# Wavelet U-Net for Automated Crack Segmentation and Detection

Navodita Mathur[1], Brittany Gomez[2]
University of Pittsburgh

1 School of Computing and Information Science
Email- nam266@pitt.edu
2 Department of Biomedical Informatics
Email- b.gomez@pitt.edu

## Abstract

Road crack detection is an important task in infrastructure maintenance and safety. Detecting cracks in roads helps identify areas that need repair and ensures road safety. Automatic crack detection is always a challenging task due to the inherent complex backgrounds, uneven illumination, irregular patterns, and various types of noise interference. This study proposes an effective, fully connected network, that uses a self-attention mechanism to capture contextual information effectively. A well-optimized U-Net-based model like ours can achieve high accuracy, with IoU scores often exceeding 0.8 or 0.9 on high-quality datasets.

## 1. Introduction

Image classification has a rich history that has continued for decades. Computer vision began with edge detection via matrices of particular values that would yield images with similar patterns given the filter. The similarity in patterns was exploited for classification tasks. Eventually, neural networks started to develop their own filters which were usually black box parameters with the ability to classify images. Traditional neural nets were improved when convolution neural nets (CNN) stepped in to tackle image classification. It continues to be state of the art with more scientists adding layers and finding tricks to bring solutions to the vanishing gradient within deep layered models. Although concrete crack images can come from large datasets, which would generally capture a multitude of conditions, crack detection shares the high noise seen in medical imaging modalities along with the potential of the classification of images relying on limited pixels, or fine detail.

Over the past few years, computational medicine has turned to models that can combine low and high-level features along with recovering spatial information. Compared to state-of-the-art CNNs, U-Net models have skipped connections between encoder and decoder components which allows information to flow directly from the encoder to the decoder. Since there are shared weights between the encoder and decoder this causes fewer parameters which reduces the chance of overfitting. U-Nets also make sure the feature maps of the encoder and decoder are the same size. This isn't always true for CNNs which causes the loss of information during upsampling. U-Nets are better at dealing with imbalanced datasets as the loss function concentrates on the boundary regions between the classes, which improves the segmentation of smaller classes. This could be useful in concrete crack detection by triaging the severity and possibly automating a schedule of tasks.

Even with the improvements of skipped connections to incorporate local and global information, there are still irreversible losses of information when max pooling is used after each layer. On the other hand, without pooling layers, there is a risk of overfitting, especially when dealing with limited training data. To try and reorient the problem a new pooling operation will be tried, wavelet U-Net (WUNet). The WUNet uses a wavelet transform for the downsampling and upsampling. The encoder uses discrete wavelet transformation (DWT) which will replace the pooling operation. This will preserve the features' frequency and location, which increases texture information. Inverse wavelet transform (IWT) will be used to restore resolution within the decoder. Since there is invertibility between DWT and IWT no information is lost. This will be an asset when determining the differences between small nuances in pixels where no information has been lost.

The WUNet will have a densely cross-level connection

which will encourage features to be reused and increase complementarity among cross-level information.

Figure 1. Road Crack

Traditionally crack detection includes manual inspection by engineers, which can be time-consuming. An automated crack detection would save time, and money, and prevent engineers from going into unsafe conditions.

## 2. Related Work

During the 1990s and 2000s, researchers began to explore edge detection techniques, such as the Canny edge detector and the Sobel operator, to locate the boundaries of cracks. Additionally, morphological operations, including erosion and dilation, were used to refine crack segmentations and improve the accuracy of crack boundary detection. Researchers explored the use of texture analysis and feature-based methods to identify cracks. Techniques like Gabor filters and Local Binary Patterns (LBP) were employed to capture the unique textural characteristics of cracks, which could be used for segmentation. In recent years, machine learning methods, including classical algorithms like Support Vector Machines (SVM) and Random Forest, have been applied to crack segmentation. These methods use labeled data to train models to classify pixels or regions as crack or non-crack. The advent of deep learning, particularly convolutional neural networks (CNNs), has brought a significant transformation to crack segmentation. CNNs, such as U-Net and Mask R-CNN, have demonstrated remarkable success in pixel-wise and object-level crack segmentation. Transfer learning, data augmentation, and the availability of large datasets have contributed to the rapid progress in deep learning-based crack segmentation. [3] outlines the development history, research results and related applications of computer vision in the field of concrete crack recognition.

