SIT UP STRAIGHT, A REAL-TIME POSTURE DETECTOR

Yam Gui Peng David, Zhao Yazhi

Institute of Systems Science, National University of Singapore, Singapore 119615

ABSTRACT

In this project, the team successfully built a system to detect improper seating posture using a common computer webcam. This system relies on open-source algorithms such as face detection, facial landmarks and OpenPose body estimation. It visualizes the output in the form of a visual traffic light and provides real-time feedback for the user to adjust his/ her posture. The system has been evaluated and works well in various experimental environments.

***Index Terms*—** Posture Detection, real-time, OpenPose, Facial Landmarks

1. BUSINESS PROBLEM BACKGROUND

With workers spending a large portion of office hours in front of a computer, proper seating posture is necessary to reduce strain and aching. One study showed that "static working positions and poor postures are both associated with the development of musculo-skeletal disorders and discomfort". [1]

It was also found that improper sitting posture has a direct impact on back pain [2] [3]. With the current situation where most office workers have to Work From Home (WFH), this situation is even further exacerbated due to the improper setup of the tables/ chairs at home.

Furthermore, it is observed that most computers come pre-installed with a functioning webcam, and that it can be used for video conferencing and recording. It is thus the objective of this project to utilize the common webcam in laptops/ desktops to provide a means to monitor one's seated posture and provide real-time feedback in the form of audio & visual cues.

2. OBJECTIVES AND SUCCESS MEASUREMENTS

2.1. Objectives

1. To be able to detect improper posture based on the following positions:
    - Head is uneven (i.e. head tilted to the left/right)
    - Individual is hunching or slouching
    - Shoulders are uneven (i.e. shoulders slanted to the left/right)
2. To be able to do the real-time detection with common hardware components:
    - Front-facing computer webcam
    - Computer with CPU only

As an added feature, we also attempt to estimate if the user is looking at the screen or off to the side/ elsewhere.

2.2. Success Measurements

The performance of the Sit-Up-Straight system shall be evaluated by three ways:

- The system should be albe to provide corrective alerts & recommendations.
- The system should be able to work in different experimental environments.
- The system should be fast enough to fulfill real-time detection. (For example, a waiting time below 2 seconds for each action would be desirable.)

3. LITERATURE REVIEW

3.1. Facial Landmarks

"Accurate face landmarking and facial feature detection are important operations that have an impact on subsequent tasks focused on the face, such as ... expression ..., gaze detection..., etc." [4] The authors review 2 decades worth of facial landmarking papers and techniques. They split the area into model-based & texture-based methods and categorize the findings within. More recently, the authors of [5] train an ensemble of cascading regression trees to detect facial landmarks efficiently and relatively accurately. The model enables the detection of 194 landmarks on the HELEN dataset within milliseconds, which enables real-time use cases.

3.2. OpenPose Landmarks

OpenPose [6] represents the first real-time multi-person system to jointly detect human face, body, hands, and feet key-points (a total of 135 key points) on single images. [7] OpenPose initially detects key-points belonging to all individuals in an image, and subsequently assigns each key-point

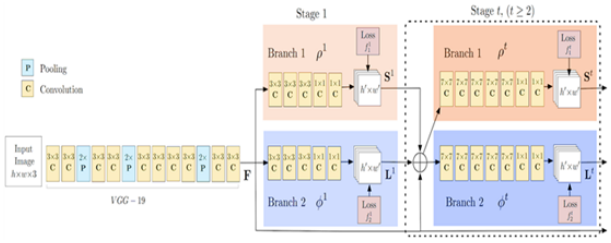

Fig. 1: OpenPose Neural Network Architecture

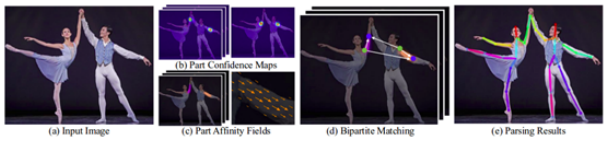

Fig. 2: OpenPose Example Images

to distinctive individuals in the image. The architecture of the OpenPose model is shown in Fig.1.

