Crowd Counting via Adversarial Cross-Scale Consistency Pursuit

Zan Shen[1], Yi Xu[1], Bingbing Ni[1], Minsi Wang[1], Jianguo Hu[2], Xiaokang Yang[1]
[1]Shanghai Institute for Advanced Communication and Data Science [2]Minivision
[1]Shanghai Jiao Tong University, Shanghai 200240, China
(sz128ve980, xuyi, nibingbing, xkyang)@sjtu.edu.cn, mswang1994@gmail.com, hujianguo@minivision.cn

Abstract

Crowd counting or density estimation is a challenging task in computer vision due to large scale variations, perspective distortions and serious occlusions, etc. Existing methods generally suffer from two issues: 1) the model averaging effects in multi-scale CNNs induced by the widely adopted ℓ_2 regression loss; and 2) inconsistent estimation across different scaled inputs. To explicitly address these issues, we propose a novel crowd counting (density estimation) framework called Adversarial Cross-Scale Consistency Pursuit (ACSCP). On one hand, a U-net structured generation network is designed to generate density map from input patch, and an adversarial loss is directly employed to shrink the solution onto a realistic *subspace, thus attenuating the blurry effects of density map estimation. On the other hand, we design a novel scale-consistency regularizer which enforces that the sum up of the crowd counts from local patches (i.e., small scale) is coherent with the overall count of their region union (i.e., large scale). The above losses are integrated via a joint training scheme, so as to help boost density estimation performance by further exploring the collaboration between both objectives. Extensive experiments on four benchmarks have well demonstrated the effectiveness of the proposed innovations as well as the superior performance over prior art.*

1. Introduction

With the rapid increase of population of major cities, crowd scene analysis [11, 33] has already become an important security technique in video surveillance [16, 20, 36]. However, generating high-quality crowd density map (crowd count) is a challenging task due to complex illumination, severe occlusions, perspective distortions and diverse distributions of people sizes. Among them, scale variation problem is the major obstacle.

Recent CNN-based works [20, 37, 25] utilize multi-path architectures to address the scale variation issue and have achieved good improvements in crowd density estimation.

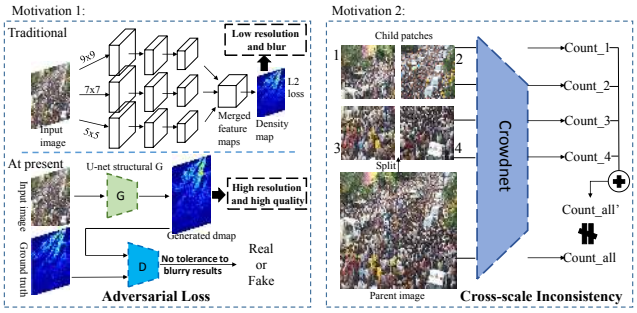

Figure 1. Two motivations of our proposed algorithm: 1) Adversarial loss to "generate" density map for sharper and higher resolution; 2) Cross-scale consistency constraints.

Namely, different sizes of convolutional kernels are applied to the input images to deal with different scaled humans, and the convolution maps from multiple-scale paths are fused to yield the final density estimation. However, most of these methods suffer inherent algorithmic drawbacks. On one hand, only traditional Euclidean loss is employed to optimize these models, which is known to have certain disadvantages [10] such as sensitivity to outliers and image blur. In particular, although different sizes of convolutional kernels are used to extract multi-scale features, each sub-network path attempts to minimize the regression loss independently (i.e., multi-scale model **competition**) and to predict the correct density map for patches with all human scales. As shown in Figure 1 Motivation 1, since each scale-specific sub-network can only work well on its corresponding scale and its performance drop drastically on other scales, it is easy to result in low-quality and blurry results. On the other hand, most existing approaches do not explore the coherence between the estimated density maps from different scales. Namely, the sum up of the crowd counts from local patches (i.e., small scale) does NOT necessarily correspond to the overall count of their region union (i.e., large scale) as shown in Figure 1 Motivation 2. Further, as most algorithms employ sliding window scheme, accumulation of boundary loss of local patches will affect the global crowd count. It is thus demanding to develop a cross-scale consistency regularization scheme which is beneficial for further improving crowd density estimation.

To address these issues, we propose a novel crowd counting framework called Adversarial Cross-Scale Consistency Pursuit Networks (ACSCP). On one hand, inspired by the recent success of GANs in image translation [9], we propose a patch-to-density generation network endowed with an adversarial training loss, to mitigate blurring effect caused by optimization only over traditional Euclidean loss. Further, the proposed multi-scale U-net [24] generation architecture executes a pixel-wise translation from every crowd image pixel to its corresponding density value, which ensures high resolution and high quality density map estimation. On the other hand, a new regularizer is proposed to further enforce cross-scale model calibration and encourage different scale paths to work collaboratively. In particular, our model is made of two complementary density map generators: one takes large scale patch input, and the other takes small scale patch input. We enforces that the sum up of the crowd counts from local patches (i.e., small scale) is coherent with the overall count of their region union (i.e., large scale). The above objectives are integrated via a joint training scheme, so as to help boost density estimation performance by further exploring their collaboration. Extensive experiments on four benchmarks have well demonstrated the effectiveness of the proposed innovations as well as the superior performance over prior art.

