Towards Accurate Multi-person Pose Estimation in the Wild

George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev,
Jonathan Tompson, Chris Bregler, Kevin Murphy
Google, Inc.
[gpapan, tylerzhu, kanazawa, toshev, tompson, bregler, kpmurphy]@google.com

Abstract

We propose a method for multi-person detection and 2-D pose estimation that achieves state-of-art results on the challenging COCO keypoints task. It is a simple, yet powerful, top-down approach consisting of two stages.

In the first stage, we predict the location and scale of boxes which are likely to contain people; for this we use the Faster RCNN detector. In the second stage, we estimate the keypoints of the person potentially contained in each proposed bounding box. For each keypoint type we predict dense heatmaps and offsets using a fully convolutional ResNet. To combine these outputs we introduce a novel aggregation procedure to obtain highly localized keypoint predictions. We also use a novel form of keypoint-based Non-Maximum-Suppression (NMS), instead of the cruder box-level NMS, and a novel form of keypoint-based confidence score estimation, instead of box-level scoring.

Trained on COCO data alone, our final system achieves average precision of 0.649 *on the COCO test-dev set and the* 0.643 *test-standard sets, outperforming the winner of the 2016 COCO keypoints challenge and other recent state-of-art. Further, by using additional in-house labeled data we obtain an even higher average precision of* 0.685 *on the test-dev set and* 0.673 *on the test-standard set, more than* 5% *absolute improvement compared to the previous best performing method on the same dataset.*

1. Introduction

Visual interpretation of people plays a central role in the quest for comprehensive image understanding. We want to localize people, understand the activities they are involved in, understand how people move for the purpose of Virtual/Augmented Reality, and learn from them to teach autonomous systems. A major cornerstone in achieving these goals is the problem of human pose estimation, defined as 2-D localization of human joints on the arms, legs, and keypoints on torso and the face.

Recently, there has been significant progress on this problem, mostly by leveraging deep Convolutional Neural Networks (CNNs) trained on large labeled datasets [45, 27, 44, 10, 33, 2, 7, 6, 20, 25, 8]. However, most prior work has focused on the simpler setting of predicting the pose of a single person assuming the location and scale of the person is provided in the form of a ground truth bounding box or torso keypoint position, as in the popular MPII [2] and FLIC [40] datasets.

In this paper, we tackle the more challenging setting of pose detection 'in the wild', in which we are not provided with the ground truth location or scale of the person instances. This is harder because it combines the problem of person detection with the problem of pose estimation. In crowded scenes, where people are close to each other, it can be quite difficult to solve the association problem of determining which body part belongs to which person.

The recently released COCO person keypoints detection dataset and associated challenge [31] provide an excellent vehicle to encourage research, establish metrics, and measure progress on this task. It extends the COCO dataset [32] with additional annotations of 17 keypoints (12 body joints and 5 face landmarks) for every medium and large sized person in each image. A large number of persons in the dataset are only partially visible. The degree of match between ground truth and predicted poses in the COCO keypoints task is measured in terms of object keypoint similarity (OKS), which ranges from 0 (poor match) to 1 (perfect match). The overall quality of the combined person detection and pose estimation system in the benchmark is measured in terms of an OKS-induced average precision (AP) metric. In this paper, we describe a system that achieves state-of-the-art results on this challenging task.

There are two broad approaches for tackling the multi-person pose estimation problem: *bottom-up*, in which keypoint proposals are grouped together into person instances, and *top-down*, in which a pose estimator is applied to the output of a bounding-box person detector. Recent work [35, 25, 8, 24] has advocated the bottom-up approach; in their experiments, their proposed bottom-up methods outperformed the top-down baselines they compared with.

In contrast, in this work we revisit the top-down approach and show that it can be surprisingly effective. The proposed system is a two stage pipeline with state-of-art constituent components carefully adapted to our task. In the first stage, we predict the location and scale of boxes which are likely to contain people. For this we use the Faster-RCNN method [37] on top of a ResNet-101 CNN [22], as implemented by [23]. In the second stage, we predict the locations of each keypoint for each of the proposed person boxes. For this we use a ResNet [22] applied in a fully convolutional fashion to predict activation heatmaps and offsets for each keypoint, similar to the works of Pishchulin et al. [35] and Insafutdinov et al. [25], followed by combining their predictions using a novel form of heatmap-offset aggregation. We avoid duplicate pose detections by means of a novel keypoint-based Non-Maximum-Suppression (NMS) mechanism building directly on the OKS metric (which we call OKS-NMS), instead of the cruder box-level IOU NMS. We also propose a novel keypoint-based confidence score estimator, which we show leads to greatly improved AP compared to using the Faster-RCNN box scores for ranking our final pose proposals. The system described in this paper is an improved version of our G-RMI entry to the COCO 2016 keypoints detection challenge.

