Development Plan

Drake Svoboda
Hassan Bazzi
Hadil Mohamed
Safayeth Khan

1.1 Project Overview

Before road damage can be repaired, government officials must be aware of the presence of road damage. The process of locating and identifying road damage requires manual inspection, expensive sensors, specialty vehicles, or public reports. Many of these methods are either cost prohibitive, slow, or ineffective. We will create a mobile application that will detect the presence of road damage in pictures captured by dashboard mounted cell phones. Cell phones running our application could be dashboard mounted in government vehicles to automatically detect and record road damage locations. We will also create an accompanying web application to make this information accessible to government officials.

1.2 Project purpose, scope, objectives

The purpose of our project is to assist governments with the identification of road damage in a more optimized and automated manner. To do this, we will need two components: 1) A mobile application that will take the pictures and detect road damage; and 2) an interface for government officials to retrieve road damage data. Using this data, government officials will be able to take timely and appropriate action to improve road conditions.

To achieve these objectives, we will create a machine learning model that will detect the presence of road damage in pictures captured by dashboard mounted cell phones. The model will be able to detect and classify road damage into 8 categories (crack, pothole, faded crosswalk lines, etc.). If road damage is detected, a data entry will be stored containing the location and type of damage detected. The data will be accessible via an easy-to-use web application. This data will also be accessible through a web API for governments to integrate into existing systems.

1.3 Team organization (roles and responsibilities)

Each member will be clearly assigned aspects of the project that he/she is responsible for. These roles are not rigid; all members may be expected to work on each aspect of the project.

It important to note that each member will be completely responsible for his/her own documentation and all team members will be expected to contribute to submitted assignments and presentations. In this regard, the presentation and documentation leads will not be solely responsible for presenting and documenting and will only oversee that each is done properly and professionally.

Drake Svoboda

Team Lead

Drake was chosen to be team lead because he initiated this project and has the most knowledge of it. As team lead, Drake is responsible for overseeing the project progress and delegating work to the rest of the team.

Computer-Vision Model/Machine Learning Lead

Drake will be working on training/creating the computer-vision model that will be used to detect and identify road damage.

Hassan Bazzi

Web Application Lead

Hassan is responsible for leading the development of the web application that will display the road damage locations. Hassan will also engineer the API for communication with the mobile app and potential third parties.

Database Lead

With the database being part of the web application, Hassan will set up and design the database that will be used to store damage location data.

Hadil Mohamed

Mobile App Co-Lead

Hadil will be (co)-leading the design and development of the mobile application, which will run the computer-vision model and communicate with the web server.

Presentation Lead

Hadil will be responsible for designing and formatting presentations. She will also be responsible for reviewing the presentations for any misspelled words or grammatical errors and ensure that a professional and polished look is attained.

Safeyeth Khan

Mobile App Co-Lead

Safetyeth will be (co)-leading the design and development of the mobile application, which will run the computer-vision model and communicate with the web server.

Documentation Lead

Safeyeth is responsible for managing the documentation to guarantee its overall correctness, and he will ensure that all documentation is professionally designed and properly formatted.

1.4 Problem resolution policies

Every team member is expected to be an active participant and contributor to this project, but policies must be established to resolve potential problems if/when these expectations are not met at any point in the semester. The team has elected to employ the standard problem resolution policies, which take effect in 3 levels – the team lead, TA, and instructor(s). This 3-level system will be explained in greater detail below along with the expectations of all team members.

Team Expectations

All team members are expected to show up to each scheduled meeting, as agreed upon by the team. If a team member is unable to attend a meeting, he/she is required to notify the team and the TA by sending out an email at least 24 hours in advance to allow enough time for the team to decide whether to adjust the meeting agenda or postpone the meeting until a later date. In cases of extenuating circumstances, such as emergencies, accidents, or significant prior-engagements, members will be excused for missing a meeting.

All team members are expected to deliver their work at the scheduled times, as agreed upon by the team, and are expected to produce high-quality work. If a team member does not expect to meet a deadline, it is his/her responsibility to ask the team or the TA for extra help several days in advance to ensure there is adequate time to complete the work and meet the deadline.

All team members are expected to communicate with one another in a respectful and professional manner. The opinions of all team members should be respected and taken into consideration. If a team member recommends a decision or course of action, he/she must make a rational justification for its use and describe its benefits and drawbacks. The team must then discuss the decision or course of action and decide whether it should be implemented.

