VIT-AP UNIVERSITY

Machine Learning Lab
Slot:- L47+L48

Team Name – TechnoSapiens
How to run the application

Team Members:

Vennagiri Bhanu Sri Mohith -23BCE20010
Shaik Arshiya Anjum. -23BCE7053
V.C.Premchand Yadav -23BCE7167
MD Nasir Alam -23BCE8680
P.V.Satya Laxmi Aswitha -23BCE9302
Velaga Mouli -23BCE9374

Faculty Name:- Prof. Sai Babu Veesam

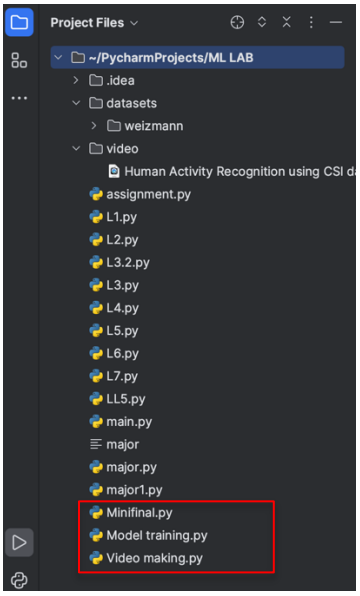
Running the Weizmann Action Recognition Application in PyCharm

This document provides step-by-step instructions on how to run the Weizmann action recognition application *within PyCharm*. For information on setting up your Python environment, installing PyCharm, and creating the necessary files, please refer to Help Document 1.

Prerequisites:

Make sure you have the following:

- You have followed the instructions in Help Document 1 to install Python, PyCharm, and the required libraries.
- You have created the necessary Python files: Video making.py, Model training.py, and Minifinal.py, and you have your trained model file (ending in .h5), like action_recognition_model.h5.
- You have the Weizmann Dataset located on your system.



Preparing the Video in PyCharm (using Video making.py):

1. **Open your project in PyCharm:** Open the PyCharm project containing your Python files and model.
2. **Edit Video making.py:**
    - Open Video making.py in the PyCharm editor.
    - Locate the line dataset_path = ... **Replace the existing path with the actual path to your Weizmann dataset folder.** For example: dataset_path = r"C:\Users\YourName\Documents\Weizmann" (Windows) or dataset_path =

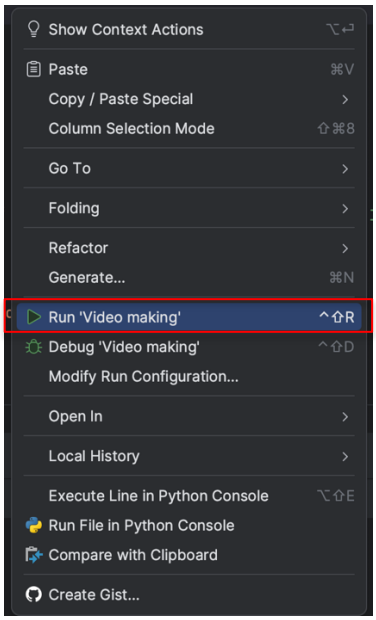
"/Users/YourName/Documents/Weizmann" (macOS/Linux). Make sure the path is inside quotes.

- Ensure that the output video file name in Video making.py is set to Combined_video.avi. It should look something like this (adjust the path if needed): output_video_name = "Combined_video.avi" or output_video_name = r"C:\path\to\your\output\Combined_video.avi".

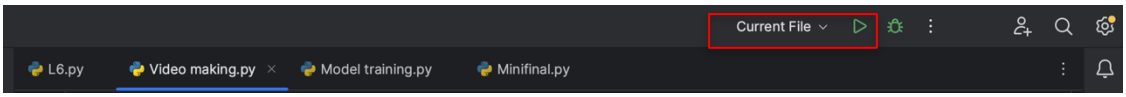
```python
import cv2
import os
from natsort import natsorted

# Path to the Weizmann dataset folder
dataset_path = r"C:\Users\velaga mouli\OneDrive\Desktop\Weizmann"  # Change this to your actual dataset path
output_video_name = "combined_video.avi"  # Final output video file

# Get the list of action folders (run, walk, wave, jump)
action_folders = [f for f in os.listdir(dataset_path) if os.path.isdir(os.path.join(dataset_path, f))]

# List to store all images from different folders
all_images = []

# Process each action folder and collect image paths
for action in action_folders:
    action_path = os.path.join(dataset_path, action)
```

- Save the file (Ctrl+S or Cmd+S).

3. **Run Video making.py:**
   - Right-click on the Video making.py file in the Project view (the panel on the left side of PyCharm).
   - Select Run 'Video making'.



- Alternatively, you can select current file from the drop down menu and click the green "Run" button in the top right corner of PyCharm. If Video making.py is the currently active file, it will run.



- This will create a video file named Combined_video.avi in your project directory (or the path you specified in output_video_name).

