BrainStation Capstone: Concrete Crack Detection

Prepared By:
- Sepehr Salimi

Prepared For:
- BrainStation Capstone

Problem Statement

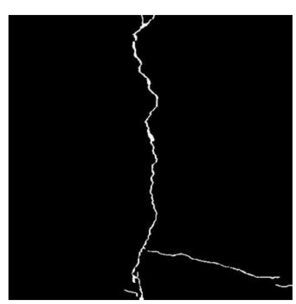
The purpose of this project was to develop a concrete crack detection model that identifies and crack masks in concrete. The presence of cracks in concrete can make structures vulnerable to external effects, accelerate aging and potentially lead to structural collapse. Existing solutions such as Ultrasonic Pulse Velocity and Ultrasonic Guided Waves are expensive for small companies, while high-security companies like those in the Nuclear industry, require many clearances. By offering a cost-effective alternative solution through software and drone imagery, regular maintenance can be performed more efficiently. This approach aims to decrease costs for companies and enhance safety for workers and civilians.

Background

There is a notable acceleration in innovation in the Nuclear Energy industry. While U-Net concrete crack detection models do exist, they have predominately been designed for road pavements. However, the U-Net model that was designed for this capstone specializes in wall concretes that are from standing structures, addressing a crucial aspect of nuclear infrastructure maintenance and safety.

Dataset Summary

To train the model, three suitable datasets were utilized, providing 32,640 positive (cracked) and 21,544 negative (non-cracked) concrete images along with segmented masks. These datasets included the Surface Crack Detection dataset from Kaggle [1], Concrete Crack Conglomerate Dataset from Virginia Tech [2], and the CCSS-DATA dataset from Kaggle [3].

Cleaning & Preprocessing

To prepare the dataset for training, all image and mask file paths, heights, widths, and aspect ratios were stored in data frames. It was essential that the images will all be resized to 64x64 and 256x256 to be suitable for U-Net's architecture. At first, all images were suitable for U-Net's architecture, so a decision was made to resize all images to dimensions of 64x64 and 256x256 pixels^2 after analysis.

During the preprocessing stage, it was confirmed that all images have corresponding masks of the same size and that they were all 1:1 aspect ratio. Subsequently, the images and masks were resized to both 64x64 and 256x256 pixels^2 to accommodate for U-Net's model requirements.

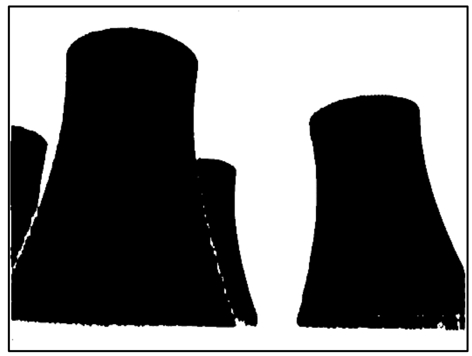
Lastly, it was noticed that there was an imbalanced dataset. Due to the nature of the dataset having more negative pixels (non-cracked) than positive pixels (cracked), it was determined that negatively classified images would not be suitable due to contributing to even more of an imbalanced dataset of positive and negative pixels. As such, the negatives from these datasets have been separated and were not utilized in the training process.

Prior to training, the dataset was split into three sets: 70% for training data, 15% for validation, and 15% for testing. The links to all of these datasets can be found in the References section.

Insights, Modeling, Results

Object Segmentation Model

META's Segmant Anything Model (SAM) API was utilized to set up a solution to segment large structures and patch them into thousands of 256x256 images to feed into the Concrete Crack Detection Model for masking. Below is the result of using SAM on a nuclear plant structure:



Concrete Crack Detection Model

U-Net, a popular Convolutional Neural Network (CNN) for pixel segmentation, was utilized for this problem. A manual iterative approach was taken to tune the model for optimum validation and training loss. Training began with 64x64 images but due to unsatisfactory results, the transition was made to 256x256 images. Furthermore, the imbalanced ratio was addressed through weighted binary crossboundary coefficients for the loss function. Below are the parameters tested and the optimum configuration found through 19 cycles of tuning.

