Crowd counting via scale-adaptive convolutional neural network

Lu Zhang*†
Tencent Youtu
v_youyzhang@tencent.com

Miaojing Shi*
Inria Rennes & Tencent Youtu
miaojing.shi@inria.fr

Qiaobo Chen†
Shanghai Jiaotong University
chenqiaobo@sjtu.edu.cn

Abstract

The task of crowd counting is to automatically estimate the pedestrian number in crowd images. To cope with the scale and perspective changes that commonly exist in crowd images, state-of-the-art approaches employ multi-column CNN architectures to regress density maps of crowd images. Multiple columns have different receptive fields corresponding to pedestrians (heads) of different scales. We instead propose a scale-adaptive CNN (SaCNN) architecture with a backbone of fixed small receptive fields. We extract feature maps from multiple layers and adapt them to have the same output size; we combine them to produce the final density map. The number of people is computed by integrating the density map. We also introduce a relative count loss along with the density map loss to improve the network generalization on crowd scenes with few pedestrians, where most representative approaches perform poorly on. We conduct extensive experiments on the ShanghaiTech, UCF_CC_50 and WorldExpo'10 datasets as well as a new dataset SmartCity that we collect for crowd scenes with few people. The results demonstrate significant improvements of SaCNN over the state-of-the-art.

1. Introduction

The crowd counting task in computer vision is to automatically count the pedestrian number in images/videos. To help with crowd control and public safety in many scenarios such as public rallies and sports, accurate crowd counting is demanded.

Early methods estimate the pedestrian number via detection, where each individual pedestrian in a crowd is detected by trained detectors [35, 13, 7, 10]. This can be very hard if pedestrians are heavily occluded or densely spread. Present methods instead regress the pedestrian number on the whole image and achieve significant improvements [3, 4, 16]. Handcrafted features like HOG [9] were soon outperformed
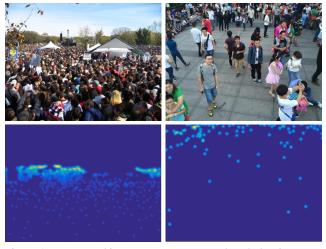

Figure 1. Top: crowd images. Bottom: ground truth density maps.

by modern deep representations [19, 33].

Crowd counting via regression suffers from the drastic changes in perspective and scale, which commonly exist in crowd images (see Fig. 1). To tackle it, multi-column convolutional neural networks (CNN) were recently adopted and have shown robust performance [26, 1, 34]. Different columns corresponding to different filter sizes (small, medium and large) are combined in the end to adapt to the large variations of perspective and scale. They regress a density map of the crowd (see Fig. 1). Pedestrian number is obtained by integrating the density map. [26] further introduced a switch classifier to relay the crowd patches in images to the best CNN column. Each CNN column is trained with its own samples.

We notice that the selection among the multiple columns in [26] is not balanced. Using one single column is able to retain over 70% accuracy of the multi-column model on some datasets. Building upon this observation, in this paper, we propose a scale-adaptive CNN architecture (SaCNN, see Fig. 2) for crowd counting. It offers several new elements as our contributions: (1) We design a single column CNN with a single filter size as our backbone, which is easy to train from scratch. Small-sized filters preserve the spatial resolution and allow us to build a deep network. (2) We

---

*Equal contribution.
†Lu Zhang and Qiaobo Chen have worked on this paper during their internships at Tencent Youtu Lab.

combine the feature maps of multiple layers to adapt the network to the changes in pedestrian (head) scale and perspective. Different layers share the same low-level feature representations, which results in fewer parameters, fewer training data required, and faster training. (3) We introduce a multi-task loss by adding a relative head count loss to the density map loss. It significantly improves the network generalization on crowd scenes with few pedestrians, where most representative works perform poorly on. (4) We collect the new SmartCity dataset with high-angle shot for crowd counting. The existing datasets are taken from outdoors and do not adequately cover those crowd scenes with few pedestrians. The new dataset contains both indoor and outdoor scenes and has an average pedestrian number of 7.4 per image. (5) We conduct extensive experiments on three datasets: ShanghaiTech [34], UCF_CC_50 [9] and WorldExpo'10 [33]; the results show that our SaCNN significantly outperforms the state-of-the-art crowd counting methods. We also compare SaCNN with other representative works on SmartCity to show the generalization ability of ours. Our code and dataset are publicly available[1].

2. Related work

We categorize crowd counting methods as either 1) detection-based methods or 2) regression-based methods. We shall describe both categories below and compare representative works to our method.

