PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model

George Papandreou(✉), Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy

Google Research, Los Angeles, USA
gpapan@google.com, tylerzhu@google.com, lcchen@google.com, spyrosg@google.com, tompson@google.com, kpmurphy@google.com

Abstract. We present a box-free bottom-up approach for the tasks of pose estimation and instance segmentation of people in multi-person images using an efficient single-shot model. The proposed PersonLab model tackles both semantic-level reasoning and object-part associations using part-based modeling. Our model employs a convolutional network which learns to detect individual keypoints and predict their relative displacements, allowing us to group keypoints into person pose instances. Further, we propose a part-induced geometric embedding descriptor which allows us to associate semantic person pixels with their corresponding person instance, delivering instance-level person segmentations. Our system is based on a fully-convolutional architecture and allows for efficient inference, with runtime essentially independent of the number of people present in the scene. Trained on COCO data alone, our system achieves COCO test-dev keypoint average precision of 0.665 using single-scale inference and 0.687 using multi-scale inference, significantly outperforming all previous bottom-up pose estimation systems. We are also the first bottom-up method to report competitive results for the person class in the COCO instance segmentation task, achieving a person category average precision of 0.417.

Keywords: Person detection and pose estimation
Segmentation and grouping

1 Introduction

The rapid recent progress in computer vision has allowed the community to move beyond classic tasks such as bounding box-level face and body detection towards

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01264-9_17) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11218, pp. 282–299, 2018.
https://doi.org/10.1007/978-3-030-01264-9_17

more detailed visual understanding of people in unconstrained environments. In this work we tackle in a unified manner the tasks of multi-person detection, 2-D pose estimation, and instance segmentation. Given a potentially cluttered and crowded 'in-the-wild' image, our goal is to identify every person instance, localize its facial and body keypoints, and estimate its instance segmentation mask. A host of computer vision applications such as smart photo editing, person and activity recognition, virtual or augmented reality, and robotics can benefit from progress in these challenging tasks.

There are two main approaches for tackling multi-person detection, pose estimation and segmentation. The *top-down* approach starts by identifying and roughly localizing individual person instances by means of a bounding box object detector, followed by single-person pose estimation or binary foreground/ background segmentation in the region inside the bounding box. By contrast, the *bottom-up* approach starts by localizing identity-free semantic entities (individual keypoint proposals or semantic person segmentation labels, respectively), followed by grouping them into person instances. In this paper, we adopt the latter approach. We develop a box-free fully convolutional system whose computational cost is essentially independent of the number of people present in the scene and only depends on the cost of the CNN feature extraction backbone.

In particular, our approach first predicts all keypoints for every person in the image in a fully convolutional way. We also learn to predict the relative displacement between each pair of keypoints, also proposing a novel recurrent scheme which greatly improves the accuracy of long-range predictions. Once we have localized the keypoints, we use a greedy decoding process to group them into instances. Our approach starts from the most confident detection, as opposed to always starting from a distinguished landmark such as the nose, so it works well even in clutter.

In addition to predicting the sparse keypoints, our system also predicts dense instance segmentation masks for each person. For this purpose, we train our network to predict instance-agnostic semantic person segmentation maps. For every person pixel we also predict offset vectors to each of the K keypoints of the corresponding person instance. The corresponding vector fields can be thought as a geometric embedding representation and induce basins of attraction around each person instance, leading to an efficient association algorithm: For each pixel x_i, we predict the locations of all K keypoints for the corresponding person that x_i belongs to; we then compare this to all candidate detected people j (in terms of average keypoint distance), weighted by the keypoint detection probability; if this distance is low enough, we assign pixel i to person j.

We train our model on the standard COCO keypoint dataset [1], which annotates multiple people with 12 body and 5 facial keypoints. We significantly outperform the best previous bottom-up approach to keypoint localization [2], improving the keypoint AP from 0.655 to 0.687. In addition, we are the first bottom-up method to report competitive results on the person class for the COCO instance segmentation task. We get a mask AP of 0.417, which outperforms the strong top-down FCIS method of [3], which gets 0.386. Furthermore

our method is very simple and hence fast, since it does not require any second stage box-based refinement, or clustering algorithm. We believe it will therefore be quite useful for a variety of applications, especially since it lends itself to deployment in mobile phones.

2 Related Work

2.1 Pose Estimation

Proir to the recent trend towards deep convolutional networks [4,5], early successful models for human pose estimation centered around inference mechanisms on part-based graphical models [6,7], representing a person by a collection of configurable parts. Following this work, many methods have been proposed to develop tractable inference algorithms for solving the energy minimization that captures rich dependencies among body parts [8–16]. While the forward inference mechanism of this work differs to these early DPM-based models, we similarly propose a bottom-up approach for grouping part detections to person instances.

Recently, models based on modern large scale convolutional networks have achieved state-of-art performance on both single-person pose estimation [17–26] and multi-person pose estimation [27–34]. Broadly speaking, there are two main approaches to pose-estimation in the literature: top-down (person first) and bottom-up (parts first). Examples of the former include G-RMI [33], CFN [35], RMPE [36], Mask R-CNN [34], and CPN [37]. These methods all predict key point locations within person bounding boxes obtained by a person detector (*e.g.*, Fast-RCNN [38], Faster-RCNN [39] or R-FCN [40]).

In the bottom-up approach, we first detect body parts and then group these parts to human instances. Pishchulin *et al.* [27], Insafutdinov *et al.* [28,29], and Iqbal *et al.* [30] formulate the problem of multi-person pose estimation as part grouping and labeling via a Linear Program. Cao *et al.* [32] incorporate the unary joint detector modified from [31] with a part affinity field and greedily generate person instance proposals. Newell *et al.* [2] propose associative embedding to identify key point detections from the same person.

