EE/CprE/SE 491 WEEKLY REPORT 01

9/1/2019 - 9/14/2019

Group number: 20-18

Project title: Development of Image Analysis Algorithms for Crack Detection Using a Smartphone

Client &/Advisor: Bo Yang/Halil Ceylan

Team Members/Role: Akira Demoss, Maggie Dalton, Modeste Kenne, Nik Thota

---

- **Weekly Summary**

These first few weeks have primarily been spent discussing the project, beginning research into applicable areas of machine learning, and setting up environments to begin work on prototyping a potential solution. We have been largely successful with that goal and have identified a possible plan that may allow us to identify cracks with relatively quick speed and classify them with accuracy while maintaining flexibility in the design of the application. Following conception of the plan, we have been researching technologies to build each of the necessary components.

- **Past week accomplishments**
    - **Maggie Dalton**
        - Researched object detection algorithms
            - Learned a bit about the history/development of popular algorithms
            - Learned about the distinctions between HOG/CNN/YOLO
                - YOLO looks like a good option for object detection (and some level of classification)
                - YOLO is more lightweight than some of its counterparts and has a few options to pick from if we want faster/less accurate or slower/more accurate
        - Installed Tensorflow on personal computer

    - Researched the general process of creating and training a neural network
    - Researched AWS options for running tasks (like training)
        - EC2 might be an option for faster training than we can accomplish on our personal computers
- **Modeste Kenne**
    - Researched how to make a good data set
        - Identified types of cracks to detect: Alligator Cracking, Block, Longitudinal, Transverse, Slippage cracking, etc…
        - Learned about about data selection, data processing and data transformation.
        - Literature review on pavement a crack detection paper: “Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection”
        - Read about “SDNET2018”, a dataset for training artificial intelligence based crack detection algorithm of concrete surfaces.
- **Akira DeMoss**
    - Did key testing on technologies to establish a standardized technology stack / development environment
        - Got a training environment setup and configured with CUDA and cuDNN module.
        - Tested multiple machine learning frameworks (namely keras and tensorflow) also installed tensorflow backend. Identified Darknet backend as the best way to get started training models for object detection.
        - Trained a custom model, identified a sample dataset to test the framework on, and practiced labeling images in labelIng.
        - Configured custom configuration of OpenCV to build with CUDA and cuDNN modules among other custom modules. Saved the custom compile flag commands for reproducing the build on other machines - **side note:** hopefully this will save us all time as this was quite time consuming even with my year of experience working with and building open source C++ software… however definitely worth the effort!
        - Visualized results of my model using OpenCV 4 which drew the bounding boxes around the object detection predictions.
    - Learned how to label images for training a dataset in a convolutional neural network

        - Researched how to integrate model for detection into Android - identified Tensorflow Lite as a solid solution for this.
    - Nik Thota
        - Downloaded Ubuntu Linux to obtain Yolo software.
            - Obtained extra computer from the university to run on.
        - Read through Yolo documentation and viewed sample videos.
        - Looked are previously implemented crack detection algorithms to see what has been done in the past and how we may do things differently.
            - Looked at several projects on github.
        - Read LTPP Distress Manual.
            - Looked at crack and pothole classifications.

- **Pending issues**

None so far

- **Individual contributions**

Name	***Individual Contributions***	***Hours this week***	***Hours Cumulative***
Akira Demoss		24	24
Maggie Dalton	Researched object detection algorithms, installed Tensorflow	6	6
Modeste Kenne	Researched how to make a good data set	6	6
Nik Thota	Researched how to use Yolo and looked at previously existing crack detection algorithms.	6	6

- **Comments and extended discussion**

N/A

- **Plans for the upcoming week**

- **Maggie Dalton**
    - Experiment with implementing/training a YOLO algorithm to better understand pros/cons
        - If time, maybe do same for versions of CNN/R-CNN
    - Complete AWS learning module for machine learning to see if their suggestions for AWS services could be useful for project
        - If so, would it be possible to remain in the free tier?

- **Modeste Kenne**
    - Set up working environment
        - Identify machine that would be used for client/server communication
        - Install Tensorflow and any other require software.
        - Configure client/server
    - Test client/server communication and make sure it is up and running efficiently,

- **Akira DeMoss**
    - Priority
        - Help team members to get familiar with complex technology stack / development environment.
        - Read more about deep learning to develop better intuition for meeting future requirements
    - If time permits
        - De-couple previous Android Camera Application and add to project repo
- Nik Thota
    - Set up training environment
        - Install Tensorflow
    - Design screens
        - Draw screen diagrams and screen flows
    - Familiarize myself with Tensorflow
        - Look at documentation online and watch some informative videos

- **Summary of weekly advisor meeting** (*If applicable/optional)*

  Met with our client, Bo Yang, to discuss what kind of deliverables he's looking for at

the end of the project. Decided upon creating an Android application first (because everyone in our group has Android phones), then creating a modified iOS version. We also discussed potential difficulties (namely driving speed impacting the integrity of collected images/videos) and where we can find information on crack classification standards and existing datasets.