The OpenPose [8] network begins by extracting features from an image using a VGG-19 backbone. The features are then fed into two parallel branches of convolutional layers. The first branch predicts a set of 18 confidence maps, with each map representing a particular part of human pose skeleton. The second branch predicts a set of 38 Part Affinity Fields (PAFs) which represents the degree of association between parts.

Successive stages are then used to refine the predictions made by each branch. Using the part confidence maps, bipartite graphs are formed between pairs of key points. Using the PAF values, weaker links in the bipartite graphs are pruned. Through the steps above, human pose skeletons can be estimated and assigned to each individual person in the image. Show in Fig.2 are example visualizations of the various branches and components.

3.3. C3D

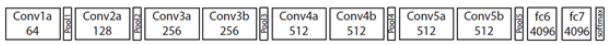

Fig. 3: C3D Network Architecture

C3D is a deep 3-dimensional convolutional neural network with a homogenous architecture containing 3*3*3 convolutional kernels followed by 2*2*2 pooling at each layer. The C3D architecture is shown in Fig.3

The C3D [9] model extracts both spatial and temporal components relating to motion of objects, human actions, human-scene or human-object interaction and appearance of those objects, humans and scenes. The features are further fed into a temporal network for action localization task, action recognition or action-word mining.

4. PROPOSED SOLUTION

4.1. Selection of Modules

Based on the algorithms and frameworks explored, the following are 3 proposed methods to implement a real-time video analysis system to detect if the user sits with proper/ improper posture:

1. Implement C3D to train a video classification model on the entire video.
2. Implement transfer learning to train a classification model on individual frames, and subsequently predict on the video sampled by frames.
3. Implement pre-trained facial detection, facial landmarks & OpenPose models, and create a rule-based system for posture detection on the video sampled by frames.

It is noted that a large and general dataset is paramount to training a deep learning model successfully. However, there is currently no available public video dataset for sitting office workers. Due to limited project time and human resources (with 2 indviduals on the team), it was difficult to create a sufficiently large training dataset. Therefore, with the insufficient data, it was found that the testing accuracies of the C3D and transfer learning model were unacceptable. Furthermore, the inference speed was insufficient to meet the requirement of a real-time detection system. Upon evaluating the trade off between prediction accuracy and processing time, the team decided to implement the pre-trained models and create a rule-based system to detect if users are in the proper posture. Subsequently, the system will provide corresponding alerts and recommendations based on the posture.

4.2. System Architecture

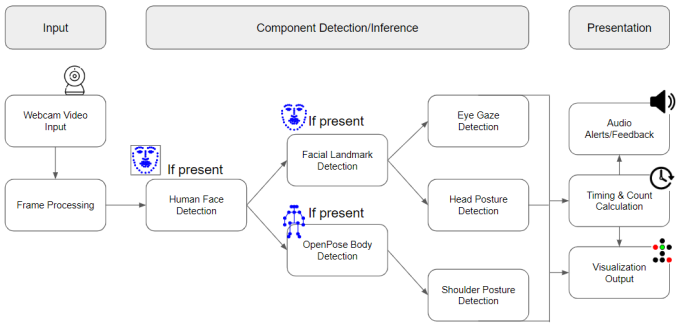

Fig. 4: System Architecture of Sit-Up-Straight posture detector

In Fig.4, the system architecture of the Sit-Up-Straight posture detector is shown. It has 9 modules of which the following 3 utilize pre-trained models for detection: Human Face, Facial Landmarks & OpenPose. (It was found that the landmarks for OpenPose were not granular enough for the requirements, hence Facial Landmarks were required). The following 6 are newly developed modules: Eye Gaze, Head Posture, Shoulder Posture, Audio Alerts, Timing & Count Calculation & Visualization Output. Table.1 shows the existing and newly developed models/modules.

