PaDNet: Pan-Density Crowd Counting

Yukun Tian[1], **Yimei Lei**[1], **Junping Zhang**[1], **James Z. Wang**[2]
[1]Department of Computer Science, Fudan University, Shanghai 200082, China
[2]The Pennsylvania State University
{17210240203, 17110240016, jpzhang}@fudan.edu.cn, {jwang}@ist.psu.edu

Abstract

Crowd counting in varying density scenes is a challenging problem in artificial intelligence (AI) and pattern recognition. Recently, deep convolutional neural networks (CNNs) are used to tackle this problem. However, the single-column CNN cannot achieve high accuracy and robustness in diverse density scenes. Meanwhile, multi-column CNNs lack effective way to accurately learn the features of different scales for estimating crowd density. To address these issues, we propose a novel pan-density level deep learning model, named as Pan-Density Network (PaDNet). Specifically, the PaDNet learns multi-scale features by three steps. First, several sub-networks are pre-trained on crowd images with different density-levels. Then, a Scale Reinforcement Net (SRN) is utilized to reinforce the scale features. Finally, a Fusion Net fuses all of the scale features to generate the final density map. Experiments on four crowd counting benchmark datasets, the ShanghaiTech, the UCF_CC_50, the UCSD, and the UCF-QRNF, indicate that the PaDNet achieves the best performance and has high robustness in pan-density crowd counting compared with other state-of-the-art algorithms.

Introduction

Crowd counting has broad applications in public safety, businesses, emergency evacuation, and smart city planning. However, due to problems including perspective distortions, severe occlusion, and high scale density variation, automated pan-density crowd counting (*i.e.* when the density varies from sparse to extremely dense) has been a challenging problem for AI and pattern recognition researchers. Earlier methods (Wu and Nevatia 2005; Wang and Wang 2011) count sparse pedestrians by using a sliding window detector. Regression-based approaches (Chan and Vasconcelos 2009; Ryan et al. 2009) utilize hand-crafted features extracted from local image patches to count sparse crowds. These methods perform ineffectively in dense scenes due to serious occlusions and high scale variations. To handle occlusions, researchers have employed convolutional neural networks (CNNs) based methods to predict a density map which includes important spatial information of crowd images for dense crowd counting.

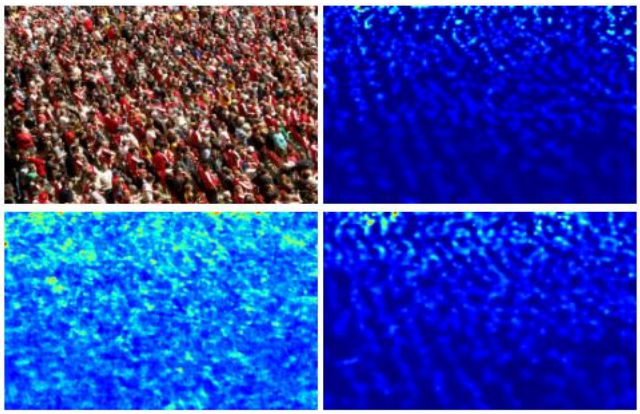

Figure 1: Density map predicted by different multi-column based networks. The first row images are the original and the ground truth. The second row images are MCNN prediction (Zhang et al. 2016) and our PaDNet prediction. Our result is much closer to the ground truth.

Single-column CNNs (Wang et al. 2015; Fu et al. 2015) adopt multiple convolutional layers to extract features, followed by a fully connected layer to predict the count of crowd. Due to the significant variation of crowd density and different spatial distributions of an image, these single-column based methods cannot achieve high accuracy and robustness in varying density scenes. So a number of multi-column network based methods have been developed (Zhang et al. 2016; Zeng et al. 2017). These architectures contain several columns of convolutional neural networks whose filters are in different sizes. Then final predictions are obtained by averaging individual predictions of all deep neural networks. But because these methods only roughly extract features of crowd images, they cannot achieve high prediction performance for pan-density crowd counting. An example can be seen in Figure 1. Li, Zhang, and Chen proposed a single-column based dilated convolutional network, and point out that the existing multi-column network didn't learn different features for each column, resulting in an ineffective branch structure (2018). To overcome these drawbacks, we propose a novel multi-column based network, named as PaDNet, to accurately learn different scale features

for the corresponding sub-networks (Figure 2).

