CS CAPSTONE REQUIREMENTS DOCUMENT

NOVEMBER 3, 2017

CS461 FALL 2017

REAL-TIME SEED IDENTIFICATION

PREPARED FOR

OREGON STATE UNIVERSITY CROP SCIENCE DEPARTMENT

DANIEL CURRY

PREPARED BY

GROUP 11

GREEN TEAM

OMAR ELGEBALY
XIAOYI YANG
VINAYAKA THOMPSON

Abstract

The purpose of this project is to identify off-type seeds in real-time. The seeds go through a conveyor belt with a camera constantly feeding images to the classifier via a USB connection to a computer. If an off-type is detected, the computer will send a signal and the conveyor belt will trash any seeds in the picture involving a bad seed, The client allows false negatives but limits false positives. The scope of this project involves implementing a machine learning algorithm to identify seeds in real-time. Anything beyond the software is out ff this project's scope. However, it is important the software is designed to fit within the hardware constraints. The classifier should be binary. This means it only has to determine if a seed is an off-type or desired type. In this case, the desired type will mainly be Tall Fescue and Perennial Ryegrass.

CONTENTS

1 INTRODUCTION

1.1 Purpose

In the world today there is a need to sort seed. Reasons for sorting seed may include grading for pricing purposes or cleaning for reseeding. This is an arduous task requiring, for example, a researcher to go through the seed on by one under a microscope to determine the seed type and sort it. The difficulty is compounded when working with small seeds like those produced by grass plants. Usually, 90% or more of the seed is desired seed. Thus we are working on a device to reduce the work required to sort seed with a focus on sorting grass seed. The product we are working on is being designed for the OSU Seed Lab, making the lab-techs working for them our primary audience. There is, however, a secondary group that this device could be targeted to namely farmers needing to clean small batches of seed for reseeding and other such use cases.

1.2 Scope

The product to be created by our team is a program that is to work with a theoretical set of hardware components to sort seed. We must design our software with our hardware constraints in mind. This program is designed to take a photo of the seed (in this case grass), process it locally, and then send a signal only if an off-type is detected. The algorithm we make should be able to accurately identify alien seed within a maximum 20% margin of error. When testing seed, the fully assembled device should be able to process about 14 seeds per second. The client would like it to be able to process a batch of 25,000 seeds in approximately 30 minutes. Additionally, there was a desire that the device have some sort of user interface which would allow the user to change values and get feedback. When working on the above program, we need to build it in such a way that the following extensions to our product are possible. The first is the ability for the user to train the device or be able to switch seed type through some user installable upgrade. The second was categorizing the off-type seed thus giving an estimation of what types of alien seeds are present in the sample. Should we accomplish all of the above requirements to the best of our ability with significant time left before the deadline, the client would like us to start on one of the two extension projects.

1.3 Definitions acronyms and abbreviations

Client: In this case, our client is OSU Seed Lab

Lab-tech: Someone who sorts and identifies seed for OSU Seed Lab.

Researcher: Someone who works at OSU Seed Lab.

OSU: Oregon State University

Off-type: unwanted seed.

Desired type: Perennial Ryegrass or Tall Fescue. This is the seed we want.

1.4 References

https://developer.nvidia.com/deep-learning-frameworks

https://developer.nvidia.com/deep-learning-software

https://developer.nvidia.com/cuDNN

https://www.microsoft.com/en-us/cognitive-toolkit/

https://elinux.org/Jetson_TK1#About_Jetson_TK1

2 Overall description

2.1 Product perspective

2.1.1 User Interfaces

The user will be interacting with the product in 2 primary ways. The first is quite simple. To run the base device, the user must fill a hopper at one end of the device with the seed to be sorted and then turn the device on. The second way the user may interact with the device is through a screen should we start work on the user interface extension project.

2.1.2 Hardware Interfaces

The three base hardware components we will be using are the camera, the Nvidia Jetson chip, and the conveyor belt. There is also the option to add more memory through a disk drive. The way we intend to interface with the camera is over USB. If USB is not fast enough, we may try to switch to the CSI-2 MIPI camera serial interface. The conveyor belt, on the other hand, will be communicated with over GPIO pins.