2.1. Detection-based methods

Detection-based methods consider a crowd as a group of detected individual entities [31, 32, 28, 29, 2, 21]. Early works focus on video surveillance scenario so that the motion and appearance cues can be utilized [29, 2, 21]. For example, [29] trained a dynamic detector capturing motion and appearance information over two consecutive frame pairs of a video sequence. These works are not applicable in still images for crowd counting. Pedestrians are often occluded in dense crowds, and this is particularly challenging when estimating the crowds in still images. Part-detectors were therefore employed to count pedestrian from parts in images [31, 32]. [28] took advantage of a recurrent deep network to detect the occluded heads in crowd scenes. Notwithstanding their improvements, detection-based methods overall suffer severely in dense crowds with high occlusions among people.

2.2. Regression-based methods

Regression-based methods regress either a scalar value (pedestrian number) [3, 4, 16, 9] or a density map [5, 11, 12] on various features extracted from crowd images. They basically have two steps: first, extracting effective features

from crowd images; second, utilizing various regression functions to estimate the crowd count. Early works utilize handcrafted features, like edge features [3, 5, 25, 23] and texture features [5, 9, 17]. Regression methods include linear [23, 20], ridge [5] and Gaussian [3] functions.

Due to the use of strong CNN features, recent works on crowd counting have shown remarkable progress by regressing a density map of an image [33, 34, 26, 1, 19]. A density map provides much more information than a scalar value, and CNNs have been demonstrated to be particularly good at solving problems "locally" rather than "globally". Crowd counting is casted as that of estimating a continuous density function whose integral over any image region gives the count of pedestrians within that region. [33] designed a multi-task network to regress both density maps and crowd counts. They employed fully-connected layers to regress the absolute crowd count.

2.3. Comparison to our SaCNN

[34, 26, 1] utilized multi-column networks [6] to deal with the variation of head scale in one image. Different columns correspond to different filter sizes. The large-sized filters make the whole network hard to train. The network is pretrained on each column first and finetuned together in the end [34]; whilst our SaCNN employs a single column with a single filter size which can be trained from scratch. [26] introduced a switch classifier using VGG net [27] to relay the crowd patches from images to the best CNN column. The switching is very costly and often not correct. While our SaCNN adapts the feature maps from multiple scales and concatenates them to relay each pedestrian to the best scale in the network.

Similar to [33], we perform a multi-task optimization in SaCNN to regress both the density map and head count. Differently, ours is a multi-scale fully convolutional network and we propose a relative count loss. The scale-adaptive architecture is similar in spirit to [24, 14] by combining feature maps of multiple layers; but we use the deconvolutional layer [18] to adapt the network output instead of upsampling and element-wise summation. Our network design with a backbone of single filter size and multi-scale outputs is new in the crowd counting field. Plus, we claim five-fold new elements in total in Sec. 1.

3. Scale-adaptive CNN

In this section, we first generate the ground truth density maps for training data and then present the architecture of SaCNN.

3.1. Ground truth density maps

Supposing we have a pedestrian head at a pixel z_j, we represent it by a delta function $\delta(z - z_j)$. The ground truth
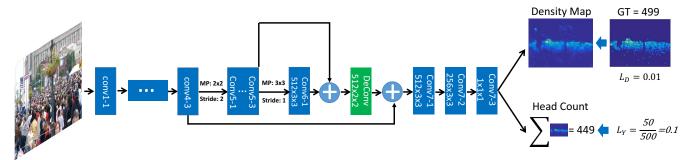

Figure 2. The structure of the proposed scale-adaptive convolutional neural network (SaCNN) for crowd counting. MP: max pooling layer; DeConv: deconvolutional layer; GT: ground truth. We build a single backbone network with a single filter size. We combine the feature maps of multiple layers to adapt the network to the changes in pedestrian scale and perspective. Multi-scale layers share the same low-level parameters and feature representations, which results in fewer parameters, fewer training data required, and faster training. We introduce two loss functions to jointly optimize the model: one is density map loss, the other is relative count loss. The relative count loss helps to reduce the variance of the prediction errors and improve the network generalization on very sparse crowd scenes.

density map $D(z)$ is computed by convolving this delta function using a Gaussian kernel G_σ normalized to sum to one:

$$D(z) = \sum_{j=1}^{M} \delta(z - z_j) * G_\sigma(z), \qquad (1)$$

where M is the total head number. The sum of the density map is equivalent to the total number of pedestrians in an image. Parameter setting follows [34, 26, 1].

