Joint Human Pose Estimation and Instance Segmentation with Pose*Plus*Seg

Niaz Ahmad, Jawad Khan, Jeremy Yuhyun Kim, Youngmoon Lee

Hanyang University, Ansan, South Korea
{niazahamd89, jkhanbk1, yuhyunkim, youngmoonlee}@hanyang.ac.kr

Abstract

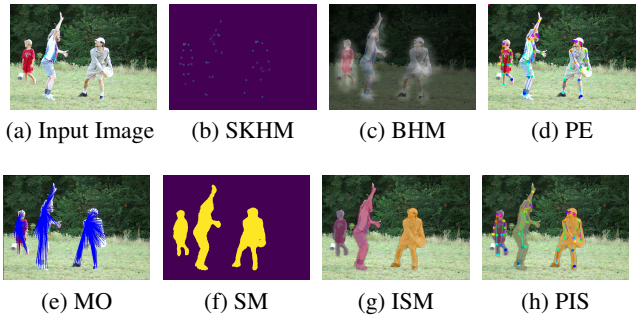
Despite the advances in multi-person pose estimation, state-of-the-art techniques only deliver the human pose structure. Yet, they do not leverage the keypoints of human pose to deliver whole-body shape information for human instance segmentation. This paper presents Pose*Plus*Seg, a joint model designed for both human pose estimation and instance segmentation. For pose estimation, Pose*Plus*Seg first takes a bottom-up approach to detect the soft and hard keypoints of individuals by producing a strong keypoint heat map, then improves the keypoint detection confidence score by producing a body heat map. For instance segmentation, Pose*Plus*Seg generates a mask offset where keypoint is defined as a centroid for the pixels in the embedding space, enabling instance-level segmentation for the human class. Finally, we propose a new pose and instance segmentation algorithm that enables Pose*Plus*Seg to determine the joint structure of the human pose and instance segmentation. Experiments using the COCO challenging dataset demonstrate that Pose*Plus*Seg copes better with challenging scenarios, like occlusions, entangled limbs, and overlapped people. Pose*Plus*Seg outperforms state-of-the-art detection-based approaches achieving a 0.728 mAP for human pose estimation and a 0.445 mAP for instance segmentation. Code has been made available at https://github.com/RaiseLab/PosePlusSeg.

Introduction

Human pose estimation and body segmentation are major cornerstones in many computer vision applications such as activity recognition, video surveillance, human-computer interaction, etc. These applications require the two-dimensional (2D) positioning of human joints and their body shape structure to identify individuals and their activities. Existing pose estimation models (Chen et al. 2018; Fang et al. 2017; Sun et al. 2019) focuses on delivering the human pose structure, but they do not leverage the human pose structure to infer the whole-body shape information. The three primary challenges in inferring the pose and body shape of multiple people, especially those who are socially engaged, call for an effective model. First, an image can have an undefined number of individuals that can appear at any location and distance. Second, human-to-human interactions induce complex spatial interference due to contacts, obstructions, and articulations of the limbs, making it difficult to associate body parts. Third, the computational cost and complexity tends to increase with the number of people in the image. Proposals for pose estimation (Chen et al. 2018; Fang et al. 2017; Huang, Gong, and Tao 2017; Li et al. 2019) aim to tackle these challenges by taking a top-down approach to first detect a person in the image and then estimate the human pose. However, these top-down approaches require the pose estimator to run iteratively for each detected person, thus degrading model efficiency. Moreover, establishing segmentation head using a top-down approach can further increase the computational cost. Recent studies (He et al. 2017; Papandreou et al. 2018) suggest that joint estimation of human pose and body segmentation can produce state-of-the-art results using a large-scale pose and segmentation dataset (e.g., COCO (Lin et al. 2014)). However, existing approaches used human poses to refine the pixel-wise clustering for segmentation and thus could not perform well on the segmentation task (Papandreou et al. 2018). Further, existing models incur overhead due to the extra computation of a person detector (He et al. 2017) and suffer scalability issues for instance segmentation (Zhang et al. 2019), which makes them unsuited for populated scenarios.

In this paper, we propose Pose*Plus*Seg, a novel bottom-up pose estimation and instance segmentation model specifically designed for joint human pose estimation and instance segmentation. Pose*Plus*Seg employs a bottom-up approach to first detect keypoints, and then connects those keypoints to form several instances of human pose. This approach detects the human body without a bounding box concept thus enabling effective pose estimation, along with instance segmentation, without incurring the runtime complexity of the top-down approach.

Pose*Plus*Seg tackles joint human pose estimation and instance segmentation via two pipelines: (i) a pose estimation pipeline and (ii) an instance segmentation pipeline. The pose estimation pipeline generates a strong keypoint heat map that estimates the relative displacement between each pair of keypoint and improves the precision of long-range, occluded, and proximate keypoints (Figure 1b). Using the strong keypoint heat map, a body heat map is produced to identify the human body and improve the keypoint detection confidence score (Figure 1c). Once the keypoints are identified, then a pose estimation algorithm is used to connect the



(a) Input Image (b) SKHM (c) BHM (d) PE

(e) MO (f) SM (g) ISM (h) PIS

Figure 1: Pose*Plus*Seg produces (b) a strong keypoints heat map (SKHM) for detecting keypoints of individuals and (c) a body heat map (BHM) for identifying the human body position to generate (d) human pose estimation (PE). Pose*Plus*Seg also produces (e) a mask offset (MO) for finding the embedding space for each individual to generate (f) a segmentation mask (SM) to determine the human estimated shape structure and (g) an instance segmentation mask (ISM) to segment each individual separately and the final (h) human pose and instance segmentation (PIS) is produced from both the PE and ISM.

keypoints into human instances to make the pose structure (Figure 1d).

