CS Capstone Requirements Document

November 3, 2017

CDK Data Stream AI

Prepared for

CDK Global

Chris Smith

Prepared by

Group 65

SigFind

Jacob Geddings

Inhyuk Lee

Juan Mugica

Abstract

Our team has been assigned to assist in the development of AI for application to CDKs existing Data Streams. These streams deal mainly with the classification of documents as well as pictures and information contained within them. This document outlines the various requirements our team has been tasked with fulfilling, as well constraints, time frames, platforms and strategies utilized for their completion. These requirements range from basic functionality pertaining to signature detection, to stretch goals such as license plate reading, car damage detection, and permit and license classification.

CONTENTS

1 INTRODUCTION

Overview of SRS

1.1 Purpose

The purpose of this requirements document is to bring clarity to the project by indicating what the hard-line requirements are for the project to be considered a success. This will be done with the consideration that the desired end product may not be achievable and as a result will be treated as a form of research project or proof of concept.

Intended target audience for this document is our client and by extension CDK given this is their idea and the success of the project directly assists them. In addition, our instructors and teaching assistant are extensively involved in this project and they will be using this as a grading guideline. The assumption is that the reader is familiar with general concepts within computer science including artificial intelligence, common programming languages, and coding environments.

1.2 Scope

The product being developed will be a variant of open source AI that will be tailored to scanning submitted forms by CDK. With the inclusion of stretch goals this platform may also analyze driver licenses, vehicle imaging, or name to vehicle identification.

This project will be capable of analyzing a submitted form to locate its signature box and check to see if it has been signed. It will also prompt the operator if it is not certain of what it is observing and if a cosigner box is also present. The format of these submissions will be in PDF format. Stretch goals include the capacity to detect expiration dates on drivers licenses and determine what make and model of a vehicle is based on a submitted image. Additional file formats are also being considered.

Benefits from this product include reduced costs for CDK in error checking forms, rapid validation of a document, and a singular platform that can scan the various form styles that CDK possesses. Another goal of this design is the ability for it to expand upon itself and adapt to new forms that may be introduced by permitting operator confirmation or override on its assessment.

1.3 Definitions, Acronyms, and Abbreviations

Open source AI is a free to access platform in which the code can be used and implemented as seen fit by the operator. OpenCV is a popular image processing example of this as well as TensorFlow and DL4j. Programming languages that may be used include C++, Python, and Java. The use of an IDE is likely, which means Integrated Development Environment which provides a more convenient setup for coding. Examples of IDEs include Microsoft Visual Studio and Eclipse. Black box design means that the user does not need to understand the internal functions of the program, just what it wants as input and what the user will receive as the output.

1.4 References

[1] IEEE Software Engineering Standards Committee, IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, June 25, 1998.

1.5 Overview

The remainder of the requirements document will contain a greater look into what precisely the project entails. This is going to involve looking at where the project stands for CDK and how it will be utilized as well as the products functions and characteristics for a user. Mention will also be made towards potential constraints placed upon the program.

The following section will be a general overview and description of the project, primary focus being placed on functions, characteristics, constraints, and assumptions/dependencies. After that there will be a section on specific requirements that will bring mention to potential interfacing concerns as well as performance metrics and standards compliance. The document is designed to follow the IEEE Software Engineering Standards [1].

2 OVERALL DESCRIPTION

2.1 Product Perspective

The core objective as well as the stretch goals all conclude that this product will be an independent platform. For this project, we will explore the potential of A.I. and create the foundation of CDKs own AI platform.

The signature detection and drivers license verification software must support various personal computers such as desktop, laptop, and tablet as an input source. This software will first receive a PDF file of a document as its input. It will then compute the data and give result or feedback to user. With that in mind, the software needs to be robust enough to compute large set of data. Similarly, within our stretch goals, vehicle model detection software must also support various personal computers and be robust enough to evaluate large data. However, it must also support portable devices with camera such as phone and tablet. Software should first take photo from camera then it evaluates the photo and returns model number.

2.2 Product Functions

Main goal The function of this software is detecting the location of the signature box and determine whether it is signed or unsigned. If document is unsigned: software must provide information of location and quantity of missing components. Also, the program must be capable of making a reasonable guess on whether a signature is sufficiently binding. This entails the AI must be capable of distinguishing between a simple mark in the signature window and an actual signature.

Stretch goal License validation software: This software must validate whether given photo is drivers license or not. First, it must validate the format of drivers license and check legal information such as expiration date.

Vehicle model detection software: Vehicle model detection software must be able to determine the specific model number from appearance of the vehicle. Software will first take photos or video of vehicle taken from different angle. It will determine vehicle model by narrowing down the information. For example, software will search sequentially for make, model, and year.

