Localizing Human Keypoints beyond the Bounding Box

Soonchan Park[1,2] Jinah Park [2]
[1] Electronics and Telecommunications Research Institute
[2] Korea Advanced Institute of Science and Technology
Republic of Korea
parksc@etri.re.kr jinahpark@kaist.ac.kr

Abstract

Since human pose is one of the most effective and popular sources for understanding human in various applications, there have been numerous researches on detecting keypoints of human body from the image source. However, when a human body is shown partially in the source image, estimation range is also restricted causing performance degradation in locating keypoints of human body. In this paper, we propose 'Position Puzzle' network and augmentation to leverage the performance of detecting keypoints including those outside the bounding box. Specifically, Position Puzzle Network expands the spatial range of keypoint localization by refining the position and the scale of the target's bounding box, and Position Puzzle Augmentation improves the performance of keypoint detector using the partial image in training. We prepare data by cropping COCO dataset and utilize them in training and evaluation. Under the prepared dataset, the proposed method enhances the performance of baseline network up to 37.6% and 30.6% in mAP and mAR, respectively, and effectively localizes keypoints positioned not only inside but also outside the bounding box. We also verify that the proposed method can localize keypoints beyond the bounding box in the original COCO dataset.

1. Introduction

Keypoint detection is one of the most popular approaches to estimate human pose by localizing pre-defined human body parts such as eyes, shoulders, and ankles [7, 11]. Numerous researches [14, 3, 6, 8, 9, 15, 23, 22, 20] have accomplished remarkable developments, and their results lead to the enhancement of subsequent researches on topic such as human object interaction[5, 21, 2, 28] and person re-identification[19, 13].

Keypoint detection from a partial image often caused by occlusion or a limited field of view of the camera is an unavoidable task. The machine generally shows deteriorated performance under the situation due to a lack of information to localize keypoints. To alleviate the problem, detecting keypoints *within* a partial image [22, 8, 4, 27, 16] has been studied and shown to improve the overall performance.

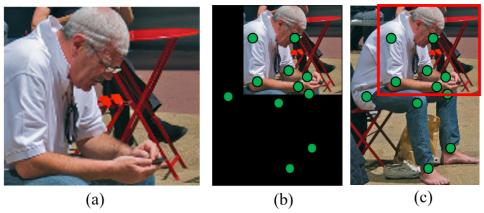


Figure 1. (a): Example of an image partially cropped by the red box in the image (c). (b): Coupled information constructed by the partial image and the complete keypoints. (c): Reference image.

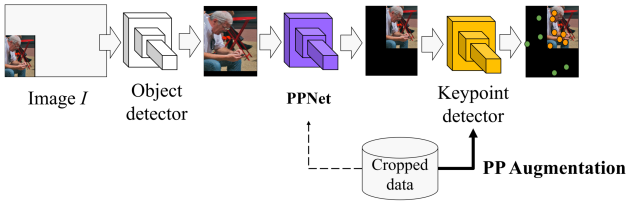


Figure 2. Position of two proposed methods in the top-down based keypoint detection. Position Puzzle Network is visualized as the purple pyramid and Position Puzzle Augmentation is involved in a training process of keypoint detector using cropped data.

On the other hand, when human attempt to analyze someone's pose from a partial image, we can localize not only the keypoints in the image but also keypoints outside the image. Visual clues in the partial image, such as size of the human, or the angle of a joint, allow us to see or locate keypoints of the human beyond the partial image. For instance, we can recognize that the man in Fig.1(a) is seated and so estimate where his hips, knees, and ankles are located by collecting cues like the length of his limbs and his bending knees. Such reasonings that human naturally do are not easy for Neural Networks, because (i) keypoint detectors have not learned patterns between a partial image and

complete keypoints like in Fig.1(b), and (ii) a spatial range of keypoint detection is inevitably restricted by the given bounding box or image.

In this paper, we mitigate the aforementioned problems by adding two components to a conventional process of top-down based keypoint detection like in Fig.2. *Position Puzzle Network*(PPNet) indicated by the purple box in Fig.2 is placed between the object detector and the keypoint detector. PPNet refines a bounding box for allowing the box to contain those excluded keypoints that might belong to the target. Consequently, the following keypoint detector expands its estimation regions and localizes keypoints even though some of them are located outside the box. Moreover, we propose a data augmentation method for keypoint detector namely *Position Puzzle Augmentation* which provides augmented data constructed by partial images and complete keypoints. The data augmentation method maximizes the effectiveness of PPNet and consequently, it enhances the performance of detecting keypoints outside the bounding box. Since there is no dataset to train complete information(bounding box and keypoints) from partial images, we also compile dataset by developing and applying a cropping method to an existing dataset to train and evaluate the proposed method.

