PostureGuardian: the slouch detector

Views about Computer Vision and this course

As an aspiring machine learning enthusiast, I recall my first touch into the world of computer vision during an internship. It was a face recognition project that served as my initial glance into the realm of machine learning. Like many others, this marked the beginning of my machine learning journey. Back then, I had easy access to numerous tutorials on building image classification tasks, and achieving high accuracy results seemed within reach. At the time, my understanding of computer vision was limited to task-specific convolutional neural network (CNN) models and the constant stream of groundbreaking papers that achieved state-of-the-art performance. It seemed that computer vision was advancing at an astonishing rate, positioning itself as the most cutting-edge field within the realm of AI. Furthermore, the practical applications of computer vision are extensive and essential in today's business landscape. This consideration, coupled with my preference for hands-on learning over theory, gradually build up my decision to delve deeper into computer vision.

Prior to undertaking this course in computer vision, my exposure to the field had been primarily focused on CNN models. I had never delved into the fundamental techniques of computer vision. Concepts such as edge detection, component labeling, and lighting correction were strange to me, as I had taken a direct path to image classification models. However, as I embarked on this course, I soon discovered that computer vision was far more expansive than I had ever imagined. The course introduced me to these fundamental techniques, enabling me to comprehend the complete workflow involved in establishing a computer vision project.

Tasks such as edge detection, component labeling, and lighting correction proved to be invaluable in breaking down complex projects into manageable and solvable problems. I realized that these sub-tasks could be resolved without relying on CNNs, highlighting the significance of algorithms and theories in the field of computer vision. I delved deeper into these tasks, figuring out the underlying mathematics and algorithms that formed the basis of them. This learning experience not only broadened my understanding of computer vision but also provided a historical perspective on its development and sparked anticipation for future advancements.

Undoubtedly, the course proved instrumental in my growth as a machine learning practitioner. However, my thirst for knowledge remained the same, and I hope to explore state-of-the-art CNN models designed explicitly for image classification and recognition tasks. I desire a deeper understanding of the underlying techniques that allowed these models to outperform traditional approaches. By understanding the intricacies of advanced CNN architectures and immersing myself in cutting-edge research papers, I aim to grasp the latest breakthroughs and remain at the forefront of computer vision advancements. This pursuit would enable me to tackle increasingly complex tasks and stay abreast of the ever-evolving landscape of the field. Armed with an understanding of the underlying techniques that drive top-performing CV models, I aspired to contribute to the advancement of computer vision and pave the way for future innovations.

Project description

National Institutes of Health (NIH) reports that 80% of Americans experience posture-related problems. In fact even one of us in this project group is suffering from bad sitting postures over time. When people are sitting with the bad postures, there will be significant pressure on the spinal disks, low back, neck and a lot other body parts. These pressures will possibly lead to spinal misalignments, muscle imbalances, back and neck pain, and even impact on mood and energy over long time. According to biomechanics experts, these severe health issues are very common to observe from people who sit in a bad posture for extended periods.

Understanding the outcomes of sitting in bad postures could send a warning to some people, but it is difficult for them to keep in good postures over time while sitting. When someone is working with his computer, they are focusing on the work with more and more degrees over time, which makes them easily unaware of their sitting postures. Rounded upper backs, forward-rolled shoulders, and heads carried in front of the center of gravity are some common problems which are easily getting ignored during working. Without any direct notifications, people will still suffer from bad postures although they want to sit in correct postures intentionally. It is crucial to find a solution which helps people to self-monitor their sitting postures.

Our project is called “PostureGuardian”, which basically detects if someone is slouching. We empowers healthy posture by continously monitoring the user’s sitting posture in real-time using webcam. This involves detecting key postural cues as back alignment, head position and even lag posture. The system should be able to process the video stream and extract relevant information about the user's posture continuously. To let users know if they are sitting correctly without deconcentrating their focuses, we want to have a notification feature. Once the user's sitting posture is analyzed, the system should be able to provide immediate feedback to the user regarding the quality of their posture. This feedback can be in the form of visual indicators, audio prompts, or other cues that notify the user about any deviations from optimal posture. The feedback should be timely and actionable, enable the user to make necessary adjustments to improve their posture. In addition to immediate feedback, the system should guidelines and instructions on how to maintain good posture while sitting, in similar forms stated above. The guidelines should be informative and clear, helping the user understand the principles of good posture and the reasons why they are sitting in bad postures.

