The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)

To Choose or to Fuse? Scale Selection for Crowd Counting

Qingyu Song[1,2] *, **Changan Wang**[1] *, **Yabiao Wang**[1],
Ying Tai[1], **Chengjie Wang**[1], **Jilin Li**[1], **Jian Wu**[2] †, **Jiayi Ma**[3]
[1]Tencent Youtu Lab, Shanghai, China
[2]College of Computer Science & Technology, Zhejiang University, Hangzhou, China
[3]Electronic Information School, Wuhan University, Wuhan, China
{changanwang, caseywang, yingtai, jasoncjwang, jerolinli}@tencent.com,
{qingyusong, wujian2000}@zju.edu.cn, jyma2010@gmail.com

Abstract

In this paper, we address the large scale variation problem in crowd counting by taking full advantage of the multi-scale feature representations in a multi-level network. We implement such an idea by keeping the counting error of a patch as small as possible with a proper feature level selection strategy, since a specific feature level tends to perform better for a certain range of scales. However, without scale annotations, it is sub-optimal and error-prone to manually assign the predictions for heads of different scales to specific feature levels. Therefore, we propose a Scale-Adaptive Selection Network (SASNet), which automatically learns the internal correspondence between the scales and the feature levels. Instead of directly using the predictions from the most appropriate feature level as the final estimation, our SASNet also considers the predictions from other feature levels via weighted average, which helps to mitigate the gap between discrete feature levels and continuous scale variation. Since the heads in a local patch share roughly a same scale, we conduct the adaptive selection strategy in a patch-wise style. However, pixels within a patch contribute different counting errors due to the various difficulty degrees of learning. Thus, we further propose a Pyramid Region Awareness Loss (PRA Loss) to recursively select the most hard sub-regions within a patch until reaching the pixel level. With awareness of whether the parent patch is over-estimated or under-estimated, the fine-grained optimization with the PRA Loss for these region-aware hard pixels helps to alleviate the inconsistency problem between training target and evaluation metric. The state-of-the-art results on four datasets demonstrate the superiority of our approach. The code will be available at: https://github.com/TencentYoutuResearch/CrowdCounting-SASNet.

Introduction

Crowd counting aims to estimate the number of people within given images or videos. Most recent state-of-the-art works (Zhang et al. 2016; Li, Zhang, and Chen 2018) adopt deep learning based methods and transform original annotations, *i.e.*, center points of people's heads, into density maps as training targets. However, a major challenge for the task is the extremely large scale variation of crowds, which becomes even more tricky due to the lack of scale annotations.

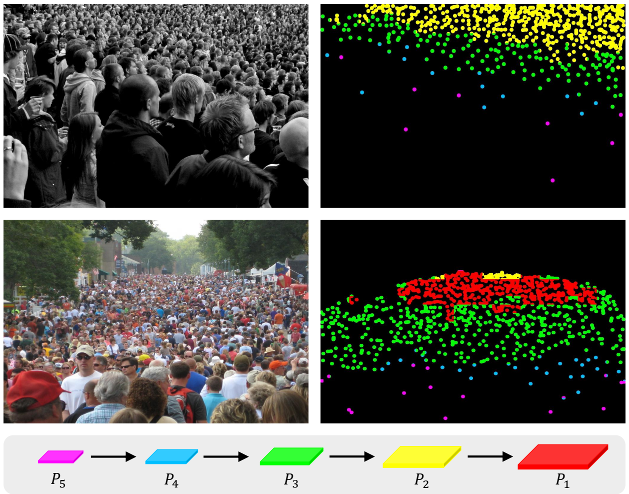


Figure 1: Original input images (left column) and visualizations of our adaptive selection results (right column). In the visualizations, we use points with different colors to represent the persons' heads. The colors represent the most appropriate feature levels selected by our SASNet for these points. There are totally five feature levels $\{P_1, ..., P_5\}$ in SASNet, and the resolution of which increases from P_5 to P_1. The rich detail information in high-resolution feature levels is helpful to the prediction of small scale heads, while the low-resolution feature levels with rich contextual information perform better for large scale heads. However, it is non-trivial to manually assign the correspondence relations since the irregular partition boundaries and the inhomogeneous crowd distribution. Instead, the proposed adaptive selection strategy automatically learns the internal relations and the visualizations demonstrate its effectiveness.

In recent years, numerous methods have been proposed to tackle with the large scale variation problem of crowd counting. One feasible workaround is to use the multi-column networks (Zhang et al. 2016; Onoro-Rubio and López-Sastre 2016; Sam, Surya, and Babu 2017), which aggregate several branches with various receptive fields to acquire features

*Equal contribution. †Corresponding author.

with rich scale information. However, these methods are unable to cover the continuous scale variation, which lead to feature redundancy among different branches. The other major approaches adopt multi-level networks (Zhang, Shi, and Chen 2018; Valloli and Mehta 2019), in which the final prediction is generated by the exploitation of all the feature levels. But these methods aggregate the features from multiple levels in a scale-agnostic way, which ignore the intrinsic correspondence between feature levels and head scales. Generally speaking, although these two kinds of methods benefit from the multi-scale feature representations, both of them fail to effectively exploit these rich scale information. To remedy the deficiencies of existing methods, one possible solution is to select the most appropriate feature level for heads of different scales. However, it is non-trivial to conduct such selection due to the continuous scale variation and the lack of scale annotations. Towards the issue of insufficient annotations, some previous methods (Li, Zhang, and Chen 2018; Zhang et al. 2016) tried to approximately estimate the heads' scales, which are not actually reliable especially for sparse areas. Besides, these methods still need additional rules to select the appropriate feature levels for specific scales, which highly depend on hand-craft design.