Using only publicly available data for training, our final system achieves average precision of 0.649 on the COCO *test-dev* set and 0.643 on the COCO *test-standard* set, outperforming the winner of the 2016 COCO keypoints challenge [8], which gets 0.618 on *test-dev* and 0.611 on *test-standard*, as well as the very recent Mask-RCNN [21] methods which gets 0.631 on *test-dev*. Using additional in-house labeled data we obtain an even higher average precision of 0.685 on the test-dev set and 0.673 on the test-standard set, more than 5% absolute performance improvement over the best previous methods. These results have been attained with single-scale evaluation and using a single CNN for box detection and a single CNN for pose estimation. Multi-scale evaluation and CNN model ensembling might give additional gains.

In the rest of the paper, we discuss related work and then describe our method in more detail. We then perform an experimental study, comparing our system to recent state-of-the-art, and we measure the effects of the different parts of our system on the AP metric.

2. Related Work

For most of its history, the research in human pose estimation has been heavily based on the idea of part-based models, as pioneered by the Pictorial Structures (PS) model of Fischler and Elschlager [16]. One of the first practical and well performing methods based on this idea is Deformable Part Model (DPM) by Felzenswalb et al. [15], which spurred a large body of work on probabilistic graphical models for 2-D human pose inference [3, 12, 39, 47, 11, 28, 34, 40, 18]. The majority of these methods focus on developing tractable inference procedures for highly articulated models, while at the same time capturing rich dependencies among body parts and properties.

Single-Person Pose With the development of Deep Convolutional Neural Networks (CNN) for vision tasks, state-of-art performance on pose estimation is achieved using CNNs [45, 27, 44, 10, 33, 2, 7, 6, 20, 25, 8]. The problem can be formulated as a regression task, as done by Toshev and Szegedy [45], using a cascade of detectors for top-down pose refinement from cropped input patches. Alternatively, Jain et al. [27] trained a CNN on image patches, which was applied convolutionally at inference time to infer heatmaps (or activity-maps) for each keypoint independently. In addition, they used a "DPM-like" graphical-model post processing step to filter heatmap potentials and to impose inter-joint consistency. Following this work, Tompson et al. [44] used a multi-scale fully-convolutional architecture trained on whole images (rather than image crops) to infer the heatmap potentials, and they reformulated the graphical model from [27] - simplifying the tree structure to a star-graph and re-writing the belief propagation messages - so that the entire system could be trained end-to-end.

Chen et al. [10] added image-dependent priors to improve CNN performance. By learning a lower-dimensional image representation, they clustered the input image into a mixture of configurations of each pair of consecutive joints. Depending on which mixture is active for a given input image, a separate pairwise displacement prior was used for graphical model inference, resulting in stronger pairwise priors and improved overall performance.

Bulat et al. [7] use a cascaded network to explicitly infer part relationships to improve inter-joint consistency, which the authors claim effectively encodes part constraints and inter-joint context. Similarly, Belagiannis & Zisserman [6] also propose a cascaded architecture to infer pairwise joint (or part) locations, which is then used to iteratively refine unary joint predictions, where unlike [7], they propose iterative refinement using a recursive neural network.

Inspired by recent work in sequence-to-sequence modeling, Gkioxari et al. [20] propose a novel network structure where body part locations are predicted sequentially rather than independently, as per traditional feed-forward networks. Body part locations are conditioned on the input image and all other predicted parts, yielding a model which promotes sequential reasoning and learns complex inter-joint relationships.

The state-of-the-art approach for single-person pose on the MPII human pose [2] and FLIC [40] datasets is the CNN model of Newell et al. [33]. They propose a novel CNN architecture that uses skip-connections to promote multi-scale feature learning, as well as a repeated pooling-

upsampling ("hourglass") structure that results in improved iterative pose refinement. They claim that their network is able to more efficiently learn various spatial relationship associated with the body, even over large pixel displacements, and with a small number of total network parameters.