The PaDNet uses different density-level image patches of a dataset to pre-train corresponding sub-networks. The image patches are divided into three classes, low, medium, and high, and the sub-networks are Low-Net, Mid-Net and High-Net, respectively. It is worth noting that we don't use all of the images to directly train the sub-network as in (Zhang et al. 2016; Sindagi and Patel 2017). We believe utilizing high-density images to train Low-Net reduces the ability of identifying the low density features. Therefore, we divide the image patches into three categories. And each specific type of data is employed to tune the respective sub-network for enhancing the ability of the sub-network in recognizing the specific scale. Meanwhile we design a Scale Reinforcement Net (SRN) to further enhance the ability of the sub-network in learning the corresponding scale feature, then use a Fusion-Network to fuse the three feature maps to generate the final density map. With this approach, our proposed PaDNet achieves high accuracy and robustness for pan-density crowd counting.

The main contributions are as follows.

- We propose a novel end-to-end architecture, the PaDNet. To our knowledge, this is the first work to address the problem of pan-density crowd counting.
- By pre-training sub-networks on image pathes with corresponding density-levels and utilizing SRN, the PaDNet accurately learns different scale features.
- Through extensive experiments on four benchmark crowd datasets, the PaDNet obtains the best performance and high robustness in pan-dendsity crowd counting compared with state-of-the-art algorithms.

Related Work

Existing crowd counting algorithms can be roughly placed in two categories (Liu et al. 2018), detection-based methods and regression-based methods.

Detection-based methods

In (Dalal and Triggs 2005; Leibe, Seemann, and Schiele 2005; Tuzel, Porikli, and Meer 2008), the authors proposed to extract some common features from appearance-based crowd images to train the counting classifiers such as Random Forest, Boosting and Naïve Bayes. In the recent decade, more researchers (Ren et al. 2017; Redmon et al. 2016) focused on CNN for detecting pedestrians because deep learning can learn more abundant features related to crowd. But detection-based methods perform ineffectively in dense scenes due to severe occlusions and high scale variations. To overcome these issues, (Felzenszwalb et al. 2010; Lin, Chen, and Chao 2001; Wu and Nevatia 2007) used part-based methods to detect the specific body parts and regions. These detection-based methods are suitable for counting sparse crowd scenes.

Regression-based methods

To address the problem of occlusion, regression-based methods were introduced for crowd counting (Chan and Vasconcelos 2009; Ryan et al. 2009). They learn a mapping from features such as histogram (HOG) and local binary pattern (LBP) extracted from local image patches to their counts. Unlike these approaches, others utilized regression methods to estimate the density map rather than the count of crowd. For example, Lempitsky and Zisserman proposed to learn a linear mapping between local patch features and corresponding object density maps, followed by estimating the total number of pedestrians by integrating over the whole density map (2010). Wang et al. proposed an end-to-end deep CNN for counting people from images in extremely dense crowds (2015). Zhang et al. pre-trained a network for certain scenes, and selected similar training data to fine-tune the pretrained network based on the perspective information and similarity in the density map (2015). Observing that the densities and appearances of image patches are of large variations, Zhang et al. further proposed a multi-column CNN architecture for estimating the density map (2016). In their work, different columns are explicitly designed for learning density variation across different image resolutions. Sindagi and Patel proposed a multi-column based method, named as CP-CNN, to incorporate contextual information of crowd images for achieving lower counting error and better quality density maps (2017). Li, Zhang, and Chen proposed a deeper single-column CSRNet to avoid the issues of previous multi-column networks having ineffective branch structure and inaccurately learning the features of different scales (2018). By refining the VGG-16 network, Sam, Surya, and Babu presented a Switch-CNN to feed the image patches into different column networks (2017). Note that in the prediction phase, the Switch-CNN can only use a single column network for a image patch, without incorporating all sub-networks they have trained. As a result, the density maps predicted by Switch-CNN have some deviation because the densities and appearances of image patches are of high vari-