In this work, we propose a novel multi-level based Scale Adaptive Selection Network (SASNet) to take full advantage of the multi-scale feature representations. The SASNet adopts an adaptive selection strategy to automatically learn the internal level-scale correspondence, without using extra scale annotations or scale estimation strategies. Specifically, for a given image patch, our SASNet predicts a score for each feature level to indicate its confidence of being the most appropriate level. Then we normalize these scores as weighting coefficients, and use them to re-weight the predicted density maps from all feature levels. Finally, the final prediction for the image patch is obtained by summing the re-weighted predictions, which helps to reduce the inconsistency between discrete feature levels and continuous scale variation. As demonstrated in Figure 1, the proposed adaptive selection strategy reasonably learns the internal correspondence between feature levels and head scales.

The proposed strategy of adaptive selection enables the patch-wise feature level selection for heads in a patch, but ignores the different difficulty degrees of learning for pixels within the patch. On the contrary, the learning of each feature level is guided by pixel-wise regression losses, which is obviously region-ignorant. To further optimize the pixels within a patch in a fine-grained way, we propose a novel Pyramid Region Awareness (PRA) Loss. The PRA Loss recursively searches the most over-estimated (or under-estimated) sub-regions in an over-estimated (or under-estimated) patch until reaching the pixel level. After the searching process, these region-aware pixels are selected as the most hard samples to be further optimized. Besides, the widely adopted training loss emphasizes the accurate pixel-wise regression, while the evaluation metric only focuses on the crowd number in a region. As a result, after taking the inclusion relations between pixels and regions into consideration, our PRA Loss also helps to reduce the inconsistency between training target and evaluation metric. In conclusion, the contributions of this work are summarized as follows:
1. We propose a patch-wise feature level selection strategy to identify the most appropriate feature level. Such a novel strategy effectively exploits the multi-scale feature representations inside a multi-level network, offering a new way for addressing the challenging large scale variation problem.
2. We propose a novel PRA Loss to recursively select the most hard pixels that are further optimized in a region-aware style, acting as a fine-grained complement for the patch-wise feature level selection strategy.
3. We conduct extensive experiments on four datasets to demonstrate the superiority of our method against the state-of-the-art competitors.

Related Works

In this section, according to the way of acquiring multi-scale representations, we group Convolutional Neural Networks (CNN) based crowd counting methods into two categories, which are highly related to our proposed method.

Multi-Column based Approaches. This kind of methods adopts multi-column architecture or stacked multi-branch module to obtain rich multi-scale feature representations. MCNN (Zhang et al. 2016) builds a three-branch network with different convolution kernel sizes to acquire features with different receptive fields. Instead of using different kernel sizes, CrowdNet (Boominathan, Kruthiventi, and Babu 2016) designs different convolution depths for each branch, and then combines the low-level features in shallow branch and the high-level features in deep branch together. Unlike above methods, DADNet (Guo et al. 2019) uses different dilated rates in each parallel column to obtain multi-scale features. Similar to MCNN, Switch CNN (Sam, Surya, and Babu 2017) also utilizes a three-branch architecture but adds an additional classifier to select the branch for prediction. RANet (Zhang et al. 2019a) proposes a local self-attention module and a global self-attention module to simultaneously obtain the local and global features. In summary, these methods try to address the scale variation problem by combining the scale-dependent features from multiple branches, but also introduce significant feature redundancy.

Multi-Level based Approaches. These approaches learn the multi-scale representations from multiple internal layers of the backbone network, which exploit the hierarchical structure of CNN and are obviously more efficient compared with the multi-column based methods. SaCNN (Zhang, Shi, and Chen 2018) is a single-column network, but fuses feature maps from different feature levels to obtain the multi-scale information. AFN (Zhang et al. 2019b) utilizes Conditional Random Fields (CRF) to aggregate multi-scale features from different levels within an encoder-decoder network. Similarly, DSSINet (Liu et al. 2019) also adopts a CRF-based module, but different from AFN, the features of each level are extracted from corresponding image in the input pyramid using the same network. Another method (Varior et al. 2019) tries to explore the multi-scale feature fusion by soft attention mechanism, which aggregates the predicted