### 2.1. Convolutional Neural Networks

Many techniques based on Deep Convolutional Neural Networks have been proposed to detect road cracks, given their remarkable success in various other computer vision tasks. These techniques can be divided into three groups based on how crack detection is carried out: pixel-level segmentation, object-based techniques, and pure image classification methods.

#### 2.1.1 Pure Image Classification

Some researchers have carried out image-level classification studies, which mainly solve the problem of determining whether a surface contains cracks and, if so, what type of cracks. Ma et al. [8] developed a deep learning method for road detection and evaluation based on convolutional neural network, Fisher vector coding, and UnderBagging random forest. Zhang et al. Pang-jo Chun et al.[2] proposes the use of a Light Gradient Boosting Machine model for automatic crack detection and compares the results with pix2pix-based approach. The study generates crack features using pixel values and geometric shapes and achieves an accuracy of 99.7%. [16] proposed a six-layer CNN network with four convolutional layers and two fully connected layers and used their convolutional neural network to train $99 \times 99 \times 3$ small patches, which were split from $3264 \times 2248$ road images collected by low-cost smartphones. Their study shows that deep CNNs are superior to traditional machine learning techniques, such as SVM and boosting methods, in detecting pavement cracks. [11] proposed a four-layered simple Convolutional Neural Network for automatic crack detection which is concluded as highly efficient yet accurate with an accuracy of 98.3%. The author argues that the model's simplicity enables it to work on low-quality images that eliminates the need for costly digital image-capturing devices. [5] uses transfer learning to develop and validate a CNN suitable for crack detection using images that are pre-processed. Zhang et al. (2017) demonstrated the feasibility of CrackNet (a form of CNN). The CrackNet unlike usual CNN algorithms uses line filters to enhance the contrast between cracks and the background for preprocessing with no pooling layers to remove the downsampling. The study verified that the CrackNet CNN is more effective than SVM and non-ML methods and that the pooling layers may not assist in crack detection.

Apart from CNN Classification, there are other models that separates the image into multiple smaller images (patches) and then performs crack classification. This is Patch-level classification. It has two main advantages; first, we can generate more data due to the division of the image into smaller patches. Second, it gets easier to localize the

existing cracks in the image by working on every patch of the original image.

#### 2.1.2 Object Detection

The classification algorithms help in identifying the presence or absence of cracks in the images of surface irrespective of their position. Object detection adds one step further in classification to find the location of cracks within an image. [6] uses 3 modules, Base architecture, Objectness Score Identification (OSI) Network and Region of Interest (ROI) pooling to first detect and then classify the cracks The base architechture is inspired from Feature Pyramid Network (FPN) to extract features from the input image before image segmentation. Cha et al. [1] adopted the modified ZF-net as the CNN feature extractor of Faster R-CNN. This helped in accelerating the process of feature extraction and was more suitable for real-time detection.

#### 2.1.3 Pixel-level image segmentation

Pixel-level image segmentation is ideal for detailed crack segmentation tasks where the goal is to provide precise delineation of crack boundaries in the image. It provides a pixel-wise binary mask that accurately identifies the location of cracks in the image, enabling exact boundary delineation. It is essential when the main objective is to precisely segment and measure cracks in images, such as in structural health monitoring and detailed defect analysis.

