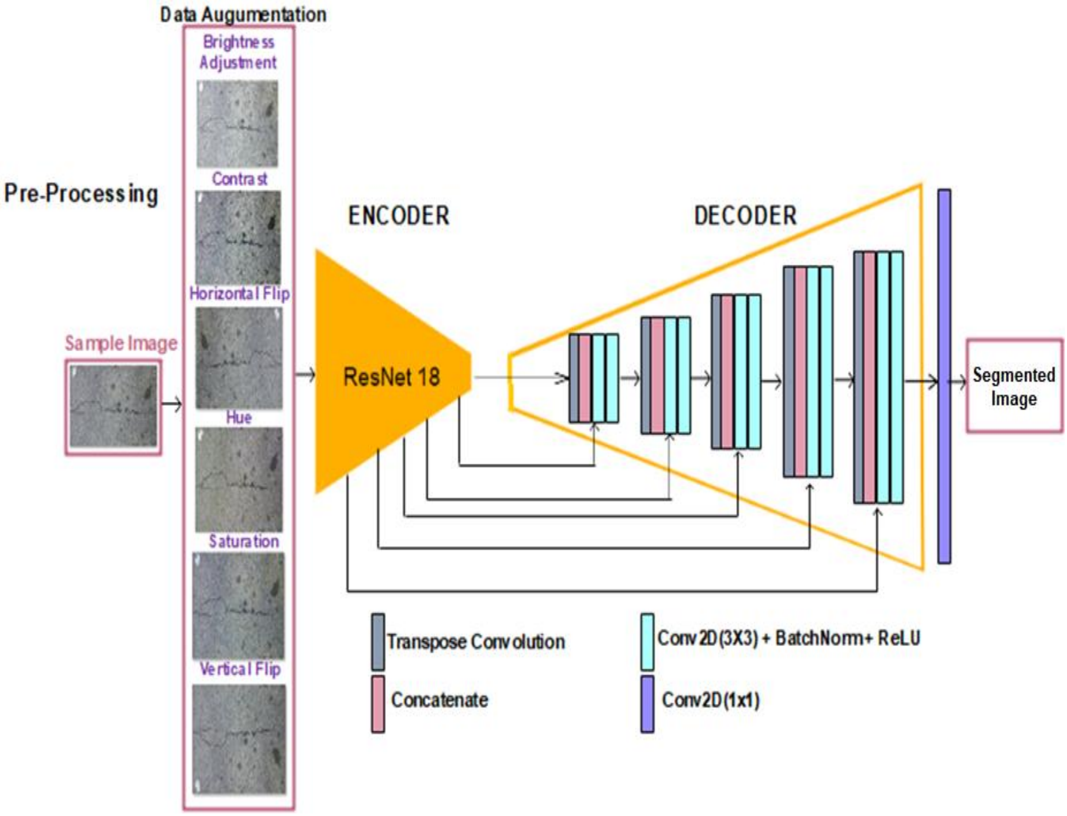


Fig.3. Encoder-decoder Representation of U-Net Architecture

3.2 FPN

A Feature Pyramidal Network (FPN) serves as a feature extractor for deep convolutional neural networks, commonly used in tasks like semantic segmentation and object detection. While U-Net and FPN share similarities, their decoding stages differ. U-Net concatenates to preserve global context and fine-grained features, whereas FPN uses element-wise addition to merge high-resolution and semantic information, creating a pyramid structure. FPN merges semantically strong features from deeper layers with high-resolution but less semantically rich features from shallower layers, creating a feature pyramid crucial for object detection. The encoder part of FPN mirrors that of U-Net.

In our implementation, we explored ResNet18 and VGG19, the latter having three additional convolutional layers compared to VGG16, which was used in prior work. VGG19 Encoder implementation is depicted in Fig.4. Following the encoder stage, the 1x1 convolution in Fig.5 normalizes the output channels from all stages. The subsequent up-and-down sampling process proceeds in a top-down manner, with each stage initially upsampled to match the resolution of the next stage. This process repeats for all stages, culminating in the final stage's upsampled output passing through the final convolution block to produce.

3.3 OTSU THRESHOLDING

Otsu thresholding is an image processing technique that employs automated threshold selection. It finds the optimal threshold value to distinguish between foreground and background pixels by optimising between-class variance while decreasing within-class variance.

By iteratively evaluating several threshold values, Otsu's method successfully determines the threshold that best distinguishes between object and background pixels. It is commonly employed in image segmentation, edge detection, and object recognition to enhance image contrast and extract features.