The segmentation pipeline generates a mask offset that defines the embedding space to associate pixels with the right instance centroid to predict the shape of each individual (Figure 1e). The mask offset helps to generate a segmentation mask to present the human body 2D shape structure (Figure 1f). Once all the labeled pixels are defined for each instance, then an instance segmentation algorithm is used to generate an instance segmentation mask (Figure 1g). Finally, we present a new pose estimation and instance segmentation algorithm to produce the joint structure of the human pose and its corresponding instance segmentation (Figure 1h).

We evaluate the performance of Pose*Plus*Seg using the COCO dataset (Lin et al. 2014). Pose*Plus*Seg outperforms existing joint pose and segmentation techniques demonstrating 0.728 mean average precision (mAP) for human pose estimation and 0.445 mAP for instance segmentation. In summary, we make the following contributions:

- We introduce the strong keypoint heat map to detect both soft and hard keypoints to accurately estimate human pose, and the body heat map for locating the individuals and increasing the keypoint confidence scoring;
- We introduce a mask offset which defines the keypoint as a center of attraction for the pixels in the embedding space to identify the human body shape structure;
- We utilize a refine network to produce refined keypoints and instance segmentation mask, and introduce a new pose and instance segmentation algorithm to visualize the joint human pose and instance segmentation;
- Our in-depth experiments demonstrate Pose*Plus*Seg is an effective method to perform joint human pose and instance segmentation.

Related Work

Multi and Single Person Pose Estimation There are two approaches for human pose estimation; top-down (Girshick 2015; Iqbal and Gall 2016; Chen et al. 2018; Fang et al. 2017; Huang, Gong, and Tao 2017; Szegedy et al. 2015; Li et al. 2019) and bottom-up (Insafutdinov et al. 2016; Pishchulin et al. 2016; Cao et al. 2017). The top-down approach identifies keypoints surrounded by a bounding box detector. The bottom-up method first detects human keypoints, and then combine the adjacent keypoints to generate a pose. The Regional Multi-Person Pose Estimation (Fang et al. 2017) is a top-down approach to extract a high-quality single-person region from inaccurate bounding boxes but takes a long execution time. DeeperCut (Insafutdinov et al. 2016) is a ResNet-based bottom-up approach to improve the body part detectors by generating effective proposals. However, it is computational constant to formulate the association between keypoints using the integer linear scheme. A partitioning and labeling formulation of a set of body-part hypotheses were proposed with CNN-based part detectors (Pishchulin et al. 2016). Such a formulation is a non-deterministic polynomial (NP) hard problem and requires significant computational power.

Instance Segmentation There are two primary approaches for instance segmentation: (1) single-stage (Dai et al. 2016; Long, Shelhamer, and Darrell 2015; Bolya et al. 2019) and (2) multi-stage (He et al. 2017; Ren et al. 2015). The single-stage approach first create intermediate and distributed feature maps based on the entire image, then assembles the extracted features for each instance to form the final mask. The InstanceFCN (Dai et al. 2016) uses fully convolutional networks to create several instance-sensitive scoring maps and applies the assembly module to the output instance. It requires repooling and other non-trivial computations (e.g., mask voting), making the real-time processing infeasible. YOLACT (Bolya et al. 2019) generates a set of prototype masks, then uses coefficient per instance mask to produce the instance-level segmentation. This way, the prototype masks make YOLACT's computation cost constant. Multi-stage instance segmentation follows the detect-then-segment paradigm. In this approach, first, it performs bounding box detection and then the pixels are classified to obtain the final mask in the bounding box region. Mask R-CNN (He et al. 2017) is based on the multi-stage instance segmentation that extends Faster R-CNN (Ren et al. 2015) by adding a branch of predicting segmentation mask for each Region of Interest (RoI). The method presented by (Liu et al. 2018) improves the accuracy of Mask R-CNN by enriching the Feature Pyramid Network (FPN) features.

Human Pose Estimation and Instance Segmentation Mask R-CNN (He et al. 2017) proposed the human pose estimation along with instance segmentation. However, this method suffers from the extra computation overheads of a person detector. Pose2Seg (Zhang et al. 2019) proposed human pose-based instance segmentation. This method separates instances based on the human pose, rather than by region proposals. However, it takes previously generated poses as input instead of a normal image. The PersonLab (Pa-