2.3   User Characteristics

The intended user of the software is vehicle retailers. We can assume a low degree of understanding on artificial intelligence. Therefore, software must be designed as black box model. To accompany this structure the user interface must be agreeable for them. Lastly, the program must not overburden the operator, it will take a single file as input to scan and return a conclusion to the user.

2.4   Constraints

There are few constraints on CDK A.I. data stream project. First, software must be standalone. The future goal of this project is to build CDKs own platform, so software must be independent. In addition, project must be kept in open source. The software must be made in such a way that, once our team has completed its work, the project be taken over by a new team within CDK. To meet this demand the need for thorough documentation of implementation is vital. This project will also have limits on testing due to the privacy issue surrounding actual consumer data. Since the data that signature detecting and license validation software compute contains personal data, testing data must be done on dummy documents and licenses. The last constraint is that product must function on its own without the additional use or maintenance of outside software. Since the users of the product are unlikely to be familiar with the internal workings the interface for them must sufficiently mask the technical workings of the program only provide a clean and simple menu.

2.5   Assumptions and Dependencies

All images provided by the client are sanitized and free of legal conflicts. Cloud based deployment services will be provided by the client. From a developing and testing standpoint, this project will be developped in C++ utilizing the OpenCV picture formatting library. This library assumes that the system is running GCC 4.4.x or later as well as CMake 2.8.7 or higher.

3   SPECIFIC REQUIREMENTS

3.1   External Interface Requirements

The focus of this project is building signature detecting artificial intelligence run on the CDK server. It must be designed as black box model, so user interface must be simple . Since software runs on the CDK server, communication interface must be secure and durable.

The structure of user interface is very simple. There are two functions in user interface. Software simply takes set of documents from the user, and it displays result of the signature detection to the user. Result of the signature will be any pages of the document where error is detected.

Communication between our software and the CDK server database is important; however, focus on this project is making accurate and reliable artificial intelligence. We are considering the use of docker and amazon web services to construct a reliable communication structure but this is a stretch goal of the project.

3.2   System Features

3.2.1   Signature detecting capability

3.2.1.1   Introduction/Purpose of feature: The program must be able to detect if a document contains a signature regardless of the documents formatting.

3.2.1.2   Stimulus/Response sequence: The program will parse a PDF image of a document, or a document in PDF format, and output a percentage based estimation as to whether it contains a signature.

3.2.1.3   Functional Requirements: Signature detection must be functional for all types of document formats regardless of the type so long as they are presented in PDF file format.

3.2.2   Missing signature detection

3.2.2.1   Introduction/Purpose of feature: The program must be able to detect if a document contains a space where there should be a signature regardless of the format of the form.

3.2.2.2   Stimulus/Response sequence: The program will parse a PDF image of a document , or a document in PDF format, and output a percentage based estimation as to whether it contains a space where a signature is missing.

3.2.2.3   Functional Requirements: Signature detection must be functional for all types of document formats regardless of the type so long as they are presented in PDF file format.

3.2.3   Multiple signature and multiple missing signature capability

3.2.3.1   Introduction/Purpose of feature: The program must be able to perform feature 3.2.1 for every signature contained within the document. The program must also be able to perform feature 3.2.2 for every space that is missing a signature within the document

3.2.3.2   Stimulus/Response sequence: The program will parse a PDF image of a document, or a document in PDF format, and output percent estimates for features 3.2.1 and 3.2.2 at every instance deemed applicable.

3.2.3.3   Functional Requirements: Multiple signature and multiple missing signature capabilities must be functional for all types of document formats regardless of the type so long as they are presented in PDF file format.

The following features are part of the groups stretch goals and are to be completed once the aforementioned features work to the satisfaction of the client.

3.2.4   Car detection within image

3.2.4.1   Introduction/Purpose of feature: The program must detect if a given picture contains a car within it.

3.2.4.2   Stimulus/Response sequence: The program will parse a PDF image and output a percentage based estimate as to whether it contains a car.

3.2.4.3   Functional Requirements: The program must be able to detect cars within any picture presented in PDF format.

3.2.5   License and Permit detection within document

3.2.5.1   Introduction/Purpose of feature: The program must detect if a document contains a license or a permit regardless of the format of the form.

3.2.5.2   Stimulus/Response sequence: The program will parse a PDF image of a document , or a document in PDF format, and output a percentage based estimation as to whether it contains a license, or a permit.

3.2.5.3   Functional Requirements: The program must be able to detect permits and licenses within any document presented in PDF format.