2.2 Instance Segmentation

The approaches for instance segmentation can also be categorized into the two top-down and bottom-up paradigms.

Top-down methods exploit state-of-art detection models to either classify mask proposals [41–47] or to obtain mask segmentation results by refining the bounding box proposals [3,34,48–51].

Ours is a bottom-up approach, in which we associate pixel-level predictions to each object instance. Many recent models propose similar forms of instance-level bottom-up clustering. For instance, Liang *et al.* use a proposal-free network [52] to cluster semantic segmentation results to obtain instance segmentation. Uhrig *et al.* [53] first predict each pixel's direction towards its instance center

and then employ template matching to decode and cluster the instance segmentation result. Zhang *et al.* [54,55] predict instance ID by encoding the object depth ordering within a patch and use this depth ordering to cluster instances. Wu *et al.* [56] use a prediction network followed by a Hough transform-like approach to perform prediction instance clustering. In this work, we similarly perform a Hough voting of multiple predictions. In a slightly different formulation, Liu *et al.* [57] segment and aggregate segmentation results from dense multi-scale patches, and aggregate localized patches into complete object instances. Levinkov *et al.* [58] formulate the instance segmentation problem as a combinatorial optimization problem that consists of graph decomposition and node labeling and propose efficient local search algorithms to iteratively refine an initial solution. InstanceCut [59] and the work of [60] propose to predict object boundaries to separate instances. [2,61,62] group pixel predictions that have similar values in the learned embedding space to obtain instance segmentation results. Bai and Urtasun [63] propose a Watershed Transform Network which produces an energy map where object instances are represented as basin. Liu *et al.* [64] propose the Sequential Grouping Network which decomposes the instance segmentation problem into several sub-grouping problems.

3 Methods

Figure 1 gives an overview of our system, which we describe in detail next.

3.1 Person Detection and Pose Estimation

We develop a box-free bottom-up approach for person detection and pose estimation. It consists of two sequential steps, detection of K keypoints, followed by grouping them into person instances. We train our network in a supervised fashion, using the ground truth annotations of the $K = 17$ face and body parts in the COCO dataset.

Keypoint Detection. The goal of this stage is to detect, in an instance-agnostic fashion, all visible keypoints belonging to any person in the image.

For this purpose, we follow the hybrid classification and regression approach of [33], adapting it to our multi-person setting. We produce heatmaps (one channel per keypoint) and offsets (two channels per keypoint for displacements in the horizontal and vertical directions). Let x_i be the 2-D position in the image, where $i = 1, \ldots N$ is indexing the position in the image and N is the number of pixels. Let $\mathcal{D}_R(y) = \{x : \|x - y\| \leq R\}$ be a disk of radius R centered around y. Also let $y_{j,k}$ be the 2-D position of the k-th keypoint of the j-th person instance, with $j = 1, \ldots, M$, where M is the number of person instances in the image.

For every keypoint type $k = 1, \ldots, K$, we set up a binary classification task as follows. We predict a heatmap $p_k(x)$ such that $p_k(x) = 1$ if $x \in \mathcal{D}_R(y_{j,k})$ for any person instance j, otherwise $p_k(x) = 0$. We thus have K independent dense binary classification tasks, one for each keypoint type. Each amounts to predicting a disk of radius R around a specific keypoint type of any person in

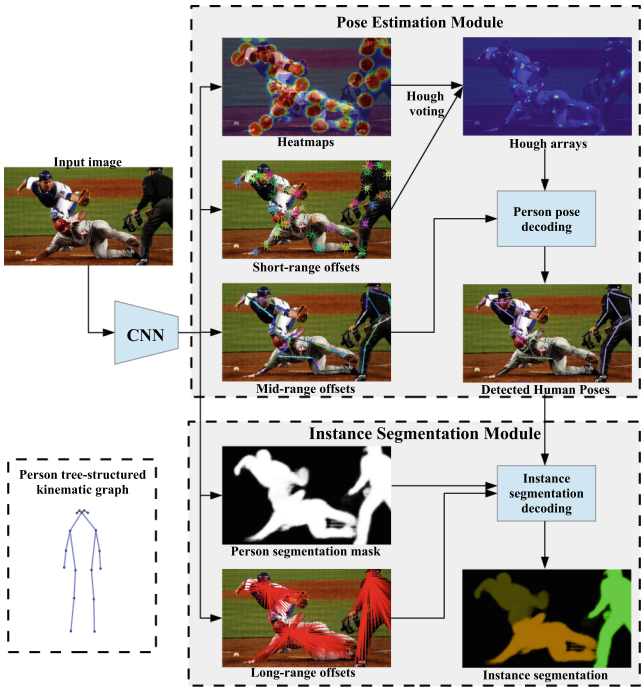


Fig. 1. Our PersonLab system consists of a CNN model that predicts: (1) keypoint heatmaps, (2) short-range offsets, (3) mid-range pairwise offsets, (4) person segmentation maps, and (5) long-range offsets. The first three predictions are used by the *Pose Estimation Module* in order to detect human poses while the latter two, along with the human pose detections, are used by the *Instance Segmentation Module* in order to predict person instance segmentation masks.

the image. The disk radius value is set to $R = 32$ pixels for all experiments reported in this paper and is independent of the person instance scale. We have deliberately opted for a disk radius which does not scale with the instance size in order to equally weigh all person instances in the classification loss. During training, we compute the heatmap loss as the average logistic loss along image positions and we back-propagate across the full image, only excluding areas that contain people that have not been fully annotated with keypoints (person crowd areas and small scale person segments in the COCO dataset).