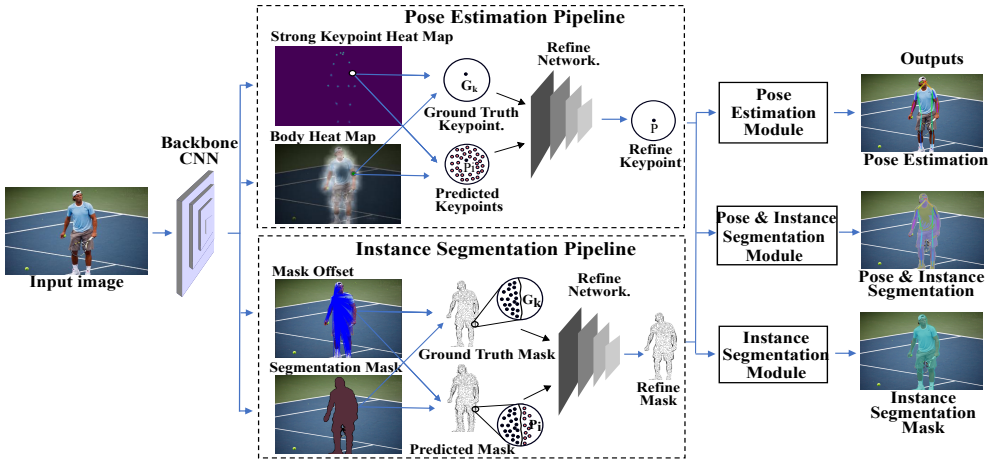


Figure 2: Pose*Plus*Seg consists of two main pipelines: (i) pose estimation pipeline uses strong keypoint heat map and body heat map to predict the human estimated pose, and (ii) instance segmentation pipeline uses segmentation mask and mask offset to define instance-level segmentation.

pandreou et al. 2018) approach is used to detect individual keypoints and use greedy decoding process to group keypoints into person instances. This method also reports a part-induced geometric embedding descriptor for human class instance segmentation but fails to perform instance segmentation for scenes with highly-entangled instances.

Pose*Plus*Seg Overview

Pose*Plus*Seg consists of (i) Pose Estimation Pipeline (ii) Instance Segmentation Pipeline Figure 2. The input image is feed into the backbone CNN to learn the feature maps of each human instance, then the learned feature maps are feed into their corresponding pipelines.

(i) Pose estimation pipeline generates the SKHM based on the keypoints feature maps and present it as a human skeleton structure. It also produce BHM utilizing the keypoints proposals to detect the position of each individual, to improve the keypoint confidence score. The output of the pose estimation pipeline feeds into the pose estimation module for the final human pose estimation.

(ii) Instance segmentation pipeline takes mask feature maps for each labeled human instance and generates MO. The MO defines the association between the pixels in the embedding space to predict the 2D shape of the individuals. The output of the instance segmentation pipeline feeds into the instance segmentation module to generate the ISM.

The system employs a refine network one for each pipeline in order to refine the keypoints and mask. Pose*Plus*Seg uses the information generated from both pipelines as inputs to the pose and instance segmentation module to generate the final human pose and instance segmentation.

Figure 3: Example of body heat map.

Pose Estimation Pipeline

Body Heat Map Unlike traditional bounding box human detection (Zhou and Yuan 2017, 2018), Pose*Plus*Seg presents a new idea of human body detection through the Body Heat Map (BHM). Along with human body detection, the BHM also helps to enhance the predicted keypoint confidence score. An example of the BHM is shown in Figure 3. To produce the BHM, we sequentially calculate a disk of pixels $D = 2\pi R$ where radius R=32 pixels for all predicted keypoints P_k belonging to each individual I summarized in Equation 1. P_k is a group of pixels that represent human body keypoints. While producing the BHM, we utilize all the pixels belonging to each keypoint disk to cover the whole human body. We reduce the intensity of each pixel in the image except those belonging to the disk surrounded by a keypoint. To maintain the bright resolution, we multiply alpha α = 0.4 to each keypoint disk.

$$I = \sum_{k=1}^{n} P_k(\alpha), \text{where } P_k = 2\pi R. \quad (1)$$

During the inference, the BHM algorithm detects the human body with high confidence and also increases the keypoint confidence score.

Strong Keypoint Heat Map The pose estimation pipeline perform two tasks: first, Pose*Plus*Seg generates a Strong Keypoint Heat Map (SKHM) is illustrated in Figure 4, which is a building base for the pose estimation. In this stage, each

Figure 4: Example of strong keypoints heat map.

individual keypoint is detected. Second, each detected keypoint is refined through the refine network.

Suppose p_i represents the 2D keypoint position in the image, where i = {1,..., N} are mapped to the positions of the pixels. A keypoint disk $D_R(q) = \{p : \|p–q\| \leq R\}$ of radius R focused at point q. Also, consider $q_{j,k}$ be the 2D position of the k^{th} keypoint of the j^{th} person instance, where j={1,...,I} and I is the number of person instances in the image. For each known keypoint type k={1,...,M} a binary classification approach is used as follows: For every predicted keypoint pixel K_i, such that K_i=1 if the pixel $\in D_R$ for each person instance j, otherwise $K_i = 0$. Thus, for every keypoint, we have independent dense binary classification tasks. The radius of the SKHM disk is set to R = 32 pixels. We predict a disk around a specific keypoint of any person in the image. For this, we empirically obtain the R value and set it to 32. The value R is constant for all experiments reported in this paper. In order to equally consider all person instances, we choose a disk radius that does not scale according to the instance size. While training the network, the SKHM loss is computed based on the annotated image positions. It then back-propagates across the entire image, excluding the range that includes individuals who are not fully annotated with keypoints (e.g., crowded areas and small individual segments).

During the model training process, a group of raw pixels in the SKHM disk feeds into the refine network, the prediction loss is penalized by the L_1 loss function, averaging and back-propagating the error only at the pixel positions $p \in D_R$ with the ground truth keypoint position. We reduce the error in the keypoint disk (radius R = 32 pixels) by normalizing them and making a dynamic range that is compatible with the heat map classification loss.

