Demo of Human Ergonomic Analysis
(Tested on Windows 10, tensorflow-gpu=1.12, CUDA 9.0)

Table of Content

1.Quick Start

Note:

- This project needs tensorflow, whichs has both gpu and cpu versions. Download either tensorflow-gpu or tensorflow-cpu version to run this project folder. By default, the requirements.txt file will have you install tensorflow-gpu, but if you would like to use cpu, then simply go to the text file and delete the gpu part, which will then be just 'tensorflow.'
- When django server is launched, launch the web address in Microsoft Edge. Chrome does not support a video player function for html in http (When officially launched, it is supported in https).

$ git clone https://github.com/stephenbaek/inseer.git
$ cd django-upload-example
$ pip3 install -r requirements.txt

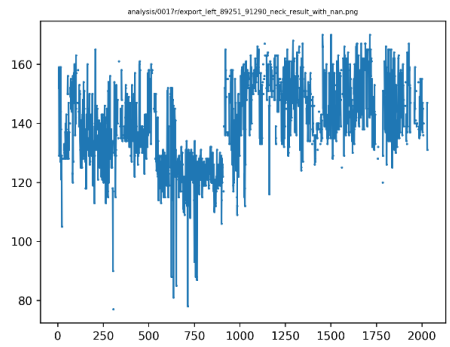
Download pre-converted yolo.h5 model file to /mysite/track/model:
https://drive.google.com/file/d/1v1B_6NxRQ-cd5G-Yy-1HpXmm8V677RLr/view?usp=sharing
Download pre-converted yolo.h5 model file to /mysite/track/model_data:
https://drive.google.com/file/d/1EBXhTU9X_8UOqbum_jXBxaydUNxspskI/view?usp=sharing

$ python manage.py runserver (Tested on Microsoft Edge)

Results

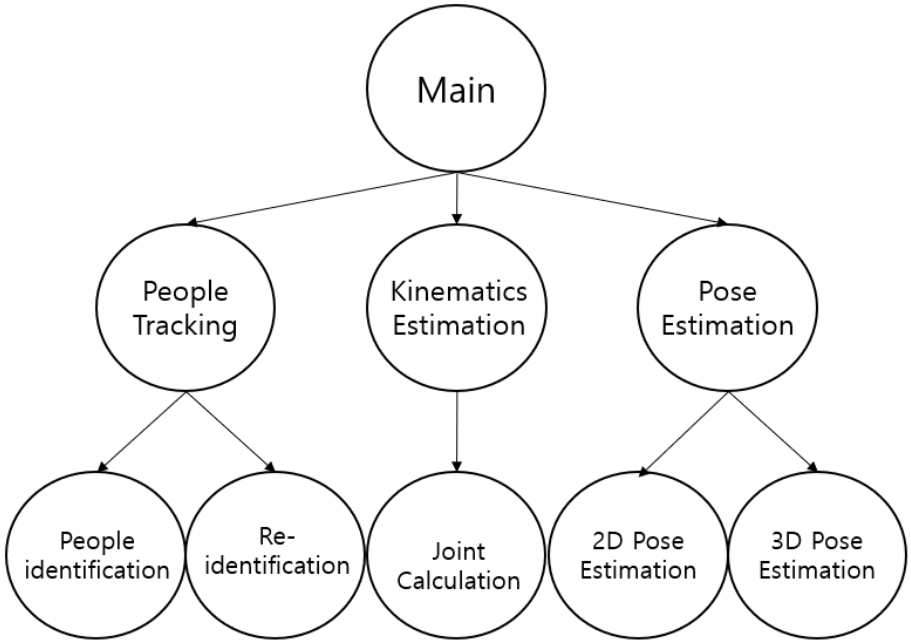
After completion, following files will be uploaded to the webpage for the user to download

- Original file
- Video with only box detection
- Video with both box and pose detection
- Visualization of shoulder and neck angles
- Text files of box coordinates, shoulder angles, neck angles



Structure of the project

The structure is organized in a way that each process does not interfere with another process. The processes execute in a unique manner, so that it is kept safe from the possible underlying errors.



Core Concepts:

People Tracking

- Detecting a person and then tracking the person throughout the video provides a fundamental base for other algorithms. This step includes person detection and person re-identification. A text file containing identification number and box coordinates will be saved.

Pose Estimation (2D & 3D)

- Based on the identification number provided from people tracking, the 2d pose estimation will be performed only on the given bounding box.
- 3D pose estimation is done based on the given 2D pose estimation

Kinematic Estimation

- A calculation based on careful calibration of how person is moving.
- The calculation is based on how long the person maintains that movement per category.

Trial and Error

People Tracking

1. Originally feature tracking was used simply extracts feature from a bounding box created from people detection algorithm and then compare the image to the next frame's extracted feature. It was not able to track a same person when detected in different cameras and the person's identification number suddenly changed.
2. Feature tracking was a little faster than deep sort algorithm, but the accuracy was much higher in deep sort algorithm
3. Deep sort algorithm (https://arxiv.org/abs/1703.07402) is able to track a person as long as it stays in the video, but when the person moves out of the frame and then comes back in, the person is perceived as a different person.

Pose Estimation (2D & 3D)

1. The original Caffe model (https://github.com/CMU-Perceptual-Computing-Lab/openpose) runtime was too long to estimate pose in real time, so a faster implementation had to be used
2. There are several different models, which accuracy seems to differ depending on the models.

It seems as if mobilenet_large works the best

3. In regard to performances on different models, mobilnet_v2_large (trained in 432x368) seemed to have the highest accuracy, so the mobilnet_v2_large model was used for this project

Kinematic Estimation

1. The calculations seem to vary depending on the person in the video. Also, the metrics used for defining the range of joint angles was not unified across different assessment, so an optimization in angles for calculating shoulder and neck is still needed.

Observations

- Person's neck angle is in mild angle from the start., which algorithm detects as a stress contributor. However, since it is a neutral stance for the person, the stress might not exist, which could lead to difference between algorithm and real person analysis.

- Videos that had a lot of frames recording the back was not able to get the poses from them. However, in other videos, the back of people were detected, so there is something wrong with the pose detection algorithm. This could be fixed and produce better result, if a better algorithm was used.

- In the analysis made by person, there are some parts that are 100%. This seems a little odd, because there are some frames that seem to be mild or severe, but the person just wrote 100% neutral.

- There are videos with only body shots. However, the pose algorithm assumes the whole body poses from few points, which leads to less missing rate. The person analyzing does not see the neck posture, so he just assumes missing.

- Person analyzing either underestimate or overestimate, because he cannot not accurately calculate the angle of the workers. This results in miscalculation of a worker assumed to be overworked when instead he was in neutral position.

- The pose detection algorithm correctly calculate the angles if the poses are detected accordingly. I could tell there were errors made by human analysist when comparing with the excel based on pose detection. Although pose detection may not be perfect for now, but if the probability of detection poses is enhanced, then the algorithm could really better understand how a working is doing.

Dependencies

Django=2.1.3
Tensorflow=1.12.0
Utils=0.9.0
Numpy
Pandas=0.23.4
Opencv=3.4.4.19
Numpy=1.16
Pillow
Xlwt=1.3.0
Sklearn
Keras
Matplotlib

Source:

Deep sort algorithm with Yolo:
https://github.com/Qidian213/deep_sort_yolov3
2D && 3D pose estimation repo (MobileNet):
https://github.com/ildoonet/tf-pose-estimation/tree/dc4a6a53f6be496d82b6d23d7ed85e75b63fbe10