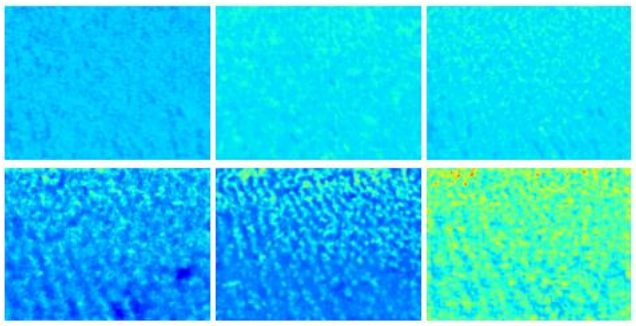

Figure 2: From left to right, there are the feature maps of the Low, Mid and High sub-network. The first row is generated by MCNN (Zhang et al. 2016), through adding a convolutional layer with 1 filters whose size is 1×1 to the last layer of each sub-network of MCNN . And the second row is generated by PadNet. As can be seen, there is only a little difference among these three feature maps of MCNN. But for PaDNet, each sub-network has learned discriminative scale features. The corresponding original image and density map show in Figure 1

ations.

Our Approach

As aforementioned, there are two key issues remaining unsolved: (i) the inappropriate training scheme that uses the whole images to train all column networks, and (ii) the ineffective architecture that cannot properly utilize multi-scale information from each sub-network. To address these, we propose a novel multi-column based framework for crowd counting with pan-density scene, named as PaDNet. The architecture of the PaDNet is illustrated in Figure 4. The fore-end of the PadNet is a modified fine-tuned VGG-16 shared by three sub-networks for extracting features from crowd image patches. And each specific density-level image patch is employed to pre-train the corresponding Low, Mid, and High Nets to enhance their abilities in recognizing the scale features. Then we apply all image patches to fine-tune the whole PaDNet. Note that the Scale Reinforcement Net and the Fusion Net will not be used in the pre-training process. We provide details of the PaDNet below.

Density map generation

The ground truth is generated by blurring the head annotations each with a normalized Gaussian kernel (sum to one). Geometry-adaptive kernel is used for generating the density map as in (Zhang et al. 2016). Geometry-adaptive kernels is defined as:

$$F(x) = \sum_{i=1}^{N} \delta(x - x_i) \times G_{\sigma_i}(x), \text{with } \sigma_i = \beta \overline{d_i} \qquad (1)$$

where x_i is the position of ith head in the ground truth δ, and $\overline{d_i}$ is the average distance of k nearest neighbors. We convolve $\delta(x - x_i)$ with a Gaussian kernel with parameter σ_i. For the ShanghaiTech (Zhang et al. 2016), the UCF_CC_50 (Idrees et al. 2013), and the UCF-QNRF (Idrees et al. 2018) datasets, we set β to 0.3 and k to 5. But because the UCSD dataset (Chan, Liang, and Vasconcelos 2008) does not satisfy the assumptions that the crowd is evenly distributed, we set the σ of the density map to 3.

Data preparation

Sam, Surya, and Babu suggested that to represent the density degree of image patches, the average distance between heads is more effective than the sum of head counting (2017). Therefore, we calculate the density degree of an image as follows:

$$D = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} d_{ij} \qquad (2)$$

where N is the number of people in the image patch. d_{ij} represents the distance between the ith people and the jth nearest neighbor of i. K represents that at most K nearest neighbors are calculated. Intuitively, the smaller the value of D, the denser the crowd.

We resize the training images to 720 $\times$ 720, and crop nine patches from each image. Four of them contain four quarters of the image without overlapping. The remained five patches are randomly cropped from the image. By using horizontal flip for these patches, we can get 18 patches for each image. We calculate the density D for every patch. In order to divide these patches into three classes: low, mid, and high densities, the K-means algorithm is performed to cluster image patches into three classes. To avoid sample imbalance, we continue to crop the patches from the original images to augment patches so that each category will have equivalent number of patches. In the ShanghaiTech, the UCF_CC_50 and the UCF_QNRF dataset, we set K to 5, and 2 for the UCSD dataset.

Algorithm 1 Training

Input: input crowd image patches dataset S
Output: output the parameters Θ_{PaDNet}
Init: Dividing the whole image patches S into K clusters $S_1, S_2 ... S_K$ via K-means clustering algorithm.

```
for i = 1 to epoch_1 do
    for j = 1 to K do
        Training jth sub-network with S_j update Θ_j
        Saving the best state Θ_j of jth sub-network
Loading the best {Θj}_1^K for PaDNet
for i = 1 to epoch_2 do
    Training PaDNet with S update Θ_PaDNet
return Θ_PaDNet
Adam is applied with learning rate at 10^-5 and weight dacay at 10^-4
```

