

Home

A Comprehensive Guide on Human Pose Estimation



Mrinal Singh Walia — Updated On February 10th, 2022

[Algorithm](#) [Computer Vision](#) [Deep Learning](#) [Guide](#) [Intermediate](#) [Machine Learning](#) [Object Tracking](#)

This article was published as a part of the [Data Science Blogathon](#).

Human Pose estimation is a computer vision task that represents the orientation of a person in a graphical format. This technique is widely applied to predict a person's body parts or joint position. It is one of the most exciting areas of research in computer vision that has gained a lot of traction because of its abundance of applications that can benefit from such a technology.

Today's article will shed light on human pose estimation and how it works. Also, I will try to cover some top and open-source pose estimation projects on Github along with some state-of-the-art models, techniques, and types of pose estimation that prove to make tremendous advancements in this field.

Going through this comprehensive guide, you will come across below topics:

1. What is Human Pose Estimation?
2. Importance of Human Pose Estimation
3. Difference Between 2D and 3D Human Pose Estimation
4. Types of Human Pose Estimation Models
5. Bottom-Up VS. Top-Down Methods of Pose Estimation
6. How Does Human Pose Estimation Works?
7. Simple Human Pose Estimation Code
8. Applications Of Human Pose Estimation
9. Final Words
10. About Author
11. References

What is Human Pose Estimation?



<https://www.analyticsvidhya.com/blog/2021/10/human-pose-estimation-using-machine-learning-in-python/>

Human Pose Estimation identifies and classifies the poses of human body parts and joints in images or videos. In general, a model-based technique is used to represent and infer human body poses in 2D and 3D space.

Essentially it is a way to capture a set of coordinates by defining the human body joints like wrist, shoulder, knees, eyes, ears, ankles, and arms, which is a key point in images and videos that can describe a pose of a person.

Then, when an image or video is given to the pose estimator model as input, it identifies the coordinates of these detected body parts and joints as output and a confidence score showing precision of the estimations.

Importance of Human Pose Estimation



Recommended For You

[Become a full stack data scientist](#)

[Learning Pose Estimation Using New Computer Vision Techniques](#)

[Know all About 2D and 3D Pose Estimation!](#)

[Human Pose Estimation Using Machine Learning in Python](#)

[Pose Estimation using OpenCV](#)

[5 Exciting Computer Vision Applications With Relevant Datasets!](#)

[18 All-Time Classic Open Source Computer Vision Projects for Beginners](#)

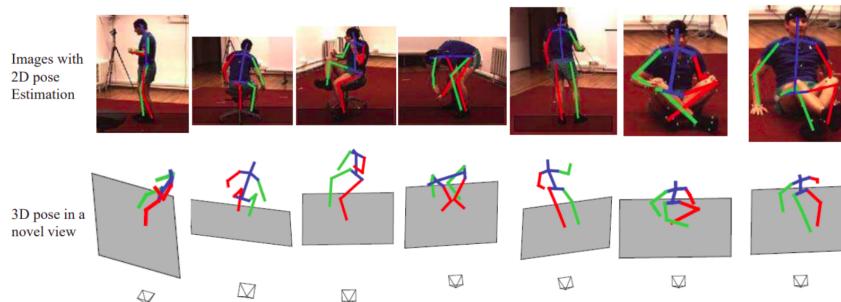
Detection of people has long been a primary centre of discussion for various applications in traditional object detection. With recent developments in machine-learning algorithms, computers can now understand human body language by performing pose detection and pose tracking. The accuracy of these detections and the hardware requirements to run them has now reached a point where it has become commercially viable.

In addition, the technology's growth is also profoundly transformed amid the coronavirus pandemic, where high-performing real-time pose detection and tracking will bring some of the most influential trends in computer vision. For instance, it can be employed for social distancing by combining human pose estimation and distance projection heuristics. It assists people in maintaining physical distance from each other in a crowded place.



Human pose estimation will significantly impact various industries, including security, business intelligence, health and safety, and entertainment. One such area where this technique has already proved its existence is autonomous driving. With the help of real-time human pose detection and tracking, computers can sense and predict pedestrian behaviour much thoroughly – allowing more consistent driving.