Through various experiments based on the original data and the cropped dataset, we verify the effectiveness of the proposed method. The method enhances the performance of detecting keypoints from the partial image by approximately 37.6% and 30.6% in precision and recall, respectively. The visualizations of the estimation process also support these improvements. The major contributions of our work can be summarized as follows:

- We address a problem of localizing keypoints beyond the bounding box from partial images, and introduce related approaches to prepare datasets and evaluate performance.
- We propose a neural architecture called Position Puzzle Network which estimates the appropriate position and scale to refine the bounding box for the machine to look beyond the bounding box so that those missing keypoints nearby can be identified.
- We introduce a data augmentation technique called Position Puzzle Augmentation which enhances the performance of detecting keypoints beyond the bounding box without adding computational load.

2. Related work

2D keypoint detection for human Top-down based approach [14, 23, 8, 20] and bottom-up based approach [3, 9, 15] are two main streams of keypoint detection. In top-down based approaches using bounding box obtained by preceding human detector[12, 6, 17, 25], the final output is probability maps (i.e., heatmaps) which represent possibilities that a certain keypoint could be located at a certain pixel in the bounding box. Owing to the correspondence between the region of bounding box and the probability map, the keypoint detector only focuses on estimating the location of keypoints in the bounding box. However, in this paper, we proposed a method that allows the keypoint detector to expand its estimation range and localize keypoints beyond the bounding box.

Enhancing keypoint detection from partial image The bounding box of the target human does not always cover the complete human object due to occlusions or the limited field of view of the camera. The uncertainty generally diminishes the performance of detecting keypoints. Various studies have attempted to achieve invariant performance under these situations[22, 8, 4, 27, 16] by refining estimations in the occluded region[4] or applying data augmentation methods to produce such data [22, 8, 27, 16] for training. However, the researches still aim to enhance estimating information *within image* although the image only shows a part of the target. Thus, in this paper, we focus on allowing keypoint detector to expand its spatial range of estimation to outside the image.

3. Method

In this section, we introduce the definition of the problem to localize human keypoints beyond the bounding box. We explain a method of Position Puzzle Network which refines the bounding box to include complete human although a part of target human is excluded from given image. Then, we will discuss Position Puzzle Augmentation to train keypoint detector using augmented patterns coupled by partial image and entire keypoints.

Since we define a novel problem, there is no dataset for estimating a bounding box nor keypoints of a complete human body from a partial image given. Therefore, we need to prepare such a dataset to tackle the problem. We first explain strategies to prepare the dataset for our study.

3.1. Data preparation: 3×3 Cropping

With an input image I, each human object is represented by its bounding box B and keypoints K like $H = (I, B, K)$. Elements of $B = (x, y, w, h)$ indicate relative 2D position of the top-left corner, width, and height according to I. K is composed by n keypoints $K = \{K_1, ..., K_n\}$, and each K_i consists of 2D pixel coordinates and a visibility status of the keypoint $K_i = (px, py, v)$. Visibility status v can have three values: 2 denotes the keypoint is visible, 1 denotes invisible but can be estimated, and 0 denotes impossible to estimate.

First, we collect $H_{comp} \subset H$ where H_{comp} has human objects which contain their entire body in the image. We

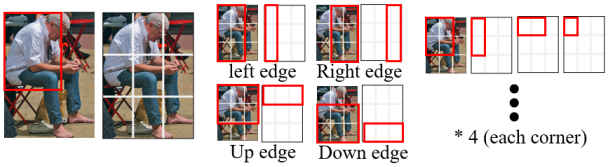


Figure 3. (a): Original image is divided into 3×3 grid. (b): Cropped data are collected from the 3×3 grid by holding one edge of the bounding box. (c): Cropping examples by maintaining one corner of the bounding box.

Figure 4. (a): Example of cropping case b visualized in the red box. (b): Cropped example $I(b)$. (c) The ideal output of PPNet. PPNet estimates 4D vector $\hat{R}$ from $I(b)$, and modifies the image (b) to the image (c). (d): Example of augmented data generated by PP Augmentation, which consists of the partial image and entire keypoints (red circles).

assume the object has an entire human body when v of at least one of keypoints on the head, both wrists, and both ankles are either 1 or 2. Then, sampling bounding box $b \subseteq B$ from H_{comp}, we compose puzzle data P like Eq.1.

$$P = (H_{comp}, b) \tag{1}$$

In the process of sampling b, we employ a fixed policy[22] rather than randomized schemes [27, 16] to stabilize the performance achieved by the cropping policy. Specifically, we first divide B into 3×3 grid and gather cropped samples by holding one edge or one corner at a time, such as in like Fig.3(b) and (c) to simulate cases when only a part of human is in the natural images[16]. Then, samples of b are refined to optimally contain the segmentation region of the target. We exclude the sub-bounding box, which includes a region of the target that is too small (less than 5%) or too large (larger than 95%). Adding the original bounding box itself, we can prepare 25 samples at most from each of H_{comp}. The following sections will introduce Position Puzzle Network and Position Puzzle Augmentation utilizing the puzzle dataset P.