3.2. Network architecture

Many state-of-the-art works [34, 26, 1] adopt the multi-column architecture with different filter sizes to address the scale and perspective change in crowd images. We instead propose a scale-adaptive CNN (SaCNN) where we use a single backbone network with a single filter size (see Fig. 2). Density map estimation is performed at a concatenated layer which merges the feature maps from multiple scales of the network; it can therefore easily adapt to the pedestrian scale and perspective variations.

We use all $3 * 3$ filters in the network. The computation required for $3 * 3$ filters is far less than for large filters [34, 26, 1]. Small filters preserve the spatial resolution of the input so that we can build a deep network. "Deeper" has a similar effect as "Wider" (multi-column architecture) does in a network: for instance, a large filter can be simulated by a small filter from deeper layer. The backbone design (until conv4_3 in Fig. 2) follows the VGG architecture [27] in the context of using 3*3 filters and three convolutional layers after each pooling layer.

SaCNN is built from low-level to high-level by gradually adding layers of different scales. For concatenation, we need to carefully adapt their output to be the same size. We first present a *single-scale* model that is illustrated in Fig. 3 (a): it takes the output feature map from conv4_2 and projects it to a density map via 1×1 filters in p-conv. Re-
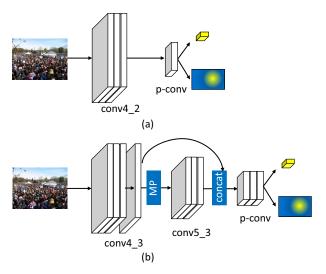

(a)

(b)

Figure 3. (a) single-scale model; (b) two-scale model. MP: max pooling layer of stride 1. The outputs of the networks are the density maps (blue-yellow heatmaps) and head counts (yellow cubes). p-conv denotes 1×1 convolutional layer.

sults in Fig. 5 show that our single-scale model converges fast and its performance is close to the state-of-the art.

Like the multi-column CNN, we want our model to fire on pedestrians (heads) of different sizes. We decide to go deeper. We have three max-pooling layers and several convolutional layers until conv4_3. We add pool4 and three convolutional layers (conv5_1 to conv5_3) after conv4_3 in the network (see Fig. 3 (b)). We propose a *two-scale* model by extracting feature maps from conv4_3 and conv5_3; we set the stride of pool4 to be 1 so that the two feature maps have the same size; we concatenate them to produce the final density map (refer to Fig. 3 (b)) . Results in Fig. 5 demonstrate a clear improvement of our two-scale model over the single-scale model.

Building upon the two-scale network, we construct our *scale-adaptive* model in Fig. 2. We add pool5 and conv6_1 afterwards in the network. In order to combine conv5_3 and conv6_1, we set the stride of pool4 as 2 and pool5 as 1. We upsample the concatenated output to 1/8 resolution of the input using DeConv layer to further concatenate it with conv4_3. The final density map therefore has a spatial resolution of 1/8 times the input Hence, we downsample the ground truth density map by factor 8.

Notice that we have also tried to regress the density map at a lower resolution: *e.g.* 1/16 times the input resolution. The performance is slightly declined compared to concatenating at 1/8 resolution. We suggest it is due to the receptive field being too big in the deeper layer.

3.3. Network loss

Our network training first adopts the Euclidean loss to measure the distance between the estimated density map and the ground truth [34, 26, 1]:

$$L_D(\Theta) = \frac{1}{N} \sum_{i=1}^{N} \|F_D(X_i; \Theta) - D_i\|^2, \qquad (2)$$

where Θ is a set of learnable parameters in the proposed network. N is the number of training images. X_i is the input image and D_i is the corresponding ground truth density map. $F_D(X_i; \Theta)$ denotes the estimated density map for X_i. The Euclidean distance is computed at pixels and summed over them.

Apart from the density map regression, we introduce another loss function regarding the head count (see Fig, 2). We notice that most representative approaches perform poorly on crowd scenes with few pedestrians. This problem can not be resolved via (2), because the absolute pedestrian number is usually not very large in sparse crowds compared to that in dense crowds. To tackle this, we propose a relative head count loss:

$$L_Y(\Theta) = \frac{1}{N} \sum_{i=1}^{N} \|\frac{F_Y(X_i; \Theta) - Y_i}{Y_i + 1}\|^2, \qquad (3)$$

where $F_Y(X_i; \Theta)$ and Y_i are the estimated head count and the ground truth head count, respectively. The denominator Y_i is added by 1 to prevent division by zero. (3) concentrates the learning on those samples with relatively large prediction errors. Results on those very sparse crowd scenes (Sec. 4: Table 6) demonstrate significant improvements by employing the relative head count loss in the network.