Table 1: Existing & newly developed models/models

Model	Type of Model	Model Description
Human Face Detection	Existing open source model	Viola Jones algorithm to detect a human face in an image
Facial Landmarks Detection	Existing open source model	Takes in the bounding box which contains a face and outputs facial landmarks
OpenPose Body Detection	Existing open source model	Takes in an image and outputs body landmarks
Eye Gaze Detection	New developed module	Estimates if a user's eyes are looking at the screen or in another direction
Head Posture Detection	New developed module	Estimates if the user's head is tilted or too forward/backward
Shoulder Posture Detection	New developed module	Estimates if the user's shoulders are slanted
Audio Alerts/ Feedback	New developed module	Provides real-time audio alerts
Timing & Frequency Calculation	New developed module	Records the time the user spends in each improper posture and the overall time spend seated
Visualization Output	New developed module	To visualize key-points detected on the frame and in the form of a traffic light

5. WORKING DETAILS OF DETECTION MODULES

5.1. Head Posture Detection

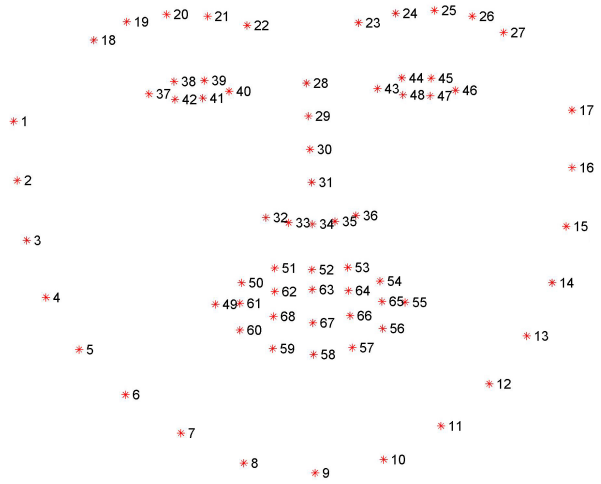

Fig. 5: 68 Facial Landmarks

The head posture and eye gaze detection relies heavily on the 68 landmarks (shown in Fig.5) from the pre-trained facial landmarks detection algorithm. Specifically it utilizes the landmarks for the T-zone (28), chin (9), left ear (1), right ear (17), left eye (37-42) and right eye (43-48).

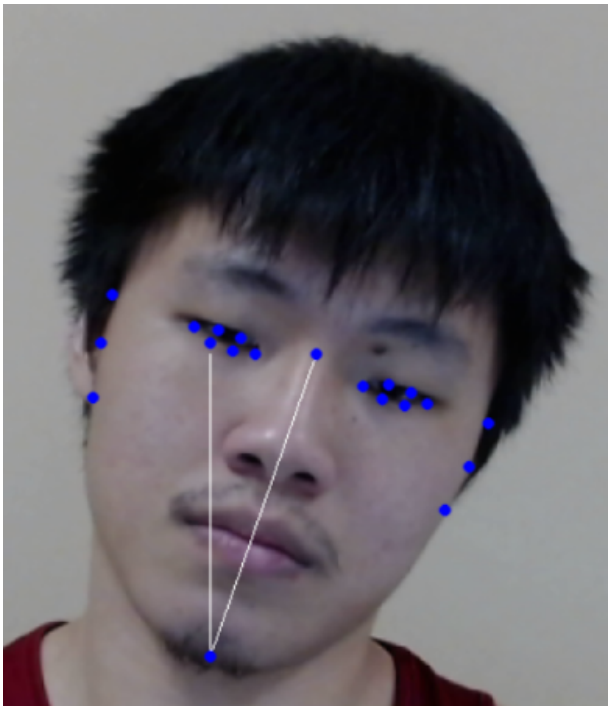

Fig. 6: Head tilted to the right

For head posture detection (in the left & right direction), the detection is dependent on the angle between the vertical and a line drawn from the chin (landmark 9) to the T-zone (landmark 28). The vertical line is drawn by extruding landmark 9 up in the y-axis, and the other line is drawn between the chin (landmark 9) and the T-zone (landmark 28). As the visualized white lines indicate in Fig.6, this enables us to determine the tilt of the face with respect to the vertical by using simple trigonometric formulas. If the face tilt angle is over a provided threshold, we specify that it is an improper posture.