3.2. Position Puzzle Network

Notations and architectures. PPNet f_{PP} is a function to place a given *piece* of an image to solve *position puzzle* for the complete human. For example, if b is sampled according to Fig.4(a), then the input of PPNet becomes a cropped image $I(b)$, as shown in Fig.4(b). The goal of PPNet is to reconstruct the bounding box to place the given piece(i.e., partial image), such as in Fig.4(c). In detail, PPNet regresses 4D vector $\hat{R}$ from image $I(b)$ to reconstruct B using b as shown in Eq. 2.

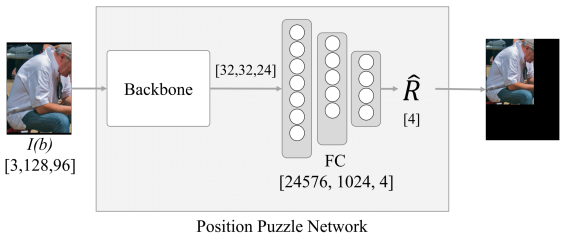


Figure 5. Architecture of PPNet. The output of PPNet is a 4D vector and we utilize the output for reconstructing human images.

$$\begin{aligned} f_{PP}(I(b)) &= \hat{R} \\ &= (r_x, r_y, r_w, r_h) \\ &= (\frac{(x_b - x)}{w}, \frac{(y_b - y)}{h}, \frac{w_b}{w}, \frac{h_b}{h}) \end{aligned} \tag{2}$$

where $B = (x, y, w, h)$ and $b = (x_b, y_b, w_b, h_b)$. All elements of $\hat{R}$ are bounded from 0.0 to 1.0. For a specific example, $f_{pz}(I(B))$ returns $(0.0, 0.0, 1.0, 1.0)$ if B has a complete human body in the image.

An overview of PPNet's architecture is visualized in Fig.5. The backbone of PPNet extracts features of an input image using a series of convolutional layers and poolings. Following fully connected layers take the features and narrow them to estimate 4D vector $\hat{R}$ to reconstruct the bounding box. The specific dimensions in Fig.5 can be changed depending on environmental settings.

Training loss The goal of training PPNet is minimizing loss $\mathcal{L}(R, \hat{R})$ similar to object detection[12, 6, 17, 25]. Existing studies have shown that losses based on intersection over union(IoU) are more effective [24, 18, 26] for training than using regression losses such as ℓ_1 or ℓ_2. Among the IoU based losses, Generalized-IoU(GIoU) loss [18](Eq.3) is employed by a result of preliminary study.

$$\begin{aligned} IoU &= \frac{|B_R \cap B_{\hat{R}}|}{|B_R \cup B_{\hat{R}}|} \\ GIoU &= IoU - \frac{|S/(B_R \cup B_{\hat{R}})|}{|S|} \\ L_{GIoU} &= 1 - GIoU \end{aligned} \tag{3}$$

where B_R and $B_{\hat{R}}$ are reconstructed bounding box using R and $\hat{R}$, respectively. S is the smallest convex box for B_R and $B_{\hat{R}}$.

The accuracy of detecting keypoints can deteriorate when a puzzle becomes too small because the puzzle is the only source to extract useful features. Therefore, we add another loss named *PuzzleSizeLoss*, which provides a penalty

in terms of the relative size of the puzzle (i.e.,$r_w r_h$) in the reconstructed bounding box. Eq.4 denotes PuzzleSizeLoss, and Eq.5 shows the final loss for PPNet.

$$L_{size,\alpha} = C((\alpha - r_w r_h), [0, \alpha]) \tag{4}$$

$$L = L_{GIOU} + L_{size,\alpha} \tag{5}$$

where α is a constant that is used to control the impact of size loss, and function $C(v, [m, M])$ bounds v between m and M.

3.3. Position Puzzle Augmentation for Keypoint Detector

PPNet modifies input images to create a proper margin to contain keypoints outside the bounding box, but keypoint detector is not used to images like Fig.4(c) as its input image. Therefore, we need to provide such training data to maximize the performance of keypoint detector using the output of PPNet.

The original keypoint detector learns patterns between a cropped image by target's bounding box $I(B)$ and keypoints K for all elements in H. Among them, PP Augmentation is only applied to the data belongs to H_{comp} which has the sampled sub-bounding boxes b created by the 3×3 cropping method. PP Augmentation masks the region $B \cap b^c$ of $I(B)$ in black while maintaining K, and the result of the alteration is illustrated by Fig.4(d).

As mentioned in Sec.3.1, one element of H_{comp} has at most 25 cropping cases including the original box. While PPNet treats them as all different training cases, PP Augmentation merges them as a list to handle them as one case of training. In each actual training epoch, we utilize a probability p_{aug} which represents a chance to use augmented data rather than the original data as existing works proposed [27, 22]. For example, when $p_{aug} = 0.3$, the probability that one of the cropping cases is randomly sampled and delivered to the machine as a training data is 30%. Under the policies we mentioned in this section, PP Augmentation maintains the number of training data per an epoch, and consequently, the method does not require additional computational load after pre-processing for arranging puzzle data P.