A key aspect of the system is its simplicity and accessibility. By relying solely on a webcam and built-in notification tools, the system eliminates the need for additional devices or complex setup procedures. This makes it convenient for users to adopt and use the system without any significant barriers or requirements beyond a standard webcam.

Overall, the system's goals align with promoting proper sitting posture, raising user awareness about their posture habits, and providing real-time feedback and guidance to help users maintain good posture while sitting. By leveraging computer vision and user-friendly interfaces, the system aims to be an accessible and effective tool for supporting healthy sitting habits.

Approaches and design

The slouch detector utilizes three major libraries and dependencies: OpenCV, mediapipe and Plyer. OpenCV is used for capturing video frames, image processing and drawing annotations on the image. In this project, OpenCV draws facial and body landmarks detected from the video frames and shows them on the monitoring window to help users understanding the underlying baseline of posture evaluation. Mediapipe provides pre-trained models for face mesh detection and pose estimation. It is a machine learning solution for high-fidelity body pose tracking, which is trained with input images and videos frames along with annotations of pose landmarks. This ML pose-tracking tool predicts 3D coordinates of positions of various body joints or landmarks, such as shoulders, elbows and knees. In this project, mediapose is solely used for detecting landmarks of shoulders and face. Plyer is used for displaying system notifications. It generates different notification messages using Python notification tool.

Generally, we implemented a real-time posture monitoring system using a webcam. It combines two models from the Mediapipe library: FaceMesh and Pose. The FaceMesh model is used to detect facial landmarks, while the Pose model is used to detect body landmarks, particularly the shoulders. The system continously captures frames from the webcam, processes them using the models, and calculates various posture-related metrics and angels.

Regarding our metrics of posture checking, we analyzed the user's posture by detecting face and shoulder landmarks using FaceMesh and Pose and performing several calculations:

1) Calculating the center of the face and the area of the convex hull of the face landmarks.

- The center of the face is calculated by scaling the first landmark pont (index 0 indicating nose) to the dimensions of the image.
- The convex hull is a polygon that encloses all the face landmarks, which includes the edge and corner of the face. The area of convex hull could represent the shape and size of the face.

2) Calculating the distance between the left and right shoulder landmarks.

- It measures the straight-line distance between two shoulder points and provides a base shoulder width for calculating relative distance later.

3) Calculating the relative distance between the face center and the shoulder width.

- This is computed by dividing the distance between the face center and the should-line by the shoulder width.
- This metric provides a measure of how far the face center is from the shoulder line. A higher relative distance indicates that the face is positioned further away from the shoulder, which could be an indicator of leaning forward or incorrect posture.

4) Calculating the angle of the shoulder with respect to the image frame.

- This is the angle between the line connecting the left and right shoulder landmarks and the horizontal line in the image frame.
- This metric provides information about the orientation of the shoulders relative to the image frame. A horizontal shoulder angle indicates that the shoulder are properly aligned, while a tilted angle suggests misalignment.

5) Calculating the face area with respect to the shoulder width.

- This is computed by dividing the face area by the shoulder width.
- This metric provides a measure of how close the person’s face is to the camera. A value out of the threshold suggests that the person’s face is too close or too far to the camera, indicating an incorrect posture, as the face should ideally be positioned at a reasonable distance from the shoulder.

6) Calculating the angle of the face with respect to the shoulder using the shoulder and face landmarks.

- Taking the shoulder and face landmarks as input, getting their difference in vector and calculating the angle between the vector and the horizontal plane.
- The angle provides information about how much the face is tilted or inclined relative to the shoulder. A correct posture would have a relatively small angle, indicating that the face is aligned properly.

7) Calculating shoulder z difference.

- This is computed by the difference in the z-coordinates of the left and right shoulder landmarks.

- This metric provides information about whether the person's shoulders are uneven or at different depths. A value exceeding the certain threshold may be an indicator of an incorrect or asymmetrical body position.

Based on the analyzed posture metrics and conditions, we implemented the features of providing immediate feedback to the user. It uses the Plyer library to display system notifications on the user's device, indicating whether the posture is good or if any posture issues are detected. We also included visual indicators on the video feed, such as colored circles or text, to represent the posture status in back stage for monitoring.