Zhang et al. put forward CrackNet [16], a study on pixel-level crack detection based on CNN in earlier years. The prominent feature of CrackNet is using a CNN model without a pooling layer to retain the spatial resolution. Fei et al. [4] have upgraded it to Cracknet-V, an improved version of it with deeper architecture but fewer parameters, resulting in improved accuracy and computation efficiency. While CrackNet and its series versions perform well, they are primarily used for 3D road crack images

In recent years, Semantic segmentation using fully convolutional networks (FCN), encoder-decoder architectures, and related methods has become a major research focus in the field of pixel-level image segmentation. Some of its pioneer methods being but not limited to FCN, SegNet, and U-Net. Zou et al. [17]proposed an end-to-end deep convolutional neural network (DeepCrack) to realize the automatic detection of cracks by learning high-level characteristics of cracks. DeepCrack incorporates multi-scale deep convolutions to capture linear structures at different levels. This allows the network to learn hierarchical features and better detect cracks with varying widths and scales. In [12], a lightweight end-to-end pixel by pixel classification network (SegNet) was used to detect cracks. SegNet uses max-pooling indexes obtained during the encoder's pooling steps to implement non-linear upsampling in the decoder. This approach simplifies the learning process and reduces the need to learn how to up-sample. However, it can be computationally intensive and may require a significant amount of labeled data for training.

### 2.2. Transformer-Based Methods

In recent years, transformers have made great breakthroughs in CV, and it was quickly introduced into the field of crack segmentation. In [14], a novel SegCrack model for pixel-level crack segmentation is proposed using a hierarchically structured Transformer encoder to output multiscale features and a top-down pathway with lateral connections to progressively up-sample and fuse features from the deepest layer of the encoder. Furthermore, it adopted an online hard example mining strategy to strengthen the detection of hard samples and improve the model performance, resulting in precision, recall, and F1 score of 96.66%, 95.46%, 96.05%, and 92.63% respectively. In [13], a convolutional-transformer network based on an encoder-decoder architecture with Dilated Residual Block (DRB) which is combined with a lightweight transformer that captures global information to serve as an effective encoder and a Boundary Awareness Module (BAM) was proposed. The DRB captures the local detail of cracks and adjusts the feature dimension for other blocks as needed whereas the BAM learns the boundary features from the dilated crack label. This study proposes [15] proposes a dual-encoder network fusing transformers and convolutional neural networks (DTrC-Net) to alleviate the influence of irregularly shaped cracks, complex image backgrounds, and to overcome limitations in acquiring global contextual information.

Transformers have shown promising results and, in some cases, have outperformed traditional convolutional neural networks (CNNs) by leveraging the advantages of self-attention for capturing long-range dependencies and modeling global image information. Transformer architectures need more training data to achieve equal or improved accuracy than CNNs.[10]

The proposed model tries to leverage accuracy by using the best of both. It uses the same attention mechanism along with traditional convolutional layers to improve accuracy in the semantic segmentation of road cracks.

## 3. Method

### 3.1. Network Architechture

The WU-Net model is named for it's U-like architecture. 2 The WU-Net is defined by the contracting path and the expanding path. On the contracting path, it can be seen that there is a combination of convolution layers, common in CNNs, and DWT. This combination captures contextual information and downsamples the feature maps. ReLU activation functions are used within the hidden layers of the contracting path to address the vanishing gradient problem. It also employs a cross-level fusion strategy to increase the efficiency of feature map reuse and to fuse feature information across the downsampling level.

The expansive path is composed of transposed convolutional layers to upsample the feature maps, which recover the original spatial resolution of the image. There are concatenations from the contracting feature maps to the expansive feature maps; these are referred to as skipped connections. The skipped connections allow the model to combine the low-level and high-level features. However, the problem with skip connections is that low-level feature maps are simply passed backwards, and there is a significant amount of redundancy in these feature maps. The naive skip connections cannot distinguish the more valuable and important parts of the information. To overcome this, we employ self-attention modules, which can assign more weight coefficients to important regions to make the network more focused on specific local regions.

The activation for the final layer is a softmax activation. Softmax is a popular activation function due to it simplifying the training process with smoother gradients which can lead to faster convergence time. Although the goal of the segmentation is to produce a binary mask, softmax is often used to help normalize the outputs and provide a probabilistic interpretation. This would generally lead to the softmax function producing a two-dimensional probability distribution that sum to one. However, a U-Net with a softmax is able to produce a more nuanced output than a simple binary classification. For image segmentation, the model may be able to predict $20\%$ likely one one class and $80\%$ to another which is useful when distinguishing between classes of finely detailed pixels.