Network architecture

Limited by a lack of training image in each crowd dataset, we choose the first 10 convolutional layers of a fine-tuned VGG-16 for feature extraction to leverage its ability of strong transfer learning. The shortened VGG-16 is shared by three sub-networks: Low-Net, Mid-Net and High-Net, where the three column networks are similar to MCNN (Zhang et al. 2016). The Low-Net utilizes large filters to recognize the sparse density scene, and the High-Net has small filters to recognize the dense scene. Different from MCNN, the Low-Net includes more filters than the High-Net in each layer. The reason is that compared with sparse scenes, the features are relatively simple in dense scenes. Meanwhile, there is no pooling layer in the sub-networks of PaDNet. (Li, Zhang, and Chen 2018) suggested that too many pooling layers reduce the spatial resolution meaning the spatial information of feature map is lost. Therefore, for PaDNet, there are three pooling layers in the shorten VGG-16. Apart from these, the architecture of the PaDNet is deeper and wider than that of MCNN to strengthen the feature extraction ability.

To reinforce the scale feature corresponding to the feature map of each sub-network, we also design a Scale Reinforcement Net (SRN). Three sub-networks generate their respective feature maps, $\boldsymbol{FM_1}$, $\boldsymbol{FM_2}$, and $\boldsymbol{FM_3}$. We concatenate them as input for the SRN. The SRN consists of a Spatial Pyramid Pooling (SPP) layer (He et al. 2015) and a Fully Connected (FC) layer. SPP is used to enhance scale invari-

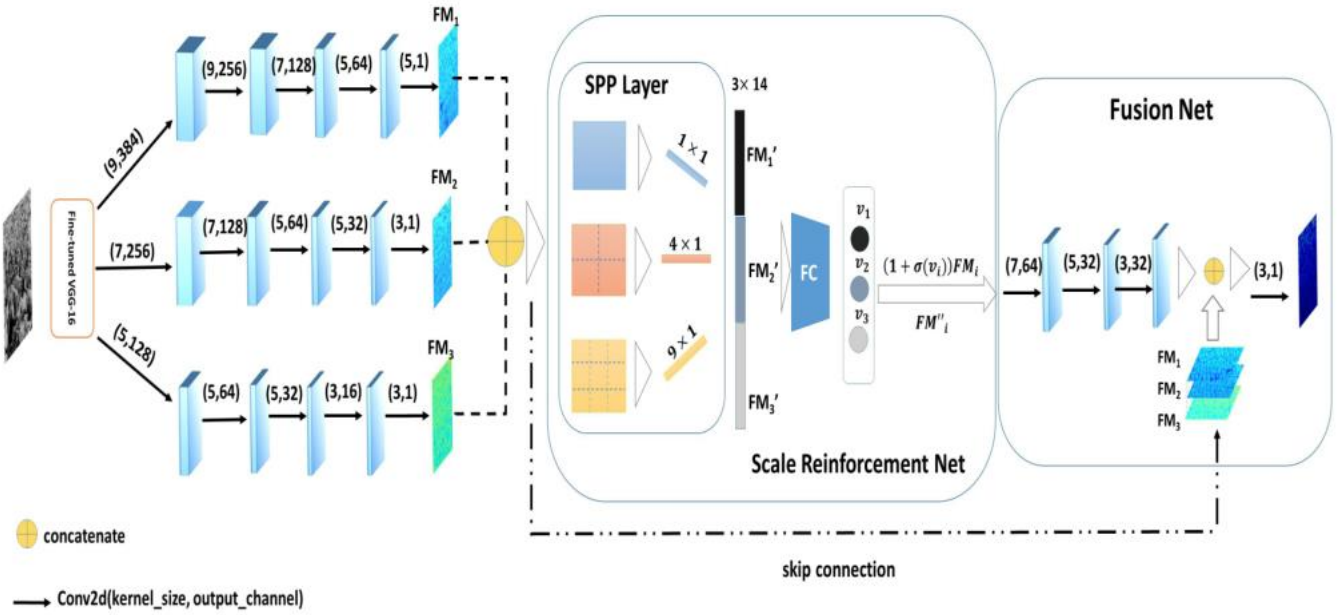

Figure 3: The PaDNet consists of three sub-networks, a Scale Reinforcement Net (SRN), and a Fusion Net. The three sub-networks are used for learning the different scale features by pre-training. Then the SRN is utilized to reinforce the scale features, followed by using a Fusion Net to generate the final density map. Apart from the last of the three sub-network and the Fusion Net, all convolutional layers are followed by BN and ReLU.

