**Roadmap for the Hand Keypoints Detection Project**

**Phase 1: Project Planning and Setup**

**Objectives:**

* Define the project scope.
* Set clear outcomes: extract, annotate, and export hand keypoints from images.
* Identify necessary hardware, software, and dependencies.

**Tasks:**

* Review the project documentation (e.g., “Hand Keypoints Labeling - Documentation.docx”).
* Install required packages (MediaPipe, OpenCV, and any other dependencies) using pip:
* pip install opencv-python mediapipe
* Set up your project repository with the following structure:

├── 21keypoints.py

├── 21keypointsTest.py

├── dataset\_test/ # Contains input hand images

├── dataset\_test\_keypoints/ # Will contain annotated images

└── hand\_keypoints.csv # Will contain CSV output

**Deliverables:**

* A working development environment.
* A clearly defined project folder structure.

**Phase 2: Data Preparation and Organization**

**Objectives:**

* Ensure that the dataset is correctly organized.
* Validate that all input images are in the correct formats (e.g., PNG, JPG, BMP).

**Tasks:**

* Place all hand image files in the dataset\_test folder.
* Verify file names and formats; update the script variables if necessary.
* Optionally, perform basic checks on the dataset (e.g., using a Python script) to list images.

**Deliverables:**

* A well-structured dataset folder ready for processing.

**Phase 3: Implementation of Keypoint Extraction**

**Objectives:**

* Execute the core functionality using the provided code.
* Extract hand keypoints and annotate images with visual markers.

**Tasks:**

* Evaluate the two provided scripts:
  + **21keypoints.py:** Extracts keypoints and writes them into a CSV.
  + **21keypointsTest.py:** In addition to keypoint extraction, it annotates images and saves them for visual verification.
* Update file paths and parameters within the scripts if your folder names or file names differ.
* Run the chosen script (recommended: **21keypointsTest.py** for enhanced functionality):
* python 21keypointsTest.py

**Deliverables:**

* A CSV file (e.g., hand\_keypoints.csv) containing the normalized coordinates of detected hand keypoints.
* Annotated images saved in the dataset\_test\_keypoints folder.

**Phase 4: Testing and Validation**

**Objectives:**

* Confirm that the detection and annotation processes work as expected.
* Validate the accuracy and quality of the extracted keypoints.

**Tasks:**

* Review console logs for progress and error messages.
* Open the CSV file to ensure that data is recorded with:
  + Filename
  + Hand index
  + Keypoint index
  + Normalized x, y, and z values
* Visually inspect annotated images in the dataset\_test\_keypoints folder to verify the correct drawing of landmarks.
* Check error handling (e.g., images that can’t be read or where no hands are detected).

**Deliverables:**

* A list of confirmed outputs (CSV records and annotated images).

**Phase 5: Analysis and Feedback**

**Objectives:**

* Assess the performance of the hand keypoint detection.
* Identify opportunities to improve detection accuracy and efficiency.

**Tasks:**

* Analyze sample images and CSV data for consistency.
* Adjust detection parameters (for example, the min\_detection\_confidence) based on empirical results.
* Gather feedback on detection performance for further iteration.

**Deliverables:**

* An evaluation document (even if informal) noting how the current version performs.
* Recommendations for any potential modifications or improvements.

**Phase 6: Integration and Deployment**

**Objectives:**

* Prepare the system for integration into a larger application or for production use.
* Ensure the code is robust, modular, and maintainable.

**Tasks:**

* Integrate the keypoint detection module into a larger workflow if required (such as a gesture recognition system or UI application).
* Set up version control (e.g., using Git) and, if applicable, continuous integration and automated tests.
* Package the code and documentation as needed for end-users or further development.

**Deliverables:**

* An integrated system ready for deployment.
* Versioned code that can be maintained and updated.

**Phase 7: Documentation**

**Objectives:**

* Finalize project documentation.
* Plan future iterations and additional features.

**Tasks:**

* Update the project documentation to reflect any changes or improvements.
* Detail the installation, execution, and troubleshooting steps clearly.
* Consider additional features such as real-time video processing or integration with machine learning models for gesture recognition.
* Set a roadmap for future enhancements (e.g., refining landmark detection, adding support for additional image formats, or implementing an interactive GUI).

**Deliverables:**

* Comprehensive project documentation that can be shared with stakeholders.

**Timeline Overview (Example)**

* **Week 1: Planning & Setup**
  + Environment setup, dependency installation, folder structure creation.
* **Week 2: Data Preparation & Initial Implementation**
  + Organize dataset, update scripts, run initial tests.
* **Week 3: Testing & Validation**
  + Debugging, visual verification, and CSV data validation.
* **Week 4: Analysis, Integration & Documentation**
  + Evaluate performance, integrate with larger systems if required, and finalize documentation.