While training, a $50\%$ dropout at the end of the contracting path along with the l2 regularizer with lambda as $1e-4$, helps reduce the overfitting of the model.

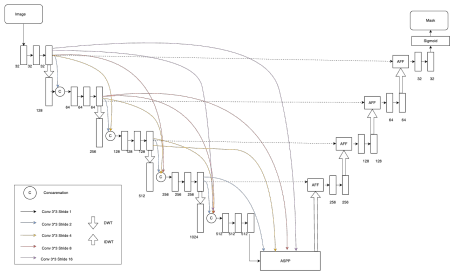

Figure 2. The architecture of a WU-Net.

#### 3.1.1 Wavelet Transformations

The contracting path employs both convolutional layers and discrete wavelet transform (DWT). Given an image x, we use 2D DWT with four convolutional filters, i.e. low-pass filter $f_{LL}$, and high-pass filters $f_{LH}$, $f_{HL}$, $f_{HH}$, to decompose x into four subband images, i.e. $x_{LL}$, $x_{LH}$, $x_{HL}$, and $x_{HH}$.

We have used Haar wavelet transformation because of its simplicity with stride 2. The filters here are defined as $f_{LL}=\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}, f_{LH}=\begin{bmatrix}-1 & -1\\ 1 & 1\end{bmatrix}, f_{HL}=\begin{bmatrix}-1 & 1\\ -1 & 1\end{bmatrix}, f_{HH}=\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}$.

The operation of DWT is defined as $x_{LL} = (f_{LL} \circledast x) \downarrow_2$, $x_{LH} = (f_{LH} \circledast x) \downarrow_2$, $x_{HL} = (f_{HL} \circledast x) \downarrow_2$ and $x_{HH} = (f_{HH} \circledast x) \downarrow_2$, where $\circledast$ denotes convolution operator, and $\downarrow_2$ means the standard downsampling operator with factor 2. In other words, DWT mathematically involves four fixed convolution filters with stride 2 to implement the downsampling operator.

Moreover, according to the theory of Haar transform [9], the (i, j)-th value of $x_{LL}$, $x_{LH}$, $x_{HL}$, and $x_{HH}$ after 2D Haar transform can be written as:

$$x_{LL}(i,j)=x(2i-1,2j-1)+x(2i-1,2j)+x(2i,2j-1)+x(2i,2j)$$
$$x_{LH}(i,j)=-x(2i-1,2j-1)-x(2i-1,2j)+x(2i,2j-1)+x(2i,2j)$$
$$x_{HL}(i,j)=-x(2i-1,2j-1)+x(2i-1,2j)-x(2i,2j-1)+x(2i,2j)$$
$$x_{HH}(i,j)=x(2i-1,2j-1)-x(2i-1,2j)-x(2i,2j-1)+x(2i,2j)$$

Figure 3 shows an illustration of 2 levels of DWT.

Although the downsampling operation is deployed, due to the biorthogonal property of DWT, the original image x can be accurately reconstructed without information loss by the Inverse Wavelet Transform, IWT, i.e., $x = IWT(x_{LL}, x_{LH}, x_{HL}, x_{HH})$. For the Haar wavelet, the IWT can defined as:

$$x(2i-1,2j-1)=(x_{LL}(i,j)-x_{LH}(i,j)-x_{HL}(i,j)+x_{HH}(i,j))/4$$
$$x(2i,2j-1)=(x_{LL}(i,j)-x_{LH}(i,j)+x_{HL}(i,j)-x_{HH}(i,j))/4$$
$$x(2i-1,2j)=(x_{LL}(i,j)+x_{LH}(i,j)-x_{HL}(i,j)-x_{HH}(i,j))/4$$
$$x(2i,2j)=(x_{LL}(i,j)+x_{LH}(i,j)+x_{HL}(i,j)+x_{HH}(i,j))/4$$

These transformations can be sequentially decomposed by DWT for further processing, resulting in multi-level architecture.