Top-Down Multi-Person Pose The problem of multi-person pose estimation presents different challenges, unadressed by the above work. Most of the approaches for multi-person pose aim at associating person part detections with person instances. The *top down* way to establish these associations, which is closest to our approach, is to first perform person detection followed by pose estimation. For example, Pishchulin et al. [36] follow this paradigm by using PS-based pose estimation. A more robust to occlusions person detector, modeled after poselets, is used by Gkioxari et al. [19]. Further, Yang and Ramanan [47] fuse detection and pose in one model by using a PS model. The inference procedure allows for pose estimation of multiple person instances per image analogous to PS-based object detection. A similar multi-person PS with additional explicit occlusion modeling is proposed by Eichner and Ferrari [13]. The very recent Mask-RCNN method [21] extends Faster-RCNN [37] to also support keypoint estimation, obtaining very competitive results. On a related note, 2-D person detection is used as a first step in several 3D pose estimation works [41, 4, 5].

Bottom-Up Multi-Person Pose A different line of work is to detect body parts instead of full persons, and to subsequently associate these parts to human instances, thus performing pose estimation in a *bottom up* fashion. Such approaches employ part detectors and differ in how associations among parts are expressed, and the inference procedure used to obtain full part groupings into person instances. Pishchulin et al. [35] and later Insafutdinov et al. [25, 24] formulate the problem of pose estimation as part grouping and labeling via a Linear Program. A similar formulation is proposed by Iqbal et al. [26]. A probabilistic approach to part grouping and labeling is also proposed by Ladicky et al. [29], leveraging a HOG-based system for part detections.

Cao et al. [8] winning entry to the 2016 COCO person keypoints challenge [32] combines a variation of the unary joint detector architecture from [46] with a part affinity field regression to enforce inter-joint consistency. They employ a greedy algorithm to generate person instance proposals in a bottom-up fashion. Their best results are obtained in an additional top-down refinement process in which they run a standard single-person pose estimator [46] on the person instance box proposals generated by the bottom-up stage.

3. Methods

Our multi-person pose estimation system is a two step cascade, as illustrated in Figure 1.

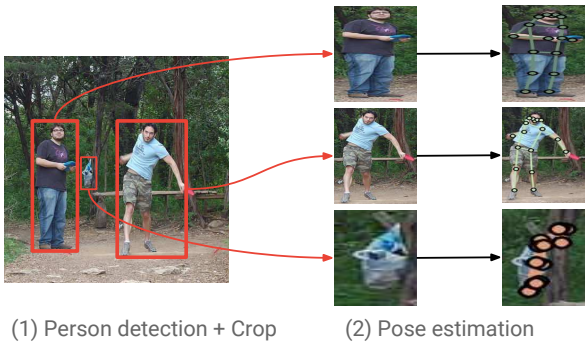

Figure 1: Overview of our two stage cascade model. In the first stage, we employ a Faster-RCNN person detector to produce a bounding box around each candidate person instance. In the second stage, we apply a pose estimator to the image crop extracted around each candidate person instance in order to localize its keypoints and re-score the corresponding proposal.

Our approach is inspired by recent state-of-art object detection systems such as [17, 43], which propose objects in a class agnostic fashion as a first stage and refine their label and location in a second stage. We can think of the first stage of our method as a proposal mechanism, however of only one type of object – person. Our second stage serves as a refinement where we (i) go beyond bounding boxes and predict keypoints and (ii) rescore the detection based on the estimated keypoints. For computational efficiency, we only forward to the second stage person box detection proposals with score higher than 0.3, resulting in only 3.5 proposals per image on average. In the following, we describe in more detail the two stages of our system.

3.1. Person Box Detection

Our person detector is a Faster-RCNN system [37]. In all experiments reported in this paper we use a ResNet-101 network backbone [22], modified by atrous convolution [9, 30] to generate denser feature maps with output stride equal to 8 pixels instead of the default 32 pixels. We have also experimented with an Inception-ResNet CNN backbone [42], which is an architecture integrating Inception layers [43] with residual connections [22], which performs slightly better at the cost of increased computation.

The CNN backbone has been pre-trained for image classification on Imagenet. In all reported experiments, both the region proposal and box classifier components of the Faster-RCNN detector have been trained using only the person category in the COCO dataset and the box annotations for the remaining 79 COCO categories have been ignored. We use the Faster-RCNN implementation of [23] written in Tensorflow [1]. For simplicity and to facilitate reproducibility we do not utilize multi-scale evaluation or model ensembling

in the Faster-RCNN person box detection stage. Using such enhancements can further improve our results at the cost of significantly increased computation time.

3.2. Person Pose Estimation

The pose estimation component of our system predicts the location of all $K = 17$ person keypoints, given each person bounding box proposal delivered by the first stage.