4. Data and Implementation

Data Among existing datasets [7, 1, 11], we utilize COCO dataset[11], which is one of the most massive and popular datasets. Only COCO has a segmentation region of the target, which is essential to verify whether the sampled sub-bounding box has a part of the target or not. COCO provides three subsets, namely, training set, validation set, and test set, and their scales are approximately 118K, 5K, and 40K, respectively.

Each PPNet and PP Augmentation has its own training dataset and policy, so we do not employ end-to-end training. In the training of PPNet, we choose 39,267 and 1,526 human objects as H_{comp} from the training set and the validation set of COCO data. Then, the policies in Sec.3.1 collect sub-bounding boxes from H_{comp}, and the training set and the validation set of PPNet consist of 596,810 and 23,149 samples in total. For testing PPNet, we prepare three test cases which randomly crop the validation set of COCO while maintaining the cropping policies holding an edge or a corner like in Fig.4. On the other hand, keypoint detector is trained by 118K samples whose scale is identical to the scale of the original training set of COCO, because the cropping cases from the data are merged by a list as mentioned in Sec.3.3.

Position Puzzle Network PPNet utilizes a light version of HR-Net$_{w32}$ [20, 10] as a backbone network to minimize the computational load of employing PPNet. PPNet uses one basic block at each module whereas the original architecture is constructed by four blocks. Using a 128×96 input image, the backbone extracts 32×16×32 dimensional features. The following two fully connected layers whose size are [24576, 1024] and [1024, 4], analyze the features to estimate 4D vector $\hat{R}$. The total GFLOP of the architecture is 0.775. By using Adam optimizer, we utilize four GTX 1080Ti and set 256 batch size for each. Through 160 total training epochs, the learning rate starts from 5e-5, and we divide into half the rate in the 90th, 130th, and 150th epoch. Preliminary experiments (Sec.5.3.3) show that $\alpha = 0.7$ is optimal for PuzzleSizeLoss mentioned in Eq.4.

Data augmentation using random rotation is not applied in the training because we assume that the input and output of PPNet are axis-aligned bounding boxes. Instead, we employ data augmentation that controls the size of the sampled bounding box in training to compensate for the fixed sampling policy mentioned in Sec.3.1, which may not produce enough variant training cases. Specifically, each cropping box changes its width and height up to half of the grid size, and the ground truth R is also changed according to the alterations.

Keypoint Detector and Position Puzzle Augmentation We use HR-Net$_{w32}$ for keypoint detector [20, 10]. Using 256×192 input resolution, we utilize three GTX 1080Ti with 64 batch size for each. The other settings are identical with the settings in [10]. We employ random rotation, horizontal flipping, and random scaling, but exclude half-body augmentation[22]. In addition, all hyper-parameters for training like learning rate, its schedule, and total epochs are maintained to minimize alterations of the baseline.

When we apply PP Augmentation, we modify a data loader to utilize puzzle data P if it is available. Specifically, the data loader decides to pass whether the original data or the augmented data to the machine by the probability p_{aug}

	S_{random}		S_{easy}		$S_{moderate}$		S_{hard}	
	mAP	mAR	mAP	mAR	mAP	mAR	mAP	mAR
HR-Net[20]	21.3	25.8	69.0	73.3	41.2	46.1	11.4	15.8
PPNet$_{\alpha=0.7}$ +HR-Net$_{aug}$	**29.3**	**33.7**	**78.1**	**81.6**	**55.6**	**60.3**	**21.4**	**25.1**

Table 1. Mean averaged precision(mAP) and mean averaged recall(mAR) comparison in Crop-COCO dataset.

	mAP$_{easy}$				mAP$_{moderate}$				mAP$_{hard}$			
Crop direction	left	right	top	bottom	left	right	top	bottom	left	right	top	bottom
HR-Net[20]	80.5	79.0	44.9	71.5	57.5	51.6	7.4	48.3	15.6	10.7	0.2	19.2
PPNet$_{\alpha=0.7}$ + HR-Net$_{aug}$	**83.3**	**82.3**	**63.9**	**83.0**	**68.3**	**64.8**	**23.0**	**66.1**	**29.8**	**24.2**	**0.3**	**31.2**

Table 2. Result of mean averaged precision(mAP) on each Crop-COCO dataset by the cropping directions.

	mAR$_{easy}$				mAR$_{moderate}$				mAR$_{hard}$			
Crop direction	left	right	top	bottom	left	right	top	bottom	left	right	top	bottom
HR-Net[20]	83.2	81.7	52.6	75.5	61.4	56.3	13.7	53.0	21.1	15.7	0.8	25.7
PPNet$_{\alpha=0.7}$ + HR-Net$_{aug}$	**85.9**	**84.8**	**69.6**	**85.9**	**71.6**	**68.7**	**30.7**	**70.3**	**33.8**	**28.4**	**2.2**	**36.0**

Table 3. Result of mean averaged recall(mAR) on each Crop-COCO dataset by the cropping directions.

if the training case has sub-bounding boxes. Preliminary experiments have been performed and we decide $p_{aug} = 0.5$ as the optimal value.