Pose Generator Pose*Plus*Seg utilizes a pose generator algorithm to collect all keypoints and turn them into individual instances. Initially, a queue store all the keypoints along with the keypoint positions p_i. However, two or more keypoints can be discovered for one keypoint (x and y coordinates). These points are used as building blocks to detect instances. At each iteration, if the position p_i of the current detection point is inside a disk $D_R(q_{j,k})$ of a previously detected person instance j, then the algorithm skips such point because it is already recognized. It usually occurs when two keypoints overlap or partially touch. For this matter, we utilize non-maximum suppression. Then a new detection instance j' starts with the k-th keypoint at location $q_{j,k} = p_i$, and delivers it to the new point. We then follow the pose kinematic graph to greedily join pairs of adjoined keypoints.

Most approaches for pose estimation are based on the torso or nose keypoints, which sometimes fail to estimate an individual pose when they are not clearly visible in the image. In contrast, our pose generator algorithm does not serve any keypoint preferentially. But, starts from the most confident detection, such as the nose, and connects all of the keypoints one by one. Moreover, it manages rigorous situations where a significant part of the individual is not visible.

Instance Segmentation Pipeline

Mask Offset Human instance segmentation is a pixel-level classification challenge, i.e., how to connect pixels with the right instance I. We define the mask offset (MO) at each image position x_i inside the segmentation mask of an annotated person instance j with 2-D mask pixel positions $y_{j,k}$ where $k = 1, ..., M$, which point from image position x_i to the position of the k^{th} keypoint of the corresponding instance I_j. At each image position x_i of a semantically identified human instance, the embedding vector $e(x_i)$ reflects our local approximation of the absolute location of each mask pixel of an individual to whom it corresponds, i.e., it represents the person's expected shape structure. To this end, for each pixel, we learn the MO, illustrated in (Figure 5a), which points to the centroid (right shoulder).Here, we take advantage of the keypoint localization to use them as a center of attraction for each instance mask pixel. We also define the MO edge boundary to bound the MO pixel vectors in the embedding space. The purpose of the instance segmentation is to cluster a group of pixels $\mathcal{P} = \{p_0, p_1, p_2, ..., p_i\}$ and its 2-dimensional embedding vectors $e(p_i)$, into a set of instances $\mathcal{I} = \{S_0, S_1, S_2, ..., S_j\}$ to provide a shape along with pose. Pixels are assigned to their corresponding centroid:

$$C_k = \frac{1}{N} \sum_{p_i \in S_j} p_i. \tag{2}$$

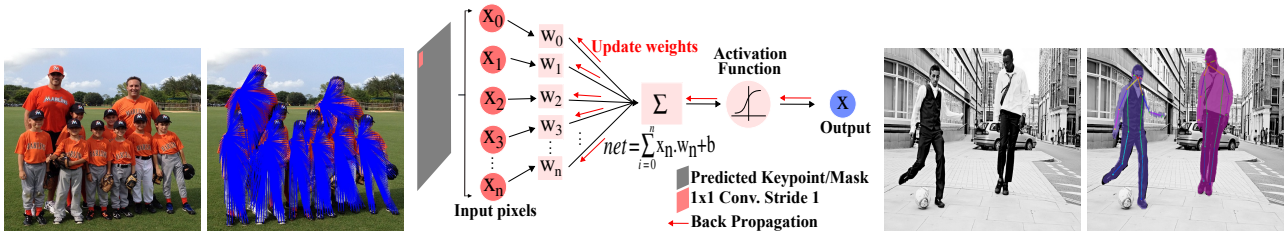
This is attained by defining mask offset vector v_i for each known pixel p_i, so that the resulting embedding $e_i = p_i + v_i$ points from its respective instance centroid. We penalize MO loss by an L_1 loss function throughout model training, averaging and back-propagating the loss at the only image position x_i that corresponds to an instance of a specific individual entity:

$$L = \sum_{i=0}^{n} \|v_i - \hat{v}_i\|, \tag{3}$$

where $\hat{v}_i = C_k - p_i$ for $p_i \in S_j$. In order to cluster the pixels to their centroid, first, it is important to specify the positions of the instance centroids and second to assign pixels to a particular instance centroid. We utilize a density-based clustering algorithm to first locate a set of centroids as a center of attraction. Having obtained an array of centroids $\mathcal{C} = \{C_0, C_1, ..., C_K\}$, we next to add pixels to a particular instance based on a minimum distance-to-centroid metric:

$$e_i \in S_j : k = \arg\min_{\mathcal{C}} \|e_i - C\|. \tag{4}$$

Instance Segmentation Generator Similar to the pose estimation pipeline, the instance segmentation pipeline performs two tasks. First Segmentation Mask (SM) is generated for each individual employing the backbone network.



(a) Mask offset (b) Refine Net. (c) Pose and Instance Segmentation

Figure 5: (a) Shows mask offsets where the right shoulder is defined as a center of attraction. (b) Describes the refine network. (c) Depicts an example pose estimation and instance segmentation.

Second, the segmentation mask is refined through the refine network. Each SM provides raw pixels matrix corresponding to each individual, and ignore the background, and other classes. We follow the embedding-based method for this task, similar to (Newell, Huang, and Deng 2017; Fathi et al. 2017; De Brabandere, Neven, and Van Gool 2017). At each pixel position x_i, we compute an embedding vector $e(x_i)$, and cluster them to obtain the final object instance.