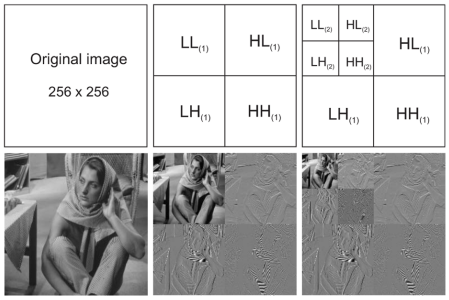


Figure 3. Discrete Wavelet Transformations

DWT can provide a compact representation of both high and low-frequency components in the image. While both multi-level architecture and CNN are capable to being fully-functional on their own, in some architectures, CNNs and DWT can be used in conjunction to leverage both spatial hierarchies (learned by CNNs) and frequency information (provided by DWT)[7]. The key idea is to insert CNN blocks into the architecture before (or after) each level of DWT to replace the pooling layers4.

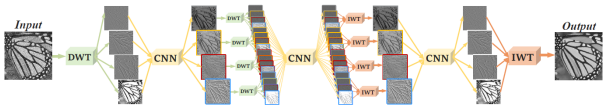


Figure 4. Discrete Wavelet Transformations

### 3.1.2 Cross-level Fusion

As shown in Fig. 2, we use the CLF strategy in four downsampling layers and an ASPP module as shown in Fig. 6 to ensure enhanced cross-level feature connection and complementarity between cross-level information.

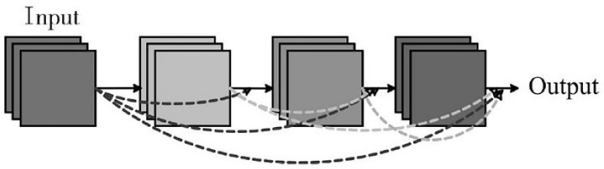


Figure 5. Cross-level Connections

For each feature map before the downsampling operation, we use the convolution with specific strides to downsample it and concatenate it with the corresponding feature map obtained by DWT. The downsampled layer is formed by convolving the previous layer's output with strides 2 concatenated with the one formed by convolving the layer before the previous's layer output with stride 4 and so on, as shown in fig. 5

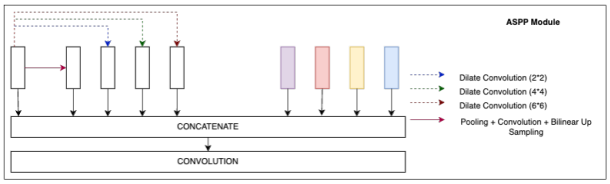


Figure 6. Atrous spatial pyramid pooling

The four levels of features from the downsampling path are combined with the four scale features obtained using dilated convolution at rates 2,4, and 6 along with one level with average pooling and bilinear upsampling in the ASPP to produce features of nine scales. Thereby, the network not only obtains five scale features from high-level semantic information but also texture and position information from CLF in decoding.

### 3.1.3 Attention Feature Fusion Module

Attention mechanisms, including self-attention, have been employed to capture long-range dependencies and relationships within an image. The architecture employed here is relatively simple, as shown in Fig 7.

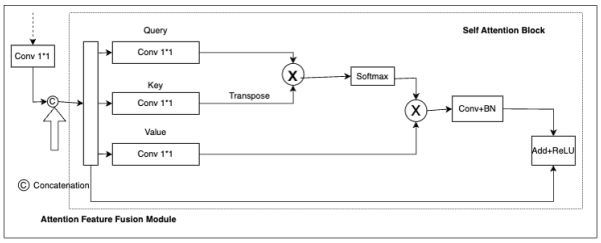


Figure 7. Attention Feature Fusion

Key, Query, and Value transformations are applied to

the input tensor x using Conv2D layers. The block first calculates attention scores by measuring the similarity between the Query vector and the Key vector using a dot product. The attention scores are normalized using a softmax function to obtain attention weights. These weights represent how much focus should be given to each element in the sequence. The Value vectors are multiplied by their corresponding attention weights and summed up. This weighted sum becomes the output. To make it fully functional, we add a convolution layer followed by batch normalization, which is then element-wise added to the input. This is followed by a ReLU activation.

