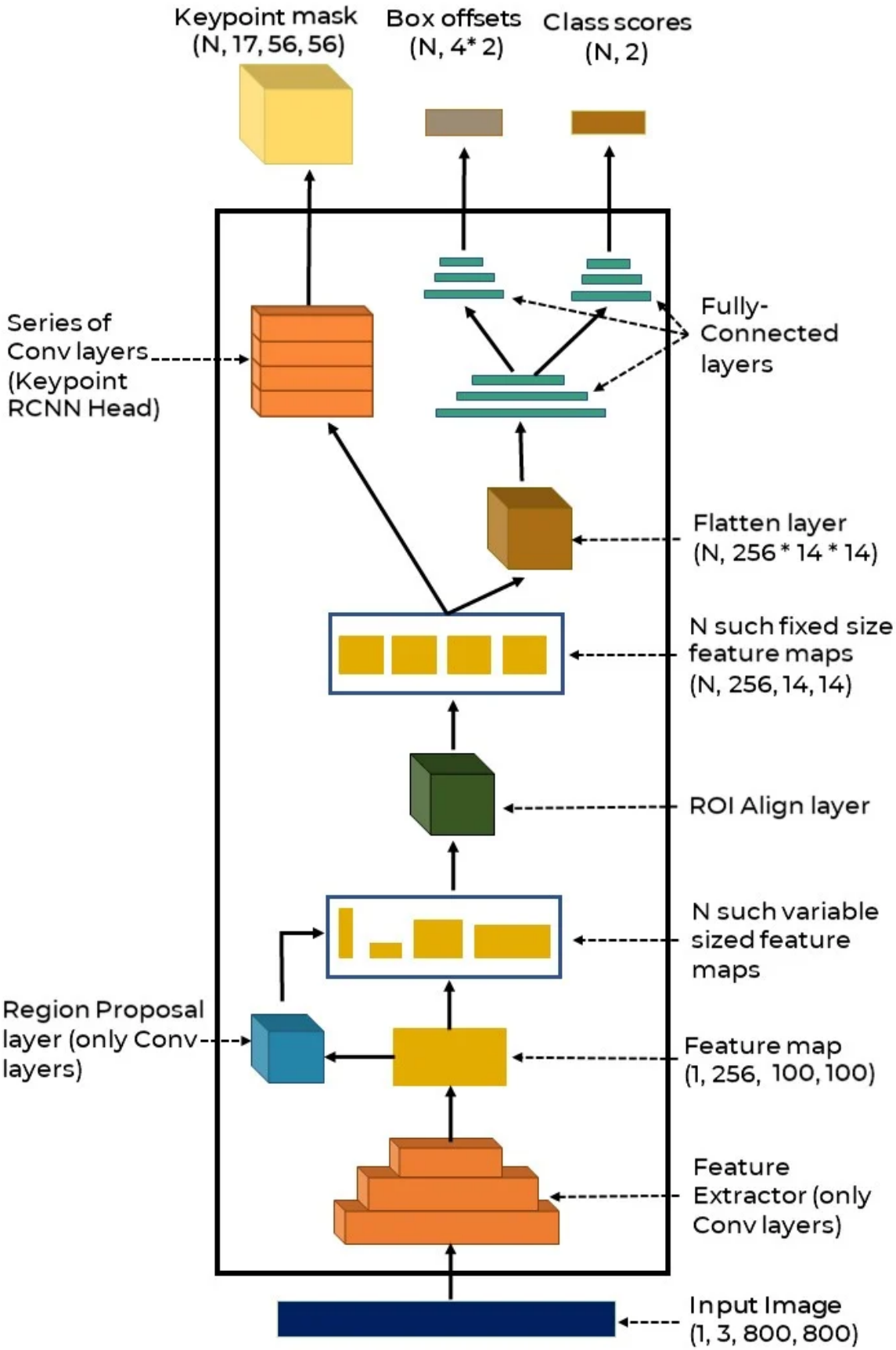
Human Pose Estimation using Keypoint RCNN in PyTorch笔记

- Blog: Human Pose Estimation using Keypoint RCNN in PyTorch
- Code: spmallick/learnopencv/PyTorch-Keypoint-RCNN

