# Human Learning from Demonstrations through Real-Time Segmentation and Pose Estimation

**Andrei Spiride**
spiridea@mit.edu
MIT

**Andrew Jenkins**
awj@mit.edu
MIT

## Abstract

This study introduces an innovative approach to human learning from demonstrations, leveraging real-time segmentation and pose estimation models. We aim to develop a virtual personal trainer application that provides feedback on workout techniques, potentially replacing the need for expensive personal trainers and increasing accessibility to correct exercise guidance. Our system utilizes pose segmentation models to count exercise repetitions and encourage full-range motion in various exercises. We explore the application of advanced neural networks, specifically focusing on pose estimation through convolutional neural networks (CNN) and segmentation techniques. The application is built on the Mediapipe model, incorporating a scoring system to assess pose accuracy and an intuitive user interface. The core of our system lies in its ability to generate a mask of the user's location within an image, providing easy-to-follow feedback for exercise adjustments. This paper details the development process, including the methodology, application flow, model efficiency, and the user interface's design. We also present our findings on pose-matching accuracy and user interface responsiveness. The study concludes with an evaluation of the system's effectiveness as a virtual personal trainer and discusses limitations and potential future enhancements to improve its capability and performance.

## 1 Introduction

Technology is commonly created and used with the intent of improving our day-to-day lives in unique ways. At first, these pieces of technology might be slightly cumbersome, or inconvenient to use, however over time, their integration into our daily lives improves. In this investigation, we explore the use of pose segmentation models in making a virtual personal trainer, that can count reps, and encourage full range of motion on a variety of different exercises.

Currently, personal trainers are expensive to hire. Creating a free piece of software that helps perform some of the same jobs as personal trainers could increase the accessibility of feedback on workout techniques and prevent injuries associated with incorrect form. Having a piece of software that runs on a user's phone, or tablet, that they can take with them to the gym, or use at home would be very practical to use. Currently, there are several pieces of software that aim to use 2d and 3d pose estimation in order to provide users feedback on their exercises. However, these apps often require precise setup of cameras in relation to where a user is performing the exercise, and produce feedback (poses) that can be difficult for people to understand and incorporate into adjusting their poses. Some of these applications also don't work in real time[2][6]. Instead, we focus on using pose segmentation- a technique in which we can create a mask of the user's location within an image. These images provide feedback (masks) that is easy to adjust pose off of- a user just needs to try to occupy the same space as the mask in order to achieve the correct pose. Below, we provide some context on how these models are typically built and applied, in various domains. Such a tool would be

Preprint. Under review.

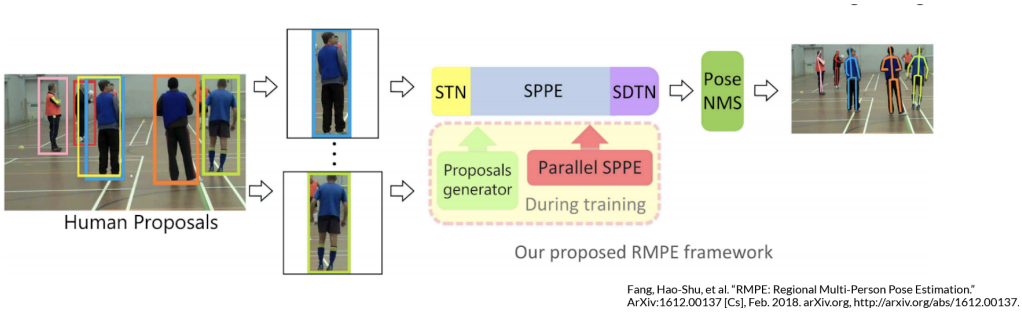


Figure 1: Alpha Pose Methodology

useful for people who are learning new exercises or people doing Physical Therapy, which involves a lot of exercises that are static- you hold a position for prescribed amounts of time.

## 2 Related Work

### 2.1 Pose Estimation

There is significant previous research using neural networks for pose estimation. Many networks have been created and tested against data sets such as MPII[3], COCO[5], or PoseTrack[1]. These typically use CNN-based networks to output heatmaps for every key point within an image. More complex models have been developed that add to this architecture. Notably, Alpha Pose is a CNN based architecture that uses a Spatial Transformer Network to increase prediction accuracy[4] 1. There are many competing networks that perform very well on images/videos and can predict poses in real time. Common pose estimation networks include PoseNet, developed by Google, and DeepCut.

Top-down approaches, in which bounding boxes are first drawn around people in an image, rely on Single-Person Pose Estimation (SPPE) to make predictions. SPPE is usually done with CNN based networks and identify the most likely locations key points will appear in. Bounding boxes isolate individuals in an image and simplify the task of drawing a skeleton that matches the person's pose. Because of its relative simplicity compared to a bottom-up approach, a top-down approach is used in this investigation.

