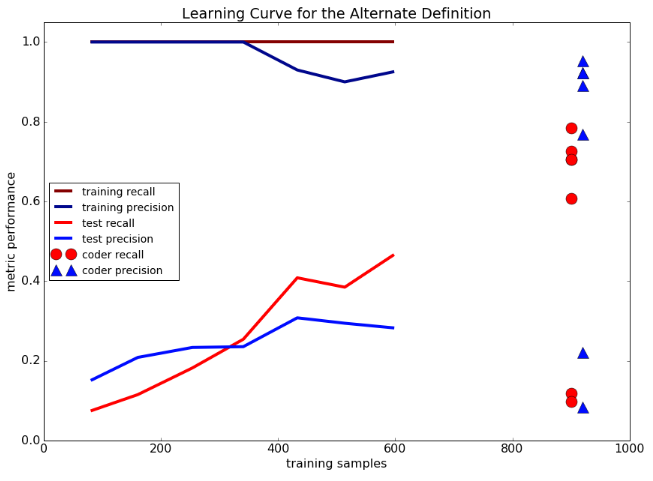
# Basic Info

Dashboard for Behavior Analysts to review and code images

# Background and Motivation

Currently Behavior Analysts at the University of Utah will observe classrooms in person or through video to evaluate behaviors of clients. Specific items are looked for such as out of seat, hitting, non-compliance, etc and the information is used to assess or change treatment plans.

Going through video is time-consuming and error prone (~70% of actual events are detected by any one observer). In an ideal situation an algorithm would process video and provide clips as well as statistics to the behavior analyst.



The goal of this project is to create the gui portion of a dashboard that allows a behavior analyst to view what a model thinks are events of interest image as well as enable a way to correct misclassifications.

# Project Objectives

Provide a way for non-technical educators to

1. View images of behaviors that a model thinks are of interest
2. Correct model misclassifications (so the model will get better with more use)

I would like to learn more about creating visualizations that add value to people who use them. In particular the security dashboard where the developer interacted with clients to iterate through solutions. Metrics such as adoption and user ratings were considered when evaluating effectiveness

.

Benefits of project

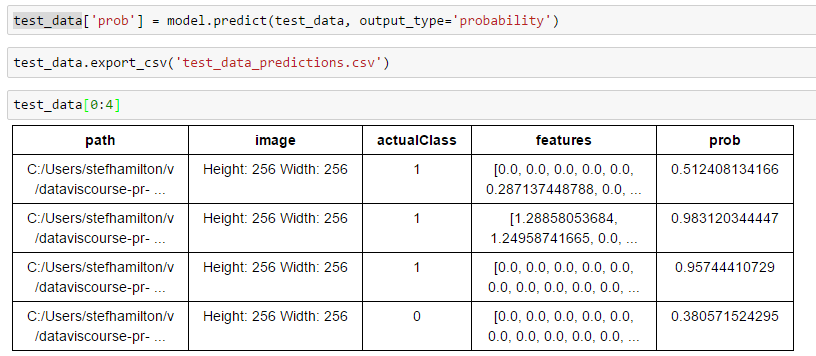
1. This could potentially lead to better treatments of Behavior Analyst clients since it could be cost effective to quantify the degree to which treatments are working
2. More specifically, having the images and ability to interact and view them in different manners, should improve the understanding of a situation. For example by looking at 1min before an incident, behavior triggers can be identified and integrated into a treatment plan for a client.

# Data

Since the actual data is HIPPA protected, I will pull images of hand raising off the internet. See data\unprocessedImages for the images that are being used.