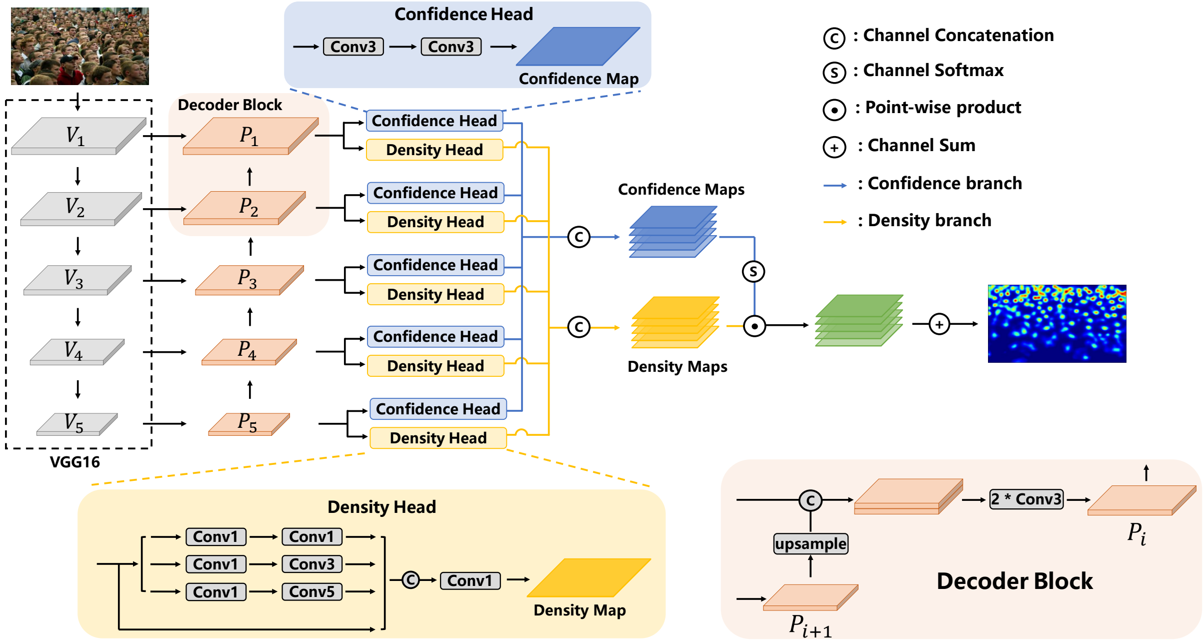


Figure 2: The overall architecture of the proposed SASNet mainly consists of three components: U-shape backbone, confidence branch and density branch. Firstly, the U-shape backbone is adopted to extract multi-level feature representations for a given image. Then these features are fed into both confidence heads and density heads to obtain multi-level confidence maps and density maps respectively. Finally, with the guidance from the multi-level confidence maps, we combine the multi-level density maps via weighted average to obtain the final prediction. "Conv$n$" represents a convolution with kernel size of $n \times n$.

density maps from multi-level features into the final prediction. TedNet (Jiang et al. 2019) incorporates multiple decoders with dense skip connections to hierarchically aggregate the multi-level features. Multi-level networks are designed to exploit the natural hierarchical structure of CNN to extract multi-scale feature representations. This kind of methods has been demonstrated efficient and effective but still fails to explicitly build the correspondence relations between feature levels and head scales.

Our Approach

As illustrated in Figure 2, there are two parallel branches: density branch and confidence branch. The density branch predicts density maps using five feature levels, while the confidence branch is responsible for predicting confidence scores to indicate the most appropriate feature level for a certain image patch. After the final predicted density map is obtained using an adaptive selection strategy, we further select region-aware hard pixels in the density map by PRA Loss and optimize them in a fine-grained way.

Scale-Adaptive Selection

As shown in Figure 2, the multi-scale feature representations are denoted to $\{P_1, P_2, P_3, P_4, P_5\}$, which are generated hierarchically using the features of different levels in the backbone network. However, due to the limited receptive field for specific P_i, it is only suitable for predicting heads within a narrow range of scales. To take full advantage of the multi-scale feature representations, we firstly make predictions independently using P_i in a scale-agnostic way with the density branch. Then we obtain final prediction by selecting the prediction from the most appropriate feature level, which is assisted by the confidence branch. With the observation that heads in a specific patch share roughly a same scale, such a selection strategy is conducted in a patch-wise style. And the selected feature level for each patch is considered be able to achieve lower counting error for corresponding prediction.

Density Branch. As illustrated in Figure 2, the density branch consists of five density heads which are responsible for the predictions using $\{P_1, P_2, P_3, P_4, P_5\}$ respectively. In order to obtain high-quality density maps in each density head, we aggregate context information across multiple fine-grained scales using a multi-branch module. The module consists of three convolution branches and one skip connection branch. For the convolution branches, we firstly use a 1×1 convolution for channel reduction, and then adopt convolutions with different kernel sizes to acquire context information from various receptive fields. The output features from the multi-branch module are concatenated along channel dimension, after which we use a 1×1 convolution to obtain the prediction for the i-th feature level. Finally, we upsample the predicted density map of each feature level to D_i, which has the same size as original input.

These density heads are supervised with the same ground-truth density map D^{gt}. And we use the Euclidean distance between the predicted density map D_i and D^{gt} as the loss function. Then, the losses from different levels are summed

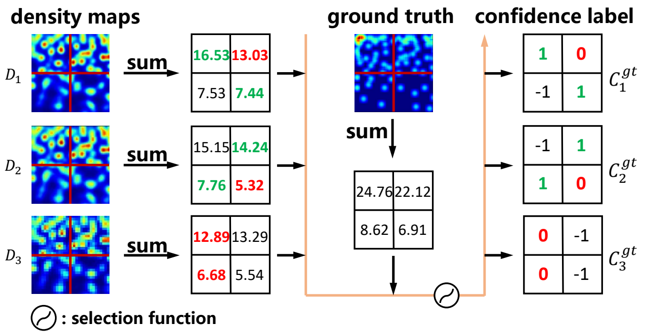


Figure 3: The ground-truth label generation for confidence branch. For a specific patch, the selection function assigns positive label of 1 to the feature level with the closest estimated crowd number to the ground truth, and negative label of 0 to the feature level with the max prediction error. The other feature levels are assigned with label of -1 which are ignored during training. The green and red colors represent positive and negative samples respectively.