One approach would be to use a single regressor per keypoint, as in [45], but this is problematic when there is more than one person in the image patch (in which case a keypoint can occur in multiple places). A different approach addressing this issue would be to predict activation maps, as in [27], which allow for multiple predictions of the same keypoint. However, the size of the activation maps, and thus the localization precision, is limited by the size of the net's output feature maps, which is a fraction of the input image size, due to the use of max-pooling with decimation.

In order to address the above limitations, we adopt a combined classification and regression approach. For each spatial position, we first classify whether it is in the vicinity of each of the K keypoints or not (which we call a "heatmap"), then predict a 2-D local offset vector to get a more precise estimate of the corresponding keypoint location. Note that this approach is inspired by work on object detection, where a similar setup is used to predict bounding boxes, e.g. [14, 37]. Figure 2 illustrates these three output channels per keypoint.

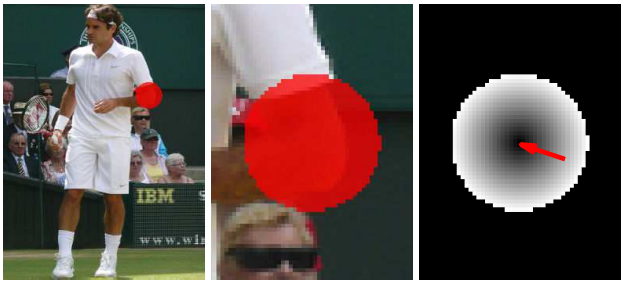

Figure 2: Network target outputs. *Left & Middle*: Heatmap target for the left-elbow keypoint (red indicates heatmap of 1). *Right*: Offset field L2 magnitude (shown in grayscale) and 2-D offset vector shown in red).

Image Cropping We first make all boxes have the same fixed aspect ratio by extending either the height or the width of the boxes returned by the person detector without distorting the image aspect ratio. After that, we further enlarge the boxes to include additional image context: we use a rescaling factor equal to 1.25 during evaluation and a random rescaling factor between 1.0 and 1.5 during training (for data augmentation). We then crop from the resulting box the image and resize to a fixed crop of height 353 and width 257 pixels. We set the aspect ratio value to $353/257 = 1.37$.

Heatmap and Offset Prediction with CNN We apply a ResNet with 101 layers [22] on the cropped image in a fully convolutional fashion to produce heatmaps (one channel per keypoint) and offsets (two channels per keypoint for the x- and y- directions) for a total of $3 \cdot K$ output channels, where $K = 17$ is the number of keypoints. We initialize our model from the publicly available Imagenet pretrained ResNet-101 model of [22], replacing its last layer with 1x1 convolution with $3 \cdot K$ outputs. We follow the approach of [9]: we employ atrous convolution to generate the $3 \cdot K$ predictions with an output stride of 8 pixels and bilinearly upsample them to the 353x257 crop size.

In more detail, given the image crop, let $f_k(x_i) = 1$ if the k-th keypoint is located at position x_i and 0 otherwise. Here $k \in \{1, \ldots, K\}$ is indexing the keypoint type and $i \in \{1, \ldots, N\}$ is indexing the pixel locations on the 353x257 image crop grid. Training a CNN to produce directly the highly localized activations f_k (ideally delta functions) on a fine resolution spatial grid is hard.

Instead, we decompose the problem into two stages. First, for each position x_i and each keypoint k, we compute the probability $h_k(x_i) = 1$ if $||x_i - l_k|| \leq R$ that the point x_i is within a disk of radius R from the location l_k of the k-th keypoint. We generate K such heatmaps, solving a binary classification problem for each position and keypoint independently.

In addition to the heatmaps, we also predict at each position i and each keypoint k the 2-D offset vector $F_k(x_i) = l_k - x_i$ from the pixel to the corresponding keypoint. We generate K such vector fields, solving a 2-D regression problem for each position and keypoint independently.

After generating the heatmaps and offsets, we aggregate them to produce highly localized activation maps $f_k(x_i)$ as follows:

$$f_k(x_i) = \sum_j \frac{1}{\pi R^2} G(x_j + F_k(x_j) - x_i) h_k(x_j)\,, \qquad (1)$$

where $G(\cdot)$ is the bilinear interpolation kernel. This is a form of Hough voting: each point j in the image crop grid casts a vote with its estimate for the position of every keypoint, with the vote being weighted by the probability that it is in the disk of influence of the corresponding keypoint. The normalizing factor equals the area of the disk and ensures that if the heatmaps and offsets were perfect, then $f_k(x_i)$ would be a unit-mass delta function centered at the position of the k-th keypoint.

