

Implementation of Human Pose Estimation Using Angle Calculation Logic on The Elder of The Hands as a Fitness Repetition

Irfan, Muchlis abdul muthalib*

Department of Electrical Engineering, Faculty of Engineering, Universitas Malikussaleh, Aceh, Indonesia

*Corresponding author E-mail: muchlis.abd@unimal.ac.id

Manuscript received 19 July 2022; revised 1 Sept 2022; accepted 15 Sept 2022. Date of publication 4 Nov 2022

Abstract

Sport cannot be separated from everyday life. Sport is a hobby for every age group. Everyone needs exercise to keep the body healthy and fit. Many people are busy with their respective activities, so they need more time to exercise and go to the gym. Apart from having to adjust the time, People stuck at Home do not have free access to go the Gym during the global pandemic. This can be overcome by tracking people's movements virtually. Human Pose Estimation can track the movements being carried out by a person in time. Data collection in the form of optimal detection distance, Human Pose Estimation in calculating the number of fitness repetitions, and data from the application of Human Pose Estimation when someone does fitness sports. The optimal distance webcam detects Human Pose Estimation for fitness reps is three meters. Webcams successfully track and predict the movements made by a person. The program can calculate fitness reps based on the angle at the elbow. The key points on the human body are connected by utility lines to form the body's skeleton. Application, The repetitions in this detection utilize the value of the elbow angle when the hand is straight and bent. If the tip is >300 , such as lifting dumbbells, then a stage up is counted as one rep. Meanwhile, when the angle forms >1700 , a stage down and one repetition also count. The use of Media pipe for detection results is accurate and effective.

Keywords: Human Pose Estimation, Sports Fitness, Computer Vision, OpenCV, Python, Media Pipe.

1. Introduction

Exercise can help you lose weight and keep your body in shape. Practice makes blood circulation flow well, maintains ideal body weight, and calms the mind. Many people go to the gym to keep their bodies in shape. Visiting a gym and being trained by a trainer is both costly and affordable for everyone. Apart from adjusting the time between the two parties, people are also stuck at home, not having free access to the gym during this global pandemic [1]. Maintaining endurance and balance with maximum nutritional intake is the primary key to fighting the coronavirus. The coronavirus pandemic has caused many changes in everyday life. Maintaining healthy is very important during this pandemic for a perfect body [18].

The Media pipe library is a solution for tracking human movements when performing sports movements that can be controlled virtually. Detection of human activity is based on the ability of computer vision, which acts as a vision on the computer. OpenCV is a library used as a computer vision technology in this programming. Today, computer vision technology has improved health care through exercise and fitness in detecting body movements. A fitness trainer program can see dangerous or wrong poses for fitness practitioners. The program will track a person's position during exercise to minimize the possibility of hazardous movement so as not to suffer injury. In fitness sports, such as lifting dumbbells, pushups, and pull-ups, there is a movement process in the form of repetitions and sets. The repetition is the repetition of one exercise movement done several times without a break during exercise. Reps are also part of the set. At the same time, the group is a collection of repetitions we have done. Detection Human pose estimation can also help someone calculate the number of repetitions done. Relying on the calculation of the angle formed at the elbow during exercise, reps will be achieved when the arm is bent and the arm is swinging.

Human movement tracking is achieved by studying key points and combinations of poses on a person or object to detect a person's location. In humans. Key attributes consist of various joints, including the wrists, elbows, and knees. Since objects are built-in, this critical point has angles and other essential features. The main objective is to adopt pose estimation to track the above key points in videos or photos. A keypoint is an unlabeled point, and a good keypoint detector generates outputs at pairs of points with locally similar shapes across a couple of similar 2D images or 3D scans[19].

Pose estimation uses the library Media pipe as the primary mode of detection with output, and OpenCV, whose role is to display images or videos. Ideally, the video will be taken with a webcam or from videos in the working directory. Later in the video, each key point will



be connected to the rest of the body, with utility lines forming a human skeleton following the poses performed in real-time. Writing a series of codes will be written in the Visual Studio Code application using Python to help run the project.