4. OPTIMIZER AND LOSS FUNCTION

In our models, we use the Adam Optimizer for training, which adjusts the learning rate based on the average of previous gradients and squared gradients. It utilizes separate learning rates for individual parameters, allowing it to respond effectively to each parameter's unique characteristics. The optimizer also incorporates momentum to accelerate gradient descent and dampen oscillations. Adam corrects biases in estimates of first and second moment gradients, leading to more stable convergence behavior. The algorithm involves initializing the first and second moment vectors to zero, setting the time step notation to zero, and initializing hyper parameters such as the learning rate alpha, decay rates for moments α_1 and α_2, and a small constant ε to prevent division by zero.

The Adam optimizer updates parameters using the following steps:

1. Update the first moment vector:

$$a_t = \alpha_1 a_{t-1} + (1-\alpha_1)x_t$$

2. Update the second moment vector:

$$b_t = \alpha_2 b_{t-1} + (1-\alpha_2)(x_t^2)$$

3. Bias correction for the first moment: $\hat{a}_t = \dfrac{a_t}{1-\alpha_1^t}$

4. Bias correction for the second moment: $\hat{b}_t = \dfrac{b_t}{1-\alpha_2^t}$

5. Compute the parameter update: $\Delta\theta_t = -\dfrac{\beta}{\sqrt{\hat{b}_t} + \varepsilon}\hat{a}_t$

6. Update the model parameters: $\theta_{t+1} = \theta_t + \Delta\theta_t$

Adam offers several advantages over SGD, including faster convergence, robustness to noisy gradients, and adaptability to various optimization landscapes. We use the Binary Focal Loss function, whereas other works may use binary categorical loss or sigmoid cross-entropy loss. The Binary Focal Loss is defined in Eq.(1)

$$FL(G,P) = -G\beta(1-P)^{\delta}\log(P) - (1-G)\beta P^{\delta}\log(1-P) \quad (1)$$

where, P is the predicted image, G is the ground truth image, β is set to 0.25, and δ is set to 0.2.