Difference Between 2D and 3D Human Pose Estimation



https://openaccess.thecvf.com/content_cvpr_2017/papers/Chen_3D_Human_Pose_CVPR_2017_paper.pdf

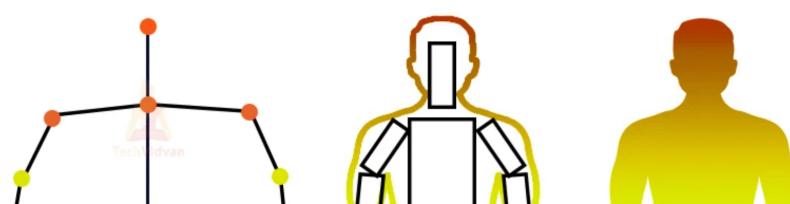
There are majorly two techniques in which pose estimation models can detect human poses.

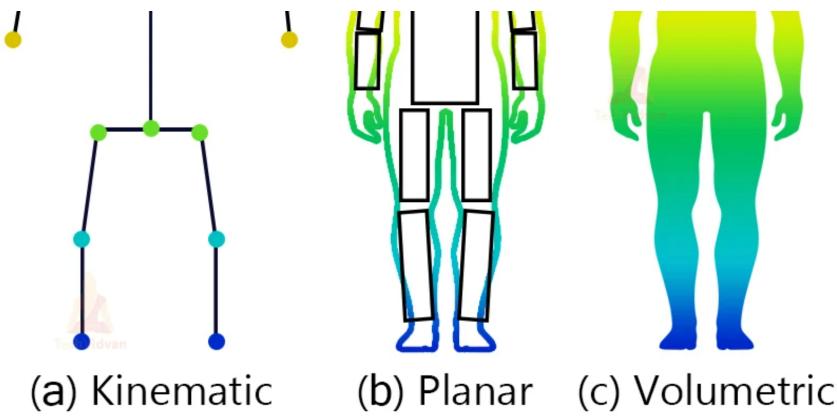
1. **2D Pose Estimation:** In this type of pose estimation, you simply estimate the locations of the body joints in 2D space relative to input data (i.e., image or video frame). The location is represented with X and Y coordinates for each key point.
2. **3D Pose Estimation:** In this type of pose estimation, you transform a 2D image into a 3D object by estimating an additional Z-dimension to the prediction. 3D pose estimation enables us to predict the accurate spatial positioning of a represented person or thing.

3D pose estimation is a significant challenge faced by machine learning engineers because of the complexity entailed in building datasets and algorithms that estimate several factors, such as an image's or video's background scene, light conditions, and besides.

Types of Human Pose Estimation Models

There are three main types of human pose estimation models used to represent the human body in 2D and 3D planes.





“ ”

<https://www.semanticscholar.org/paper/Deep-Learning-Based-Human-Pose-Estimation%3A-A-Survey-Zheng-Wu/0edef16d8fb78625ec5a050e2a7ae4efffe3689>

#1. Skeleton-based model: also called the kinematic model, this representative includes a set of key points (joints) like ankles, knees, shoulders, elbows, wrists, and limb orientations primarily utilized for 3D and 2D pose estimation.

This flexible and intuitive human body model comprises the human body's skeletal structure and is frequently applied to capture the relations between different body parts.

#2. Contour-based model: also called the planar model, it is used for 2D pose estimation and consists of the contour and rough width of the body, torso, and limbs. Basically, it represents the appearance and shape of a human body, where body parts are displayed with boundaries and rectangles of a person's contour.

A famous example is the [Active Shape Model](#) (ASM), which captures the entire human body graph and silhouette deformations employing the principal component analysis (PCA) technique.

#3. Volume-based model: also called the volumetric model, is used for 3D pose estimation. It consists of multiple popular 3D human body models and poses represented by human geometric meshes and shapes, generally captured for deep learning-based 3D human pose estimation.

Bottom-Up VS. Top-Down Methods of Pose Estimation

All methods for human pose estimation can be classified into two primary approaches: bottom-up and top-down.