2. Literature Review

2.1. Human Pose Estimation

The pose can be defined as the arrangement of human joints in a certain way. Therefore, the Human Pose Estimation can be defined as the localization of human joints or predetermined landmarks. Several types of pose estimation exist in pictures and videos, including body, face, and hands, especially in computer vision [3]. The analysis of human poses from videos plays a vital role in various applications such as measuring physical exercise, sign language recognition, and control of whole-body movements. For example, it can form the basis for yoga, dance, and fitness applications. It can also enable the overlay of digital content and information over the physical world in augmented reality (AR). Human pose estimation aims to determine the position of human joints from images, image sequences, depth images, or skeleton data provided by motion-capturing hardware [20].

The utilization of artificial intelligence is a vital thing whose performance is to adopt the expertise of an expert and store his knowledge in a computer that can solve uncertainties [17]. Pose estimation is one of the contributing factors to the shift from looking for a gym to engaging in home workouts. It works because an artificial intelligence-enhanced personal trainer points the camera at a person while doing an exercise. Then the video from that person will be detected using Pose estimation to see the movement and posture of the person to decide whether the movement made is correct or not. Computer vision detects and predicts athletic activities such as yoga poses, dead-lifts, and other strength training poses.

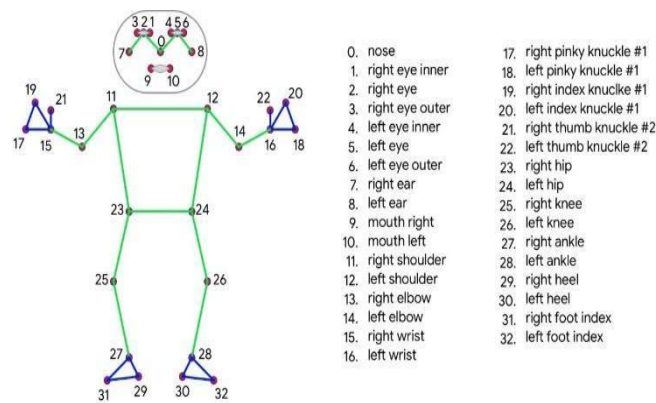


Fig 1. Landmarks Media pipe Blaze Pose

The input image is fed to the Media pipe blaze pose detection key point from the user's body. The output is a list of coordinates on the X, Y, and Z axes for the 33 major critical issues of the human body. This list of coordinates determines the location of each major body part in the input image. The output from the library contains only the coordinates of the user's primary key point in the picture. In Figure 2, the markers show the significant joints and locations in the human body [2].



Fig 2. Blaze Pose.

2.2. Media Pipe Library

MediaPipe purpose is to create a perception pipeline that can be built as a graph of modular components, including model inference, media processing algorithms and data transformations, etc. [15]. Media pipe Hands is a solution for tracking or detecting hands and fingers with high accuracy. Media pipe uses machine learning (ML) to infer 21 3D landmarks using one hand and only take from one frame. As for its cutting-edge approach, Media pipe, for now, relies on a good desktop. With this, the media pipe method achieves a performance up to which it can scale to many hands [12].

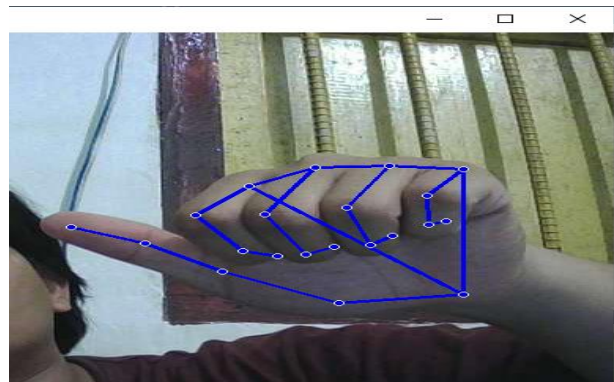


Fig 3. Media pipe Hand Landmarks

Media pipe is intended for tracking movements made by a person. On each limb displayed points connected by lines. This line will connect the coordinate points and form a person's frame. The library must be executed in conjunction with the library for the video from the tracking image to be displayed. To improve detection quality and speed, it is necessary to use backend GPU TensorFlow lite on the device. Media pipe offers a customizable Python solution as a prebuilt in PyPI, which can be installed simply by `pip install media pipe`. It also provides tools for users to build their solutions.

2.3. OpenCV Library

Open-Source Computer Vision (OpenCV) is an open-source library whose purpose is devoted to displaying images or videos. The point is that computers have capabilities similar to visual processing in humans. OpenCV has provided many basic computer vision algorithms and object detection modules using computer vision [11]. OpenCV is also one of many libraries compatible with the real-time focused python programming language, created by Intel and now supported by Willow Garage and It zeas. OpenCV already has many features, including face recognition, face tracking, face detection, Kalman filtering, and various AI (Artificial Intelligence) methods. OpenCV also provides a variety of simple algorithms related to Computer Vision.