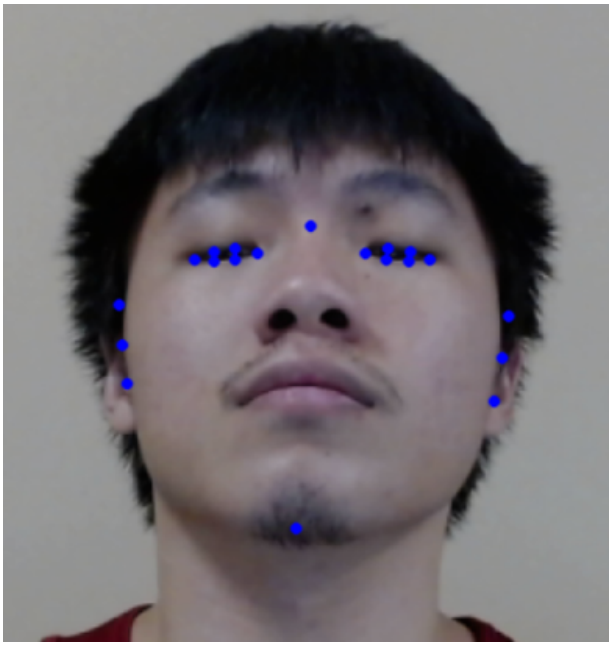

Fig. 7: Head tilted too far back

For head posture detection (in the front & back direction), we simply measure the difference in pixels between the T-zone (landmark 28) and the average position of the ears (landmark 1 & 17), as shown in Fig.7. If the difference in pixels is over a provided threshold, we specify that it is an improper posture. A possible further improvement would be to calculate the average angle forward and backwards between the landmarks.

5.2. Eye Gaze Detection

To achieve Eye Gaze Detection (as shown in Fig.8) we implement a simplistic model to estimate the direction the user is looking at. First, the left & right eye landmarks (37-42 & 43-48) are utilized to extract a greyscale image of the eye as shown in Fig.9.

Next, binary thresholding is done to extract a mask of the surrounding eyelids and another mask of the surrounding eyelids and iris, as shown in Fig.10 and Fig.11.

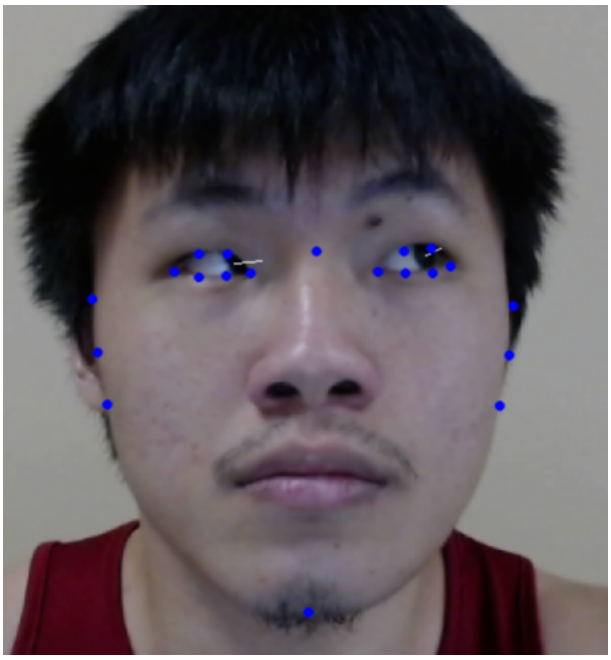

Fig. 8: Eye gaze visualized

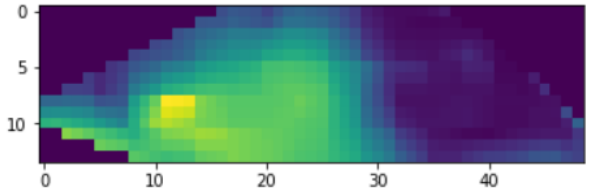

Fig. 9: Left eye (greyscale)

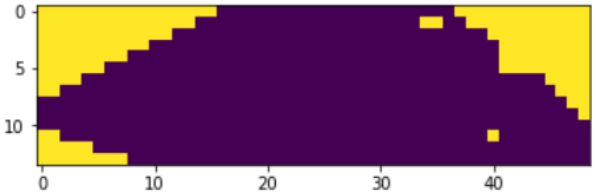

Fig. 10: Left eye threshold (eyelids)

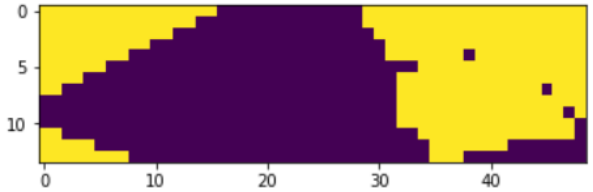

Fig. 11: Left eye threshold (eyelids & iris)