2. Related Work

A large number of algorithms have been proposed to tackle crowd counting task in computer vision. Early works estimate the number of pedestrians via head or body detection [32, 18, 31]. Such detection-based methods are limited by severe occlusions in extremely dense crowd scenes. Methods [1, 6, 5, 12, 2, 22, 4] use regressors trained with low-level features (e.g. HOG, SIFT, Fourier Analysis, detections and trajectories) to predict global counts. These methods cannot provide the distribution of crowd, and such low-level features are outperformed by features extracted from CNN [34] which have better and deeper representations.

In recent years, crowd counting has entered the era of deep CNN. A comprehensive survey of recent CNN-based methods for crowd counting can be found in [29]. Wang *et al*. [30] trained a classic Alexnet style CNN model to predict crowd counts. Regrettably, this model has limitation in crowd analysis as it does not provide the estimation of crowd distribution. Zhang *et al*. [34] proposed a deep convolutional neural network for crowd counting which is alternatively regressed with two related learning objectives: crowd count and density map. Such switchable objective-learning helps improve the performance of both objectives. But the application of this method is limited as it requires perspective maps which are not easily available in practice during the process of both training and testing.

Multi-column CNN is employed by [37, 3]. Different CNN columns with varied receptive fields are designed to capture scale variation and perspective, and then features from these columns are fused together by a 1×1 convolutional layer to regress crowd density. Switch-CNN [25] inspired by MCNN [37] proposes a patch-based switching architecture before the crowd patches go into multi-column regressors. The switch-net is trained as a classifier to intelligently choose the most appropriate regressor for a particular input patch, which takes advantage of patch-wise variations in density within a single image. These methods have made great contributions to the progress of crowd counting by deep learning. By using max pooling layers and ℓ_2 loss, they pay more attention to the accuracy of predicted crowd count, and neglect the quality of the regressed density map. As a result, these poor quality maps adversely affect other higher level cognition tasks such as counting and scene recognition which depend on them. The latest research CP-CNN [28] proposes a contextual Pyramid CNNs for incorporating global and local contexts which are obtained by learning various density levels. Contextual information is fused with high-dimensional feature maps extracted from a multi-column CNN by a Fusion-CNN consisting of a set of convolutional and fractionally-strided layers. Both our method and CP-CNN are contemporary works starting to consider the quality of density map. Besides proposing a patch-to-density translation through adversarial training, we further introduce a novel regularizer to enforce cross-scale model calibration and encourage different scale paths to work collaboratively.

3. Methodology

3.1. Density Regression Revisited

As discussed in Section 1, recent state-of-the-art methods [34, 37, 3, 20, 25] dominantly choose L2 based loss function to regress crowd density map. In most cases [37, 3, 20, 28], to deal with human scale changes, multiple convolution paths (sub-networks) with varying sized kernels are fused to yield the final density map prediction. Suppose the network forward computation of scale path i is denoted as S_i, the overall loss function could be expressed as:

$$L = min_{_F} \ \|F\left(S_1, S_2, S_3...\right) - M\|_2^2, \tag{1}$$

where M is the ground truth density map and $F(S_1, S_2, S_3...)$ is output map fused from multiple scale paths. These state-of-the-art methods have two major issues:

1. First, although different sizes of convolutional kernels are utilized to extract multi-scale features [37, 28], (i.e., as each sized kernel is sensitive to different human scales), different scale-paths work in a "**competing way**" rather than

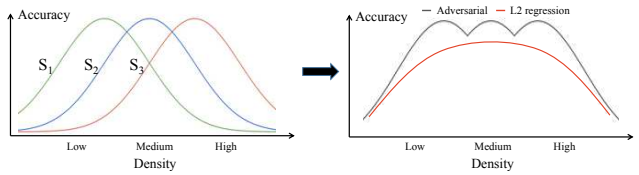

Figure 2. Explainations of inability of L2 regression loss for multi-scale density estimation. Left: three scale-sensitive models; Right: L2 regression (red) leads to model competition, which yields degraded accuracy in all density regions, while adversarial loss (black) encourages model collaboration, thus yields consolided performances in all density regions.

a "**collaborating way**", to deal with human scale variation. Namely, each scale-sensitive network path attempts to minimize the regression loss independently and to predict the correct density map for patches with all human scales. There, it is very easy for each sub-network to generate a blurry estimation due to the model averaging effect (i.e., widely acknowledged to result in low-quality and blurry results especially for image reconstruction tasks [9, 15]). Indeed, this is the inherent drawback/fundamental problem of regression based density map estimation methods, which CANNOT be alleviated by simply applying multi-scale convolution paths. See explainations in Figure 2.

2. Second, there lacks a calibration among various scale-sensitive paths of the multi-scale crowd density estimation network. Namely, as each sub-network behaves differently for input paths with varying human scales, given the exactly same input patch, the output density maps from different sub-networks are different (i.e., sometimes the gap might be very significant). This directly causes unreliable density estimation. That is, if we divide a large parent patch into several non-overlapping small child patches, it is highly possible that the sum of the human counts from all child patches is NOT equal to the direct estimation of the human counts from the parent patch. In other words, the existing multi-scale crowd density estimation network is very sensitive to how we extract local patches. A small change of patch sizes might cause large error of human count estimation.

To deal with these two issues, a novel crowd counting framework called Adversarial Cross-Scale Consistency Pursuit Networks (ACSCP) is proposed. Details are presented in following Sections 3.2, 3.3, 3.4.