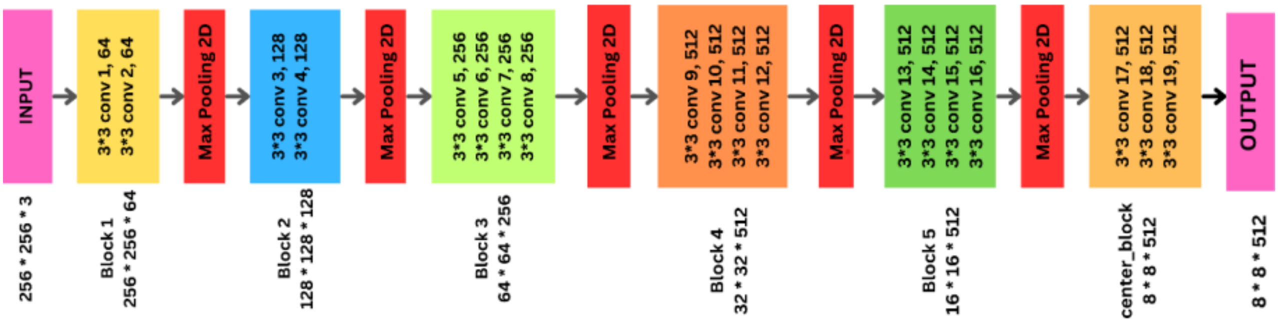


Fig.4. VGG19 Encoder implementation

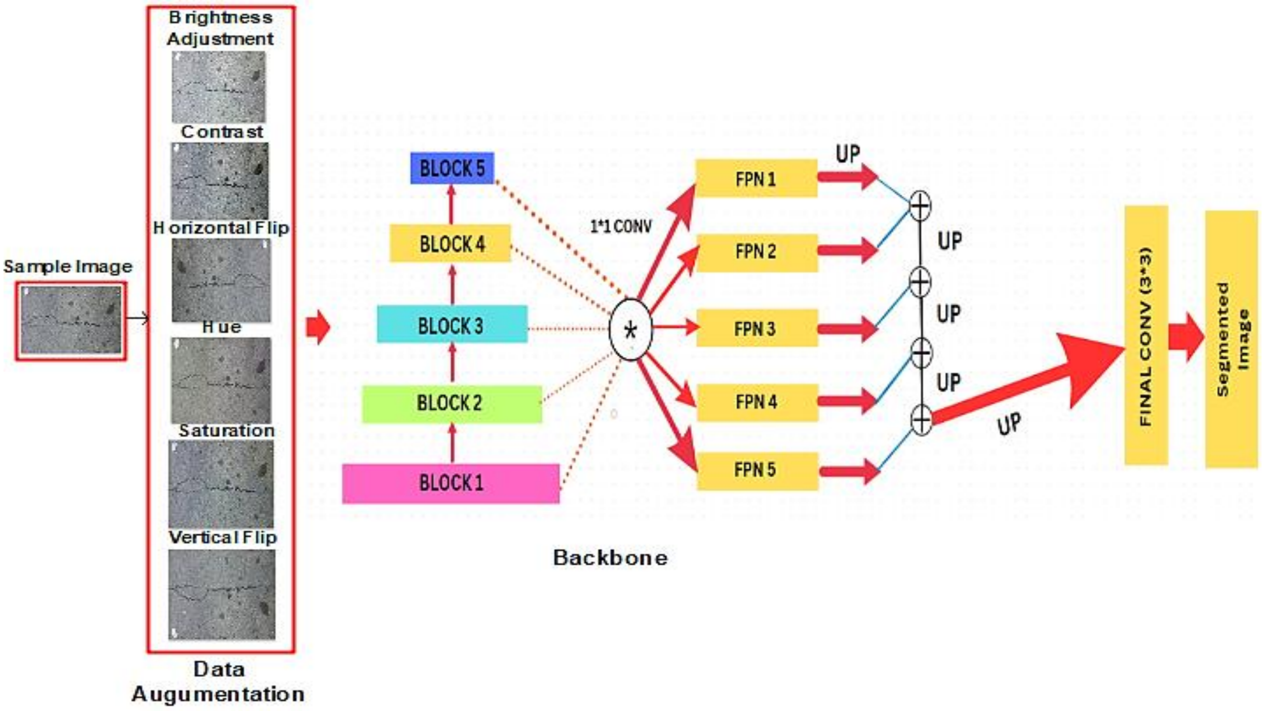


Fig.5. VGG19-FPN Block Diagram

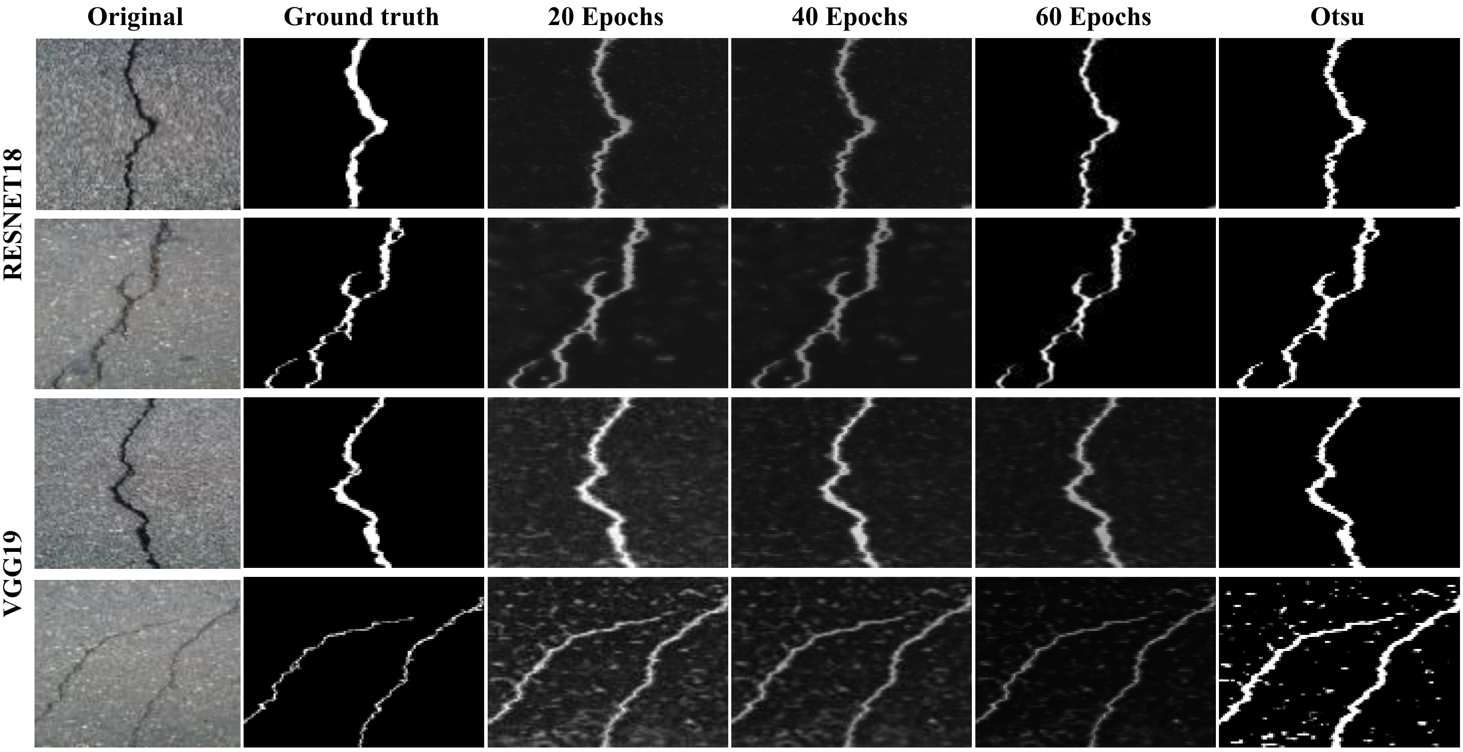


