**Task3/**

**── Dataset/**

**── train/**

**── Rock/**

**── Paper/**

**── Scissor/**

**── None/**

**── test/**

**── Rock/**

**── Paper/**

**── Scissor/**

**── None/**

**── valid/**

**── Rock/**

**── Paper/**

**── Scissor/**

**── None/**

**── Arrange\_Dataset.py**

**── Model\_Utils.ipynb**

**── CNN\_Training.ipynb**

**── CNN\_Inference.ipynb**

**── Local\_Prediction.ipynb**

**── app.py**

**Here's an overview of each component:**

**Dataset:** This directory contains your image dataset, split into training, testing, and validation sets. Each set is further divided into classes (Rock, Paper, Scissor, None).

**Arrange\_Dataset.py:** This script is responsible for arranging your dataset into the desired directory structure as you mentioned in your question. It will take care of moving images into the appropriate class folders.

**Model\_Utils.ipynb:** This notebook contains code related to your CNN model architecture, transformations, and any utility functions needed for your model.

**CNN\_Training.ipynb:** This notebook contains code for training your CNN model. It may include sections for data loading, model training, validation, testing, and saving the best model checkpoint. You mentioned two versions: one with cross-validation and one without.

**CNN\_Inference.ipynb:** This notebook contains code for image and video prediction functions using your trained model. It may include functions for processing images and videos and using your model to make predictions.

**Local\_Prediction.ipynb:** This notebook is for running CNN inference locally on your machine. It might contain examples of how to load your model and make predictions on local images or videos.

**App.py:** This is your Flask application for deploying your model as a web service. It should include routes for accepting image uploads and returning predictions. Make sure to load your model and preprocessing functions in this script.

**To run the project, follow these steps:**

Use **Arrange\_Dataset.py** to organise your dataset.

Train your model using **CNN\_Training.ipynb**, either with or without cross-validation. Save the best model checkpoint.

Use **CNN\_Inference.ipynb** for image and video predictions.

If you want to deploy your model, use **app.py** with Flask to create a web service for making predictions over HTTP requests.

Remember to install the necessary Python libraries (e.g., PyTorch, Flask) and import the required modules in your scripts and notebooks to ensure everything works correctly.