



International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Analysis of Environmental Factors Affecting Real-Time Hand Gesture Recognition Using MediaPipe Landmarks

Smita Mangesh Junnarkar¹, Rohit Bose²

¹Assistant Professor Department of Data Science, SIES College Of Arts, Science And Commerce (Empowered Autonomous), Sion (W), Mumbai. smitaj@sies.edu.in

²Department of Data Science, SIES College Of Arts, Science And Commerce (Empowered Autonomous), Sion (W), Mumbai. rohit25820060@gmail.com

ABSTRACT—

Touch-free control through hand gestures looks convincing in a demonstration, yet many prototypes stumble as soon as they leave the room in which they were built. This paper investigates why. It studies gesture recognition systems that pass MediaPipe hand landmarks to a small classifier, and it follows the route by which a disturbance in the scene travels from the camera, through landmark estimation, to the final gesture label. Eight families of disturbance are examined: lighting, background, distance, camera hardware, viewpoint, occlusion, motion together with frame rate, and differences between users and devices. For each family the paper explains the mechanism, the symptom a developer is likely to notice, and a practical remedy. A test protocol is then proposed, with condition levels, normalisation and stability formulas, and reporting metrics, so that other groups can repeat the measurements on their own equipment. The work is analytical and reports no new benchmark numbers. Its purpose is to help researchers and students plan fair experiments and build gesture interfaces that stay dependable in everyday rooms.

Keywords—*Hand gesture recognition, MediaPipe, hand landmarks, environmental factors, illumination, robustness, real-time systems, human-computer interaction*

1. INTRODUCTION

A camera that understands the hand is attractive for many reasons. A teacher can move through slides from the middle of a classroom, a cook with messy fingers can scroll through a recipe, and a person who cannot use a mouse can still operate a computer. Interest in such touch-free control grew sharply once shared screens and kiosks began to worry the public, and the inexpensive cameras in phones and laptops now supply enough image quality to make the idea affordable [4], [5].

Before learned hand models became available, a vision-based gesture program typically began by isolating skin-coloured pixels, subtracting the background, and measuring the outline of the remaining blob. These steps were quick but brittle: a change in lamp colour, or a wooden table in the frame, was enough to break them [4], [6]. MediaPipe Hands altered the situation by supplying a ready-trained network that finds a hand and returns 21 landmark points in real time, even on a phone [2]. A developer can now train a compact classifier on a few numbers per frame instead of on whole images.

That simplicity hides a risk. Landmarks look like exact measurements, but each one is a prediction made from pixels, and the prediction becomes worse when the pixels do. The classifier receives the same 63 numbers whether the hand was captured in good light or in poor light, and nothing in those numbers announces that they have become less trustworthy. A model that scores almost perfectly at its author's desk can therefore perform much worse beside a bright window in a classroom.

This paper examines that weakness through three questions. First, by which routes do scene conditions reach each stage of a MediaPipe-based recogniser? Second, which conditions deserve the most attention, and how do they combine? Third, what can be done at the data, feature, and system levels to lessen the sensitivity? The answers are developed through a description of the pipeline, a classification of scene conditions, a test protocol, and a set of countermeasures.

The paper is organised as follows. Section 2 summarises earlier studies, and Section 3 explains how MediaPipe Hands processes a video frame. Section 4 classifies the scene conditions that disturb the pipeline. Section 5 presents the test protocol, Section 6 explains how observations should be interpreted, and Section 7 lists countermeasures. Section 8 discusses limitations and ethical issues, Section 9 suggests future directions, and Section 10 concludes.

2. BACKGROUND AND RELATED STUDIES

2.1 Surveys of Vision-Based Gesture Recognition

Review articles on gesture recognition now span almost twenty years. Mitra and Acharya surveyed approaches for hands, arms, faces, and bodies, including hidden Markov models, finite-state machines, and early neural networks [4]. Rautaray and Agrawal narrowed the focus to vision-based human-computer interaction and split the problem into detection, tracking, and recognition, remarking that changes in lighting and background kept causing trouble [5]. Even earlier, a survey of motion capture had pointed out that appearance-based tracking is affected by clothing, illumination, and self-occlusion [13]. Pisharady and Saerbeck examined published datasets and found that many were captured under restricted settings, so reported accuracy often does not carry over to ordinary use [6].