as density branch loss $\mathcal{L}_{den}$, which is defined as follows:

$$\mathcal{L}_{den} = \sum_{i}^{5} \left|\left|D_i - D^{gt}\right|\right|_2^2. \tag{1}$$

Confidence Branch. After the predicted density maps $\{D_1, D_2, D_3, D_4, D_5\}$ are acquired, we try to select the prediction with minimum counting error for a specific patch to take full advantage of the multi-scale feature representations. However, since we do not have access to the ground truth during inference, it is infeasible to determine the most appropriate feature level with minimum counting error. Therefore, we propose a confidence branch to assist the feature level selection, which contains five confidence heads, one for each feature level. The confidence head predicts a score to indicate its confidence of being the most appropriate feature level for a certain patch.

We denote the confidence maps as $\{C_1, C_2, C_3, C_4, C_5\}$, which are generated with the confidence heads using features $\{P_1, P_2, P_3, P_4, P_5\}$. Specifically, since we conduct a patch-wise selection, we first down-sample P_i to $1/k$ size of original input image. Then the downsampled features are processed with two 3×3 convolutions, which are then modulated by a Sigmoid function to obtain C_i. The score value in the confidence map C_i represents the confidence that the i-th feature level is the most appropriate level to make the prediction for a specific $k \times k$ patch. The ground-truth label for the confidence map C_i is generated by the comparison of the predicted density map D_i and the ground truth D^{gt}. We illustrate the process in Figure 3, in which we take three feature levels out of five as an example for simplicity. Firstly, we divide each predicted density map D_i into several patches of size $k \times k$ without overlapping. Then, we obtain a counting map M_i to represent the crowd numbers of all patches, and the crowd number is the summation of the density values inside each patch. Similarly, we have the counting map M^{gt} for the ground truth D^{gt}. Finally, the ground-truth label of the confidence map C_i is generated by:

$$C_{i,m,n}^{gt} = \begin{cases} 1, & \text{if } \underset{l\in[1,5]}{\arg\min} \left|M_{l,m,n} - M_{m,n}^{gt}\right| = i, \\ 0, & \text{if } \underset{l\in[1,5]}{\arg\max} \left|M_{l,m,n} - M_{m,n}^{gt}\right| = i, \\ -1, & \text{otherwise}, \end{cases} \tag{2}$$

where $C_{i,m,n}^{gt}$ represents the ground-truth label on the i-th feature level for the patch located in (m, n), $M_{l,m,n}$ represents the predict crowd number on the l-th feature level, and $M_{m,n}^{gt}$ represents the ground-truth crowd number for the patch. As a result, the positive label for $C_{i,m,n}^{gt}$ indicates that the i-th feature level is the most appropriate level for the prediction of the patch located in (m, n). While the negative label indicates that the i-th feature level is the most inappropriate feature level. The patches with label of -1 are ignored in the training phase.

The confidence heads are supervised with Binary Cross Entropy (BCE) loss:

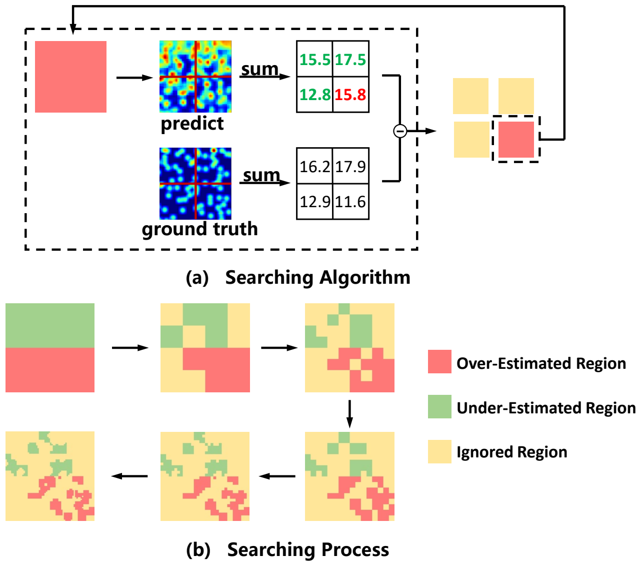
$$\begin{aligned} \mathcal{L}_{ce}(C_{i,m,n}, C_{i,m,n}^{gt}) = C_{i,m,n}^{gt} \cdot \log(C_{i,m,n}) \\ +(1 - C_{i,m,n}^{gt}) \cdot (1 - \log(C_{i,m,n})), \end{aligned} \tag{3}$$

$$\mathcal{L}_{conf} = \frac{\sum_{i=1}^{5} \sum_{(m,n)\in\mathcal{K}_i} \mathcal{L}_{ce}(C_{i,m,n}, C_{i,m,n}^{gt})}{\sum_{i=1}^{5} |\mathcal{K}_i|}, \tag{4}$$

where $\mathcal{K}_i$ represents the set of patches with confidence label of 0 or 1 in the i-th level, $|\mathcal{K}_i|$ represents the number of patches in $\mathcal{K}_i$, $\mathcal{L}_{ce}$ denotes the BCE loss and $\mathcal{L}_{conf}$ denotes the total loss for the confidence branch.