3.2.6   Parse license plate, license, and permit text into machine readable format

3.2.6.1   Introduction/Purpose of feature: Once the program can perform features 3.2.3 and 3.2.4, it must then parse the contents of these forms into machine readable format. This will allow the user to utilize this information to validate a clients identity, ties to the vehicle, etc.. For this task, we will utilize Googles Vision API.

3.2.6.2   Stimulus/Response sequence: The program will parse a license, permit, or license plate image in PDF format and extract the information presented into a machine readable format.

3.2.6.3   Functional Requirements: The program must be able to read information within any license plate, license, or permit regardless of the origin so long as they are presented in a PDF format.

3.2.7   Damage recognition

3.2.7.1   Introduction/Purpose of feature: The program must be able to parse a picture of a vehicle and detect whether this vehicle has sustained damage. Often CDK employees must go through pictures of vehicles in order to assess if they have been damaged. This feature aims at discarding all vehicles with no damage lessening the amount of pictures that must be assessed for damage.

3.2.7.2   Stimulus/Response sequence: The program will parse a picture of a car in PDF format and output a percentage estimate as to whether the car has been damaged.

3.2.7.3   Functional Requirements: The program must be able to detect damage within any picture of a car so long as it is submitted in PDF format.

3.2.8   Make and Model detection

3.2.8.1   Introduction/Purpose of feature: The program will parse the picture of a car and attempt to discern the make and the model.

3.2.8.2   Stimulus/Response sequence: The program will parse an image of car in PDF format and output a number of percentages assessing what the possible make and model combination might be for the vehicle.

3.2.8.3   Functional Requirements: The program must be able to detect the make and model of a vehicle regardless of the make and model so long as the picture is of a car, and is submitted in PDF format.

3.3   Performance requirements

This program will utilize deep learning algorithms to achieve all features except for 3.2.5. To do so, the program must run continuously for extended periods of time in order to ensure optimal learning time frames. The program must be able to catch errors and continue running unless the error is fatal. Segmentation faults, memory depletion and hardware errors are all considered fatal, all other errors must be caught, handled, and notified to the user.

3.4 Design Constraints

The program is free of design constraints so long as the interface ensures ease of access and interaction between the different components for all end users. This pertains to the ease of submission of pictures and documents to the software, as well as the presentation of all results in a readable and comprehensive manner.

3.5 Software system attributes

The program will require a platform in which to run uninterrupted for extended periods of time. The platform must be accessible to ensure that modification can be made whenever deemed necessary. The platform must be portable to ensure that the software can be deployed in a wide range of environments. Finally, the platform must be able to support heavy usage of memory resources to ensure the software does not encounter depletion of memory issues.

4 APPENDIX

This section details the groups schedule in order to complete the tasks per the school year of 2017. Included is a Gantt chart detailing what the specific task is and the time frame over which they're scheduled to take place.

Fall Term

Week 6: 10/29

End: Friday, December 8, 2017
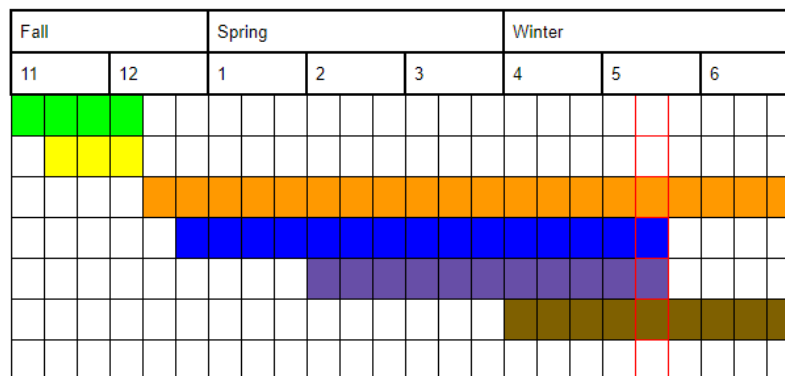
Winter term

Begin: Monday, January 8, 2018

End: Friday, March 23, 2018

Spring Term

Begin: Monday, April 2, 2018

Expo: Tentatively May 18, 2018

End: Friday, June 15, 2018

Fall		Spring			Winter		
11	12	1	2	3	4	5	6

Practice

–   Study machine learning framework

–   Build Small texture detecting program

Main: Build first prototype

–   Study on texture detecting A.I. algorithm

–   Build program that detects handwritten signatures from document

–   Adapt program to accept ranging document formats

–   Data feeding

- – Check License information
- – Check permit information

- – Optimize the program

  - – Debugging

Stretch Goals (to be scheduled after completion of original goals)

- – Detect the model of a vehicle
- – Detect the make of a vehicle
- – Detect/read license plate information
- – Detect/read license and permit information
- – Gather information of the condition of the vehicle (e.g.: dents, cracks and scratches)