In addition to the heatmaps, we also predict *short-range* offset vectors $S_k(x)$ whose purpose is to improve the keypoint localization accuracy. At each position x within the keypoint disks and for each keypoint type k, the short-range 2-D offset vector $S_k(x) = y_{j,k} - x$ points from the image position x to the k-th keypoint of the closest person instance j, as illustrated in Fig. 1. We generate K such vector fields, solving a 2-D regression problem at each image position and keypoint independently. During training, we penalize the short-range offset prediction errors with the L_1 loss, averaging and back-propagating the errors

only at the positions $x \in \mathcal{D}_R(y_{j,k})$ in the keypoint disks. We divide the errors in the short-range offsets (and all other regression tasks described in the paper) by the radius $R = 32$ pixels in order to normalize them and make their dynamic range commensurate with the heatmap classification loss.

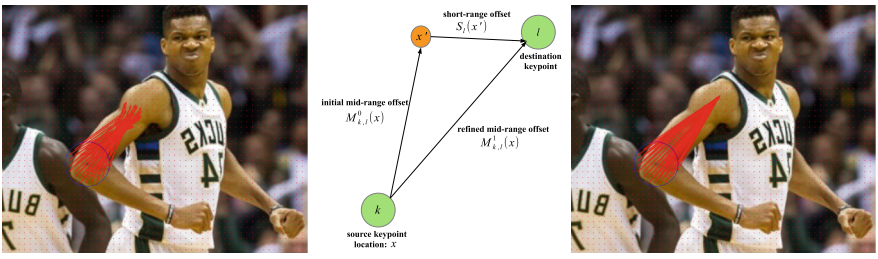
We aggregate the heatmap and short-range offsets via Hough voting into 2-D Hough score maps $h_k(x), k = 1, \ldots, K$, using independent Hough accumulators for each keypoint type. Each image position casts a vote to each keypoint channel k with weight equal to its activation probability,

$$h_k(x) = \frac{1}{\pi R^2} \sum_{i=1:N} p_k(x_i) B(x_i + S_k(x_i) - x)\,, \tag{1}$$

where $B(\cdot)$ denotes the bilinear interpolation kernel. The resulting highly localized Hough score maps $h_k(x)$ are illustrated in Fig. 1.

Grouping Keypoints into Person Detection Instances.

Mid-Range Pairwise Offsets. The local maxima in the score maps $h_k(x)$ serve as candidate positions for person keypoints, yet they carry no information about instance association. When multiple person instances are present in the image, we need a mechanism to "connect the dots" and group together the keypoints belonging to each individual instance. For this purpose, we add to our network a separate pairwise *mid-range* 2-D offset field output $M_{k,l}(x)$ designed to connect pairs of keypoints. We compute $2(K-1)$ such offset fields, one for each directed edge connecting pairs (k, l) of keypoints which are adjacent to each other in a tree-structured kinematic graph of the person, see Figs. 1 and 2. Specifically, the supervised training target for the pairwise offset field from the k-th to the l-th keypoint is given by $M_{k,l}(x) = (y_{j,l} - x)I(x \in \mathcal{D}_R(y_{j,k}))$, since its purpose is to allow us to move from the k-th to the l-th keypoint of the same person instance j. During training, this target regression vector is only defined if both keypoints are present in the training example. We compute the average L_1 loss of the regression prediction errors over the source keypoint disks $x \in \mathcal{D}_R(y_{j,k})$ and back-propagate through the network.



(a) (b) (c)

Fig. 2. Mid-range offsets. (a) Initial mid-range offsets that starting around the *RightElbow* keypoint, they point towards the *RightShoulder* keypoint. (b) Mid-range offset refinement using the short-range offsets. (c) Mid-range offsets after refinements.

Recurrent Offset Refinement. Particularly for large person instances, the edges of the kinematic graph connect pairs of keypoints such as *RightElbow* and *RightShoulder* which may be several hundred pixels away in the image, making it hard to generate accurate regressions. We have successfully addressed this important issue by recurrently refining the mid-range pairwise offsets using the more accurate short-range offsets, specifically:

$$M_{k,l}(x) \leftarrow x' + S_l(x') \,, \text{where } x' = M_{k,l}(x) \,, \tag{2}$$

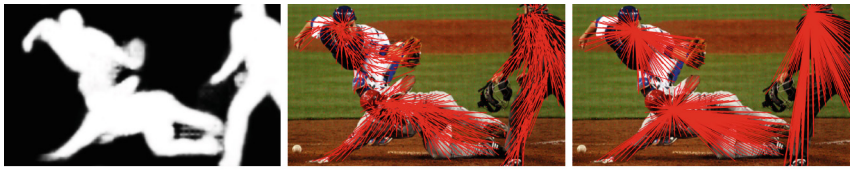
as illustrated in Fig. 2. We repeat this refinement step twice in our experiments. We employ bilinear interpolation to sample the short-range offset field at the intermediate position x' and back-propagate the errors through it along both the mid-range and short-range input offset branches. We perform offset refinement at the resolution of CNN output activations (before upsamling to the original image resolution), making the process very fast. The offset refinement process drastically decreases the mid-range regression errors, as illustrated in Fig. 2. This is a key novelty in our method, which greatly facilitates grouping and significantly improves results compared to previous papers [28,32] which also employ pairwise displacements to associate keypoints.

Fast Greedy Decoding. We have developed an extremely fast greedy decoding algorithm to group keypoints into detected person instances. We first create a priority queue, shared across all K keypoint types, in which we insert the position x_i and keypoint type k of all local maxima in the Hough score maps $h_k(x)$ which have score above a threshold value (set to 0.01 in all reported experiments). These points serve as candidate seeds for starting a detection instance. We then pop elements out of the queue in descending score order. At each iteration, if the position x_i of the current candidate detection seed of type k is within a disk $\mathcal{D}_r(y_{j',k})$ of the corresponding keypoint of previously detected person instances j', then we reject it; for this we use a non-maximum suppression radius of $r = 10$ pixels. Otherwise, we start a new detection instance j with the k-th keypoint at position $y_{j,k} = x_i$ serving as seed. We then follow the mid-range displacement vectors along the edges of the kinematic person graph to greedily connect pairs (k, l) of adjacent keypoints, setting $y_{j,l} = y_{j,k} + M_{k,l}(y_{j,k})$.

