Deep Learning Models for Seated Posture Detection from a User-Facing Webcam

Lindsey Bang, Ryan Wilson, Michelle Bolner, and Jack Galvin
{lindseyejbang, bolnerm, rjwilson, jgalvin}@berkeley.edu
UC Berkeley, April 2023

Abstract

Sitting for prolonged periods of time with bad posture can result in serious spinal and health complications. While certain chairs and cushions promote good posture, they are expensive and not easily transported. We present four machine learning models that are capable of detecting good and bad posture from a user-facing webcam. We show that using existing pose estimation models to extract body keypoints from single frames and feeding these latent representations through a fully-connected classification head produces more accurate results than fine-tuning existing image recognition models. We also show that three dimensional feature extraction is more accurate than two dimensional extraction. Two of these models have been quantized, deployed on an NVIDIA Jetson Xavier device, and include a user-facing chime to correct sustained (60s) bad posture.

Introduction

Bad posture for extended periods of time can result in major back problems [1, 2]. Many organizations implemented work-from-home policies during Covid-19. People who were not accustomed to sitting in front of a computer screen for a prolonged period of time suddenly found themselves doing so. In fact, a recent study shows that 48% of participants worked from home sometime during the pandemic and 33% exclusively worked from home[3]. Before that, only 5% regularly "telecommuted" [3]. As a result, good posture becomes an important part of preventing serious spinal and health complications.

While sitting on medicine balls and using special cushions on high-end office chairs are viable ways to encourage good posture, such materials can be very expensive. Recent work in the computer vision community has produced machine learning models that use cameras to detect bad posture and notify individuals to correct it [4]. More specifically, Kapoor et. al. run single frames from a livestream video through a pose estimation model (BlazePose) and feed these latent representations into a deep neural network to classify bad posture. However, no existing models do so from a forward-facing view (i.e., they detect bad posture from the side using spine angle). Such a setup can be cumbersome and is not easily portable.

Several human pose estimation models already exist and provide a good start for extraction of body keypoints. MoveNet, which locates (x, y) coordinates of 17 body parts (Figure 1), consists of a MobileNetV2 feature extractor and four prediction heads. It was trained on the COCO dataset and an internal Google dataset called Active. The model comes in two separate versions (Lightning and Thunder) and each runs at over 30 frames per second. The MoveNet API is available through Tensorflow.js and as a TFLite model from Tensorflow Hub.

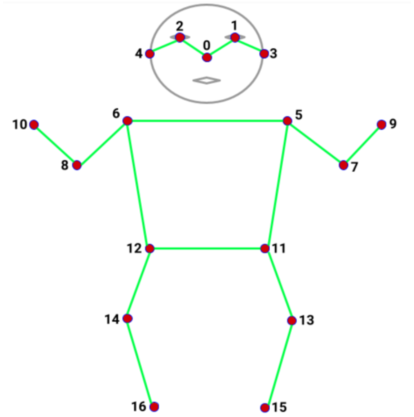

Figure 1   Keypoints extracted by MoveNet and PoseNet

PoseNet is a convolutional neural network built for real-time human pose detection. It is a modified version of GoogLeNet and has 23 layers[5]. More specifically, the 3 softmax layers in GoogLeNet were replaced with affine regressors to output a 7-dimensional vector representing position and orientation. Like MoveNet, PoseNet extracts the coordinates of 17 body keypoints detailed in Figure 1.

Perhaps one of the most popular and widely-used real-time pose detection models is MediaPipe Pose. Unlike MoveNet and PoseNet, MediaPipe Pose extracts 3-dimensional (x, y, z) coordinates of 33 keypoints with visibility scores, indicating wheather the point is visible or hidden on a frame. MediaPipe Pose, like MoveNet and PoseNet, extracts keypoints for just a single person, but comes with an optional segmentation mask that can be used to separate the human from the background. However, MediaPipe offers a face geometry solution unlike other models.