To obtain the final Instance Segmentation Mask (ISM): (i) We find the image positions x_i belonging to an individual, i.e., pixels with a high probability to lie in the ground truth mask region. A high probability indicates that the pixel embedding e_i is near to the centroid of an instance and is likely to correspond to that particular instance, However a low probability implies that the pixel is more likely belong to the background or another instance. More precisely, if the probability $p(e_i) \geq 0.5$, then that pixel at location x_i will be assigned to instance I. (ii) We equate each pixel p_i with each observed individual instance I_j that satisfies the embedding distance metric, as given in Equation 4. The relative distance threshold as t = 0.25 in all proclaimed experiments. To evaluate the COCO segmentation task and obtain the average accuracy output figures, we use the same instance-level score provided by the previous pose estimation stage.

Refine Network

(Figure 5b) shows a refine network using a 1x1 convolutional filter with a stride of 1. It generates a dense vector of raw pixels x_n and computes the probability x_m for each pixel belong to at least one person in the image:

$$\mathbf{P}(x_m = 1|x_n) = \frac{e^{w.\varphi(x_n)}}{1 + e^{w.\varphi(x_n)}}. \quad (5)$$

We established two refine networks for both the pose estimation and instance segmentation pipelines. The predicted keypoints generated by the backbone network for pose estimation produces the SKHM disk, which is further fed to the refine network for refining. Where, in the case of Instance Segmentation, the predicted mask pixels pass to the refine network. The main aim of the refine network is to refine the predicted keypoints and mask to improve the precision. The input vector of pixels x_n is multiplied by random weights w and bias b is added. The sum function is applied to combine all of the weights and biases. The sigmoid activation function is used to find the predicted probability for each pixel belongs to an instance.

We compute and back-propagate the average loss of all the regions in the image that have been annotated. We use the L_1 loss function to minimize the distance between the estimated and ground-truth coordinates. The loss function L_{ref} is defined as follows:

$$L_{ref} = \frac{1}{N} \sum_{n=1}^{m} \|x_n - \hat{x}_n\|, \quad (6)$$

where x_n is the ground truth mask or keypoint pixel coordinates and $\hat{x}_n$ indicate the predicted coordinates. The final refined pixels are fed into their corresponding modules to generate the output visually.

Pose and Instance Segmentation

We introduce a new PIS algorithm that takes the refine pixels produced by the refine network and present them visually, as illustrated in (Figure 5c). As shown in Figure 2, the PIS module takes the pose and instance segmentation information from both pipelines and use them for pose and instance segmentation. Initially, the algorithm identifies all the keypoints and their 2D coordinates and stores them in a priority queue. However, in some cases, more than one keypoint can be identified for a single keypoint position p_i. This occurs when the two keypoints are overlapped or entangled. We handle such issues by leveraging non-maximum suppression. Next, a new instance j' starts with the k^{th} keypoint detected at image position x_i considers point.

Simultaneously, the PIS performs instance segmentation for all detected human instances. The system identifies pixels positions x_i belonging to an instance, i.e., those pixels with a high probability to lie in the embedding space e_i. Then, these pixels are assigned to the relevant instance if the pixel embedding is close to the instance centroid C_k. More specifically, If the pixel probability $p(s_i) \geq 0.5$, then that pixel at position x_i is assigned to the relevant human instance. However, if the pixel value is less than the threshold value 0.5, then it is more likely to belong to another instance or background. We process both the pose estimation and instance segmentation tasks independently to prevent performance degradation and maintain a high score.

Experiments

We evaluate the performance of Pose*Plus*Seg model on the standard COCO keypoint dataset. Our model is trained end-to-end using the COCOPersons training set. Experiments and ablation studies are conducted on the COCO *test* and *minival* set.

Training Setup We use the CNN backbone networks ResNet-101 (RN101) and ResNet-152 (RN152) (He et al. 2016) for training and testing. The hyperparameters for training are: learning rate = $0.1 \times e^{-4}$, image size = 401×401, and batch size = 2 implemented on one NVIDIA GeForce GTX 1080 Ti. We conduct synchronous training for 500 epochs with stochastic gradient descent using TensorFlow 1.13.

Experimental Results Table 1 compares Pose*Plus*Seg with 5 SOTA methods on the COCO *minival* dataset: 8-stage Hourglass (Newell, Yang, and Deng 2016), CPN (Chen et al. 2018), SimpleBaseline (Xiao, Wu, and Wei 2018), CMU-Pose with refinement (Cao et al. 2017), and PersonLab (Papandreou et al. 2018); The first three methods are the top-down approaches while the last two methods are the bottom-up methods. Compared with top-down approaches Pose*Plus*Seg achieves 10.8%, 2.3%, and 3.3% gains, respectively. However, compared with bottom-up approaches Pose*Plus*Seg gains an increment of 21.9%, and 11.8%.

Models	Backbone	AP	AP$^{.50}$	AP$^{.75}$	AP^{M}	AP^{L}
Top-down:						
Hourglass	8-stage	0.671	-	-	-	-
CPN	RN50	0.727	-	-	-	-
SimpleBaseline	RN152	0.720	0.893	0.798	0.687	0.789
Bottom-up:						
CMU-Pose	-	0.610	0.849	0.675	0.563	0.693
PersonLab	RN152	0.665	0.862	0.719	0.623	0.732
Pose*Plus*Seg (ours)	RN101	0.717	0.873	0.726	0.648	0.776
Pose*Plus*Seg (ours)	RN152	0.744	0.894	0.748	0.675	0.811

Table 1: Performance comparison on the COCO keypoint *minival* set.