Specifically, we use Mediapipe, a library that uses a top-down approach to first locate a region-of-interest (ROI) within the frame. The underlying model within Mediapipe is called BlazePoze. It has 3 versions, each of different sizes, for different use cases. Once the ROI is identified, a neural network predicts pose and a segmentation mask within that ROI. For video inference, the ROI does not change until a frame is reached in which the network cannot accurately identify body presence. Then the ROI is re-derived. Mediapipe has been shown to be competitive with other SOTA pose estimation networks, such as Alpha Pose 2.

### 2.2 Human Segmentation

Image segmentation is a separate, but related problem for our use-case. Typically, image segmentation networks focus on segmenting all aspects of an image. In our case, we have a focused person-segmentation network. Mediapipe conveniently has an optional mask that can be output from the pose-prediction network. The original image is turned into a mask that outputs values from 0-1 which reflects the probability of a pixel being part of a human or not being part of a human. Segmentation networks such as these are often used to identify the locations of various objects within scenes and identify areas of interest in medical applications on CT scans, X-ray scans, etc.

For our purposes, we focus on using Mediapipe, which is built on the BlazePose network, and has 3 main outputs. First, the locations of keypoints in the poses of people in the image (a collection of 2-d coordinates in the image). Second, a normalized 3-d pose calculated by estimating camera location and performing a constrained optimization to calculate the most likely person orientation in 3d space of the 2d pose spit out by the model. Finally the model outputs a mask outlining the most likely

| Method | Yoga mAP | Yoga PCK@0.2 | Dance mAP | Dance PCK@0.2 | HIIT mAP | HIIT PCK@0.2 |
|---|---|---|---|---|---|---|
| BlazePose GHUM Heavy | 68.1 | **96.4** | 73.0 | **97.2** | 74.0 | **97.5** |
| BlazePose GHUM Full | 62.6 | **95.5** | 67.4 | **96.3** | 68.0 | **95.7** |
| BlazePose GHUM Lite | 45.0 | **90.2** | 53.6 | **92.5** | 53.8 | **93.5** |
| AlphaPose ResNet50 | 63.4 | **96.0** | 57.8 | **95.5** | 63.4 | **96.0** |
| Apple Vision | 32.8 | **82.7** | 36.4 | **91.4** | 44.5 | **88.6** |

Figure 2: BlazePose Benchmarks

position of the subject in the image. Because of the limitations of using the 2d or 3d pose outlined in the beginning of the introduction, in order to guide a user to match poses, we use the mask.

## 3 Method

Below, we describe each piece of our system as well as its function as a whole. It is composed of the Mediapipe model, which functions as the application backbone, a scoring system to decide if a user is accurately matching the desired pose, and the user-interface.

### 3.1 Model and Efficiency

For pose and segmentation information extraction, our application employs Google's Pose Landmarker model, leveraging its advanced convolutional neural network architecture, which is akin to MobileNetV2 [7]. This model is specifically engineered for efficient performance in mobile-device environments, making it an ideal choice for our application's requirements. We opted for the *Heavy* variant of the model which is approximately 26 MB in size and is advertised to operate at around 4 frames per second (FPS).

In practical application scenarios, our system achieved a slightly lower performance, averaging about 3.5 FPS. We hypothesize that this minor reduction in speed could be attributed to additional processing overhead introduced by our application's functionalities. Another possible factor is that we used an inference mode designed for processing singular images, as opposed to the streaming mode, which is tailored for handling continuous data streams. This discrepancy in operational modes could account for the observed performance variance.

The Pose Landmarker model processes inputs of 256x256x3 RGB images, delivering precise outputs that include 33 body landmark locations along with a detailed human-body segmentation mask. Despite the slight deviation from the advertised speed, the model's performance remains highly effective for our application's purposes, balancing accuracy and efficiency in a mobile environment.

### 3.2 Application Flow

The flow of the user-interface was designed to be as simple as possible. There are 2 main stages to the application. First, a user can create a new exercise. This exercise creation screen allows the user to define an exercise as 2 static poses- one that represents the top of the exercise and one that represents the bottom of the exercise. Once the user collects these two poses, the exercise is saved under a name and can be accessed on the other screen. When a user wants to do an exercise, they simply load up the exercise they want to do by name, and a "shadow" of the target pose for the exercise is overlayed over the current camera feed.