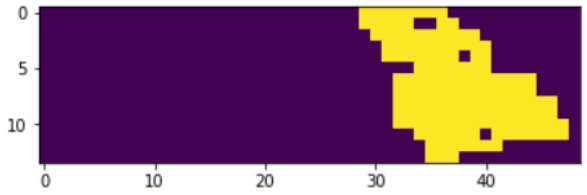

Fig. 12: Left eye threshold (iris)

Subsequently, as shown in Fig.12 the two masks are subtracted to extract only the pixels relating to the iris.

Lastly, the centre mass (CM) of the iris (in x,y) is calculated and compared with the CM of the eye (in x,y). If the CM of the iris deviates from the CM of the eye by more than a provided threshold, we specify that the user is not looking straight at the screen. As an added visualization, the direction the user is looking at will be added to the image using white lines.

5.3. Shoulder Posture Detection

Similar to the head posture and eye gaze detection, the shoulder posture detection relies heavily on the landmarks provided by the OpenPose Body detection. Specifically it utilizes the landmarks for the left shoulder (2) and right shoulder (5).

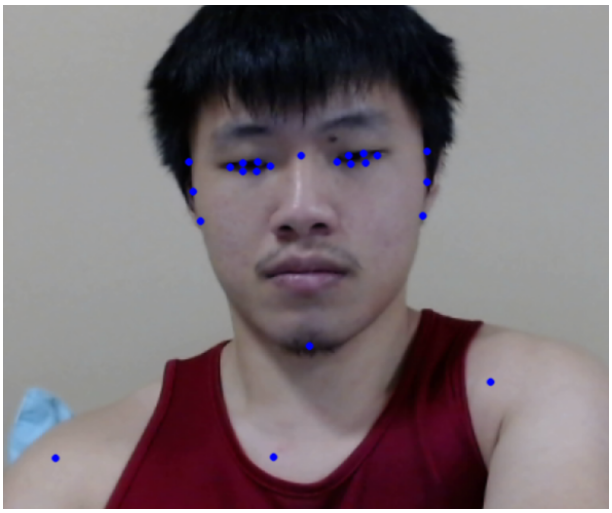

Fig. 13: Shoulder slanted to the left

For shoulder posture detection (as shown in Fig.13, the pixel difference between the left & right shoulder (landmarks 2 & 5) is calculated. If the difference in pixels is over a provided threshold, we specify that it is an improper posture.

6. DESIGN OF USER INTERFACE

6.1. Traffic Light Output

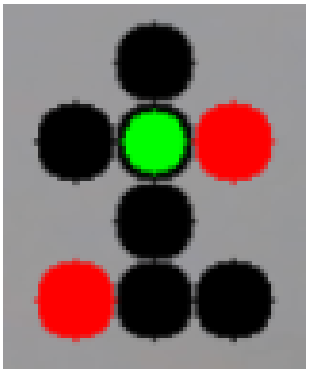

Fig. 14: Example of Traffic Light Visualization

The UI design presents the user with a simple but effective visual guide to improving his/her posture. The main visualization of the posture is output in the form of a traffic light as shown in Fig.14. In Traffic light output, the top cross (of 5 circles) relates to the head position, the bottom line (of 3 circles relates to the shoulders and the centre circle relates to the eye gaze.

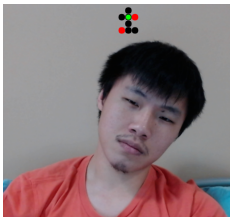

Fig. 15: Image Corresponding to Traffic Light

In this example from Fig.14, the user's head is tilted too much to the right, the shoulders are slanted to the left and the eye gaze is detected as forward, a full image of the user is shown below in Fig.15

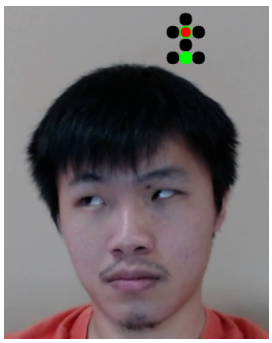

Fig. 16: Example of User Not Looking at Screen

Should the head position and should position be proper, the middle circle in the cross and line will be green respectively. However if the eye gaze is not towards the screen, the middle inner circle will be red. An example of the current situation is shown in Fig.16

