***OBD-II Reader Image Detection And Text Extraction***

Requirements Document

# 1. Introduction

## 1.1 Overview

The goal of this project is to build a solution to classify OBDII Reader images from in a set of images and also to extract text from those images captured by ACV's car Inspector. This text will help ACV's team to prepare condition reports of vehicle.The extracted text contain OBDII codes which are used for vehicle diagnostics.

## 1.2 Scope of the Product

The product accepts the images from the vehicle inspectors as input through. The classification web service is called and the classification is performed and the classified OBD-II images will be separately stored. Further the image of the OBD-II reader is given as an input to the text extraction API is called. This image is feeded to the trained text extraction model and text information is extracted. A JSON object will be constructed using the extracted text information and this JSON object will be sent as response to the API caller.

## 1.3 Business Case for the Product

The current approach used by the company is to capture the OBD II reader information and directly display it to the customers. This approach does not help the company to obtain more information about the vehicle and does not aid in the price prediction. The product will automate this process so that the obtained information will be used by the company to determine the accurate price of the vehicle. The information will be delivered to the company which can be added in the condition report given to the customers.

# 2. General Description

## 2.1 Product Perspective

The product accelerates the process of reporting the condition of the vehicle by getting the error code information of the vehicle displayed in the OBD II reader. The tool will be a part of the business model of the company and will be an integral part of their vehicle information gathering process and price prediction process.

## 2.2 Product Functions

1. Classifying OBD2 reader images from all vehicle image dataset.
2. Upload images to the web API.
3. Refactoring the image into smaller resolution and applying image processing techniques.
4. Extracting the text from obd2 reader images and provide user control to correct the text manually.
5. Display the extracted text with related information in a user-friendly format.

## 2.3 User Characteristics

The product will be used by the vehicle inspectors. The user of the product require no technical knowledge for using our product. The inspector needs to verify if the extracted text is correct or requires any alteration. Moreover, if the input image is of poor quality, the inspector might require to fill out the information manually as instructed to do so by our product.

## 2.4 General Constraints

Usage of any third party APIs has to be verified with ACV auction to ensure license and any such implementation constraints. The product would perform better if the input images have minimum resolution specified by the system with no skew.

## 2.5 Assumptions and Dependencies

The project assumes that the image database used to train our ML model will be large. It is also assumed that the input images are high resolution for image processing tasks.

# 3. Specific Requirements

## 3.1 User Requirements

1. Upload the images of the vehicle.
2. Classify the images of OBD-II reader and other images.
3. Display classified images.
4. Extraction of the text in the OBD-II reader image.
5. Display extracted text from the OBD-II reader image.
6. Feedback from the vehicle inspector for the tasks performed.

## 3.2 System Requirements

1. The system requires large dataset for improving the accuracy of ML classification model.
2. The webpage requires an authentication mechanism to for secure access to the API’s.
3. The product uses a temporary database to store the results of classification and extraction before verifying it through the feedback.
4. System logs the classification and extraction events that are wrong which are identified during the feedback and adds the correct data to the training set.

## 3.3 Interface Requirements

This application would require a simple interface with a provision to load an image. The image would be feeded to the trained text extraction model through an API. The extracted text is in the form of JSON object which is further verified by the Vehicle Inspector before it is stored in the database.

# 4. Appendices

1. **On-board diagnostics** (**OBD**) is an automotive term referring to a vehicle's self-diagnostic and reporting capability. OBD systems give the vehicle owner or repair technician access to the status of the various vehicle subsystems. The amount of diagnostic information available via OBD has varied widely since its introduction in the early 1980s versions of on-board vehicle computers. Early versions of OBD would simply illuminate a malfunction indicator light or "idiot light" if a problem was detected but would not provide any information as to the nature of the problem. Modern OBD implementations use a standardized digital communications port to provide real-time data in addition to a standardized series of diagnostic trouble codes, or DTCs, which allow one to rapidly identify and remedy malfunctions within the vehicle.
2. **Image classification** refers to the task of extracting information classes from a multiband raster image. The resulting raster from image classification can be used to create thematic maps. Depending on the interaction between the analyst and the computer during classification, there are two types of classification: supervised and unsupervised.
3. **ACV Auctions** provides a mobile app and web platform that allows car dealers to quickly buy and sell wholesale vehicles through live auctions from the comfort of being at their own location.

# 5. Glossary

* OBD-II - On Board Diagnostics 2
* Auction ID’s - Unique identity through which different components of the vehicle are associated to that vehicle by the company.

# 6. References

1. https://support.pix4d.com/hc/en-us/articles/115002495966-Computer-requirements
2. https://en.wikipedia.org/wiki/On-board\_diagnostics
3. http://desktop.arcgis.com/en/arcmap/latest/extensions/spatial-analyst/image-classification/what-is-image-classification-.htm
4. https://www.acvauctions.com/