User Manual for Video Streaming, Recording, Hand Landmark Extraction and Object Detection

Prepared for: Haig’s Quality Printing
Version 1.0
August 2024
Prepared by: Tarek Zahid

Table of Contents

1. Overview

This document provides instructions for running Python scripts developed for Haig's Quality Printing. It outlines the setup, requirements, and usage of scripts related to:

- Streaming and Recording
- Hand Landmark Detection and Extraction
- Object Detection

2. Setup and Installation

a. Unzipping the installation folder

Unzip the provided installation folder to your desktop. This will create a folder on your desktop containing all the necessary scripts and files.

b. Starting the terminal

1. Open Command Prompt (Terminal) on Windows:
   - Press **Win + R** to open the Run dialog.
   - Type **cmd** and press **Enter**. This will open the Command Prompt.

Or you can search for the app named Terminal from the start menu.

2. Navigate to the haigs_app Directory:
   - Once the Command Prompt is open, you need to change the directory to where the haigs_app folder is located on your desktop.
   - your username is YourUsername, type the following command and press Enter:

```
cd C:\Users\YourUsername\Desktop\haigs_app
```

c. Installing necessary python libraries

1. Ensure You Are in the Correct Directory:
   - First, make sure you have navigated to the **haigs_app** directory in the Command Prompt as explained earlier:

```
cd C:\Users\YourUsername\Desktop\haigs_app
```

2. Run the following command to install all the libraries listed in the **requirements.txt** file:

```
pip install -r requirements.txt
```

3. Execution and Usage

a. Streaming

This section explains how to run the stream.py script to monitor RTSP streams, check their active status, and calculate the frame rate for each active stream. The script requires a file containing RTSP stream URLs and allows you to set the interval for refreshing the stream checks.

Running the Script

1. Open the Command Prompt:

    - Start by opening the Command Prompt on your Windows machine.

2. Navigate to the haigs_app Directory:

    - Change the directory to where the stream.py script is located:

      ```
      cd C:\Users\YourUsername\Desktop\haigs_app
      ```

3. Prepare the Feeds File:
    - Ensure you have a text file (e.g., feeds.txt) containing the RTSP stream URLs. Each line in the file should be formatted as **stream_name:url**.
4. Run the Script:
    - Execute the script using the following command, specifying the path to your feeds file and the refresh interval in minutes:

      ```
      python stream.py --feeds_file feeds.txt --refresh_time_min 10
      ```

5. To stop the script, press the q key at any time.

b. Recording

This section explains how to run the record.py script to record RTSP streams using VLC, save the recordings as MP4 files, and handle non-recording periods. The script reads stream URLs from a file, checks for active streams, and records them for a specified duration, with the ability to stop recordings by pressing the 'q' key.

Running the Script

1. Open the Command Prompt:
   - Start by opening the Command Prompt on your Windows machine.
2. Navigate to the haigs_app Directory:
   - Change the directory to where the record.py script is located:

     cd C:\Users\YourUsername\Desktop\haigs_app
3. Prepare the Feeds File:
   - Ensure you have a text file (e.g., feeds.txt) containing the RTSP stream URLs. Each line should be formatted as stream_name:url.
4. Run the Script:

   Execute the script using the following command, specifying the path to your feeds file and the recording parameters:

```
python record.py --feeds_file feeds.txt --record_time_min 1 --record_max_time_min 100 --non_record_start 11:19 --non_record_end 11:20
```

Adjust the parameters as needed:

- **--record_time_min:** Duration to record each stream in minutes.
- **--record_max_time_min:** Maximum duration before the recording session ends.
- **--non_record_start:** Start time of the non-recording period in HH format.
- **--non_record_end:** End time of the non-recording period in HH format.

5. Press the q key at any time to stop the script and terminate recordings.

c. Hand Landmark Detection and Extraction

This section explains how to run the `extract_hand_landmarks.py` script to process videos, extract hand landmarks using MediaPipe, and save the results as CSV files. The script will process all videos in a specified input folder and create output files in a local folder named 'landmark_extracts'.

Running the Script

1. Open the Command Prompt
    - Start by opening the Command Prompt on your Windows machine.
2. Navigate to the haigs_app Directory:
    - Change the directory to where the extract_hand_landmarks.py script is located: **cd C:\Users\YourUsername\Desktop\haigs_app**
3. Prepare the Input Folder:
    - Ensure you have a folder containing your video files. The folder should be specified in the command line argument.
4. Run the Script:
    - Execute the script using the following command, specifying the path to your input folder:

      python extract_hand_landmarks.py C:\path\to\input\folder

    - Replace C:\path\to\input\folder with the path to your actual folder containing video files.
5. Check the Output:
    - The extracted hand landmarks will be saved as CSV files in a folder named 'landmark_extracts', which will be created in the same directory where the script is run.

d. Object Recognition

This section explains how to run the YOLO object detection script to perform inference on images and visualize the results. The script uses a trained YOLO model to detect objects in an image and displays the annotated image with bounding boxes and labels.

Running the Script

1. Open the Command Prompt:
    - Start by opening the Command Prompt on your Windows machine.
2. Navigate to the haigs_app Directory:
    - Change the directory to where the infer.py script is located:

    ```
    cd C:\Users\YourUsername\Desktop\haigs_app
    ```
3. Prepare the Model and Image:
    - Ensure you have the YOLO model file (trained_model.pt) and the image file in the local folder or specify their paths.
4. Run the Script:
    - Execute the script using the following command, specifying the path to your model file and the image:

      ```
      python infer.py trained_model.pt path\to\image.jpg
      ```
5. View Results:
    - The annotated image with detected objects will be displayed in a window.