Group 33 | Reviewer: Khoa Tran (Group 26)

Catego ry	Description	Reviewers Comment	Action taken by reviewed group
Build	Could you clone from Git and build using the README file?	N/A. Because I did not have access to an Intel RealSense depth camera, and I did not want to install a long list of dependencies, compile, and run something that will not work without a camera.	No actions needed.
Legibil ity	Was the flow sane and were variable names and methods easy to follow? Does the code adhere to general guidelines and code style?	I did not see long or confusing statements. In fact, the code is easy to follow, smartly factored, and well formatted. One thing though is I'm not sure why semicolons are used, even though they are optional in Python.	The members of our group who worked on the code for these portions agreed to use semicolons to the end of lines because that is what we are most used to from other languages.
Implem entatio n	Is it shorter/easier/faster/cleaner/safer to write functionally equivalent code? Do you see useful abstractions?	As mentioned, code is excellently modularized. Even if I did not fully understand the various libraries and algorithms used, I was able to follow the abstract flow of the code.	No actions needed.
Maintai n ability	Are there unity tests? Should there be? Are the test covering interesting cases? Are they readable?	I did not see unit tests. However, because the team is using many libraries to help with image processing and data collection, the critical errors would originate from these libraries. Smaller errors can be tested manually.	No actions needed.
Requir ements	Does the code fulfill the requirements?	Based on what the team demo at the code review, the code seems to fulfill the requirements. However, because we only saw the end-result of the training and image processing which have been done prior, I was not able to judge the quality of the training and data collection algorithms.	We demonstrated the data collection during the code review, and the training of the algorithm takes too long to demonstrate. We can easily test our model with our testingMetrics.py file to prove that the testing set is functional.
Other	Are there other things that stand out that can be improved?	Nothing comes to mind. I wish I had been able to compile and run the code to test it, but the team performed an excellent demo and walk-through of the code.	No actions needed.

Group 33 | Reviewer: Aidan Grimshaw (Group 26)

Category	Description	Reviewers Comment	Action taken by reviewed group
	Could you clone from Git and build using the README file?	I did not have the depth camera required to run the image gathering python script. I was not able to install dependencies using `{python version} -m pip install {package} –user` Instead I ended up using pip3 install {package} –user Additionally, when installing torch I had to use: pip install torch==1.4.0+cpu torchvision==0.5.0+cpu -f https://download.pytorch.org/whl/torch_stable.html When I tried to install the PIL dependency I found it was deprecated, and I had to use: pip3 install Pillow instead I would recommend adding a requirements.txt or similar so that all the packages don't have to be installed manually. I was not able to train the model because differences in the windows file system and Unix. This resulted in FileNotFoundError: [WinError 3] The system cannot find the path specified: '..\\datasets'	We made a setup to install all of the packages using pipreq and updated our README appropriately so the right packages would be installed. We have fixed the error on windows in order to update our paths.
Legibility	Was the flow sane and were variable names and methods easy to follow? Does the code adhere to general guidelines and code style?	I have a limited knowledge of Pytorch so I can't comment on the pytorch specifics. Commenting on transferLearning.py I would suggest removing all commented-out print statements. Also purely from a styling perspective I would remove semicolon line endings as they are unnecessary in Python. Otherwise, looks good.	We have made sure to comment more of the code, as well as remove commented out lines. The members of our group who worked on the code for these portions agreed to use semicolons to the end of lines because that is what we are most used to from other languages.

Implementation	Is it shorter/easier/faster/cleaner/safer to write functionally equivalent code? Do you see useful abstractions?	Commenting on transferLearning.py I would recommend splitting up the code into more functions rather than just train_model and main, I would recommend substituting in an additional function in the 3rd for loop in the train_model section.	We have reactored our transferLearning.py script to make more sense to those that might be unfamiliar to the algorithms used.
Maintainability	Are there unit tests? Should there be? Are the test covering interesting cases? Are they readable?	The project is a classifier, so it doesn't use testing in the traditional sense. However it uses a python script, testingMetrics.py, to determine the accuracy of the classifier. The image gathering code does not have unit tests but I don't really think that is important, as they are not meant to be productionized.	No actions needed.
Requirements	Does the code fulfill the requirements?	I tried testing on windows using testingMetrics.py and ran into the same file system issue I had in the build section. I tried running the code under WSL which didn't work either, but I don't neccesarily think this is the fault of the developers because WSL is a VM and can be finicky sometimes. Overall the demo looked good when the showed it via video, but I was not able to fully verify the accuracy of the classifier.	We have added more windows support in order to get the code to work better on the windows platform. We recommend using git bash on windows to resolve some errors with WSL.
Other	Are there other things that stand out that can be improved?	This is probably too much work to get done by the time of code freeze, but I would move the code to a platform like Floydhub or Kaggle. That would make it easier to demo your work as well as separate the data from the code into different storage units for ease of use.	We are not going to refactor our data storage at this point into this project, although we have considered doing this.