6.2. Available Modes

Possible modes are shown on the screen at the bottom right. The visualization allows for the following modes, which can be activated by pressing the keys:

1. V - Verbose (plot the face/shoulder landmarks)
2. Q - Quiet (hide the face/shoulder landmarks; default)
3. S - Small (Minimize the display, showing only the traffic light)
4. E - Expand (Expand the display, showing the webcam output; default)
5. ESC - Exit (Exit the program)

Should the Verbose mode be used, the image facial landmarks and shoulder landmarks will be plotted on the output in blue, as shown in multiple figures above.

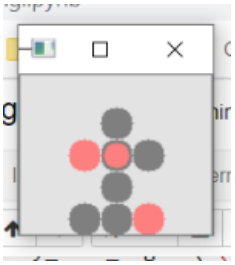

Fig. 17: Example of Improper Posture (Small Mode)

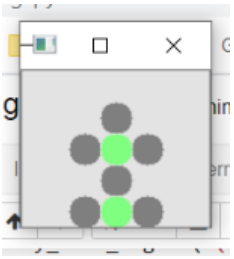

Fig. 18: Example of Proper Posture (Small Mode)

Should the Small mode be used, the visualization display will be minimized to only show the traffic light output. This mode is provided so that users can utilize the real-time posture detector whilst doing other tasks. Examples of the minimized output is shown in Fig.17 and Fig.18.

6.3. Timestamp & Comments

A timestamp and some hints are provided at the bottom left of the screen. This is to aid the user in understanding the traffic light visualization and to keep track of the time. An example is shown in Fig.19

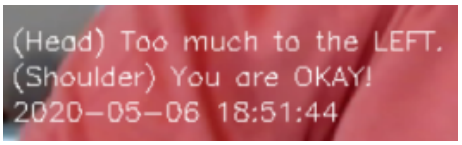

Fig. 19: Example of Timestamp & Hints

6.4. Alerts & Alarms

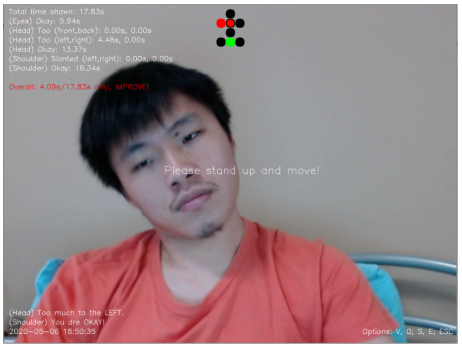

Fig. 20: Alert to get up

If the user spends longer than a pre-defined time in an improper posture, a short audio alarm will be played to remind the user to re-adjust his/her posture. Also if the user spends too long in the seated position, the posture detector will provide the user with the prompt “Please stand up and move!”. An example is shown in Fig.20

6.5. Exit Prompt & Evaluation

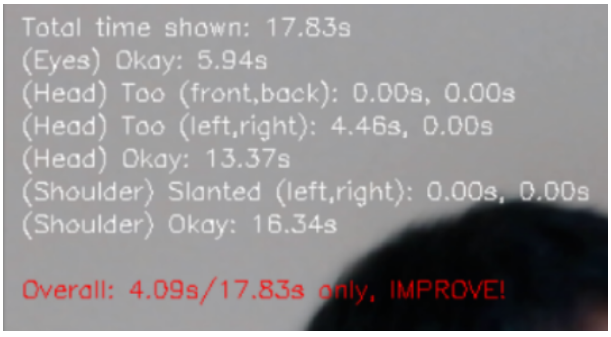

Fig. 21: Descriptive Information provided upon exit

Upon exiting the posture detector, some descriptive information will be provided such as the total amount of time shown and the time in each posture (eyes, head & shoulder). Finally an overall time will be calculated based on the duration spent in each posture.

7. EXPERIMENTAL TESTING ON DIFFERENT SCENARIOS

To ensure that the Sit-Up-Straight system can work well in different scenarios, the system was tested using various experimental setups. The results of AB testing are discussed below.