2.4. Computer Vision

Computer vision is a science that allows a computer to see objects captured by the camera. The purpose of seeing this is so the computer can identify the image in front of it so that information can be converted into various commands. After capturing a person's image or video and detecting Human Pose Estimation using the system, computer vision can also identify the level of visibility. Then the video data will be saved automatically, or run other commands as needed. The purpose of computer vision is to find the best way so that computers can have "human vision," or the same vision as humans, to detect things that can help us recognize human poses based on their body movements while doing daily activities.

2.5. Python Programming Language

Python is a general-purpose interpretive programming language with a design philosophy focusing on code readability. Python aims to be a language with an extensive and comprehensive standard library feature that combines skills and abilities with a clear code syntax. A large community also supports Python. Python supports several programming paradigms, especially object-oriented, imperative, and functional programming [10].

The inventor of Python was a programmer named Guido Van Rossum in 1990 in Amsterdam, Netherlands. It started with a request from Andrew S. Tenenbaum from Vrije University Amsterdam to Guido Van Rossum. Who wants to create a programming language that can handle Distributed Operating Systems on computers? Python was inspired by a comedy sketch show entitled Mothy Python Flying Circus. Python was initially developed as a response to the ABC programming language.

The advantages of the python programming language include being easier to understand because of writing simple code, available for free and open source, and being flexible. It can be run on almost all operating systems, being versatile because it can be implemented on web development, mobile apps, and desktop apps. Then the main advantage is that it has a library and can be accessed easily. Install each library can be done at the command prompt using `pip install`.

2.5.1. Python Modules

The module is a file that contains a collection of function, class, and variable code stored in a single file .py extension and can be executed by the Python interpreter itself. The name of the module is the name of the file itself. For example, in this study, there is a file called "DeteksiHolistic.py", then a module called "DeteksiHolistic" has been created. The module can have different contents, be it functions, classes, or variables.

2.5.2. Python

The package is a collection of python modules in a folder with a single module constructor (`__init__.py`). This package is a way to manage and organize python modules in a directory form, allowing a module to be accessed using the "namespace" and dot locations. The file constructor tells the python interpreter that the folder is a package. So, any directory or folder containing the module constructor `__init__.py` will be treated as a package. The box is a collection of modules, where the module is a file. Python Contains an array of classes, functions, variables, and other Python code.

2.6. Fitness Exercises

Fitness training is a sports activity to increase endurance and muscle strength. In this case, fitness training can improve the flexibility and balance of the body. The effect of time spent on fitness exercises is added strength and endurance exercise modality per session on the adaptation of muscle strength and hypertrophy [16]. The body will become mentally and spiritually healthy by doing fitness exercises regularly. Several disease risks can be lowered, and calories can be burned effectively if you exercise regularly. Some physical

fitness exercises such lifting weights to increase muscle strength, aerobic exercise for endurance, yoga for flexibility and body balance, pushups, and pull-ups.

3. Result and Discussion

3.1. Implementation

Implementation of this research is the process of making software based on the specifications of the components needed in the system. This system component will then be used for final project research. Members are divided into two, namely hardware components and software components. Later the two parts are interrelated in preparing a system that can detect tracking Human Pose Estimation. The following is a list of components needed for this research:

Hardware

The hardware that will be used in Table 1.

Table 1. Hardware.	
Hardware	Specifications
Processor	Intel(R) Celeron(R) CPU N3350
RAM	4 GB
System Type	64-bit Operating System
Camera	X3 Webcam 1080P FHD
Hard Disk	500 GB
Laptop	Asus X441N

Software

that will be used in This final project research is available in Table 2

Table 2. Software.	
Software	Specifications
Documentation	Microsoft Office 2016
Operating System	Windows 10
Web Browser	Google Chrome, Google Scholar
Programming Language	Python
Library	OpenCV, Numpy, Mediapipe
Code Editor	Visual Studio Code

3.2. Media pipe Result of Distance Detection

This test aims to test the optimal distance for detecting human pose estimation using media pipe and OpenCV when displaying images or videos containing the framework of landmarks on the human body. Every multiple of the distance of three meters in the video. The results can be seen in the following figure.