Resolving Conflicts

If a team member does not meet all the expectations previously stated at any point, the three-level system will take effect. If a member does not meet the expectations in some way (e.g., misses a meeting or is late on a deadline) once, then level 1 of the system gets implemented. At this level, the team lead will confront the team member(s) about the conflict and the team members will work together to prevent it from reoccurring. If the conflict reoccurs more than twice or persists, level 2 is implemented. In this case, the team lead will notify the TA about the conflict and they will work together to come up with a better solution, with the TA having the final say. If the conflict is still not resolved after the TA intervenes, the third and final level is implemented. In this level, the team lead and the TA will meet with one or both instructors to discuss the matter and the instructor(s) will have the final say as to how to resolve the conflict.

1.5 Project plan (iterations, project schedule)

Team Meeting Schedule

The team's main forms of communication will be through Slack, GitHub and text messaging. All team members and the TA will meet weekly on Mondays after class. Team members can coordinate additional meetings if required.

Project Schedule

Web App

Deliverable	Description
Prototype 1	**09/15 - 10/01**
1.1 Basic models and controllers	Implement basic model and controller for the road damage entries with create/read/update/delete functions available via an HTTP-based RESTful API with no authentication.

1.2 Authentication	Implement users and user roles/permissions. Implement JavaScript Web Tokens for API authentication.
1.3 Basic Front End	Implement a basic React.js frontend with login/registration pages and a google maps integration.
Prototype 2	**10/01 - 10/29**
2.1 Verify and Label Damage Instances	Users can verify road damage and label instances as ignored, to be fixed, under construction, completed.
2.2 Search and Sort	Users can search and sort the data both in the API and in the frontend.
2.3 Front End Styling	Enhance the styling of the frontend.
Prototype 3	**10/29 - 11/19**
3.1 Integration with Continuous Training	Implement API endpoint so verified images can be gathered for continuous training.

Mobile App

Deliverable	**Description**
Prototype 1	**09/15 - 10/01**
1.1 Recording Images	Implement simple mobile app that uses the phones camera to be tested in a moving vehicle.
1.2 Communication with Web App	Images and GPS location data is communicated with the web server.
1.3 Log In / Register Features	Users can login and register through the mobile app.
Prototype 2	**10/01 - 10/29**
2.1 Running Neural Net	Deploy machine learning model to run natively.
2.2 Data View Front End	Road damage instances are displayed in the front end.
Prototype 3	**10/29 - 11/19**
3.1 Offline Mode	Data is gathered and stored while the mobile device is offline. Once an internet connection is established, gathered data is sent to the web server.
3.2 Improved Location Reporting	Travel direction and which road the mobile device is on is reported along with raw GPS coordinates.

Deep Learning Model

Deliverable	**Description**
Prototype 1	**09/15 - 10/01**
1.1 First Iteration of Model	Train a multi-label classification model.
Prototype 2	**10/01 - 10/29**
2.1 Deploy Model in Mobile App	Deploy model to run natively on a mobile device.
2.2 Improve Performance	Continue to improve model performance. Prototype bounding box object detection.
Prototype 3	**10/29 - 11/19**
3.1 Train on Verified Images	Continuously and automatically train on road damage instances marked verified by web users.

1.6 Configuration Management Plan

Source code will be managed through GitHub. Each team member has been added as a contributor to the deepditch GitHub organization. A thorough git workflow strategy document can be found in our documentation repository. This document covers our strategy for branching and merging code and contains the relevant git console commands.

All features and bug-fixes will be developed in separate branches. Once a change has been completed in a separate branch, a pull request can be made through GitHub. All merge conflicts should be resolved and tested within the separate branch; this will allow for clean merges back into master. Ideally, one developer will be working in a branch at a time, however, this is not a strict requirement.

1.7 Technologies

Machine Learning Model

The machine learning model will be developed in Python using the deep learning library PyTorch. The reason we have chosen PyTorch over TensorFlow is due to PyTorch's object-oriented API. PyTorch's API makes it easy for any python developer to quickly get up to speed whereas TensorFlow's graphs and sessions add a level of indirection.

Web App

PHP utilizing the Laravel framework. PHP was chosen as it allows for rapid development of web applications, especially with the usage of a framework such as Laravel. Laravel makes it easy to implement models, views, and controllers. It has a built in ORM (Object Relation Mapper) which will help us map our objects to tables in a straightforward manner, without us having to write raw SQL queries. Other advantages are that it makes authentication, sessions, and unit-testing a breeze. Our web frontend will utilize React.js, a popular and powerful frontend JavaScript framework.

Database

We will be using MySQL as our database driver. However, all interaction with the database from mobile and/or the dedicated server will be through a web service API as best-practice.