1. **Bottom-up methods** evaluate each body joint first and then arrange them to compose a unique pose.
2. **Top-down methods** run a body detector first and determine body joints within the discovered bounding boxes.

How Does Human Pose Estimation Works?

Now that you know what pose estimation is, why it is essential, and the difference between various methods, models, and techniques, it's now time to look into its working. Yes, we are going to talk about how human pose estimation works, and this section is divided into 3 sub-categories respectively:

- Basis Structure
- Model Architecture Overview
- Different Approaches For Human Pose Estimation

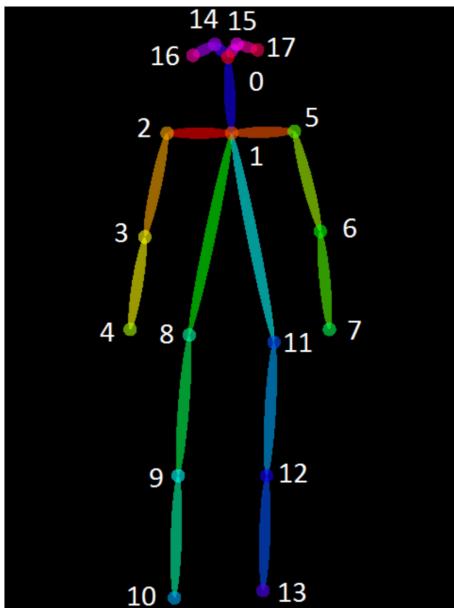




‘ ‘ ‘

<https://indatalabs.com/resources/human-activity-recognition-fitness-app>

Basic Structure



‘ ‘ ‘ ‘

<https://nanonets.com/blog/human-pose-estimation-2d-guide/>

There are several solutions proposed for the problem of human pose estimation. However, on the whole, the existing methods can be subcategorized into three groups, namely Absolute Pose Estimation, Relative Pose Estimation, and the appropriate Pose estimation, which is the combination of both.

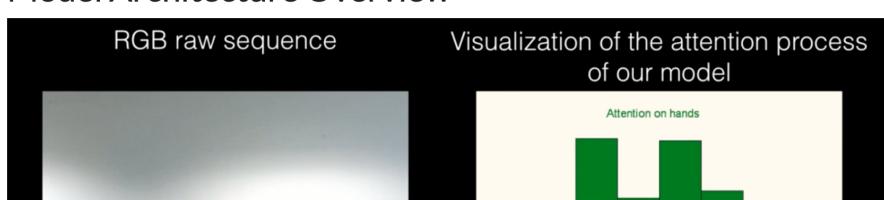
The first: Absolute pose estimation method is based on satellite-based navigation signals, navigation beacons, active and passive landmarks, and heatmap matching. The second: Relative pose estimation method is based on dead reckoning, which incrementally updates the human pose by estimating the distance from a known joint, i.e., a human's initial position and orientation.

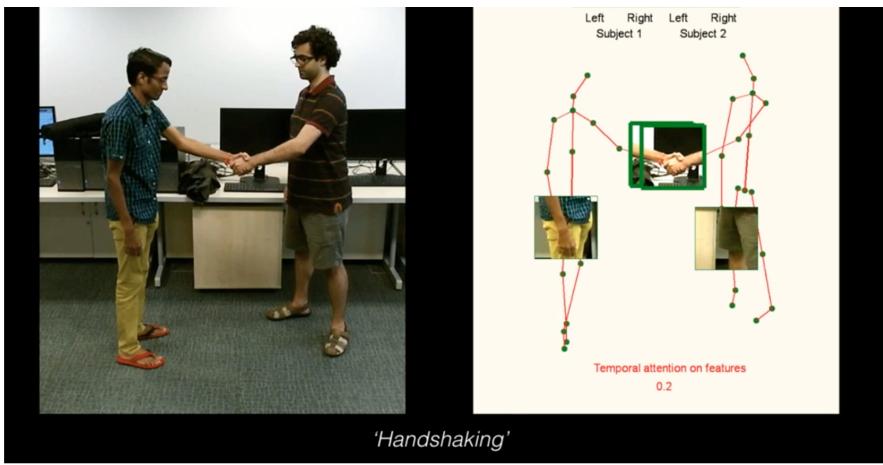
Fundamentally, most algorithms use human pose and orientation to predict a person's location concerning the background. It is a 2 steps framework that identifies human bounding boxes and then evaluates each box's pose.

