

pose estimation

1.Introduction

Posture estimation includes locating various body parts (also known as key points) that form a skeletal topology (also known as links). There are various applications for posture estimation, including gestures, AR/VR, HMI (Human Machine Interface), and posture/gait correction. A pre trained model is provided for human and hand pose estimation, which can detect multiple individuals in each frame.

The poseNet object accepts images as input and outputs a list of object poses. Each object pose contains a list of detected key points, as well as their positions and links between key points. You can query these to find specific features. PoseNet can be used from Python and C++.

As an example of using the poseNet class, there are sample programs in C++and Python. These samples can detect the poses of multiple individuals in images, videos, and cameras. More information about the various types of input/output streams supported.

2.Image pose estimation

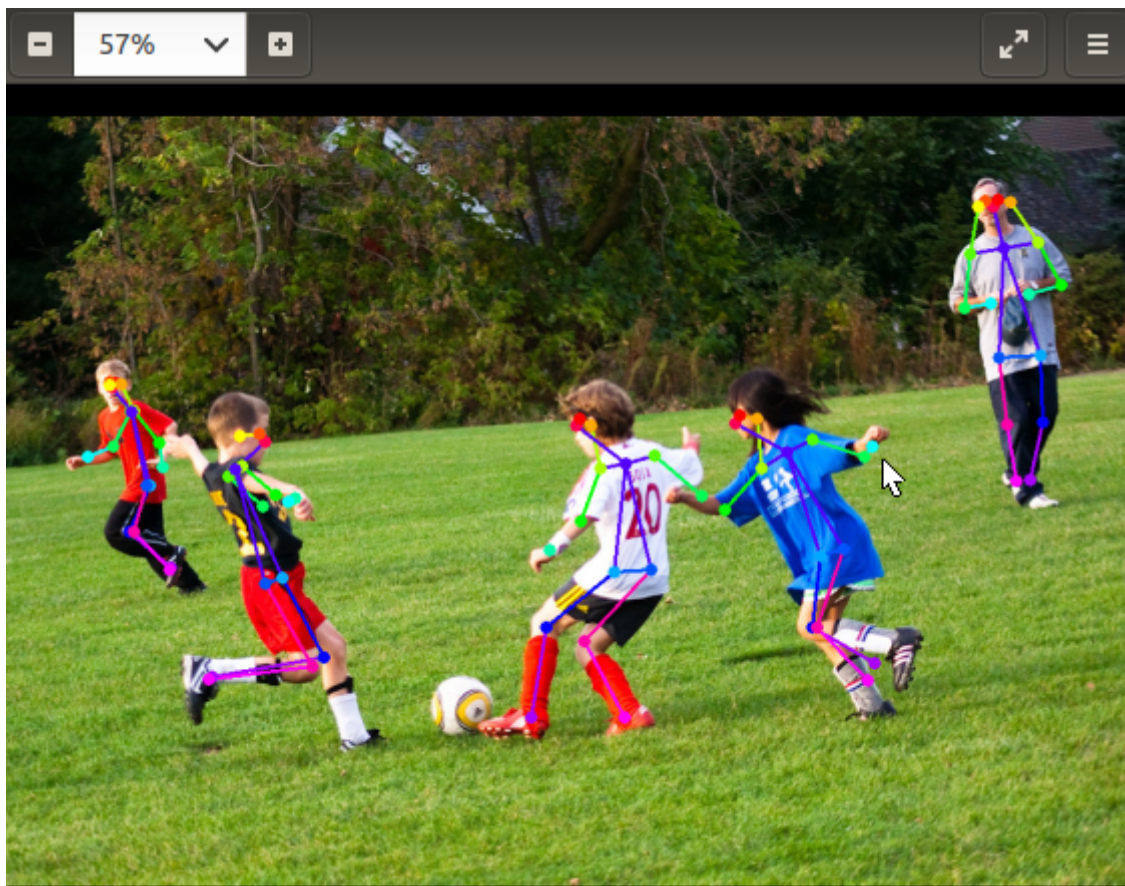
Firstly, let's try running the posenet example on some sample images. It is recommended to save the output image to the directory where images/test is mounted. Then, under Jetson inference/data/images/test, these images can be easily viewed from the host deviceAfter building the project, please ensure that your terminal is located in the aarch64/bin directory:

```
cd jetson-inference/build/aarch64/bin
```