Once the user hits the target pose, the screen flashes green, and it alternates to the other side of the exercise- if you hit the bottom of the exercise, it switches the goal to the top, and vice versa. Using

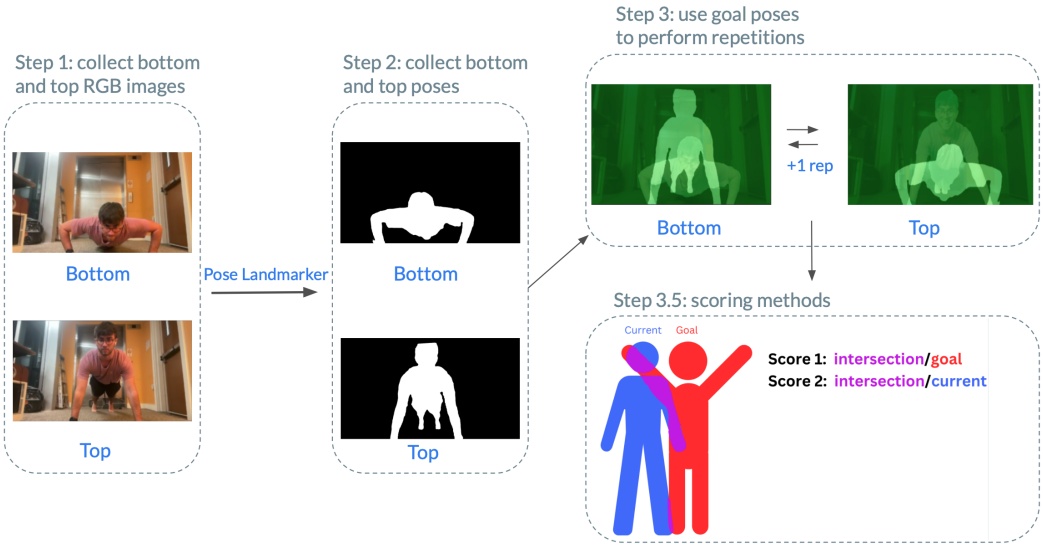


Figure 3: Application Flowchart

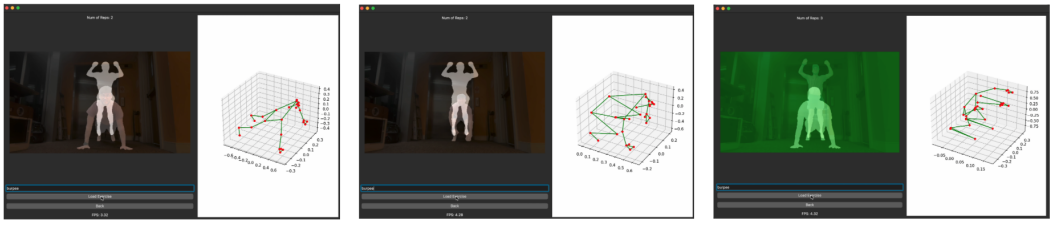

Figure 4: Burpee Demo

this method, it's very easy to count reps for the user. This user-interface gives the user constant, simple feedback while they do their exercises. It's simple for them to tell what part of their body is not correctly aligned with the target pose. Compared to showing a target 2d or even 3d pose, the pose mask is easier to use 3.

### 3.3 Human Pose Segmentation

The image above shows the final user interface that the user sees. 3 screenshots taken during the burpee exercise are shown. On the left of the user interface, the user sees their current image being streamed from a camera. It has an overlay that shows the "shadow" of the target pose the user must achieve in order to advance through the exercise. This is the main use case of the masks showing person-segmentation.

We included the 3d pose estimates on the right because they allow the user to observe their pose from any angle. It is a feature that currently adds no functional value, however, it is a cool illustration of how powerful these tools can be 4. Again, the limitations of asking the user to assume a certain pose by providing them with the 2d and 3d pose estimates are evident- first, it is hard to match up your current pose annotations to a target pose annotation, especially for the case where the poses are in 3d. The mask is much more intuitive when it comes to allowing the user to make adjustments to their pose to hit their goals. Additionally, with 3d poses, a user would have to precisely match camera positioning for all reps of an activity. Using masks mitigates this issue.

Figure 5: Pose Matching

### 3.4 Pose-Matching

Being able to tell whether a user has successfully matched a pose is a relatively simple problem, however it requires a clever trick. The most naive approach is to look at the user's current mask, shown in blue in Step 3.5 of Figure 3. If the blue mask covers the entirety of the red mask, that would most likely mean the user is covering the target pose. However, that leads to an error in the edge case in which the user covers the entire target, but also covers other areas. An easy way to visualize this case is if the user is much closer to the camera than when the target pose was taken. As shown in Figure 5, we see an example of this edge case. The user is covering the entire target, but the user's arms are outside of the target.

To combat this, we introduce another score. This score is the ratio of the intersection to the total mask of our current position. In the edge case example shown above, this score will fall beneath the threshold because the arms are not covered by the intersection between the current pose and the target pose.