It is worth noting that our decoding algorithm does not treat any keypoint type preferentially, in contrast to other techniques that always use the same keypoint type (*e.g. Torso* or *Nose*) as seed for generating detections. Although we have empirically observed that the majority of detections in frontal facing person instances start from the more easily localizable facial keypoints, our approach can also handle robustly cases where a large portion of the person is occluded.

Keypoint- and Instance-Level Detection Scoring. We have experimented with different methods to assign a keypoint- and instance-level score to the detections generated by our greedy decoding algorithm. Our first keypoint-level scoring method follows [33] and assigns to each keypoint a confidence score $s_{j,k} = h_k(y_{j,k})$. A drawback of this approach is that the well-localizable facial keypoints typically receive much higher scores than poorly localizable keypoints

like the hip or knee. Our second approach attempts to calibrate the scores of the different keypoint types. It is motivated by the object keypoint similarity (OKS) evaluation metric used in the COCO keypoints task [1], which uses different accuracy thresholds κ_k to penalize localization errors for different keypoint types.



(a) (b) (c)

Fig. 3. Long-range offsets defined in the person segmentation mask. (a) Estimated person segmentation map. (b) Initial long range offsets for the *Nose* destination keypoint: each pixel in the foreground of the person segmentation mask points towards the *Nose* keypoint of the instance that it belongs to. (c) Long-range offsets after their refinements with the short-range offsets.

Specifically, consider a detected person instance j with keypoint coordinates $y_{j,k}$. Let λ_j be the square root of the area of the bounding box tightly containing all detected keypoints of the j-th person instance. We define the *Expected-OKS* score for the k-th keypoint by

$$s_{j,k} = E\{OKS_{j,k}\} = p_k(y_{j,k}) \int_{x \in \mathcal{D}_R(y_{j,k})} \hat{h}_k(x) \exp\left(-\frac{(x - y_{j,k})^2}{2\lambda_j^2 \kappa_k^2}\right) dx\,, \quad (3)$$

where $\hat{h}_k(x)$ is the Hough score normalized in $\mathcal{D}_R(y_{j,k})$. The expected OKS keypoint-level score is the product of our confidence that the keypoint is present, times the OKS localization accuracy confidence, given the keypoint's presence.

We use the average of the keypoint scores as instance-level score $s_j^h = (1/K)\sum_k s_{j,k}$, followed by non-maximum suppression (NMS). We have experimented both with hard OKS-based NMS [33] as well as a soft-NMS scheme adapted for the keypoints tasks from [65], where we use as final instance-level score the sum of the scores of the keypoints that have not already been claimed by higher scoring instances, normalized by the total number of keypoints:

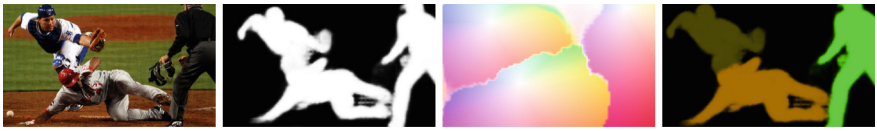
$$s_j = (1/K) \sum_{k=1:K} s_{j,k}[\|y_{j,k} - y_{j',k}\| > r, \text{for every } j' < j]\,, \quad (4)$$

where $r = 10$ is the NMS-radius. In our experiments in the main paper we report results with the best performing Expected-OKS scoring and soft-NMS but we include ablation experiments in the supplementary material.

3.2 Instance-Level Person Segmentation

Given the set of keypoint-level person instance detections, the task of our method's segmentation stage is to identify pixels that belong to people (recognition) and associate them with the detected person instances (grouping).

We describe next the respective semantic segmentation and association modules, illustrated in Fig. 4.



(a) (b) (c) (d)

Fig. 4. From semantic to instance segmentation: (a) Image; (b) person segmentation; (c) basins of attraction defined by the long-range offsets to the *Nose* keypoint; (d) instance segmentation masks.

Semantic Person Segmentation. We treat semantic person segmentation in the standard fully-convolutional fashion [66,67]. We use a simple semantic segmentation head consisting of a single 1×1 convolutional layer that performs dense logistic regression and compute at each image pixel x_i the probability $p_S(x_i)$ that it belongs to at least one person. During training, we compute and backpropagate the average of the logistic loss over all image regions that have been annotated with person segmentation maps (in the case of COCO we exclude the crowd person areas).

Associating Segments with Instances Via Geometric Embeddings. The task of this module is to associate each person pixel identified by the semantic segmentation module with the keypoint-level detections produced by the person detection and pose estimation module.

Similar to [2,61,62], we follow the embedding-based approach for this task. In this framework, one computes an embedding vector $G(x)$ at each pixel location, followed by clustering to obtain the final object instances. In previous works, the representation is typically learned by computing pairs of embedding vectors at different image positions and using a loss function designed to attract the two embedding vectors if they both come from the same object instance and repel them if they come from different person instances. This typically leads to embedding representations which are difficult to interpret and involves solving a hard learning problem which requires careful selection of the loss function and tuning several hyper-parameters such as the pair sampling protocol.

