

- **Run final implementation and reproduce results presented in the GitHub - 2pt.**

It took much time to reproduce the results because of many non dynamic links. I had to manually change paths. After that I managed to reproduce the results. The results presented by the authors are the same I've got.

- **Describe produced results. Are they reasonable? - 2pt.**

The idea of the project was to reforge code of existing projects and add some functionality. The reported functionality works well. Comments and examples look better than several examples I checked in the original folder.

- **Would you recommend using this package / library for real-world problems? Explain your answers. If yes - provide examples. If not - explain what is missing - 2pt. About 300 words.**

The author did a great job on documenting and reforging the code. Overall, I would recommend it . Because the package was created to solve real world problems aka raster to vector transformation. And have comparably good documentation and methods on how to use it. It even has setup.py for package installation.

Real life problem examples are engineering drawing vectorization or any other black and white raster to vector transformation of technical drawings. Python methods are rewritten in an industrial way so it would be easy even to use some of them. There are plots showing real word applications. For instance, in https://github.com/adasegroup/Deep-vectorization-PR/blob/master/notebooks/pretrain_model_loading_and_evaluation_for_line.ipynb

Second reason I would recommend it is that the author created docker images so people would not have to deal with installation of every package.

In my opinion, the code is currently missing some try-catches for possible errors and some small refactoring like moving all imports to the top of the jupyter. It might be an option to use something like tensorboard for better presentation and customization of the result

Overall, after some final polishing steps this would be used by people.