2.2 Locating Hand Keypoints

Locating finger joints in a single photograph is hard because the fingers hide one another and look alike. Simon et al. tackled the lack of labelled images by photographing a hand from many angles at once; predictions from each view were combined geometrically to yield better labels, which were then used to retrain the detector [3]. Their approach helped make precise keypoint models realistic and encouraged smaller models that run on the device itself. The BlazePose body-tracking model follows a similar plan, pairing a quick detector with a regression network designed for tracking [12].

2.3 MediaPipe and On-Device Perception

MediaPipe offers a way to assemble perception systems from reusable processing nodes, and the same graph can run on desktops, phones, and web browsers [1]. Its hand solution pairs a palm detector with a landmark regressor and recycles the previous frame's result so that detection need not run every time [2]. The networks owe part of their speed to mobile-oriented building blocks such as depthwise separable convolutions [7]. Many application papers pair the landmarks with a support vector machine [8], a random forest [9], or a recurrent network [10]. Few of them, however, explain why accuracy shifts when the setup changes, and that question is taken up here.

2.4 Robustness and Fairness

Audits of commercial vision products have shown that accuracy can differ between demographic groups, notably between skin tones, when the training data are unbalanced [14]. That study concerned faces, but the lesson carries over to hands, because skin appearance interacts with lighting and background in ways a model may never have encountered. For this reason the present paper treats differences between users as part of the environment rather than as a side issue.

3. HOW MEDIAPIPE HANDS PROCESSES A FRAME

To see where scene conditions enter, it helps to follow one frame through a typical application. There are five steps: capture, palm detection, landmark regression, feature construction, and classification. The scene affects the first four directly. The fifth is affected only indirectly, through the quality of the features it is given.

3.1 Capture

A camera, often driven through a library such as OpenCV [11], supplies frames at some resolution and rate. Exposure, white balance, focus, and compression are normally managed by the camera driver without any request from the application. These automatic corrections suit human viewers, but they can make brightness jump between neighbouring frames, which is awkward for a model that expects steady input.

3.2 Palm Detection

The first network scans the whole image for palms. Its designers chose palms over complete hands because a palm is fairly rigid and roughly square, and so is easier to localise than a hand with spread fingers [2]. The output is an oriented box around each candidate, accepted only if its score exceeds the minimum detection confidence chosen by the developer.

3.3 Landmark Regression

The cropped region goes to a second network that predicts 21 points: the wrist and four joints on each of the five fingers. Every point has horizontal and vertical positions in image-relative units and a depth value measured relative to the wrist. The network also reports a presence score and a left-or-right label [2]. According to the original report, training data combined hand-annotated photographs with images rendered from a synthetic three-dimensional hand, which was intended to widen the range of poses and appearances covered [2].

3.4 Tracking Between Frames

Running the palm detector on every frame would waste effort. The pipeline instead uses the landmarks of the current frame to place a search region in the next one, and it calls the detector again only when the presence score drops below the minimum tracking confidence [2]. This has a practical consequence. Once a hand has been acquired, performance depends mostly on the landmark network and the tracking region, so anything that breaks tracking, such as rapid movement or sudden occlusion, forces a fresh detection and may leave visible gaps.

3.5 Features and Classifiers

The final stage receives 63 numbers, which are three coordinates for each of 21 points. Raw values depend on where the hand sits in the frame and on how large it looks, so applications normally shift the points to place the wrist at the origin and then divide by a reference length. This removes position and most of the scale. It cannot repair errors in the landmarks themselves, and it cannot restore information lost when a finger is hidden. The classifier may be a static model for single-frame poses or a sequence model for movements such as swipes and circles.

4. SCENE CONDITIONS THAT DISTURB THE PIPELINE

The conditions below are sorted into eight families. For each one the discussion covers how the image changes, how the change spreads through the pipeline, and what a developer is likely to see. The reasoning rests on how the models work and on behaviour commonly reported for vision systems. It is meant to guide experiments, not to replace measurement on a particular device.

4.1 Lighting

Lighting is the condition most often blamed for failures. When a scene is dim, the camera raises its sensor gain, which amplifies noise and weakens the contrast between hand and background. Fingertips suffer first, since they cover only a few pixels. Very bright scenes cause parts of the hand to saturate and lose texture. A single strong lamp throws shadows between the fingers that resemble extra edges, and a window behind the user turns the hand into a dark shape with little surface detail. Coloured or flickering lamps may cause banding or a drifting white balance.