Table 2 compares the performance on the COCO keypoint *test* dataset demonstrating that Pose*Plus*Seg outperforms bottom-up approaches, CMU-Pose (Cao et al. 2017), Associative Embedding (Newell, Huang, and Deng 2017), PersonLab (Papandreou et al. 2018), and MultiPoseNet (Kocabas, Karagoz, and Akbas 2018). Specifically, the Pose*Plus*Seg yields a mAP of 0.728 on the ResNet-152 base architecture. The test results also show that the performance of Pose*Plus*Seg surpasses that of top-down approaches, i.e., Mask-RCNN (He et al. 2017), G-RMI (Papandreou et al. 2017), Integral Pose Regression (Sun et al. 2018), and CPN (Chen et al. 2018).

Table 3 and Table 4 present the results of the COCO Segmentation *minival* and *test* sets. Pose*Plus*Seg demonstrates a mAP of 0.563 on the *minival* set, and improves the AP by 0.145 compared to PersonLab (Papandreou et al. 2018) and by a 0.008 AP compared to Pose2Seg (Zhang et al. 2019). Moreover, on the test set, Pose*Plus*Seg achieves a mAP of 0.445 and improves the AP by 0.074 over Mask-RCNN (He et al. 2017) and by 0.028 over PersonLab.

Models	Backbone	AP	AP$^{.50}$	AP$^{.75}$	AP^{M}	AP^{L}
Top-down:						
Mask-RCNN	RN50-fpn	0.631	0.873	0.687	0.578	0.714
G-RMI COCO-only	RN101	0.649	0.855	0.713	0.623	0.700
Integral Pose Regress	RN101	0.678	0.882	0.748	0.639	0.740
G-RMI + extra data	RN101	0.685	0.871	0.755	0.658	0.733
CPN	RN50	0.721	0.914	0.800	0.687	0.772
Bottom-up:						
CMU-Pose	-	0.618	0.849	0.675	0.571	0.682
Ass. Emb. (m-scale)	Hourglass	0.630	0.857	0.689	0.580	0.704
Ass. Emb. (mscale,ref)	Hourglass	0.655	0.868	0.723	0.606	0.726
PersonLab (s-scale)	RN152	0.665	0.880	0.726	0.624	0.723
PersonLab (m-scale)	RN152	0.687	0.890	0.754	0.641	0.755
MultiPoseNet	-	0.696	0.863	0.766	0.650	0.763
Pose*Plus*Seg (ours)	RN101	0.706	0.865	0.768	0.655	0.774
Pose*Plus*Seg (ours)	RN152	0.728	0.884	0.787	0.678	0.794

Table 2: Performance on the COCO keypoint *test* set. The AP at IOU=.5:.05:.95, AP$^{.05}$ at IOU=.05 (Pascal VOC metric), AP$^{.75}$ at IOU=.75 (strict metric), AP$^{\mathrm{M}}$ corresponds to the AP for medium objects: 32^2 <area <96^2, and AP$^{\mathrm{L}}$ corresponds to the AP for large objects: area >96^2. S-scale refers to single-scale, m-scale to multi-scale, and ref to refinement.

Models	Backbone	AP	AP$^{.50}$	AP$^{.75}$	AP^{M}	AP^{L}
PersonLab (s-scale)	RN101	0.382	0.661	0.397	0.476	0.592
PersonLab (s-scale)	RN152	0.387	0.667	0.406	0.483	0.595
PersonLab (m-scale)	RN101	0.414	0.684	0.447	0.492	0.621
PersonLab (m-scale)	RN152	0.418	0.688	0.455	0.497	0.621
Pose2Seg	RN50-fpn	0.555	-	-	0.498	0.670
Pose*Plus*Seg (ours)	RN101	0.556	0.699	0.546	0.499	0.673
Pose*Plus*Seg (ours)	RN152	0.563	0.701	0.557	0.509	0.683

Table 3: Performance comparison on the COCO Segmentation (human category) *minival* set.

Models	Backbone	AP	AP$^{.50}$	AP$^{.75}$	AP^{M}	AP^{L}
Mask-RCNN	RNext101	0.371	0.600	0.394	0.399	0.535
PersonLab (s-scale)	RN101	0.377	0.659	0.394	0.480	0.595
PersonLab (s-scale)	RN152	0.385	0.668	0.404	0.488	0.602
PersonLab (m-scale)	RN101	0.411	0.686	0.445	0.496	0.626
PersonLab (m-scale)	RN152	0.417	0.691	0.453	0.502	0.630
Pose*Plus*Seg (ours)	RN101	0.432	0.699	0.469	0.515	0.648
Pose*Plus*Seg (ours)	RN152	0.445	0.794	0.471	0.524	0.651

Table 4: Performance comparison on the COCO Segmentation (human category) *test* set.

Impact of the SKHM on Keypoint Detection Now, we run a set of ablations to qualitatively analyze the impact of each component of Pose*Plus*Seg. We first compare SKHM with different keypoint detection algorithms that rely on keypoint heatmaps for keypoint detection. Table 5 presents the performance of the SKHM with the SOTA bottom-up approaches: CMU-Pose (Cao et al. 2017), MultiPoseNet (Kocabas, Karagoz, and Akbas 2018), and PersonLab (Papandreou et al. 2018). We noted that the SKHM out-performs in all categories by introducing a keypoint disk around keypoints. Using ResNet152 as a backbone network, the SKHM shows a spike increase in the AP, except for MultiPoseNet (Kocabas, Karagoz, and Akbas 2018), which has a lead of 0.002 in AP $^{.75}$ due to a specially designed Pose Residual Network (PRN). The SKHM also leads in keypoint detection for AP^{medium} and AP^{large} humans instances.