# Data Processing

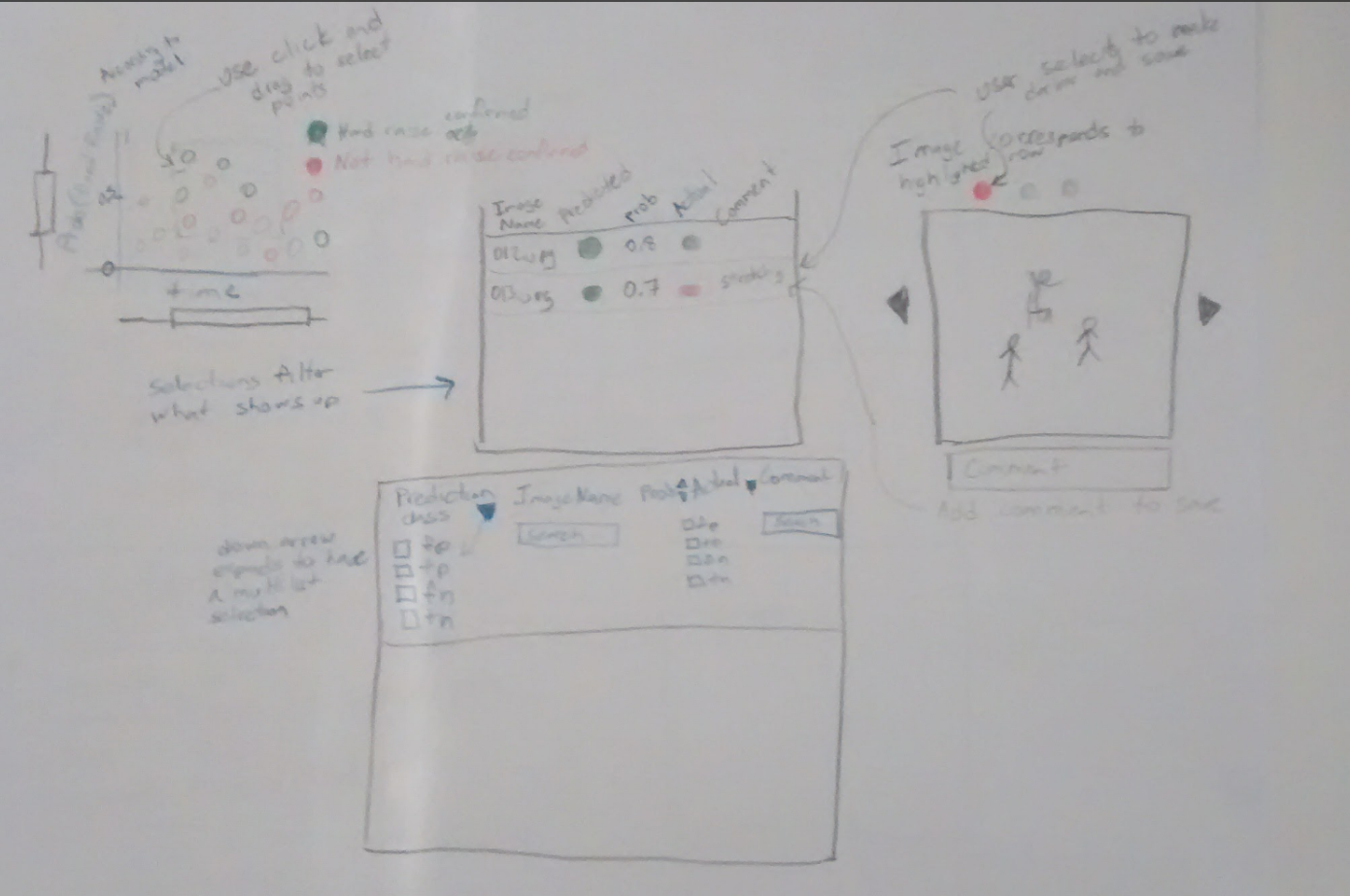
1. ~100 images of people with a hand raised and 100 images with no hand raises will be manually classified
2. A neural net classifier trained on Imagenet images will be repurposed to classify whether an image has a hand raise in it. See Reference: Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. “Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. 2012 or ModelAndImageProcessing.ipynb for details.