Here, we opt instead for a considerably simpler, geometric approach. At each image position x inside the segmentation mask of an annotated person instance j with 2-D keypoint positions $y_{j,k}, k = 1, \ldots, K$, we define the *long-range offset* vector $L_k(x) = y_{j,k} - x$ which points from the image position x to the position of the k-th keypoint of the corresponding instance j. (This is very similar to the short-range prediction task, except the dynamic range is different, since we require the network to predict from any pixel inside the person, not just from inside a disk near the keypoint. Thus these are like two "specialist" networks. Performance is worse when we use the same network for both kinds of tasks.) We

compute K such 2-D vector fields, one for each keypoint type. During training, we penalize the long-range offset regression errors using the L_1 loss, averaging and back-propagating the errors only at image positions x which belong to a single person object instance. We ignore background areas, crowd regions, and pixels which are covered by two or more person masks.

The long-range prediction task is challenging, especially for large object instances that may cover the whole image. As in Sect. 3.1, we recurrently refine the long-range offsets, twice by themselves and then twice by the short-range offsets

$$L_k(x) \leftarrow x' + L_k(x')\,, x' = L_k(x) \text{ and } L_k(x) \leftarrow x' + S_k(x')\,, x' = L_k(x)\,, \quad (5)$$

back-propagating through the bilinear warping function during training. Similarly with the mid-range offset refinement in Eq. 2, recurrent long-range offset refinement dramatically improves the long-range offset prediction accuracy.

In Fig. 3 we illustrate the long-range offsets corresponding to the *Nose* keypoint as computed by our trained CNN for an example image. We see that the long-range vector field effectively partitions the image plane into basins of attraction for each person instance. This motivates us to define as embedding representation for our instance association task the $2 \cdot K$ dimensional vector $G(x) = (G_k(x))_{k=1,\ldots,K}$ with components $G_k(x) = x + L_k(x)$.

Our proposed embedding vector has a very simple geometric interpretation: At each image position x_i semantically recognized as a person instance, the embedding $G(x_i)$ represents our local estimate for the absolute position of every keypoint of the person instance it belongs to, i.e., it represents the predicted shape of the person. This naturally suggests shape metric as candidates for computing distances in our proposed embedding space. In particular, in order to decide if the person pixel x_i belongs to the j-th person instance, we compute the embedding distance metric

$$D_{i,j} = \frac{1}{\sum_k p_k(y_{j,k})} \sum_{k=1}^{K} p_k(y_{j,k}) \frac{1}{\lambda_j} \|G_k(x_i) - y_{j,k}\|\,, \quad (6)$$

where $y_{j,k}$ is the position of the k-th detected keypoint in the j-th instance and $p_k(y_{j,k})$ is the probability that it is present. Weighing the errors by the keypoint presence probability allows us to discount discrepancies in the two shapes due to missing keypoints. Normalizing the errors by the detected instance scale λ_j allows us to compute a scale invariant metric. We set λ_j equal to the square root of the area of the bounding box tightly containing all detected keypoints of the j-th person instance. We emphasize that because we only need to compute the distance metric between the N_S pixels and the M person instances, our algorithm is very fast in practice, having complexity $\mathcal{O}(N_S * M)$ instead of $\mathcal{O}(N_S * N_S)$ of standard embedding-based segmentation techniques which, at least in principle, require computation of embedding vector distances for all pixel pairs.

To produce the final instance segmentation result: (1) We find all positions x_i marked as person in the semantic segmentation map, *i.e.* those pixels that have

semantic segmentation probability $p_S(x_i) \geq 0.5$. (2) We associate each person pixel x_i with every detected person instance j for which the embedding distance metric satisfies $D_{i,j} \leq t$; we set the relative distance threshold $t = 0.25$ for all reported experiments. It is important to note that the pixel-instance assignment is non-exclusive: Each person pixel may be associated with more than one detected person instance (which is particularly important when doing soft-NMS in the detection stage) or it may remain an orphan (*e.g.*, a small false positive region produced by the segmentation module). We use the same instance-level score produced by the previous person detection and pose estimation stage to also evaluate on the COCO segmentation task and obtain average precision performance numbers.

3.3 Imputing Missing Keypoint Annotations

The standard COCO dataset does not contain keypoint annotations in the training set for the small person instances, and ignores them during model evaluation. However, it contains segmentation annotations and evaluates mask predictions for those small instances. Since training our geometric embeddings requires keypoint annotations for training, we have run the single-person pose estimator of [33] (trained on COCO data alone) in the COCO training set on image crops around the ground truth box annotations of those small person instances to impute those missing keypoint annotations. We treat those imputed keypoints as regular training annotations during our PersonLab model training. Naturally, this missing keypoint imputation step is particularly important for our COCO instance segmentation performance on small person instances. We emphasize that, unlike [68], we do not use any data beyond the COCO *train* split images and annotations in this process. Data distillation on additional images as described in [68] may yield further improvements.

4 Experimental Evaluation

4.1 Experimental Setup

Dataset and Tasks. We evaluate the proposed PersonLab system on the standard COCO keypoints task [1] and on COCO instance segmentation [69] for the person class alone. For all reported results we only use COCO data for model training (in addition to Imagenet pretraining). Our *train* set is the subset of the 2017 COCO training set images that contain people (64115 images). Our *val* set coincides with the 2017 COCO validation set (5000 images). We only use *train* for training and evaluate on either *val* or the *test-dev* split (20288 images).

Model Training Details. We report experimental results with models that use either ResNet-101 or ResNet-152 CNN backbones [70] pretrained on the Imagenet classification task [71]. We discard the last Imagenet classification layer and add 1×1 convolutional layers for each of our model-specific layers. During model training, we randomly resize a square box tightly containing the full

Table 1. Performance on the COCO keypoints **test-dev** split.