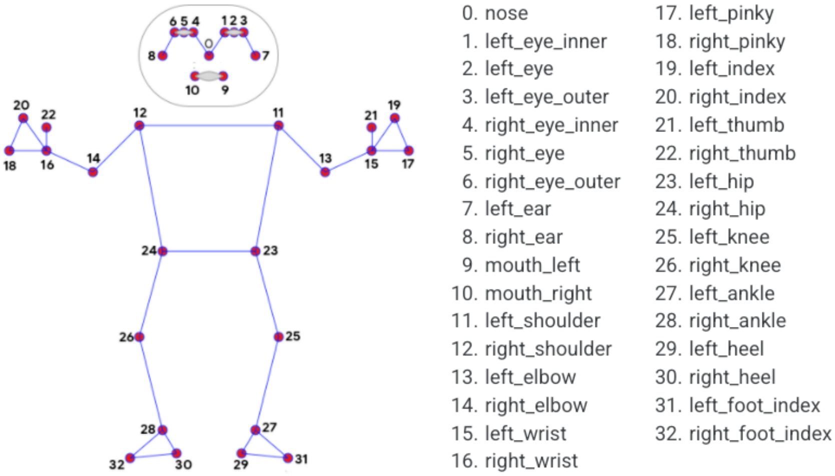

Figure 2   Keypoints extracted by MediaPipe Pose

Finally, even traditional computer vision models like VGG-19 and YOLOv7 have been adapted for pose detection. As its name suggests, VGG 19 is a deep convolutional neural network with 19 layers and was trained on over one million images from ImageNet[6]. YOLOv7 has been adapted specifically for pose detection, and, unlike all other models mentioned previously, works for multi-person pose detection[7].

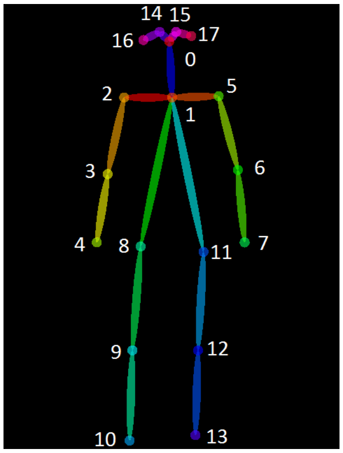

Figure 3   Keypoints extracted by YOLOv7 Pose

Each of the aforementioned pose estimation models has its own set of benefits and limitations. Despite these, each can be used to extract keypoint coordinates which can be fed to a custom classification network aimed at deciphering good posture from bad.

Data

Our dataset consists of 3,473 training, 682 validation, and 869 test images collected by the authors of six unique individuals. We used the user-facing webcam from our own laptops to collect video of ourselves (upper torso) in good and bad postures in a variety of environments and split these videos into frames using ffmpeg. The images are split into two classes - "looks good" and "sit up straight" - and were labeled by the authors. Despite efforts to create additional, more specific classes under the umbrella of "bad posture," we opted for a simpler implementation due to time constraints associated with the additional amount of data that would have to be gathered and labeled.

In general, "looks good" posture is classified as (1) not leaning toward the screen or to one side, (2) shoulders pulled back, and (3) neck is relatively straight. Images labeled as "sit up straight" are anything but what is described above and often include hunched necks and backs. See Figure 4 below for examples.