The first sign is usually jitter, meaning small frame-to-frame wobbles in landmark positions. Complete loss of the hand appears later. Accuracy alone is therefore a poor indicator of lighting problems, and stability should be recorded as well.

4.2 Background

A plain wall makes detection easy. A busy scene contains edges and textures that can resemble parts of a hand, which raises the number of false alarms. Skin-toned surfaces such as wooden furniture, or another person's face and hands, may draw the detector toward the wrong target. Because palms are detected first, rounded palm-like objects can also trigger false detections. People walking behind the user can disturb the tracking region.

4.3 Distance and Apparent Size

Distance decides how many pixels the hand occupies. Very close, the hand leaves the frame and the fingertips are cut off. Far away, the hand is small and the landmark network works from a coarse crop. Since positions are expressed in image-relative units, an error of a fixed number of pixels is a larger fraction of a small hand. The usable range therefore depends on resolution and field of view as well as on the model.

4.4 Camera Hardware

Resolution, field of view, lens quality, sensor size, and compression all shape the signal. Cheap webcams and heavily compressed streams, such as those produced by video-call software, wipe out detail around the fingers. Wide lenses curve straight lines near the edges and alter the relative distances between landmarks. Rolling-shutter sensors in phones can skew a fast hand, and autofocus that keeps searching causes intermittent blur. Because users seldom own the camera the developer used, this family is among the most realistic sources of change.

4.5 Viewpoint and Hand Orientation

Orientation belongs to the user more than to the room, yet it is tied to where the camera is placed, so it is included here. A camera above, below, or beside the hand sees a different projection of the same gesture. When the palm faces away, or the hand is seen edge-on, several fingers overlap in the image and depth becomes the only cue that separates them. The depth value from MediaPipe is a relative estimate and is noticeably less dependable than the horizontal and vertical values, so classifiers that rely heavily on it tend to be fragile.

4.6 Occlusion

A hand may be partly hidden by the other hand, by an object it holds, by a sleeve, glove, ring, or bandage, or by its own folded fingers. The network still returns 21 points when some are hidden, filling the gap with plausible positions based on the visible parts and on the structure it has learned. Such guesses can look anatomically sensible and still be wrong for the gesture actually made. Two gestures that differ only in one concealed finger may therefore yield almost identical landmark sets.

4.7 Motion and Frame Rate

A fast-moving hand blurs, especially when the exposure time is long because the room is dim. Blur softens the edges the networks rely on and often breaks tracking. A low frame rate has a similar effect, since the hand travels far between frames and may leave the search region predicted from the previous one. Movement-based gestures suffer twice, because sequence models expect samples at even intervals and irregular or dropped frames distort the temporal pattern. As exposure links lighting and frame rate, a dim room frequently produces blur and a slower effective rate together.

4.8 Differences Between Users and Devices

People differ in hand size, finger proportions, skin tone, nail decoration, and range of movement, and a classifier trained on only a few of them may absorb their personal habits. Devices differ too. A laptop, a phone, and a single-board computer process frames at different speeds, and when processing lags behind the camera, frames are dropped or queued, which adds delay. Phones may also slow down from heat during long sessions. None of these is purely environmental, but in deployment they act like environmental variables because the developer cannot control them.

5. TEST PROTOCOL

A fair comparison changes one condition at a time and keeps everything else as steady as practical. The protocol below is independent of any dataset, so it can be repeated on different hardware.

5.1 Gestures and Participants

A vocabulary of eight to ten static gestures with clearly different finger layouts is suggested: open palm, closed fist, pointing, victory sign, thumbs up, an OK sign, and a pinch, for example. A few dynamic gestures, such as swipes to the left and right and a circle, can be added. At least ten participants are advisable, covering a range of hand sizes and skin tones, and each should repeat every gesture several times per condition. The people used for training should not appear in the test set, so that the result reflects performance on new users.

5.2 Condition Levels

Table 1 lists suitable levels. A baseline should be recorded first, with moderate indoor light, a plain wall, a distance of about 60 cm, and a 720p camera at 30 frames per second. Each factor is then changed on its own, and a few pairs, such as dim light with a cluttered desk, are tested to expose interactions. Illuminance should be measured with a lux meter or a calibrated phone application and logged for every session.

