

Crowd Counting Using Scale-Aware Attention Networks

Mohammad Asiful Hossain

Mehrdad Hosseinzadeh

Omit Chanda

Yang Wang

University of Manitoba

{hossaima, mehrdad, omitum, ywang}@cs.umanitoba.ca

Abstract

In this paper, we consider the problem of crowd counting in images. Given an image of a crowded scene, our goal is to estimate the density map of this image, where each pixel value in the density map corresponds to the crowd density at the corresponding location in the image. Given the estimated density map, the final crowd count can be obtained by summing over all values in the density map. One challenge of crowd counting is the scale variation in images. In this work, we propose a novel scale-aware attention network to address this challenge. Using the attention mechanism popular in recent deep learning architectures, our model can automatically focus on certain global and local scales appropriate for the image. By combining these global and local scale attentions, our model outperforms other state-of-the-art methods for crowd counting on several benchmark datasets.

1. Introduction

We consider the problem of crowd counting in arbitrary static images. Given an arbitrary image of a crowded scene without any prior knowledge about the scene (e.g. camera position, scene layout, crowd density), our goal is to estimate the density map of the input image, where each pixel value in the density map corresponds to the crowd density at the corresponding location of the input image. The crowd count can be obtained by integrating the entire density map. In particular, we focus on the setting where the training data have dotted annotations, i.e. each object instance (e.g. people) is annotated with a single point in the image.

Crowd counting has many real-world applications, such as surveillance, public safety, traffic monitoring, urban planning [17]. The methods developed for crowd counting can also be used in counting objects in many other domains, such as counting cells or bacteria in microscopic images [28], counting animals for ecologic studies [1], counting vehicles in traffic control [14, 26, 27].

The challenges of crowd counting are manifold, including severe occlusion, perspective distortion, diverse crowd



Figure 1. Sample images from crowd counting datasets. These images tend to have severe occlusions, perspective distortion, diverse crowd densities, etc. These factors make crowd counting a challenging problem.

densities, and so on (see Fig. 1 for some sample images). Some early work on crowd counting is based on head detection [7]. In recent years, convolutional neural networks (CNNs) have become popular in crowd counting. Most of the CNN-based approaches [25, 28, 17, 20] work by estimating a density map from the image, then obtain the crowd count based on the density map. The accuracy of crowd counting largely depends on the quality of the estimated density map.

Due to the variations of people/head sizes in crowded scenes, a standard feed-forward CNN model with a single scale usually cannot effectively capture the diverse crowd densities in an image. Several recent work [28, 17, 20] has demonstrated the importance of using multi-scale CNN architectures in crowd counting. Zhang et al. [28] propose a multi-column CNN (MC-CNN) architecture with several branches, where each branch uses filters of different sizes. The features from these branches are combined together for the density map estimation. The final features will capture the multi-scale information of the image. Sam et al. [17] use a similar intuition and develop an architecture called switching convolutional neural network (Switch-CNN). Switch-CNN also uses multiple branches to extract features using filters of different sizes. However, instead of concatenating the features maps from all the branches, Switch-CNN learns a classifier that predicts a discrete density class (i.e. scale) of an input image, then uses this predicted scale to choose one of the branches and uses the features from that branch for the density estimation. Sindagi et al. [20] pro-

pose the contextual pyramid CNNs (CP-CNN). In addition to global density class of the whole image, CP-CNN also predicts the local density class for each patch in the image. The contextual information obtained from both global and local density class predictions are combined for the final density estimation.

In recent years, attention models have shown great success in various computer vision tasks [13, 24]. Instead of extracting features from the entire image, the attention mechanism allows models to focus on the most relevant features as needed. Our model is partly inspired by some work of Chen et al. [5] in semantic segmentation. This work uses attentions to focus on the relevant scale dimension of an image for semantic segmentation. In this paper, we use a similar idea and develop a CNN architecture with scale-aware attentions for crowd density estimation and counting. The attention in our model plays a similar role to the “switch” (i.e. density classifier) in Switch-CNN [17]. Switch-CNN makes a hard decision by selecting a particular scale based on the density classifier output and only uses the features corresponding to that scale for the final prediction. The problem is that if the density classifier is not completely accurate, it might select the wrong scale and lead to incorrect density estimation in the end. In contrast, the attention in our model acts as a “soft switch”. Instead of selecting a particular scale, we re-weight the features of a particular scale based on the attention score corresponding to that scale.

