

Nicholas Kiddle

CS 463

5/13/20

Code Review: Addressing Responses

Category	Reviewer's Comment	Action Taken by Reviewed Group
Build	I can get accessible to the GitHub. There is a readme.md file. Readme file is good to read and follow. I download all files, but I failed to compile.	This failure may have been do to a hardware incompatibility, a NVIDIA GPU is needed. Reviewer did not specify...
Legibility	The code is modular, clean, easy to read, and has no redundancy. The structure between files is also good. I reviewed you code this afternoon, even though you explained your code this morning, I can't understand your code 100 percent. However, I still think your code is concise and good quality so far, because I am not developer and I think these functions are not complex and easy understood.	Thanks! Glad you think it is good.
Implementation	I checked the implementation video. I think your guys have good implementation.	Thanks!
Maintainability	There is one thing you can do is to make your code more understood is giving function information before your functions. I review your code; your team have comments in the code. If you guys can have function details comments that would be very helpful to read. Tell us how this function works for whole project, and that helps group members to maintain and us to understand your structure.	I think you are totally right. This project is very large though and I am just a single person group. I did not author much of this so I do not have the time nor expertise to go through all of the functions and meet my requirements.  I will add comment headers to the functions that I worked on though.
Requirements	I checked the requirements document. I think you guys have met the requirements.	Great!

Category	Reviewers Comment	Actions Take by Reviewed Group
Build	I was able to clone the repo and build from the README. If possible, I suggest using a shell script to set up the environment to simplify the installation process.	Good point. But actually the docker container is now available on dockerhub and that has the entire environment built in! Just requires docker and nvidia-docker to be installed. Then just a simple docker run will pull and run the latest version.
Legibility	The code for this project flowed as expected. Variable names represented their content and function names represented their use case. No suggestions necessary.	Thanks!
Implementation	I lack sufficient knowledge of the technology to improve upon the code base.	Fair enough.
Maintainability	There are no unit tests that I can find. Several functions could be tested using mock data, for example: <b>init_mask_from_points</b> function returns a mask that looks like a numpy array. This function could be tested by comparing its output with a mock array checking for equality. I suggest using python's unittest module since it is fairly easy to understand.	The numpy array that is output is pixel data. It would be near impossible to create dummy data to pass in. Let alone predict what the machine vision algorithm will identify as the object. This would end up being a project in of itself. Visual testing is going to work better for this.
Requirements	Yes the requirements are being fulfilled	Cool!
Other	Other than testing and improving the install experience, no I think group 22 did a great job!	Concern addressed above ^^ and thanks!

Category	Reviewers Comment	Action Taken by Reviewed Group
Build	<p>I did manage to get the project running eventually but it was a major timesink. I had to spend over an hour troubleshooting to get it working. The instructions were pretty clear I think but it's still a relatively complicated process to get started, not sure how this could be improved but if it could it would be very welcome. I initially thought I couldn't build it for example because the instructions linked to the Ubuntu distribution only. Some of the python dependencies weren't mentioned as well. Combining the two repos into 1, if possible, would be good as well.</p> <p>For parsing the actual code of the project, I found the second README from the bitbucket repo to be more useful, but I'm also not 100% clear on what was expected from group 22 based on the information provided and a lot of the information here might have been outside the scope of the project.</p>	<p>As mentioned in the github readme only the first readme needed to be executed. The BitBucket readme did not need to be done. We simply pull our code from there to later stuff into a docker container. So actually no python dependencies needed to be installed. Docker did this for us.</p> <p>The repos are separate since they perform two different tasks. The code in BitBucket is the main project code but we want to be able to separate it from docker since the client does not want to always use docker to run the project.</p> <p>Yes the second readme is just about the project code. The first readme is about building the docker image that contains the project code.</p>
Legibility	<p>The legibility here was subpar in my opinion, mostly just due to the lack of modularization of the code. The Annotator class on it's own spans over 4500 lines of code.</p>	<p>Yes, unfortunately it was like this when I joined. To refactor this it would take one person the a whole year likely and I am just a one person team. This was also outside of the project requirements.</p>
Implementation	<p>In my opinion, this code definitely needs to be modularized more. This falls into the pitfall a lot of OOP languages fall into where you have this giant master class that has become incredibly bloated over time, making it hard to read and making it much easier to just add new functions to it rather than try to sort it out, causing it to grow more and more.</p>	<p>I agree. I will pass this on to the client as a potential topic to work on for the next year. The above comment applies here too ^</p>
Maintainability	<p>There aren't any unit tests I can see and I'm honestly not sure what you would even test for this, without using some kind of default 'test' image that produces consistent results (if that's even possible?).</p>	<p>Yeah a default image would still potentially give varying results because of the machine learning algorithm changes often and is not very</p>