Next, it estimates the key points for a person that would be jointed like the elbow, knees, wrists, etc. We can estimate poses for a single person or multiple people depending on the application.

In single pose estimation, the model estimates the poses for a single person in a given scene. In contrast, in the case of multi-pose estimation, the model estimates the poses for multiple people in the given input sequence.

Model Architecture Overview





‘ ‘ ‘ ‘

<https://nanonets.com/blog/human-pose-estimation-2d-guide/>

Multiple specific neural network architectures cannot be covered here in a single article, but we'll talk about a few robust, reliable ones that make good points to start.

Human pose estimation models come in a few varieties, i.e., bottom-up and top-down approaches mentioned above. The most famous architecture begins with an encoder that takes an input image and extracts features utilizing a series of narrowing convolution blocks. The next step after an encoder varies with the method used for pose estimation.

The most conceptually simplistic system practices a regressor to final output predictions of each keypoint location by accepting an input image and outputs X, Y, and Z coordinates for each key point you're attempting to predict. However, practically this architecture is not used as it does not produce accurate results without further refinement.

A somewhat more complex approach practices an encoder-decoder architecture. Instead of calculating joint coordinates directly in this architecture, the encoder is fed into a decoder, generating heatmaps. These heatmaps represent the likelihood of a joint detected in a given section of an input image.

The precise coordinates are chosen by selecting heatmap locations with the highest joint likelihood during the post-processing. Further, in the case of multi-pose estimation, a heatmap comprises multiple regions of high keypoint likelihood, for instance, 2 or more left hands in an image. It is done to assign each location to a specific human model.

The architectures discussed above apply equally to 2D and 3D pose estimation.

Different Libraries For Human Pose Estimation

With the rapid development in the classical computer vision methods, pose estimation, including image segmentation and object detection, has outperformed in various tasks.

This section will list and review the top five most popular pose estimation libraries available on the internet for public use. You can implement a custom human pose estimator using the below libraries.

#1. OpenPose



Documentation: <https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/index.html>

Github Link: <https://github.com/CMU-Perceptual-Computing-Lab/openpose>

Github Stars: 22.8K

Github Fork: 0.0K

OpenPose is a free human joints detection library that works in real-time. It detects key points for the body, face, hands, and foot estimation. It is the first multi-person system to jointly detect 135 key points in total on a single input image. It is one of the most popular multi-person human pose estimation libraries that uses a bottom-up approach.

OpenPose is an open-source API that gives users the flexibility of selecting input images from camera fields, webcams, and other sources for embedded systems applications. It supports different hardware architectures, including CUDA GPUs, OpenCL GPUs, and CPU-only systems. It is widely used for 2D pose estimation (whole-body), 3D pose reconstruction, and estimation (whole-body), and unity plugin.

#2. PoseDetection



Github Link: <https://github.com/tensorflow/tfjs-models/tree/master/pose-detection>

Github Stars: 10.3K

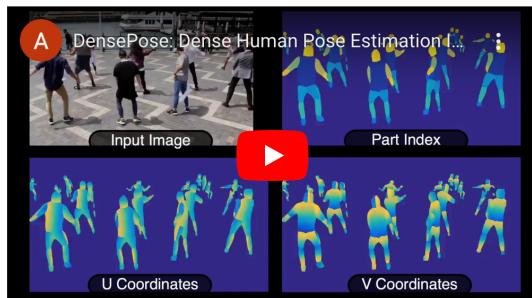
Github Fork: 3.2K

Pose detection is an open-source real-time pose detection library that can detect human poses in images or videos. It is a pose estimator architecture built on tensorflow.js and allows you to detect body parts such as elbows, hips, wrists, knees, ankles, and others for either a single pose or multiple poses.

