Uncommon Core Autograder Proposal Notes

**Use Case:**

What It Does:

Teachers will use this product in order to have a seamless experience for grading math worksheets in bulk.

Problem Area:

We aim to solve the hardships of grading massive quantities of math worksheets for teachers. Because grading large quantities of worksheets is difficult, public school teachers avoid giving proper quantities of math worksheets to students in order to help them really improve. Services that provide proper quantities of worksheets such as Kumon cost $100 a month, and still require dedicated parents to grade these sheets.

By solving this problem, we can make high quality math education more feasible for teachers and parents who have even low levels of dedication.

How Far We Go:

(MVP) Have our auto grader grade specific set of designed worksheets from a picture. Record the grades and student answers in a database for teachers to be able to view later through a visually pleasing web app.

(Reach) Have the additional feature of our autograder grading packets in real time by holding a mobile phone in video mode against a math packet and flipping the pages quickly.

ECE Areas:

Image Processing and Machine Learning (Classifying student handwriting), Web App Development (Hosting database backend on AWS/Heroku),

**Requirements:**

**Handwriting Bounding Box Accuracy -** Be able to find where students handwriting is with 90 percent accuracy.

**Grading Accuracy** - Be able to grade students with 90% accuracy on given worksheets if we correctly find where there handwriting is.

**Grading Robustness** - Be able to grade multiple different worksheet formats

(5 kinds of worksheets.)

**Web App Latency** - Grade Pages with 250ms of latency.

**Solution/Approach:**

Handwriting recognition - We would like to use deep learning techniques in order to make bounding boxes around the students answers using deep learning techniques. Then we would like to use the MNIST dataset as a prior to recognize the students handwriting. Then we would like to compare these answers to answers we have in our database and grade students.

Grading - We would like to take the students answers and compare them with correctly stored answers in a MYSQL database hosted on AWS. We would then like to store the performances of the students in the database. Preferably we could scrape the names of the students and the worksheet level and page numbers off of the sheets to store the performances in the right place in the database.

Presentation - We would like to make these student performances accessible for teachers to see on a Heroku web application.

**Design Trade Studies**

* Gradescope implements fairly rough autograding features.
* ZipGrade Implements autograding features on multiple choice scantrons.
* Bakpax sells handwritten autograding proprietarily.

However, what makes our product different is that it will be integrated within the Uncommon Core curriculum that automatically levels and progresses students eventually, and gives teachers analytics on the students progress.

**Project Schedule/ Work Distribution**

**Workflow :** Malcom and Jacob will plan the whole project through Asana.

**Weeks 1-2 :** Malcom and Jacob will pair program the database and the django backend.

**Weeks 3-4 :** Malcom and Jacob will work on bounding box handwriting detection.

**Weeks 5-6 :** Malcom and Jacob will work on reading student handwriting.

**Weeks 7-8 :** Malcom and Jacob will work on database backend and web app.

**Weeks 9-10 :** Malcom and Jacob will work on video autograding.

Division of labor in a strict manner isn’t ideal for a group of two. Use of Asana to eliminate tasks in a dynamic agile way is the proper way to go for a group of two.

**Risk Management**

We have designed the MVP to be fairly risk averse. If our MVP is overly ambitious, we will make the requirements less strict.

**Costs**

Potentially $0.