5. Evaluation

We compare the performance of the baseline (i.e., HR-Net[20]) and our proposed method which consists of PP-Net with $\alpha = 0.7$ and HR-Net trained by PP Augmentation (i.e., PPNet$_{\alpha=0.7}$ + HR-Net$_{aug}$). The baseline directly uses a given image and a bounding box to localize keypoints, while our method firstly utilizes PPNet to refine the bounding box and then following HR-Net$_{aug}$ uses the refined image to estimate the position of the keypoints. All images in our experiments are resized according to the input resolution of each neural network, and we mask the regions outside the bounding box in black. We apply a horizontal flip test for all machines.

Sec.5.1 includes experiments to evaluate the performance of localizing keypoints including those outside the bounding box. Given that COCO has no ground-truth for excluded keypoints, we arrange '*Crop-COCO*' dataset by cropping images in COCO and perform various experiments under the Crop-COCO. In Sec.5.2, we also evaluate performance on the original COCO to examine performance changes in the original task which might be affected by our method. Finally, various experimental results that scrutinize the effectiveness of our proposed method are introduced in Sec.5.3.

5.1. Evaluations on Crop-COCO Dataset

The accuracy of the estimations for keypoints outside the bounding box cannot be evaluated under the conventional evaluation method and dataset, because they have no ground truth for such keypoints. Therefore, we construct modified datasets by cropping the validation set of COCO to evaluate the performance of detecting keypoints including those positioned outside the bounding box.

5.1.1 Crop-COCO dataset

We employ the cropping method in Sec.3.1 to construct Crop-COCO dataset. The only difference is that we control the amount of cropping by random. S_{random} is the name of the dataset and we prepare three random cases to alleviate bias. Additionally, we create additional datasets by controlling cropping amount and direction to investigate performance changes of proposed method in particular circumstances. Specifically, we crop 20%, 40%, and 60% of the images from the left, right, top, and bottom edges. For each 20%, 40%, and 60% cases, we name the sets as S_{easy}, $S_{moderate}$, and S_{hard}, respectively. We exclude 80% cropping because localizing entire keypoints using only 20% of images is too excessive. All cropping results are refined so that the cropping box optimally contains the segmentation region of the target object. Meanwhile, we do not modify the ground truth of keypoints to allow a machine to localize entire keypoints from the partial image.

Original box(B): **green**
Cropped box(b): **red**
Reference: val 329456

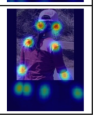
	(a)	(b)	(c)
HR-Net	-		
PPNet +HR-Net$_{aug}$			

Figure 6. Example of 60% crop from bottom in Crop-COCO. The green box and the red box on the reference image are the ground-truth and a cropped box, respectively. (a): Input of PPNet. (b): Input of keypoint detector, and simultaneously the output of PP-Net. (c): Estimated heatmap by using (b).

5.1.2 Quantitative Evaluation

Table 1 compares the mean averaged precision(mAP) and the mean averaged recall(mAR) on S_{random}, S_{easy}, $S_{moderate}$, and S_{hard}. For S_{random}, AP and AR from three random sets are averaged. The proposed method enhances the performance of keypoint detection for all prepared tests. In S_{random}, the proposed method outperforms the baseline in mAP and mAR by 37.6% and 30.6% respectively. The differences are noticeable under the difficult set. Specifically, the proposed method increases mAP and mAR by approximately 13.2% and 11.3%, 35.0% and 30.8%, and 87.7% and 58.9% on average under the S_{easy}, $S_{moderate}$, and S_{hard} respectively.

Table 2 and Table 3 show the details of mAP and mAR according to the cropping directions. Similarly, the proposed method outperforms in all cases, and mAP and mAR are leveraged by 27.5% and 23.5% on average. When we examine performance changes according to cropping direction (i.e., left, right, top, and bottom), mAP enhancements of each case are 18.1%, 21.2%, 66.1%, and 29.7% on average, and mAR improvements are 15.4%, 18.3%, 52.8%, and 24.6% on average. The top-crop case is relatively difficult, because a large number of keypoints are positioned at the top of the image such as nose, eyes, ears, and shoulders.