Mobile App

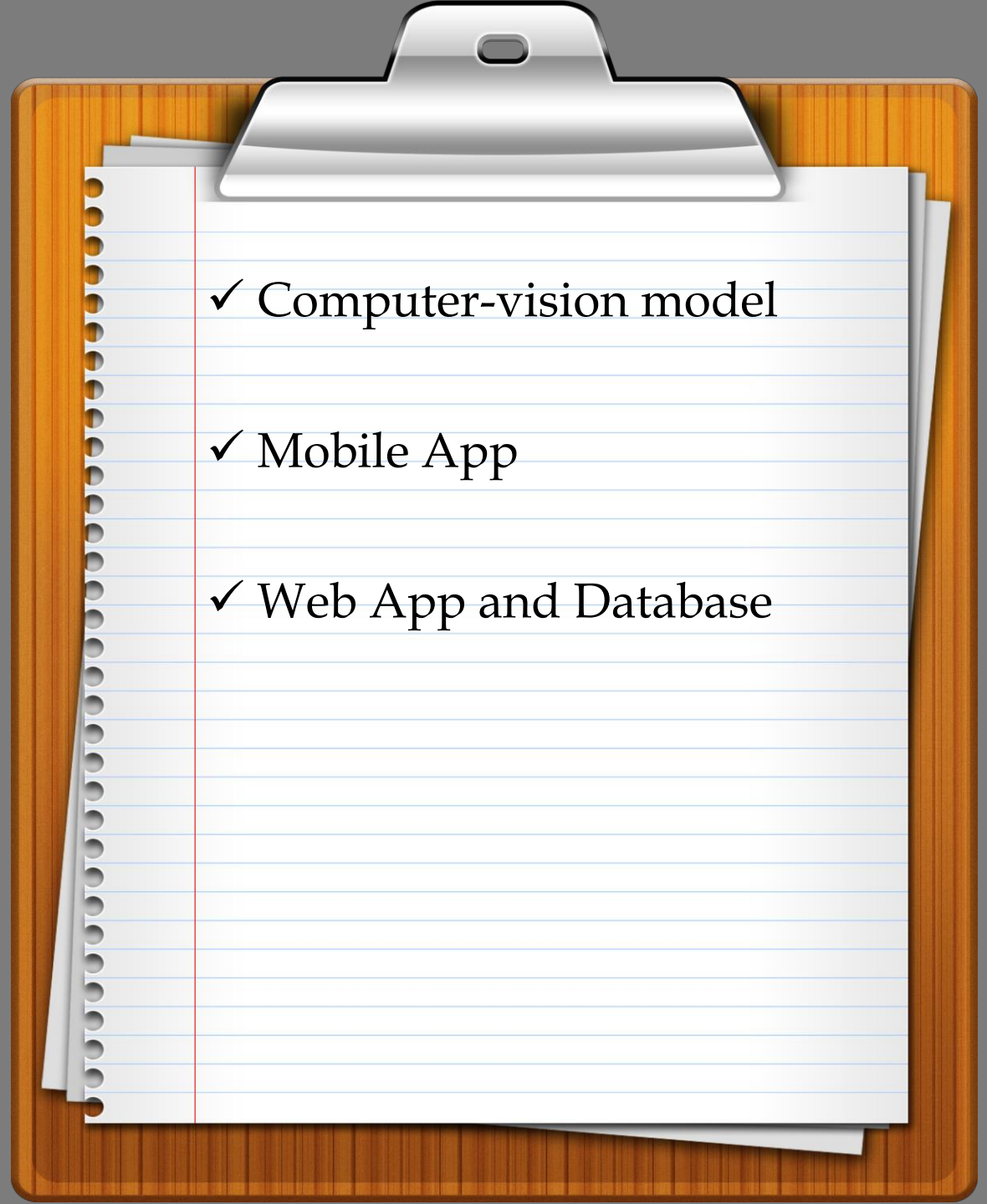
We will use React Native to develop our mobile application. React native was chosen due to its expanding market share and its ability to 'hot-reload' JS modules. We expect to benefit from an active development community and quickly prototype without waiting for costly build times. We will focus our efforts on iOS first; we may also develop for Android as a stretch goal.

Road Damage Detector

Drake Svoboda, Hassan Bazzi,
Hadil Mohamed, and Safayeth Khan

What is the purpose?

Objectives



Team Organization

Drake Svoboda: Team Lead, Computer-Vision Model Lead

Hassan Bazzi: Web App Lead, Database Lead

Hadil Mohamed: Mobile App Co-Lead, Presentation Lead

Safayeth Khan: Mobile App Co-Lead, Documentation Lead

Project Schedule

DELIVERABLE	START	END
Mobile App		
Prototype 1	9/15/18	10/1/18
1.1 Recording Images		
1.2 Communication with Web App		
1.3 Log In / Register Features		
Prototype 2	10/1/18	10/29/18
2.1 Running Neural Net		
2.2 Data View Front End		
Prototype 3	10/29/18	11/19/18
3.1 Offline Mode		
3.2 Improved Location Reporting		