The process is illustrated in Figure 3. We see that predicting separate heatmap and offset channels and fusing them by the proposed voting process results into highly localized activation maps which precisely pinpoint the position of the keypoints.

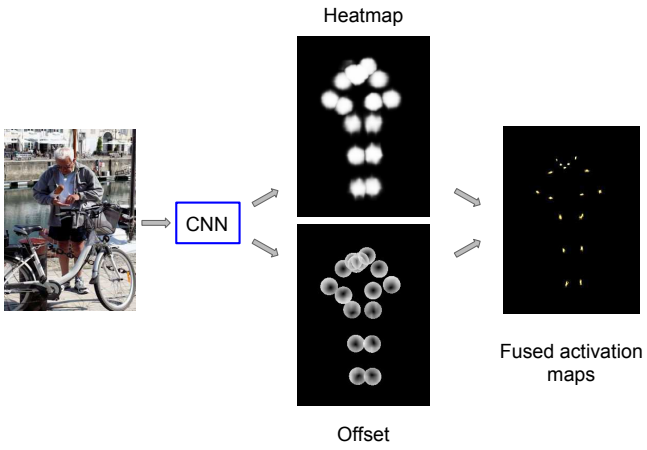

Figure 3: Our fully convolutional network predicts two targets: (1) Disk-shaped heatmaps around each keypoint and (2) magnitude of the offset fields towards the exact keypoint position within the disk. Aggregating them in a weighted voting process results in highly localized activation maps. The figure shows the heatmaps and the pointwise magnitude of the offset field on a validation image. Note that in this illustration we super-impose the channels from the different keypoints.

Model Training We use a single ResNet model with two convolutional output heads. The output of the first head passes through a sigmoid function to yield the heatmap probabilities $h_k(x_i)$ for each position x_i and each keypoint k. The training target $\bar{h}_k(x_i)$ is a map of zeros and ones, with $\bar{h}_k(x_i) = 1$ if $||x_i - l_k|| \leq R$ and 0 otherwise. The corresponding loss function $L_h(\theta)$ is the sum of logistic losses for each position and keypoint separately. To accelerate training, we follow [25] and add an extra heatmap prediction layer at intermediate layer 50 of ResNet, which contributes a corresponding auxiliary loss term.

For training the offset regression head, we penalize the difference between the predicted and ground truth offsets. The corresponding loss is

$$L_o(\theta) = \sum_{k=1:K} \sum_{i:||l_k - x_i|| \leq R} H(||F_k(x_i) - (l_k - x_i)||)\,, \quad (2)$$

where $H(u)$ is the Huber robust loss, l_k is the position of the k-th keypoint, and we only compute the loss for positions x_i within a disk of radius R from each keypoint [37].

The final loss function has the form

$$L(\theta) = \lambda_h L_h(\theta) + \lambda_o L_o(\theta)\,, \quad (3)$$

where $\lambda_h = 4$ and $\lambda_o = 1$ is a scalar factor to balance the loss function terms. We sum this loss over all the images in a minibatch, and then apply stochastic gradient descent.

An important consideration in model training is how to treat cases where multiple people exist in the image crop in the computation of heatmap loss. When computing the heatmap loss at the intermediate layer, we exclude contributions from within the disks around the keypoints of background people. When computing the heatmap loss at the final layer, we treat as positives only the disks around the keypoints of the foreground person and as negatives everything else, forcing the model to predict correctly the keypoints of the person in the center of the box.

Pose Rescoring At test time, we apply the model to each image crop. Rather than just relying on the confidence from the person detector, we compute a refined confidence estimate, which takes into account the confidence of each keypoint. In particular, we maximize over locations and average over keypoints, yielding our final instance-level pose detection score:

$$\text{score}(\mathcal{I}) = \frac{1}{K} \sum_{k=1}^{K} \max_{x_i} f_k(x_i) \quad (4)$$

We have found that ranking our system's pose estimation proposals using 4 significantly improves AP compared to using the score delivered by the Faster-RCNN box detector.

OKS-Based Non Maximum Suppression Following standard practice, we use non maximal suppression (NMS) to eliminate multiple detections in the person-detector stage. The standard approach measures overlap using intersection over union (IoU) of the boxes. We propose a more refined variant which takes the keypoints into account. In particular, we measure overlap using the object keypoint similarity (OKS) for two candidate pose detections. Typically, we use a relatively high IOU-NMS threshold (0.6 in our experiments) at the output of the person box detector to filter highly overlapping boxes. The subtler OKS-NMS at the output of the pose estimator is better suited to determine if two candidate detections correspond to false positives (double detection of the same person) or are true positives (two people in close proximity to each other).