This helps the model know which features to pay attention to before passing to convolution layers present at that level.

# 4. Experiment

## 4.1. Data

The dataset consists of surface cracks for segmentation from Kaggle. The sample contains 11,200 images that are merged from 12 available crack segmentation data sets. The images have been segmented to create a mask of the cracked portion.

## 4.2. Implementation Details

We implemented the model using TensorFlow with GPU. The model was trained for 30 epochs using 5-fold cross-validation, with the validation set as 20% of the data. The model is compiled using the Adam optimizer with the learning rate set as 1e-4. The buffer size is set as 1000 and the batch size as 8.

The input image size is adjusted to (256,256) pixels, and the encoder uses convolutional layers with a convolution kernel size of 3. Early stopping with patience as 5 is used to help generalize the model.

## 4.3. Evaluation

The loss function chosen is binary cross entropy (BCE). BCE measures the difference between predicted probabilities and the ground truth labels. BCE is commonly used for binary classification problems, like image segmentation. The main function of BCE is to enforce class balance. The combination of BCE and softmax makes this model ideal for tasks where a probabilistic interpretation of the results is necessary.

It is used along with metrics, Dice coefficient and IOU both of which help measure the overlap of the predicted mask with the ground-truth mask, to evaluate the performance and fine-tune the model.

# 5. Results and Discussion

The evaluation for a U-Net binary mask considers the predicted mask against the ground truth mask for each pixel in the image. In binary mask problems, this means that 1 is equivalent to foreground while 0 is background. This yields a percentage of pixels that were correctly classified.

## 5.1. Metrics

### 5.1.1 Dice Coefficient

The Dice Coefficient, or F1 Score, measures the similarity between the two sets. It is calculated using the following formula:

$$DiceCoefficient = \frac{2|A \cap B|}{|A| + |B|}. \tag{1}$$

Where intersection is the number of pixels that are common to both the predicted and ground truth sets and the total number of pixels in both sets is the sum of pixels in the predicted set and the ground truth set.

The Dice Coefficient ranges from 0 to 1, with 0 indicating no overlap between the sets (complete dissimilarity) and 1 indicating a perfect match (complete similarity). Higher Dice Coefficient values correspond to better segmentation performance. In the context of image segmentation, the sets being compared are often the pixels predicted by a model (the segmentation mask) and the true segmentation mask (the ground truth). The Dice Coefficient provides a measure of how well the predicted segmentation aligns with the actual segmentation.

### 5.1.2 Intersection over Union

This metric is commonly used in image segmentation and object detection tasks. IOU measures the overlap between the predicted and ground truth regions. It is also known as the Jaccard Index. The IOU is calculated using the formula:

$$IOU = \frac{A \cap B}{A \cup B}. \tag{2}$$

Where,
Intersection: Intersection is the area (or volume in 3D) common to both the predicted and ground truth regions.
Union: Union is the total area (or volume) encompassed by both the predicted and ground-truth regions.

IOU values range from 0 to 1, with 0 indicating no overlap (complete dissimilarity) and 1 indicating a perfect match (complete similarity). It provides a measure of the spatial overlap between the predicted and true regions, offering insights into the accuracy of the segmentation.

#### 5.1.3 Area Under Curve

Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC) are commonly used metrics for evaluating the performance of binary classification models. They are particularly useful when dealing with imbalanced datasets.

The ROC curve is a graphical representation of the model's ability to discriminate between the positive and negative classes across various threshold values. The curve represents the trade-off between true positive rate and false positive rate as the classification threshold varies. AUC represents the area under the ROC curve. It provides a single scalar value summarizing the overall performance of the model.

AUC ranges from 0 to 1. A model with an AUC of 0.5 performs no better than random chance, while a model with an AUC of 1.0 indicates perfect classification. Generally, a higher AUC indicates better discrimination ability of the model.