Table of Contents

Evolution of Keypoint RCNN Architecture

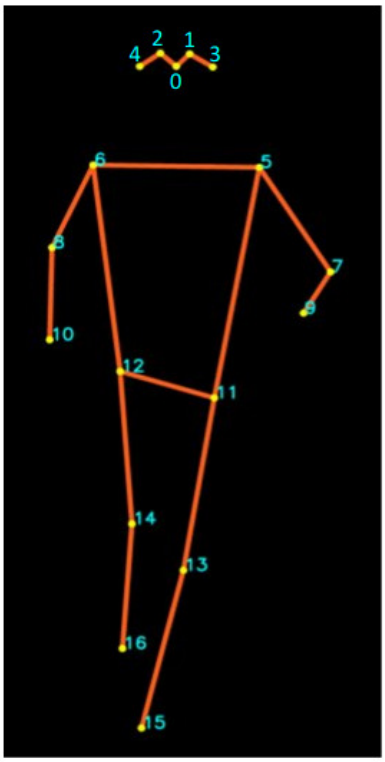
1. `N` is the number of objects proposed by the Region-Proposal Layer.
2. `C` is `2`, the MS-COCO Dataset offers keypoints only for the person class.
3. `K` is the number of keypoints per person, which is `17`.



KEYPOINT RCNN

- The output from Keypoint-RCNN is now sized `[N, K=17, 56, 56]`. Each of the K channels corresponds to a specific keypoint (for eg: left-elbow, right-ankle etc).
- The final class-scores will be of size `[N, 2]`:
  - one for background
  - the other for the person class
- The box-predictions will be sized `[N, 2 * 4]`.

Index	Key point
0	Nose
1	Left-eye
2	Right-eye
3	Left-ear
4	Right-ear
5	Left-shoulder
6	Right-shoulder
7	Left-elbow
8	Right-elbow
9	Left-wrist
10	Right-wrist
11	Left-hip
12	Right-hip
13	Left-knee
14	Right-knee
15	Left-ankle
16	Right-ankle



Input Output Format

Input to the model a tensor of size `[batch_size, 3, height, width]`. Note that the original image should be normalized (i.e. the pixel values in the image should range between 0 and 1).

```
output = model([img_tensor])[0]
```

The variable `output` is a dictionary, with the following keys and values:

- ***boxes*** – A tensor of size `[N, 4]`, where `N` is the number of objects detected.
- ***labels*** – A tensor of size `[N]`, depicting the class of the object.
  - This is always 1 because each detected box belongs to a person.
  - 0 stands for the background class.
- ***scores*** – A tensor of size `[N]`, depicting the confidence score of the detected object.
- ***keypoints*** – A tensor of size `[N, 17, 3]`, depicting the 17 joint locations of `N` number of persons. Out of 3, the first two numbers are the coordinates `x` and `y`, and the third one depicts the visibility.
  - 0, when keypoint is invisible
  - 1, when keypoint is visible
- ***keypoints_scores*** – A tensor of size `[N, 17]`, depicting the score for all the keypoints, for each detected person.

Loss Function in Keypoint-RCNN

$$\frac{-\sum_{h,w}\left[Y_{k,h,w}==1\right]\left(Y_{k,h,w}*\log\left(\text{softmax}\left(\widehat{Y}_{k,h,w}\right)\right)\right)}{\sum_{h,w}\left[Y_{k,h,w}==1\right]}$$

Evaluation Metric in Keypoint Detection

- Object Keypoint Similarity (OKS)

$$OKS=\exp\left(-\frac{d_i^2}{2s^2k_i^2}\right)$$

Where `d_i` is the Euclidean distance between predicted and ground-truth, `s` is the object's scale, and `k_i` is a constant for a specific keypoint.

How to fix the values for `k`?

Well, as we mentioned earlier, k is a constant factor for each keypoint and it remains the same for all samples. It turns out that k is a measure of standard-deviation of a particular keypoint. Essentially, the value of `k` for keypoints on the face (eyes, ears, nose) have a relatively smaller standard deviation than the keypoints on the body (hips, shoulders, knees).

更新于2021-06-22