4. Experimental Evaluation

4.1. Experimental Setup

We have implemented out system in Tensorflow [1]. We use distributed training across several machines equipped with Tesla K40 GPUs.

For person detector training we use 9 GPUs. We optimize with asynchronous SGD with momentum set to 0.9. The learning rate starts at 0.0003 and is decreased by a factor of 10 at 800k steps. We train for 1M steps.

Figure 4: Detection and pose estimation results using our system on a random selection from the COCO test-dev set. For each detected person, we display the detected bounding box together with the estimated keypoints. All detections for one person are colored the same way. It is worth noting that our system works in heavily cluttered scenes (third row, rightmost and last row, right); it deals well with occlusions (last row, left) and hallucinates occluded joints. Last but not least, some of the false positive detections are in reality correct as they represent pictures of people (first row, middle) or toys (fourth row, middle). Figure best viewed zoomed in on a monitor.

Table 1: Performance on COCO keypoint **test-dev** split.

	AP	AP .5	AP .75	AP (M)	AP (L)	AR	AR .5	AR .75	AR (M)	AR (L)
CMU-Pose [8]	0.618	0.849	0.675	0.571	0.682	0.665	0.872	0.718	0.606	0.746
Mask-RCNN [21]	0.631	0.873	0.687	0.578	0.714					
G-RMI (ours): *COCO-only*	0.649	0.855	0.713	0.623	0.700	0.697	0.887	0.755	0.644	0.771
G-RMI (ours): *COCO+int*	**0.685**	0.871	0.755	0.658	0.733	0.733	0.901	0.795	0.681	0.804

Table 2: Performance on COCO keypoint **test-standard** split.

	AP	AP .5	AP .75	AP (M)	AP (L)	AR	AR .5	AR .75	AR (M)	AR (L)
CMU-Pose[8]	0.611	0.844	0.667	0.558	0.684	0.665	0.872	0.718	0.602	0.749
G-RMI (ours): *COCO-only*	0.643	0.846	0.704	0.614	0.696	0.698	0.885	0.755	0.644	0.771
G-RMI (ours): *COCO+int*	**0.673**	0.854	0.735	0.642	0.726	0.730	0.898	0.789	0.675	0.805

For pose estimator training we use two machines equipped with 8 GPUs each and batch size equal to 24 (3 crops per GPU times 8 GPUs). We use a fixed learning rate of 0.005 and Polyak-Ruppert parameter averaging, which amounts to using during evaluation a running average of the parameters during training. We train for 800k steps.

All our networks are pre-trained on the Imagenet classification dataset [38]. To train our system we use two dataset variants; one that uses only COCO data (*COCO-only*), and one that appends to this dataset samples from an internal dataset (*COCO+int*). For the *COCO-only* dataset we use the COCO keypoint annotations [32]: From the 66,808 images (273,469 person instances) in the COCO *train+val* splits, we use 62,174 images (105,698 person instances) in *COCO-only* model training and use the remaining 4,301 annotated images as *mini-val* evaluation set. Our *COCO+int* training set is the union of *COCO-only* with an additional 73,024 images randomly selected from Flickr. This in-house dataset contains an additional 227,029 person instances annotated with keypoints following a procedure similar to that described by Lin et al. [31]. The additional training images have been verified to have no overlap with the COCO training, validation or test sets.

We have trained our Faster-RCNN person box detection module exclusively on the *COCO-only* dataset. We have experimented training our ResNet-based pose estimation module either on the *COCO-only* or on the augmented *COCO+int* datasets and present results for both. For *COCO+int* pose training we use mini-batches that contain COCO and in-house annotation instances in 1:1 ratio.

4.2. COCO Keypoints Detection State-of-the-Art

Table 1 shows the COCO keypoint test-dev split performance of our system trained on *COCO-only* or trained on *COCO+int* datasets. A random selection of test-dev inference samples are shown in Figure 4.

Table 2 shows the COCO keypoint test-standard split results of our model with the pose estimator trained on either *COCO-only* or *COCO+int* training set.

Even with *COCO-only* training, we achieve state-of-the-art results on the COCO test-dev and test-standard splits, outperforming the COCO 2016 challenge winning CMU-Pose team [8] and the very recent Mask-RCNN method [21]. Our best results are achieved with the pose estimator trained on *COCO+int* data, yielding an AP score of 0.673 on *test-standard*, an absolute 6.2% improvement over the 0.611 *test-standard* score of CMU-Pose [8].