The contributions of this paper are manifold. First, we introduce the attention mechanism in CNN-based crowd counting models. Although attention models have been successful in many other vision tasks, our work is the first to use attention models in crowd counting. Second, previous work usually uses attention models to focus on certain spatial locations in an image. In our work, we instead use attentions to focus on certain scales. Previous methods [28, 17, 20] in crowd counting select the scale using a learned classifier. In contrast, the attention mechanism allows our model to “softly” select the scales. Finally, we demonstrate that our proposed approach outperforms other state-of-the-art methods on several benchmark datasets.

2. Related Work

Most existing crowd counting approaches work by first extracting low-level features from images, then map these features to density maps or counts using various techniques. Loy et al. [11] has categorized the existing methods into three groups: (1) detection-based methods (2) regression-based methods, and (3) density estimation based methods. In the following, we briefly review some of these methods. Interested reader may refer to [11] for a more extensive review.

Early work [21] on crowd counting uses detection-based

approaches. These approaches usually apply a person or head detector on an image. Detection-based approaches often cannot handle high density crowd. To address the limitation of detection-based methods, some work [7] uses a regression-base method that directly learns the mapping from an image patch to the count.

In recent years, convolutional neural networks (CNNs) have been popular in almost all vision tasks including crowd counting. Walach et al. [22] propose a method for learning CNN-based crowd counting model in a layer-wise fashion. Zhang et al. [25] introduce a cross-scene crowd counting method by fine-tuning a CNN model to the target scene.

One particular challenge in crowd counting is the scale variation in crowd images. Zhang et al. [28] propose a multi-column architecture (MC-CNN) for crowd counting. This multi-column architecture uses three branches of CNNs. Each branch works at a different scale level of the input image. The three branches are fused in the end to produce the output. Onoro-Rubio and López-Sastre [14] addressed the scale variation issue by proposing a scale-aware counting model called Hydra CNN. This model is formulated as a regression model in which the network learns the way of mapping of the image patches to their corresponding object density maps. Also, Boominathan et al. [3] tackle the issue of scale variation using a combination of deep and shallow networks.

Although these methods are proved to be robust to scale variations, they have some adverse effect that limits the size of input image during training. By reducing the training image size, they would not be capable of learning the features of original image size. To address this drawback, Sindagi et al. [19] presented a novel end-to-end cascaded CNN that jointly produces the estimated count and high quality density map. The high-level features of their network enables it to learn globally relevant and discriminative features.

Sam et al. [17] introduce a coarse to fine Switch-CNN network that chooses a branch corresponding to an estimated scale instead of fusing all branches. Sindagi et al. [20] develop a contextual pyramid CNN (CP-CNN) that combines both global and local contextual information for crowd counting. Most recently, Liu et al. [10] introduce a method called DecideNet. DecideNet has two counting models – a regression mode and a detection mode. Depending on the real density condition at a location in an image, DecideNet will learn to automatically switch between these two modes.

Inspired by the previous work, we propose a new approach for handling scale variations in crowd counting. Our proposed model uses attentions to automatically focus on a particular scale, both at the whole image level and at the local patch level. Our model is conceptually simpler than DecideNet [10], since we do not need to switch between two different counting modes.

