

Project Plan Assignment for CS 499

1. Planning and Estimating

a) Size estimate of the project in terms of Story Points.

- [K-d trees \(introduction\)](#) - 1 story points
 - Research k-d trees and algorithm(s) with which they work.
- [Researching TensorFlow \(introduction\)](#) - 2 story points
 - Research TensorFlow and other deep learning techniques to acquire an understanding of how the algorithm(s) function.
- [Code test bench for TensorFlow](#) - 8 story points
 - Create a simple working program that utilizes TensorFlow at the lowest level. This wouldn't yet work with the OCR program, but would rather be a functional rubric that can later be tuned to it.
- [Apply TensorFlow to OCR](#) - 9 story points
 - Use our understanding of TensorFlow to create a deep learning algorithm for an OCR.
- [Unit Testing and Debugging](#) - 5 story points
 - Run unit testing and quality control on tensorflow program applied to the OCR to make sure it is functional in all cases.
- [Integrate our deep learning algorithm into a .c file that can be called instead of the original kd.c file](#) - 3 story points
 - Make our program is callable from the original OCR program.
- Give user the option to use our algorithm - 1 story point
 - Edit the GUI so that the user can select either of the two classification algorithms.

b) Risk List

- Risk: One of the group members drops the class.
 - Mitigation: Regroup and reprioritize, scaling back on the more ambitious features of the project and cutting back on some “bells and whistles”.
- Risk: Our deep learning TensorFlow implementation fails to properly work on the OCR project, and we have to reconsider our entire design in the middle of the semester.

- Mitigation: Start off by creating a simple working program that utilizes tensorflow at the lowest level. This would ensure the logic of our algorithms, and due to the fact that we already have a huge database of correct OCR to test, it should be able to effectively recognize the correct characters once trained correctly.
- Risk: GitHub crashes and we lose our code repository.
 - Mitigation: Backup the entire GitHub on one of our own machines weekly in order to ensure safety of project.
- Risk: Group members have other time commitments, making it difficult to manage a strict timeline of completing the tasks.
 - Mitigation: Schedule two hourly weekly meetings for Monday and Wednesday at 2 pm at the Arts Library for progress reports and figuring out the next step. Schedule additional times as necessary.

c) **Schedule and allocation of resources**

- | | |
|---------------------------------------|----------------------|
| ● Research existing OCR program | complete |
| ● Run a .tif in English file thru OCR | Friday, September 20 |
| ● Researching k-d trees | Friday, September 20 |
| ● Researching tensorflow | Monday, September 23 |
| ● Testbench for tensorflow | Friday, October 12 |
| ● Apply tensorflow to OCR | Friday, October 26 |
| ● Unit testing and debugging | Monday, November 5 |
| ● Integration | Monday, November 12 |
| ● GUI | Friday, November 17 |

2. Requirements

Refer to the Requirement Specifications Document, which can be found here:

<https://github.com/BlakePrice/tensorflow-ocr/blob/master/Requirement%20Specifications%20Document.pdf>

Word count: 2704

3. Metrics

Estimated Story Points:

Copy existing OCR program and run it on our own machines - 1 story point (*complete*)
Run a new .tif file in English on the existing OCR program - 2 story points (*in progress*)
Research k-d trees - 1 story points ☐ (*in progress*)
Research deep learning and TensorFlow - 2 story points ☐ (*in progress*)
Code test bench for our deep learning algorithm and/or TensorFlow - 8 story points
Train our algorithm using the existing nearest neighbor search - 5 story points
Apply our deep learning algorithm to OCR - 9 story points
Unit Testing and Debugging - 5 story points
Integrate our algorithm into the existing kd.c file - 3 story points
Change the GUI - 1 story point

Product Size (to be completed):

User Stories:

- As a user, I should be able to choose between running the deep learning and nearest neighbor classification algorithm.
- As the product owner or software engineer, I should be able to compare the outputs of the existing nearest neighbor algorithm and the new deep learning algorithm.
- As a user, I should be able to see a green line over any character which may be incorrectly identified and a red line over any character which is distant from all possible matches.
- As the product owner, I need to ensure that the new algorithm is compatible with the existing OCR program.

Product Effort:

Atanas: 7 hours

Blake: 7 hours

Leah: 7 hours

Defects:

We initially weren't able to the ocr.h file from the Multilab because we didn't have read permission. However, this problem was resolved and there's currently no issues.

4. Web Page and Developer Notebook

Our blog can be found here: <https://github.com/BlakePrice/tensorflow-ocr>

Word count: 458