3.2. Network Architecture

Figure 3 shows the architecture of our proposed patch-to-density map generation network, which is partly motivated by the recent success of pixel-to-pixel translation architecture [9]. In our method, a generator network G learns an end-to-end mapping from input crowd image patch to its corresponding density map with the same scale. More specific, following [9, 21, 10], a U-net [24] structure is employed for constructing the generator G, as an encoder-decoder structure. To handle scale variation, we employ a structure of two back-to-back encoder-decoder structures, i.e., G_{large} and G_{small}. These two complementary generators cooperate with each other. Generator G_{large} extracts large-scale information, while G_{small} concentrates on small-scale details. For generator G_{large}, eight convolutional layers along with batch normalization layers and LeakyReLU activation layers are stacked in the encoder part acting as feature extraction layers, which are followed by eight deconvolutional layers along with batch normalization layers and ReLU activation layers (except for the last one) in the decoder part. The decoder layer is further connected to a tanh function. Note that the deconvolutional layers are a mirrored version of the foregone convolutional layers. In addition, three dropout layers are added after the first three deconvolutional layers with dropout ratio as 0.5 in order to alleviate overfitting. Skip connections are also added between mirror-symmetry convolutional and deconvolutional layers to help improve the performance and efficiency. G_{small} shares a similar structure with G_{large}. The detailed architecture parameters of generator G_{large} and G_{small} are depicted in Table 1. The input sizes are 240×240 and 120×120 respectively, and the output sizes are the same as the input ones.

Layer	G_{large}	Layer	G_{small}
1	6 x 6 x 64 conv, stride 2	1	4 x 4 x 64 conv, stride 2
2-7	4 x 4 x 64 conv, stride 2	2-6	4 x 4 x 64 conv, stride 2
8	4 x 4 x 64 conv, stride 1	7	4 x 4 x 64 conv, stride 1
9	4 x 4 x 64 decv, stride 1	8	4 x 4 x 64 decv, stride 1
10-15	4 x 4 x 64 decv, stride 2	9-13	4 x 4 x 64 decv, stride 2
16	6 x 6 x 3 decv, stride 2	14	4 x 4 x 3 decv, stride 2

Table 1. Network architectures of G_{large} and G_{small}.

3.3. Density Estimation via Adversarial Pursuit

As pointed out above, using L2 based regression for training the multi-scale path network leads to blurry estimation, because of its average effect. To alleviate this issue and motivated by recent success of generative adversarial networks (GANs) [7, 23, 17, 14, 19], we propose an adversarial loss. The adversarial loss usually involves a Generator G and Discriminator D playing a two-player minimax game: G is trained to generate images to fool D while D is trained to distinguish synthetic images from ground truth. More specifically, in our problem, the adversarial loss of generating crowd density map from image patch is denoted as:

$$L_A(G,D) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{data}(\mathbf{x},\mathbf{y})}[logD(\mathbf{x},\mathbf{y})] + \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[log(1-D(\mathbf{x},G(\mathbf{x})))], \quad (2)$$

where $\mathbf{x}$ denotes a training patch and $\mathbf{y}$ denotes corresponding ground-truth density map. G tries to minimize this ob-