ance and to convert the feature maps into a fixed dimension as the input for FC layer. And FC layer is used to classify the image and reinforce the scale feature as follows.

$$\sigma(v_i) = \frac{\exp(v_i)}{\sum_{j=1}^{C} \exp(v_j)} , \tag{3}$$

where v_i is the ith output of FC layer, and C denotes the number of neurons. Therefore, we have:

$$\boldsymbol{FM_i^{''}} = (1 + \sigma(v_i))\boldsymbol{FM_i} , \tag{4}$$

where the number 1 denotes that retaining the original feature of ith sub-network and $\sigma(v_i)$ denotes the reinforcement for the feature. The cross-entropy loss for training the SRN is defined as follows.

$$L_{cls} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}[(\mathrm{y}_i = j)F(X_i, \Theta)] , \tag{5}$$

where N is the number of training images. C represents the number of classes and y_i is the density-level label of ith sample. $F(X_i, \Theta)$ is the prediction of classification.

By concatenating $\boldsymbol{FM_1^{''}}$, $\boldsymbol{FM_2^{''}}$, and $\boldsymbol{FM_3^{''}}$ as input for Fusion Net, we thus incorporate all of the scale features of the sub-networks to generate the final density map. Before the last convolutional layer, further, we add a skip connection to help improve the performance and efficiency. The detail of the training procedure is shown in Algorithm 1, and

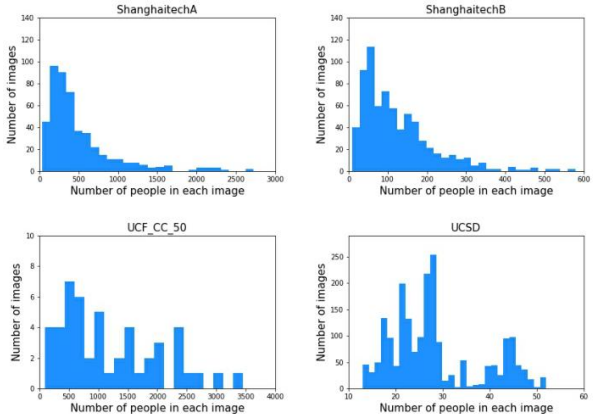

Figure 4: Histograms of the crowd counts in the four datasets.

the loss function for training the PadNet is given as follows.

$$L = L(\Theta) + \lambda L_{cls} , \tag{6}$$

$$L(\Theta) = \frac{1}{N}\sum_{i=1}^{N}\|Z(X_i;\Theta) - Z_i^{GT}\|_2^2 , \tag{7}$$

where λ is the weight factor of L_{cls}, and N is the number training images. $Z(X_i;\Theta)$ is a density map evaluated by the PaDNet and Z_i^{GT} is the ground truth.

Experiments

We now evaluate the PaDNet using four crowd counting benchmark datasets with different crowd densities: the *ShanghaiTech* (Zhang et al. 2016), the *UCSD* (Chan, Liang, and Vasconcelos 2008), the *UCF_CC_50* (Idrees et al. 2013), and the *UCF-QNRF* (Idrees et al. 2018). Figure 4 shows the histogram of crowd counts of these datasets. Furthermore, we compare the PadNet with five state-of-the-art algorithms, the D-ConvNet (Shi et al. 2018), the ACSCP (Shen et al. 2018), the ic-CNN (Ranjan, Le, and Hoai 2018), the SaNet (Cao et al. 2018) and the CSRNet (Li, Zhang, and Chen 2018). We will detail experimental setting and results.

Evaluation metric

In our work, the mean absolute error (MAE) and the mean squared error (MSE) are used as evaluation metrics. Here the MAE is defined as

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|C_{X_i} - C_{X_i}^{GT}| \,, \tag{8}$$

and the MSE is defined as

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(C_{X_i} - C_{X_i}^{GT})^2} \,, \tag{9}$$

where N is the number of test samples, C_{X_i} and $C_{X_i}^{GT}$ are the estimated number of people and the ground truth in the ith image, respectively. Moreover, the MAE and the MSE reflect the algorithm's accuracy and robustness.