4.3. Ablation Study: Box Detection Module

An important question for our two-stage system is its sensitivity to the quality of its box detection and pose estimator constituent modules. We examine two variants of the ResNet-101 based Faster-RCNN person box detector, (a) a fast 600x900 variant that uses input images with small side 600 pixels and large side 900 pixels and (b) an accurate 800x1200 variant that uses input images with small side 800 pixels and large side 1200 pixels. Their box detection AP on our COCO person *mini-val* is 0.466 and 0.500, respectively. Their box detection AP on COCO *test-dev* is 0.456 and 0.487, respectively. For reference, the person box detection AP on COCO *test-dev* of the top-performing multi-crop/ensemble entry of [23] is 0.539. We have also tried feeding our pose estimator module with the ground truth person boxes to examine its oracle performance limit in isolation from the box detection module. We report our COCO *mini-val* results in Table 3 for pose estimators trained on either *COCO-only* or *COCO+int*. We use the accurate Faster-RCNN (800x1200) box detector for all results in the rest of the paper.

4.4. Ablation Study: Pose Estimation Module

We have experimented with alternative CNN setups for our pose estimation module. We have explored CNN network backbones based on either the faster ResNet-50 or the more accurate ResNet-101, while keeping ResNet-101 as CNN backbone for the Faster-RCNN box detection module. We have also experimented with two sizes for the image crops that are fed as input to the pose estimator:

Table 3: Ablation on the box detection module: Performance on COCO keypoint *mini-val* when using alternative box detection modules trained on *COCO-only* or ground truth boxes. We use the default ResNet-101 pose estimation module trained on either *COCO-only* or *COCO+int*. We mark with an asterisk our default box detection module used in all other experiments.

Box Module	Poser Train	AP	AP .5	AP .75	AP (M)	AP (L)	AR	AR .5	AR .75	AR (M)	AR (L)
Faster-RCNN (600x900)	COCO-only	0.657	0.831	0.721	0.617	0.725	0.699	0.856	0.754	0.634	0.788
Faster-RCNN (800x1200)*	COCO-only	0.667	0.851	0.730	0.633	0.726	0.708	0.874	0.763	0.652	0.786
Ground-truth boxes	COCO-only	0.704	0.904	0.771	0.684	0.746	0.736	0.911	0.794	0.693	0.796
Faster-RCNN (600x900)	COCO+int	0.693	0.854	0.757	0.650	0.762	0.730	0.871	0.786	0.665	0.819
Faster-RCNN (800x1200)*	COCO+int	0.700	0.860	0.764	0.665	0.760	0.742	0.888	0.800	0.686	0.820
Ground-truth boxes	COCO+int	0.745	0.925	0.815	0.725	0.783	0.774	0.930	0.835	0.735	0.831

Table 4: Ablation on the pose estimation module: Performance on COCO keypoint *test-dev* when using alternative pose estimation modules trained on *COCO+int*. We use the default ResNet-101 box detection module trained on *COCO-only*. We mark with an asterisk our default pose estimation module used in all other experiments.

Pose Module	Poser Train	AP	AP .5	AP .75	AP (M)	AP (L)	AR	AR .5	AR .75	AR (M)	AR (L)
ResNet-50 (257x185)	COCO+int	0.649	0.853	0.722	0.627	0.693	0.699	0.890	0.763	0.650	0.766
ResNet-50 (353x257)	COCO+int	0.666	0.862	0.734	0.638	0.717	0.714	0.894	0.774	0.661	0.787
ResNet-101 (257x185)	COCO+int	0.661	0.862	0.734	0.641	0.708	0.712	0.895	0.777	0.662	0.782
ResNet-101 (353x257)*	COCO+int	0.685	0.871	0.755	0.658	0.733	0.733	0.901	0.795	0.681	0.804

Table 5: Performance (AP) on COCO keypoint *mini-val* with varying values for the OKS-NMS threshold. The pose estimator has been trained with either *COCO-only* or *COCO+int* data.

Threshold	0.1	0.3	0.5*	0.7	0.9
AP (*COCO-only*)	0.638	0.664	0.667	0.665	0.658
AP (*COCO+int*)	0.672	0.699	0.700	0.701	0.694

Smaller (257x185) for faster inference or larger (353x257) for higher accuracy. We report in Table 4 COCO *test-dev* results for the four CNN backbone/ crop size combinations, using *COCO+int* for pose estimator training. We see that ResNet-101 performs about 2% better but in computation-constrained environments ResNet-50 remains a competitive alternative. We use the accurate ResNet-101 (353x257) pose estimator with disk radius $R = 25$ pixels in the rest of the paper.