Fig.6. U-Net architecture Experimental Results on CRACK500 Dataset

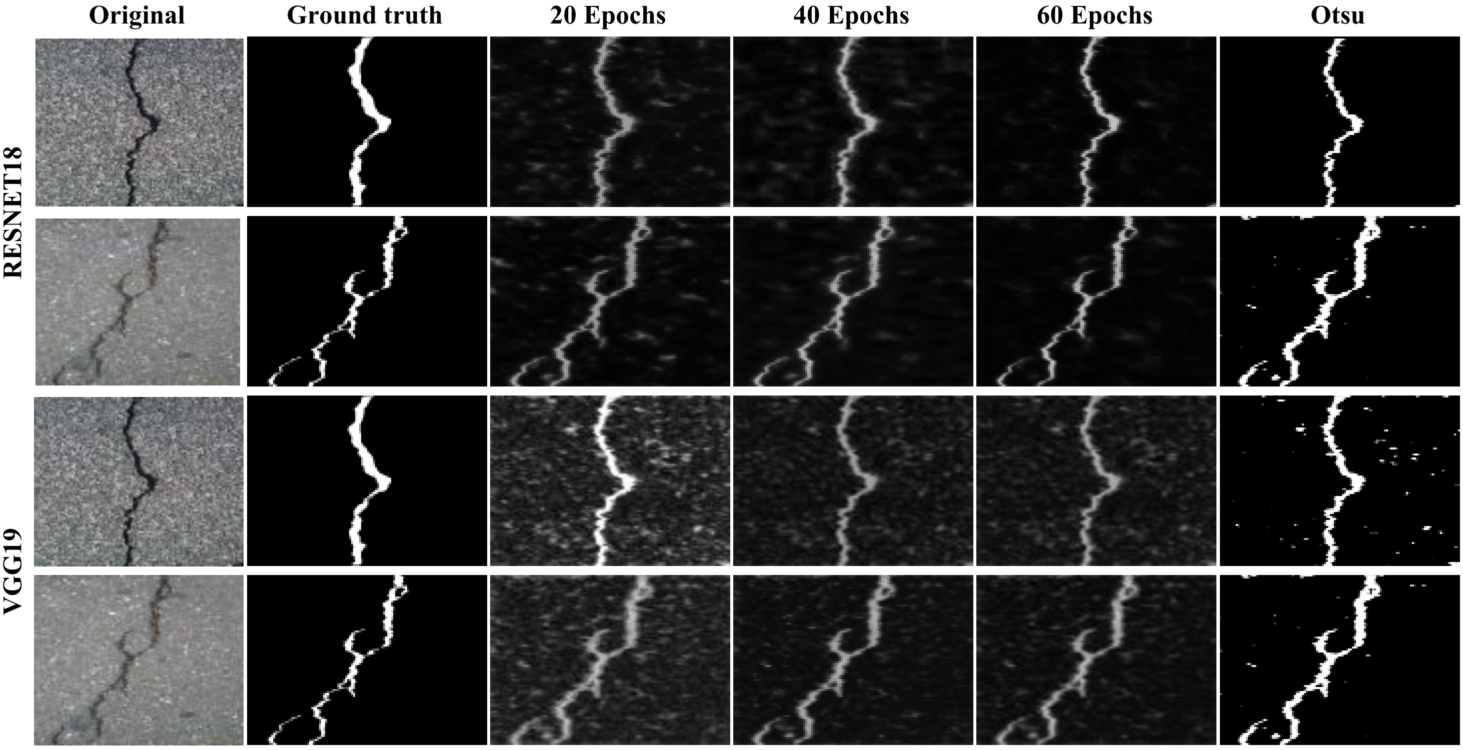


Fig.7. FPN architecture Experimental Results on CRACK500 Dataset