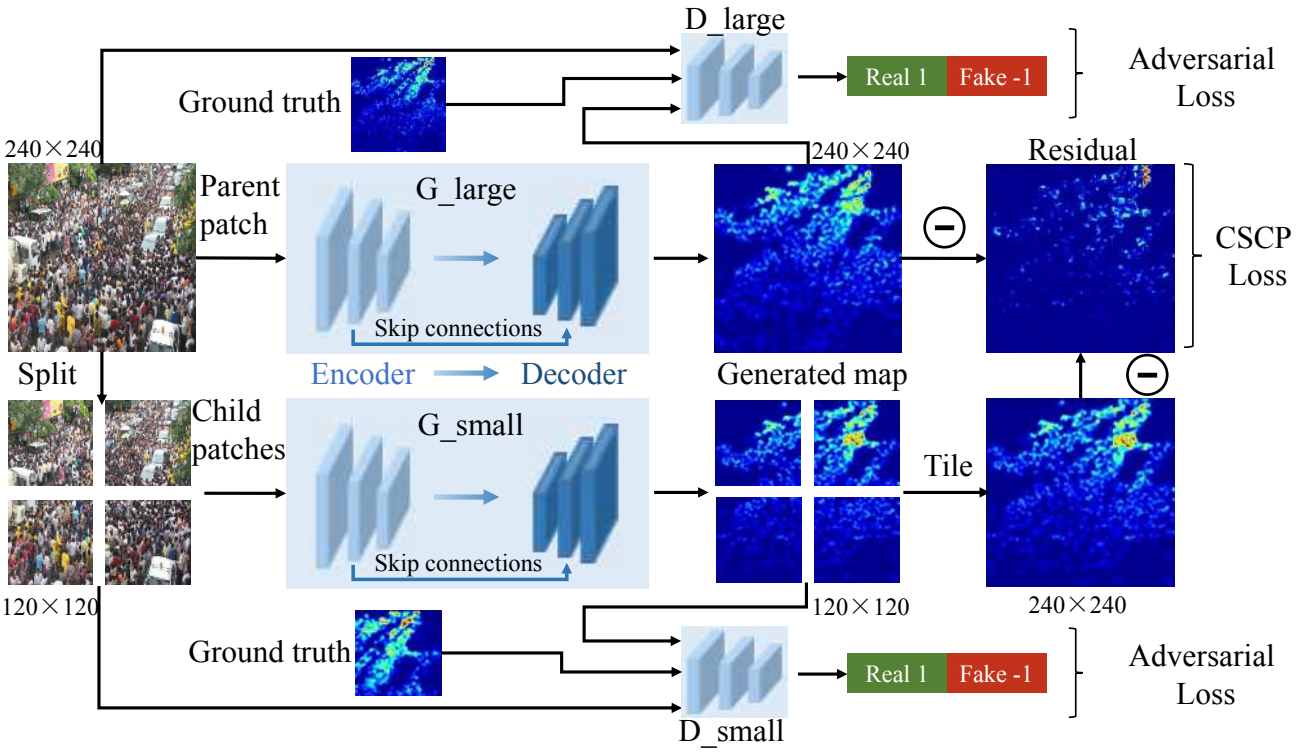

Figure 3. The architecture of the proposed Adversarial Cross-Scale Consistency Pursuit Networks (ACSCP). Two scale G/D jointly trained via cross-scale consistency loss.

jective, whereas D tries to maximize it.

Our discriminative structure is illustrated in Figure 3. The input is concatenated pairs of crowd patches and (generated/ground truth) density maps. Note that a generated density map is exactly the same size as its ground truth map. Five convolutional layers along with batch normalization layers and LeakyReLU activation layers (except for the last one) act as a feature extractor. A tanh function is stacked at the end of these convolutional layers to regress a probabilistic score ranging from -1.0 to 1.0, the value of which indicates whether the input is real (close to 1.0) or fake (close to -1.0). The architecture and network parameters are described as follows: $C(48,4,2) - C(96,4,2) - C(192,4,2) - C(384,4,1) - C(1,4,1) - tanh$, where C is convolutional layer and the numbers inside every brace successively represents the number of filters, filter size and stride. According to our multi-scale generators G_{large} and G_{small}, we have correspondingly two discriminators D_{large}, D_{small}. D_{small} shares the same structure with D_{large}.

The advantage of an adversarial loss over a regression loss is as follows. As the backward gradient of traditional pixel-wise Euclidean loss depends on the magnitude of deviation of the certain pixel, it tends to incentivize a blur when it confronts sharp edges and outliers, thus results in average and blurry maps on image generation problems [13]. An adversarial loss, however, gives each pixel a binary judgement whether it is 'real' or 'fake', and encourages matching the true distribution. It can in principle avoid blur as well as incentivize sharp images since blurry outputs seem unrealistic [7].

For the lack of direct punishment from ground truth, simply using an adversarial loss might sometimes results in aberrant spatial structure even it does not exist in the input label space, as shown in previous works [23, 9]. As suggested by previous works [9, 21, 35], we further include two conventional losses to smooth/improve the solution, which is detailed as follows.

- Euclidean loss: In our model, ℓ_2 loss is chosen to force the estimated density map generated from G to not only fool D but also be close to the ground truth in an L2 sense. Given a $W \times H$ crowd image with C channels, the pixel-wise ℓ_2 loss is defined as:

$$L_E(G) = \frac{1}{C}\sum_{c=1}^{C} \left\| \mathbf{p}^G(c) - \mathbf{p}^{GT}(c) \right\|_2^2, \tag{3}$$

where $\mathbf{p}^G(c)$ represents the pixels in generated density map and $\mathbf{p}^{GT}(c)$ represents the pixels in ground-truth density map, $C = 3$.

- Perceptual loss: Perceptual loss is first introduced by Johnson *et al.* [10] for image transformation and super resolution task. In our model, high-level perceptual features of

the synthetic image and the objective image are respectively extracted from a pre-trained VGG-16 [26] model at layer relu2_2. The basic idea is: by minimizing the perceptual differences between the two images, the synthetic image can be more semantically similar to the objective image. Formally, a perceptual loss is defined as:

$$L_P(G) = \frac{1}{C}\sum_{c=1}^{C}\left\|\mathbf{f}^G(c) - \mathbf{f}^{GT}(c)\right\|_2^2, \tag{4}$$

where $\mathbf{f}^G(c)$ represents the pixels in high level perceptual features of generated density map and $\mathbf{f}^{GT}(c)$ represents the pixels in high level perceptual features of ground-truth density map, $C = 128$.

Therefore, the integrated loss is expressed as:

$$\begin{aligned} L_I = & arg \min_G \max_D L_A(G, D) + \\ & \lambda_e L_E(G) + \lambda_p L_P(G). \end{aligned} \tag{5}$$

Here, λ_e and λ_p are predefined weights for Euclidean loss and perceptual loss. Suggested by previous works [35], we set $\lambda_e = \lambda_p = 150$.

