OPEN-SOURCE SOFTWARE VERSION CONTROL SYSTEM

ECE 49595 - Spring 2024

Team Members:

- Vishal Muthuraja - vmuthura@purdue.edu
- Rohan Malavathu - rmalavat@purdue.edu
- Dev Thakkar - duthakka@purdue.edu

GitHub: https://github.com/Rmalavathu/Open-Source-Software-Senior-Design-Project

Repository Structure:

We are using a typical repository structure for ML projects and slightly tweaked it to fit our project needs.

LICENSE
Makefile
README.md
requirements.txt

Data
  Raw - Original, unedited videos
  Processed - Parsed videos

Docs - Folder storing documentation

Models - Folder containing trained models

Notebooks - Folder with Jupyter notebooks

Src - Folder containing the project code
  Aws - Scripts for cloud operations
  Models - Scripts for training models and making predictions
  Features - Folder for feature-related scripts

Web - Code for the web interface

Test - Folder with scripts for project testing

Api - Folder containing API endpoint scripts

OPEN-SOURCE SOFTWARE VERSION CONTROL SYSTEM

ECE 49595 - Spring 2024

Branching Strategy

Our branching strategy is a variant of the feature-branch branching strategy.

- **Master branch:** This branch is for production-ready code. It should be stable and only contain code that has been thoroughly tested.
- **Dev branch:** This branch is for staging releases, integration testing, and general QA testing. New features should be developed on this branch.
- **Feature branches:** These branches are for developing new features. They should be created from the develop branch and merged back into develop once the feature is complete.

Workflow

1. Create a new feature branch from the dev branch.
2. Develop the new feature on the feature branch.
3. Submit a pull request of feature branch into dev branch
4. Merge the feature branch into the dev branch.
5. Submit a pull request of dev branch into master branch
6. Merge the dev branch into the master branch when a new release is ready.

Review and Merge Policy

All code changes must be reviewed by both teammates before they can be merged into the master branch.

All code changes must be reviewed by at least one teammate before being merged into the dev branch.

In addition to the branching strategy and code review process, the pull request should mandatorily pass a continuous integration pipeline for merging to the master/dev branch. The CI pipeline should automate building, and testing, ensuring their quality and preventing build or test failures.While there would be a CD pipeline, to deploy code changes, for just the master branch.

Code reviews should be conducted constructively and collaboratively. The goal is to improve the code's quality and identify any potential issues.

OPEN-SOURCE SOFTWARE VERSION CONTROL SYSTEM

ECE 49595 - Spring 2024

Reviewers should check for the following:

- Correctness: Is the code free of bugs and does it meet the requirements?
- Readability: Is the code easy to understand and maintain?
- Maintainability: Is the code modular and easy to extend?
- Testability: Is the code easy to test?
- Documentation: Is the code well-documented?

Merges should only be done when all code changes have been reviewed and approved.

Additionally, we'll create a pull request template that is structured in the following manner:

- Description: What needed correction/What features were added
- Relevant Issues: List of issues in the repo that this addresses
- Methods: Thought process behind the code, addressing how the commits progressed, explaining what are the changes in the code

The following steps should be followed when merging code:

1. Create a pull request for the code changes.
2. Ensure that all code changes have been reviewed and approved.
3. Resolve any conflicts.
4. Merge the code changes into the master/dev branch.