Automatic Crack Detection Using Encoder-Decoder Architecture

Nathan Loh, Nikolaus Schultze
Department of Computer Science
Illinois Institute of Technology

April 17, 2023

Problem Statement

Several factors, including stress from heavy loads, water intrusion, wear and tear, and shrinkage over time, can cause concrete cracking. These cracks can pose health risks to humans, animals, and vehicles. They can be a tripping hazard, scrap vehicles, or allow mold to form. Cracks can be prevented with regular maintenance and proper monitoring. However, many places are not capable of being able to afford this regular maintenance and monitoring. [3]

Background Materials

Data: The data we obtained from Kaggle and contains 2 folders: Positive and Negative. The Positive folder contains 20,000 images of concrete with cracks and the Negative folder contains 20,000 images with no cracks.

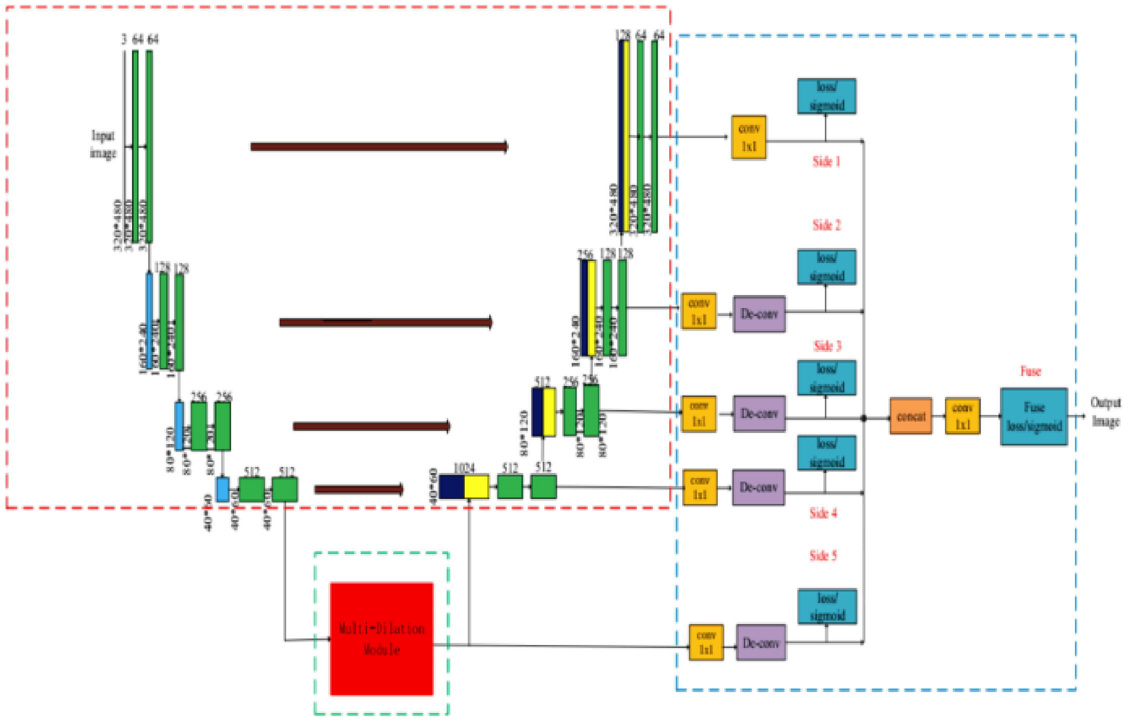
Proposed Solution

Our proposed solution would be to create a U-Hierarchical Dilated Network (U-HDN), which utilizes an encoder-decoder algorithm to detect a crack. In addition, we would test other models, such as a simple Convolutional Neural Network, a complex Convolutional Neural Network and VGG16. We will then compare the results of these models' results with the U-HDN algorithm's results.

Implementation Details

The U-HDN algorithm consists of three components: a U-net architecture to extract the features and restore the image, a multi-dilation module to get more information about the features, and a hierarchical feature learning module to obtain multi-scale features. We will take multiple images of road pavement, where some have cracks while others do not, and split them into training, validation, and testing groups before feeding them into the network. The model will be trained on parts of the dataset, and when tested by inputting images, it will state whether there are cracks or not. [1]