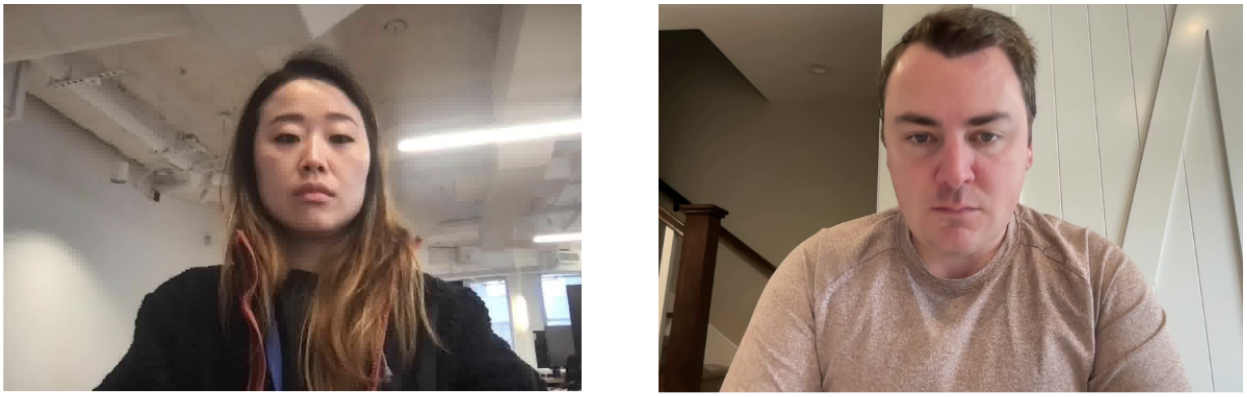

Figure 4   Examples of good (left) and poor posture (right)

Experiments

We carried out the below experiments and used accuracy on the test set as our metric for success.

<u>MoveNet</u>

We first validated that extraction of keypoints is actually helpful for this task. To do so, we used transfer learning to fine-tune ResNet50 on our training set, thus treating this as a pure image recognition task with no regard for extraction of body keypoints. We built a custom classification head on top of ResNet50 with 4 fully connected layers of dimensions 512, 256, 128, and 64, each followed by batch normalization and dropout layers. Validation accuracy and loss were highly variable and final validation accuracy was just below 0.6, which is much lower than validation accuracy (0.84) using the architecture described below.

We ran each image in the dataset through MoveNet. Since MoveNet returns raw (x, y) coordinates of each keypoint, the output is not standardized in a coordinate system and is not robust to orientation changes. We normalized the output of MoveNet according to the steps in Kapoor et. al. Briefly, we restricted the output to just the keypoints of the upper torso (points 0 to 6 in Figure 1) and subtracted the average of the shoulder coordinates from all others. This removed translational variations and moved the center of the pose closer to the origin of the cartesian coordinate system. Next, we calculated a scaling factor by finding the maximum distance between the shoulder average and any other keypoint. Each keypoint was divided by this scaling factor, which brought all coordinate values in the range of [0, 1].

After normalizing the keypoints returned by MoveNet on each image, we fed these latent representations into a neural network with 3 fully connected layers of dimensions 512, 256, and 128, each followed by a dropout layer. We used Adam as our optimizer to minimize sparse categorical cross entropy loss, a learning rate of 1e-4, a batch size of 16, and trained for 15 epochs.

MediaPipe Pose

MediaPipe offers both landmark coordinates and a facemesh. We collected 2004 coordinates from each image; x, y, z and v with 501 keypoints (468 from facemesh and 33 from keypoint landmarks). Since MediaPipe outputs normalized values, we did not need to take into account the same preprocessing steps as we did with MoveNet and YOLOv7 before feeding these coordinates into the model.

After using MediaPipe to extract the coordinates from each image, we fed these representations into a neural network with 3 fully connected layers of dimensions 512, 256, and 128, each followed by a dropout layer. We used Adam as our optimizer to minimize sparse categorical cross entropy loss, a learning rate of 1e-4, a batch size of 16, and trained for 100 epochs.

YOLOv7

We performed the exact same preprocessing steps here as we did for MoveNet, but used a learning rate of 1e-3, binary cross entropy loss, and trained for 100 epochs.

VGG-19

VGG19 is not a pose estimation model and does not output keypoints. VGG19 is an image classification model. Accordingly, we used data generators to transform images into tensors, applied random transformations to each image, and fed the transformed image tensors into the same classification network described above.

Since VGG19 is 19 layers deep, we used early stopping callback in order to save training time. The model completed learning after 11 epochs.