Aaron Leenknecht (Group 26)

Category	Description	Reviewers Comment	Action taken by reviewed group
Build	Could you clone from Git and build using the README file?	The README file is extensive and would theoretically be able to run. Not having the camera that is necessary to complete the task prevents proper use of the application. I think no actions are necessary.	No actions necessary.
Legibility	Was the flow sane and were variable names and methods easy to follow? Does the code adhere to general guidelines and code style?	I think more commenting should be done throughout all src code files. The three main directories: ML, SW, UI should be renamed to something that makes sense. Code guidelines/style seem to be adhered to and look good. The flow of the repo, for example: Capstone/src/code/* is a valid place to put all the code and would be easily identified/found.	We have named our directories in a way that separates the different components of our code. The naming to us makes sense, since this is how we divide the project up with the client. We recently refactored the code into this structure since we found it confusing to have everything in one folder.
Implementation	Is it shorter/easier/faster/cleaner/safer to write functionally equivalent code? Do you see useful abstractions?	Most src code files started out with a long list of imports, which can be cleaned up by: "import lib1, lib2, etc." all in one line. I'm not familiar with 99% of the project, so I'm not sure how I would re-factor it to better results. Based on the code review, multiple things have been done to improve implementation.	We have combined our imports unless they have a commented line above them explaining their import usage and reason. We felt like this is important documentation in the code and wanted to keep this.
Maintainability	Are there unity tests? Should there be? Are the test covering interesting cases? Are they readable?	No unit tests were written for simple tasks such as checking values. They have a testing system to test accuracy of their symbol prediction. I think that the majority of unit tests they could write would essentially be useless. The testing they have written are the most vital and complicated tests. They are readable in the sense that each time a prediction is given, their testing prints out a percent accuracy. A few unit tests could be wrote for vital computations.	Client requests of testing the ML model capabilities through a confusion matrix have been completed, and can be found in ML/testingMetric.py. Manual tests such as program launching without the presence of a camera have also been completed by our group.

Requir e ments	Does the code fulfill the requirements?	All requirements that I read were met except for the 3.2.2 subsection that says multiple testing methods will be implemented for the GUI. It mentions regression testing and unit testing to make sure output to user is accurate.	Manual testing was the only form of testing accomplished by our group for the UI. Accomplishment of regression/unit testing of UI was a stretch goal for our group and not required by our clients. We will update the requirements document to reflect these changes now that we are at the end of our project.
Other	Are there other things that stand out that can be improved?	Nothing stands out as something that needs to change. From the perspective of constantly improving things, I think that image processing and predicting could be faster and more accurate. More accurate can most likely be achieved by larger and more varied data sets. Github only allows a certain amount of data so maybe find a larger resource container.	Image processing has been refactored within the past few weeks, to handle batch processing of images, and background removal. Furthermore multithreading has been implemented to speedup the collection of training data for our group. As far as moving data to a new source, our group has already refactored the storage of our data from matrices of pixel values to raw images and GitHub has supported our needs now that we have finished collecting data and training mdoel.