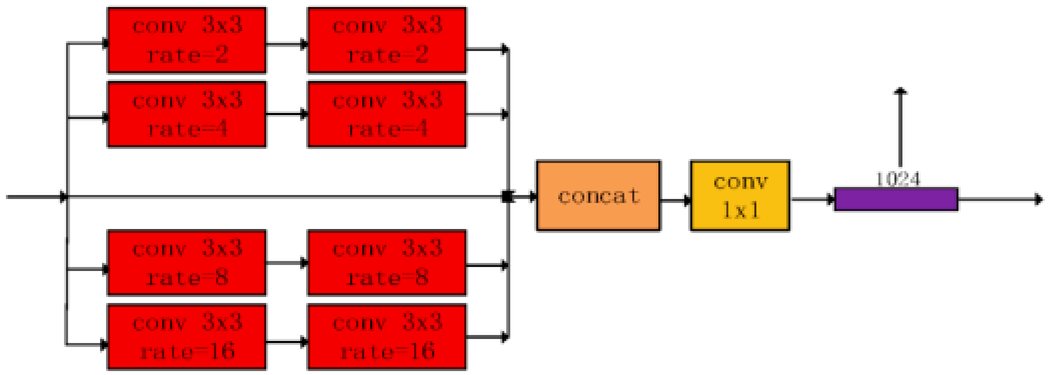
UDHN Architecture



The proposed U-HDN architecture consists of three components: U-net architecture, multi-dilation module, and hierarchical feature learning module. The red dotted box presents the modified U-net; the green dotted box is a multi-dilation module; the blue dotted box shows the hierarchical feature learning module. [1]

U-Net Architecture:
A U-Net contains 2 parts: an encoder and decoder
Encoder(Contracting Path): Consists of two 3x3 Convolution Layers, with a batch normalization and dropout layer in between, that are followed by an ReLU and a 2x2 Max Pooling Layer that is used for down sampling and another batch normalization layer before the pooling layer.
Decoder (Expansive Path): Consists 2x2 Up Convolution, the features that were cropped out of the Encoder, two 3x3 Convolutional Layers, followed by an ReLU. It also has a batch normalization and dropout layer between the two convolutional layers and another batch normalization layer after the second convolution layer. [1]

Multi-Dilation Module (MDM):
Consists of 4 paths of two 3x3 Convolutional Layers with different rates (2, 4, 8, 16). These are then concatenated and then imputed into a 1x1 Convolution Layer.



[1]

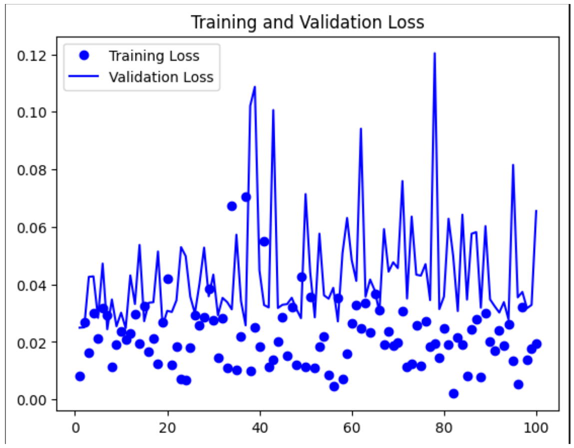
Hierarchical Feature (HF):
Consists of convolutional layers with different kernel sizes of 1, 3, and 5, as well as different dilation rates of 2, 4, and 8 before being concatenated.

U-HDN Results

Recall	.396
Precision	0.496
F1-Score	0.440
ODS	0.597
OIS	0.025 (Invalid)
Accuracy	0.497
Positive Recall	0.50
Negative Recall	0.50
Positive Precision	0.40
Negative Precision	0.60
Positive F1-Score	0.44
Negative F1-Score	0.54

The model is only about 50% accurate at predicting whether or not an image is positive or negative for a crack. The model predicts Positive for about half of the positive samples but also positive for about half of the negative samples. This model can be easily improved by switching the threshold by which we determine whether or not the image is Positive or Negative, which can be shown by the elevated ODS.