Here are some examples of using the default pose ResNet18 Body model for human pose estimation:

```
# C++
$ ./posenet images/humans_4.jpg images/test/pose_humans_%i.jpg

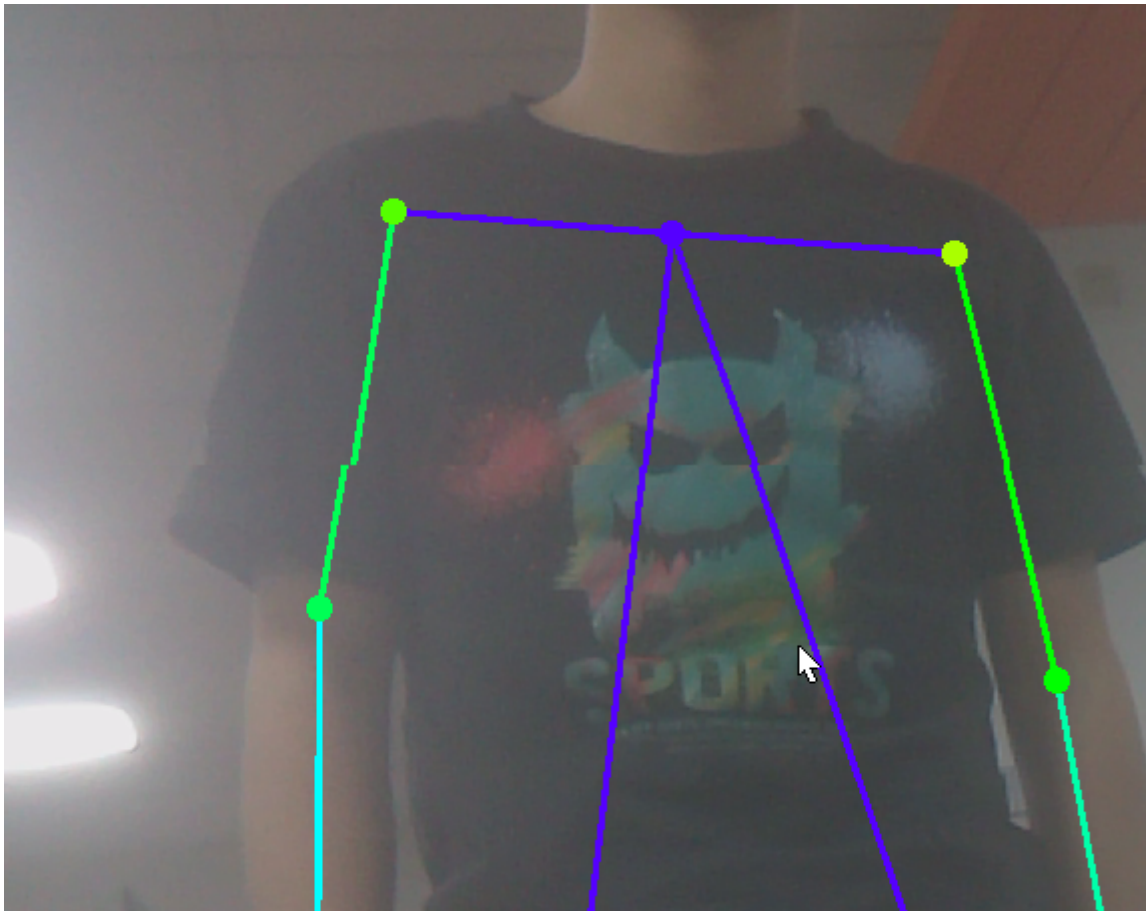
# Python
$ ./posenet.py images/humans_4.jpg images/test/pose_humans_%i.jpg
```