	Tested	Optimum Configuration
Image Size	64x64, 256x256	256x256
Model Layers	5, 6, 7	6
Model Kernels	3x3, 5x5	3x3
Kernel Initializer	He Normal, He Uniform	He Normal
Activation Function: Leaky ReLu Coefficients	0.1 to 2.0	Leaky ReLu Coefficient 0.375
Optimizer	Adam, RMSprop, SGD with momentum 0.9	Adam
Loss Function: Weighted Binary Crossboundary Coefficients	0.5 to 1.5	Weighted Cross Boundary (0.9 negative pixels,1.1 positive pixels)

And as shown below, through this tuning, the losses were halved and each score increased by more than 10%. An image of the scores per iteration cycle can be found in Appendix A.

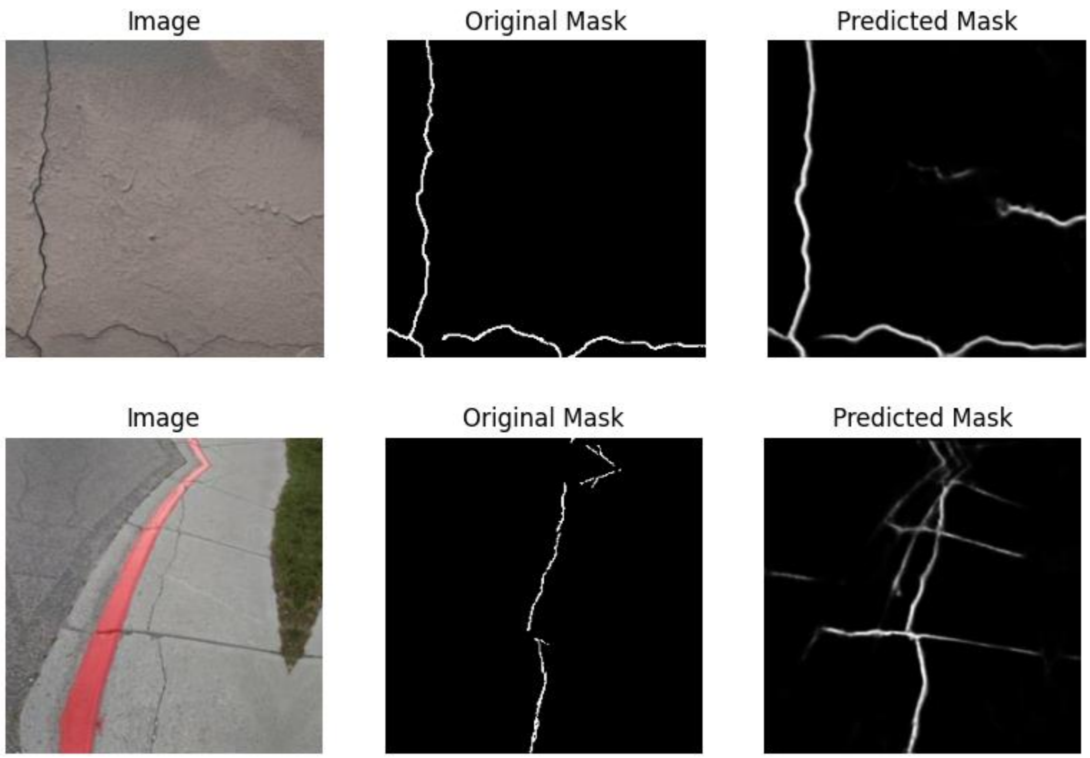
Run #	Training Loss	Validation Loss	Validation F1 Score	Validation IoU Score	Validation Recall
1	0.065	0.07	0.771	0.509	0.725
19	0.03	0.037	0.869	0.705	0.866

After this extensive manual tuning process, the model was tested on the remaining 15% of unseen test data and achieved the following scores and generated masks:

Loss	F1 Score	IoU Score	Recall Score	Precision Score
0.0377	0.8651	0.6930	0.8639	0.8771

Due to recall being the most important scoring metric for the model's performance since not identifying cracks can be costly, the model's final performance scores are satisfactory with recall being almost over 90%.