It is built to run efficiently on lightweight devices such as browsers or mobile devices. This package offers three state-of-the-art models for running real-time pose estimation:

- MoveNet (detects 17 key points & runs at 50+ fps)
- BlazePose (detects 33 key points)
- PoseNet (capable of detecting multiple poses, and each pose contains 17 key points)

#3. DensePose



Documentation: <http://densepose.org/>

Github Link: <https://github.com/facebookresearch/Densepose>

Github Stars: 6.2K

Github Fork: 1.2K

Dense human pose estimation is a free, open-source library that can map all human pixels of 2D RGB images to a 3D surface-based model of the body in real-time. This library is implemented in the detectron framework, powered by caffe2, and can also be used for single and multiple pose estimation problems.

#4. AlphaPose





Documentation: <https://www.mvig.org/research/alphapose.html>

Github Link: <https://github.com/MVIG-SJTU/AlphaPose>

Github Stars: 5.7K

Github Fork: 1.6K

Alphapose is an open-source real-time multi-person pose estimation library that uses a popular top-down approach and is very accurate. This library helps detect poses in the presence of inaccurate human bounding boxes, and it is an optimal architecture for determining human poses by optimally detected bounding boxes.

Alphapose also provides an efficient online pose tracker to associate poses that indicate the same person across frames. It is the first open-sourced online pose tracker and is called PoseFlow. This library can detect accurate real-time multi-person and single-person keypoints in images, videos, and image lists.

#5. HRNet (High-Resolution Net)



Documentation: <https://jingdongwang2017.github.io/Projects/HRNet/PoseEstimation.html>

Github Link: <https://github.com/leoxiaobin/deep-high-resolution-net.pytorch>

Github Stars: 3.5K

Github Fork: 810

HRNet is an architecture used for human-pose estimation to find what we know as key points with respect to the specific objects or person in an image. It maintains high-resolution representations throughout the process and predicts a very accurate key point heatmap.

Additionally, this architecture is suitable for detecting human posture in televised sports. Many other dense prediction tasks, such as segmentation, face alignment, object detection, etc., have benefited from HRNet.

Simple Human Pose Estimation Code

There are a lot of public datasets available both for 3D and 2D pose estimation.

3D Pose Estimation Datasets

- [DensePose](#)

- [UP-3D](#)
- [Human3.6m](#)
- [3D Poses in the Wild](#)
- [HumanEva](#)
- [Total Capture](#)
- [SURREAL \(Synthetic hUmans foR REAL tasks\)](#)
- [JTA Dataset](#)
- [MPI-INF-3DHP](#)

2D Pose Estimation Datasets

- [MPII Human Pose Dataset](#)
- [Leeds Sports Pose](#)
- [Frames Labeled in Cinema](#)
- [Frames Labeled in Cinema Plus](#)
- [YouTube Pose \(VGG\)](#)
- [BBC Pose \(VGG\)](#)
- [COCO Keypoints](#)

In our example, we will use the pre-trained model by the Openpose team using Caffe on MPI Dataset, which has 15 key points to identify various joints in the human body.

```
"Head": 0, "Neck": 1, "RShoulder": 2, "RElbow": 3, "RWrist": 4,
"LShoulder": 5, "LElbow": 6, "LWrist": 7, "RHip": 8, "RKnee": 9,
"RAngle": 10, "LHip": 11, "LKnee": 12, "LAngle": 13, "Chest": 14,
"Background": 15
```

Define the pose pairs used to create the limbs that will connect the key points. Then download the pre-trained models.

```
Pose_Pairs = [ ["Head", "Neck"], ["Neck", "RShoulder"], ["RShoulder", "RElbow"],
               ["RElbow", "RWrist"], ["Neck", "LShoulder"], ["LShoulder", "LElbow"],
               ["LElbow", "LWrist"], ["Neck", "Chest"], ["Chest", "RHip"], ["RHip", "RKnee"],
               ["RKnee", "RAngle"], ["Chest", "LHip"], ["LHip", "LKnee"], ["LKnee", "LAngle"]
MODEL_URL="http://posefs1.perception.cs.cmu.edu/OpenPose/models/"
POSE_FOLDER="pose/"
MPI_FOLDER=${POSE_FOLDER}"mpi/"
MPI_MODEL=${MPI_FOLDER}"pose_iter_160000.caffemodel"
wget -c ${MODEL_URL}${MPI_MODEL} -P ${MPI_FOLDER}
```