3.4. Cross-Scale Consistency Pursuit Loss

As mentioned earlier, we propose a new regularizer called cross-scale consistency constraint to restrain the cross-scale consistency of parent-child-relationship density maps. In other words, this novel constraint targets at minimizing the residual error between the overall human count estimation of a large image patch and the sum up of counts from its four child patches (i.e., we divide the large patch into four equal sized non-overlapping small patches). This regularization scheme is to address the inability of previous density estimation methods, which easily produce inconsistent results since each of their multiple scale sub-networks is ONLY sensitive to a certain human scale and these sub-models do not work in a collaborative way (i.e., thus induces large cross-scale errors). More specific, at training time, a crowd patch is fed into G_{large} and G_{small} to get the estimated density map P_{parent} and four density maps P_{child}. Then, these four density maps P_{child} are concatenated to get P_{concat} according to P_{parent} . Cross-Scale Consistency Pursuit loss, being defined as the discrepancy/distance between P_{concat} and P_{parent}, is computed by L2-norm in this work. Mathematically, the Cross-Scale Consistency Pursuit loss of a $W \times H$ density map with C channels can be described as follows:

$$L_C(G) = \frac{1}{C}\sum_{c=1}^{C}\left\|\mathbf{p}^{prt}(c) - \mathbf{p}^{cnt}(c)\right\|_2^2 \tag{6}$$

where $\mathbf{p}^{prt}(c)$ represents the pixels in density map P_{parent} and $\mathbf{p}^{cnt}(c)$ represents the pixels in density map P_{concat}, $C = 3$. Via minimizing this regularizer, density estimation gap between parent and child scales is forced to be small. It is worth nothing that if we know the ground-truth human counts for child patches, we might also define ℓ_2 losses for each P_{child}, which may yield similar effect as the proposed cross-scale consistency loss. We MUST emphasize here that in most cases, we are ONLY given the overall human count annotation for the entire image (i.e., without any local annotation such as head locations or ground-truth density maps), therefore ONLY our proposed cross-scale consistency regularizer could be applied (i.e., it does not require density map annotation, and a human count annotation is sufficient). Therefore, our proposed new regularizer is more generally applicable.

Final objective: The above four loss functions are weightedly combined to get the final objective,

$$L_{II} = L_I + \lambda_c L_C(G). \tag{7}$$

Here, λ_c is predefined weight for cross-scale consistency pursuit loss. If λ_c is set to 0, then two generators in our model will be trained independently. In order to determine its value, we have made an experiment on parameter sensitivity in Section 4.4, the λ_c is finally set to 10.

3.5. Density Map for Training

During training and testing, paired crowd image patch and its corresponding ground-truth density map are required. We follow the same scheme as in [34] for preparing ground-truth density maps. As all crowd datasets are given in the form of point annotation at the center of the head of each person, point cloud to density map conversion is required. To this end, Gaussian kernels are applied to match the center (mean) and area (variance) of each person head. The number of Gaussian modals therefore represents the number of people in the image. To deal with head size variations and perspective distortions on datasets that do not provide perspective information, we follow the method proposed by Zhang *et al*. [37] to utilize geometry-adaptive Gaussian kernels to generate density maps.

3.6. Training Details

During training, inputs are image pairs composed of a crowd patch and its corresponding density map. Such image pair is first input to the large-scale sub-network G_{large}, which is then evenly divided into 4 equational image pairs without overlapping and input to the small scale sub-network G_{small}. Both sub-networks are trained jointly. RMSprop optimizer, the learning rate of which is set to 0.00005, is used to update the parameters of our network. We follow the rule of update as: four updates of G_{small} are followed by one of G_{large} in each iteration.

To augment training data, one of the general approaches is to resize the input image pair to a larger size and random-

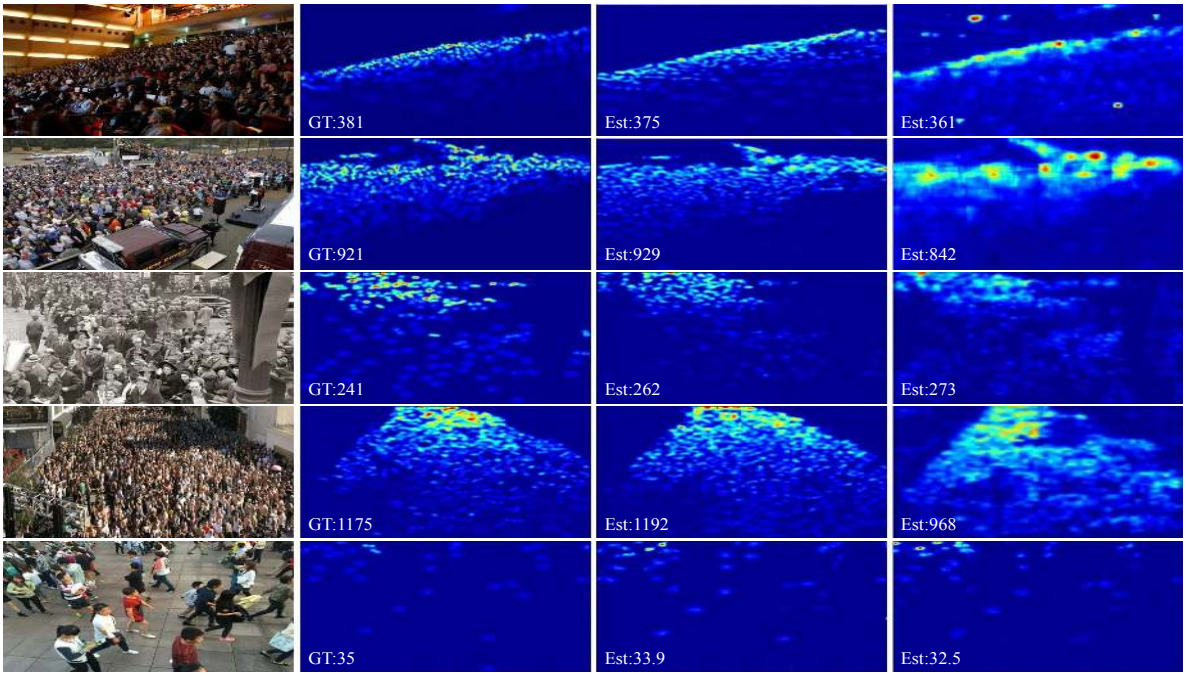

Figure 4. Comparison of estimated density maps. First column: test image; Second column: ground-truth density maps with crowd count; Third column: estimated density maps by our method (ACSCP); Forth column: estimated density maps by MCNN [37].

