

Motion Analysis Report

1. Model Used And Why:

We used **MediaPipe** as the pose-estimation model because it is lightweight, fast, and runs efficiently on CPU in Google Colab without requiring a GPU. It provides 33 stable and well-tracked body keypoints, including detailed facial points like the nose, making it suitable for frame-to-frame geometric motion measurement. Additionally, it is easy to install and integrate using pip, produces reliable confidence scores for accuracy evaluation, and allows pose-overlay visualization, which supports both quantitative analysis and result verification in representative frames.

When compared with other common pose models:

- **MoveNet:** MoveNet is accurate and works well for single-person detection, but requires loading TensorFlow models, additional preprocessing, and often GPU for smooth large-batch frame processing. Its keypoints are fewer and less stable for face tracking (nose jitter is higher), which affects head-distance metrics.
- **YOLOv8-Pose:** YOLOv8-Pose performs multi-person pose detection and is robust to partial body visibility, but depends on running the Ultralytics stack which increases package weight, load time, and inference latency in CPU-only environments. It is designed more for detection than for smooth tracking, so it may have frame-to-frame keypoint inconsistency when not using GPU.
- **OpenPose:** OpenPose is extremely detailed and accurate, but very heavy, slow in Colab, and requires complex setup using C++ binaries or CUDA compilation. This makes it impractical for automated lightweight analysis.

Our key reasons for choosing MediaPipe Pose:

1. **No GPU requirement:** In contrast to OpenPose and YOLOv8-Pose, MediaPipe runs efficiently on Colab CPU, enabling smooth pose overlay for hundreds of frames.
2. **Tracking stability:** It internally uses a pose-tracking pipeline instead of pure detection, meaning joint positions don't jump drastically between frames, which explains your extremely high PCK score (~0.99+ consistency).
3. **Reliable face keypoints:** Nose keypoint is highly stable compared to MoveNet or detection-only models, which is critical for head movement distance measurement.

4. **Ease of integration:** Installation is simple using Python package, overlay drawing is built-in, and keypoints can be converted directly to pixel space without normalization confusion.
5. **Sufficient precision:** Although YOLOv8-Pose is more accurate overall, MediaPipe gives strong enough joint localization to compute angles (arm, torso, knee) with minimal noise while maintaining high performance.

2. The metric computed

The metric measures how far the person's head moves between frames by tracking the nose keypoint coordinates predicted by the pose model. Since each keypoint gives an (x, y) pixel location, we compute the geometric distance between the nose positions in two frames using the Euclidean distance formula. This formula calculates the straight-line displacement in 2D space and is independent of arm or leg movement, camera zoom, or rotation. The Python function `math.dist()` directly implements this calculation by evaluating $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$, giving the head movement in pixels for each frame pair compared.

- **Detection Rate: 0.81** → poses were found in most of the sampled frames, meaning the model captured a person in 81% of cases.
- **Keypoint Confidence: 0.89** → detected joints were highly reliable and visible.
- **PCK: ~0.999** → keypoints stayed within the allowed distance threshold, showing very stable tracking between frames.
- **OKS: ~0.196** → low score here mainly reflects that the body/face position changed noticeably between frames, not that detection was wrong, since OKS decreases when movement magnitude is large.

The head moved significantly between different motion stages, which increased the geometric distance between nose positions in frame pairs, confirming noticeable 2D motion. Since PCK and confidence are very high, the detected head locations are accurate, and the low OKS score simply indicates larger spatial shifts rather than detection errors. Overall, the pose model reliably tracks head movement in pixel space, and the nose displacement directly represents the person's motion path across frames in 2-dimensional geometry.

3. The numeric result

Frames used for metric:

[100, 300, 500, 700, 900] (sampled as representative motion stages)

Head (nose keypoint) movement between frame pairs:

Frame Pair Nose Displacement (pixels)

100 → 300 **36.0 px**

300 → 500 **22.0 px**

500 → 700 **40.7 px**

700 → 900 **18.9 px**

Best overall movement observed:

- **Max head movement: ~40.7 pixels** between 500 → 700
- **Min head movement: ~18.9 pixels** between 700 → 900

(Values are rounded for simplicity as allowed in the assignment rules.)

The head-motion metric was computed by extracting the 2D (x, y) pixel coordinates of the **nose keypoint (0)** from two frames and applying the Euclidean distance formula to measure straight-line displacement. This was repeated for consecutive representative frame pairs to observe movement variation across motion stages. The values above directly reflect geometric head movement in pixel space.

"The head moved between 18.9px and 40.7px across representative motion stages, with the largest movement (40.7px) occurring between frames 500 and 700."

4. Interpretation

We measured **Head Movement Distance** using the 2D position of the nose keypoint across 5 representative frames from a YouTube video processed in **YouTube**. The head displacement ranged from ~18.9 to 40.7 pixels, meaning the head changed its screen position noticeably between motion stages, especially between frames 500 and 700, where movement was largest. High keypoint confidence and near-perfect PCK indicate the nose locations were consistently tracked, so the numeric distances reflect actual 2D spatial shifts rather than detection instability.