2.1.3 Software Interfaces

Our product will have three main levels of software interfaces.

The first is the operating system. The current assumption is we are using NVIDIA's a modified Ubuntu 13.04 Linux distro.

The second level of software interfaces is the libraries we are going to use. We are using a Jetson processor. Therefore we will probably want to use cuDNN to get better GPU acceleration and OpenCV to help see the seeds. As for the library we plan on using for deep learning, we are still testing between Caffe2, CNTK, TensorFlow, Torch and dmlc mxnet. Because there are many choices with the deep learning library and it is such a big part of the project we plan to look into this in more detail and run some tests before picking one. The package we will be using is the most modern supported version of the library we pick as of November 2017.

In addition, we have the requirement to build our system with two expansion projects in mind. We need to allow for the interface expansion. This means a user should be able to train a different type of seed as a desired type. We also need top allow for the off-type classification. This means any image that is classified as an off-type should be placed into a folder for off-types only. This allows for a future expansion in classifying those off-types.

2.1.4 Memory constraints

The device we are working on has 2GB ram as primary storage. The secondary storage option on this device is provided by a full-size SD slot and 16GB fast eMMC 4.51 onboard the chip. This limits our options for secondary storage by giving us a maximum SD card capacity of 512 GB. There is also the option of mini-PCIe and SATA port that can be used to expand the storage space on the device should we need it. There should be one USB 3.0 port on the device that will probably be used for the camera limiting our ability to use a flash drive for extended secondary storage.

2.1.5 Operations

The basic operations of this device are as simple as turning it on. This will not be the case if someone creates the expansion project. If someone does work on that project, there will be some sort of screen added to the device allowing the user to interact with the added functionality.

2.2 Product functions

The primary function of this product is to sort grass seed with the requirement of removing 80% or more of the desired seed from the original seed mix with zero false positives. In this case, Tall Fescue and Perennial Ryegrass are the main targets for sorting. Therefore, they are the main focus of this project. In the future, the client expressed a desire to have the device determine the type of each seed running through the device including the off-types however as they require 100% success this is something beyond the scope of this project.

2.3 User characteristics

There are two target users for this device. The first and primary users will be the lab workers at the Seed Lab. In this case, the user would be some sort of lab tech who would be using the device to reduce their workload and more efficiently identify the off-types. This user would be a trained professional hired by the Seed Lab. These users would have in-depth knowledge about the process the device is trying to automate but may have little to no computer experience. The other potential user is a seed enthusiast in the public. Our client said something about wanting to, potentially in the future, merge capstone projects and expand the project into some sort of product that could be used by anyone who wanted to sort seed. This user could be at any stage of understanding the process the device is trying to automate and may or may not have computer experience.

2.4 Constraints

This project has a few true constraints provided by the client. The first constraints is that we must have no false positives. The device must be able to remove at least 80% of the good seed from the bad seed with no false positives. It has been stressed that there must be no alien seed in the good seed at the end of the devices run. We must program it to identify about 14 seeds per second. This means it should be able to process a batch of 25,000 seeds in approximately 30 minutes. In addition to this, the client expressed a desire for some additional systems in the future and wanted us to take that into consideration when building our system.

There are two future projects we discussed with the client. The first is allowing the client to change the seed sorted on the device by letting the client train the device on a different type of desired seed or install some sort of an update to the device. The other possible expansion project was to identify the types of bad seed that were removed from the good seed.