Models	AP	$AP^{.50}$	$AP^{.75}$	AP^M	AP^L
CMU-Pose	0.610	0.849	0.675	0.563	0.693
MultiPoseNet	0.643	0.882	0.750	0.596	0.739
PersonLab	0.665	0.862	0.719	0.623	0.732
PosePlusSeg:					
RN101 (SKHM)	0.717	0.873	0.726	0.648	0.776
RN152 (SKHM)	0.744	0.894	0.748	0.675	0.811

Table 5: Keypoint detection comparison between Pose*Plus*Seg's SKHM and keypoint heatmap approaches on the COCO *minival* set.

Impact of the BHM on the Keypoint Confidence Score We examine the effect of BHM on the keypoint confidence score. Table 6 shows the 17 keypoint detection confidence scores generated by the keypoint disks of radius $R = 8$, 16, and 32. The keypoint confidence detection score for a bigger disk (R=32) is high because it provides a larger radius to the classifier to reach the ground-truth value.

Left	R=8	R=16	R=32	Right	R=8	R=16	R=32
Nose	0.775	0.804	0.830	-	-	-	-
Eye	0.678	0.726	0.756	Eye	0.662	0.698	0.738
Ear	0.676	0.701	0.721	Ear	0.634	0.676	0.719
Shoulder	0.618	0.643	0.653	Shoulder	0.617	0.643	0.662
Elbow	0.548	0.604	0.625	Elbow	0.559	0.582	0.599
Wrist	0.583	0.618	0.636	Wrist	0.526	0.553	0.608
Hip	0.521	0.558	0.586	Hip	0.553	0.583	0.613
Knee	0.618	0.648	0.674	Knee	0.647	0.677	0.692
Ankle	0.671	0.683	0.729	Ankle	0.638	0.654	0.707

Table 6: Average keypoint detection confidence scoring based on different keypoint disk radius R values.

Impact of the BHM on the SKHM We examine the effect of the BHM on SKHM in terms of the quality of the keypoint confidence detection. For this, we built a version of Pose*Plus*Seg with the BHM disabled. Table 7 shows the keypoint mAP generated from the SKHM and the SKHM+BHM using the ResNet101 and ResNet152 backbone networks. We see that the SKHM along with the BHM improves model keypoint AP by 1.6% and 0.5% while using the ResNet101 and ResNet152 backbones, respectively. The average increases of each keypoint confidence detection score using the BHM are listed in Table 8.

Model	AP	$AP^{.50}$	$AP^{.75}$	AP^M	AP^L
PosePlusSeg:					
RN101 (SKHM)	0.717	0.873	0.726	0.648	0.776
RN152 (SKHM)	0.744	0.894	0.748	0.675	0.811
RN101 (SKHM+BHM)	0.729	0.877	0.734	0.658	0.794
RN152 (SKHM+BHM)	0.748	0.893	0.751	0.675	0.816

Table 7: Impact of using the BHM with the SKHM.

Impact of the Mask Offset on Human Segmentation We experiment the Mask Offset that plays an important role in the task of human instance segmentation by defining centroids as a center of attraction for embedded pixels. We compare Pose*Plus*Seg's Mask Offset with PersonLab (Papandreou et al. 2018) and Pose2Seg (Zhang et al. 2019) human segmentation models. Table 9 shows the mask offset of an instance segmentation pipeline achieves a high accuracy trade-off relative to PersonLab and Pose2Seg.

Left	w/o BHM	w BHM	Right	w/o BHM	w BHM
Nose	0.655	0.830	-	-	-
Eye	0.631	0.756	Eye	0.635	0.738
Ear	0.608	0.721	Ear	0.606	0.719
Shoulder	0.583	0.653	Shoulder	0.562	0.662
Elbow	0.523	0.625	Elbow	0.487	0.599
Wrist	0.506	0.636	Wrist	0.514	0.608
Hip	0.458	0.586	Hip	0.476	0.613
Knee	0.551	0.674	Knee	0.583	0.692
Ankle	0.598	0.729	Ankle	0.621	0.707

Table 8: Average keypoint detection confidence score with and without the BHM.

Models	AP	$AP^{.50}$	$AP^{.75}$	AP^M	AP^L
PersonLab	0.418	0.688	0.455	0.497	0.621
Pose2Seg	0.555	-	-	0.498	0.670
PosePlusSeg:					
RN101 (mask offset)	0.556	0.699	0.546	0.499	0.673
RN152 (mask offset)	0.563	0.701	0.557	0.509	0.683

Table 9: Mask offset performance on the COCO Segmentation (human category) *minival* set.

Runtime Performance and Number of Parameters Finally, we compare runtime performance with SOTA approaches to quantitatively analyze the model efficiency as shown in Table 10. We also compare GFLOPs and number of parameters in Table 11. Results demonstrate Pose*Plus*Seg enables real-joint pose estimation and instance segmentation with less computation and parameters.