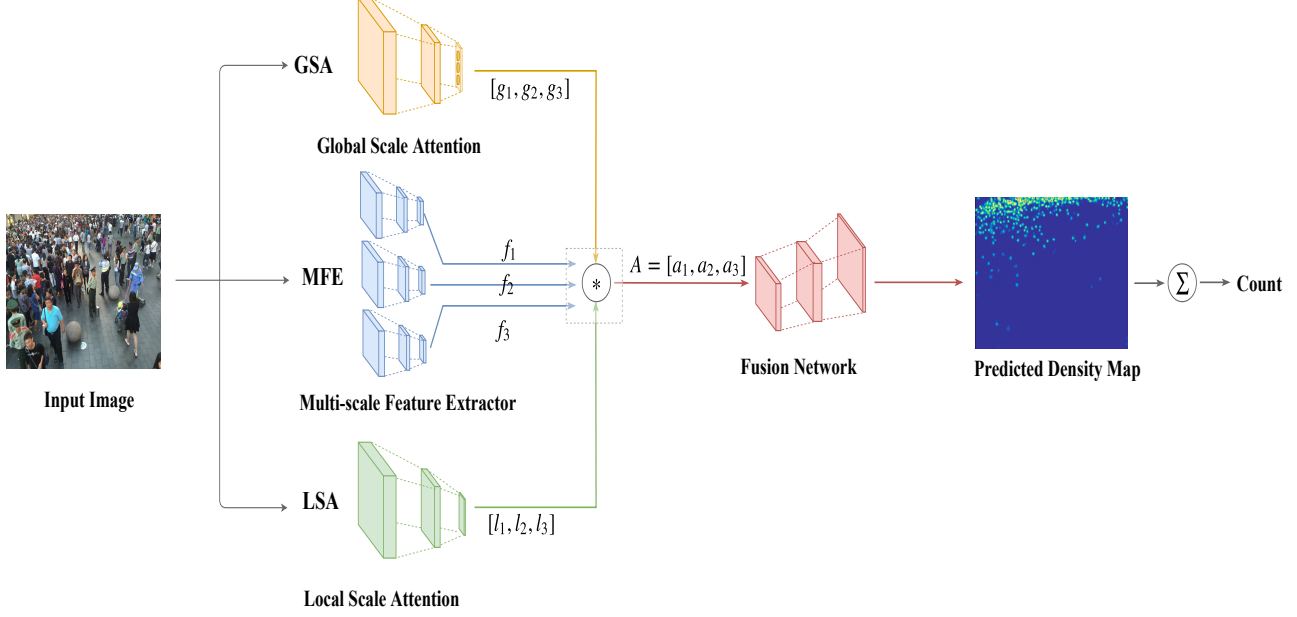


Figure 2. Overview of the proposed method. The input image is passed simultaneously through three sub-networks: multi-scale feature extractor (MFE), global scale attentions (GSA), and local scale attentions (LSA). MFE extracts feature maps in three different scales. GSA and LSA produce three global scores and three pixel-wise local attention maps, respectively. Multi-scale feature maps are then weighted by the corresponding GSA and LSA outputs. The attention-weighted features are then used as the input to the fusion network to predict the density map. Finally, the crowd count can be obtained by summing over the entries in the predicted density map.

3. Our Approach

Crowd densities can vary dramatically both across different images and across different spatial locations within the same image. We propose to use both global and local attention weights to capture inter-image and intra-image variations of crowd density. This allows our model to adaptively use features at appropriate scales. Our proposed approach has several modules (see Fig 2): multi-scale feature extractor (MFE), global scale attentions (GSA), local scale attention (LSA), and the fusion network (FN) for density estimation. Following, we describe each module in details.

3.1. Multi-Scale Feature Extractor

The goal of this module is to extract multi-scale feature maps from an input image. Inspired by the success of using multi-branch architectures in crowd-counting [28], we use a similar multi-branch architecture to extract feature maps at different scales. The architecture (see Fig. 3) has three branches associated with three different scales. It takes an image of arbitrary size as its input. Each branch then independently processes the input at its corresponding scale level. Each branch consists of multiple blocks of convolution layers with different filter sizes. By choosing different filter sizes in these branches, we can change the receptive field in each branch and capture the features at different scales. Each branch also has several max-pooling layers.

We choose the filter sizes in max-pooling so that the output feature map in each branch has a spatial dimension equal to one-fourth of the input image dimension.

3.2. Global Scale Attention

Previous work [17, 20, 19] has shown the benefit of leveraging global scale information in crowd counting problem. In our model, we use a global scale attention (GSA) module to capture global contextual information about how dense the image is. This module takes an input image and produces three attention scores. Each score corresponds to one of the three pre-defined density label level: low-density, mid-density, and high-density. The number of density levels is equal to the number of scales in the multi-scale feature extraction module (Sec. 3.1). The architecture of GSA is illustrated in Fig. 4.

GSA outputs three scores g_i ($i = 1, 2, 3$) for each input image, representing the extent to which the input image belongs to each of the three density labels. A softmax layer is used at the end of the pipeline to normalize the scores (which can be interpreted as “attentions”) to sum to one.