Table 1. Suggested condition levels for controlled testing

Factor	Suggested test levels
Lighting	Dim (under 50 lux), normal indoor (300 to 500 lux), bright (over 1000 lux), window behind the user, mixed warm and cool lamps
Background	Plain wall, moderately cluttered desk, skin-toned surface, second person moving behind the user
Distance	0.3 m, 0.6 m, 1.0 m, 1.5 m, 2.0 m
Camera	480p, 720p, 1080p at 15, 30, and 60 frames per second; wide and standard lenses
Viewpoint	Camera in front, above, below, and beside the hand; palm facing and not facing the lens
Occlusion	None, partly hidden finger, held object, glove, other hand overlapping
Motion	Still, slow movement, fast movement

5.3 Feature Construction

Let p_i denote the three-dimensional position of landmark i , with landmark 0 at the wrist and landmark 9 at the base of the middle finger. Each landmark is shifted and rescaled as follows.

$$\hat{p}_i = (p_i - p_0) / |p_9 - p_0|, \quad i = 0, \dots, 20$$

The 63 values of the normalised vector are then given to the classifier. Joint angles and distances between fingertips can be appended, since both stay unchanged when the hand turns within the image plane. For moving gestures, the latest 30 normalised frames can be stacked and passed to a sequence model. A frame without a detected hand should be marked as missing rather than silently dropped, because the gap itself is information about the scene.

5.4 Classifiers

Three families are proposed because they trade off differently. A support vector machine with a radial kernel [8] and a random forest [9] give strong baselines at low training cost and are available in standard libraries [15]. A small multilayer perceptron provides a compact neural option, and a long short-term memory network [10] can model the order of frames in dynamic gestures. Comparing several classifiers helps separate the harm done by the scene to the landmarks from the weaknesses of one particular classifier.

5.5 Metrics

Table 2 defines the measures. Detection rate and jitter describe the landmark stage, accuracy and F1 describe the final decision, and latency and throughput show whether the system is truly real-time. Using all of them gives a fuller picture than accuracy alone.

Table 2. Evaluation metrics

Metric	Definition	Purpose
Detection rate	Share of frames with a hand present in which a hand is found	Reliability of the detector
Landmark jitter	Spread of each landmark over frames of a held pose, divided by palm width	Stability of landmarks
Accuracy	Share of samples labelled correctly	Overall decision quality
Macro F1-score	Average of the per-class F1 values	Fairness across classes
Latency	Time from capture to final label	Suitability for real-time use
Throughput	Frames processed per second	Detection of dropped frames

The first two metrics can be written compactly. If N_{found} frames out of N_{present} contain a detected hand, and σ_i is the standard deviation of the position of landmark i over a held pose with palm width w , then

$$DR = N_{\text{found}} / N_{\text{present}}, \quad J = (1/21) \sum_i (\sigma_i / w), \quad i = 0 \dots 20$$

Every reported value should carry a confidence interval. Because the same people appear in every condition, differences between conditions should be checked with a paired test across participants, for example the Wilcoxon signed-rank test, which does not assume normally distributed scores.

6. INTERPRETING OBSERVATIONS

6.1 Finding Where a Failure Starts

When accuracy drops, the first question is where in the pipeline the trouble begins. Three kinds of failure can be told apart. In a detection failure no hand is found and the application receives nothing. In a landmark failure a hand is found but its points are inaccurate or unsteady. In a classification failure the landmarks are acceptable but the label is wrong. Distance, background clutter, and strong backlighting mostly produce detection failures. Dim light, blur, and low resolution mostly produce landmark failures, shown as jitter. Occlusion and unusual viewpoints produce landmark and classification failures together, because hidden joints are guessed and the gestures become ambiguous.

The distinction matters because the remedies differ. A detection failure calls for changes to the scene or to the confidence thresholds, a landmark failure for better input or output smoothing, and a classification failure for better training data or features. Table 3 summarises the links.