Results

Movenet

We achieved final training and validation accuracies of 0.86 and 0.84, respectively. The loss and accuracy curves can be seen in Figure 5 below.

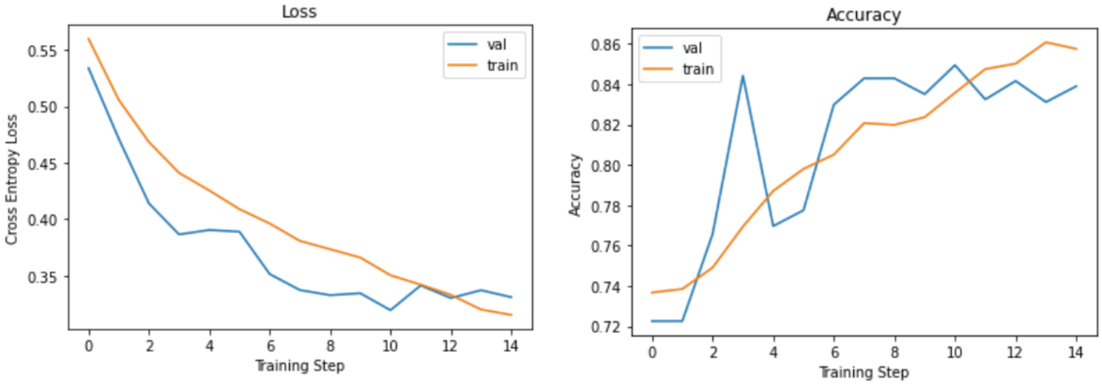

Figure 5   Loss and accuracy curves for MoveNet. The model achieved final training and validation accuracies of 0.86 and 0.84, respectively.

MoveNet achieves an accuracy of 0.79 on the test set. Additionally, we determined a decision threshold of 0.31 to optimize F-score, but ultimately abandoned this in an effort to optimize precision over recall for the reasons described in the Conclusions. This model was quantized to FP-16 and achieved the same accuracy after quantization (Figure 9).

This model was containerized and deployed to an NVIDIA Jetson Xavier. Both MoveNet and our trained classification head were wrapped in a cv2 capture loop, which plays a chime to correct bad posture if such posture is measured consistently (frame-by-frame) for 60s.

__MediaPipe Pose__

We achieved an accuracy of 0.87 on both the training and validation datasets. Additionally, loss steadily decreases and accuracy steadily increases on both the training and validation sets.

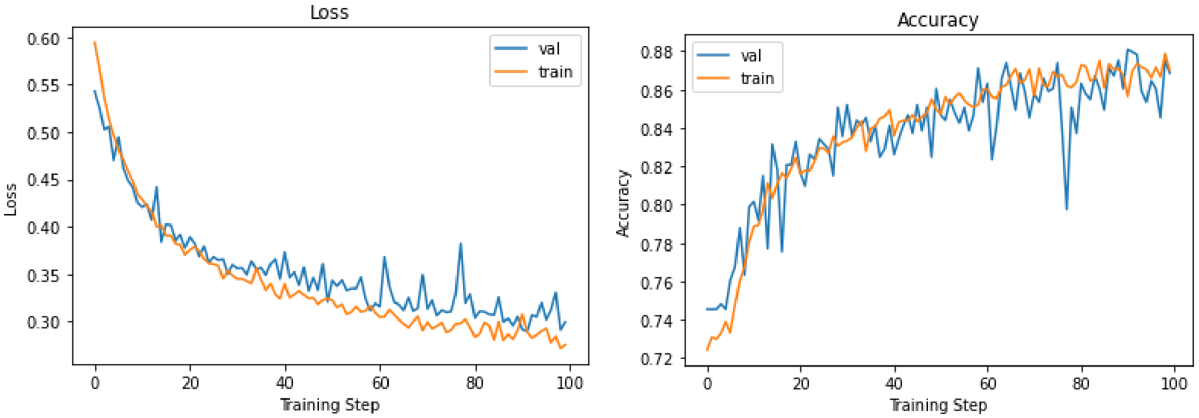

Figure 6   Loss and accuracy curves for MediaPipe

An accuracy on the test dataset was 0.86 which was the higher than any other models we have experimented. In addition, f1 score for the “looks good” posture was 0.75 and the “sit up straight” posture was 0.91, which make sense since we have 2.7x more “sit up straight” images compared to the “looks good” images.