## 4 Results

Our investigation into the use of real-time human body segmentation masks as dynamic movement goals has yielded encouraging results, demonstrating the substantial potential of this innovative approach. The core achievement of our application lies in its ability to accurately segment and interpret human body movements in real-time, facilitated by advanced machine learning algorithms. This technology has proven not only highly accurate but also exceptionally responsive, ensuring a seamless user experience.

Looking ahead, several areas present opportunities for future expansion. Enhancing the level of personalization to adapt not just to physical movements but also to individual user preferences and specific movement goals is a promising direction. Expanding the application's scope to include various fields such as rehabilitative therapy, athletic training, and virtual reality experiences could provide valuable insights and increase its utility. There is also a significant opportunity to refine the underlying algorithms for more precise segmentation and reduced latency, which is particularly crucial in dynamic environments. In parallel, focusing on user experience and interface design is essential for making the technology more accessible and intuitive for a broader audience.

### 4.1 Pose Matching and Rep Counting

Pose matching proved to be consistent and reliable, effectively identifying user poses during dynamic movements. Given these results, we could further enhance the robustness of our scoring algorithm by combining pose predictions with segmentation masks, allowing for more accurate assessments. Additionally, the application excelled in repetition counting, accurately tracking the number of completed movements, a critical feature for exercises where monitoring rep counts is essential.

An area identified for enhancement is the application's feedback mechanism. We propose leveraging Large Language Models (LLMs) to provide real-time, descriptive feedback on user performance. For example, during a squat exercise, the system could analyze the user's form and offer specific advice like, "You need to keep your back straight and go lower into the squat position." This level of detailed, real-time feedback could significantly improve user experience and effectiveness in physical training and other applications requiring precise body movement analysis.

## 5 Conclusion

We demonstrated our system was capable of acting as a virtual personal trainer. It could accurately provide specific bottom and top poses and ensure the user hits the correct poses during an exercise. It is versatile in the sense that it allows the user to define any kind of exercise, as long as it has a top and bottom pose. This tool therefore provides additional feedback when people workout and increases the accessibility of this feedback.

### 5.1 Limitations and Future Work

Although we believe our system is a good start, it still has several limitations that currently hold it back from being a full replacement for a personal trainer. First, we recognize that we have a strict definition for an exercise- it is defined as 2 images- a top pose and a bottom pose. We, in this initial implementation, do not provide feedback on the technique between the 2 poses. A user can move however they want between the poses. As long as they achieve the endpoints of the exercise, they get credit for that rep. Along similar lines, we are also currently limited to having only 2 important stages in an exercise. Some exercises might have multiple important poses that need to be achieved at some point. The burpee is a good example of this: it's a push up, combined with a small jump. Adding support for multi-stage exercises improves the versatility of the system. It has currently been designed in a functional way that allows someone to easily expand the code base to support these features. Finally, the third limitation is the speed of the system. Currently, the bottleneck is inference speed from the BlazePose model. We run inference independently on each individual frame. The next step to improve speed is to incorporate the model's ROI persistence. It has a setting in which the region of interest in which a person's pose should be found is not necessarily recomputed for every individual frame that is seen. This can help decrease the time necessary to make inferences in real-time, perhaps enough to have the application running on a low-power mobile device.

In order to further reduce model size, which is useful for running on embedded systems, and improving model speed, we could prune model weights, as well as quantize the model weights. These 2 methods could cut down on the size and number of computations necessary to make predictions. Once pruned and quantized, we would need to refine-tune the refined model in order to preserve accuracy as much as possible. In order to check accuracy is maintained, we would run the refined models on several pose estimation benchmarks.

## References

[1] Andriluka, M., Iqbal, U., Ensafutdinov, E., Pishchulin, L., Milan, A., Gall, J., & Schiele, B. (2018) PoseTrack: A Benchmark for Human Pose Estimation and Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[2] Baek, S., Kim, K.I. & Kim, T. (2020) 3D Hand Shape and Pose Estimation from a Single RGB Image. *arXiv preprint arXiv:2006.11718*. Available at: https://arxiv.org/pdf/2006.11718.pdf.

[3] Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. (2020) Deep Learning for Human Pose Estimation: A Survey. *arXiv preprint arXiv:2006.10204*. Available at: https://arxiv.org/pdf/2006.10204.pdf.

[4] Fang, H.-S., Xie, S., Tai, Y.-W., & Lu, C. (2017) RMPE: Regional Multi-person Pose Estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[5] Lin, T.-Y., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C.L. (2014) Microsoft COCO: Common Objects in Context. *arXiv preprint arXiv:1405.0312*. Available at: http://arxiv.org/abs/1405.0312.

[6] Mahendran N (2021) Multi-Hand Tracking and Gesture Recognition by 3D Hand Pose Estimation with Neural Network. *arXiv preprint arXiv:2109.01376*. Available at: https://arxiv.org/pdf/2109.01376.pdf.

[7] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.