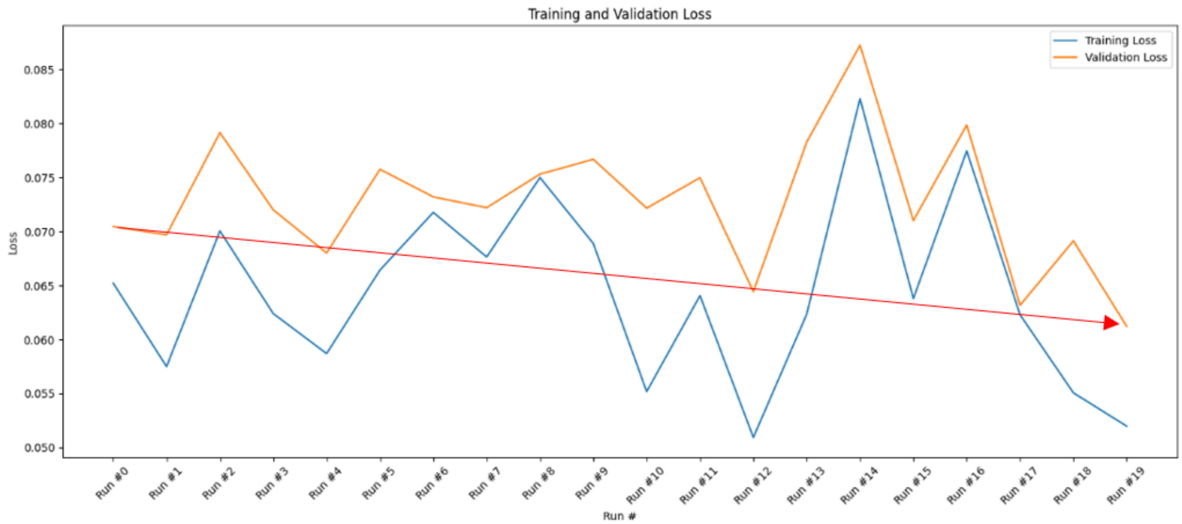
Furthermore, upon investigation, the model did exceptionally well with cracked images that were clean, but poorly when there were pavement tiles. These pavement tiles were mistaken for cracks. While this model can be used for pavement, it is primarily intended to be used on standing concrete structures. So for its application, it is expected to do even better. See below for examples. In the first row, it’s evident that the model detects cracks better than the original mask, but in the second poorer.