Simple CNN Results

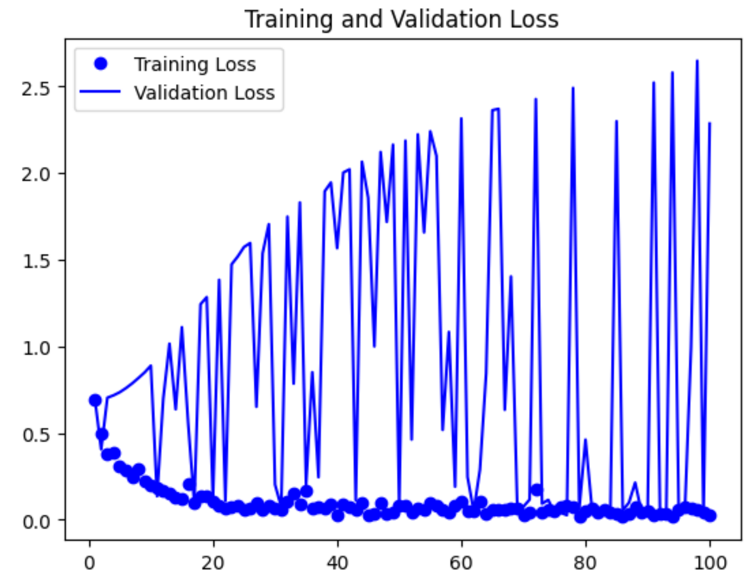


The simple CNN model overfits which can be shown by the Validation Loss being higher than the Training Loss.

Recall	0.496
Precision	0.499
F1-Score	0.497
ODS	0.501
OIS	0.02475 (Invalid)
Accuracy	0.499
Positive Recall	0.50
Negative Recall	0.50
Positive Precision	0.50
Negative Precision	0.50
Positive F1-Score	0.50
Negative F1-Score	0.50

The model is only about 50% accurate at predicting whether or not an image is positive or negative for a crack. The model predicts Positive for about half of the positive samples but also positive for about half of the negative samples.

Complex CNN Results



The complex CNN model overfits which can be shown by the Validation Loss being higher than the Training Loss. The spikes heights are also increasing, enforcing the point that the model has been overfit on the Training data.

Recall	0.982
Precision	0.500
F1-Score	0.662
ODS	0.666
OIS	0.0245 (Invalid)
Accuracy	0.500
Positive Recall	0.50
Negative Recall	0.52
Positive Precision	0.98
Negative Precision	0.02
Positive F1-Score	0.66
Negative F1-Score	0.04

The model is not very good or accurate at predicting whether or not an image is positive or negative for a crack. The model predicts Positive for most of the samples, resulting in a near-perfect Positive Recall and a near 0 Negative Recall.

CNN with Encoder and Decoder Results

Recall	0.0
Precision	0.0
F1-Score	0.0
ODS	0.0
OIS	0.0 (Invalid)
Accuracy	0.5

Positive Recall	0.0
Negative Recall	1.0
Positive Precision	0.0
Negative Precision	0.5
Positive F1-Score	0.0
Negative F1-Score	0.67

The model is not very good or accurate at predicting whether or not an image is positive or negative for a crack. The model predicts Negative for every sample, resulting in a perfect Negative Recall and a 0 Positive Recall.

Results Comparison

	U-HDN	Simple CNN	Complex CNN	CNN
Recall	.396	0.496	0.982	0.0
Precision	0.496	0.499	0.500	0.0
F1-score	0.440	0.497	0.662	0.0
ODS	0.597	0.501	0.666	0.0
OIS	0.025 (Invalid)	0.02475 (Invalid)	0.0245 (Invalid)	0.0 (Invalid)
Accuracy	0.497	0.499	0.500	0.5

Conclusion:

In Conclusion, we see that the model that performs the best for correctly predicting Positive and negative samples is the U-HDN. The U-HDN can predict Positive and Negative samples correctly about half the time. We could potentially increase the accuracy of this model by training the model with an increased number of Epochs and by finding the optimal threshold. The Complex CNN predicts almost only Positive, which results in its Positive results being near perfect and its Negative Results being near 0. The CNN with Encoder and Decoder predicts only Negative, which results in its Negative Results being perfect and its Positive results being 0.

References

[1] Fan, Z., Li, C., Chen, Y., Wei, J., Loprencipe, G., Chen, X., & Di Mascio, P. (2020). Automatic crack detection on road pavements using encoder-decoder architecture. Materials, 13(13), 2960. https://doi.org/10.3390/ma13132960

[2] Gad, A. F. (2021, April 9). Accuracy, precision, and recall in deep learning. Paperspace Blog. Retrieved April 15, 2023, from https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/

[3] Hardrock. (2021, April 8). Beware of floor cracks: The dangers concrete cracks can cause. Hard Rock Concrete Coatings | Utah Concrete Coating Contractor. Retrieved March 25, 2023, From https://www.hardrockconcretecoatings.com/beware-of-floor-cracks-the-dangers-concrete cracks-can-cause/#:~:text=Mold%2D%20if%20water%20is%20able,a%20source%20o %20potential%20injury.

[4] Özgenel, Çağlar Fırat (2019), "Concrete Crack Images for Classification", Mendeley Data, v2 http://dx.doi.org/10.17632/5y9wdsg2zt.2

[5] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.