7.1. Different Background Experimental Setups

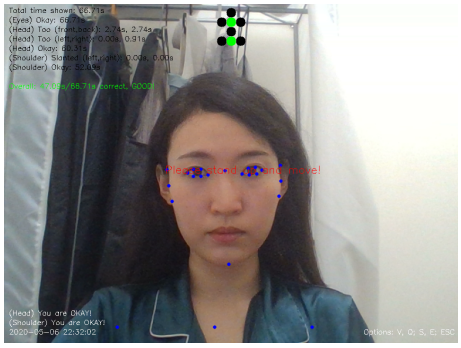

Fig. 22: Messy Background Setup

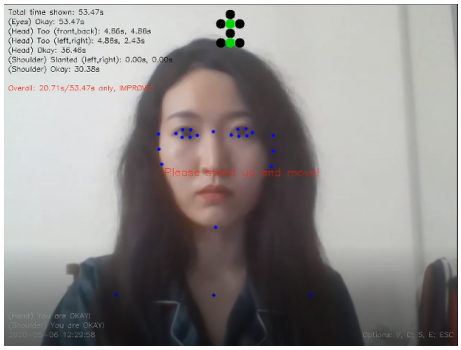

Fig. 23: Clear Background Setup

The Sit-Up-Straight system is able to work in both clear or messy background setups. Fig.22 represents the messy background while Fig.23 has a clearer background. In both background experimental setups, the system has managed to detect key-points, display visualization output as the traffic lights and provide the corresponding real-time feedback.

7.2. Different Illumination Experimental Setups

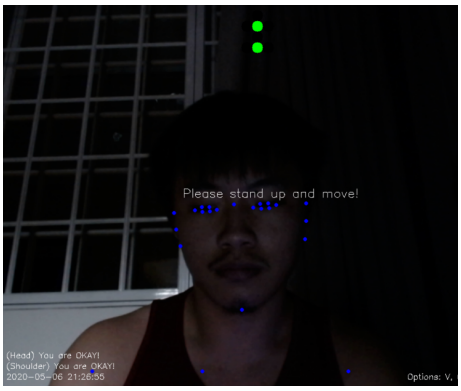

Fig. 24: Low Illumination Setup

Fig. 25: High Illumination Setup

Illumination is a key factor used to test computer vision systems. As such, the team has tested the Sit-Up-Straight system in different lighting conditions. It was found that whilst there is limited light captured via the laptop webcam, the natural/white/orange lights do not have much differences. In testing, the system works well in all three lighting conditions. Furthermore, the team also tested the low light condition as shown in Fig.24.

Compared with the white light condition (in Fig.25, the system is still able to detect key-points and provide real time feedback. The results above clearly show that the Sit-Up-Straight system can work well in different illumination conditions.

7.3. User Covers Ears & Shoulders with Hair

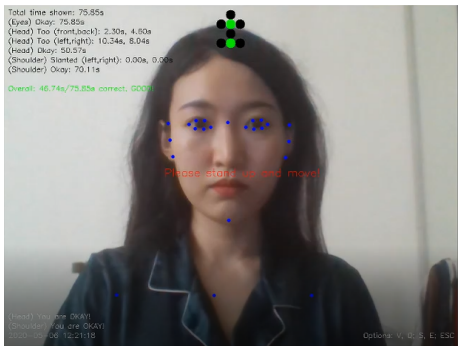

Fig. 26: User showing Ears & Shoulders uncovered

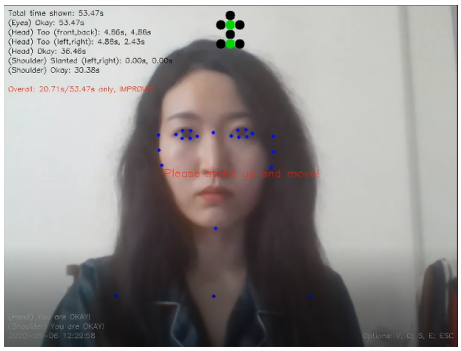

Fig. 27: User with Hair Covering Ears & Shoulders

AB testing results shown in Fig.26 and Fig.27 indicate that the Sit-Up-Straight system can work well even if some of the key points are covered or occluded by other objects. In Fig.27, the user's ears and shoulder are covered by hair. Desipte this, the Head Posture Detection and Shoulder Posture Detection modules can still correctly detect the key-points, which ensures that the system can correctly capture sitting posture of the user and provide real-time feedback.