3.3. Local Scale Attention

The GSA module captures the overall density level of an image. But an image may have different density levels at different locations. The global density level may not sufficient to capture the fine-grained local contextual infor-

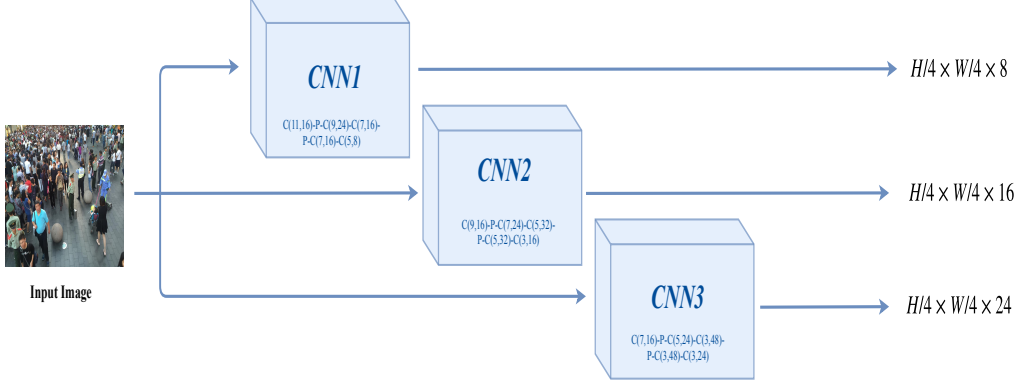


Figure 3. Illustration of the multi-branch feature extractor (MFE) module. Here we use three branches associated with three different scales. Each branch uses different filter sizes to extract features at different scales. Using max-pooling, the output of each branch has a spatial dimension equal to one-fourth of the input image size. In this figure, we use $C(i, j)$ to denote a 2D convolution layer with filter sizes of $i \times i$ and j output feature maps. We use P to denote max-pooling.

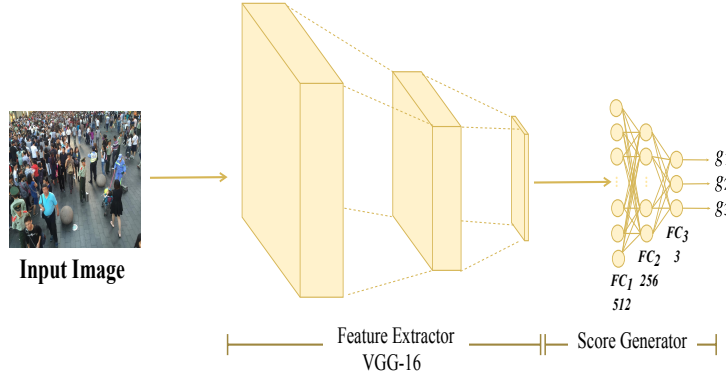


Figure 4. Illustration of the global scale attention (GSA) module. Given an input image, this module generates three attention scores. Each score corresponds to one of the three pre-defined density labels: low-density, mid-density, and high-density.

mation at different locations in an image. Inspired by [20], we incorporate a *local scale attention (LSA)* module to capture the local scale information at different locations in an image. LSA generates pixel-wise attention maps representing the scale information at different locations. Similar to the global attention module, here we also consider three different scale levels. Unlike GSA that produces three scalar scores, LSA will produce three pixel-wise attention maps. These attention maps have the same spatial dimensions as the corresponding multi-scale feature maps (Sec. 3.1).

The LSA module consists of eight convolution and two max-pooling layers, followed by three fully connected layers (see Fig. 5). A sigmoid layer is placed at the end of the module to ensure the values in the attention maps are between 0 to 1.

3.4. Fusion Network

The last component of the proposed method is the fusion network (FN) which produces the final density map for an input image. This module takes the extracted feature maps from the image which are re-weighted by the global and

local attention scores. The output of this component is a predicted density map. The final crowd count can be obtained by summing over all entries of the estimated density map.