DELIVERABLE	START	END
Web App		
Prototype 1	9/15/18	10/1/18
1.1 Basic Models and Controllers		
1.2 Authentication		
1.3 Basic Front End		
Prototype 2	10/1/18	10/29/18
2.1 Verify and Label Damage Instances		
2.2 Search and Sort		
2.3 Front End Styling		
Prototype 3	10/29/18	11/19/18
3.1 Integration with Continuous Training		

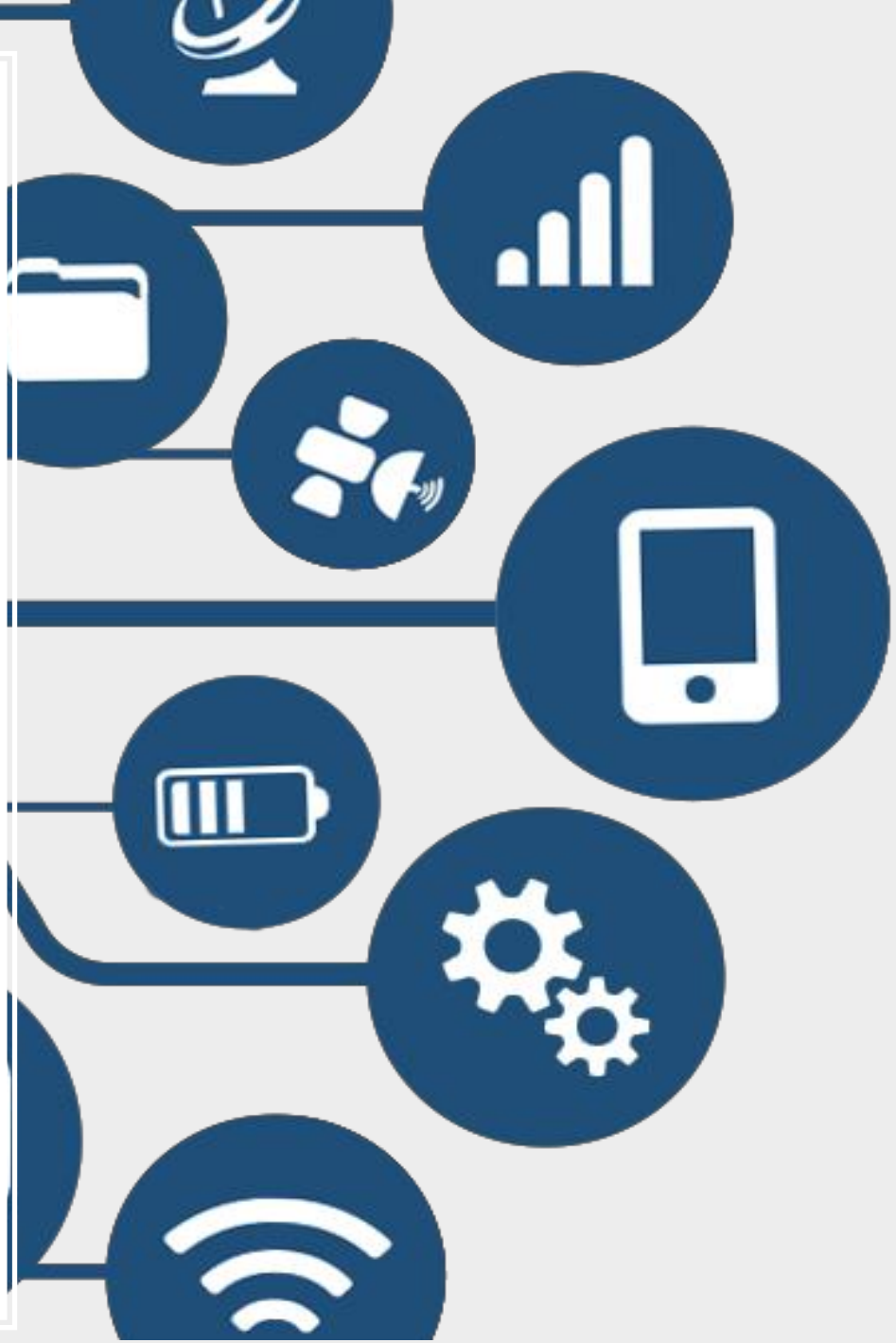
DELIVERABLE	START	END
Deep Learning Model		
Prototype 1	9/15/18	10/1/18
1.1 First Iteration of Model		
Prototype 2	10/1/18	10/29/18
2.1 Deploy Model in Mobile App		
2.2 Improve Performance		
Prototype 3	10/29/18	11/19/18
3.1 Train on Verified Images		

Technologies

Python & PyTorch

PHP, Laravel & MySQL

React Native



Configuration Management Plan

GitHub

Team Expectations

Conflict Resolutions

Team Meeting Schedule

Questions?