We directly train our SaCNN from scratch by randomly initializing the network parameters. Because the head count regression is an easy task, we first regress the model on the density map (2); once it converges, we add (3) in the objective loss to jointly train a multi-task network for a few more epochs. We set the density map loss weight as 1 and the relative count loss weight as 0.1.

We note that it will be 1.5× faster for the model to converge if we train a two-scale model first and then finetune SaCNN on it; the performance difference between the two training manners is trivial.

4. Experiments

We shall first briefly introduce three standard datasets and a new created dataset for crowd counting. Afterwards, we evaluate our method on these datasets.

4.1. Datasets

ShanghaiTech dataset [34] consists of 1198 annotated images with a total of 330,165 people with head center annotations. This dataset is split into two parts: PartA and PartB. Refer to Table 1 for the statistics of the dataset: the crowd images are sparser in PartB compared with PartA. Following [34], we use 300 images for training and 182 images for testing in PartA; 400 images for training and 316 images for testing in PartB.

WorldExpo'10 dataset [33] includes 3980 frames, which are from the Shanghai 2010 WorldExpo. 3380 frames are used as training while the rest are taken as test. The test set includes five different scenes and 120 frames in each one. Regions of interest (ROI) are provided in each scene so that crowd counting is only conducted in the ROI in each frame. Some statistics of this dataset can be found in Table 1.

UCF_CC_50 dataset [9] has 50 images with 63974 head annotations in total. The head counts range between 94 and 4543 per image. The small dataset size and large variance in crowd count make it a very challenging dataset. Following [9], we perform 5-fold cross validations to report the average test performance. Some statistics of the dataset is in Table 1.

SmartCity dataset is collected by ourselves. There are in total 50 images collected from ten city scenes including office entrance, sidewalk, atrium, shopping mall *etc.*. Some examples are shown in Fig. 4. They are all high-angle shot for video surveillance. Existing crowd counting datasets consist of images of hundreds or even thousands of pedestrians, and nearly all the images are taken outdoors. We therefore specifically create the dataset that has few pedestrians in images and consists of both outdoor and indoor scenes. Some statistics of this dataset is shown in Table 1: the average number of pedestrians per image is only 7.4 with minimum being 1 and maximum being 14. We use this dataset to test the generalization ability of the proposed framework on very sparse crowd scenes.

Dataset		Resolution	#Images	Min	Max	Avg	Total
ShanghaiTech	PartA	different	482	33	3139	501.4	241,677
	PartB	768 * 1024	716	9	578	123.6	88,488
WorldExpo'10		576 * 720	3980	1	253	50.2	199,923
UCF_CC_50		different	50	94	4543	1279.5	63,974
SmartCity		1920*1080	50	1	14	7.4	369

Table 1. Dataset statistics in this paper. #Images is the number of images; Min, Max and Avg denote the minimum, maximum, and average pedestrian numbers per image, respectively.

Figure 4. Examples for the SmartCity dataset. They consist of both indoor and outdoor scenes with few pedestrians.
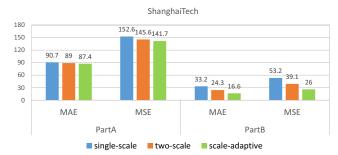

Figure 5. Multi-scale ablation test on ShanghaiTech dataset. The MAE and MSE are clearly decreased from single-scale to two-scale and scale-adaptive model.

4.2. Implementation details and evaluation protocol

Ground truth annotations for each head center in the standard benchmarks are publicly available. Given a training set, we augment it by randomly cropping 9 patches from each image. Each patch is $1/4$ size of the original image. All patches are used to train our model SaCNN. We train the model using the stochastic gradient descent (SGD) optimizer. The learning rate starts from 1e-6 and decays to 1e-8 with multistep policy. Momentum is 0.9 and batch size is 1, we train 250 epochs in total.

We evaluate the performance via the mean absolute error (MAE) and mean square error (MSE) commonly used in previous works [33, 34, 26, 1, 19, 30]:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |F_Y - Y_i|,$$
$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{(F_Y - Y_i)^2}. \quad (4)$$

One can refer to (3) for notations F_Y and Y_i. Small MAE and MSE indicate good performance.

4.3. Results on ShanghaiTech

Ablation study. We report an ablation study to offer the justification of our SaCNN in multi-scale and multi-task setup. Referring to Sec. 3, SaCNN is built from low-level to high-level by gradually adding layers of different scales. Network training is at first focused on the density map loss, af-

ter it converges, the relative count loss is added to continue the training.