Let $f_i \in \mathbb{R}^{H_i \times W_i \times D_i}$ ($i = 1, 2, 3$) denote the feature maps corresponding to the three different scales, where $H_i \times W_i$ is the spatial dimension and D_i is the number of channels for the i^{th} feature map. We use $g_i \in \mathbb{R}$ ($i = 1, 2, 3$) and $l_i \in \mathbb{R}^{H_i \times W_i}$ ($i = 1, 2, 3$) to denote the corresponding global and local attention scores, respectively. We can use the global and local attention scores to re-weight the feature maps. Let $f_i^{h,w,d}$ denote the (h, w, d) entry of the feature map corresponding to the i -th scale. Similarly, let $l_i^{h,w}$ denote the (h, w) entry of the corresponding local attention map l_i . We use $a_i \in \mathbb{R}^{H_i \times W_i \times D_i}$ to denote the attention-weighted feature map for the i -th scale. The (h, w, d) entry $a_i^{h,w,d}$ of a_i is calculated as follows:

$$a_i^{h,w,d} = g_i \cdot l_i^{h,w} \cdot f_i^{h,w,d}, \quad \text{where } i = 1, 2, 3 \quad (1)$$

As shown in Fig. 3 and described in Sec. 3.1, the extracted feature maps for all three scales are of size $H/4 \times W/4$



Figure 5. Local Scale Attention Network. This module produces three feature maps of size $H/4 \times W/4$ each of which associated to a density scale. $C(i, j)$ denotes a 2D convolution with a filter size of $i \times i$ and with j output feature maps. P stands for max pooling.

where $H \times W$ is the spatial dimension of the input image. However, the feature maps can have different depth (i.e. number of channels) among different scales. Here the depth dimension is set to 24, 16 and 8 for the low-density, mid-density, and high-density scales, respectively.

The attention-weight feature maps for different scales are then concatenated together and fed into the fusion network (FN) module to produce the density map. The FN module consists of several convolution layers along with 2 de-convolution layers which resize the feature maps into size of $H/2 \times W/2$ and eventually $H \times W$, where $H \times W$ is the spatial dimensions of the original input image. This output of this stage is a feature map of size $H \times W \times 16$. Finally, we apply a 1×1 convolution to produce the density map (DM) of size $H \times W$ (i.e. of depth 1). To obtain the final crowd count from the 2-D density map DM , we simply sum over the entries in the density map as follows:

$$Count = \sum_{j=1}^H \sum_{k=1}^W DM(j, k) \quad (2)$$

where $DM(j, k)$ is the value of at the spatial position (j, k) in the predicted density map.

3.5. Loss Function

In order to train the model parameters, we define an overall loss function L_{final} consisting of three losses:

$$L_{final} = L_{DM} + \lambda_g \cdot L_{GSA} + \lambda_l \cdot L_{LSA} \quad (3)$$

In Eq. 3, L_{DM} is a loss function defined on the predicted density map. This loss will encourage the model to predict density maps close to the ground-truth density maps on the training data. We also use two auxiliary losses L_{GSA} and L_{LSA} . These two losses will encourage the predicted global and local density scale attentions to be similar to the ground-truth global and local scales, respectively. The hyperparameters λ_g and λ_l are used to control the relative contributions of the two auxiliary losses. In the following, we provide details of these loss functions. To simplify the notation, we focus on the definition of each loss function on one single training image. The final loss will be accumulated over all training images in the end.

Let $C \in \mathbb{R}^{H \times W}$ be the predicted density map on a training image and $C_{gt} \in \mathbb{R}^{H \times W}$ be the corresponding ground-truth density map. Here $H \times W$ is the spatial dimension of the input image. The loss L_{DM} is defined as the square of the Frobenius norm between C and C_{gt} , i.e. $L_{DM} = \frac{1}{2} \|\text{vec}(C) - \text{vec}(C_{gt})\|^2$ where $\text{vec}(\cdot)$ concatenates entries of a matrix into a vector.

The auxiliary loss L_{GSA} is used to encourage the predicted global scale attention scores to be close to the ground-truth global scale on a training image. We obtain the ground-truth global scale as follows. First, we find the maximum and minimum crowd count (denoted as $Count_{max}$ and $Count_{min}$, respectively) on the training data. We split the range $[Count_{min}, Count_{max}]$ into three bins of equal sizes. For a training image, we assign its ground-truth global scale g_{gt} ($g_{gt} \in \{1, 2, 3\}$) according to the bin that the ground-truth crowd count falls into. Let $g \in \mathbb{R}^3$ be a vector of the global attention scores (corresponding to 3 different scales) on this training image. We can consider g to be the score of classifying the image into one of the three global scales. We define L_{GSA} using the standard cross-entropy loss as $L_{GSA} = CE(g, g_{gt})$ where $CE(\cdot)$ denotes the multi-class cross-entropy loss function.