We utilized OpenCV to create a graphical user interface (GUI) that displays the webcam video feed, annotated with landmarks and posture feedback. It overlays text on the image to show the calculated metrics and posture status. The GUI continuously updates with each frame captured from the webcam.

To optimize performance, we set the resolution of the video capture to reduce processing time. We also added time gap for processing each input video frame to avoid overwhelming notification messages.

Result and analysis

The slouch detector system was tested by assuming the role of a user and setting threshould values for each metric based on the optimal sitting posture. It was evaluated by monitoring real-time postures through a displayed monitor window.

We started our testing by sitting in the correct posture. The monitor window displayed a green dot, indicating the posture was deemed correct by the system. However, there was a slight delay of several seconds before the system displayed a notification on the top of the screen, while the previous notification was indicating incorrect posture.

We tried various incorrect sitting postures to test the system's responsiveness and accuracy. These postures included lowering/raising/tilting head, shruging one of the shoulders, moving closer/farther to the camera and slouching. Each of these postures violated one or more of the predefined metrics, causing the monitor window to display a red dot, indicating an incorrect posture as expected. The system also successfully displayed notifications corresponding to the detected incorrect postures, with instructions on how to correct the specific posture deviation.

However, there were some issues existed, for example delay in notification. The purpose of introducing a time gap between notifications is to prevent users from being overwhelmed with constant alerts, but this turns out to make users receive feedback on their postures a few seconds after the actual posture change occurred. This delay can be misleading, as users may have already adjusted their posture by the time they receive the notification, therefore getting incorrect instructions.

Another issue is the inconsistency between different metrics, for example shoulder angle and relative distance metrics. There are instances where violations of the shoulder angle trigger notifications related to the relative distance metric, since that posture also violates the relative distance metric and its alerts come before shoulder angle. This can be explained by the reduced relative distance when someone is shrugging one of his shoulder to a large extent, as the midpoint of shoulder line moves up. Although the system is still giving notification of incorrect posture, this inconsistency can confuse users and compromise the reliability of the system, because users are receiving instructions on fixing slouching instead of uneven shoulders.

The system's sensitivity to minor posture changes is another limitation that affects its performance. Even small adjustments or no body movements can trigger notifications, creating unnecessary interruptions. This is caused by the predefined thresholds of each metric. It is difficult to determine the range of the threshold as well as imagining the real-time situation of user running this slouch detector. Sometimes the optimal posture may not indicate the exact median value of each metric, which causes users more likely to go beyond the upper or lower boundary of each metric when they are adjusting their postures. A larger range of the threshold may result in failure of detecting incorrect postures, while a smaller range will definetly amplify this sensitivity issue.

The system's dependence on the user's direction in relation to the camera introduces a constraint on the user's sitting position, giving the condition that threshold of metrics are predefined and unchangeable. Users must align themselves in a specific direction for accurate posture detection when they start running this detector. This dependency on camera angles limits the system's flexibility and usability, as users may need to adjust their sitting position based on the environment, chair or desk setup. Besides, this issue violates the goal of full-time back stage running of the monitor window, as users have to manually check and fix their starting postures.

In conclusion, the testing of the slouch detector yielded positive results. The system accurately identified incorrect postures based on the predefined metrics and provided users with visual cues through the monitor window. The notifications served as valuable reminders to maintain proper sitting posture and offered instructions on how to make necessary adjustments. However, there are areas that require improvement. Addressing the delay in notifications, resolving metrics inconsistencies, reducing sensitivity to minor posture changes, and enhancing adaptability to different sitting orientations will significantly improve the system's performance and user experience. By overcoming these limitations, the slouch detector can become a more reliable tool for promoting healthy sitting habits and overall well-being.

Remarks and future work

During implementing and testing the slouch detector system, several remarks and areas for improvement were identified. These observations provide valuable insights into the system's performance and serve as a basis for refining the system's functionality.

The slouch detector employed multiple metrics defined by ourselves, which considered various aspects of posture. This metric-based assessment provided a comprehensive evaluation of the user's sitting postion and offered a more holistic understanding of posture quality. Within what I can imagine of, the slouch detector is able to detect any kind of incorrect posture using some of its metrics. This exceeds our expectation and turns out to be a success in self-defined metric based assessment.

