All That It's Cracked up to Be:
an Infrastructure Crack Detector

Scott Wigle

/github.com/Swig-DPI
in/scott-wigle/
scott.mwigle@gmail.com

Background

Currently engineers have to physically climb structures such as dams, bridges, and various other buildings, on a regular basis and following an event to inspect the facility. This process is labor intensive and dangerous especially following an earthquake or during a flood. One aspect of these inspections involves observations of cracks in concrete which may be an indicator of ongoing structural failure or performance issues.

This project aims to detect surface cracks through the utilization of machine learning algorithms, focusing on concrete cracks, with the goal of reducing the time spent on structures while simultaneously increasing safety and reducing cost.

Objectives

- Detect the presence of cracks in a photo of a concrete structure.
- Determine incorrect classifications due to model bias.
- Extrapolate information regarding misclassification.

Data

- The data set, SDNET2018, is from the University of Utah and consists of 56,000 images, labeled cracked or not cracked.
- Images depict three types of structures: dams, bridge decks, and walkways.
- Data split into 3 balanced sets: train, test, and hold out. All sets contain 5,600 labeled images.

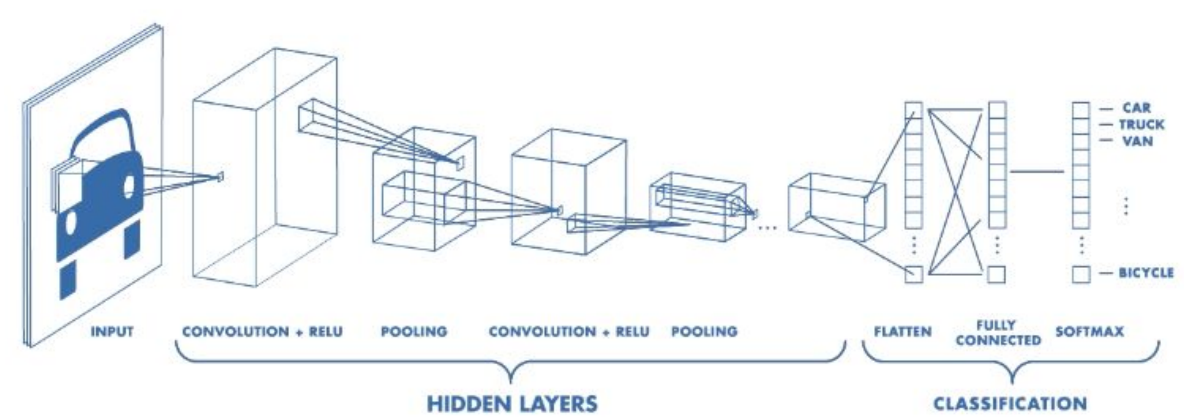
Methods

Baseline Model: Logistic Regression
This method is a binary classification. It uses a combination of features (pixels) to determine the presence of a crack.

$$Log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_n X_n$$

Where p = probability of a crack.

Second Model: Simple Convolutional Neural Network (CNN)



Third Model: Transfer Learning CNN - Takes a pre-trained CNN and re-trains the CNN according to the new data set. Input and output layers are added to fit the needs of the classification problem.

