1. Human Posture Estimation

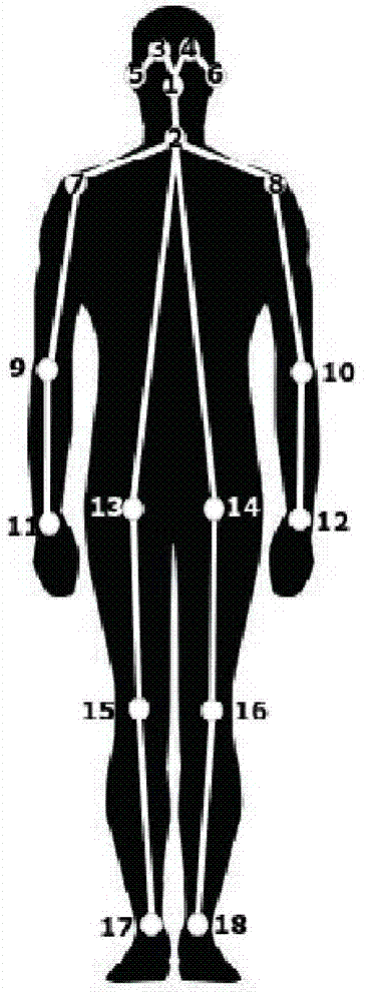
1. Human Posture Estimation

Note: The AI camera in this case does not have computing power bonus, and is called as a normal camera!

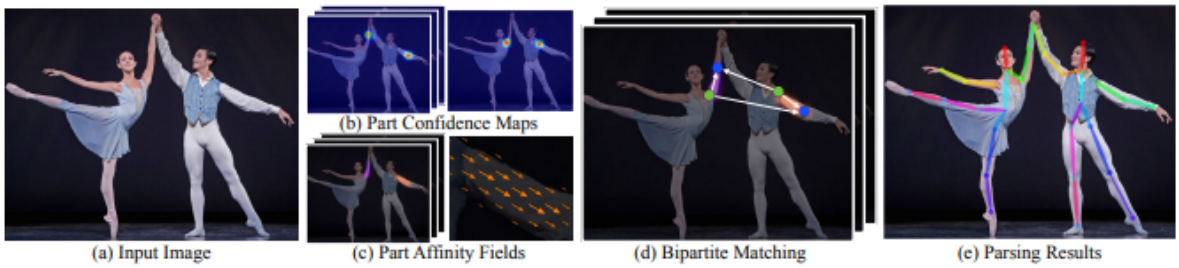
Source code path: /home/pi/yahboomcar_ws/src/yahboomcar_visual/detection

1.1. Overview

Human Posture Estimation estimates the human posture by correctly connecting the key points of the human body detected in the image. The key points of the human body usually correspond to joints with a certain degree of freedom on the human body, such as the neck, shoulder, elbow, wrist, waist, knee, ankle, etc., as shown in the figure below.



1.2, Principle



Input an image, extract features through the convolutional network, get a set of feature maps, and then split into two forks, using the CNN network to extract Part Confidence Maps and Part Affinity Fields respectively;
After obtaining these two pieces of information, we use Bipartite Matching in graph theory to find Part Association, connect the joints of the same person, and due to the vector nature of PAF itself, the generated bipartite matching is very correct, and finally merged into a person's overall skeleton;
Finally, Multi-Person Parsing based on PAFs—>Convert the Multi-person parsing problem into a graphs problem—>Hungarian Algorithm (Hungarian algorithm)
(The Hungarian algorithm is the most common algorithm for partial graph matching. The core of the algorithm is to find augmenting paths. It is an algorithm that uses augmenting paths to find the maximum matching of bipartite graphs.)

1.3. Start

```
cd /home/pi/yahboomcar_ws/src/yahboomcar_visual/detection
python target_detection_CSI.py
```

After clicking the image box, use the keyboard [f] key to switch target detection.

```
if action == ord('f') or action == ord('F'):state = not state
```

Input image

Output image


84.90ms