#1. Read the .prototxt file and load the pretrained weights on the network.

```
net = cv.dnn.readNetFromCaffe(args.proto, args.model)
```

#2. Next, load images in a batch and pass them through the neural network.

```
blob = cv.dnn.blobFromImage(image, scalefactor, size, mean, swapRB, crop)
```

#3. Call the forward function to run the inference on the input images. Then generate the confidence map for each keypoint.

```
inp = cv.dnn.blobFromImage(frame, 1.0 / 255, (inWidth, inHeight),
                           (0, 0, 0), swapRB=False, crop=False)
net.setInput(inp)

out = net.forward()
```

#4. Display these critical points on the original image.

```
for i in range(len(BODY_PARTS)):
    # Slice heatmap of corresponding body's part.
    heatMap = out[0, i, :, :]
    # Originally, we try to find all the local maximums. To simplify a sample
    # we just find a global one. However only a single pose at the same time
    # could be detected this way.
    _, conf, _ = cv.minMaxLoc(heatMap)
    x = (frameWidth * point[0]) / out.shape[3]
    y = (frameHeight * point[1]) / out.shape[2]
    # Add a point if its confidence is higher than threshold.
    points.append((int(x), int(y)) if conf > args.thr else None)

for pair in POSE_PAIRS:
    partFrom = pair[0]
    partTo = pair[1]
    assert(partFrom in BODY_PARTS)
    assert(partTo in BODY_PARTS)
    idFrom = BODY_PARTS[partFrom]
    idTo = BODY_PARTS[partTo]
```

```

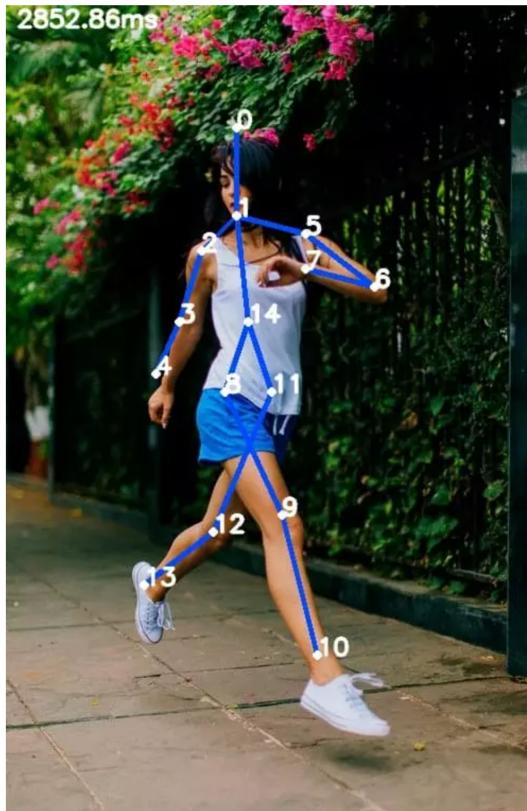
if points[idFrom] and points[idTo]:
    cv.line(frame, points[idFrom], points[idTo], (255, 74, 0), 3)
    cv.ellipse(frame, points[idFrom], (4, 4), 0, 0, 360, (255, 255, 255), cv.FILLED)
    cv.ellipse(frame, points[idTo], (4, 4), 0, 0, 360, (255, 255, 255), cv.FILLED)
    cv.putText(frame, str(idFrom), points[idFrom], cv.FONT_HERSHEY_SIMPLEX, 0.75, (255,

```

#5. Save the file and run from the command prompt using the assigned arguments.

```
python3 sample.py --input sample.jpg --proto pose/mpi/pose_deploy_linevec_faster_4_stage
```

#6. Results



Applications Of Human Pose Estimation

Human pose estimation is the most talked about topic in computer vision, and it has been utilized in a wide array of applications and use-cases. Some include human-computer interaction, motion analysis, augmented reality, and robotics.