5.1.3 Qualitative Evaluation

Fig.6 depicts estimation results using the Crop-COCO data. The reference image in Fig.6 illustrates the original bounding box in green, and 60% cropping box from the bottom in red. Column (b) shows that, without PPNet, the image in the red box directly becomes the input of a keypoint detector as it is shown in the 1st row. Consequently, the keypoint detector is not able to localize keypoints outside the box such as both hands and both hips of the target. By comparison, PPNet refines the bounding box to allow the modified bounding box to include more margins below as the 2nd row at column (b) and to provide an opportunity to keypoint detector so that both hands and both hips are localized by the detector (column(c)).

5.2. Evaluation on the Original COCO Dataset

Although experimental results in Sec.5.1 show the effectiveness of our method to localize keypoints beyond the bounding box, we need to investigate the performance changes caused by proposed method on the original dataset and task. In this section, we evaluate our method using the conventional evaluation of keypoint detection on the original COCO dataset.

The table 4 shows the results from the validation set of COCO(*COCO-val*) and the test set of COCO(*COCO-test*). For COCO-val, using human detection results whose AP is 56.4[20], our proposed method slightly enhances the performance of keypoint detection in the validation set of COCO, and mAP and mAR are improved by 0.6% and 0.5%, respectively. For the test set of COCO, each environment localizes human keypoints on the basis of the detection result whose AP is 60.9[20]. The proposed method slightly outperforms the baseline by 0.8% and 0.9% for mAP and mAR, respectively. The results from the experiments indicate that our proposed method improves the performance of detecting keypoints in the image although the method aims to enhance the performance of localizing keypoints including those positioned outside the bounding box.

The visualization of the estimation results from both environments verifies the effectiveness of the proposed method. Each row in Fig.7 represents the results of the baseline and our proposed method. The results verify that PPNet properly reconstructs the input image to contain keypoints outside the bounding box, and the refinement allows the excluded keypoints to be possibly estimated by the following keypoint detector. For example, as the 1st and 4th columns depict, PPNet understands the context of missing body parts from the partial image and enlarges the given bounding box to bottom-side and top-side, respectively. Although slight differences are achieved by PPNet in the 2nd and 3rd columns, we can examine the advantage of PP Augmentation from those columns. The baseline has enough margin to localize excluded keypoints of the target, but keypoint detector rarely localizes those keypoints. On the other hand, keypoint detector trained by PP Augmentation estimates a position of the excluded keypoints using partial

		mAP	$AP_{.5}$	$AP_{.75}$	AP_m	AP_l	mAR	$AR_{.5}$	$AR_{.75}$	AR_m	AR_l
COCO-val$_{AP=56.4}$	HR-Net	72.5	88.6	78.8	68.6	**80.2**	78.4	93.0	84.2	73.6	**85.3**
	PPNet$_{\alpha=0.7}$ +HR-Net$_{aug}$	**73.0**	**89.1**	**80.0**	**69.8**	79.6	**78.8**	**93.4**	**85.2**	**74.6**	84.8
COCO-test$_{AP=60.9}$	HR-Net	71.8	91.0	78.8	68.0	**78.3**	77.4	94.5	83.9	73.0	83.6
	PPNet$_{\alpha=0.7}$ +HR-Net$_{aug}$	**72.4**	**91.3**	**80.0**	**69.3**	78.2	**78.1**	**94.9**	**85.0**	**74.0**	**83.7**

Table 4. Performance on the original COCO's validation set(COCO-val) and test set(COCO-test). We use the human detection results whose AP is 56.4 and 60.9, respectively.

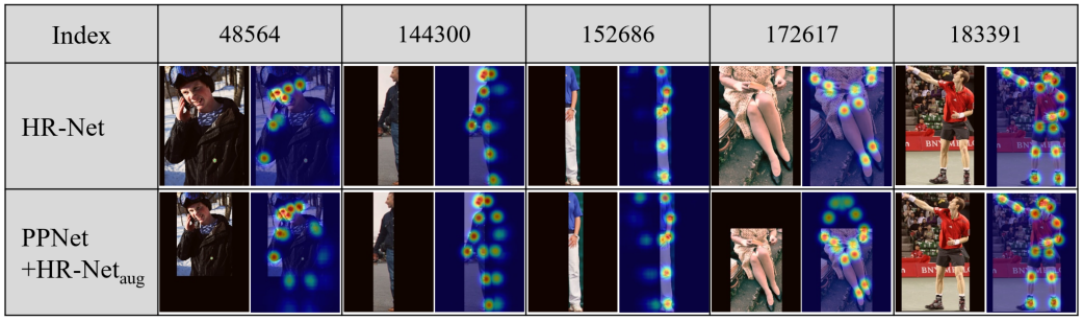


Figure 7. Qualitative evaluation on the original validation set of COCO dataset. The first row shows the result of the baseline and the second row describes the result of our proposed method. In each cell, images on the left side are input images of the keypoint detector, and the other images are converged probability maps estimated by keypoint detector. The maps are visualized by Jet colormap.

image because the machine already learns the patterns between partial image and the complete set of keypoints. The 5th column shows an example if the bounding box has the entire human. PPNet maintains the bounding box because the training set of PPNet has not only cropping cases but also the data itself to learn the cases when the bounding box has the complete human.