The system allowed customizable thresholds by setting threshold values for each metric based on users' preferred correct sitting postures. This customization feature enables users to adapt the system to their individual needs and preference, making it more personalized and flexible. Although calibration is required for each different user, they can decide how severe the slouch detector has to be by trying different threshold ranges, and find the one which suits them the best.

The delay in notification keeps users from receiving constant alerts, but may mislead them with incorrect instructions. To address this issue, a future work could be optimizing the slouch detector to provide more real-time feedback by reducing the delay or implementing a more adaptive notification mechanism. For example, the system only displays the notification when it makes sure that the user is sitting in a bad/good posture (probably this posture is lasting for a certain period of time). This would ensure that users receive timely and valuable feedback on their posture.

The inconsistency between different metrics is an issue regarding the algorithms and methods designing them. To resolve this, these algorithms and methods can be revised to become more comprehensive and have less interference with each other, but this may result in defining the metrics from scratch. Another solution could be fine-tuning the thresholds and conditions for each metric to help ensure that violations are correctly associated with the relevent metric, but this will spend large efforts for every possible user.

The system's sensitivity to minor posture changes negatively impacts its effectiveness. To address this issue, the slouch detector should be capable of filtering out insignifcant changes and focus on sustained or significant posture deviations. Implementing advanced filtering techniques, such as signal smoothing or adaptive thresholding, can help differentiate intential posture adjustments from minor movements, resulting in more reliable and meaningful notifications.

The system's dependence on the user's direction and environment dependency has similar reasons of its sensitivity of minor changes. To overcome this limitation, the slouch detector should be designed to accommodate various sitting orientations and adapt to different camera angles. Machine learning techniques, such as automatic calibration, can enable the system to adjust dynamically to different environment factors.

In conclusion, the slouch detector demonstrated both positive and negative attributes, but it performed to be effective on detecting bad postures in general. Future work should focus on optimizing the user calibration and metric-based assessment, making them automatic adjustable according to real-time video frames.

Course feedback and suggestions

This CV course offered a comprehensive coverage of a wide range of topics related to computer vision. It provided a solid foundation in the field, introducing basic concepts and techniques of different CV skills specifically.

The instructor demonstrated a strong understanding of the subject and communicated effectively when explaing complex concepts. His passion for computer vision and humorous teaching style brought energy to the topic which sounds boring. The lectures were well-structured and organized, presenting the material in a logical sequence, with difficulties gradually increased. The demonstrations and examples used in the slides were easy to understand, breaking down the complex topic.

The course included practical assignments that allowed us to apply the concepts learned in the lectures. These hand-on execises were valuable in gaining a deeper understanding of computer vision techniques and honing my practical skills. The lecture slides were well designed with specific knowledges covered in different topics which were used in completing the assignment. By working on these assignments, I had the opportunity to review lecture slides and absorb every detail which were easily overlooked during the lecture.

One of the most exciting aspects of this course is the opportunity to work on a real-lief project that encouraged brainstorming and the application of computer vision skills. The project may not require high-level CV skills or huge coding, but it challenged me to think creatively and see the practical application of CV in real-world scenarios. The open-ended nature of the project motivated me to explore various techniques and algorithms to better capture the human body landmarks and evaluate sitting postures. It was facinating to witness how my theorectical knowledge could be translated into a tangible solution which potentially achieved the goal of detecting slouching. On the other hand, I improved my problem-solving abilities and collaboration skills when working in a small group. This project was undoubtly a highlight of this course.

If there is any chance to improve this course, I would say making the assignment instructions clearer. Some of them were unclear and lack specific details in expected code/result, causing confusions. My points were deducted from a few assignments since my report didn't include all the requirements, but these requirements were not specified in the instructions either. Providing more explict guidelines and examples would help students complete assignments more effectively and accurately.

Detailed feedback on assignments would be highly appreciated, since it helps with understanding the strength and areas for improvents of my implementations. Although there is not much to comment for correct solutions, I would like to see suggestions of any alternatives or optimizations to better enhance my learning experience.

Besides, guest lectures or workshops by industry professionals working in computer vision would be helpful for us to understand real-world applications, industry trends and future developments in this field. Since this course is heavily based on concepts and algorithms, a little taste of real-world practices can be beneficial to absorb the knowledge.

Finally, I appreciate the opportunity to participate in this course. It was a helpful learning experience to expand my knowledge and skills in the field of computer vision.