3.Video pose estimation

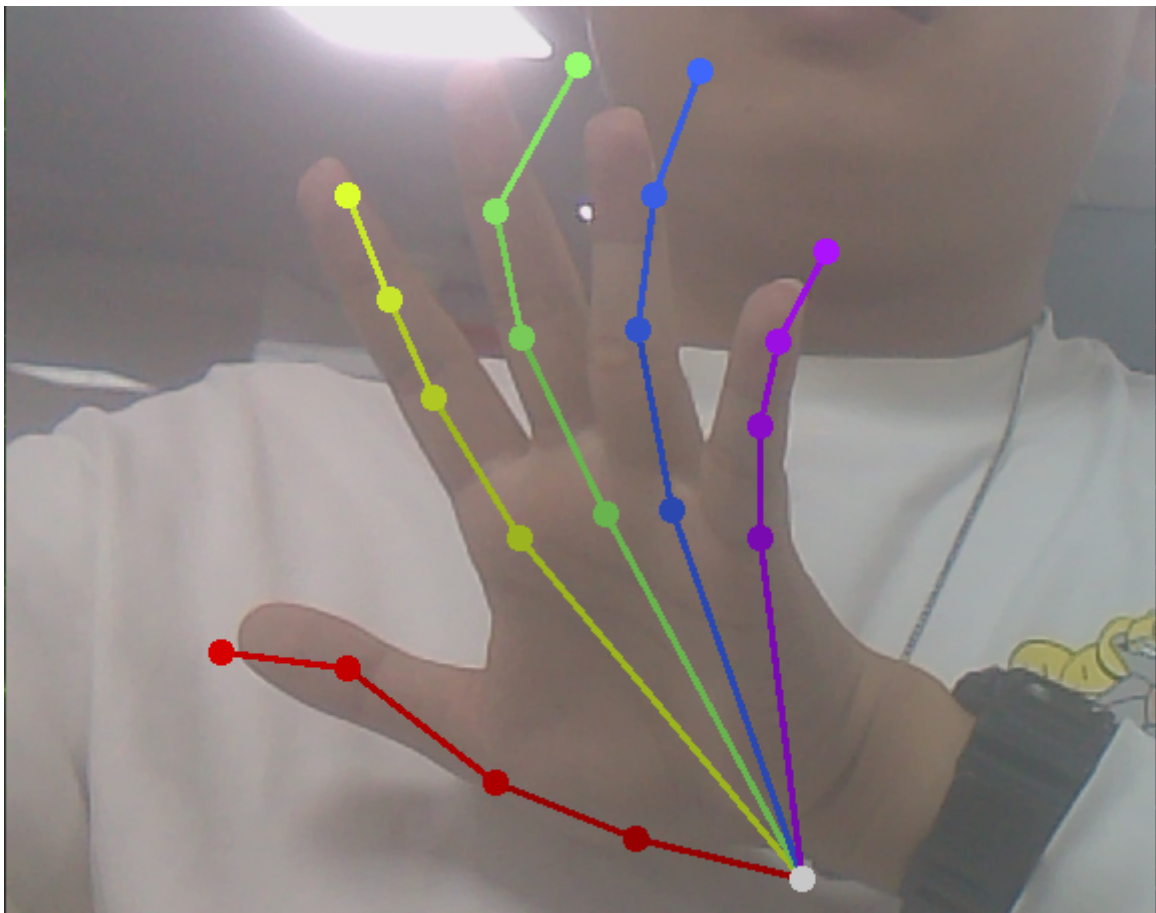
To run pose estimation on real-time camera streams or videos, enter the device or file path from the "Camera Streaming and Multimedia" page.

```
# C++  
$ ./posenet /dev/video0      # csi://0 if using MIPI CSI camera  
  
# Python  
$ ./posenet.py /dev/video0  # csi://0 if using MIPI CSI camera
```



There is also a hand recognition:

```
# C++  
$ ./posenet --network=resnet18-hand /dev/video0  
  
# Python  
$ ./posenet.py --network=resnet18-hand /dev/video0
```



Note: If you are building your own environment, you must download the model file to the network folder to run itFace program. By using the image we provide, you can directly input the above program