Multi-scale ablation test. We train three models separately corresponding to single-scale, two-scale, and three-scale (scale-adaptive) CNNs in Sec. 3. Fig. 5 illustrates their performance. It can be seen that the MAE and MSE are clearly decreased from single-scale to two-scale models; eventually scale-adaptive model reaches the lowest MAE and MSE on both PartA and PartB. There are on average 4 times more people in PartA than in PartB, pedestrian heads are therefore quite small; whilst the feature maps from the deep layers tend to fire on big heads (Sec. 3.2). Thus combining multi-scale outputs does not result in a significant decrease in MAE and MSE on PartA. But still, the results validate our argument that the complementary scale-specific feature maps produce a strong scale-adaptive crowd counter. We use the three-scale model in the following experiments. Notice in this test we only employ the density map loss during training.

Multi-task ablation test. We introduce a relative count loss (rcl) to the above scale-adaptive model to improve the network generalization ability on crowd scenes with few pedestrians. We denote by SaCNN(w/o cl) the scale-adaptive model without count loss and SaCNN(rcl) with relative count loss. It can be seen from Table 2 that, compared to SaCNN(w/o cl), SaCNN(rcl) decreases both MAE and MSE on PartA and PartB.

To justify the relative count loss over the absolute count loss (acl), we train another model SaCNN(acl) following [33], where they train a multi-task network by alter-

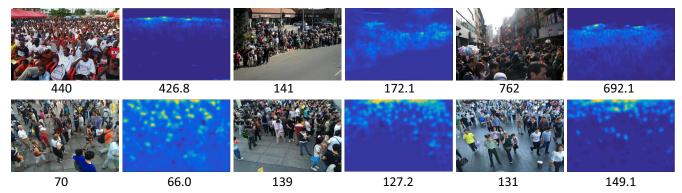| 440 | 426.8 | 141 | 172.1 | 762 | 692.1 |
| 70 | 66.0 | 139 | 127.2 | 131 | 149.1 |

Figure 6. Result on ShanghaiTech dataset. Top: PartA; Bottom: PartB. We present six test images and their estimated density maps on the right. Ground truth and the estimated pedestrian numbers are beneath the real images and the corresponding density maps, respectively.

ShanghaiTech	PartA		PartB	
Method	MAE	MSE	MAE	MSE
SaCNN(w/o cl)	87.4	141.7	16.6	26.0
SaCNN(acl)	102.7	164.9	17.5	26.5
SaCNN(rcl)	**86.8**	**139.2**	**16.2**	**25.8**

Table 2. Multi-task ablation test of SaCNN on ShanghaiTech dataset. cl is short for count loss; acl denotes the absolute count loss while rcl denotes the relative count loss proposed in this paper.

ShanghaiTech	PartA		PartB	
Method	MAE	MSE	MAE	MSE
Zhang *et al.* [33]	181.8	277.7	32.0	49.8
Zhang *et al.* [34]	110.2	173.2	26.4	41.3
Sam *et al.* [26]	90.4	**135.0**	21.6	33.4
SaCNN	**86.8**	139.2	**16.2**	**25.8**

Table 3. Comparison of SaCNN with other state-of-the-art on ShanghaiTech dataset.

nating between training on the density map loss and the absolute count loss. The result in Table. 2 shows that adding the absolute loss however impairs the performance. This is because the absolute count loss varies drastically among crowd images. In the following experiments, we use SaCNN to signify our best model SaCNN(rcl). We shall also show the significant improvement of adding rcl on SmartCity dataset (Sec. 4.6).

Comparison to state-of-the-art. We compare our best model SaCNN with state-of-the-art [33, 34, 26] on both PartA and PartB (Table 3). Our method achieves the best MAE: 86.8 and 16.2 on PartA and PartB. Compared to [26], SaCNN achieves 3.6 point decrease in MAE on PartA and 5.4 point on PartB; 7.6 point decrease in MSE on PartB and a comparable MSE (second best) on PartA. SaCNN requires less computation compared to [26], where they split the test image into patches and employ both a classification and a regression network to test each patch. At test time SaCNN is 2 times faster than [26] (293ms vs. 580ms per image).

Fig. 6 shows some examples on both PartA and PartB.

WorldExpo'10	S1	S2	S3	S4	S5	Avg.
Zhang *et al.* [33]	9.8	14.1	14.3	22.2	3.7	12.9
Zhang *et al.* [34]	3.4	20.6	12.9	13.0	8.1	11.6
Sam *et al.* [26]	4.4	15.7	**10.0**	11.0	5.9	9.4
SaCNN (GT with perspective)	4.4	26.3	16.8	**9.0**	4.9	12.3
SaCNN (GT w/o perspective)	**2.6**	**13.5**	10.6	12.5	**3.3**	**8.5**

Table 4. Comparison of SaCNN with other state-of-the-art on WorldExpo'10 dataset. MAE is reported for each test scene and averaged in the end.