4.5. OKS-Based Non Maximum Suppression

We examine the effect of the proposed OKS-based non-maximum suppression method at the output of the pose estimator for different values of the OKS-NMS threshold. In all experiments the value of the IOU-NMS threshold at the output of the person box detector remains fixed at 0.6. We report in Table 5 COCO *mini-val* results using either *COCO-only* or *COCO+int* for pose estimator training. We fix the OKS-NMS threshold to 0.5 in the rest of the paper.

5. Conclusion

In this work we address the problem of person detection and pose estimation in cluttered images 'in the wild'. We present a simple two stage system, consisting of a person detection stage followed by a keypoint estimation stage for each person. Despite its simplicity it achieves state-of-art results as measured on the challenging COCO benchmark.

Acknowledgments

We are grateful to the authors of [23] for making their excellent Faster-RCNN implementation available to us. We would like to thank Hartwig Adam for encouraging and supporting this project and Akshay Gogia and Gursheesh Kour for managing our internal annotation effort.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

[3] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009.

[4] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3d pictorial structures for multiple human pose estimation. In *CVPR*, pages 1669–1676, 2014.

[5] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3d pictorial structures revisited: Multiple human pose estimation. In *CVPR*, 2015.

[6] V. Belagiannis and A. Zisserman. Recurrent human pose estimation. In *arxiv*, 2016.

[7] A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, 2016.

[8] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv:1611.08050v1*, 2016.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016.

[10] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, 2014.

[11] M. Dantone, J. Gall, C. Leistner, and L. V. Gool. Human pose estimation using body parts dependent joint regressors. In *CVPR*, 2013.

[12] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *BMVC*, 2009.

[13] M. Eichner and V. Ferrari. We are family: Joint pose estimation of multiple persons. In *ECCV*, pages 228–242. Springer, 2010.

[14] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *CVPR*, pages 2147–2154, 2014.

[15] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[16] M. A. Fischler and R. Elschlager. The representation and matching of pictorial structures. In *IEEE TOC*, 1973.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[18] G. Gkioxari, P. Arbelaez, L. Bourdev, and J. Malik. Articulated pose estimation using discriminative armlet classifiers. In *CVPR*, 2013.

[19] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. Using k-poselets for detecting people and localizing their keypoints. In *CVPR*, pages 3582–3589, 2014.

[20] G. Gkioxari, A. Toshev, and N. Jaitly. Chained predictions using convolutional neural networks. In *ECCV*, 2016.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *arXiv:1703.06870v2*, 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[23] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv:1611.10012*, 2016.

[24] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, B. Andres, and B. Schiele. Articulated multi-person tracking in the wild. *arXiv:1612.01465*, 2016.

[25] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016.

[26] U. Iqbal and J. Gall. Multi-person pose estimation with local joint-to-person associations. In *ECCV*, pages 627–642. Springer, 2016.

[27] A. Jain, J. Tompson, M. Andriluka, G. Taylor, and C. Bregler. Learning human pose estimation features with convolutional networks. In *ICLR*, 2014.

[28] S. Johnson and M. Everingham. Learning Effective Human Pose Estimation from Inaccurate Annotation. In *CVPR*, 2011.

[29] L. Ladicky, P. H. Torr, and A. Zisserman. Human pose estimation using a joint pixel-wise and part-wise formulation. In *CVPR*, pages 3578–3585, 2013.

[30] Y. Li, K. He, J. Sun, et al. R-FCN: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, pages 379–387, 2016.

[31] T.-Y. Lin, Y. Cui, G. Patterson, M. R. Ronchi, L. Bourdev, R. Girshick, and P. Dollr. Coco 2016 keypoint challenge. 2016.

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[33] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[34] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *CVPR*, 2013.

[35] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, 2016.

[36] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele. Articulated people detection and pose estimation: Reshaping the future. In *CVPR*, pages 3178–3185, 2012.

[37] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time object detection with region proposal networks. In *NIPS*, 2015.

[38] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[39] B. Sapp, C. Jordan, and B.Taskar. Adaptive pose priors for pictorial structures. In *CVPR*, 2010.

[40] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *CVPR*, 2013.

[41] M. Sun and S. Savarese. Articulated part-based model for joint object detection and pose estimation. In *ICCV*, pages 723–730, 2011.

[42] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv:1602.07261*, 2016.

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[44] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Join training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014.

[45] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

[46] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *arXiv*, 2016.

[47] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures of parts. In *CVPR*, 2011.