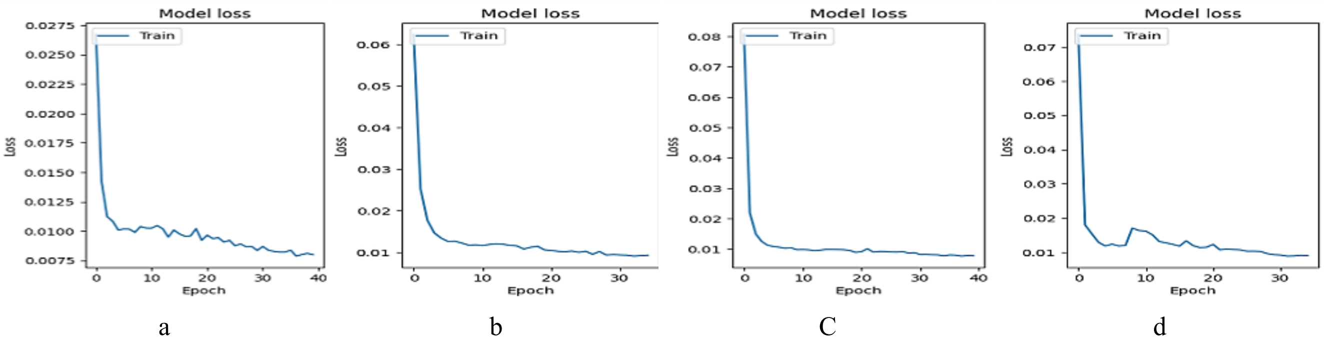


Fig.8. Loss (a) VGG19-UNet (b) VGG19-FPN (c) ResNet18-UNet (d) ResNet18-FPN

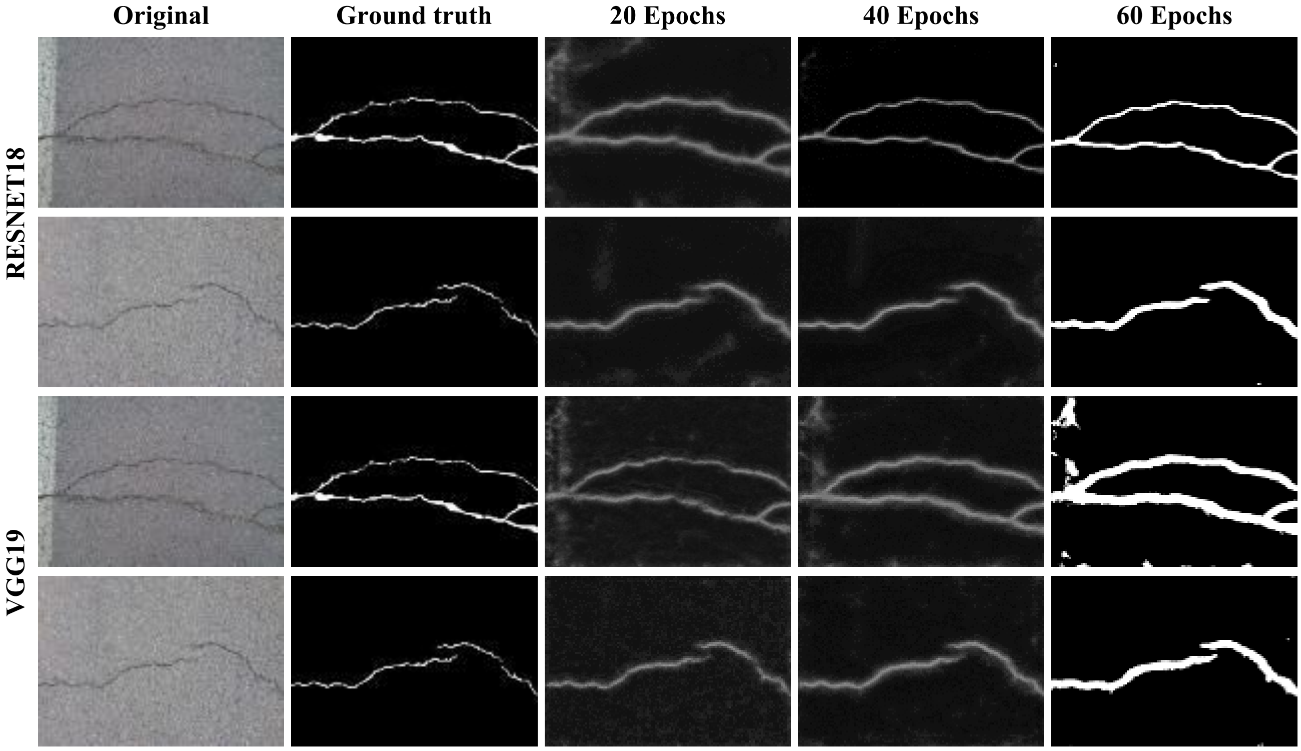
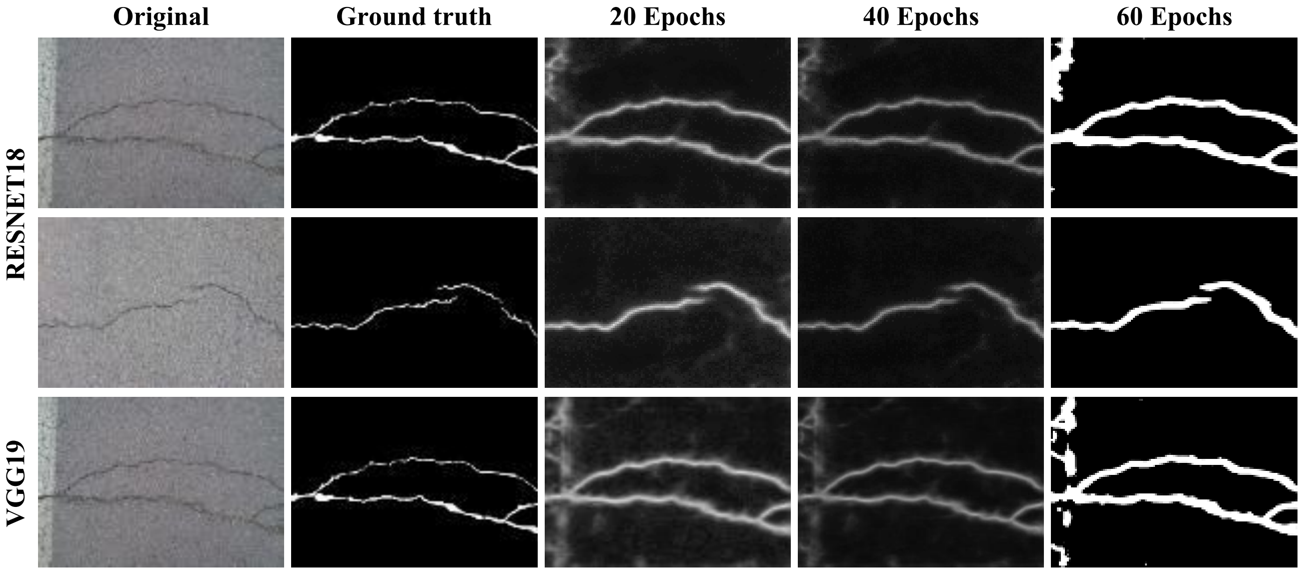