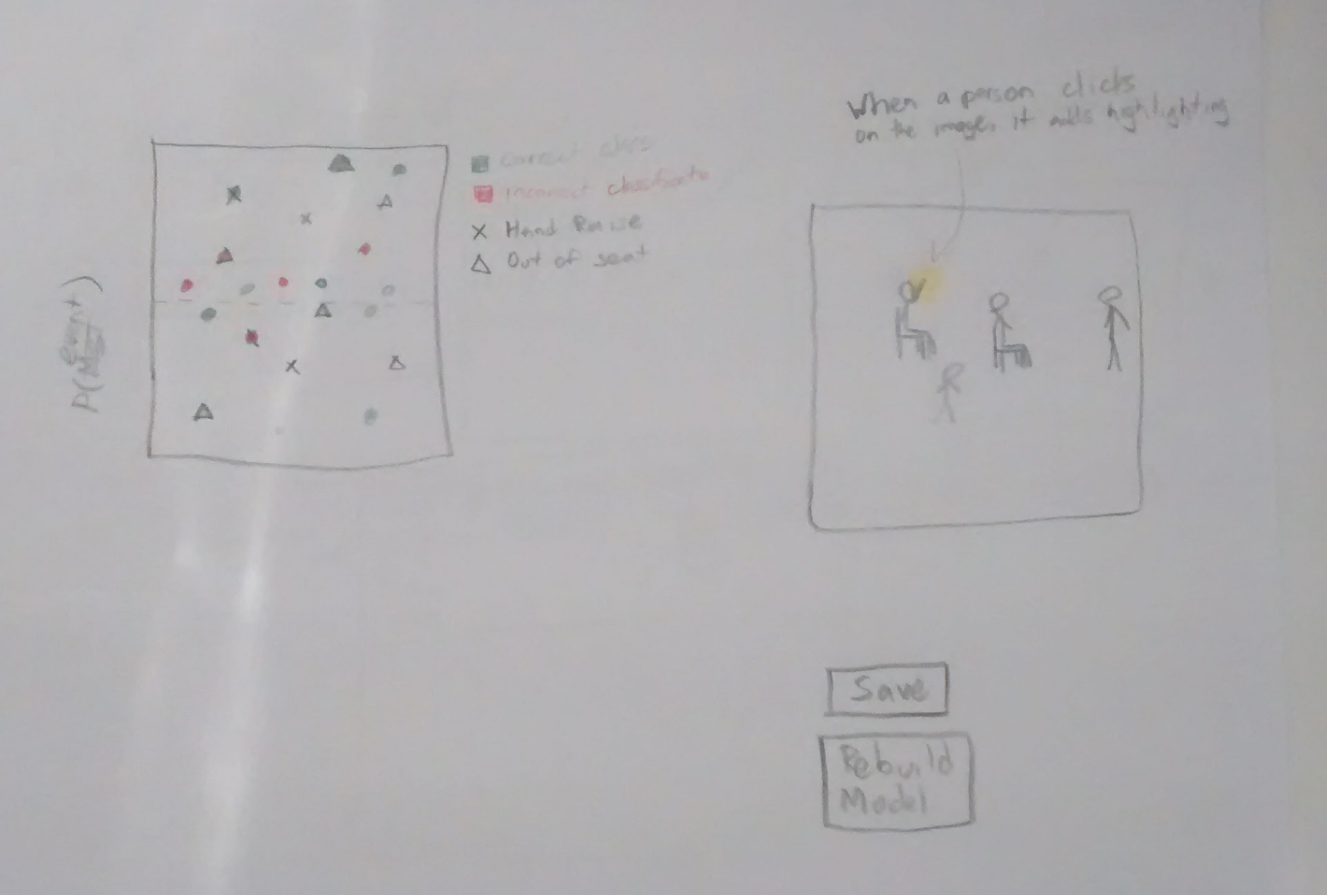
The 4 rows correspond to the 4 images below. Note how the confidence on the first is low since there weren’t many cartoon examples