Fig 4. Media pipe Detection Distance of 3 Meters.

The detected image object has been successfully read and displayed on software with the help of OpenCV. Detection Blaze Pose Python programming language using media pipe can work with high accuracy. The video image is also evident that landmarks pose of all limbs, from landmarks to body skeleton landmarks and landmarks, are detected properly.



Fig 5. Media pipe Detection 18 Meters Distance It is.

Detected that a person's pose can be read only when he is still and slightly moving, but when the person makes a fast movement, such as running, the Human Pose is no longer detected on the software with the help of OpenCV. For Blaze Pose, the Python programming language uses media pipe, the accuracy starts to decrease, and sometimes even Human Pose is lost. People began to look a bit blurry because of the long distance. All head limb landmarks from skeleton landmarks, hand, and landmarks were detected well, although very small.

After testing several ranges of meters with the help of a webcam from the media pipe, the webcam successfully detects the pose of the whole body accurately at 3 meters, 6 meters, 9 meters, 12 meters, 15 meters, and 18 meters. Blaze pose can be detected by media pipe with the library's help to display videos. Every detail of the landmarks in the full body pose can be executed by software using the Python programming language. However, at an estimated distance of more than 19 meters, the detection accuracy declines and sometimes even disappears. This is because the size of the object captured by the webcam has shrunk, or there is a problem with lighting the surrounding environment. Because the lighting of the surrounding environment can be sufficiently bright and light. So, the optimal distance estimation for using the library media pipe blaze pose using a webcam is three meters, while the maximum distance is 18 meters.

The first step to do Pose Human Estimation is to import the library, namely by importing cv2 so that it can display videos and import the library so that it is a solution for tracking someone's movement. To capture videos, use the function `cap = cv2.VideoCapture(0)`, OpenCV is ready to capture the detection video. Entering the media pipe function, i.e., `MP Pose = MP. solutions. Pose` as a solution for overall detection. Then enter `MP Draw = MP. solutions. drawing utils` which serve as a solution in the form of drawing videos. The loop function is used `While`. Loop `While` is needed because the video will continue to be captured frame by frame repeatedly if the image is read successfully. If the image is not read successfully, the program will stop, namely the break.

The last step is providing a timer function, `cv2.waitKey(0)`. If you press any key on the keyboard, the program will continue to run. If the value 0 is filled in brackets, the image will still be displayed; it waits indefinitely to be hit. It can also be set to detect pressing specific keys, such as if the q key is pressed. While the function `0xff == Ord('q')` is to stop the loop, stop the video display if the letter q is pressed.

3.3. Human Pose Estimation Model Fitness Repetition

AI Counter Repetition is an AI fitness personal trainer model that measures the number of repetitions of a person's fitness when lifting dumbbells. Repetition is the number of repetitions performed when performing a sports movement. The repetitions focused on the fitness exercise of lifting dumbbells, pushups, and pull-ups.

The joint angle was measured as the relative angle between the longitudinal axis of two adjacent segments. These segments were composed of three points in the 2D space: a starting point, a middle point, and an endpoint [14]. This model uses the logic of calculating the angle formed along the arm by using three landmarks, namely landmark 11 (shoulder), landmark 13 (elbow), and landmark 15 (wrist). This model can be achieved by estimating the angle landmark elbow. Starting position starts with arms facing down to form an angle value of 1800. Then the associate will lift the weight up to create an angle of 300 angle values from the elbow, and the arm swings down again to form a straight line. When the component includes a slope of 300, the display on the detection will enter data that there has been one movement of lifting the load up (Up). Meanwhile, when the arm forms an angle of 1800, the display on the detection will enter data that there has been one downward movement of the load.

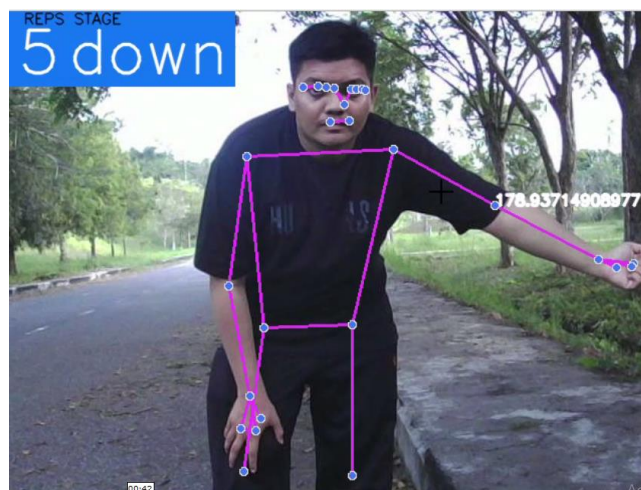


Fig 6. Semicircle Angle Pose 1800

The above display is a human pose estimation when the position will lift dumbbells that form an angle of 178.930 with all hands straight. So that the repetition counter indication box displays the words down.