Fig.9. FPN architecture Experimental Results on CFD Dataset



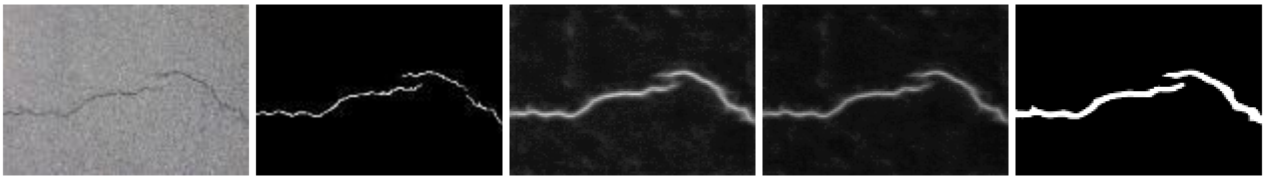


Fig.10. FPN architecture Experimental Results on CFD Dataset

Table.3. Metrics analysis of U-Net Model on CRACK500 Dataset

Metrics	Dice Coefficient (ODS)				Pixel Accuracy				AIU Score			
Epochs Count	**20**	**40**	**60**	**Otsu**	**20**	**40**	**60**	**Otsu**	**20**	**40**	**60**	**Otsu**
ResNet18	0.345	0.494	0.607	0.702	0.481	0.612	0.793	0.825	0.381	0.457	0.532	0.584
VGG19	0.092	0.242	0.302	0.383	0.343	0.394	0.487	0.526	0.214	0.243	0.290	0.330

Table.4. Metrics analysis of FPN Model on CRACK500 Dataset

Metrics	Dice Coefficient (ODS)				Pixel Accuracy				AIU Score			
Epochs Count	**20**	**40**	**60**	**Otsu**	**20**	**40**	**60**	**Otsu**	**20**	**40**	**60**	**Otsu**
ResNet18	0.305	0.379	0.494	0.572	0.450	0.523	0.694	0.751	0.312	0.362	0.481	0.546
VGG19	0.103	0.290	0.323	0.398	0.356	0.430	0.512	0.568	0.278	0.324	0.356	0.412