		predictable. An unfortunate pitfall of working with a black box.
Requirements	The requirements weren't clear to me on what the user was expected to complete for the project. The rectangle tool worked, which he mentioned working on, and the docker environment worked after a significant amount of tweaking.	<p>Not sure what tweaking was required. But the user did try to use the second readme which was not required per the first github readme linked.</p> <p>This is now all built into a public docker container anyways, so building is only 3 steps.</p>
Other	Really the readme workflow and code refactor seem like the biggest things to me.	Addressed in above comments.

Category	Reviewers Comment	Action Taken by Reviewed Group
Build	<p>I was not able to build the program as it is designed for use on a Linux system, and even if I could have built it I would not have been able to run the code as I do not have the proper GPU. Because of this, my evaluation for this section is based on just reading through the build instructions.</p> <p>The Readme in the Docker repo is well detailed with installation, build, and run instructions for setting up and using the container in conjunction with the larger project code on BitBucket. I do believe if I had the proper OS and hardware I would be able to successfully build the project based on the instructions. Running it is a bigger question as, compared to the Docker file documentation, the readme for the annotator repo is less detailed than it could be, but I am unsure how much of that repo was part of the project or if only the Docker work was, in which case it is unfair to grade the student based on shortcomings in the other code.</p>	<p>No problem.</p> <p>Yes the readme in Bitbucket has since been updated to be more clear. However, the BitBucket readme install instructions are not required to be completed. Docker compiles the whole requirements for us.</p>
Legibility	<p>The Docker file is well commented, making it clear what each part of the configuration is for and how it runs.</p> <p>The annotator file is cohesive in style, and fairly well commented. I would not say the code is easy to follow, but not due to poor design only because I am not familiar enough with this kind of programming to have a clear understanding of what it is all doing. There is clearly a logic to the design, I can see that the components work together, simply not the details of how.</p> <p>As before, judging the capstone work based on the entirety of the massive annotator file would not be right because the student only contributed some of this code. Viewing the commit history and focusing in on the contributions he made, specifically the implementation of the boundary box, the code is clean and fairly easy to follow.</p>	<p>Thanks!</p> <p>I did not design the layout of the annotator file. It certainly should be refactored, but this would be a project in of itself let alone for a one person team. This also was not part of the requirements.</p> <p>Thank you, glad it was readable.</p>

Implementation	Based on the demo video that was provided in the event the code could not be run first hand, efficiency was already improved from how the project was initially received by adding the boundary box in the annotator to only have to process pixels that are of interest. Being unfamiliar with the inner workings of this type of program, I am unable to see any obvious further improvements that could be made beyond this.	Cool!
Maintainability	There are not defined unit tests, but I don't know that there should be for the specific work involved in the capstone project. Unit tests for the annotator would be nice to confirm its functionality, but that seems outside the scope of this project if the student was only responsible for implementing some specific features in the annotator and then wrapping it in the Docker container. The test for that work is simply that it runs.	Yes, I agree. The annotator is difficult to test. It would take a long time to get a suite setup.
Requirements	Yes; the student was not responsible for the entire annotator, they were asked to add the user experience improvement and the efficiency improvement and create a Docker container for running the application, all of which were accomplished.	Great!
Other	The annotator file is so extensive that it may be easier for someone to understand the code if it was modularized a bit; though I don't know how feasible that is for how everything works together, and it would be out of the scope of this student's project to assign such a task to him, but it could be a note for the future.	Yes, outside of the scope of the project and very difficult to do as a single person unfamiliar with all the code. This would take the whole year.