ly crop a specific sized image pair. However, in our crowd counting task, such data augmentation is not appropriate as image interpolation algorithms such as nearest and bilinear inevitably change the crowd count from a density map. We therefore replace image-resizing with image padding and flip image pairs with a probability of 50% for data augmentation in our experiments.

Our model takes about 300 epochs of training to converge. For the sake of balanced training for both sub-networks, in the first 100 epochs, the predefined weight λ_c in Equation 7 is set as 0 and afterwards adjusted to 10 and the training procedure continues. Finally, the sufficiently trained generator G_{large} is employed to predict density maps for test images. Training and testing of the proposed network are implemented on Torch7 framework.

4. Experiments

4.1. Crowd Counting Datasets

We evaluate our method on four major crowd counting datasets:

ShanghaiTech. ShanghaiTech dataset is created by Zhang *et al*. [37] that consists of 1198 annotated images, including internet images and street view images. Our model is trained and tested on the training and testing sets split by author respectively. To augment the training data, we resize all the images to 720 $\times$ 720 and crop patches from each image. Each patch is 240 $\times$ 240 and is cropped from different locations. Ground-truth density maps are generated by geometry-adaptive Gaussian kernels mentioned in Section 3.5. At test time, a window of size 240 $\times$ 240 slides on a test image to crop patches with 50% overlapping as inputs of the well trained generator. The above steps are similarly applied to the other three datasets.

WorldExpo'10. WorldExpo'10 dataset is created by Zhang *et al*. [34] with 1132 annotated video sequences captured by 108 surveillance cameras from Shanghai 2010 WorldExpo. 3380 frames are treated as training set, and the rest 600 frames are used as test set which are sampled from 5 different scenes, each containing 120 frames. The pedestrians' number in the test scene ranges from 1-220. This dataset provides perspective maps, the value of which represents the number of pixels in the image covering one square meter at realistic location. Different from ShanghaiTech dataset, we choose the crowd density distribution kernel introduced in [34]. To follow the previous methods, only the crowd in ROI regions are taken into consideration.

UCF_CC_50. The UCF_CC_50 dataset is firstly introduced by Idrees *et al*. [8] which is a very challenging dataset composed of 50 annotated crowd images with a large variance in crowd counts and scenes. The crowd counts range from 94 to 4543. We follow the work of [8] and use Five-fold cross-validation to evaluate the proposed method.

UCSD. This dataset consists of 2000 labeled frames with size of 158 $\times$ 238. Ground truth is labeled at the center of every pedestrian and the largest number of people is under 46. The ROI and perspective map are provided as well. In order to cover the pedestrian contour, we choose a bivari-

	Part A		Part B		WorldExpo'10
Objective	MAE	MSE	MAE	MSE	AMAE
L_E	95.8	149.4	24.1	36.4	9.95
L_I	83.2	131.3	18.4	28.8	8.48
L_{II}	**75.7**	**102.7**	**17.2**	**27.4**	**7.5**

Table 2. Comparisons of errors for training with different losses.

ate normalized distribution kernel shaped ellipse to generate density maps. We follow the same train-test setting in [5]: 800 frames from 601 to 1400 are treated as training set and the rest 1200 frames as test set.

To follow previous works, mean absolute error (MAE) and mean squared error (MSE) are used to evaluate the performance of all comparative methods in our experiments.

4.2. Algorithmic Study

In this section, we perform a study to demonstrate the effect of adversarial pursuit and cross-scale consistency regularizer.

Adversarial pursuit takes advantage of adversarial loss, perceptual loss and U-net structured generator to improve the quality of generated density maps as shown in Figure 4. It is noted that our predicted density maps conform to the distribution of crowd much better than MCNN's with less blur and noise. Furthermore, comparative experiments are implemented on ShanghaiTech [37] and WorldExpo'10 [34] datasets in Table 2. It can be observed that training with additional adversarial loss and perceptual loss (i.e. L_I) results in much lower errors than training with Euclidean loss only.

In order to show the effect of the cross-scale consistency regularizer, we plot the mean human count estimation errors between the parent patch and its corresponding sum up from child patches, over all testing patches of various datasets in Figure 5(a). We note that the proposed cross-scale consistency regularizer effectively reduces the estimation gaps from different scales. Figure 5(b) visualizes an example of the residual error maps (i.e., the difference between map of parent patch and the tiled map of its four child patches), which further consolidates the effectiveness. Combined with CSCP loss, the final loss L_{II} achieves the superior results as indicated in Table 2. The performance improvement highlights the benefits of exploiting adversarial training and cross-scale consistency regularizer.