Table.5. Metrics analysis of U-Net Model on CFD Dataset

Metrics	Dice Coefficient (ODS)			Pixel Accuracy			AIU Score		
Epochs Count	**20**	**40**	**Otsu Thresh**	**20**	**40**	**Otsu Thresh**	**20**	**40**	**Otsu Thresh**
ResNet18	0.527	0.664	0.828	0.490	0.714	0.852	0.126	0.352	0.424
VGG19	0.376	0.543	0.615	0.219	0.506	0.625	0.103	0.235	0.357

Table.6. Metrics analysis FPN Model on CFD Dataset

Metrics	Dice Coefficient (ODS)			Pixel Accuracy			AIU Score		
Epochs Count	**20**	**40**	**Otsu Thresh**	**20**	**40**	**Otsu Thresh**	**20**	**40**	**Otsu Thresh**
ResNet18	0.485	0.632	0.756	0.507	0.615	0.781	0.107	0.260	0.354
VGG19	0.239	0.453	0.537	0.412	0.495	0.552	0.091	0.187	0.302

5. RESULTS OBTAINED

5.1 EVALUATION CRITERION

In evaluating pavement crack segmentation, maximizing accurate judgments while minimizing inaccurate detections is crucial. Evaluation involves four states: true positive (TP), false negative (FN), false positive (FP), and true negative (TN). The following performance metrics were used:

5.1.1 Pixel Accuracy (PA):

It is defined as the ratio of correctly predicted pixels to total pixels in ground truth images. The formula is given in Eq.(2)

$$PA = \frac{TP + TN}{FP + FN + TP + TN} \tag{2}$$

5.1.2 Average Intersection over Union (AIU) Score:

Measures similarity/diversity of sample images as the ratio of area of intersection to area of union. The formula is given in Eq.(3)

$$I(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{3}$$

where, *X, Y* represents Ground truth image and Predicted image

5.1.3 Dice coefficient (ODS):

Weighted average of precision and recall, ranging from 0 to 1. The formula is given in Eq.(4)

$$F_{\alpha}(precision, recall) = \frac{(1+\alpha^2)(precision \times recall)}{\alpha^2 \times precision + recall} \tag{4}$$

where α is a constant and it is assigned with an value of 0.25

5.2 CRACK500 DATASET RESULTS

Adam optimizer with a learning rate of 0.001 is employed, along with binary focal loss function parameters set to β = 0.25 and δ = 0.2. The training initially spans 20 epochs and then extends to 60 epochs in increments of 20, resulting in improved image quality by mitigating noise and artifacts, and enhancing clarity for visual assessment.

The epoch increase is halted, followed by Otsu thresholding at an appropriate level to closely match ground truth images. The output images of U-Net architecture and FPN architecture for CRACK500 dataset is depicted in Fig.6 and Fig.7. The output of U-Net architecture slightly outperforms the FPN architecture.

5.3 CFD DATASET RESULTS

The same optimizer is used for both architectures, and the impact on information loss during training is observed. Graphical representations for CFD dataset images are shown below. From Fig.8, it can be inferred that the VGG19 Encoder structure exhibits higher loss, which affects the predictions by introducing noise. The output images of U-Net architecture and FPN architecture for CFD dataset is depicted in Fig.9 and Fig.10.

5.4 METRICS ANALYSIS

We evaluate the image results against the standard ground truth images of both the CRACK500 and CFD datasets. The metrics for each 20-epoch count are listed below in the Table.3 - Table.6, showing an exponential increase in values with an increase in epochs.

5.5 STATE OF ART DETECTION

Next, we compare our results with other state-of-the-art detection methods using evaluation metrics shown in Table.7 and Table.8. Our FPN and U-Net architecture results show better performance with the ResNet18 encoder structure, which offers faster execution speed. Our model, represented by the U-Net Architecture with ResNet18 as the backbone, achieves superior AIU and ODS values compared to other methods. Specifically, we observe a significant improvement of 25% in AIU for the CFD dataset and 10% in the CRACK500 dataset.

Table.7. State-of-art comparison on CRACK500 Dataset

Methods	AIU	ODS
FPHBN [1]	0.489	0.604
DAUNet [2]	0.565	0.676
HED [6]	0.481	0.575
RCF [8]	0.403	0.490
FCN [9]	0.379	0.513
Proposed Method	0.584	0.702

Table.8. State-of-art comparison on CFD Dataset

Methods	AIU	ODS
FPHBN [1]	0.173	0.683
DAUNet [2]	0.370	0.812
HED [6]	0.154	0.593
RCF [8]	0.105	0.542
FCN [9]	0.021	0.585
Proposed Method	0.424	0.828

6. CONCLUSION

In this work, we propose the use of U-Net and Feature Pyramidal Network (FPN) architectures for crack segmentation. The U-Net architecture involves an encoder-decoder structure, where the encoder is based on a classification model with the last two layers removed, and the decoder follows a standard U-Net design. On the other hand, FPN is a type of convolutional neural network that focuses on merging semantically strong features from deeper layers with high-resolution but less semantically rich features from shallower layers to create a feature pyramid. The performance of the proposed models is evaluated using traditional metrics such as AIU, ODS, and PA. Additionally, extensive experiments are conducted to demonstrate the superiority and generalizability of our model.