7.4. User Wearing Glasses

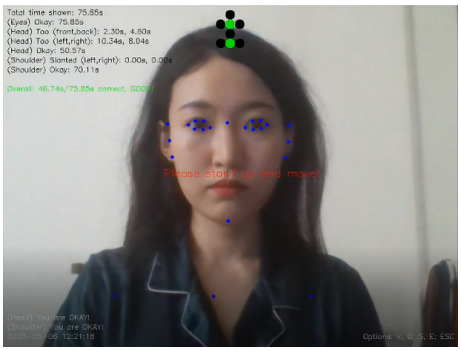

Fig. 28: User without Glasses

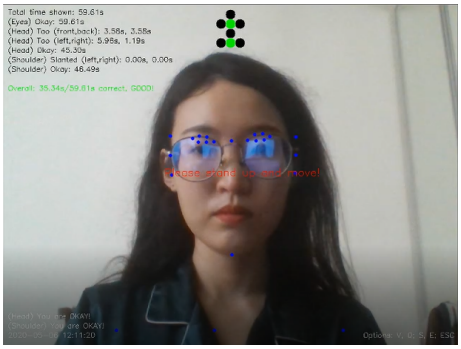

Fig. 29: User with Glasses

AB testing results shown in Fig.28 and Fig.29 indicate that the Sit-Up-Straight system can work well whether the user wears glasses or not.

8. CHALLENGES

Here are some challenges that were faced when developing the system:

- If the face is tilted by more than 30 degrees, the Viola Jones face detection algorithm fails and the facial landmarks are not computed.
- If some of the key points are out of webcam, the pretrained open source models will not be able to capture the required key points, which may affect the whole system.
- Due to the compute capability of some laptops, there might be a lag experienced when plotting the keypoints/ showing the visualization.
- The eye gaze detection is simplistic and might not work on all individuals.

9. CONCLUSION

In this project, the team successfully built a system to detect proper/improper seating postures using a common computer webcam. Tested on comodity laptops, it is able to provide real-time feedback (with alerts and recommendations) to users. The system implements open-source algorithms such as face detection, facial landmarks and OpenPose detection with rule based modules. Lastly, it has been shown to work well in various experimental environments.

10. APPENDIX

Link to code in Github: `https://github.com/davidygp/sit_up_straight`

Link to Experimental Setups: `https://drive.google.com/open?id=15mR1aJbMhohxLX8kDxy4uavPA-sdfjuq`

Link to Demo Video: `https://www.youtube.com/watch?v=mNELtDJexVE`

11. REFERENCES

[1] M Graf, U Guggenbühl, and H Krueger, "An assessment of seated activity and postures at five workplaces," *International Journal of Industrial Ergonomics*, vol. 15, no. 2, pp. 81–90, 1995.

[2] Justyna Drzal-Grabiec, Slawomir Snela, Justyna Rykala, Justyna Podgorska, and Maciej Rachwal, "Effects of the sitting position on the body posture of children aged 11 to 13 years," *Work*, vol. 51, no. 4, pp. 855–862, 2015.

[3] Yuri Kwon, Ji-Won Kim, Jae-Hoon Heo, Hyeong-Min Jeon, Eui-Bum Choi, and Gwang-Moon Eom, "The effect of sitting posture on the loads at cervico-thoracic and lumbosacral joints," *Technology and Health Care*, vol. 26, no. S1, pp. 409–418, 2018.

[4] celiktutan, Oya and Ulukaya, Sezer and Sankur, Bulent, "A comparative study of face landmarking techniques," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, pp. 13, 2013.

[5] Vahid Kazemi and Josephine Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1867–1874.

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.

[7] Shih-En Wei Yaser Sheikh Zhe Cao, Tomas Simon, "Realtime Multi-Person Pose Estimation," `https://github.com/ZheC/Realtime_Multi-Person_Pose_Estimation`, 2020, [Online; accessed 8-May-2020].

[8] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *arXiv preprint arXiv:1812.08008*, 2018.

[9] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.