4.3. Comparisons with State-of-the-art

The proposed method is compared with several state-of-the-art approaches on four benchmarks. The results are shown in Table 3, 4, 5, 6. From all tables, we note that our method consistently outperforms previous methods by a good margin. Table 3 and Table 4 indicate comparisons on ShanghaiTech Part_B and WorldExpo'10 datasets, the images of which are closer to the realistic monitoring screens than the others. Our proposed ACSCP obtains quite appreciable improvement over prior art since a large amount of cross-scale inconsistencies occur on these datasets . In addition, Table 5 shows that our approach acquires the best MAE, and comparable MSE among six recent approaches on UCF_CC_50 dataset. This indicates that the proposed approach can also achieve decent results in the case of a

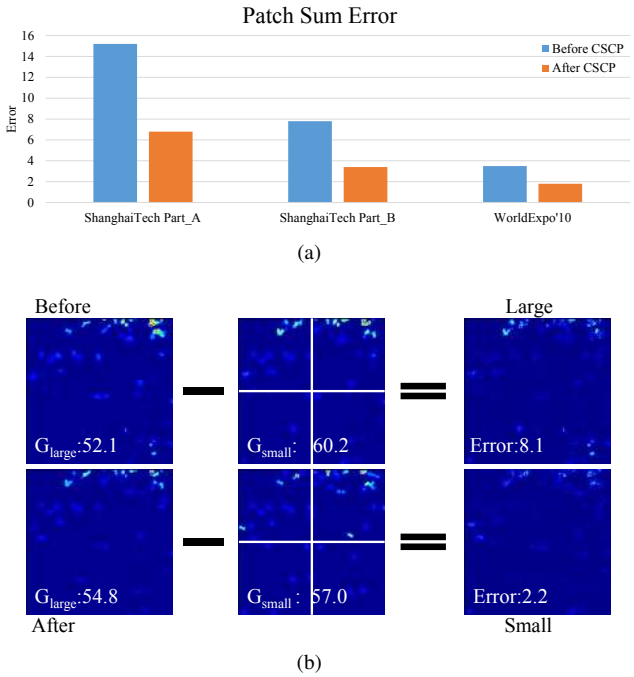

Figure 5. (a) Comparisons of patch sum errors before and after using CSCP loss; (b) Visualization of the effect.

	Part_A		Part_B	
Methods	MAE	MSE	MAE	MSE
Zhang *et al*. [34]	181.8	277.7	32.0	49.8
MCNN [37]	110.2	173.2	26.4	41.3
Switch-CNN [25]	90.4	135.0	21.6	33.4
CP-CNN [28]	**73.6**	106.4	20.1	30.1
ACSCP (ours)	75.7	**102.7**	**17.2**	**27.4**

Table 3. Comparisons on ShanghaiTech dataset [37].

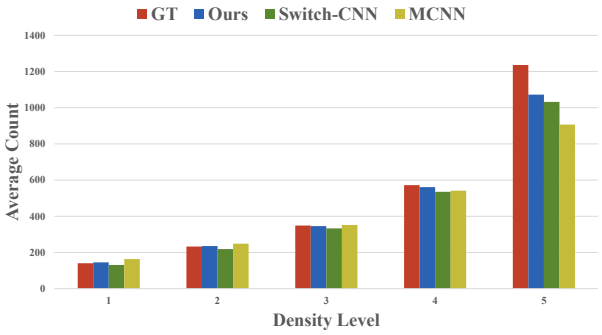

Figure 6. Histogram: average crowd number estimated by different methods on 5 groups split from Part_A according to increasing density level.

Network	Zhang *et al*. [34]	MCNN [37]	Switch-CNN [25]	CP-CNN [28]	**ACSCP (ours)**
Number of parameters	22.5	0.13	15.1	68.4	**5.1**

Table 7. Number of parameters(in millions).

Methods	S1	S2	S3	S4	S5	Ave
Zhang *et al*. [34]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [37]	3.4	20.6	12.9	13.0	8.1	11.6
Switch-CNN [25]	4.4	15.7	10.0	11.0	5.9	9.4
CP-CNN [28]	2.9	14.7	10.5	10.4	5.8	8.9
ACSCP (ours)	**2.8**	**14.05**	**9.6**	**8.1**	**2.9**	**7.5**

Table 4. Comparisons on WorldExpo'10 dataset [34]. Only MAE.

Methods	MAE	MSE
Idrees *et al*. [8]	419.5	541.6
Zhang *et al*. [34]	467.0	498.5
MCNN [37]	377.6	509.1
Cascaded-MTL [27]	322.8	341.4
Switch-CNN [25]	318.1	439.2
CP-CNN [28]	295.8	**320.9**
ACSCP (ours)	**291.0**	404.6

Table 5. Comparisons on UCF_CC_50 dataset [8].

Methods	MAE	MSE
Kernel Ridge Regression [1]	2.16	7.45
Cumulative Attributes [6]	2.07	6.86
Zhang *et al*. [34]	1.60	3.31
MCNN [37]	1.07	1.35
Switch-CNN [25]	1.62	2.10
ACSCP (ours)	**1.04**	**1.35**

Table 6. Comparisons on UCSD dataset [5].

small number of training samples (i.e., UCF_CC_50 has only 50 samples). In Table 6, our ACSCP attains the lowest MAE and MSE errors over other five state-of-the-art methods on UCSD dataset, which states that our algorithm has a good performance on estimating not only images with dense crowd but also images with relatively sparse people (i.e., single scene and maximum count under 46).