The loss L_{LSA} is used to encourage the predicted local scale at each spatial location to be consistent with the ground-truth local scale on a training image. We generate the ground-truth local scale as follows. For a pixel location in a training image, we obtain a local crowd count at this location by summing over the ground-truth density map in the 64×64 neighborhood of this location. We then find the minimum/maximum of local crowd count on training images. Similarly, we split the range to three bins and assign the ground-truth scale at a pixel location according to the bin that the local crowd count (over 64×64 neighborhood) falls into. Let $l_{gt} \in \mathbb{R}^{H \times W}$ be a matrix of ground-truth local scales of a training image, where $H \times W$ denotes the spatial dimension. Each entry l_{gt} ($l_{gt} \in \{1, 2, 3\}$) indicates the ground-truth local scale at the corresponding spatial location. Let $l \in \mathbb{R}^{H \times W \times 3}$ denote a tensor of predicted local scale attentions. We define L_{LSA} as the sum of cross-entropy losses over all spatial locations, i.e. $L_{LSA} = \sum_{h=1}^H \sum_{w=1}^W CE(l[h, w, :], l_{gt}[h, w])$ using Matlab notations.

Empirically, we have found that the auxiliary losses help regularizing the model and improve the performance. In the experiments, we will provide ablation analysis on the impact of these auxiliary losses.

4. Experiments

We first introduce the datasets and our experimental setup. We then present experimental results on three benchmark datasets. Finally, we perform ablation studies to further analyze our proposed approach.

4.1. Datasets and Setup

Datasets: We evaluate our proposed method on three benchmark datasets: ShanghaiTech PartB [28], Mall dataset [4] and UCF_CC_50 [7]. Table 1 shows various statistics of these datasets. The ShanghaiTech PartB dataset contains 716 where 400 images are for training and the other 316 ones for testing. The Mall dataset [4] has 2000 frames captured from a shopping mall. The first 800 frames are used as training frames and the remaining 1200 frames are used for testing. The UCF_CC_50 dataset [7] contains a total of 50 images from web sources. Clearly, the limited number of images in these datasets raises the need for data augmentation in order to prepare the data for training a deep network. Consequently, we used data augmentation to address the issue. We followed the same data augmentation technique used in existing methods [10, 7, 28, 17, 20].

Evaluation Metric: Following previous work [25, 28, 17, 20, 19], we use Mean Absolute Error (MAE) and Mean Square Error (MSE) as the evaluation metrics. Let N be the number of test images, $Count_{gt}^{(n)}$ be the ground truth count and $Count^{(n)}$ be the predicted count for the n -th test image. These two evaluation metrics are defined as follows:

$$MAE = \frac{1}{N} \sum_{n=1}^N |Count^{(n)} - Count_{gt}^{(n)}| \quad (4)$$

$$MSE = \sqrt{\frac{1}{N} \sum_{n=1}^N |Count^{(n)} - Count_{gt}^{(n)}|^2} \quad (5)$$

Ground-truth Density Map: On each dataset, head annotations (i.e. center of the head of a person) are provided as points. Following [10], we generate the ground-truth density map from these point annotations by applying a Gaussian kernel normalized to have a sum of one.

Training Details: We follow the data augmentation technique used in previous methods [10, 7, 28, 17, 20]. Since there are lots of parameters in our model, directly learning all parameters from scratch is challenging. In our implementation, we use a two-phase training scheme to train our proposed model. During the first phase, we ignore the

LSA module and only learn the parameters of the GSA, MFE, and FN modules. We assign each training image to one of three global scale classes (namely, low-density, mid-density, high-density) according to its ground-truth density map. We then learn the parameters of GSA, MFE, and FN modules by optimizing $L_{DM} + \lambda_g \cdot L_{GSA}$. During the second phase, we train all the modules (including LSA) together. The parameters of the GSA, MFE, and FN modules are initialized with the parameters obtained from the first phase. The parameters of the LSA module are initialized randomly from scratch.

4.2. Experimental Results

On the ShanghaiTech PartB dataset and the Mall dataset, we follow the standard training/testing split used in previous work [10]. On the UCF_CC_50 dataset, we follow [20] and perform 5-fold cross-validation. We apply the same data augmentation used in previous methods [10, 7, 28, 17, 20] on all datasets.

