Rogez, Weinzaepfel, Schmid

LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images

1 Summary

The contribution of this paper is a joint, end-to-end method for multi-person 2D and 3D human pose estimation. The eponymous LCR-Net architecture contains a *localization* component (which produces pose proposals at different locations in the image), a *classification* component (which scores the pose proposals), and a *regression* component (which refines the pose proposals in both 2D and 3D). All components originate from the same series of convolutional layers.

The architecture can be seen below (*reference: the original paper*).

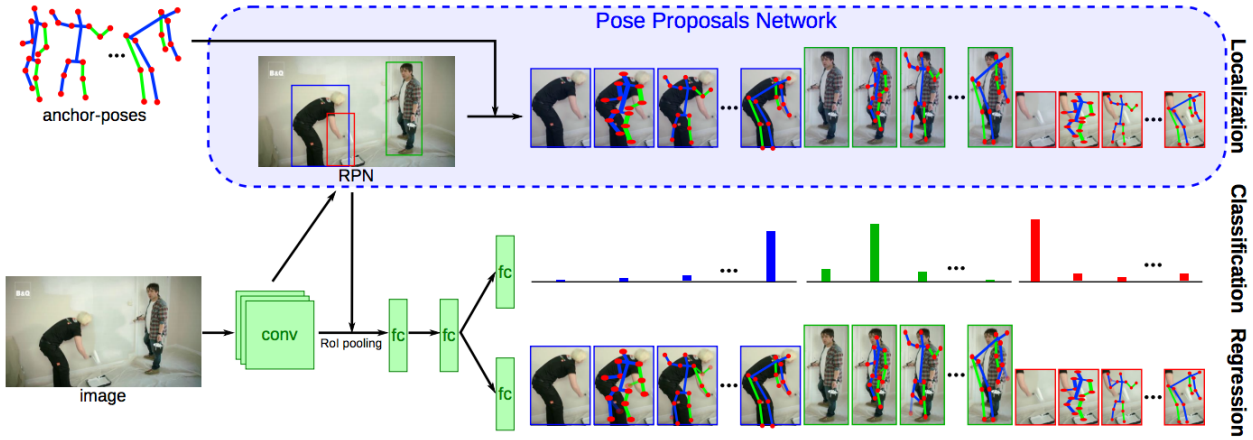

Fig. 2. Overview of our LCR-Net architecture (poses only shown in 2D for better readability). We first extract candidate regions using a Region Proposal Network (RPN) and obtain pose proposals by placing a fixed set of anchor-poses into these boxes (top). These pose proposals are then scored by a classification branch and refined using class-specific regressors, learned independently for each anchor-pose.

source: Rogez, Weinzaepfel, and Schmid in [1]

2 Introduction

- **Method**: sample, score, refine, and combine *pose proposals* to estimate both *location* (global root) and the 2D/3D pose of all persons in the image. This method recovers full-body poses even in cases with occlusion.
- CNN approaches have previously involved either **regression** or **classification**.
  - **regression**: directly estimate the 2D or 3D location of the joints
    - fails for multi-modal outputs, tends toward average of modes
  - **classification**: define pose classes, return average pose of class with highest score
- This method *combines* regression and classification. In it, we divide the space of valid full-body poses into k partitions and jointly train
  (a) a k-way classifier on this space
  (b) one pose regression model per class
- Then we have *coarse pose classification* and *class-specific pose regression*.

- The overall algorithm is as follows:
  1. Extract candidate regions for person localization.
  2. *Obtain pose proposals*: locate set of k hypothetical pose classes (anchor-poses) in the candidate regions.
  3. Score each pose proposal using classification branch of network.
  4. For each pose proposal, independently regress a pose for each anchor-pose.
  5. Aggregate similar pose proposals to obtain final pose estimation.

3 Background: Monocular 3D Human Pose Estimation

- Multiple paradigms:
  (a) compute 2D poses, then use them to estimate 3D poses (*limited by 2D pose estimator*)
  (b) directly learn mapping from image to 3D poses
  (c) jointly/iteratively learn 2D and 3D (*e.g. using multi-stage CNN architecture*)

4 LCR-Net **Localization-Classification-Regression Network**

- Define **human pose** as (p, P), where p is the 2D pixel coordinates of each joint and P is the 3D location of each joint relative to the body center.
- Assume fixed set of k 2D/3D anchor-poses $\{(a_i, A_i)\}_{i=1,\dots,k}$.

The method, one more time:

1. Extract convolutional features from image.
2. **Localization** (pose proposals network):
   - includes an RPN; outputs a list of *pose proposals*
     - *pose proposal*: set of candidate locations where the anchor-poses are hypothesized to be
   - use RoI pooling to aggregate features inside each candidate region
   - pass through two fully connected layers, then branch into two components:
     3. **Classification:**
        - estimate probability for each anchor-pose of being correct at each location
     4. **Regression:**
        - estimate difference between true human pose and pose proposal for each anchor-pose
3. Loss: $\mathcal{L}_{\text{localization}} + \mathcal{L}_{\text{classification}} + \mathcal{L}_{\text{regression}}$
4. End-to-end training for localizing humans and estimating their 2D/3D poses!

4.1 Localization: Pose Proposals Network

- Outputs a set of $n \times k$ pose proposals, i.e. 2D/3D pose hypotheses obtained by placing the k anchor-poses in the n bounding boxes generated by the region proposal network (RPN)

4.2 Classification

- Assigns a probability for each anchor-pose to each bounding box. In doing so, we predict the best-fit anchor-pose, i.e. the correct label, for each bounding box.
- Do iterative estimation similar to convolutional pose machines.

4.3 Regression

- Refines coarse anchor-poses located in region proposals (one regressor for each anchor-pose). Estimates updates for both 2D and 3D positions.
- Do iterative estimation similar to convolutional pose machines.

4.4 Pose Proposals Integration

- The LCR-Net outputs a set of refined pose proposals with associated classification scores.
- Integration process (to turn this output into a final set of poses):
  1. Re-score proposals; penalize ones with joints outside of the candidate bounding box.
  2. Aggregate proposals which are close in terms of image location and 3D pose.

5 Clarifications

- There are originally $n \times k$ pose proposals, and not all of the anchor-poses will apply to each candidate region. So when do we filter out the bad ones? Do we only use one anchor-pose per bounding box, and is it the refined highest-scoring anchor-pose?
  - This is was my first interpretation – and then there would be n of these pose proposals, for n the number of bounding boxes (basically, we'd reduce the $n \times k$ pose proposals over anchor-poses).
  - However, upon a further inspection, I think that *all* of the $n \times k$ pose proposals are passed to the PPI stage, where they are grouped according to 2D IoU, then reduced within groups, then thresholded according to score. So the "filtering" would correspond to the thresholding.

References

[1] Grgory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images. *arXiv preprint arXiv:1803.00455 (2018).*