YOLOv7

Despite extensive preprocessing, this model appeared to overfit. Training accuracy and loss were very strong, but the model did not learn to generalize these patterns to the validation set, as can be seen in Figure 7 below. Ultimately we decided to abandon optimization of this model in favor of stronger performance by MoveNet and MediaPipe Pose.

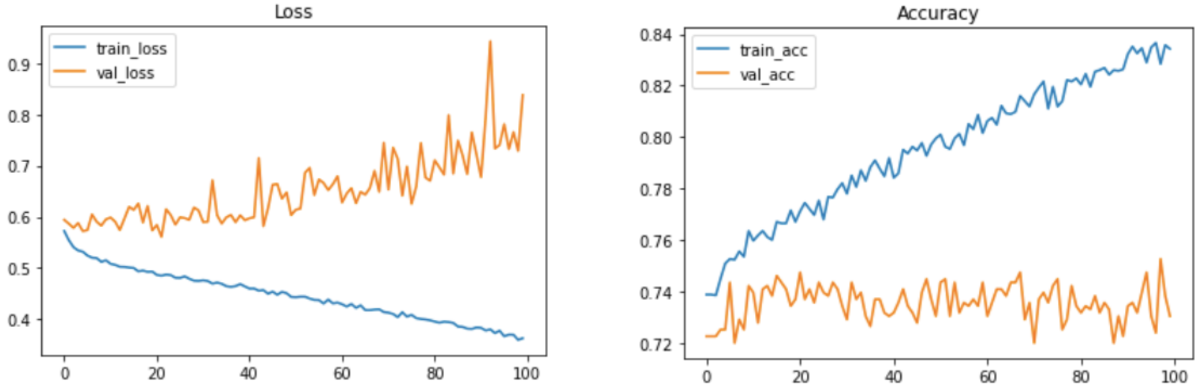

Figure 7 Loss and accuracy curves for YOLOv7 Pose

VGG-19

This model stopped training after 11 epochs since we implemented early stopping. Although VGG19 does not use keypoints, its accuracy on the test set is the second highest of all four models we trained.

We achieved a final training accuracy of 0.92 and validation accuracy of 0.88. The loss and accuracy curves can be seen in Figure 8 below.