5.3. In-depth study

In this section, we introduce further experimental results to investigate the effect of the proposed method. All experiments in this section utilize the ground truth bounding boxes from COCO and Crop-COCO to eliminate the changes due to the accuracy of the bounding box. Sec.5.3.1 will mention experimental results to separately investigate performance changes on keypoints in the box and outside the box. Sec.5.3.2 will introduce the ablation study of PPNet and PP Augmentation. Lastly, Sec.5.3.3 describes the performance analysis by controlling α to attain the optimal size of the impact of PuzzleSizeLoss.

5.3.1 Performance by the location of keypoints

We separately evaluate the performance of our proposed method by preparing two subsets for evaluation. One is Crop-COCO$_{in}$, which has keypoints only in the bounding box, and the other is Crop-COCO$_{out}$ which is constructed by only keypoints outside the bounding box.

	Crop-COCO$_{in}$		Crop-COCO$_{out}$	
	mAP	mAR	mAP	mAR
HR-Net	81.5	85.2	10.2	16.2
PPNet$_{\alpha=0.7}$ +HR-Net$_{aug}$	**84.1**	**87.4**	**24.1**	**32.0**

Table 5. Performance comparison by separating keypoints of Crop-COCO dataset into two subsets. Crop-COCO$_{in}$ consists of keypoints in the bounding box, whereas Crop-COCO$_{out}$ is constructed by keypoints outside the bounding box.

Table 5 shows the results of the experiment. Although the experiment on Crop-COCO$_{in}$ focuses on keypoints in the bounding box, the proposed method slightly enhances mAP and mAR of keypoint detection up to 3.2% and 2.6% respectively. Considering PPNet inevitably produces lower resolution images to the keypoint detector, PP Augmentation effectively compensates for the weakness by training keypoint detector with partial images. When we compare the detecting performance for keypoints outside the bounding box, the difference becomes considerable. PPNet and PP Augmentation leverage mAP and mAR for Crop-COCO$_{out}$ up to 136.3% and 97.5%, respectively. The results verify that PPNet, which brings excluded keypoints into the estimation range of the keypoint detector, and PP Augmentation, which trains the keypoint detector to localize keypoints outside the box using the partial image, forms a synergy to increase the performance of the keypoint detection under the Crop-COCO dataset.

	mAP	mAR	missing kpts(%)
HR-Net[20]	40.5	45.1	24.0
HR-Net$_{aug}$	42.0	46.7	24.0
PPNet$_{\alpha=0.7}$ +HR-Net	45.2	49.6	**14.1**
PPNet$_{\alpha=0.7}$ +HR-Net$_{aug}$	**51.7**	**55.7**	**14.1**

Table 6. Ablation study using PPNet and PP Augmentation on Crop-COCO dataset. '*missing kpts*' denotes an averaged percentage of the keypoints that are excluded from the region of interest of keypoint detector.

5.3.2 Ablation Study for PPNet and PP Augmentation

We perform an ablation study to examine how much PPNet and PP Augmentation contribute to the enhancements. We define four cases from combinations of PPNet and PP Augmentation and evaluate them on Crop-COCO dataset.

Table 6 shows changes in mAP, mAR, and the number of missing keypoints that are outside the estimation range of the keypoint detector. Specifically, the 1st and 2nd rows show that PP Augmentation can lead to performance enhancement of detecting keypoints from a partial image by 3.7% and 3.5% on average for mAP and mAR, respectively. The increase is attributed to training the patterns between the partial image and the entire keypoints, which is the contribution of PP Augmentation. Examining the effect of PPNet as shown in the 1st and 3rd rows, PPNet improves the performance without PP Augmentation by 11.6% and 10.0% on average for mAP and mAR, respectively. The improvement is attributed to the decrease in missing keypoints, as the 3rd column of the table describes. 58.8% of the missing keypoints are back in the modified bounding box of PPNet and have the opportunity to be localized by the keypoint detector. Lastly, the 4th row shows that we achieve the prominent enhancement by 27.7% and 23.5% on average for mAP and mAR, respectively, when we employ both PPNet and PP Augmentation.

5.3.3 Deicidng the optimal impact of PuzzleSizeLoss

We have introduced PuzzleSizeLoss(Eq.4) which prevents PPNet from making a piece of the image too small, and α of PuzzleSizeLoss controls the size of impact in training PPNet. Table 7 denotes performance changes on Crop-COCO dataset and the original COCO dataset in various α. The results of the Crop-COCO dataset indicate a larger α generally deteriorates the performance because PPNet cannot sufficiently refine a bounding box by a penalty generated from PuzzleSizeLoss. However, in the case of the original COCO dataset, a large α improves performance in detecting keypoints on the standard metric of COCO. When we attempt to decide the optimal α, we consider that keypoint detector cannot localize all excluded keypoints although PPNet perfectly refines the bounding box. For example, with an image containing a target's head and shoulders, we cannot accurately estimate where the target's ankles are located because the image does not have enough clues to estimate their position. Hence, we decide $\alpha = 0.7$ is the optimal value which minimizes the performance damage in the original COCO and simultaneously maintains the advantage in Crop-COCO dataset.