Libraries

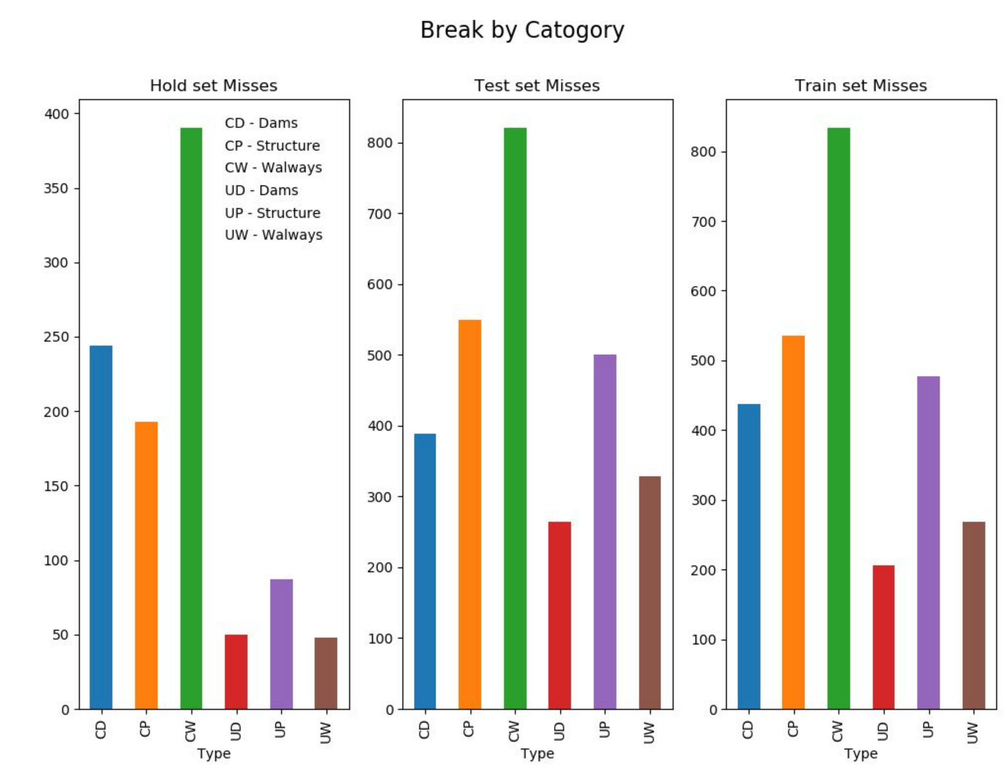
aws, Keras, OpenCV, NumPy, pandas, scikit-learn, Flask, Python, TensorFlow

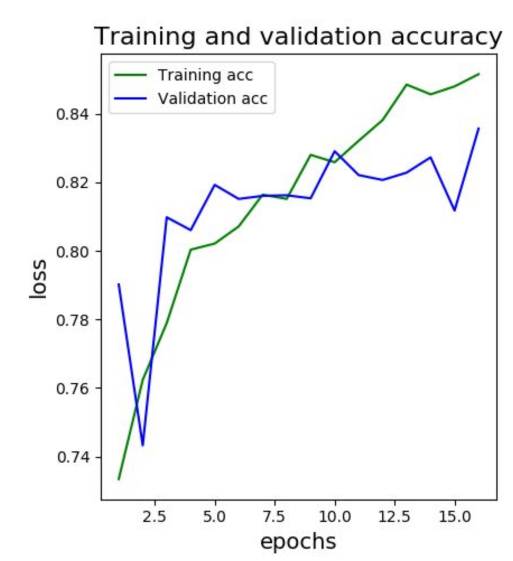
Results

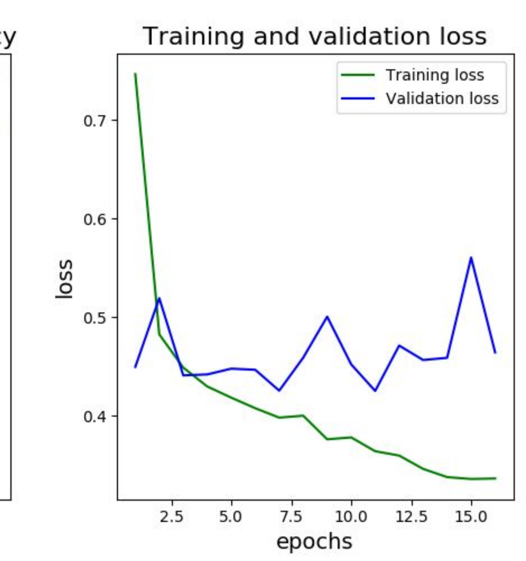
Baseline Model: Logistic Regression
- Best accuracy score ~52%.
- Primarily predicted absence of cracks.