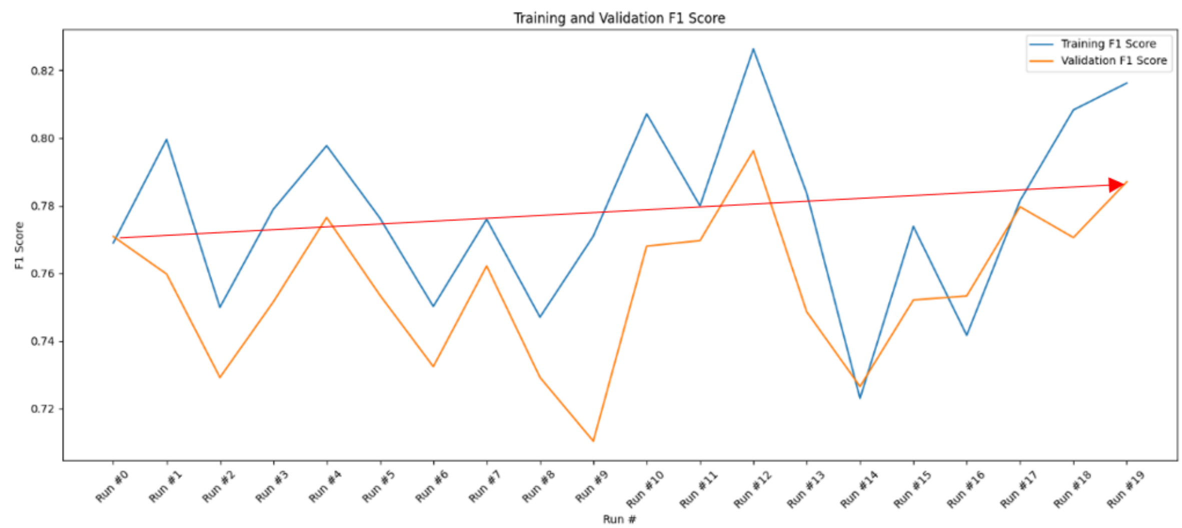


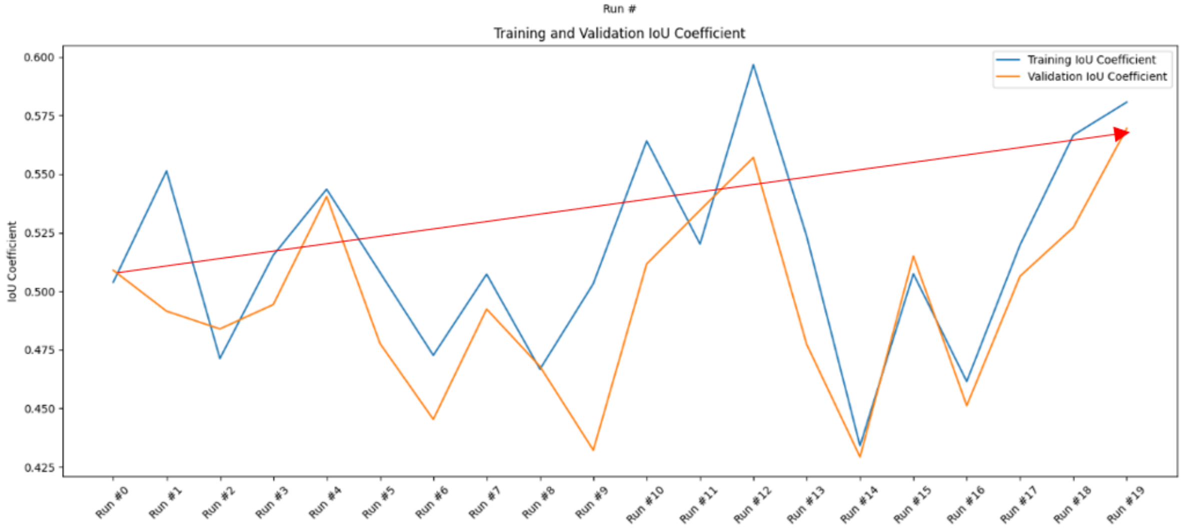
FINDINGS & CONCLUSION

In conclusion, the integration of META's Segment Anything Model (SAM) for structure segmentation and the U-Net concrete crack detection model has yielded promising results for future implementation. While there are ongoing efforts to finish the code for separating and stitching the segmented objects into thousands of 256x256 pictures and back, the current implementation has showcased remarkable performance. As of now, the focus of the model has been on detecting cracks in structural walls. However, future iterations will include a more diverse set of pavement images during training, aiming to enhance the model's performance for road applications. Additionally, to further refine the model, the validation dataset will be merged with the training data during training of the model, further increasing its performance.

APPENDIX A: TUNING CYCLE PERFORMANCES







Appendix B: References

[1] "Surface Crack Detection," www.kaggle.com. https://www.kaggle.com/datasets/arunrk7/surface-crack-detection

[2] A. ISLAM, "Concrete Surface Image Processed with Match Filter," www.kaggle.com. https://www.kaggle.com/datasets/ahsanulislam/concrete-surface-image-filtered-with-match-filter (accessed May 15, 2023).

[3] E. Bianchi and M. Hebdon, "Concrete Crack Conglomerate Dataset," data.lib.vt.edu, Oct. 2021, doi: https://doi.org/10.7294/16625056.v1.

[4] P. Shokhri, "CCSS-DATA," www.kaggle.com. https://www.kaggle.com/datasets/parniashokri/ccssdata (accessed May 15, 2023).