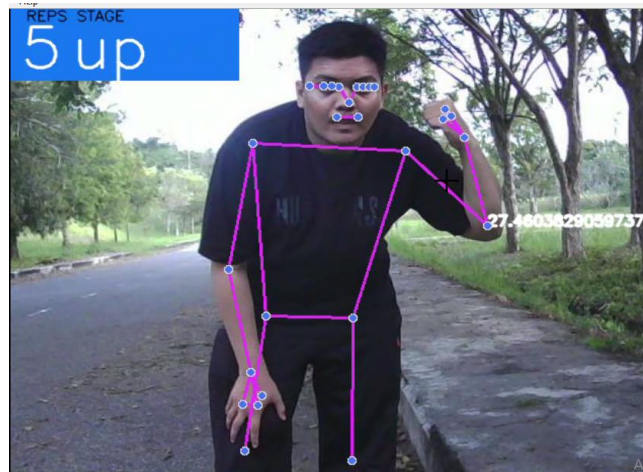


Fig 7. Sharp Angle Pose 300

The arm forms an acute angle of 27.460 with the wrist and elbow bent. Terms of the curve when the stage is up are fulfilled because the tip has reached and passed 300. In the indication of the repetition counter box, the words up appear, which means that when the detection is running, it is lifting the load upwards.

3.4. Angle Calculation Logic

The initial step of this line of code is to calculate the particular joint on the left arm. The three unique connections are left shoulder or left shoulder, left elbow, or left elbow then left wrist or left wrist. The calculate angle function is a function that calculates the under-score of an angle with the example variables ab and c. Variable a is the first value representing the left shoulder, the variable b (middle) means the left elbow, and the last variable c represents the wrist. The following function is to determine the angle of 1800 because the maximum grade obtained from the arm when straight is equal to the tip of the semicircle.

The counter logic of the above line of program code describes the sense of calculating angles. If the angle value is more significant than 1700, then the counter will count one repetition down. If the angle value is less than 300, then the counter will count one repetition up. The print(counter) function is a function that will display the number of calculations from the counter.

```
# Curl counter logic
if angle > 170:
    stage = "down"
if angle < 30 and stage == 'down':
    stage = "up"
    counter += 1
```

3.5. Pull-Up and Push Up

The logic of the three landmarks can also be applied to calculate the repetitions of pull-ups and pushups. The pull-up is the movement of lifting the body repeatedly with the position of both hands hanging on a bar. At the same time, pushups are sports movements that push the body up using both hands in a prone position.



Fig 8. Push-Up Pose Elbow Sharp Angle

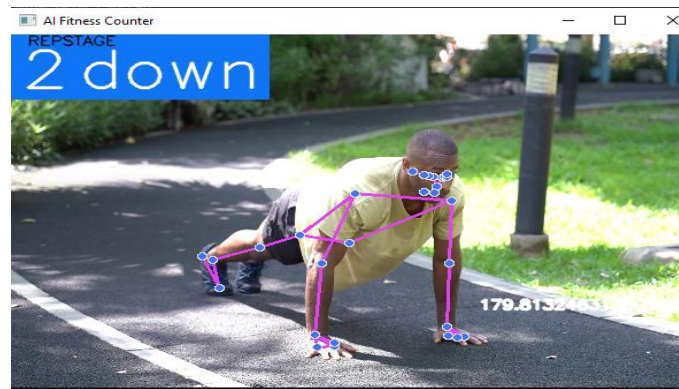


Fig 9. Push-Up Pose Elbow Straight

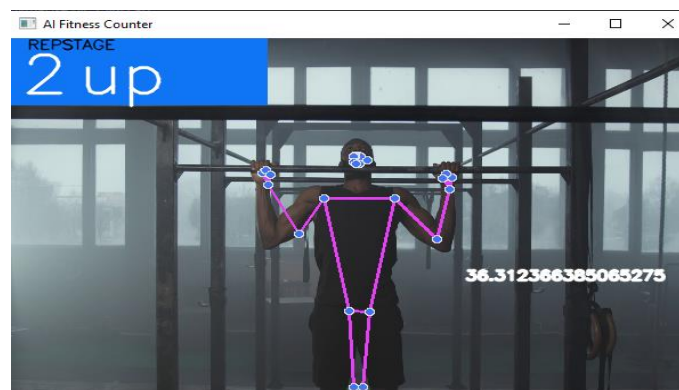


Fig 10. Pull Up Pose

3.6. Landmarks

The following is the pose of a person doing badminton sports movements which will serve as an example for comparison of landmarks in the image below.

