Project FRIDA Setup

Prerequisite Requirements:

- Windows 10 64-bit with admin privileges.

  Note: While the program may be able to work on a different OS, it is not guaranteed.

- A webcam if wanting to use the live video feed option.

  Note: Can be internal or external. An external webcam is preferred as it would provide better ability for different angles for fall detection.

- A stable Internet connection for downloading.

Installation:

1. Extract the provided *3_FRIDA.zip* file in a location on your computer that you will remember.

   Note: If using GitHub, just navigate to where the repo folder was downloaded from this point onward.

2. Install Anaconda Individual Edition by going to: https://www.anaconda.com/products/individual

   Note: The download link is toward the bottom of the page under *Anaconda Installers*

3. Install PyCharm Community by going to: https://www.jetbrains.com/pycharm/download/#section=windows

   Note: You may need to restart your computer after this installation.

4. Open Anaconda Navigator (anaconda3).
5. Go to the *Environments* tab on the left-hand side.
6. Click *Import*
7. Type in *frida* next to *Name:*
8. Click the folder icon next to *Specification File*
9. Select *Conda explicit specification files (*.txt)* as the file type.
10. Navigate to the location of the *frida* folder and choose the provided *frida_libraries.txt* file within it, and wait for it to fully import the environment.
11. Make sure the *frida* environment is chosen instead of the default *base (root)* environment within Anaconda Navigator.
12. Go to the *Home* tab on the left-hand side.

13. Launch PyCharm.
14. Select *Open*
15. Navigate to the location of the *frida* folder and click *OK*
16. Wait for the program to fully load.

    Note: The loading status is displayed on the bottom of the IDE.
17. Go to *File* and then *Settings* on the top left-hand corner of the IDE.

    Alternatively, press Ctrl+Alt+S.
18. Type in *Project Interpreter* in the search bar on the top left-hand corner of the window and press Enter.
19. Click the gear icon to the left of *Project Interpreter* and click *Add…*
20. Select the *Conda Environment* section on the left-hand side of the window.
21. Check the *Existing environment* option.
22. Make sure next to *Interpreter:* that it is prefilled with a file path like the following:

    C:\Users\Name\anaconda3\envs\frida\Scripts\conda.exe

    Note: *Name* should be whatever your username is for your local account on your computer.
23. Click *OK* until you are out of all settings windows and wait for the program to reload.

    Note: You should notice from this point onward that near the bottom right-hand corner of the IDE it says *Python 3.7 (frida)*.
24. Open *Terminal* near the bottom of the IDE.
25. Type in the following commands one at a time:

    pip3 install -Iv opencv-python==4.1.2.30 --user

    pip3 install -Iv onnx==1.7.0 --user

    pip3 install -Iv onnxruntime==0.5.0 --user

    Note: You may receive incompatibility and PATH warnings upon installation. Please ignore them as the program should still work.
26. Close PyCharm and reopen it per steps 12 – 14 and wait for the program to fully load.
27. Click the *1:Project* tab on the left-hand side of the IDE and expand the *frida* folder.
28. Right-click *regular.py* and select *Run 'regular'*

    Alternatively, right-click *condensed_space.py* and select *Run 'condensed_space'*

Operating Instructions:

Video Options

As a user, you have an option to use either an .mp4 video file, such as the video dataset files provided in the *adl*, *fallcam0*, and *fallcam1* folders, or your webcam (internal or external). Below are the instructions on how to use either option:

- **Video Option 1** – Live video feed via a webcam
    - To use this option, comment out:

        camera = CameraSetUpVideoPlayback((“fallcam0/fall1cam0.mp4”)
    - Then uncomment:

        camera = CameraSetUpLiveVideo(0)
    - Note: Using this option will take the program much longer to initiate.
    - If your webcam will not load, please try replacing the argument *0* with *1* or *-1*.
- **Video Option 2** – Video file playback
    - This is the initially chosen option and the program should automatically play a video file from the *fallcam0* dataset folder.
    - To change which video file to be played, replace *fallcam0/fall1cam0.mp4* in *camera = CameraSetUpVideoPlayback((“fallcam0/fall1cam0.mp4”)* to the folder location of the desired video file.

        Note: The video file must be in .mp4 format as it is not guaranteed to work otherwise.

Video Frame

As a user, you will also have the option to use a grayscale, which is the default option, video frame, the background subtraction video frame, or both. Below are the instructions on how to use either or both options:

- (Default) **Grayscale**
    - By default, the program will automatically load a single video frame in grayscale.
    - If wanting to use an additional background subtraction video frame alongside the grayscale video frame, simply uncomment:

        cv2.imshow("Background Subtraction", frameTransform.frame_transform)

- Note: Running both the grayscale and background subtraction video frames simultaneously may cause increased latency and program initiation time.

- (Optional) **Background Subtraction**
  - If only a single background subtraction video frame is wanted, comment out:

    cv2.imshow("Video Feed", frame)
  - Then uncomment:

    cv2.imshow("Background Subtraction", frameTransform.frame_transform)

Output

Upon initiating the program run, you will notice in PyCharm's built-in console that it will display a loading status. If the program launches successfully, it will then display a success statement and load the desired video frame(s). If it does not, for instance it cannot detect a webcam or video input file, then it will either display an error in the console within the video frame itself as a form of a camera icon.

Once a grayscale video frame loads successfully, it will display a heads-up display within the video frame itself, showing a status of either *Idle*, which signifies no fall is detected, or *FALL DETECTED*, which signifies a person has fell. For the optional background subtraction video frame, no heads-up display will appear.

Within PyCharm's console, it will continuously display fall predictions (shown as *FP*) or non-fall predictions (shown as *NFP*). If a fall occurs, it will then display *FALL DETECTED*, and then continue displaying the fall and non-fall predictions.

If using the live video feed option, you will need to press 'q' on your keyboard to terminate the program, of which then you should notice a termination statement displayed in PyCharm's console. If using the video file option, the video frame will automatically close once the video ends, and it will terminate the program. A process finished statement will then be displayed in PyCharm's console.