### 5.2. Results

The model is trained for 30 epochs, showing a loss reduction8. This method would hold out 10% of the data as a testing set that was not used in the training of the models. The performance of the model on the test dataset is detailed in Table 1.

The loss is calculated as:

$$BCE = -\frac{1}{N}\sum_{i=1}^{n}[Y_i \log p_i + (1 - Y_i) \log (1 - p_i)] \quad (3)$$


Figure 8. Results: Loss and Performance Metrics on Training and Validation sets

Table 1. Model Performance

| | |
|---|---|
| Dice Coefficient | 0.8 |
| IOU | 0.64 |
| Area under the curve (AUC) | 0.88 |

The model has led to an increase in Test F1-score by 0.4 as compared to a normal ResNet model.

The masks predicted are visualized as:

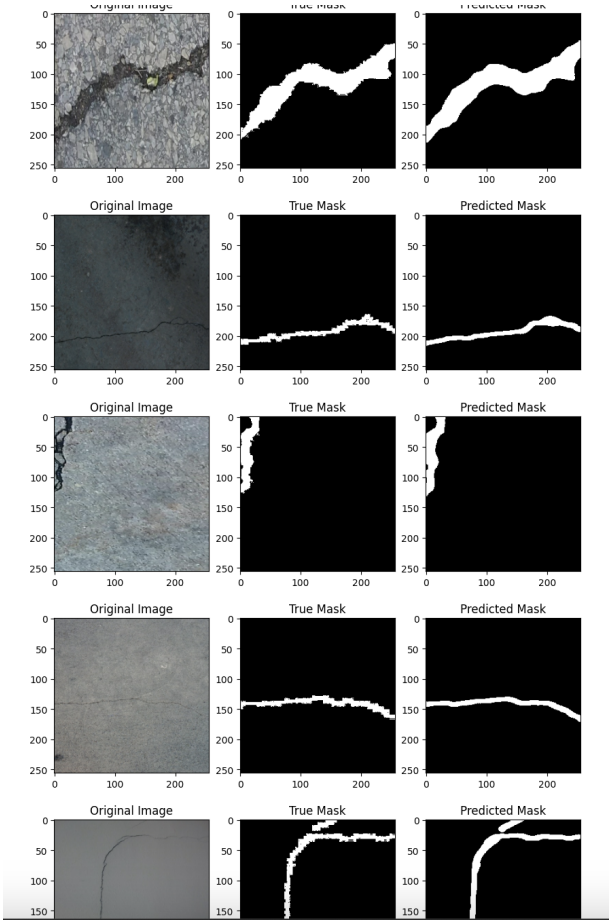


Figure 9. Results: Predicted masks

## 6. Future Work

### 6.1. Data Augmentation

Augmentation is a technique used in deep learning to artificially increase the diversity of the training dataset by applying various transformations to the existing images. The goal is to improve the model's generalization and robustness by exposing it to different variations of the input

data. Common image augmentations include:
Rotation: Randomly rotating images to different angles.
Flip: Flipping images horizontally or vertically.
Zoom: Randomly zooming in or out of images.
Translation: Shifting images horizontally or vertically.
Brightness and Contrast: Adjusting the brightness and contrast of images.
Color Jitter: Introducing random variations in color.

Augmentation is particularly useful when dealing with limited training data, as it helps prevent overfitting and improves the model's ability to handle variations in the input.

### 6.2. Transfer Learning

Transfer learning involves leveraging knowledge gained while solving one problem and applying it to a different but related problem. In the context of deep learning, transfer learning often refers to using pre-trained models on large datasets for a specific task and adapting them to a new task with a smaller dataset. This can help increase the accuracy of the predictions.

## 7. Conclusion

Crack segmentation is still an important research field in the engineering field of image recognition technology, a task that is of great significance for prolonging the service life of roads and enhancing safety. To help mitigate this issue, we have developed a WU-Net model with dense cross-level connections and self-attention mechanisms to capture the fine details as well as the context information necessary for crack segmentation.

