

EE/CprE/SE 491 WEEKLY REPORT 02

9/15/2019 – 9/30/2019

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**

This week we continued our research. We built upon the base knowledge of machine learning we gained last week and read about different libraries/technologies/etc. that will allow us to translate that into a mobile application. This consisted largely of identifying past projects that were similar, learning how to set up TensorFlow, and reading about TensorFlowLite for mobile.

○ **Past week accomplishments**

- Akira
 - Learned about the file inputs and outputs for Darknet and Tensorflow
 - Converted model and weights generated via Darknet to Tensorflow format
 - Read 6 chapters of the book Deep Learning for Computer Vision by Adrian Rosebrock
 - Identified Tensorflowlite as the best deep learning framework for mobile deep learning applications by reading the academic paper AI Benchmark: Running Deep Neural Networks
 - Read the articles “Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Image”, “A Deep Learning Approach for Road Damage Detection from Smartphone Images”, “Road Damage Detection and Classification with Faster R-CNN”, and “A Conditional Wasserstein Generative Adversarial Network for Pixel-level Crack

Detection using Video Extracted Images”. Two of these articles were published in the *2018 IEEE International Conference on Big Data*

- Encouraged and facilitated the use of a standardized development. Identified resources that would allow our team to effectively collaborate.
- Maggie
 - Researched TensorFlow Lite
 - Is optimized for Android but also works for iOS
 - Can help access hardware specific speed boosts
 - On Android it allows us to use Java instead of C++ with TensorFlow functions
 - Got an example from Google working on a personal phone and have been working through the companion codelab
 - Now understand where/how to input the trained model into an application
 - Created/pushed base Android application to group GitLab
 - After reading about different Android API/Versions decided upon API 27+ due to the addition of TensorFlow Lite compatibility
 - Learned how to label datasets
 - Created a labeled dataset
 - Examined past work on similar projects
 - Got a similar application from a group in Japan working on a personal phone that also detects road cracks, it is very slow
 - Been looking through how they implemented their application. They used MobileNet (a predecessor to TensorFlow Lite)
 - Looked at how they labeled their dataset
- Nik
 - Researched existing projects of the same goal
 - Stitched together first lightning talk
 - Clipped audio files from each member and attached them to the appropriate slides
 - Researched what software may be useful to us looking ahead
- Modeste
 - Learned how to implement YOLO algorithm for object detection in images
 - Lookup existing detection methods and weigh the advantages of using one over the other.
 - Read about TensorFlow platform
- **Pending issues**
 - Maggie

- Been working with ETG on getting a VM set up for hosting the server/database/classification algorithm for the project. Because the communications between the phone and the server will happen while the phone is off campus we would like the VM to be open to communications outside the ISU firewall. This has required some additional information/communications with ETG that are in the process of being resolved.
- The TensorFlowLite requires the model to be in format “.tflite”. Haven’t been able to successfully convert a TensorFlow model to “.tflite” yet.
- Akira
 - Issues converting tensorflow module to tensorflow lite module, need to identify a way to freeze the graph. Tried several commands and custom scripts, but found people had previously solved the problem by using a version of tensorflow compiled from source as opposed to the precompiled binaries. Planning to try again after building tensorflow from sources.

○ **Individual contributions**

Name	Individual Contributions	Hours this week	Hours Cumulative
<i>Akira Demoss</i>	Overcame challenges in combining several tensorflow files into a single file also referred to as freezing a graph, documented steps for this process, identified resources for deep learning, helped a team member install 3rd parCuda	20	44
<i>Maggie Dalton</i>	Research, got examples working on personal device, created base for Android portion of project	12	18
<i>Modeste Kenne</i>	Literature review on client/server communication. Refresher on Android Studio projects.	8	14
<i>Nik Thota</i>		10	16

- **Comments and extended discussion**
 - We have been struggling with meeting as a group due to how different our schedules are and changes in our personal schedules (work, family, etc.). We are working to find a consistent weekly time and space where we can count on meeting.

- **Plans for the upcoming week**
 - Maggie
 - Create a working prototype of an Android application running an algorithm that will be input into the application
 - Use the created dataset to retrain a YOLO model
 - Convert that model to “.tflite” and input into the application
 - Add functionality for the app to record the GPS location when something has been detected
 - Work with Akira to find a suitable dataset of cracked road images for training
 - Repeat the process of labeling, training, and converting it to “.tflite”
 - Swap out other model for this one, keeping other functionalities of the application
 - Akira
 - Continuing to help members get the software installed
 - Creating and updated documentation as needed during this process
 - Goal is to have more face to face meetings so I can help everyone get the software installed and we can all work on training and parameter tweaking.
 - Nik
 - Install appropriate software using the guidelines set forth by Akira
 - Create project timeline
 - Edit lightning talk
 - Modeste
 - Install OpenCV and Cuda
 - Look up Volley tutorial
 - Create activity in Android Studio to send a test image to server

- **Summary of weekly advisor meeting:** No meetings during this period due to

scheduling conflicts