In general, human pose estimation has endless applications in almost every domain. Some of the most common applications which are in development are:

1. Human Activity & Movement Estimation

One of the most apparent dimensions applicable to pose estimation is tracking and measuring human activity and movement. Many architectures like OpenPose, PoseNet, and DensePose are often practised for action, gesture, or gait recognition. Some examples of human activity tracking are:

- AI-powered sports coaches or personal gym trainer
- Sitting gestures detection
- Workplace activity monitoring
- Sign language communication for disabled
- Traffic policemen signal detection
- Cricket umpire signal detection
- Dance techniques detection
- Monitoring movements in security and surveillance
- Crowd counting and tracking for retail outlets

2. Augmented Reality & Virtual Reality (AR/ VR)

When clubbed with augmented and virtual reality applications, human pose estimation presents an

opportunity to create more realistic and responsive experiences. For example, you can learn to play various games like tennis or golf via virtual tutors who are pose illustrated. More so, the US army has implemented AR programs in combat. It helps soldiers distinguish between enemies and friendly troops.

3. Robotics

Traditional industrial robots are based on 2D vision systems with many limitations. In place of manually programming robots to learn movements, a 3D pose estimation technique can be employed. This approach creates more responsive, flexible, and true-to-life robotics systems. It enables robots to understand actions and movements by following the tutor's posture, look, or appearance.

4. Animation & Gaming

Modern advancements in pose estimation and motion capture technology make character animation a streamlined and automated process. For example, Microsoft's Kinect depth camera captures human motion in real-time using IR sensors data and uses it to render the characters' actions virtually into the gaming environment. Likewise, capturing animations for immersive video game experiences can also effortlessly be automated by different pose estimation architectures.

EndNotes

Pose estimation is a captivating computer vision component utilized by multiple domains, including technology, healthcare, gaming, etc. I hope my comprehensive guide on Human Pose Estimation helped explain the basics of human pose estimation, its working principles, and how it can be utilized in the real world.

About Author

Mrinal Walia is a professional Python Developer with a Bachelor's degree in computer science specializing in Machine Learning, Artificial Intelligence, and Computer Vision. Mrinal is also a freelance blogger, author, and geek with four years of experience in his work.

If you liked reading my article, connect with me on my social media profiles and follow me for a quick virtual cup of coffee.

[Instagram](#) | [Facebook](#) | [LinkedIn](#) | [Github](#) | [Email](#) |

Thank you, and don't have a good day, have a great day!

Read more articles [here](#) on Human and Object detection.

References

- https://www.researchgate.net/publication/225561390_A_Novel_Pose_Estimation_System_for_Indoor_N
- <https://viso.ai/deep-learning/pose-estimation-ultimate-overview/>
- <https://www.fritz.ai/pose-estimation/#part-how>
- <https://learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/>
- <https://cv-tricks.com/pose-estimation/using-deep-learning-in-opencv/>

The media shown in this article is not owned by Analytics Vidhya and are used at the Author's discretion.

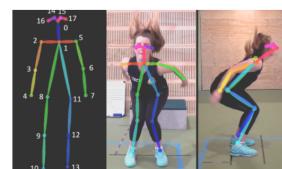
Related



[Learning Pose Estimation Using New Computer Vision Techniques](#)



[Know all About 2D and 3D Pose Estimation!](#)



[Human Pose Estimation Using Machine Learning in Python](#)



Mrinal Singh Walia

Our Top Authors

view
more

Download

Analytics Vidhya App for the Latest blog/Article



Previous Post

Quick Web Scraping using Gazpacho

Next Post

Analyzing the Income Level of US Census Data

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name*

Email*

Website

Notify me of follow-up comments by email Notify me of new posts by email.

Submit

Top Resources



10 Best AI Image Generator Tools to Use in 2023

avcontentteam - AUG 17, 2023



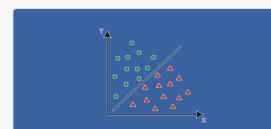
Understand Random Forest Algorithms With Examples (Updated 2023)

Sruthi E R - JUN 17, 2021



How to Read and Write With CSV Files in Python?

Harika Bonthu -
AUG 21, 2021



Guide on Support Vector Machine (SVM) Algorithm

Anshul Saini - OCT 12, 2021

