2. Human posture estimation

2. Human posture estimation
2. Human posture estimation
2.1. Overview
2.2. Principle
2.3.start

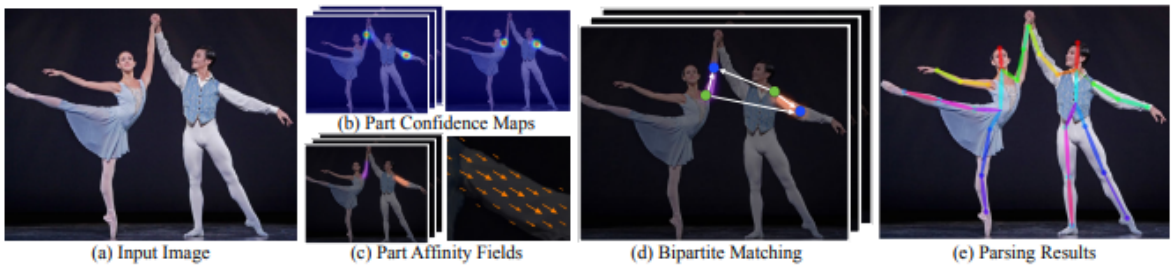
2. Human posture estimation

Source code path: ~/yahboomcar_ws/src/yahboomcar_visual/detection

2.1. Overview

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

2.2. Principle



Input an image, extract features through a convolutional network, obtain a set of feature maps, and then divide into two forks, use the CNN network to extract Part Confidence Maps and Part Affinity Fields respectively;
After obtaining these two pieces of information, we use Bipartite Matching (even matching) in graph theory to find the Part Association, and connect the joint points of the same person. Due to the vector nature of PAF itself, the generated even matching is very correct, and finally merged For the overall skeleton of a person;
Finally, find 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 the augmented path. It is an algorithm that uses the augmented path to find the maximum matching of a bipartite graph.)

2.3.start

```
cd ~/yahboomcar_ws/src/yahboomcar_visual/detection
python target_detection.py
```

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

```
if action == ord('f') or action == ord('F'):state = not state  # 功能切换
```

input image

output image