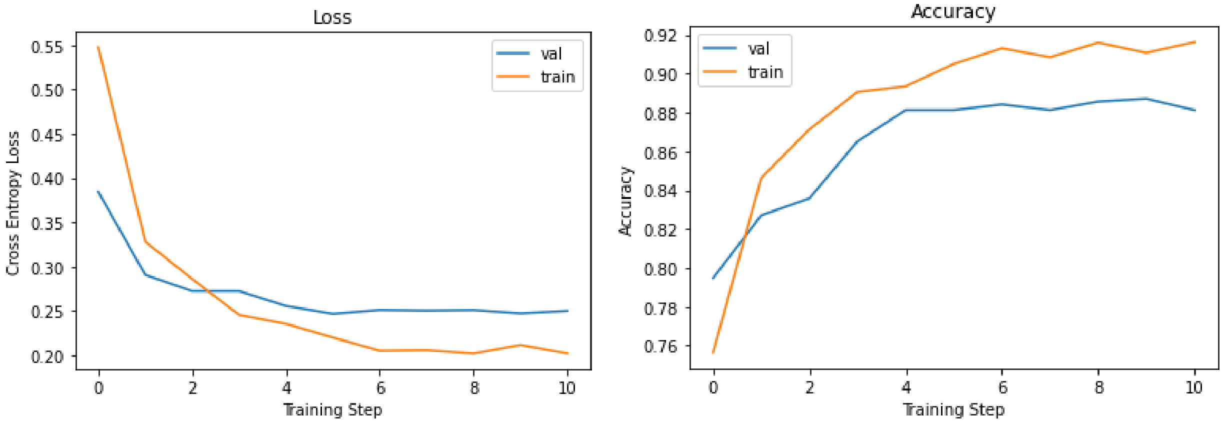

Figure 8 Loss and accuracy curves for VGG-19

VGG19 achieves an accuracy of 0.84 on the test set.

	Accuracy	Size	Compressed Size	Accuracy of Compressed
MoveNet	0.79	2.3 MiB	355 KiB	0.79
MediaPipe	0.86	147 KiB	–	–
YOLOv7	0.73	706 KiB	–	–
VGG-19	0.84	133 M	48.8MiB	0.84

Figure 9   Accuracies on the test set for each model before and after compression.

Conclusions

Based on the above experiments, we conclude that prediction of good or bad posture from a frontal facing camera benefits from the extraction of body keypoints. This can be seen in the large difference in accuracy between our MoveNet model and a fine-tuned ResNet50 with a custom classification head.

Furthermore, extraction of keypoints in three dimensions produces more accurate results compared to two-dimensional extraction, as can be seen in the differences between metrics from MediaPipe Pose (3-dimensional) and MoveNet (2-dimensional). We conclude that MediaPipe Pose is naturally more appropriate for this task as it is inherently three dimensional.

We faced a number of challenges conducting the above experiments. Perhaps the largest challenge was consistency in labeling images. We originally sought to divide the dataset into five classes, but found that subtle differences in poses and our interpretations therein produced inconsistent results. Collapsing the dataset into two classes significantly improved results.

Another challenge was the diversity of people within our images. We only have six unique individuals within our dataset, which is imbalanced both in terms of the distribution of images by individual and by class. If given more time, we would label additional images of more people and balance the dataset by class. We even experimented with changing the decision threshold post-training in an effort to rectify this limitation (i.e., optimizing for precision), but found that doing so resulted in measurements that would “flicker,” which would allow someone to stay in bad posture indefinitely without ever receiving an alert.

References

1. Wilhelmina E Hoogendoorn, Paulien M Bongers, Henrica CW De Vet, Marjolein Douwes, Bart W Koes, Mathilde C Miedema, Geertje AM Ariëns, and Lex M Bouter. 2000. Flexion and rotation of the trunk and lifting at work are risk factors for low back pain: results of a prospective cohort study. Spine 25, 23 (2000), 3087–3092.

2. Joshua Scott Will, David C Bury, and John A Miller. 2018. Mechanical low back pain. American family physician 98, 7 (2018), 421–428.
3. European Foundation for the Improvement of Living, Working Conditions, et al. 2020. Living, Working and COVID-19—First Findings—April 2020. (2020).
4. Rithik Kapoor, Ashish Jaiswal, and Fillia Makedon. 2022. Light-weight seated posture guidance system with machine learning and computer vision. PETRA '22: Proceedings of the 15th International Conference on PErvasive Technologies Related to Assistive Environments, 595–600.
5. Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: Convolutional networks for real-time 6-dof camera relocalization. In Intl. Conf. on Computer Vision (ICCV), 2015.
6. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations (ICLR 2015), 1–14.
7. Debapriya Maji, Soyeb Nagori, Manu Mathew, and Deepak Poddar. 2022. YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2637–2646.
8. Bansal, M., Kumar, M., Sachdeva, M. et al. Transfer learning for image classification using VGG19: Caltech-101 image data set. J Ambient Intell Human Comput 14, 3609–3620 (2023). https://doi.org/10.1007/s12652-021-03488-z