	Part_A		Part_B	
Method	MAE	MSE	MAE	MSE
Zhang et al.	181.8	277.7	32.0	49.8
MCNN (2016)	110.2	173.2	26.4	41.3
Switch-CNN (2017)	90.4	135.0	21.6	33.4
CP-CNN (2017)	73.6	106.4	20.1	30.1
TDF-CNN (2018)	97.5	145.1	20.7	32.8
D-ConvNet (2018)	73.5	112.3	18.7	26.0
ACSCP (2018)	75.7	**102.7**	17.2	27.4
ic-CNN (2018)	68.5	116.2	10.7	16.0
CSRNet (2018)	68.2	115.0	10.6	16.0
SaNet (2018)	**67.0**	107.5	**8.4**	**13.6**
PaDNet (Ours)	**63.3**	**95.6**	8.8	**13.5**

Table 1: Comparison on the ShanghaiTech dataset

The ShanghaiTech dataset

This dataset contains 1,198 annotated images from a total of 330,165 people, each of which is annotated at the center of the head. And the dataset is divide into two parts, Part_A and Part_B. Part_A contains 482 images randomly crawled from the Internet. The training set has 300 images and the testing set has 182 images not in the training set. Part_B contains 716 images taken from the busy streets of the metropolitan areas in Shanghai. The training set has 400 images and the testing set has 316 images not in the training set. The density of Part_A is higher than Part_B, and the density varies significantly. We test the performance of the PaDNet on Part_A and Part_B as the other approaches did, and report the best performance in Table 1. The PaDNet achieves the best performance among all approaches. For instance, it has a 5.5% MAE and a 11.1% MSE improvement for the Part_A dataset compared with the second best approach, the SaNet. And the PadNet achieves competitive performance as the SaNet at Part_B dataset.

The UCF_CC_50 dataset

The UCF_CC_50 (Idrees et al. 2013) is an extremely dense crowd dataset. It contains 50 images of different resolutions with counts ranging from 94 to 4,543 with an average of 1,280 individuals in each image. The training set only has 40 images and the testing set only has 10 images. To more accurately verify the performance of the PaDNet, we adopt a 5-fold cross-validation following the standard setting in (Idrees et al. 2013). Experiments shown in Table 2 indicate that the PadNet achieves a 11.8% MAE improvement compared with the SaNet, and 6.9% MSE improvement compared with the CP-CNN. It indicates that the PaDNet is suitable for extremely dense scenes.

Method	MAE	MSE
Zhang et al.	467.0	498.5
MCNN (2016)	377.6	509.1
Switch-CNN (2017)	318.1	439.2
CP-CNN (2017)	295.8	**320.9**
TDF-CNN (2018)	354.7	491.4
D-ConvNet (2018)	288.4	404.7
ACSCP (2018)	291.0	404.6
ic-CNN (2018)	260.9	365.5
CSRNet (2018)	266.1	397.5
SaNet (2018)	**258.4**	334.9
PaDNet (Ours)	**228.0**	**298.7**

Table 2: Comparison on the UCF_CC_50 dataset

The UCSD dataset

The UCSD dataset (Chan, Liang, and Vasconcelos 2008) is a sparse density dataset that is a 2,000-frame video dataset chosen from one surveillance camera on the UCSD campus. The ROI and the perspective map are provided in the dataset. The resolution of each image is 238 $\times$ 158, and the crowd count in each image varies from 11 to 46. As Chan, Liang, and Vasconcelos did, we use frames from 601 to 1400 as the training set and the remained frames for testing. All the frames and density maps are masked with ROI. The results are listed in Table 3. Our method not only achieve superior performance on highly dense crowd dataset, but also on sparse crowd dataset. It has a 19.6% MAE and a 20.2% MSE improvement for the UCSD dataset compared with the second best approach, the SaNet. Meanwhile, the CSRNet is worse than the MCNN, the ACSCP, and the SaNet in predicting sparse crowd count.

Method	MAE	MSE
Zhang et al.	1.60	3.31
MCNN (2016)	1.07	1.35
Switch-CNN (2017)	1.62	2.10
ACSCP (2018)	1.04	1.35
CSRNet (2018)	1.16	1.47
SaNet (2018)	**1.02**	**1.29**
PaDNet (Ours)	**0.82**	**1.03**

Table 3: Comparison on the UCSD dataset

The UCF-QNRF dataset

To further validate our method, we test on the UCF-QNRF dataset (Idrees et al. 2018), which is a new and the largest crowd dataset. The UCF-QNRF (Idrees et al. 2018) contains 1.25 million humans marked with dot annotations and consists of 1,535 dense crowd images with wider variety of scenes containing the most diverse set of viewpoints, densities, and lighting variations. The minimum and the maximum counts are 49 and 12,865, respectively. Meanwhile, the median and the mean counts are 425 and 815.4, respectively. We use 1,201 images as the training set and the remaining 334 images for testing, following (Idrees et al. 2018). Results are shown in Table 4. The PaDNet obtains the lowest MAE performance, and a 18.3% MAE refinement compared with the second lowest approach, *i.e.,* (Idrees et al. 2018).