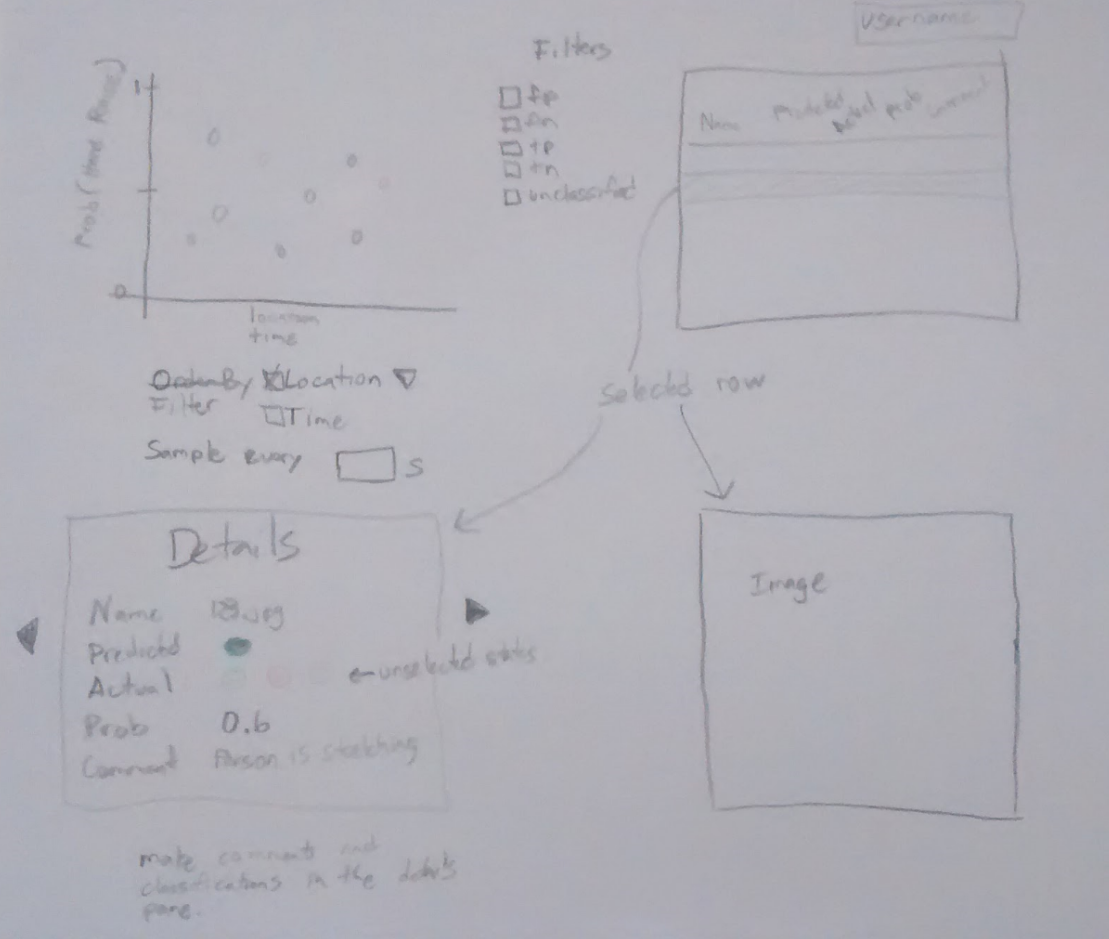
# Visualization Designs

.Design 1

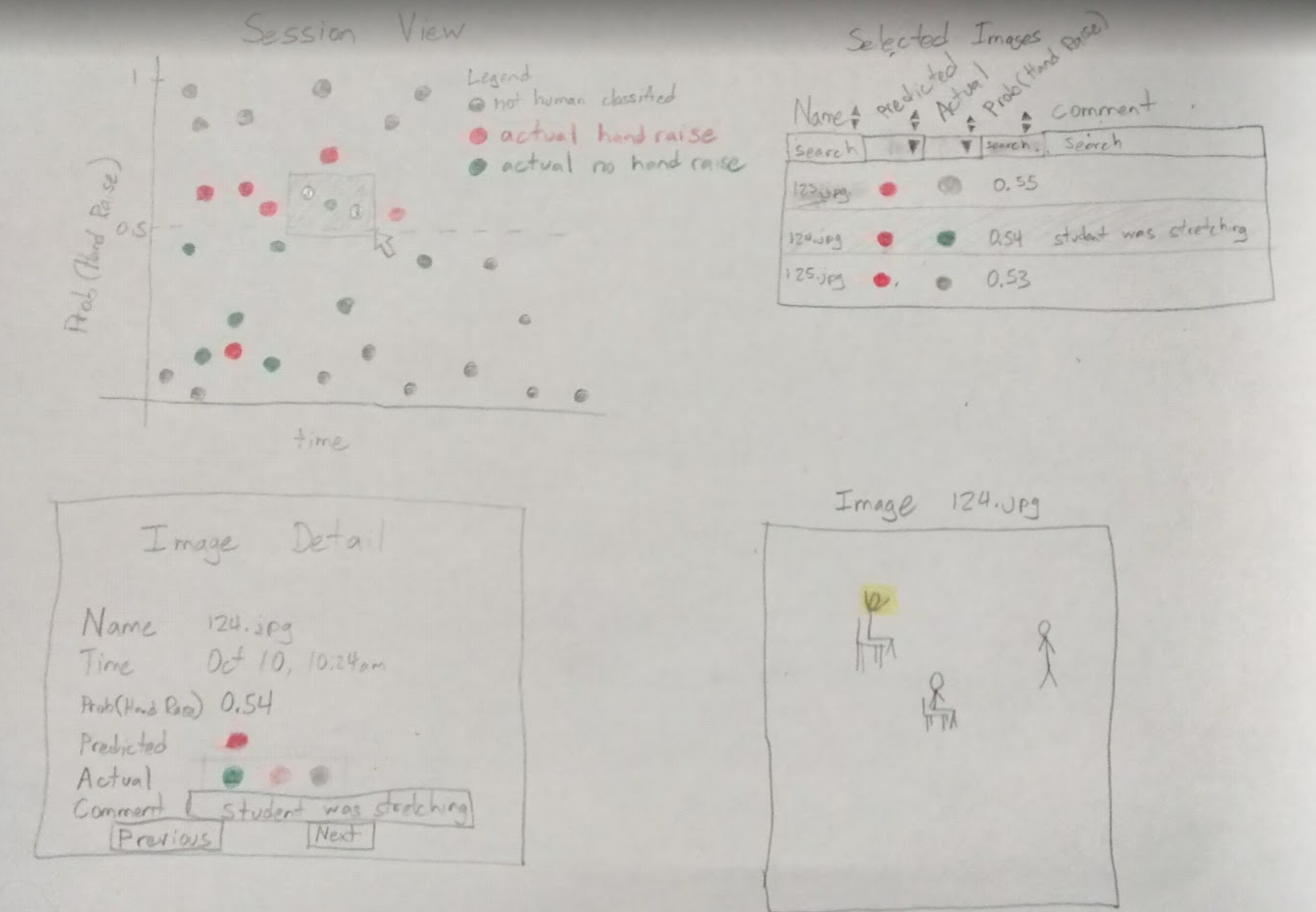
Design 2 (below)



Design 3 (is below)



Final Visualization based on the best ideas is below



**Must-Have Features.** List the features without which you would consider your project to be a failure.

* Top-left graph that shows event probabilities vs an x-axis. Brushing is implemented so that points selected show up in a list at top right
* Top-right list of selected images. When a row is selected, details show up at bottom left and the corresponding image shows up at bottom right
* Bottom-left details allows one to edit the actual classification which causes the corresponding top left graph to update with an indicator of actual classification. It is also possible to add a comment to the image that can be viewed and filtered in the selected images list
* Bottom right image appears when an image is selected

**Optional Features.** List the features which you consider to be nice to have, but not critical.

* Allow highlighting of the event in the image
* Load and save previous selections
* Order and text filters on the image list
* Keyboard shortcuts
* Stub in buttons / features of the ideal product such as rebuild model
* Integrate bootstrap styling

**Project Schedule.** Make sure that you plan your work so that you can avoid a big rush right before the final project deadline, and delegate different modules and responsibilities among your team members. Write this in terms of weekly deadlines.

By Oct 25

* Incorporate feedback from group review,
* Have sample data in github [SH]
* Decide on division of labor

Oct 27 Meet with Aaron to go over mockup

Nov 1

* 7pm phone meeting to discuss roadblocks / are we on track / rebalance workload

By Nov 11

* Figure out how to deploy to website
* functioning X by Y graph
* Functioning selected image list (filters are not necessary)
* Details are displayed
* Image displays

Nov 14-18

* Meet with an instructor to receive feedback

Nov 21

* Software development is complete

Nov 25

* Recording / writeups and entire project is ready to turn in

Dec 2

* Absolute deadline

Design Worksheets