Selection Strategy. During training, the most appropriate feature level is considered to be the feature level with minimum counting error. While during inference, with the assistance from the confidence branch, the most appropriate feature level is indicated by the confidence scores. For a patch located in (m, n), we select the feature level j with the max confidence score as the most appropriate feature level, *i.e.*, $j = \arg\max_{i\in[1,5]} C_{i,m,n}$. However, the scale distribution of heads is continuous, while the multi-scale features in the multi-level network are discrete. As a result, during inference, it is sub-optimal to directly use the corresponding region on the density map of the j-th feature level as the prediction for the patch in (m, n). Instead, we also consider the predictions from the other feature levels via weighted average, which helps to mitigate the gap between discrete feature levels and continuous scale variation. Specifically, as illustrated in Figure 2, we firstly aggregate the confidence maps $\{C_1, C_2, C_3, C_4, C_5\}$ by concatenation and normalize them across the five feature levels with a Softmax function. Then the normalized confidence map of C_i is resized using the nearest interpolation to C_i', which has the same size as D_i. Finally, we use C_i' as the weighting coefficients to apply the weighted average on the predicted density maps $\{D_1, D_2, D_3, D_4, D_5\}$. Thus for a pixel (j, k) in the final predicted density map, we have:

$$D_{j,k}^{est} = \sum_{i}^{5} (C_{i,j,k}' \cdot D_{i,j,k}), \tag{5}$$



(a) Searching Algorithm

(b) Searching Process

Figure 4: (a) Sub-region searching algorithm in the PRA Loss. (b) Overview of the recursive searching process. As illustrated in (a), we firstly divide an input region into four sub-regions and determine whether each sub-region is over-estimated or under-estimated. Then as illustrated in (b), for the whole image, we apply the searching algorithm recursively until the most hard pixels are obtained.

where $D^{est}_{j,k}$ represents the density value for pixel (j, k) in the final estimated density map.

The Pyramid Region Awareness Loss

The PRA Loss is designed to conduct a fine-grained optimization for the most hard pixels within a patch, which is motivated by the following observation. The proposed adaptive selection strategy selects the most appropriate feature level for final prediction in a patch-wise style, which is obviously a coarse selection. However, the pixels within a patch have various difficulty degrees of learning and contribute different counting errors. In particular, the over-estimated pixels, whose estimated crowd number is larger than the ground truth, in an over-estimated patch contribute the max counting error, and vice versa. Therefore, if we pay more attention to the optimization for these most hard pixels, the overall counting error might be further reduced. To be more specific, taking the over-estimated regions as an example, if we equally divide an over-estimated region into four smaller sub-regions, there must be at least one over-estimated sub-region. Since these over-estimated sub-regions make major contributions to the overall over-estimation problem, they should be further optimized. On the contrary, the under-estimated sub-regions are helpful to alleviate the over-estimation problem of the parent region to some extent, thus are ignored in the PRA Loss and just optimized as normal regions.

We begin the introduction for the PRA Loss with the search process of the most hard pixels. As illustrated in Figure 4 (a), we conduct a recursive search process from the region level (the whole image) until the pixel level. For a given over-estimated region, we firstly divide it into four sub-regions and calculate the crowd number inside these sub-regions. Then, by comparing with corresponding ground truth, we can select the over-estimated sub-regions from them. These selected sub-regions are fed into the searching algorithm again, and this process is repeated until reaching the pixel level. The final selected pixels are considered as the most hard pixels in the whole image, which are further optimized with the PRA Loss. The visualization of the searching process is shown in Figure 4 (b).

The selected most hard pixels are further optimized by a weighted Euclidean distance loss. Thus the final PRA Loss is defined as follows:

$$\mathcal{L}_{pra} = \left|\left| D^{est}_{p\in\mathcal{G}} - D^{gt}_{p\in\mathcal{G}} \right|\right|_2^2 + \gamma \left|\left| D^{est}_{p\in\mathcal{H}} - D^{gt}_{p\in\mathcal{H}} \right|\right|_2^2 \tag{6}$$

where p denotes the pixel in the final predicted density map D^{est}, $\mathcal{G}$ and $\mathcal{H}$ represent the set of all pixels in D^{est} and the set of the most hard pixels respectively, and γ is a weight term for the hard pixels.

It is worth mentioning that the PRA Loss also helps to reduce the inconsistency between training target and evaluation metric. During training, the widely adopted training loss emphasizes the accurate pixel-wise regression, while the evaluation metric only focuses on the crowd number in a region. Therefore, the converged model with minimum training loss cannot ensure the optimal counting accuracy when testing. Since the most hard pixels are selected with awareness of whether the parent region is over-estimated or under-estimated. After taking such inclusion relations between pixels and regions into consideration, our PRA Loss helps to reduce the above inconsistency.

Model Learning

The final training loss $\mathcal{L}_{final}$ is the summation of the above three losses, *i.e.*, $\mathcal{L}_{den}$, $\mathcal{L}_{conf}$ and $\mathcal{L}_{pra}$. To be more formal, $\mathcal{L}_{final}$ is defined as follows:

$$\mathcal{L}_{final} = \mathcal{L}_{den} + \lambda\mathcal{L}_{conf} + \mathcal{L}_{pra} \tag{7}$$

where λ is the weight term to balance the effect of $\mathcal{L}_{conf}$.