Table 3. Factors, affected stages, and countermeasures

Factor	Stage most affected	Typical symptom	Main countermeasure
Lighting	Landmark regression	Jitter, missed fingertips	Front lighting, fixed exposure, noise augmentation
Background	Detection	False or wrong-hand detection	Plain backdrop, higher detection threshold
Distance	Detection and landmarks	Clipped or coarse hand	Working-range guidance, higher resolution
Camera	Capture and landmarks	Blur, distortion, lost detail	Adequate resolution, calibrated lens
Viewpoint	Landmarks and classifier	Overlapping fingers, confusion	Invariant features, multi-view data
Occlusion	Landmarks and classifier	Guessed joint positions	Masking augmentation, distinct gestures
Motion	Tracking	Lost tracking, gaps	Higher frame rate, short exposure
Users and devices	Classifier and timing	Uneven accuracy, delay	Diverse participants, lighter model

6.2 Likely Order of Importance

The true ranking depends on the camera and on the gesture set, yet the mechanisms above suggest a general pattern. Lighting and distance tend to act first, since they set the basic quality of the pixels covering the hand. Occlusion and viewpoint come next, because they remove information that no image enhancement can bring back. Background clutter is often less harmful after the tracker has locked on, although it raises the risk of false starts and of the tracker jumping to the wrong target. Motion and frame rate matter more for dynamic gestures. Differences between users are easy to underestimate, because developers usually test only on themselves.

6.3 Combined Effects

Conditions seldom occur alone. A dim room forces long exposures, which increases blur and may lower the frame rate. A busy background together with weak light lowers contrast further and invites false detections. A great distance combined with a low-resolution camera leaves almost no usable detail on the fingertips. Such combinations explain how a system can pass every single-factor test and still fail in a real room, and they are the reason the protocol includes paired conditions.

6.4 Confidence Scores as Early Warnings

The presence and tracking scores are more than settings; they carry information. A falling presence score is an early sign that conditions are getting worse. An application can use it to pause recognition, show a short hint such as “move closer” or “add light”, or switch to a stricter decision rule. If thresholds are too low, false activations rise. If they are too high, the tracker lets go of the hand too often and spends more time re-detecting. The best values depend on the device and the scene and should be found by experiment.

6.5 Latency Budget

Real-time behaviour must be judged from the moment the hand moves to the moment the application responds. Writing the contributions as separate terms makes it easier to see which one to reduce.

$$L = t_{cap} + t_{det} + t_{lm} + t_{feat} + t_{clf} + t_{disp}$$

Here the terms stand for capture, detection, landmark regression, feature computation, classification, and display. Many interface designers aim for a total well under a tenth of a second for direct manipulation, although the acceptable value varies with the task. On limited hardware, a lighter landmark model and a smaller input image can recover speed at some cost in precision, and that trade-off interacts with every scene condition discussed above.

7. COUNTERMEASURES

7.1 Data

The first defence is to train and validate on data that resemble the place of use. Samples from several rooms, lighting levels, cameras, and people are worth more than a very large number from a single spot. Since the classifier receives landmarks rather than images, augmentation can be done directly on the coordinates: small rotations, rescaling, mirroring, and Gaussian noise imitate the jitter produced by poor conditions. Hiding one or two fingertips at random during training helps the model cope with partial occlusion.

7.2 Features

Joint angles and normalised distances are insensitive to camera placement and hand size, so they make the classifier steadier. Ignoring the depth value, or giving it only a small role, reduces reliance on the least dependable output. If both hands will be used, the handedness label allows a left hand to be mirrored into a right-hand frame, so that one model serves both.

7.3 Time

Smoothing the landmark stream suppresses jitter. An exponential moving average is cheap, and for a raw value x_t and a smoothing weight α between 0 and 1 it is written as follows.

$$s_t = \alpha x_t + (1 - \alpha) s_{t-1}$$

A small α gives a very calm signal but adds lag, and adaptive filters that relax the smoothing during fast movement avoid noticeable delay. For static gestures, a majority vote over the last few frames, or a rule that a gesture must persist briefly before it triggers an action, removes flicker and accidental activations. These filters give up a little responsiveness in return for a large gain in stability.

7.4 Image and Hardware

Locking exposure and white balance, where the camera permits, prevents sudden brightness jumps. Contrast-limited adaptive histogram equalisation can help in dim scenes, but it may also amplify noise, so its value should be tested rather than assumed. Placing the light source in front of the user and avoiding windows behind them costs nothing and helps a great deal. A camera with adequate resolution, a suitable field of view, and, where available, a global shutter addresses several camera problems at their source.