Sam Young (Group 24)

Category	Description	Reviewers Comment	Action taken by reviewed group
Build	Could you clone from Git and build using the README file?	Yes, I was able to. Could not build and use due to not having the camera.	No action necessary.
Legibility	Was the flow sane and were variable names and methods easy to follow? Does the code adhere to general guidelines and code style?	I would prefer more comments in the code and more function separation.	Majority of code segments have commented portions explaining what it seeks to accomplish. Functional modularity implemented where needed, such as UI/datacollection.py
Implementation	Is it shorter/easier/faster/cleaner/safer to write functionally equivalent code? Do you see useful abstractions?	I prefer more functions.	SW/datacollection.py contains clear depiction of function modularity. UI/*.py files were autogenerated through the PyQT designer GUI, so lack of functional modularity likely is a cause of the autogeneration of the file.
Maintainability	Are there unity tests? Should there be? Are the test covering interesting cases? Are they readable?	There are not. It would be good to test if the app can even open without the camera.	Client requests of testing the ML model capabilities through a confusion matrix have been completed, and can be found in ML/testingMetric.py. Manual tests such as program launching without the presence of a camera have also been completed by our group.
Requirements	Does the code fulfill the requirements?	Yes	No action necessary.
Other	Are there other things that stand out that can be improved?	None	No action necessary.

Connor Maddalozzo (Group 24)

Category	Description	Reviewers Comment	Action taken by reviewed group
Build	Could you clone from Git and build using the README file?	No, It was also difficult to try and run the UIclientview.py. I think they could have been more clear on how to use virtualenv and pip with bash to download and run. -i didnt run the gpu program, i wasnt sure how to get the files on the pelican server.	README updated with clearer instructions on how to download packages. All dependencies can now be installed with a single
Legibility	Was the flow sane and were variable names and methods easy to follow? Does the code adhere to general guidelines and code style?	It did seem pretty dense, and hard to read, their python files. at least the end user ui python file. but i think thats fine, the yhad a lot to do in the file.	EndUserUI.py file was autogenerated upon creation through the PyQT designer GUI. The denseness likely has to do with the autoformat created by the designer.
Impleme ntation	Is it shorter/easier/faster/cleaner/safer to write functionally equivalent code? Do you see useful abstractions?	I am not sure. their files were very dense. I feel ike it might have been able to be shorted up, but maybe they just needed to have those lines written.	Throughout the past few weeks, our group have been reformatting certain areas of code to adhere to client needs.
Maintain ability	Are there unity tests? Should there be? Are the test covering interesting cases? Are they readable?	no unit tests. Im not sure which file more important to do unit tests on.	Client requests of testing the ML model capabilities through a confusion matrix have been completed.
Require ments	Does the code fulfill the requirements?	Well in their document it did say a-z minus j and z and they demonstrated that	No action necessary.
Other	Are there other things that stand out that can be improved?		No action necessary.

Jared Beale (Group 24)

Category	Description	Reviewers Comment	Action taken by reviewed group
Build	Could you clone from Git and build using the README file?	No. Group 33 indicated that the application would crash without the use of the Intel RealSense camera, which only they had access to. Instead of having us run their code, they demoed it in entirety during the review session on May 1.	No action necessary
Legibility	Was the flow sane and were variable names and methods easy to follow? Does the code adhere to general guidelines and code style?	Flow was sane and variable names were clear. For the most part, the code is well commented. Take a look at ML/import_model.py, which is largely bare. Also, there's some commented print statements that hasn't been removed in ML/transfer_learning.py and in ML/import_model.py. Otherwise, great. The style was good.	Comments added to ML/import_model.py file. Unneeded/commented out code removed from ML/transfer_learning.py file.
Impleme ntation	Is it shorter/easier/faster/cleaner/saf er to write functionally equivalent code? Do you see useful abstractions?	Being no expert in machine learning algorithms, I cannot say whether there would have been safer, cleaner, or easier ways to implement functionally equivalent code. I will say that in some places, code was cleverly organized to avoid duplication while preserving flow clarity (see ML/transferLearning.py's major "for phase" loop).	No action necessary
Maintain ability	Are there unit tests? Should there be? Are the test covering interesting cases? Are they readable?	There aren't unit tests, per se. There is a test that checks the accuracy of the machine learning algorithm by computing a confusion matrix. Like the source files, this test is well-commented and legible.	No action necessary.
Require ments	Does the code fulfill the requirements?	The code fulfills the requirements. It allows capture of data from the camera to be used both to train the machine learning algorithm and to be classified by it, through different user interfaces.	No action necessary.
Other	Are there other things that stand out that can be improved?	I am a bit confused about how some of the generated files were created. UI/dev_ui.py and UI/endUserUI.py both contain comments indicating they	Comments are likely autogenerated, when created by the UI team

		were generated from separate *.ui files which are not present.	in the PyQTGUI creator. No action necessary.