	AP	$AP^{.50}$	$AP^{.75}$	AP^M	AP^L	AR	$AR^{.50}$	$AR^{.75}$	AR^M	AR^L
Bottom-up methods:										
CMU-Pose [32] (+refine)	0.618	0.849	0.675	0.571	0.682	0.665	0.872	0.718	0.606	0.746
Assoc. Embed. [2] (multi-scale)	0.630	0.857	0.689	0.580	0.704	-	-	-	-	-
Assoc. Embed. [2] (mscale, refine)	0.655	0.879	0.777	0.690	0.752	0.758	0.912	0.819	0.714	0.820
Top-down methods:										
Mask-RCNN [34]	0.631	0.873	0.687	0.578	0.714	0.697	0.916	0.749	0.637	0.778
G-RMI *COCO-only* [33]	0.649	0.855	0.713	0.623	0.700	0.697	0.887	0.755	0.644	0.771
PersonLab (ours):										
ResNet101 (single-scale)	0.655	0.871	0.714	0.613	0.715	0.701	0.897	0.757	0.650	0.771
ResNet152 (single-scale)	**0.665**	0.880	0.726	0.624	0.723	0.710	0.903	0.766	0.661	0.777
ResNet101 (multi-scale)	0.678	0.886	0.744	0.630	0.748	0.745	0.922	0.804	0.686	0.825
ResNet152 (multi-scale)	**0.687**	0.890	0.754	0.641	0.755	0.754	0.927	0.812	0.697	0.830

image by a uniform random scale factor between 0.5 and 1.5, randomly translate it along the horizontal and vertical directions, and left-right flip it with probability 0.5. We sample and resize the image crop contained under the resulting perturbed box to an 801×801 image that we feed into the network. We use a batch size of 8 images distributed across 8 Nvidia Tesla P100 GPUs in a single machine and perform synchronous training for 1M steps with stochastic gradient descent with constant learning rate equal to 1e-3, momentum value set to 0.9, and Polyak-Ruppert model parameter averaging. We employ batch normalization [72] but fix the statistics of the ResNet activations to their Imagenet values. Our ResNet CNN network backbones have nominal output stride (*i.e.*, ratio of the input image to output activations size) equal to 32 but we reduce it to 16 during training and 8 during evaluation using atrous convolution [67]. During training we also make model predictions using as features activations from a layer in the middle of the network, which we have empirically observed to accelerate training. To balance the different loss terms we use weights equal to $(4, 2, 1, 1/4, 1/8)$ for the heatmap, segmentation, short-range, mid-range, and long-range offset losses in our model. For evaluation we report both single-scale results (image resized to have larger side 1401 pixels) and multi-scale results (pyramid with images having larger side 601, 1201, 1801, 2401 pixels). We have implemented our system in Tensorflow [73]. All reported numbers have been obtained with a single model without ensembling.

4.2 COCO Person Keypoints Evaluation

Table 1 shows our system's person keypoints performance on COCO *test-dev*. Our single-scale inference result is already better than the results of the CMU-Pose [32] and Associative Embedding [2] bottom-up methods, even when they perform multi-scale inference and refine their results with a single-person pose estimation system applied on top of their bottom-up detection proposals. Our results also outperform top-down methods like Mask-RCNN [34] and G-RMI [33]. Our best result with 0.687 AP is attained with a ResNet-152 based model and multi-scale

inference. Our result is still behind the winners of the 2017 keypoints challenge (Megvii) [37] with 0.730 AP, but they used a carefully tuned two-stage, top-down model that also builds on a significantly more powerful CNN backbone.

Table 2. Performance on COCO segmentation (Person category) **test-dev** split. Our person-only results have been obtained with 20 proposals per image. The person category FCIS eval results have been communicated by the authors of [3].

	AP	AP^{50}	AP^{75}	AP^S	AP^M	AP^L	AR^1	AR^{10}	AR^{100}	AR^S	AR^M	AR^L
FCIS (baseline) [3]	0.334	0.641	0.318	0.090	0.411	0.618	0.153	0.372	0.393	0.139	0.492	0.688
FCIS (multi-scale) [3]	0.386	0.693	0.410	0.164	0.481	0.621	0.161	0.421	0.451	0.221	0.562	0.690
PersonLab (ours):												
ResNet101 (1-scale, 20 prop)	0.377	0.659	0.394	0.166	0.480	0.595	0.162	0.415	0.437	0.207	0.536	0.690
ResNet152 (1-scale, 20 prop)	**0.385**	0.668	0.404	0.172	0.488	0.602	0.164	0.422	0.444	0.215	0.544	0.698
ResNet101 (mscale, 20 prop)	0.411	0.686	0.445	0.215	0.496	0.626	0.169	0.453	0.489	0.278	0.571	0.735
ResNet152 (mscale, 20 prop)	**0.417**	0.691	0.453	0.223	0.502	0.630	0.171	0.461	0.497	0.287	0.578	0.742

Table 3. Performance on COCO Segmentation (Person category) **val** split. The Mask-RCNN [34] person results have been produced by the ResNet-101-FPN version of their publicly shared model (which achieves 0.359 AP across all COCO classes).

	AP	AP^{50}	AP^{75}	AP^S	AP^M	AP^L	AR^1	AR^{10}	AR^{100}	AR^S	AR^M	AR^L
Mask-RCNN [34]	0.455	0.798	0.472	0.239	0.511	0.611	0.169	0.477	0.530	0.350	0.596	0.721
PersonLab (ours):												
ResNet101 (1-scale, 20 prop)	0.382	0.661	0.397	0.164	0.476	0.592	0.162	0.416	0.439	0.204	0.532	0.681
ResNet152 (1-scale, 20 prop)	0.387	0.667	0.406	0.169	0.483	0.595	0.163	0.423	0.446	0.213	0.539	0.686
ResNet101 (mscale, 20 prop)	0.414	0.684	0.447	0.213	0.492	0.621	0.170	0.454	0.492	0.278	0.566	0.728
ResNet152 (mscale, 20 prop)	0.418	0.688	0.455	0.219	0.497	0.621	0.170	0.460	0.497	0.284	0.573	0.730
ResNet152 (mscale, 100 prop)	0.429	0.711	0.467	0.235	0.511	0.623	0.170	0.460	0.539	0.346	0.612	0.741

4.3 COCO Person Instance Segmentation Evaluation

Tables 2 and 3 show our person instance segmentation results on COCO *test-dev* and *val*, respectively. We use the small-instance missing keypoint imputation technique of Sect. 3.3 for the reported instance segmentation experiments, which significantly increases our performance for small objects. Our results without missing keypoint imputation are shown in the supplementary material.