DATA AVAILABILITY

The dataset on CRACK500 is available at https://github.com/fyangneil/pavement-crack-detection and the CFD dataset is available at https://github.com/cuilimeng/CrackForest.

REFERENCES

[1] L. 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, 2020.

[2] D. Polovnikov, D. Alekseev, I. Vinogradov and G.V. Lashkia, “DAUNet: A Deep Augmented Neural Network for Pavement Crack Segmentation”, *IEEE Access*, Vol. 9, pp. 125714-125723, 2021.

[3] C. Cao, Q. Liu and Z. He, “Review of Pavement Defect Detection Methods”, *IEEE Access*, Vol. 8, pp. 14531-14544, 2020.

[4] Z. Zhang, F. Yang, Y.D. Zhang and Y.J. Zhu, “Road Crack Detection using a Deep Convolutional Neural Network”, *Proceedings of IEEE International Conference on Image Processing*, pp. 3708-3712, 2016.

[5] Y. Shi, L. Cui, Z. Qi, F. Meng and Z. Chen, “Automatic Road Crack Detection using Random Structured Forests”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 12, pp. 3434-3445, 2016.

[6] S. Xie and Z. Tu, “Holistically-Nested Edge Detection”, *Proceedings of IEEE International Conference on Computer Vision*, pp. 1395-1403, 2015.

[7] D.R. Martin, C.C. Fowlkes and J. Malik, “Learning to Detect Natural Image Boundaries using Local Brightness, Colour, and Texture Features”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 5, pp. 530-549, 2004.

[8] Y. Liu, M.M. Cheng, X. Hu, K. Wang and X. Bai, “Richer Convolutional Features for Edge Detection”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5872-5881, 2017.

[9] J. Long, E. Shelhamer and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431-3440, 2015.

[10] S. Kaseko and S.G. Ritchie, “A Neural Network-Based Methodology for Pavement Crack Detection and Classification”, *Transportation Research Part C: Emerging Technologies*, Vol. 1, No. 4, pp. 275-291, 1993.

[11] Y. Liu, G. Xu, Y. Yang, X. Niu and Y. Pan, “Novel Approach to Pavement Cracking Automatic Detection based on Segment Extending”, *Proceedings of International*

Symposium on Knowledge Acquisition and Modelling, pp. 610-614, 2008.

[12] Z. Fan, C. Li, Y. Chen, J. Wei, G. Loprencipe, X. Chen, "Automatic Crack Detection on Road Pavements using Encoder–Decoder Architecture", *Materials*, Vol. 13, No. 13, pp. 2960-2983, 2020.

[13] W. Song, G. Jia, D. Jia and H. Zhu, "Automatic Pavement Crack Detection and Classification using Multiscale Feature Attention Network", *IEEE Access*, Vol. 7, pp. 171001-171012, 2019.

[14] P. Subirats, J. Dumoulin, V. Legeay and D. Barba, "Automation of Pavement Surface Crack Detection using the Continuous Wavelet Transform", *Proceedings of IEEE International Conference on Image Processing*, pp. 3037-3040, 2006.

[15] Q. Song, W. Yao, H. Tian and Y. Guo, "Two-Stage Framework with Improved U-Net Based on Self-Supervised Contrastive Learning for Pavement Crack Segmentation", *Engineering Applications of Artificial Intelligence*, Vol. 238, pp. 122406-122425, 2024.

[16] W. Huang and N. Zhang, "A Novel Road Crack Detection and Identification Method using Digital Image Processing Techniques", *Proceedings of International Conference on Computing and Convergence Technology*, pp. 397-400, 2012.

[17] L. Peng, W. Chao, L. Shuangmiao and F. Baocai, "Research on Crack Detection Method of Airport Runway based on Twice-Threshold Segmentation", *Proceedings of International Conference on Instrumentation and Measurement, Computer, Communication and Control*, pp. 1716-1720, 2015.

[18] W. Xu, Z. Tang, J. Zhou and J. Ding, "Pavement Crack Detection based on Saliency and Statistical Features", *Proceedings of IEEE International Conference on Image Processing*, pp. 4093-4097, 2013.