Running the Application in PyCharm (using Minifinal.py):

1. **Edit Minifinal.py:**
   - Open Minifinal.py in the PyCharm editor.
   - Find the line model_path = ... **Replace the existing path with the actual path to your trained model file (.h5 file).**

```
import cv2
import numpy as np
from tensorflow.keras.models import load_model
import mediapipe as mp

# Load your trained model (replace with the correct path to your model)
model_path = "action_recognition_model.h5"
model = load_model(model_path)
```

   - Find the line video_path = ... **Replace the existing path with the path to the Combined_video.avi file you just created.** For example: video_path = "Combined_video.avi" or video_path = r"C:\path\to\your\output\Combined_video.avi".

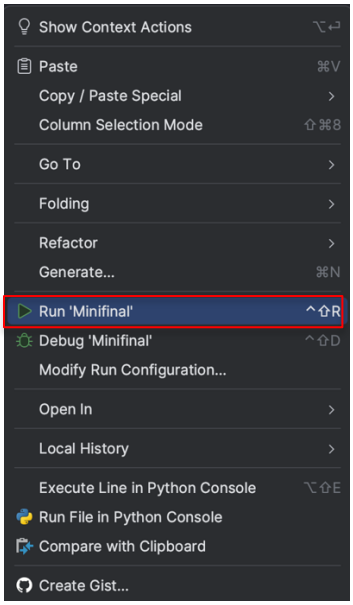
```
# Load the video file
video_path = "combined_video.avi"
cap = cv2.VideoCapture(video_path)

if not cap.isOpened():
    print("Error: Could not open video.")
    exit()
```

   - Find the line output_path = ... **Replace this with the desired path and name for the output video, which you've specified as Output_combined_video.avi.** For example: output_path = "Output_combined_video.avi" or output_path = r"C:\path\to\your\output\Output_combined_video.avi".

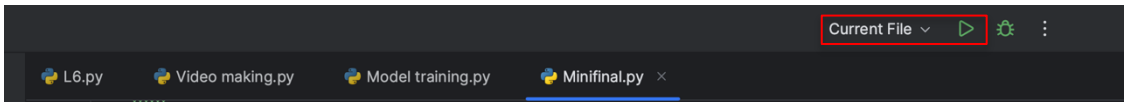
```
if not cap.isOpened():
    print("Error: Could not open video.")
    exit()

# Define the codec and create VideoWriter object
output_path = r"C:\Users\velaga mouli\OneDrive\Desktop\output_combined_video.avi"
fourcc = cv2.VideoWriter_fourcc(*'XVID')  # Codec format
fps = int(cap.get(cv2.CAP_PROP_FPS))  # Get FPS from input video
```

- Save the file.

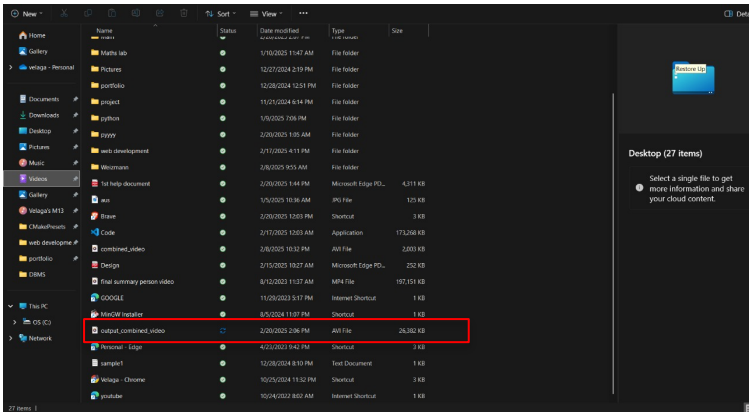
2. **Run Minifinal.py:**
   - Right-click on the Minifinal.py file in the Project view.
   - Select Run 'Minifinal'.



   - Alternatively, you can click the green "Run" button in the top right corner of PyCharm.

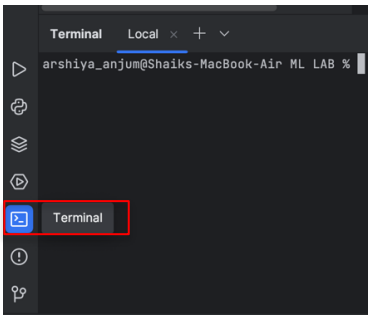


3. **View the Output:** The application will process the video, and a window will appear showing the video with action labels. The processed video will also be saved as Output_combined_video.avi to the output_path you specified.
4. **Execution Time:** The total process takes 3 minutes to complete, in the first 1 minute the application will process the video and then output is displayed for the next 2 minutes.



Troubleshooting:

- **ModuleNotFoundError:** This means you are missing some Python libraries. Open the PyCharm Terminal (View -> Tool Windows -> Terminal) and repeat the pip install commands from the "Required Libraries" section of Help Document 1.



- **File Not Found:** Double-check all file paths in the Python files. Make sure they are accurate. Copy and paste the paths if you are unsure to avoid typos. Pay close attention to the paths for Combined_video.avi and Output_combined_video.avi.
- **Errors during execution:** If you get an error message, copy and paste the error message into a search engine to find solutions.

If you encounter any problems, please provide the specific error message you're seeing for further assistance. Consult Help Document 1 for any environment-related issues.