

Week 1

Progress

This week Chase finally did something. In particular, he uploaded all the images we're going to use to train our joint detector to Google Photos, pulled the URLs from the photos using a script he found on someone's blog, and then wrote a python script to extract the urls from the HTML they were embedded in in the script and reformatted them for use in our data labelling tool.

Jared and Ethan also made some progress: Jared made it so he can save images taken in our data collection app to the server.

Problems

We are running out of time and we still don't know how the swollen joint filter and the joint filter are supposed to interact. We also haven't heard anything back about our hardware situation for Expo day, so we are assuming that the school won't give us any of the hardware we requested.

Plans

Chase plans to finish the data collecting tool by addint custom attributes and instructions. Hopefully he will get that done by midway through next week. At that point, he will publish the project on Amazon Mechanical Turk and start evaluating the results to ensure they are good.

Week 2

Progress

This week we made almost no progress. Jared made some slight improvements to the app.

Problems

We are in full-blown panic mode as we approach the deadline. We have a 1 week extension in exchange for our 7 week delay due to Professor McGrath not giving us hardware or AWS access, but that still leaves us with 10 days to finish the Amazon Mechanical Turk tool, submit the job, get the results, and train a joint detector using the data.

Plans

Chase plans to finish the data collection tool and submit some test jobs to Amazon Mechanical Turk to see how the labelling process goes.

He'll probably start with a batch of 100 images and then go from there.

Week 3

Progress

This week we made some pretty substantial progress. Ethan got the Docker installation script working so we shouldn't get docked points for having a confusing submission. Jared put some finishing touches on the app so it will be ready for Expo. And Chase deployed the data labelling

tool to Amazon Mechanical Turk so that we can get our images labelled without spending 60 hours dragging boxes around joints.

Chase also worked on getting a server running on Amazon Web Services to listen to post requests from the data labelling tool that will send the labelled data to our server. He also worked on modifying the client-side code to make it simpler for workers to submit the labelled data once they are done labelling it.

Problems

Our Amazon Mechanical Turk data labelling tool is not working correctly. We realized right before deploying that our tool had great tools for labelling joints, but a clunky method of submitting the labels when the worker was done. Currently, the only way for a worker to get the results submitted is to download a file using the tool's interface and then email that file to an email address we put in the description of the post. The first worker to complete the task did not do that. And the second one gave up half way through.

It was clear that we needed a better way to submit the project when the worker is done. Our solution is to set up a NodeJS server with publicly accessible ports that will be contacted by the data labelling tool we deploy when the user is ready to submit. We'll run the server on Amazon Web Services and add a button into the interface that says "submit project. Our current problem with this approach is that we are getting some client-side errors when trying to contact the server.

Plans

Jared and Chase will try to fix the client-side errors with the data labelling tool later today or maybe this weekend. Chase will then modify the data labelling tool to add a submit button and possibly remove other superfluous UI elements.

Chase will then re-post the job to Mechanical Turk with a small number of images to label to see if the new interface results in the labels being properly submitted. If they are properly submitted, Chase will make several posts with a few thousand images so that we can get our images labelled before the code freeze on Monday.

Week 4

Progress

This week we completed our code freeze, our poster for Expo, met with our TA, and revised our requirements and design documents. It was a productive week.

Problems

Unfortunately our grand plan to collect 500 labelled images by using data labeling via Amazon Mechanical Turk did not work as intended. Our data collection tool lacked a backend, so our instructions to workers were to download a JSON file with the annotations via the GUI they used to label, attach that file to an email, and send it to a specified email address given in the instructions.

During the first test run of the data collection tool, a worker completed the task but did not email us the file. Chase reported the job as “done improperly” via the job monitoring console. Shortly thereafter, Chase received an email warning him that a worker had reported an issue with the post. The email gave a generic list of possible reasons that it might have been reported, among which was “broken UI”. Chase assumed this was the reason the post had been reported.

Little did we know, Amazon considers asking workers to email you files to be a violations of its terms of service (specifically its privacy rules that forbid requesters from asking workers for their email addresses). Since the sending email address of the workers was visible in the email sent to us, this constituted a violation of MTurk’s terms of service, resulting in Chase’s account getting banned.

As a result, Chase and Ethan spent 4 hours on Monday evening manually labelling 420 images of hands and the knuckles in each image.

Plans

Ethan and Chase still plan to use the labelled images to train a joint detector. We may appeal the Amazon Mechanical Turk decision once we have a working backend (or at the very least to get a refund of the \$50 we spent).

Jared also plans to make some improvements to his app so that it can both store an image in its own database and send it to the image segmentation server Ethan has set up.

Chase will work on fixing the web-based data labelling tool so that we can use Mechanical Turk to get the remainder of our images labelled.

Week 5

Progress

This week we finally got the data collection tool for Amazon Mechanical Turk working. The backend is now properly talking with the front end and the JSON files submitted by the client are saved into a folder on the server. Chase also modified the UI of the data labelling tool (the client side) to make it a bit easier for workers to find the button to submit everything.

Problems

The data collection server works when Chase's terminal from which he started the server is open. It does not, however, work when the terminal is closed or when the laptop is closed. This is obviously an issue, since we need the server running all the time no matter what the status of Chase's laptop is.

Chase is also currently banned from Amazon Mechanical Turk for unwittingly violating their terms of service by asking workers to email him a JSON file. He will call Amazon to attempt to get this situation remedied.

Plans

Chase is going to run the data collection server on the school computers. This shouldn't be a problem for the school since the server is basically idle 99.9% of the time, so it shouldn't take up too many resources. Hopefully there are no secret tools that kill off student processes that have been running for too long. Given that we ran servers on flip for CS340 that were active for weeks, we are hoping that is the case.

In the meantime, Chase or Ethan will start training the segmentation network to recognize joints.

Week 6

Progress

This week Ethan worked on training the joint detector with our labelled images of hands and Chase worked on getting his Mechanical Turk account unbanned. Chase also worked on putting the finishing touches on the data collection tool so that workers could submit JSON files to our group by clicking a button in the UI instead of downloading and emailing a file.

Problems

Chase has not been unbanned, nor is he likely to be (given that he was supposed to receive an email by Wednesday and has not yet received said email). This means Chase will lose \$50. Sad.

Another issue is that we are running out of time to get the joint detector working before expo. As of this blog post, we have 1 week left.

Plans

Our goal is to get a working demonstration of our joint detector working by expo day. This will require us to write a javascript script to parse the json file, download the images at the URLs specified, place those files in a local directory, and replace the URL of the image with a local filepath referencing the image's location. The last step in that process is turning out to be quite difficult.

After that's done, we should be able to train the classifier. At the very least we can have some still photos. Hopefully we'll have a video showing real-time detection that we can play on loop on someone's laptop.

