# Human Fall Detection Readme

### Overview

The code can be used to detect, track humans and alert if it falls on to the surface. It processes the video files and saves the outputs with the fall count.

### Code Overview

It is based on the OpenPifPaf library of python which is mainly used for human pose estimation. The code continuously detects the human using the OpenCV library and classifies the pose till the body is being tracked and if it detects that the human has fallen through the code core/falldetector.py, it logs it along with printing it on the terminal.

### Code Algorithm

The main file of the code, Vehicle_Counting.py takes all the arguments which mainly consists of:

1. --source : input file name including the extension, for example input.mp4
2. --quiet : only prints on the terminal when fall is detected
3. –video-output : need to give the name of the outfile required, for example output.mp4 to save the processed video in video format
4. --show : shows the processing of the video on the screen

video.py is the main file which takes the arguments and functions detects, tracks and alerts in case of a fall. The core folder has the main functional files including fall detection and tracker.

### Setup

1. 
   ```
   # Setup Conda Environment
   $ conda create --name falldetection_openpifpaf python=3.7.6
   $ conda activate falldetection_openpifpaf
   ```
2. 
   ```
   # Download OpenPifPaf 0.11.9 (PyPI)
   $ pip3 install openpifpaf
   ```
3. 
   ```
   # Copy Source Files from this folder to the below directory which contains the openpifpaf function in the conda environment or in the global environment whatever exists
   $ cd {home_dir}/anaconda3/lib/python3.7/site-packages/openpifpaf # Global Path
   # OR
   $ cd {home_dir}/anaconda3/envs/falldetection_openpifpaf/lib/python3.7/site-packages/openpifpaf # Environment Path
   ```
4. 
   ```
   # Install Dependencies
   $ pip3 install -r requirements.txt
   ```

**Running the Code**

1.
   ```
   # Start Conda Environment
   $ conda activate falldetection_openpifpaf
   ```

2.
   ```
   # Execution
   $ python3 -m openpifpaf.video --source sample.mp4 --quiet --show --video-output
   output.mp4
   ```

**Remarks**

The code should be should strictly be ran on virtual environment considering we need to replace the internal python libraries. Using Linux would be make the process easier to implement.

**Requirements**

```
matplotlib==3.1.3
opencv-python==4.2.0.34
xmltodict===0.12.0
```

**Location**

Input: Input can be given from any folder once the original library is replaced from the given folder.

Output: The outputs are saved in the output folder with the screenshots of each fall being detected saved in img folder in the output directory.

**Argument Library**

Note: Any argument that needs to be put should be added after "--".

--source, default=None, help='OpenCV source url. Integer for webcams. Supports rtmp streams.'

--video-output', default=None, nargs='?', const=True, help='video output file'

--video-fps', default=show.AnimationFrame.video_fps, type=float

--show', default=False, action='store_true'

--horizontal-flip', default=False, action='store_true'

--no-colored-connections, dest='colored_connections', default=True, action='store_false', help='do not use colored connections to draw poses'

--disable-cuda', action='store_true', help='disable CUDA'

--json-output, default=None, nargs='?', const=True, help='json output file'

--quiet, default=False, action='store_true', help='only show warning messages or above')

**# Base Repository: https://github.com/cwlroda/falldetection_openpifpaf**