Below is a comprehensive workflow that outlines the entire project—from setup to execution and output verification. This project is focused on detecting and annotating hand keypoints using MediaPipe and OpenCV. The workflow integrates the code in my two Python scripts (one for a basic keypoint extraction and another with added image annotation and error handling) along with the project documentation.

**1. Project Overview**

**Objective:**  
Detect and annotate hand keypoints from a set of images. The project:

* Reads images from a designated dataset folder.
* Uses MediaPipe to detect up to two hands per image.
* Extracts 21 keypoints (with normalized x, y, and z coordinates) per hand.
* Saves the keypoint coordinates in a CSV file.
* Optionally annotates the images with drawn landmarks and saves the results for visual verification.

**Applications:**  
This workflow is applicable for tasks such as gesture recognition, sign language detection, human-computer interaction, and AI-based hand tracking.

**2. Pre-requisites and Environment Setup**

**A. Software Requirements**

* **Python 3.x**: Ensure you are using a compatible Python version.
* **Libraries**:
  + [OpenCV](https://opencv.org/) for image processing.
  + [MediaPipe](https://mediapipe.dev/) for hand keypoint detection.
  + **CSV module**: (part of the Python standard library).

**B. Dependency Installation**

Make sure you have the necessary packages installed. You can use pip to install them:

pip install opencv-python mediapipe

**3. Workflow Execution Steps**

**Step 1. Setup and Verification**

* Verify that Python and the required libraries are installed.
* Confirm that the dataset\_test folder exists and contains your images.
* Update the file and folder paths in the scripts if you have a different structure. For example, if you want to use a different CSV filename or input folder, modify the variables at the beginning of the scripts.

**Step 2. Script Options**

You have two main scripts:

1. **21keypoints.py:**
   * Processes images from the dataset folder.
   * Detects hand keypoints and writes them to a CSV file (e.g., cam1.csv).
   * **Note:** This script does not annotate the images.
2. **21keypointsTest.py:**
   * Includes additional error handling (e.g., skips images that cannot be read).
   * Draws the detected landmarks on the images.
   * Saves the annotated images in a separate folder (dataset\_test\_keypoints).
   * Writes the keypoints to hand\_keypoints.csv.

**Choose the script that best fits your needs:**

* Use **21keypointsTest.py** if you want both CSV output and visualization through annotated images.
* Use **21keypoints.py** for a more basic keypoint extraction process.

**Step 3. Running the Script**

Open your terminal or command prompt, navigate to the project folder, and run the script. For example, to run the test script:

python 21keypointsTest.py

**What to Expect:**

* The script will iterate over every image in dataset\_test.
* It converts each image to the RGB color space and processes it via the MediaPipe Hands model.
* If hands are detected:
  + It draws the keypoints (landmarks) on the image.
  + Saves the annotated image into the dataset\_test\_keypoints folder.
  + Logs keypoint data (filename, hand index, keypoint index, x, y, z) into the CSV file.
* You will see console messages indicating progress and any potential errors (e.g., if an image cannot be read or if no hands are detected).

**Step 4. Output Verification**

After the script finishes execution, check:

* **CSV File:**  
  Open hand\_keypoints.csv with any spreadsheet viewer or CSV reader. It should list each image file processed along with the detected hand keypoints.
* **Annotated Images:**  
  The folder dataset\_test\_keypoints should now contain the images with the hand landmarks visibly drawn. This is ideal for a quick visual inspection to verify the quality of detection.

**4. Post-Processing and Further Steps**

* **Data Analysis:**  
  Use the CSV file for further processing, such as training machine learning models for gesture or sign language recognition.
* **Improvement and Customization:**  
  You can modify the script to adjust the detection confidence, change the output format (e.g., JSON), or integrate with other modules for real-time video processing.
* **Troubleshooting:**
  + Make sure the dataset folder is correctly named and placed.
  + Check if the images are accessible and in the proper format.
  + Review the console output for any error messages and adjust the parameters (like detection confidence) if necessary.

**6. Summary of Workflow Diagram**

1. **Preparation**
   * Install dependencies
   * Organize folder structure and place images
2. **Processing**
   * Read images from dataset\_test
   * For each image:
     + Convert to RGB
     + Detect hand keypoints using MediaPipe
     + (Optional) Annotate image and save to dataset\_test\_keypoints
     + Write keypoints data to CSV
3. **Output**
   * CSV file (hand\_keypoints.csv) with numerical keypoint data
   * Annotated images for visual feedback
4. **Post-Processing**
   * Use the data for analysis or integration into other projects