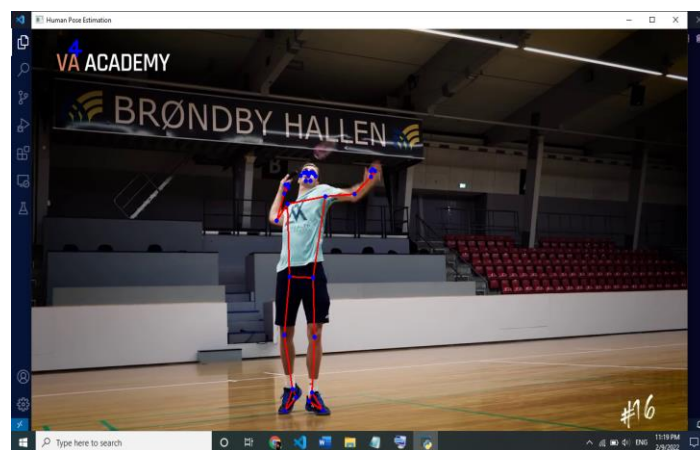


Fig 11. Badminton Smash Pose

Detected the location of each key point according to all body critical issues in the landmarks blaze pose. The utility line connecting the dots of each limb is also successfully associated with accuracy. Python programming uses media pipe to detect moving limbs and displays the body skeleton following the background. This detection is successful because all body parts are seen, starting from landmark 0, namely the nose, which is connected to both eyes and ears on the face, then to landmark the lip. For landmarks on the body, namely, the two shoulders and both hips form a quadrilateral like the constrbody's construction natural, then each shoulder is connected to the elbow, wrist, and hand. At landmarks, the heel and foot are also connected. To make it easier to observe this detection, the color of the critical point is distinguished from the color of the line connecting the key attributes. The color of the critical point will be blue on display, while the color of the line connecting each vital issue will be red. Here are the values for the X, Y, and Z coordinates and the visibility of this motion detection.

Key point 0 x: 0.46373024582862854 y: 0.35044974088668823 z: -0.1867975890636444 visibility: 0.9999932646751404	Key point 1 x: 0.46627718210220337 y: 0.34112244844436646 36568 : 05462951726 visibility: 99847412109375
Key point 2 x: 0.46812134981155396 y: 0.3417436480522156 z: -0.17263266444206238 visibility: 0.9999829530715942	Key point 3 x: 0.4699508249759674 y: 0.34248116612434387 z: -0.1726350337266922 visibility: 0.9999791979789734
Key point 4 x: 0.4600068926811218 y: 0.3396889865398407 z: -0.17250999808311462 visibility: 0.9999875426292419	Key point 5 x: 0.4571159780025482 y: 0.3394552767276764 z: -0.1725156605243683 visibility: 0.9999869465827942
Key point 6 x: 0.45424723625183105 y: 0.3395313322544098 z: -0.1724880337715149 visibility: 0.999987781047821	Key point 7 x: 0.4717233180999756 y: 0.3493706285953522 z: -0.083699069917202 visibility: 0.9999755024909973
Key point 8 x: 0.4489973187446594 y: 0.3462704122066498 z: -0.08283926546573639 visibility: 0.9999867081642151	Key point 9 x: 0.4664023816585541 y: 0.36413002014160156 z: -0.15340425074100494 visibility: 0.9999895691871643
Key point 10 x: 0.4581764340400696 y: 0.36238038539886475 z: -0.1532885730266571 visibility: 0.9999942779541016	Key point 11 x: 0.4924314618110657 y: 0.4248272776603699 z: -0.03946041688323021 visibility: 0.9999016523361206
Key point 12 x: 0.42238330841064453 y: 0.40980762243270874 z: -0.03351890295743942 visibility: 0.9999573230743408	Key point 13 x: 0.5217846632003784 y: 0.47612690925598145 z: -0.15225224196910858 visibility: 0.928395688533783
Key point 14 x: 0.3741885721683502 y: 0.4246509075164795 z: -0.14798149466514587 visibility: 0.9981265664100647	Key point 15 x: 0.5266001224517822 y: 0.4613132178783417 z: -0.332022100687027 visibility: 0.9711547493934631
Key point 16 : 0.38710859417915344 y: 0.40818284968471562250 3 visibility3Keypoint8025	Key point 17 x: 0.5297731161117554 y: 0.45496058464050293 z: -0.36909720301628113 visibility: 0.9498652815818787
Key point 18 x: 0.38717833161354065 y: 0.40378981828689575 z: -0.381781667470932 visibility: 0.9894498586654663	Key point 19 x: 0.524684906002758594 y: 0.9525469 visibility: 75019073486
Key point 20 x: 0.3923008441925049 y: 0.39989036321640015 z: -0.37780827283859253 visibility: 0.9896172881126404	Key point 21 x: 0.5225549936294556 y: 0.44969087839126587 z: -0.33694714307785034 visibility: 0.940367579460144
Key point 22 x: 0.39397555589675903 y: 0.402206689119339 z: -0.3455164134502411 visibility: 0.9859421849250793	Key point 23 x: 0.46851348876953125 y: 0.6118719577789307 z: -0.01086464338004589 visibility: 0.9991260170936584