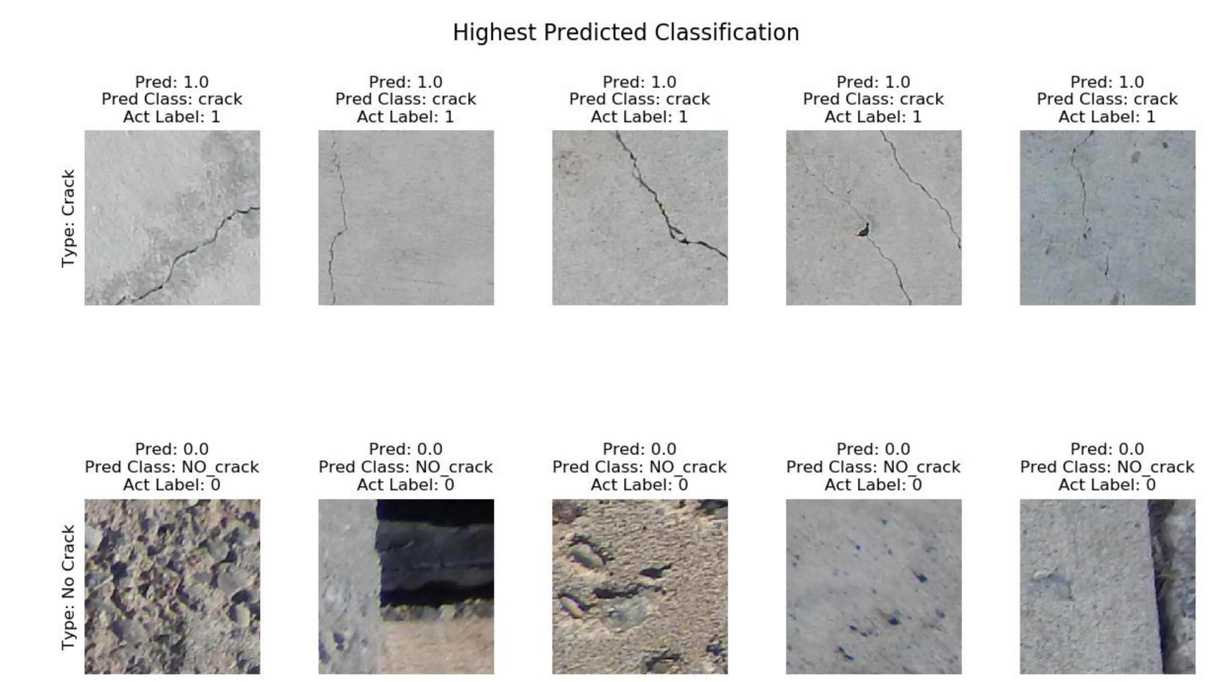
Second Model: Simple CNN
- Best accuracy score ~60%.
- Had issues with different aggregate sizes (i.e. rocks mixed into concrete).

Third Model: Transfer Learning CNN
- Best accuracy score ~82%.
- Missed fine cracks.









Future Work

- Test another transfer learning model: GoogleNet or Inception.
- Build Flask app to take images from users and detect if a crack is present.
- Use model to detect cracks from drone data. This would involve parsing over larger images to highlight cracked sections of a structure.

Special Thanks

- Galvanize Teachers and Staff (Frank Burkholder, Michael Dyer, Kelly Lutz, Alan Teran, Erin Desmond, Angela Hayes, Chris Hegg, Elliot Cohen, Land Belenky, Adam Richards, guests and anyone else I may have missed)
- HDR (Travis Moore, Brad Martin, Raj Prasad, Cheryl Rountree, Jason Page and anyone else I may have missed)
- G80 Cohort
- Editing by Diana Bagdasarova

References

1. Maguire, Marc; Dorafshan, Sattar; and Thomas, Robert J., "SDNET2018: A concrete crack image dataset for machine learning applications" (2018). *Browse all Datasets.* Paper 48. https://digitalcommons.usu.edu/all_datasets/48
2. Chollet, Francois. "The Keras Blog." *The Keras Blog*, June 2016, blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html.
3. Patel, Shyamal, and Johanna Pingel. "Introduction to Deep Learning: What Are Convolutional Neural Networks? Video." *Reconstructing an Image from Projection Data - MATLAB & Simulink Example*, 24 Mar. 2017, www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html.