Backbone Network

As illustrated in Figure 2, we use the first 13 convolutional layers in VGG-16_bn (Simonyan and Zisserman 2014) as the encoder network. Specifically, there are totally five feature levels with downsampling strides of $\{1, 2, 4, 8, 16\}$ respectively, and the corresponding feature maps are denoted as $\{V_1, V_2, V_3, V_4, V_5\}$. The feature map P_i in each decoding stage is generated by the combination of V_i and P_{i+1} as follows. Firstly, the feature map P_{i+1} from a higher level is up-sampled using nearest-neighbor interpolation to be the same size as V_i. Then the upsampled map is combined with V_i by channel-wise concatenation. Finally, we append 3×3 convolutions with ReLU activation on the merged map to generate the final feature map P_i. In particular, P_5 is generated by simply applying convolution on V_5.

Experiments

To demonstrate the superiority of our method, we conduct extensive experiments on four challenging datasets, including ShanghaiTech dataset (Zhang et al. 2016), UCF_CC_50 dataset (Idrees et al. 2013), UCF-QNRF dataset (Idrees et al. 2018) and WorldExpo'10 dataset (Zhang et al. 2015). Following (Zhang et al. 2016), we generate the ground-truth density map using Gaussian kernel with a fixed sigma of 4. As for UCF_CC_50, we use the geometry-adaptive kernel. Similar to (Zhang et al. 2016), we also adopt Mean Absolute Error (MAE) and Mean Squared Error (MSE) as the evaluation metrics. The codes are implemented using the open-source C^3-framework (Gao et al. 2019).

Implementation Details

The encoder in our backbone is composed of the first 13 convolutional layers in VGG-16_bn (Simonyan and Zisserman 2014) which has been pretrained on ImageNet. During training, we randomly select eight images for each epoch and crop four image patches with a fixed-size of 128×128 from each image. Thus the effective batch size is 32. We also use random horizontal flipping with a probability of 0.5 as data augmentation. The image patch size k in the confidence branch is set as 32. The weight term γ is set as 1 and the λ is set to 10. We optimize the model using Adam algorithm (Kingma and Ba 2014) with a fixed learning rate of 1e-5.

Ablation Study

We perform ablation studies on ShanghaiTech PartA dataset to analyze the effect of the proposed modules.

Method	MAE	MSE
Backbone + Avearage	57.48	96.79
Backbone + Adaptive (GT)	*46.19*	*82.19*
Backbone + Adaptive*	55.71	89.82
Backbone + Adaptive	**54.75**	**89.55**

Table 1: Comparison of different selection strategies. * represents directly using the prediction from the most appropriate feature level.

Effectiveness of The Scale-Adaptive Selection. We firstly show the potential improvements by taking full advantage of the multi-level features. We set up a baseline model by simply averaging the predictions from different feature levels, which are supervised with the same ground-truth target. As shown in Table 1, such an average aggregation method achieves an MAE of 57.48. However, for a specific patch, if we aggregate the predictions by selecting the feature level with minimum counting error, the MAE is improved to 46.19, denoted by *Backbone+Adaptive (GT)* in Table 1. This significant improvement demonstrates the great potential of selecting the most appropriate feature level. Unfortunately, it is infeasible to conduct such selection due to the lack of ground truth during inference.

Thanks to the proposed adaptive feature level selection strategy, we can select the appropriate feature level with the guidance from the scores in the confidence branch. As shown in Table 1, directly using the prediction from feature level with the highest score as the final estimation yields an MAE of 55.71. Compared with the sub-optimal strategy of averaging the predictions, the relative improvement of MAE is 3.1%, which demonstrates the effectiveness of our scale-adaptive selection strategy. Besides, after taking the predictions from the other feature levels into consideration via weighted average, the MAE of our method is further improved to 54.75. The improvement is quite reasonable since there exists a gap between discrete feature levels and continuous scale variation.

Method	MAE	MSE
Baseline	61.04	104.6
Baseline + PRA Loss	**57.50**	**89.38**
Backbone + Adaptive	54.75	89.55
Backbone + Adaptive + PRA Loss	**53.59**	**88.38**

Table 2: Ablation study of the PRA Loss.

Effectiveness of The PRA Loss. The widely used Euclidean distance based loss only focuses on pixel-wise learning, while our PRA Loss selects the most hard pixels with awareness of whether the parent patch is over-estimated or under-estimated. To demonstrate the effectiveness of the further optimization for these region-aware hard pixels, we set up a baseline model which only uses the top feature level P_1 as the final prediction to avoid the influence from other modules. As shown in Table 2, the baseline model achieves an MAE of 61.04. After integrating the PRA Loss to the baseline model, the MAE is 57.50 with a relative improvement of 5.8%, which demonstrates the effectiveness of our PRA Loss. When combining with the adaptive selection strategy, the MAE of our model reaches the best performance with MAE of 53.59 and MSE of 88.38. The relative improvement is 2.1%, which proves the necessity of the fine-grained region-aware optimization.

The Effect of Image Patch Size. Since our adaptive feature level selection strategy is conducted in a patch-wise style. A crucial choice is the selection for the image patch size k. As shown in Table 5, if the size k is too large, there might be a large scale variation for heads in the patch which makes the model ambiguous to select the most appropriate feature level. On the contrary, if the patch size k is too small, it is hard to cover a single head completely which makes it hard to determine the approximate scale of the head. Although the selection of the patch size might affect the performance of our method, the results with a range of sizes perform better than simply averaging the multi-level predicted density maps, which also proves the effectiveness of our scale-adaptive selection strategy. Since the patch size of 32 performs the best, we use 32 as default size in all experiments.