Key point 24 x: 0.42828255891799927 y: 0.6091117262840271 z: 0.010857395827770233 visibility: 0.9995553493499756	Key point 25 x: 0.475994348526001 y: 0.7433652877807617 z: 0.005432913079857826 visibility: 0.9918479919433594
Key point 26 x: 0.41717714071273804 y: 0.7445051670074463 z: 0.0064722225069999695 visibility: 0.996918797492981	Key point 27 x: 0.4842569828033447 y: 0.8710782527923584 z: 0.11469799280166626 visibility: 0.9910305738449097
Key point 28 x: 0.41428837180137634 y: 0.8791853189468384 z: 0.10748258978128433 visibility: 0.9950273633003235	Key point 29 x: 0.48327332735061646 y: 0.8879143595695496 z: 0.1188134178519249 visibility: 0.8963476419448853
Key point 30 x: 0.41925668716430664 y: 0.8990371823310852 z: 0.11146845668554306 visibility: 0.9168241024017334	Key point 31 x: 0.48370295763015747 y: 0.9154886603355408 z: 0.013282385654747486 visibility: 0.989659309387207
Key point 32 x: 0.4019554555416107 y: 0.923187255859375 z: 0.015206356532871723 visibility: 0.9914125800132751	

The critical point coordinates are the sample of crucial point coordinates: key point 13 is the left, the low key point, and vital point 14 is the right, the elbow. Comparing the two coordinate values for image width or key point x, image height or critical point y, and image depth or key point z, landmarks 13 are more significant than landmarks 14. As for visibility landmarks, 14 or right elbow is more important in value than landmarks 13, which is 0.998, which is almost close to one. Comparison of the values between critical points in the below graphic.

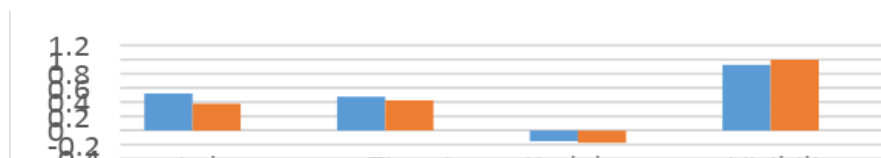


Fig 12. Landmarks Comparison

4. Conclusion

Based on the results and discussion of the implementation of human pose estimation using the angle that has been made, it can be concluded that:

1. The construction formed from the Human Pose Estimation Detection is the body frame resulting from the body's key points by utility lines that can follow a person's movements in real-time. Human Pose Estimation, equipped with AI fitness-repetition capabilities, can calculate the number of repetitions related to sports by calculating the value of the angle of the elbow.
2. After testing, the optimal distance for Detection of Human Pose Estimation is three meters.
3. Calculate the number of fitness reps by calculating the value of the angle at the elbow as the main calculation. Shoulders and wrists as endpoints form an arch. Starting position starts with arms facing down to include an angle value of 1800. Then the associate will lift the weight up to create an angle of 300 angle values from the elbow, and the component will swing back down to form a straight line. When the arm comprises a slope of 300, the program will count once the movement of lifting the load (Up). Meanwhile, when the component includes an angle of 1800, the program will calculate one motion of swinging the bag down (Down).