Category	Reviewers Comment	Action Taken by Reviewed Group
Build	No issues building	Glad to hear.
Legibility	<p>Each function is aptly named with their names describing what they handle (e.g. Point, Label, or Annotator</p> <ul style="list-style-type: none"> <li>▪ Scrolling through &gt;1000 lines starts to hinder clarity</li> <li>▪ Each frontend method is of a good length (i.e. not too long and not too short). It is easy to see what the programmer was trying to do.</li> <li>• I would have personally preferred to have everything declared/initiated at the top of each method if possible so that we can see everything being declared at once instead of having to scroll through.</li> </ul> <p>Each line is clear and whitespace is used gratuitously. This makes it each to break off code into sections and makes it easier on the eyes.</p> <ul style="list-style-type: none"> <li>• Helps with logical breaks as well</li> </ul> <p>Indentation is great and consistent.</p> <ul style="list-style-type: none"> <li>▪ Echoing what I said earlier, I would have personally preferred to have everything declared/initiated at the top of each method if possible.</li> <li>▪ Coding style is consistent across all of the student's work.</li> <li>▪ Could clean up unused code</li> <li>▪ Use ===/!== to compare with true/false or</li> </ul>	<p>Yes, I agree. Refactor would become a project itself though.</p> <p>Great!</p> <p>Fair enough. To be honest though I don't think there is much time for this to be implemented. The project has already been posted online and there are still other requirements that need to be addressed. Doing this would sacrifice completing the requirements. I will note it to the client.</p> <p>Nice.</p> <p>Nice</p> <p>Addressed above ^</p> <p>Nice</p> <p>Has been done since review.</p> <p>Will do thanks for point that out.</p>

Implementation	<p>Maybe implement dragging image in addition to scroll bars. There might be a library out there that will help you out.</p> <ul style="list-style-type: none"> <li>o Bounding box is pretty, but I would recommend against using a gradient. Maybe switch to a color that catches the eye (like bright red) or just makes you want to gouge your eyes out.</li> <li>o Move position of elements</li> <li>▪ Clear buttons closer to the tools themselves</li> <li>▪ Remove padding between preview window and "toolbar"</li> <li>o Make the toolbar buttons look like buttons</li> <li>o I'd suggest changing == to === when doing comparisons to numbers. This happens in multiple places. You can probably easily find these if you just do a search all for ==</li> <li>o Split up the annotation js file into multiple helper files. 4000 lines is too much and would make it really hard to maintain. Maybe something someone next year could pick up.</li> <li>o Clean up training whitespace where necessary. (VS Code makes this really easy)</li> </ul>	<p>Hmm good idea. Would be very complicated though since we already have so many interactions woven into the image canvas. We would need to add a tool for that and then it would defeat the purpose of being toolless and add clicks...</p> <p>Multiple colors prevents the tool from becoming invisible in low contrast situations. A red box would be invisible against a red background.</p> <p>I'll look into this.</p> <p>Good idea, look into this too.</p> <p>Done in the most recent update!</p> <p>Addressed above ^^!</p> <p>This refactor would be impossible in the time we have left and for just one person. I will pass it on to the client.</p> <p>Not sure what training whitespace is. But the you mention the indentation is good earlier.</p>
Maintainability	<ul style="list-style-type: none"> <li>▪ No tests were found.</li> <li>▪ Maybe use cypress for UI testing</li> <li>▪ Mocha or Jest for other types of testing.</li> <li>▪ Lots of commented out code. I would try to clean that up a bit.</li> <li>▪ Aligned comments are really nice. Much easier to read.</li> </ul>	<p>There is not enough time left for me to implement testing. I am still working on the requirements set by the client. I will suggest these to the client.</p> <p>Has since been cleaned up.</p> <p>Will do if there is time after the requirements are completed since this is a little nitpicky and a personal preference of theirs.</p>



	<ul style="list-style-type: none"> <li>▪ Many comments explain what each variable represents.</li> <li>• Only problem I can see is that they are all declared in the same spot (which is good) but since there are so many, it is a bit daunting and can be potentially hard to read. If you must keep it in the format that the declarations are in at the moment (see function Annotator()), consider adding whitespace before each comment (break) to categorize each of the variables.</li> <li>▪ Repo contains readme of how to install and deploy</li> <li>▪ Contains link on how to use the tool</li> <li>▪ Markdown file could use some fixing</li> <li>▪ Could potentially use readmes.md in each of the subfolders as well just giving a high-level description and view of what everything does in each of the respective subdirectories.</li> </ul>	<p>Nice.</p> <p>Okay. Not a bad idea will do if there is time since this is a little nitpicky and a personal preference of theirs.</p> <p>Yep</p> <p>Yep</p> <p>Not sure which markdown file this is talking about or why it needs fixing. The BitBucket readme has since been updated.</p> <p>The readme in BitBucker has file descriptions in it to describe the more complicated and important files.</p>
Requirements	Section not provided.	
Other	Section not provided.	