Comparisons with State-of-the-Arts

In this section, we compare our SASNet with state-of-the-art methods on four challenging datasets with various densities.

Methods	Venue	SHTech PartA		SHTech PartB		UCF_CC_50		UCF-QNRF	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
CSRNet (Li, Zhang, and Chen 2018)	CVPR	68.2	115.0	10.6	16.0	266.1	397.5	120.3	208.5
CAN (Liu, Salzmann, and Fua 2019)	CVPR	62.3	100.0	7.8	12.2	212.2	243.7	107.0	183.0
BL (Ma et al. 2019)	ICCV	62.8	101.8	7.7	12.7	229.3	308.2	88.7	154.8
SANet + SPANet (Cheng et al. 2019)	ICCV	59.4	92.5	6.5	**9.9**	232.6	311.7	-	-
DSSINet (Liu et al. 2019)	ICCV	60.63	96.04	6.85	10.34	216.9	302.4	99.1	159.2
DUBNet (Oh, Olsen, and Ramamurthy 2020)	AAAI	64.6	106.8	7.7	12.5	243.8	329.3	105.6	180.5
SDANet (Miao et al. 2020)	AAAI	63.6	101.8	7.8	10.2	227.6	316.4	-	-
ADSCNet (Bai et al. 2020)	CVPR	55.4	97.7	6.4	11.3	198.4	267.3	**71.3**	**132.5**
ASNet (Jiang et al. 2020)	CVPR	57.78	90.13	-	-	174.84	251.63	91.59	159.71
AMRNet (Liu, Yang, and Ding 2020)	ECCV	61.59	98.36	7.02	11.00	184.0	265.8	86.6	152.2
AMSNet (Hu et al. 2020)	ECCV	56.7	93.4	6.7	10.2	208.4	297.3	101.8	163.2
Ours	-	**53.59**	**88.38**	**6.35**	**9.9**	**161.4**	**234.46**	85.2	147.3

Table 3: Comparison with state-of-the-art methods on four challenging datasets.

Method	Venue	WorldExpo'10					
		S1	S2	S3	S4	S5	avg.
SCNet (Wang et al. 2018)	BMVC	1.8	**9.6**	14.2	13.3	3.2	8.4
DSSINet (Liu et al. 2019)	ICCV	1.57	9.51	9.46	10.35	2.49	6.67
ASNet (Jiang et al. 2020)	CVPR	2.22	10.11	8.89	**7.14**	4.84	6.64
Ours	-	**1.134**	10.07	**7.68**	7.61	**2.07**	**5.71**

Table 4: Comparison with state-of-the-art methods on the WorldExpo'10 dataset.

Patch size k	MAE	MSE
8×8	55.55	95.08
16×16	55.38	91.39
32×32	**54.75**	89.55
64×64	55.54	**88.38**

Table 5: Effect of different image patch size k

The results are illustrated in Table 3 and Table 4. The best performance is indicated by bold numbers and the second best is indicated by underlined numbers.

ShanghaiTech Dataset. ShanghaiTech dataset consists of two parts: ShanghaiTech partA and ShanghaiTech partB. The partA is collected from the Internet and contains highly congested scenes. While the partB is collected from a busy street and represents relatively sparse scenes. Our SASNet achieves the best performance on both partA and partB. In particular, for the highly congested partA, the SASNet reduces the MAE by 21.4% and MSE by 23.1% compared with CSRNet (Li, Zhang, and Chen 2018). Even compared with the second best performance, the SASNet can still bring a reduction of 3.3% in MAE and 4.5% in MSE respectively.

UCF_CC_50. UCF_CC_50 is a small dataset with only 50 images collected from the Internet. But this dataset has great variation of crowd numbers under complicated scenes. We follow previous work (Idrees et al. 2013) to conduct a five-fold cross validation. As shown in Table 3, our SASNet surpasses all the other methods. Compared with methods with the second best performance, the SASNet reduces the MAE by 7.7% and MSE by 3.8% .

UCF-QNRF. UCF-QNRF is a challenging dataset which has a much wider range of counts than currently available crowd datasets. Due to the existence of high resolution images, we limit the maximum size of images within 1920 pixels. Even with the missing detail information introduced by the downsampling, our method can still achieve the second best performance with MAE of 85.2 and MSE of 147.3.

WorldExpo'10. WorldExpo'10 dataset is collected from surveillance cameras in Shanghai WorldExpo 2010. To eliminate the interference from areas out of the provided RoI, we blur these areas with mean filtering. Our SASNet achieves the best performance on the average MAE, with a reduction of 14% compared with ASNet (Jiang et al. 2020).

Conclusion

In this work, we propose the SASNet, which can effectively address the large scale variation problem in crowd counting. With the proposed patch-wise scale-adaptive feature level selection strategy, the SASNet selects the prediction from the most appropriate feature level as the final prediction of a local patch. Such an adaptive selection strategy takes full advantage of the multi-scale feature representations in the multi-level network. Besides, we propose a novel PRA Loss to recursively select the most hard pixels within a patch, and these region-aware hard pixels are further optimized in a fine-grained way. The PRA Loss acts as a fine-grained complement for the patch-wise feature level selection strategy, and also helps to reduce the inconsistency between training target and evaluation metric. Extensive experiments on four challenging datasets have demonstrated the effectiveness of our contributions.

