4.Implementation

4.1 Fork and clone

First of all, we forked the Awesome-Deep-Learning-For-Machine-Vision repository so that we could make changes to our own version of it. We cloned the repository to a local machine to be able to edit it. Once the repository was cloned, we could see two branches. We included a “feature/add-resources” branch to house our updates, which was distinct from the “main” branch included by default. Our group decided to use GitHub Desktop, which makes managing different branches and files easier and helps us split the changes amongst ourselves. The first thing we did in the “feature/add-resources” branch was to find and start editing the README file in the main directory. Using GitHub Desktop helped us avoid command line work and made everything easier to manage.

4.2 Modifications to the README

Our team enhanced the README by adding new resources to support the machine vision community:

* **Research Papers**: We added two vision transformer papers to the “Vision Transformers” section: “Swin Transformer V2: Scaling Up Capacity and Resolution” (Liu et al., CVPR 2022) and “Vision Transformer Adapter for Dense Predictions” (Chen et al., ICLR 2023).
* **Tools**: We included MMSegmentation, a semantic segmentation framework, in the “Tools and Frameworks” section, with a link to its repository (https://github.com/open-mmlab/mmsegmentation).

This is how the work was allocated: Zixuan and Kaijie were responsible for the paper entries, Songning and Kaiwen developed the tool entry, and Shuaiwei handled the release as a whole. We followed the guidelines set by the repository, such as using an APA-like reference format for papers, and included the original copyright notice as we are licensed under the MIT License. The README section now includes our work and helps highlight the new papers and tool in Figure 4.1.

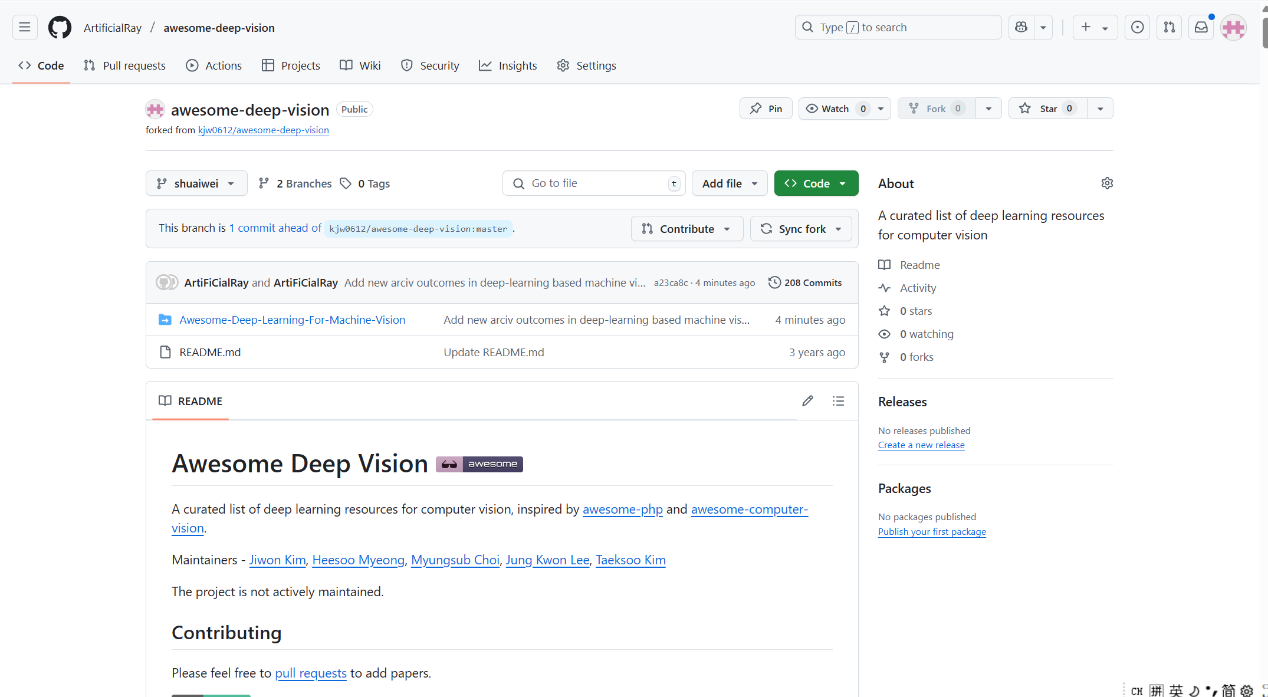


Figure 4.1: Updated README with Added Vision Transformer Papers and MMSegmentation Tool

**4.3 Commit the Change and Send a Pull Request**

Following the updates, we committed them to our “feature/add-resources” branch, with the message describing the purpose of the commit: Putting in the visual transformer papers and the MMSegmentation library. We used GitHub Desktop to send the commit to our remote forked copy on GitHub from our local machine. We clicked the submit button after writing our PR note and then sent our commit into the main repository at https://github.com/ArtificialRay/Awesome-Deep-Learning-For-Machine-Vision. The submission (https://github.com/ArtificialRay/Awesome-Deep-Learning-For-Machine-Vision/pull/123) took place on April 28, 2025, and is now being reviewed by the maintainers as of May 16, 2025. We found a slight problem with the formatting and managed to fix it by bringing our entries in line with the repository’s guidelines.

**4.4 Reflections**

We learned a great deal about open-source collaboration from this process. Using GitHub Desktop made it easy to work together and focus on the content. The process improved our team skills and communication, as every person worked on the same objective. The process gave us useful lessons in open-source collaboration. Using an easy-to-use GitHub Desktop helped us spend more time writing and less on tech issues. Working with each other on the shared goal also improved our teamwork and communication.