The experimental results on these three datasets are shown in Tables 2, Table 3 and Table 4, respectively. We also compare with existing state-of-the-art results on each of the datasets. On the ShanghaiTech PartB dataset and the Mall dataset, our proposed model significantly outperforms previous approaches in terms of both MAE and MSE. On the UCF_CC_50 dataset, our model outperforms previous approaches in terms of MAE. In terms of MSE, our model performs better than most previous approaches except for [20].

4.3. Ablation Study

We conduct ablation study on the ShanghaiTech PartB dataset to provide further analysis of relative contributions of various components of our approach.

Impact of attention modules: First, we analyze the relative contributions of global and local attentions in our model. In Table 5, we show the results of removing the global or local attention modules (GSA or LSA) in our model. The base model in Table 5 refers the architecture with only the multi-scale feature extractor and the fusion network, i.e. without any attention modules. From the results, we can see that both GSA and LSA contribute to the final performance. Using both GSA and LSA with the base model, we achieve the best performance.

Impact of auxiliary losses: We also study the impact of using the auxiliary losses (Eq. 3) as extra supervisions during training. In this analysis, we use our proposed model with all the modules. But during training, we use various combinations of the three losses: L_{DM} , L_{GSA} and L_{LSA} . The results are shown in Table 6. We can see that both L_{GSA} and L_{LSA} help improving the performance of the learned model. By using all three losses, the best performance is achieved using all three losses during training.

Dataset	Resolution	Images	Max	Min	Avg	Total
ShanghaiTech PartB[28]	1024×768	716	578	9	123.6	88,488
Mall[4]	320×640	2000	53	13	33	62325
UCF_CC_50[7]	varies	50	4543	94	1279.5	63,974

Table 1. Statistics of the three datasets used in the experiments. For each dataset, we show the following information: the image resolution, the number of images, the maximum/minimum number of people in an image, the average and total number of people annotated in the dataset.

Method	ShanghaiTech Part B	
	MAE	MSE
R-FCN [†] [6]	52.35	70.12
Faster R-CNN [†] [16]	44.51	53.22
Cross-Scene [25]	32.00	49.80
MC-CNN [28]	26.40	41.30
Switching-CNN [17]	21.60	33.40
CP-CNN [20]	20.1	30.1
FCN [12]	23.76	33.12
DecideNet [10]	20.75	29.42
Ours	16.86	28.41

Table 2. Comparison of the performance of different methods on the ShanghaiTech PartB dataset [28]. [†]These results are obtained from [10].

Method	Mall	
	MAE	MSE
DecideNet [10]	1.52	1.90
R-FCN [†] [6]	6.02	5.46
Faster R-CNN [†] [16]	5.91	6.60
SquareChn Detector [†] [2]	20.55	439.10
Count-Forest [15]	4.40	2.40
Exemplary-Density [†] [23]	1.82	2.74
Boosting-CNN [22]	2.01	–
Mo-CNN [8]	2.75	13.40
Weighted_VLAD [18]	2.41	9.12
Ours	1.28	1.68

Table 3. Comparison of the performance of different methods on the Mall dataset [4]. [†]These results are obtained from [10].

5. Conclusion

We have presented scale-aware attention networks for crowd counting in images. The novelty of our work is that we use the attention mechanism to softly select the appropriate scales at both global and local levels. Compared with DecideNet [10] that learns to switch between two different models (detection vs density estimation), our model is much simpler since it only has one model (density estimation). Our experimental results demonstrate that our proposed model outperforms other state-of-the-art approaches for crowd counting.

Method	UCF_CC_50	
	MAE	MSE
Cross-Scene [25]	467.00	498.50
MC-CNN [28]	377.60	509.10
Switching-CNN [17]	318.10	439.20
CP-CNN [20]	295.80	320.90
Lempitsky et al. [9]	493.4	487.1
Idrees et al. [7]	419.5	541.6
Crowd-Net [3]	452.5	–
Hydra-2s [14]	333.73	425.26
Ours	271.60	391.00

Table 4. Comparison of the performance of different methods on the UCF_CC_50 dataset [7].