2.5 Assumptions and dependencies

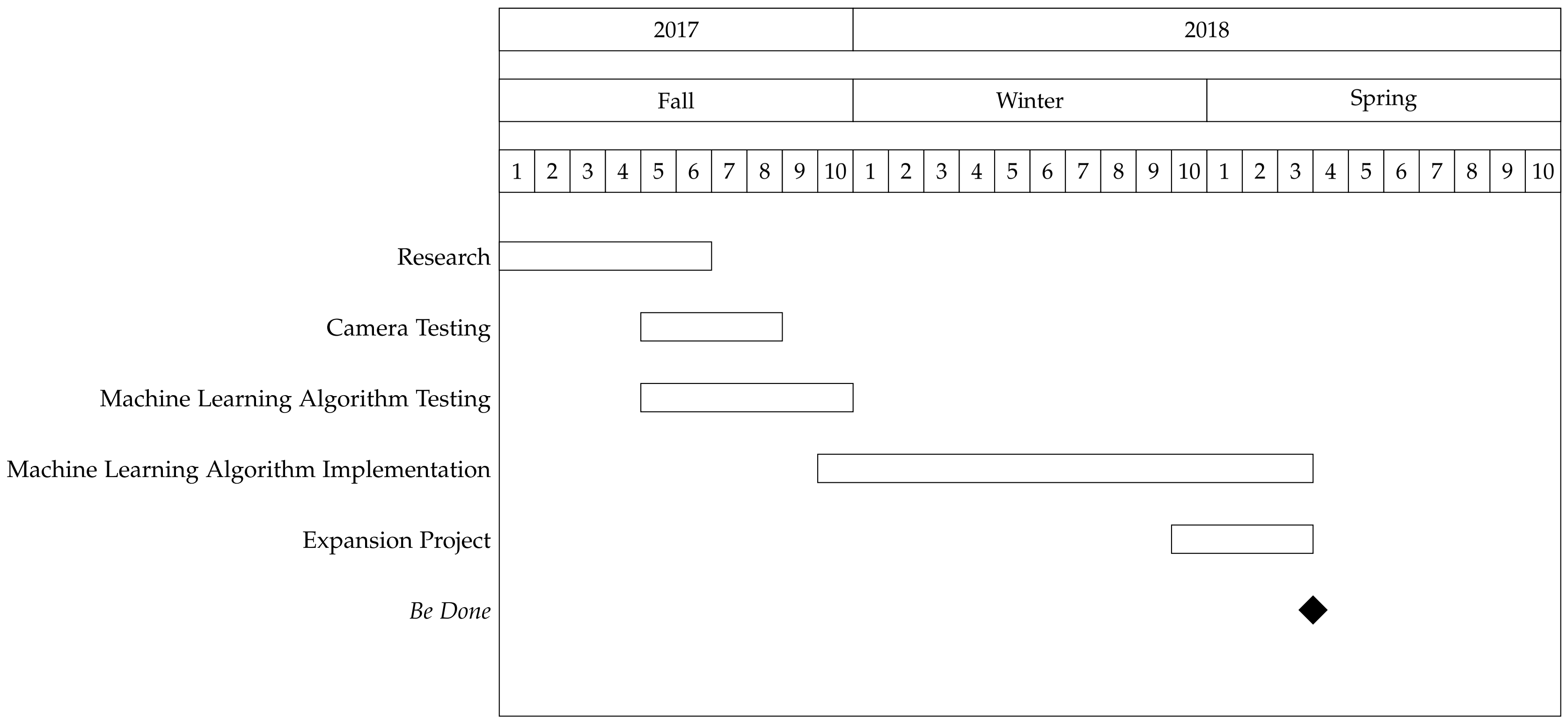
Though our project has some equipment we can use from a previous project, the client seems to be wanting us to start from scratch. This reduces the dependencies our group will have to contend with. The only two potential dependencies are a camera system the last group was able to setup and get working and the previous group's test bed as the only prebuilt system we have access to.

The first assumption is that we dont need to build any of the hardware components. We are also assuming that we only need to send a true-false signal to the hardware team. Finally, we assume that the hardware specs are what we specify regardless of what the mechanical team makes.

2.6 External Interfaces

The base project has no true user-facing external interface but does have an interface with the hardware teams project. This interface is a simple true-false signal sent through one of the GPIO pins.

3 Gantt Chart


Fall
Winter
Spring
Research
Camera Testing
Machine Learning Algorithm Testing
Machine Learning Algorithm Implementation
Expansion Project
Be Done