Meets Specifications

Congratulations!!
You have passed the project.
You have done an outstanding project with accurate results and detailed analysis.
Thank you for your effort and I wish you keep learning and never stop learning.
Good luck

Project Setup And Style

✓ **The code uses a virtual environment for project isolation.**

All required dependencies and their versions are listed in a requirements.txt file.

Tip: You should not paste the whole output of `pip freeze`. Only the required dependencies should be listed.

"Suggestion: Do not HARD Code versions in your requirements.txt file. That is not a good idea particularly when your application is going to run on multiple platforms, i.e. are you REALLY saying this application won't run with numpy 1.17? I doubt it."

- you can use pipreqs to create the dependency list for the project. This is the command you should use

  ```
  pipreqs . --force
  ```

  You can also use --no-pin flag to ommit the version number.

  ```
  pipreqs . --force --no-pin
  ```

✓ **When the program encounters an error, an exception is raised and a meaningful error message is displayed.**

The program uses logging to track important events.

I must say you have done an awesome job here

✓ **The project contains a README file with the following details:**

- **Short Introduction/Description**
- **Project Setup and Installation**
- **How to run a demo**
- **The command line options**
- **Explanation of the directory structure and overview of the files used in the project**

Inference Engine Pipeline

✓ **The program organizes reusable blocks of code into functions.**

The program encapsulates related methods and data into classes.

there is suggestion in the code review to enhance your code modularity

✓ **Based on user input, the project uses either a video file or a webcam feed to perform inference.**

✓ **The project includes an inference pipeline in which:**

- **Input frames are fed to models for inference.**
- **Outputs from multiple models are fed consecutively to other models.**

You did an excellent job creating a model pipeline. Keep it up!

✓ **The submission includes a write-up in the README. The write-up should contain:**

- **Benchmarking results for models of different precisions**
- **Discussion of the difference in the results among the models with different precisions (for instance, are some models more accurate than others?).**

✓ **The code allows the user to set a flag that can display the outputs of intermediate models.**

The output is shown using a visualization of the output model (not just printed).

Running the Program

✓ **The code uses command-line arguments to change the behavior of the program. For instance, specifying model file, specifying hardware type, etc.**

Where possible, default arguments are used for when the user does not specify the arguments.

A `--help` option should be present and should display information about the different configurations.

Excellent use of argument parser.

✓ **The program allows the user to select a hardware option on which to run the model (CPU, VPU etc). Inference then runs on the hardware that the user has chosen.**

Great job passing devices across all models at load time!