Acknowledgments

We would like to thank Hao Lu for helpful discussions and suggestions.

References

Bai, S.; He, Z.; Qiao, Y.; Hu, H.; Wu, W.; and Yan, J. 2020. Adaptive Dilated Network With Self-Correction Supervision for Counting. In *CVPR*, 4594–4603.

Boominathan, L.; Kruthiventi, S. S.; and Babu, R. V. 2016. Crowdnet: A deep convolutional network for dense crowd counting. In *ACM Multimedia*, 640–644.

Cheng, Z.-Q.; Li, J.-X.; Dai, Q.; Wu, X.; and Hauptmann, A. G. 2019. Learning spatial awareness to improve crowd counting. In *ICCV*, 6152–6161.

Gao, J.; Lin, W.; Zhao, B.; Wang, D.; Gao, C.; and Wen, J. 2019. C^3 Framework: An Open-source PyTorch Code for Crowd Counting. *arXiv preprint arXiv:1907.02724* .

Guo, D.; Li, K.; Zha, Z.-J.; and Wang, M. 2019. Dadnet: Dilated-attention-deformable convnet for crowd counting. In *ACM Multimedia*, 1823–1832.

Hu, Y.; Jiang, X.; Liu, X.; Zhang, B.; Han, J.; Cao, X.; and Doermann, D. 2020. NAS-Count: Counting-by-Density with Neural Architecture Search. In *ECCV*.

Idrees, H.; Saleemi, I.; Seibert, C.; and Shah, M. 2013. Multi-source multi-scale counting in extremely dense crowd images. In *CVPR*, 2547–2554.

Idrees, H.; Tayyab, M.; Athrey, K.; Zhang, D.; Al-Maadeed, S.; Rajpoot, N.; and Shah, M. 2018. Composition loss for counting, density map estimation and localization in dense crowds. In *ECCV*.

Jiang, X.; Xiao, Z.; Zhang, B.; Zhen, X.; Cao, X.; Doermann, D.; and Shao, L. 2019. Crowd counting and density estimation by trellis encoder-decoder networks. In *CVPR*, 6133–6142.

Jiang, X.; Zhang, L.; Xu, M.; Zhang, T.; Lv, P.; Zhou, B.; Yang, X.; and Pang, Y. 2020. Attention Scaling for Crowd Counting. In *CVPR*, 4706–4715.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Li, Y.; Zhang, X.; and Chen, D. 2018. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *CVPR*, 1091–1100.

Liu, L.; Qiu, Z.; Li, G.; Liu, S.; Ouyang, W.; and Lin, L. 2019. Crowd counting with deep structured scale integration network. In *ICCV*, 1774–1783.

Liu, W.; Salzmann, M.; and Fua, P. 2019. Context-aware crowd counting. In *CVPR*, 5099–5108.

Liu, X.; Yang, J.; and Ding, W. 2020. Adaptive Mixture Regression Network with Local Counting Map for Crowd Counting. In *ECCV*.

Ma, Z.; Wei, X.; Hong, X.; and Gong, Y. 2019. Bayesian loss for crowd count estimation with point supervision. In *ICCV*, 6142–6151.

Miao, Y.; Lin, Z.; Ding, G.; and Han, J. 2020. Shallow Feature Based Dense Attention Network for Crowd Counting. In *AAAI*, 11765–11772.

Oh, M.-h.; Olsen, P. A.; and Ramamurthy, K. N. 2020. Crowd Counting with Decomposed Uncertainty. In *AAAI*, 11799–11806.

Onoro-Rubio, D.; and López-Sastre, R. J. 2016. Towards perspective-free object counting with deep learning. In *ECCV*, 615–629.

Sam, D. B.; Surya, S.; and Babu, R. V. 2017. Switching convolutional neural network for crowd counting. In *CVPR*, 4031–4039.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* .

Valloli, V. K.; and Mehta, K. 2019. W-net: Reinforced u-net for density map estimation. *arXiv preprint arXiv:1903.11249* .

Varior, R. R.; Shuai, B.; Tighe, J.; and Modolo, D. 2019. Multi-Scale Attention Network for Crowd Counting. *arXiv preprint arXiv:1901.06026* .

Wang, Z.; Xiao, Z.; Xie, K.; Qiu, Q.; Zhen, X.; and Cao, X. 2018. In defense of single-column networks for crowd counting. *arXiv preprint arXiv:1808.06133* .

Zhang, A.; Shen, J.; Xiao, Z.; Zhu, F.; Zhen, X.; Cao, X.; and Shao, L. 2019a. Relational attention network for crowd counting. In *ICCV*, 6788–6797.

Zhang, A.; Yue, L.; Shen, J.; Zhu, F.; Zhen, X.; Cao, X.; and Shao, L. 2019b. Attentional neural fields for crowd counting. In *ICCV*, 5714–5723.

Zhang, C.; Li, H.; Wang, X.; and Yang, X. 2015. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*, 833–841.

Zhang, L.; Shi, M.; and Chen, Q. 2018. Crowd counting via scale-adaptive convolutional neural network. In *WACV*, 1113–1121.

Zhang, Y.; Zhou, D.; Chen, S.; Gao, S.; and Ma, Y. 2016. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*, 589–597.