Models	Backbone	Task	Runtime	GPU
Mask R-CNN	RN101	Boxes&Seg.&Pose	200ms (5fps)	M40
Pose2Seg	RN50-fpn	Inst. Seg.	50ms (20fps)	TitanX
Pose*Plus*Seg	RN152	Pose	56ms (17fps)	1080Ti
Pose*Plus*Seg	RN152	Inst. Seg.	59ms (16fps)	1080Ti
Pose*Plus*Seg	RN152	Pose&Seg.	68ms (14fps)	1080Ti
Pose*Plus*Seg	RN152	Pose	28ms (34fps)	RTX
Pose*Plus*Seg	RN152	Inst. Seg.	29ms (32fps)	RTX
Pose*Plus*Seg	RN152	Pose&Seg.	34ms (28fps)	RTX

Table 10: Comparison of runtime performance.

Models	Backbone	Input Size	FLOPs	#Para	mAP
Hourglass	8-stage	256x192	14.3G	25.1M	0.669
CPN	RN50	256x192	6.20G	27.0M	0.686
CPN*	RN50	384x288	6.20G	27.0M	0.694
SimpleBaseline	RN152	256x192	15.7G	68.6M	0.720
Pose*Plus*Seg	RN50	256x192	4.04G	25.5M	0.695
Pose*Plus*Seg	RN101	256x192	7.69G	44.5M	0.717
Pose*Plus*Seg	RN152	256x192	11.34G	60.1M	0.744

Table 11: Comparison of GFLOPs and parameters.

Conclusion

We propose a bottom-up approach to tackle the task of joint human pose estimation and instance segmentation. Pose*Plus*Seg generates a strong keypoint heat map and body heat map to accurately predict individual keypoints. In addition, a mask offset is used to define the association between the pixels belonging to an instance to present the human body instance segmentation. Our in-depth evaluation using the COCO keypoint challenging dataset demonstrates the effectiveness of Pose*Plus*Seg for joint human pose estimation and instance segmentation.

Acknowledgements

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Grand ICT Research Center support program (IITP-2020-0-101741) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation).

References

Bolya, D.; Zhou, C.; Xiao, F.; and Lee, Y. J. 2019. Yolact: Real-time instance segmentation. In *ICCV*.

Cao, Z.; Simon, T.; Wei, S.-E.; and Sheikh, Y. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*.

Chen, Y.; Wang, Z.; Peng, Y.; Zhang, Z.; Yu, G.; and Sun, J. 2018. Cascaded pyramid network for multi-person pose estimation. In *CVPR*.

Dai, J.; He, K.; Li, Y.; Ren, S.; and Sun, J. 2016. Instance-sensitive fully convolutional networks. In *ECCV*.

De Brabandere, B.; Neven, D.; and Van Gool, L. 2017. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*.

Fang, H.-S.; Xie, S.; Tai, Y.-W.; and Lu, C. 2017. Rmpe: Regional multi-person pose estimation. In *ICCV*.

Fathi, A.; Wojna, Z.; Rathod, V.; Wang, P.; Song, H. O.; Guadarrama, S.; and Murphy, K. P. 2017. Semantic instance segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277*.

Girshick, R. 2015. Fast r-cnn. In *ICCV*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *ICCV*.

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

Huang, S.; Gong, M.; and Tao, D. 2017. A coarse-fine network for keypoint localization. In *ICCV*.

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

Iqbal, U.; and Gall, J. 2016. Multi-person pose estimation with local joint-to-person associations. In *ECCV*.

Kocabas, M.; Karagoz, S.; and Akbas, E. 2018. Multiposenet: Fast multi-person pose estimation using pose residual network. In *ECCV*.

Li, W.; Wang, Z.; Yin, B.; Peng, Q.; Du, Y.; Xiao, T.; Yu, G.; Lu, H.; Wei, Y.; and Sun, J. 2019. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Liu, S.; Qi, L.; Qin, H.; Shi, J.; and Jia, J. 2018. Path aggregation network for instance segmentation. In *CVPR*, 8759–8768.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Newell, A.; Huang, Z.; and Deng, J. 2017. Associative embedding: End-to-end learning for joint detection and grouping. In *NeurIPS*.

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

Papandreou, G.; Zhu, T.; Chen, L.-C.; Gidaris, S.; Tompson, J.; and Murphy, K. 2018. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *ECCV*.

Papandreou, G.; Zhu, T.; Kanazawa, N.; Toshev, A.; Tompson, J.; Bregler, C.; and Murphy, K. 2017. Towards accurate multi-person pose estimation in the wild. In *CVPR*.

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

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*.

Sun, K.; Xiao, B.; Liu, D.; and Wang, J. 2019. Deep high-resolution representation learning for human pose estimation. In *CVPR*.

Sun, X.; Xiao, B.; Wei, F.; Liang, S.; and Wei, Y. 2018. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 529–545.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*.

Xiao, B.; Wu, H.; and Wei, Y. 2018. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, 466–481.

Zhang, S.-H.; Li, R.; Dong, X.; Rosin, P.; Cai, Z.; Han, X.; Yang, D.; Huang, H.; and Hu, S.-M. 2019. Pose2seg: Detection free human instance segmentation. In *CVPR*.

Zhou, C.; and Yuan, J. 2017. Multi-label learning of part detectors for heavily occluded pedestrian detection. In *ICCV*, 3486–3495.

Zhou, C.; and Yuan, J. 2018. Bi-box regression for pedestrian detection and occlusion estimation. In *ECCV*, 135–151.