The estimated density maps are visually similar to the crowd distributions in real images. We give the predicted number and the real pedestrian number under the density maps and real images, respectively. The estimated pedestrian numbers are close to the real numbers.

4.4. Results on WorldExpo'10

Referring to [33], training and test are both conducted within the ROI provided for each scene. MAE is reported for each test scene and averaged to evaluate the overall performance. We compare our SaCNN with other state-of-the-art in Table 4. Notations *i.e.* (GT w/o perspective) and (GT with perspective) signify the selection of σ in (1). [33, 34] compute the σ as a ratio of the perspective value at certain pixel. [26] however reports a better result without using perspective values. We provide both results with and w/o perspectives. It can be seen that except on S4, SaCNN (GT with perspective) is inferior to SaCNN (GT w/o perspective) on all the remaining scenes. SaCNN(GT w/o perspective) produces the best MAE on S1, S2 and S5: 2.6, 13.5 and 3.3. The average MAE of SaCNN across scenes outperforms [26] by 0.9 point and reaches the best 8.5.

4.5. Results on UCF_CC_50

We compare our method with seven existing methods on UCF_CC_50 [33, 34, 26, 12, 9, 1, 19] in Table 5. [12] proposed a density regression model to regress the density map rather than the pedestrian number in crowd counting. [9]

UCF_CC_50	MAE	MSE
Lempitsky *et al.* [12]	493.4	487.1
Idrees *et al.* [9]	419.5	541.6
Zhang *et al.* [33]	467.0	498.5
Boominathan *et al.* [1]	452.5	-
Zhang *et al.* [34]	377.6	509.1
Onoro *et al.* [19]	333.7	425.3
Sam *et al.* [26]	318.1	439.2
SaCNN	**314.9**	**424.8**

Table 5. Comparison of SaCNN with other state-of-the-art on UCF_CC_50 dataset.

SmartCity	MAE	MSE
Zhang *et al.* [34]	40.0	46.2
Sam *et al.* [26]	23.4	25.2
SaCNN(w/o cl)	17.8	23.4
SaCNN	**8.6**	**11.6**

Table 6. Comparison of SaCNN with other representative works on SmartCity dataset.

used multi-source features to estimate crowd count while [1] used a multi-column CNN with one column initialized by VGG net [27]. [19] adopted a custom CNN network trained separately on each scale; fully connected layers are used to fuse the maps from each of the CNN trained on a particular scale.

Our method achieves the best MAE 314.9 and MSE 424.8 compared to the state-of-the-art. The smallest MSE indicating the lowest variance of our prediction across the dataset.

4.6. Results on SmartCity

For SmartCity there are too few pedestrians to train a model. We employ the model trained on ShanghaiTech PartB and test it on SmartCity. Crowd scenes in ShanghaiTech PartB are outdoors and relatively sparse while in SmartCity they are both indoor and outdoor scenes with few pedestrians (7.4 on average, see Table. 1 and Fig. 4). Referring to the multi-task ablation test on ShanghaiTech (Sec. 4.3), we test both SaCNN(w/o cl) and SaCNN in Table. 6 to show how much improvement is achieved by SaCNN over SaCNN(w/o cl). By adding the relative count loss in the network, the MAE and MSE clearly drops 9.2 and 11.8 points, respectively. Fig. 8 illustrates some test examples. The density map produced by SaCNN is sparser compared to that of SaCNN(w/o cl); it is more likely to fire on the real pedestrians. While the result of SaCNN(w/o cl) reflects more subtle details of the image which are unrelated to pedestrians (*e.g.* the third row).

In Table 6 we also test [34, 26][2]: both are trained on PartB and perform poorly on SmartCity; the MAE are 40.0 and 23.4, respectively. Our best MAE is 8.6 close to the

---

[2]We used the published model from [34] and trained the model for [26] using their released code.

average pedestrian number 7.4. Adding the relative count loss has significantly improved the network generalization on very sparse crowd scenes.

Overall, our results are not perfect, and we believe this makes the dataset a challenging task for future works.