References

- [1] H. Pardeshi, A. Ghaiwat, and A. Thongire. 2021. "Fitness Freaks: A System For Detecting Definite Body Posture Using Open Pose Estimation". Mumbai
- [2] A. Anil Kumar, AKT, S. Sajan, and SKA 2021. "Pose Estimated Yoga Monitoring System,". SSRN Electron. J., No. Illiceis, doi: 10.2139/ssrn.3882498. India: Ernakulam.
- [3] R. Josyula, S. Ostadabbas. 2021 "A Review on Human Pose Estimation,". Northeastern University.
- [4] F. Zhang, Fan, Bazarevsky, Valentin, Vakunov, Andrey. 2020. "MediaPipe Hands: On-device Real-time Hand Tracking,". USA: Mountain View CA 9403.
- [5] SR Sulistiyanti, FX. A. Setyawan, M. Komarudin. 2016 "Teaching Book of Image Processing". Yogyakarta: Teknosain.
- [6] J. Main. 2011. "Digital Image Acquisition using MATLAB Programming". Maj. Unikom. Vol. 9, no. 1, pp. 71–80.
- [7] MM Sobel, R. Canny, P. Teguh, K. Putra, N. Kadek, A. Wirdiani. 2016. "Edge Detection Digital Image Processing To Compare

- Sobel, Robert and Canny Methods". J. Ilm. Merpati (Research Tower. Akad. Technol. Information), vol. 2, no. 2, pp. 253–261.
- [8] A. Ahmad Hania. 2017. "Knowing Artificial Intelligence, Machine Learning, & Deep Learning". J. Teknol. Indonesia., vol. 1, no. June, pp. 1–6.
 - [9] Syarifah, Fajriya. 2018. "Deep Learning Object Detection in Video Using Tensorflow and CNN". Yogyakarta: Indonesian Islamic University.
 - [10] IK Setia Buana, 2018. "Applications for Computer Operations By Detecting Motion Using Opencv Python". Bali: STIKOM, pp. 190–191.
 - [11] TCA-S. Zulkhaidi, E. Maria, Y. Yulianto. 2020. "Recognition of Face Shape Patterns with OpenCV". J. Technol Engineering. inf., vol. 3, no. 2, p. 181. Samarinda: Agricultural Polytechnic.
 - [12] SS Makahaube, AM Sambul, and SRU Sompie. 2021. "Implementation of Gesture Recognition Technology for Automated Education Service Kiosk". Vol. 16, no. 4, pp. 465–472. Manado: Sam Ratulangi University.
 - [13] Kartini, Bagus. 2020. "Object Detection and Leaf Damage Using Digital Image Processing Methods". Laguboti: Del Institute of Technology.
 - [14] Hernández, Oscar G, Morell. 2021. "Human Pose Detection for Robotic-assisted and Rehabilitation Environments". Spain: University of Alicante.
 - [15] C. Lugaresi, J. Tang, H. Nash. 2019. "Mediapipe: A Framework for Building Perception Pipelines". Google Research.
 - [16] N. Kikuchi, S. Yoshida, M. Okuyama. 2016. "The Effect of High-Intensity Interval Cycling Sprints Subsequent to Arm-Curl Exercise on Upper-Body Muscle Strength and Hypertrophy". Tokyo: Nippon Sport Science University.
 - [17] Zulfian Azmi. 2021. "Artificial Neural Network Model For Wind Mill". Medan: STMIK Triguna Dharma.
 - [18] I Gede Dharma Utayasa. 2021. "Effect Physical Activity and Nutrition During The Covid-19 Pandemic". Surabaya: PGRI Adi Buana Surabaya University.
 - [19] C. Creusot, N. Pears, J. Austin. 2013. "A machine-learning approach to keypoint detection and landmarking on 3D meshes". International Journal of Computer Vision, vol. 102, no 1-3, pp. 146-179.
 - [20] A. Kitsikidis, K. Dimitropoulos, S. Douka, and N. Grammalidis. 2014. "Dance analysis using multiple kinect sensors," in Proceedings of the 9th International Conference on Computer Vision Theory and Applications, VISAPP 2014, pp. 789–795.