Our method only produces segmentation results for the person class, since our system is keypoint-based and thus cannot be applied to the other COCO classes. The standard COCO instance segmentation evaluation allows for a maximum of 100 proposals per image for all 80 COCO classes. For a fair comparison when comparing with previous works, we report *test-dev* results of our method with a maximum of 20 person proposals per image, which is the convention also adopted in the standard COCO person keypoints evaluation protocol. For reference, we also report the *val* results of our best model when allowed to produce 100 proposals.

We compare our system with the person category results of top-down instance segmentation methods. As shown in Table 2, our method on the test split outperforms FCIS [3] in both single-scale and multi-scale inference settings. As shown in Table 3, our performance on the val split is similar to that of Mask-RCNN [34] on medium and large person instances, but worse on small person instances. However, we emphasize that our method is the first box-free, bottom-up instance segmentation method to report experiments on the COCO instance segmentation task.

4.4 Qualitative Results

In Fig. 5 we show representative person pose and instance segmentation results on COCO *val* images produced by our model with single-scale inference.

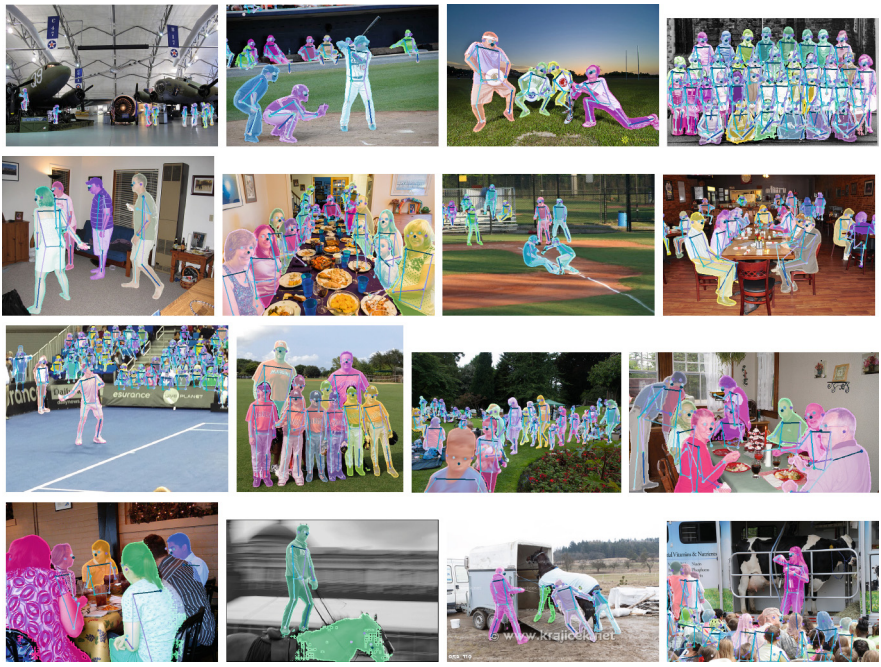


Fig. 5. Visualization on COCO *val* images. The last row shows some failure cases: missed key point detection, false positive key point detection, and missed segmentation.

5 Conclusions

We have developed a bottom-up model which jointly addresses the problems of person detection, pose estimation, and instance segmentation using a unified part-based modeling approach. We have demonstrated the effectiveness of the proposed method on the challenging COCO person keypoint and instance segmentation tasks. A key limitation of the proposed method is its reliance on keypoint-level annotations for training on the instance segmentation task. In the future, we plan to explore ways to overcome this limitation, via weakly supervised part discovery.

References

1. Lin, T.Y., et al.: Coco 2016 keypoint challenge (2016)
2. Newell, A., Deng, J.: Associative embedding: end-to-end learning for joint detection and grouping. In: NIPS (2017)
3. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: CVPR (2017)
4. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings IEEE (1998)
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
6. Fischler, M.A., Elschlager, R.: The representation and matching of pictorial structures. In: IEEE TOC (1973)
7. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
8. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: people detection and articulated pose estimation. In: CVPR (2009)
9. Eichner, M., Ferrari, V.: Better appearance models for pictorial structures. In: BMVC (2009)
10. Sapp, B., Jordan, C., Taskar, B.: Adaptive pose priors for pictorial structures. In: CVPR (2010)
11. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures of parts. In: CVPR (2011)
12. Dantone, M., Gall, J., Leistner, C., Gool., L.V.: Human pose estimation using body parts dependent joint regressors. In: CVPR (2013)
13. Johnson, S., Everingham, M.: Learning effective human pose estimation from inaccurate annotation. In: CVPR (2011)
14. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Poselet conditioned pictorial structures. In: CVPR (2013)
15. Sapp, B., Taskar, B.: Modec: Multimodal decomposable models for human pose estimation. In: CVPR (2013)
16. Gkioxari, G., Arbelaez, P., Bourdev, L., Malik, J.: Articulated pose estimation using discriminative armlet classifiers. In: CVPR (2013)
17. Toshev, A., Szegedy, C.: DeepPose: human pose estimation via deep neural networks. In: CVPR (2014)
18. Jain, A., Tompson, J., Andriluka, M., Taylor, G., Bregler, C.: Learning human pose estimation features with convolutional networks. In: ICLR (2014)