4.7. Regression v.s. detection

In this section we compare our regressed-based method SaCNN with a detection-based method YOLO9000 [22]. There are no bounding box annotations available for crowd counting datasets, and it is often not feasible to annotate every pedestrian head with a bounding box in dense crowds (see Fig. 7). Thus, we train a pedestrian detector on COCO dataset [15] following YOLO9000 with human annotations available. We also annotate the pedestrian head bounding boxes in COCO and train a head detector. We test both detectors on ShanghaiTech PartA and PartB, as well as SmartCity. The crowd scenes in these datasets vary from very sparse to very dense (see Table 1). Since the crowd images are taken from high angles, detectors trained from COCO might not generalize well on them. We thus always report the better result over the two detectors on crowd images. Results are shown in Table 7. In the detection context, a good detection is measured by the intersection-over-union (IoU) between the detected bounding box and the ground truth bounding box; the threshold for confidence score (cs) is set to be high, *e.g.* $cs > 0.5$. In the crowd counting context, a good prediction is measured via the difference between the estimated count and the ground truth count; having a lower threshold *e.g.* $cs > 0.05$ ends up with a bigger prediction. A bigger prediction however reflects a lower MAE/MSE in the crowd counting as YOLO tends to miss small objects in general in images. We illustrate both results with YOLO9000 $cs > 0.05$ and $cs > 0.5$ in Table 7: our SaCNN is significantly better than YOLO9000 on ShanghaiTech PartA and PartB; but on SmartCity it is a bit inferior to YOLO9000. Overall, we argue that the strength of our regression-based SaCNN is to provide a monolithic approach that performs close to or above the state-of-the-art on a wide range of datasets, from sparse to dense. For the very sparse case a detection-based method might be superior, but it would not perform well in the dense case. It is not clear how to combine a head detector and a density based approach, and this can be our future work.

We illustrate some examples in Fig. 7. Pedestrian crowds in the three samples range from very sparse to very dense. We illustrate the results of both pedestrian and head detectors. Using a small threshold for confidence score produces big prediction, but the bounding boxes are not always accurate (*e.g.* first row: PD - $cs > 0.05$). In general, YOLO9000 is good at detecting big objects in sparse crowds and is bad with small objects in dense crowds. In contrast, SaCNN can generalize very well from very spare to very dense case.

| Image | PD (cs > 0.05) | PD (cs>0.5) | HD (cs>0.05) | HD(cs>0.5) | SaCNN |

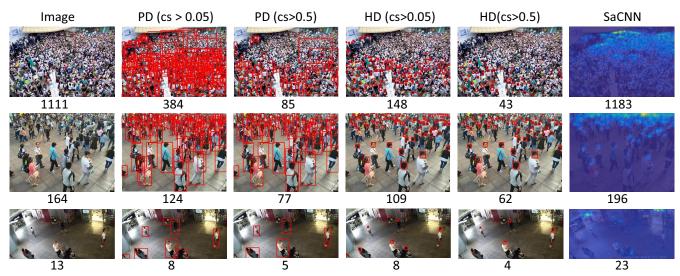| 1111 | 384 | 85 | 148 | 43 | 1183 |
| 164 | 124 | 77 | 109 | 62 | 196 |
| 13 | 8 | 5 | 8 | 4 | 23 |

Figure 7. Comparison between YOLO9000 [22] and SaCNN. PD: pedestrian detector; HD: head detector; cs confidence score. The numbers below the real images (first column) are the ground truth. Numbers below other images are the estimated pedestrian counts.

Dataset	ShanghaiTech PartA		ShanghaiTech PartB		SmartCity	
Measures	MAE	MSE	MAE	MSE	MAE	MSE
YOLO9000 [22] ($cs > 0.05$)	268.2	428.7	46.0	81.6	**3.5**	**4.7**
YOLO9000 [22] ($cs > 0.5$)	366.8	513.6	69.0	108.0	5.0	6.0
SaCNN	**86.8**	**139.2**	**16.2**	**25.8**	8.6	11.6

Table 7. Comparison between YOLO9000 [22] and SaCNN. cs denotes the confidence score. For ShanghaiTech PartA and ParB we report the pedestrian detection results, while for SmartCity we report the head detection result.

| Crowd Images | SaCNN(w/o cl) | SaCNN |

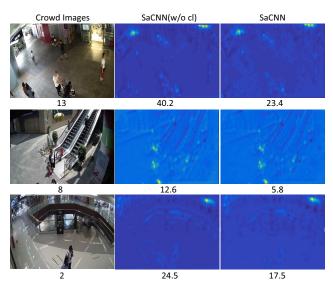| 13 | 40.2 | 23.4 |
| 8 | 12.6 | 5.8 |
| 2 | 24.5 | 17.5 |

Figure 8. Comparison of SaCNN and SaCNN(w/o cl). The numbers below the real images are the ground truth. The second column denotes the density maps produced by SaCNN(w/o cl) while the third column is produced by SaCNN. Numbers below the density maps are the corresponding estimated pedestrian counts.