Furthermore, a more detailed comparison is implemented on ShanghaiTech Part_A, where test images are divided into five groups according to increasing number of people. It can be observed from the histogram in Figure 6 that our results outperform Switch-CNN and MCNN's over all groups, even in Group 5 (i.e., the group with extremely dense crowd and very few training samples). From Table 3, we note that CP-CNN achieves the lowest MAE on this part. However, it seems unfair that the training process of CP-CNN demands extra priori density-class labels (i.e., global and local density classes) which are NOT directly provided by datasets. Moreover, as the author said, the number of density classes is determined by specific dataset, which is not a general method.

Considering practical applications of crowd counting algorithm, we perform a model complexity study. As shown in Table 7, CP-CNN owns the most parameters, 500 times more than the least MCNN, which limits its applications. In contrast, our algorithm has the second least parameters and it runs at 16 FPS on an Intel Core i7-6700K machine with a TITAN X GPU.

4.4. Parameter λ_c Study

In order to choose the optimum value of λ_c in Equation 7, comparative experiments have been performed on Part_B of ShanghaiTech dataset. As shown in Figure 7, MAE error decreases as the value of λ_c increases, and the lowest error is obtained at $\lambda_c = 10$. After that, the error rises rapidly because the weight of cross-scale consistency loss becomes too significant compared to L_I loss. Thus, we finally assign 10 to λ_c in our experiments.

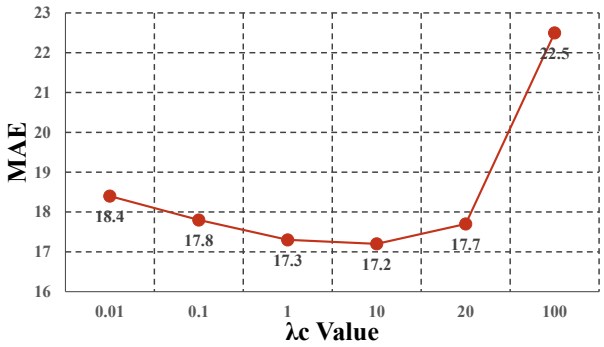

Figure 7. Comparisons of MAE for different λ_c values on ShanghaiTech Part_B [37].

5. Conclusion

In this paper, we propose a GANs-based crowd counting network which takes full advantage of excellent performance of GANs in image generation. To better confine the errors caused by different scales of crowd, we propose a novel regularizer named Cross-Scale Consistency Pursuit which provides a strong regularization constraint on cross-scale crowd density estimation. Extensive experiments indicate that our method achieves the superior performance on four major crowd counting datasets used for evaluation.

6. Acknowledgement

The corresponding author of this paper is Yi Xu (xuyi@sjtu.edu.cn) and Bingbing Ni (nibingbing@sjtu.edu.cn). This work was supported in part by NSFC (61671298, 61502301, U1611461, 61521062), STCSM (17511105401, 18DZ2270700) , 2016YFB1001003 and Chinas Thousand Youth Talents Plan.

References

[1] S. An, W. Liu, and S. Venkatesh. Face recognition using kernel ridge regression. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2007.

[2] A. Bansal and K. S. Venkatesh. People counting in high density crowds from still images. *CoRR*, abs/1507.08445, 2015.

[3] L. Boominathan, S. S. Kruthiventi, and R. V. Babu. Crowdnet: A deep convolutional network for dense crowd counting. In *ACM Multimedia*, 2016.

[4] G. J. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 1:594–601, 2006.

[5] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008.

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

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3:2672–2680, 2014.

[8] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2547–2554, 2013.

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

[10] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[11] J. C. S. J. Junior, S. R. Musse, and C. R. Jung. Crowd analysis using computer vision techniques. *IEEE Signal Processing Magazine*, 27(5):66–77, 2010.

[12] D. Kong, D. Gray, and H. Tao. A viewpoint invariant approach for crowd counting. In *ICPR*, 2006.

[13] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[14] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[15] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[16] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan. Crowded scene analysis: A survey. *IEEE transactions on circuits and systems for video technology*, 25(3):367–386, 2015.

[17] Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative face completion. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] Z. L. Lin and L. S. Davis. Shape-based human detection and segmentation via hierarchical part-template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:604–618, 2010.

[19] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[20] D. Onoro-Rubio and R. J. López-Sastre. Towards perspective-free object counting with deep learning. In *European Conference on Computer Vision*, pages 615–629. Springer, 2016.

[21] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[22] V. Rabaud and S. J. Belongie. Counting crowded moving objects. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 1:705–711, 2006.

[23] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[24] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[25] D. B. Sam, S. Surya, and R. V. Babu. Switching convolutional neural network for crowd counting. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

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

[27] V. A. Sindagi and V. M. Patel. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, pages 1–6. IEEE, 2017.

[28] V. A. Sindagi and V. M. Patel. Generating high-quality crowd density maps using contextual pyramid cnns. *2017 IEEE International Conference on Computer Vision*, 2017.

[29] V. A. Sindagi and V. M. Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 2017.

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

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

[32] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, 1:90–97 Vol. 1, 2005.

[33] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu. Crowd analysis: a survey. *Machine Vision and Applications*, 19(5-6):345–357, 2008.

[34] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 833–841, 2015.

[35] H. Zhang, V. Sindagi, and V. M. Patel. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957*, 2017.

[36] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura. Understanding traffic density from large-scale web camera data. *arXiv preprint arXiv:1703.05868*, 2017.

[37] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column convolutional neural network. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, 2016.