Methods	MAE	MSE
base model	27.63	46.65
base model+GSA	17.0	30.60
base model+LSA	18.07	31.87
base model+GSA+LSA (this paper)	16.86	28.41

Table 5. Effect of GSA and LSA modules on the ShanghaiTech PartB dataset. The “base model” only contains the multi-scale feature extractor and the fusion network, i.e. without any global or local attention modules. By adding GSA or LSA, we can achieve better performance. The best performance is obtained by using both GSA and LSA together with the base model.

Methods	MAE	MSE
L_{DM}	19.15	35.55
$L_{DM} + L_{LSA}$	17.02	31.49
$L_{DM} + L_{GSA}$	17.33	32.33
$L_{DM} + L_{LSA} + L_{GSA}$ (this paper)	16.86	28.41

Table 6. Effect of using auxiliary losses as extra supervisions on the ShanghaiTech PartB dataset. L_{DM} is the model without any auxiliary losses. By adding L_{LSA} (i.e. $L_{DM} + L_{LSA}$) or L_{GSA} (i.e. $L_{DM} + L_{GSA}$), we can improve the performance. By using both L_{LSA} and L_{GSA} as auxiliary losses (i.e. $L_{DM} + L_{LSA} + L_{GSA}$), we achieve the best performance.

Acknowledgment This work was supported by an NSERC Engage grant in collaboration with Sightline Innovation. We thank NVIDIA for donating some of the GPUs used in this work.



Figure 6. Qualitative examples of density maps. The first two columns are original input images from the Shanghaitech partB [28] dataset and their corresponding density maps respectively. The last two columns are original input images from the Mall [4] dataset and their corresponding density maps respectively.

References

- [1] C. Arteta, V. Lempitsky, and A. Zisserman. Counting in the wild. In *European Conference on Computer Vision*. Springer, 2016.
- [2] R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten years of pedestrian detection, what have we learned? In *European Conference on Computer Vision*. Springer, 2014.
- [3] L. Boominathan, S. S. Kruthiventi, and R. V. Babu. Crowdnet: A deep convolutional network for dense crowd counting. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016.
- [4] K. Chen, C. C. Loy, S. Gong, and T. Xiang. Feature mining for localised crowd counting. In *British Machine Vision Conference*, 2012.
- [5] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [6] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural Information Processing Systems*, 2016.
- [7] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [8] S. Kumagai, K. Hotta, and T. Kurita. Mixture of counting cnns: Adaptive integration of cnns specialized to specific appearance for crowd counting. *arXiv preprint arXiv:1703.09393*, 2017.
- [9] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems*, 2010.
- [10] J. Liu, C. Gao, D. Meng, and A. G. Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [11] C. C. Loy, K. Chen, S. Gong, and T. Xiang. Crowd counting and profiling: Methodology and evaluation. In *Modeling, Simulation and Visual Analysis of Crowds*. Springer, 2013.
- [12] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor. Fully convolutional crowd counting on highly congested scenes. *arXiv preprint arXiv:1612.00220*, 2016.
- [13] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, 2014.
- [14] D. Onoro-Rubio and R. J. Lopez-Sastre. Towards perspective-free object counting with deep learning. In *European Conference on Computer Vision*, 2016.
- [15] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.
- [16] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.
- [17] D. B. Sam, S. Surya, and R. V. Babu. Switching convolutional neural networks for crowd counting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [18] B. Sheng, C. Shen, G. Lin, J. Li, W. Yang, and C. Sun. Crowd counting via weighted vlad on dense attribute feature maps. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.
- [19] V. A. Sindagi and V. M. Patel. CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2017.
- [20] V. A. Sindagi and V. M. Patel. Generating high-quality crowd density maps using contextual pyramid CNNs. In *IEEE International Conference on Computer Vision*, 2017.
- [21] I. S. Topkaya, H. Erdogan, and F. Porikli. Counting people by clustering person detector outputs. In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*. IEEE, 2014.
- [22] E. Walach and L. Wolf. Learning to count with cnn boosting. In *European Conference on Computer Vision*. Springer, 2016.
- [23] Y. Wang and Y. Zou. Fast visual object counting via example-based density estimation. In *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016.
- [24] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, 2015.
- [25] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [26] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua. Visual translation embedding network for visual relation detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [27] S. Zhang, G. Wu, J. P. Costeira, and J. M. F. Moura. FCN-rLSTM: Deep spatial-temporal neural networks for vehicle counting in city cameras. In *IEEE International Conference on Computer Vision*, 2017.
- [28] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.