7.5 Interface

Good interface design can absorb some of the remaining unreliability. Showing the user whether the hand is currently detected lets them correct their own position. Choosing gestures that look clearly different from each other, and avoiding pairs that differ by one small finger movement, reduces confusion under occlusion. A confirmation step for important actions keeps the cost of a rare misclassification low.

8. LIMITATIONS AND ETHICAL ISSUES

The study has limits. It is analytical, so it contains no newly measured accuracy values, and the statements about the size of each effect should be checked against experiments on specific hardware and software versions. MediaPipe models are updated from time to time, and newer releases may behave differently from the original publication. The protocol concentrates on single-camera, two-dimensional video and does not cover depth cameras or wearable sensors. Finally, the eight families are treated as separate categories even though they overlap in practice.

Several ethical points also deserve attention. A system that works better for some skin tones, hand sizes, or physical abilities than for others creates unequal access, a concern already demonstrated for other vision systems [14]. People with tremors, missing fingers, or limited mobility may be poorly served by gesture sets designed for typical hands. Camera-based interfaces raise privacy questions too, since the video that carries gestures can also reveal faces and surroundings. Processing only the landmarks and discarding raw frames immediately is a more privacy-friendly design and should be preferred where possible. Anyone collecting data should obtain informed consent and store recordings securely.

9. FUTURE DIRECTIONS

Several directions follow from this analysis. The most immediate is to run the proposed experiments and publish the results with open protocols, so that other groups can compare setups. A second is to build a light quality estimator that predicts, from simple image statistics such as mean brightness, a blur measure, and hand size in pixels, whether the current landmarks can be trusted, and to pass that estimate to the classifier. A third is to apply domain adaptation or test-time adaptation so that a model can adjust itself to a new room with little effort from the user. A fourth is to extend the study to two-

handed gestures and to continuous sign-language recognition, where hands often hide each other. Finally, evaluation on edge hardware such as single-board computers would carry the findings to low-cost classroom and assistive uses.

10. CONCLUSION

MediaPipe has put real-time hand landmark extraction within reach of students, researchers, and developers, but easy access does not make a system immune to the scene in front of the camera. This paper traced how eight families of conditions reach the gesture classifier through capture, detection, landmark regression, and tracking. Lighting and distance set the basic quality of the evidence, occlusion and viewpoint remove evidence that cannot be recovered, and camera, motion, and user differences add variation that is easy to overlook during development. Because the conditions interact, testing them one at a time is not enough.

The protocol proposed here combines controlled condition levels, landmark normalisation, several classifiers, and metrics that include stability and latency, and so offers a structured way to measure these effects. The countermeasures at the data, feature, time, image, and interface levels suggest that much of the lost reliability can be regained with modest effort. It is hoped that the paper encourages authors to report gesture recognition results under varied conditions as well as the most favourable ones, so that systems built on MediaPipe landmarks remain dependable in the rooms where people actually use them.

REFERENCES

- [1] C. Lugaresi et al., "MediaPipe: A framework for building perception pipelines," arXiv preprint arXiv:1906.08172, 2019.
- [2] F. Zhang et al., "MediaPipe Hands: On-device real-time hand tracking," arXiv preprint arXiv:2006.10214, 2020.
- [3] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1145–1153.
- [4] S. Mitra and T. Acharya, "Gesture recognition: A survey," IEEE Trans. Systems, Man, and Cybernetics, Part C, vol. 37, no. 3, pp. 311–324, 2007.
- [5] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," Artificial Intelligence Review, vol. 43, no. 1, pp. 1–54, 2015.
- [6] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," Computer Vision and Image Understanding, vol. 141, pp. 152–165, 2015.
- [7] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.
- [8] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.
- [9] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
- [10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
- [11] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.
- [12] V. Bazarevsky et al., "BlazePose: On-device real-time body pose tracking," arXiv preprint arXiv:2006.10204, 2020.
- [13] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," Computer Vision and Image Understanding, vol. 81, no. 3, pp. 231–268, 2001.
- [14] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," Proceedings of Machine Learning Research, vol. 81, pp. 77–91, 2018.
- [15] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.