	Crop-COCO		COCO$_{val}$	
	mAP	mAR	mAP	mAR
HR-Net	40.5	45.1	75.2	78.0
HR-Net$_{aug}$	42.0	46.7	75.9	78.8
PPNet$_{\alpha=0.0}$ +HR-Net$_{aug}$	56.0	60.9	71.8	74.7
PPNet$_{\alpha=0.3}$ +HR-Net$_{aug}$	**56.5**	**61.4**	72.3	75.3
PPNet$_{\alpha=0.5}$ +HR-Net$_{aug}$	56.1	60.8	73.7	76.9
PPNet$_{\alpha=0.7}$ +HR-Net$_{aug}$	51.7	55.7	75.2	78.2
PPNet$_{\alpha=0.9}$ +HR-Net$_{aug}$	46.1	50.6	**76.0**	**78.9**

Table 7. Performance changes according to various α based on Crop-COCO dataset and the validation set of the original COCO data. The first and second rows are references which do not apply PPNet.

6. Conclusion

We have proposed Position Puzzle Network(PPNet) and Position Puzzle Augmentation(PP Augmentation) to effectively localize human keypoints beyond the bounding box. We verify that PPNet refines the given bounding box to contain excluded keypoints of a target, and a keypoint detector trained by PP Augmentation accurately estimates the position of keypoints that are not only within the image but also outside the image. The synergy of PPNet and PP Augmentation significantly leverages mAP and mAR of localizing keypoints outside the bounding box up to 136.3% and 97.5%, respectively, in Crop-COCO dataset. Furthermore, the proposed method outperforms the baseline in the existing COCO test-dataset by 0.8% and 0.9% for mAP and mAR, respectively. The various experimental results and visualizations also show the proposed method reasonably localizes those keypoints outside the image although there is no ground truth position within the dataset for them.

Acknowledgments

This research is supported by Ministry of Culture, Sports and Tourism and Korea Creative Content Agency (Project Number: R2020070002)

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3686–3693, 2014.

[2] Ankan Bansal, Sai Saketh Rambhatla, Abhinav Shrivastava, and Rama Chellappa. Detecting human-object interactions via functional generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10460–10469, 2020.

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7291–7299, 2017.

[4] Yu Cheng, Bo Yang, Bo Wang, Wending Yan, and Robby T Tan. Occlusion-aware networks for 3d human pose estimation in video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 723–732, 2019.

[5] Chen Gao, Yuliang Zou, and Jia-Bin Huang. ican: Instance-centric attention network for human-object interaction detection. In *British Machine Vision Conference*, 2018.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988. IEEE, 2017.

[7] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[8] Lipeng Ke, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-scale structure-aware network for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 713–728, 2018.

[9] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 417–433, 2018.

[10] leoxiaobin. deep-high-resolution-net.pytorch. https://github.com/leoxiaobin/deep-high-resolution-net.pytorch, 2019.

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[12] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016.

[13] Jiaxu Miao, Yu Wu, Ping Liu, Yuhang Ding, and Yi Yang. Pose-guided feature alignment for occluded person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 542–551, 2019.

[14] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 483–499. Springer, 2016.

[15] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–286, 2018.

[16] Soonchan Park, Sang-baek Lee, and Jinah Park. Data augmentation method for improving the accuracy of human pose estimation with cropped images. *Pattern Recognition Letters*, 136:244–250, 2020.

[17] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525. IEEE, 2017.

[18] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019.

[19] Chi Su, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Pose-driven deep convolutional model for person re-identification. In *Proceedings of the IEEE international conference on computer vision*, pages 3960–3969, 2017.

[20] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. *arXiv preprint arXiv:1902.09212*, 2019.

[21] Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, Xiangyu Zhang, and Jian Sun. Learning human-object interaction detection using interaction points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4116–4125, 2020.

[22] Zhig Wang, Wenbo Li, Binyi Yin, Qixiang Peng, Tianzi Xiao, Yuming Du, Zeming Li, Xiangyu Zhang, Gang Yu, and Jian Sun. Mscoco keypoints challenge 2018. *Joint Recognition Challenge Workshop at ECCV*, 2018.

[23] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 466–481, 2018.

[24] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520, 2016.

[25] Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, and Haibin Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. 2019.

[26] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *AAAI*, pages 12993–13000, 2020.

[27] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, pages 13001–13008, 2020.

[28] Tianfei Zhou, Wenguan Wang, Siyuan Qi, Haibin Ling, and Jianbing Shen. Cascaded human-object interaction recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4263–4272, 2020.