## 8. Comments

This approach took effort and time to train 3 separate models ResNet, Unet, and the final model to help see the contribution of each of the modules, if batch normalization was required, and a lot of fine-tuning with the number of layers, etc. to help arrive at this model. This model was designed primarily for biomedical imaging and then applied to crack segmentation due to a lack of cleaned data. The primary design was by Brittany, and then the incorporation of CLF and self-attention modules was done by me.

## References


[1] Young-Jin Cha, Wooram Choi, Gahyun Suh, Sadegh Mahmoudkhani, and Oral Büyüköztürk. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, 33(9):731–747, 2018. 3

[2] Pang-jo Chun, Shota Izumi, and Tatsuro Yamane. Automatic detection method of cracks from concrete surface imagery using two-step light gradient boosting machine. *Computer-Aided Civil and Infrastructure Engineering*, 36, 2020. 2

[3] Jianghua Deng, Amardeep Multani, Yiyi Zhou, Ye Lu, and Vincent Lee. Review on computer vision-based crack detection and quantification methodologies for civil structures. *Construction and Building Materials*, 356:129238, 2022. 2

[4] Yue Fei, Kelvin CP Wang, Allen Zhang, Cheng Chen, Joshua Q Li, Yang Liu, Guangwei Yang, and Baoxian Li. Pixel-level cracking detection on 3d asphalt pavement images through deep-learning-based cracknet-v. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):273–284, 2019. 3

[5] Vaughn Peter Golding, Zahra Gharineiat, Hafiz Suliman Munawar, and Fahim Ullah. Crack detection in concrete structures using deep learning. *Sustainability*, 14(13), 2022. 2

[6] Deepa Joshi, Thipendra P. Singh, and Gargeya Sharma. Automatic surface crack detection using segmentation-based deep-learning approach. *Engineering Fracture Mechanics*, 268:108467, 2022. 3

[7] Pengju Liu, Hongzhi Zhang, Wei Lian, and Wangmeng Zuo. Multi-level wavelet convolutional neural networks. *IEEE Access*, 7:74973–74985, 2019. 5

[8] Ke Ma, Minh Hoai, and Dimitris Samaras. Large-scale continual road inspection: Visual infrastructure assessment in the wild. In *BMVC*, 2017. 2

[9] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989. 4

[10] José Maurício, Inês Domingues, and Jorge Bernardino. Comparing vision transformers and convolutional neural networks for image classification: A literature review. *Applied Sciences*, 13(9), 2023. 3

[11] Mihir Padsumbiya, Vedant Brahmbhatt, and Sonal Thakkar. Automatic crack detection using convolutional neural network. *Journal of Soft Computing in Civil Engineering*, 6(3): 1–17, 2022. 2

[12] Chunge Song, Lijun Wu, Zhicong Chen, Haifang Zhou, Peijie Lin, Shuying Cheng, and Zhenhui Wu. Pixel-level crack detection in images using segnet. In *International Conference on Multi-disciplinary Trends in Artificial Intelligence*, pages 247–254. Springer, 2019. 3

[13] Huaqi Tao, Bingxi Liu, Jinqiang Cui, and Hong Zhang. A convolutional-transformer network for crack segmentation with boundary awareness. *arXiv preprint arXiv:2302.11728*, 2023. 3

[14] Wenjun Wang and Chao Su. Automatic concrete crack segmentation model based on transformer. *Automation in Construction*, 139:104275, 2022. 3

[15] Chao Xiang, Jingjing Guo, Ran Cao, and Lu Deng. A crack-segmentation algorithm fusing transformers and convolutional neural networks for complex detection scenarios. *Automation in Construction*, 152:104894, 2023. 3

[16] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In *2016 IEEE international conference on image processing (ICIP)*, pages 3708–3712. IEEE, 2016. 2, 3

[17] Qin Zou, Zheng Zhang, Qingquan Li, Xianbiao Qi, Qian Wang, and Song Wang. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE transactions on image processing*, 28(3):1498–1512, 2018. 3