19. Tompson, J., Jain, A., LeCun, Y., Bregler, C.: Join training of a convolutional network and a graphical model for human pose estimation. In: NIPS (2014)
20. Chen, X., Yuille, A.: Articulated pose estimation by a graphical model with image dependent pairwise relations. In: NIPS (2014)
21. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 648–656 (2015)
22. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29
23. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D human pose estimation: new benchmark and state of the art analysis. In: CVPR (2014)
24. Bulat, A., Tzimiropoulos, G.: Human pose estimation via convolutional part heatmap regression. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 717–732. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_44
25. Belagiannis, V., Zisserman, A.: Recurrent human pose estimation. arxiv (2016)
26. Gkioxari, G., Toshev, A., Jaitly, N.: Chained predictions using convolutional neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 728–743. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_44
27. Pishchulin, L., et al.: DeepCut: joint subset partition and labeling for multi person pose estimation. In: CVPR (2016)
28. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: DeeperCut: a deeper, stronger, and faster multi-person pose estimation model. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 34–50. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_3
29. Insafutdinov, E., Andriluka, M., Pishchulin, L., Tang, S., Andres, B., Schiele, B.: Articulated multi-person tracking in the wild. arXiv:1612.01465 (2016)
30. Iqbal, U., Gall, J.: Multi-person pose estimation with local joint-to-person associations. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9914, pp. 627–642. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_44
31. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. arXiv (2016)
32. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: CVPR (2017)
33. Papandreou, G., et al.: Towards accurate multi-person pose estimation in the wild. In: CVPR (2017)
34. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. arXiv:1703.06870v2 (2017)
35. Huang, S., Gong, M., Tao, D.: A coarse-fine network for keypoint localization. In: ICCV (2017)
36. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: RMPE: regional multi-person pose estimation. In: ICCV (2017)
37. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. arXiv:1711.07319 (2017)
38. Girshick, R.: Fast R-CNN. In: ICCV, pp. 1440–1448 (2015)
39. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)

40. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: Object detection via region-based fully convolutional networks. In: NIPS (2016)
41. Carreira, J., Sminchisescu, C.: CPMC: automatic object segmentation using constrained parametric min-cuts. PAMI **34**(7), 1312–1328 (2012)
42. Arbeláez, P., Pont-Tuset, J., Barron, J.T., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR (2014)
43. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 297–312. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10584-0_20
44. Pinheiro, P.O., Collobert, R., Dollár, P.: Learning to segment object candidates. In: NIPS (2015)
45. Dai, J., He, K., Sun, J.: Convolutional feature masking for joint object and stuff segmentation. In: CVPR (2015)
46. Pinheiro, P.O., Lin, T.-Y., Collobert, R., Dollár, P.: Learning to refine object segments. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 75–91. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_5
47. Dai, J., He, K., Li, Y., Ren, S., Sun, J.: Instance-sensitive fully convolutional networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 534–549. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_32
48. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: CVPR (2016)
49. Peng, C., et al.: MegDet: a large mini-batch object detector (2018)
50. Chen, L.C., Hermans, A., Papandreou, G., Schroff, F., Wang, P., Adam, H.: MaskLab: instance segmentation by refining object detection with semantic and direction features. In: CVPR (2018)
51. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: CVPR (2018)
52. Liang, X., Wei, Y., Shen, X., Yang, J., Lin, L., Yan, S.: Proposal-free network for instance-level object segmentation. arXiv preprint arXiv:1509.02636 (2015)
53. Uhrig, J., Cordts, M., Franke, U., Brox, T.: Pixel-level encoding and depth layering for instance-level semantic labeling. arXiv:1604.05096 (2016)
54. Zhang, Z., Schwing, A.G., Fidler, S., Urtasun, R.: Monocular object instance segmentation and depth ordering with CNNs. In: ICCV (2015)
55. Zhang, Z., Fidler, S., Urtasun, R.: Instance-level segmentation for autonomous driving with deep densely connected MRFs. In: CVPR (2016)
56. Wu, Z., Shen, C., van den Hengel, A.: Bridging category-level and instance-level semantic image segmentation. arXiv:1605.06885 (2016)
57. Liu, S., Qi, X., Shi, J., Zhang, H., Jia, J.: Multi-scale patch aggregation (MPA) for simultaneous detection and segmentation. In: CVPR (2016)
58. Levinkov, E., et al.: Joint graph decomposition & node labeling: problem, algorithms, applications. In: CVPR (2017)
59. Kirillov, A., Levinkov, E., Andres, B., Savchynskyy, B., Rother, C.: InstanceCut: from edges to instances with multicut. In: CVPR (2017)
60. Jin, L., Chen, Z., Tu, Z.: Object detection free instance segmentation with labeling transformations. arXiv:1611.08991 (2016)
61. Fathi, A., et al.: Semantic instance segmentation via deep metric learning. arXiv:1703.10277 (2017)

62. De Brabandere, B., Neven, D., Van Gool, L.: Semantic instance segmentation with a discriminative loss function. arXiv:1708.02551 (2017)
63. Bai, M., Urtasun, R.: Deep watershed transform for instance segmentation. In: CVPR (2017)
64. Liu, S., Jia, J., Fidler, S., Urtasun, R.: SGN: sequential grouping networks for instance segmentation. In: ICCV (2017)
65. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-NMS: improving object detection with one line of code. In: ICCV (2017)
66. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
67. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. TPAMI (2017)
68. Radosavovic, I., Dollár, P., Girshick, R., Gkioxari, G., He, K.: Data distillation: towards omni-supervised learning. arXiv:1712.04440 (2017)
69. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
70. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
71. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
72. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167 (2015)
73. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). tensorflow.org