5. Conclusion

In this paper we propose a scale-adaptive convolutional neural network (SaCNN) to automatically estimate the density maps and pedestrian numbers of crowd images. It concatenates multiple feature maps of different scales to produce a strong scale-adaptive crowd counter for each image; it introduces a multi-task loss including a relative count loss to improve the network generalization on crowd scenes with few pedestrians. The proposed method can easily adapt to pedestrians of different scales and perspectives. Extensive experiments on standard crowd counting benchmarks demonstrate the efficiency and effectiveness of the proposed method over the state-of-the-art.

We know that the pedestrian sizes vary in according to the perspective changes in crowd images; while the multiple feature maps in SaCNN fire on pedestrians of different sizes. In future work, we want to directly embed the perspective information as a weighting layer into SaCNN. It will produce different weights to combine the feature map outputs at every pixel depending on the perspectives.

References

[1] L. Boominathan, S. S. Kruthiventi, and R. V. Babu. Crowd-net: a deep convolutional network for dense crowd counting. In *ACM MM*, 2016. 1, 2, 3, 4, 5, 6, 7

[2] G. J. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *CVPR*, 2006. 2

[3] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *CVPR*, 2008. 1, 2

[4] A. B. Chan and N. Vasconcelos. Bayesian poisson regression for crowd counting. In *ICCV*, 2009. 1, 2

[5] K. Chen, C. C. Loy, S. Gong, and T. Xiang. Feature mining for localised crowd counting. In *BMVC*, 2012. 2

[6] D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, 2012. 2

[7] W. Ge and R. T. Collins. Marked point processes for crowd counting. In *CVPR*, 2009. 1

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[9] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In *CVPR*, 2013. 1, 2, 4, 6, 7

[10] H. Idrees, K. Soomro, and M. Shah. Detecting humans in dense crowds using locally-consistent scale prior and global occlusion reasoning. *TPAMI*, 37(10):1986–1998, 2015. 1

[11] D. Kong, D. Gray, and H. Tao. Counting pedestrians in crowds using viewpoint invariant training. In *BMVC*, 2005. 2

[12] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *NIPS*, 2010. 2, 6, 7

[13] M. Li, Z. Zhang, K. Huang, and T. Tan. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *ICPR*, 2008. 1

[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. 2017. 2

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. 7

[16] B. Liu and N. Vasconcelos. Bayesian model adaptation for crowd counts. In *ICCV*, 2015. 1, 2

[17] A. Marana, L. d. F. Costa, R. Lotufo, and S. Velastin. On the efficacy of texture analysis for crowd monitoring. In *SIBGRAPI*, 1998. 2

[18] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 2

[19] D. Onoro-Rubio and R. J. López-Sastre. Towards perspective-free object counting with deep learning. In *ECCV*, 2016. 1, 2, 5, 6, 7

[20] N. Paragios and V. Ramesh. A mrf-based approach for real-time subway monitoring. In *CVPR*, 2001. 2

[21] V. Rabaud and S. Belongie. Counting crowded moving objects. In *CVPR*, 2006. 2

[22] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017. 7, 8

[23] C. S. Regazzoni and A. Tesei. Distributed data fusion for real-time crowding estimation. *Signal Processing*, 53(1):47–63, 1996. 2

[24] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[25] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Crowd counting using multiple local features. In *DICTA*, 2009. 2

[26] D. B. Sam, S. Surya, and R. V. Babu. Switching convolutional neural network for crowd counting. In *CVPR*, 2017. 1, 2, 3, 4, 5, 6, 7

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2, 3, 7

[28] R. Stewart, M. Andriluka, and A. Y. Ng. End-to-end people detection in crowded scenes. In *CVPR*, 2016. 2

[29] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *IJCV*, 63(2):153–161, 2003. 2

[30] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao. Deep people counting in extremely dense crowds. In *ACM MM*, 2015. 5

[31] M. Wang and X. Wang. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *CVPR*, 2011. 2

[32] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV*, 2005. 2

[33] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*, 2015. 1, 2, 4, 5, 6, 7

[34] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*, 2016. 1, 2, 3, 4, 5, 6, 7

[35] T. Zhao, R. Nevatia, and B. Wu. Segmentation and tracking of multiple humans in crowded environments. *TPAMI*, 30(7):1198–1211, 2008. 1