Method	MAE	MSE
Idrees et al. (2013)	315.0	508.0
CMTL	252.0	514.0
Resnet101	190.0	277.0
Densenet201	163.0	226.0
MCNN (2016)	277.0	426.0
Switch-CNN (2017)	228.0	445.0
Idrees et al. (2018)	**132.0**	**191.0**
PaDNet (Ours)	**107.8**	**187.1**

Table 4: Comparison on the UCF-QNRF dataset

Visualization on density maps

Figure 5 shows some density maps predicted by the CSRNet (Li, Zhang, and Chen 2018), which is the state-of-the-art method based on single-column network, and by our PaDNet. The PaDNet performs significantly better in fine-grained feature extraction. Especially, in extremely dense scene, the density map evaluated by the CSRNet is blurred. The PaDNet has lower error and higher robustness for crowd scenes of different density degrees. The reason is that whether in sparse or dense scenes, both scenes have large variations in the distribution of the crowd. Meanwhile, images of dataset have a wider variety of densities and scales. Multi-column based network can learn the variable density features and is more robust than the single-column based network.

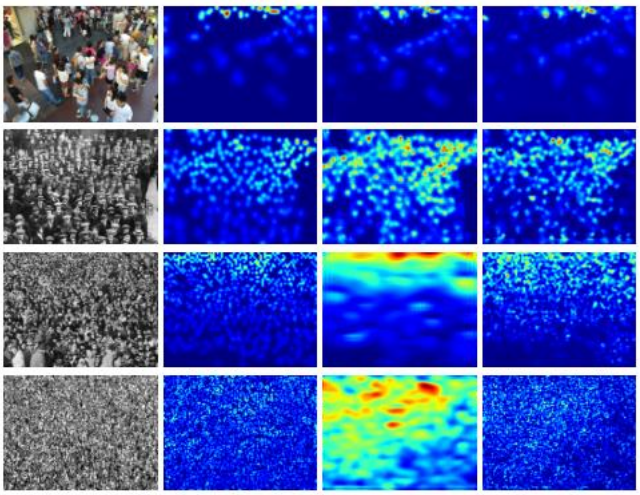

Figure 5: Example experimental results. The images in each row are original crowd image, the ground truth, the result generated by the CSRNet (Li, Zhang, and Chen 2018) (the state-of-the-art method based on single-column dilated convolutional network), and the result generated by our PaDNet, respectively.

Algorithmic study on the ShanghaiTech Part A dataset

The robustness of multi-column network is better than single-column network. Therefore, we mainly explore the effect of learning multi-scale feature with different component of the PaDNet. In Figure 6, we visualize the feature maps which is generated by three sub-networks of different multi-column based networks. PaDNet-A represents directly training the PaDNet without pre-training and the Scale Reinforcement Net (SRN). PaDNet-B represents training the PaDNet with only the SRN. PaDNet-C represents training the PaDNet with only pre-training. In order to compare the ability of sub-network for learning different scale, we normalize the feature maps as follows:

$$\boldsymbol{FM_i^{norm}} = \frac{\boldsymbol{FM_i} - v_{min}}{v_{max} - v_{min}} , \tag{10}$$

$$v_{min} = min(\boldsymbol{FM_1}, \boldsymbol{FM_2}, \boldsymbol{FM_3}) , \tag{11}$$

$$v_{max} = max(\boldsymbol{FM_1}, \boldsymbol{FM_2}, \boldsymbol{FM_3}) . \tag{12}$$

And the corresponding MAE and MSE are listed in Table 5

Method	MAE	MSE
PaDNet-A	72.4	108.5
PaDNet-B	66.1	100.7
PaDNet-C	68.3	105.6
PaDNet	**63.3**	**95.6**

Table 5: Results of different PaDNet components on the ShanghaiTech Part A dataset.

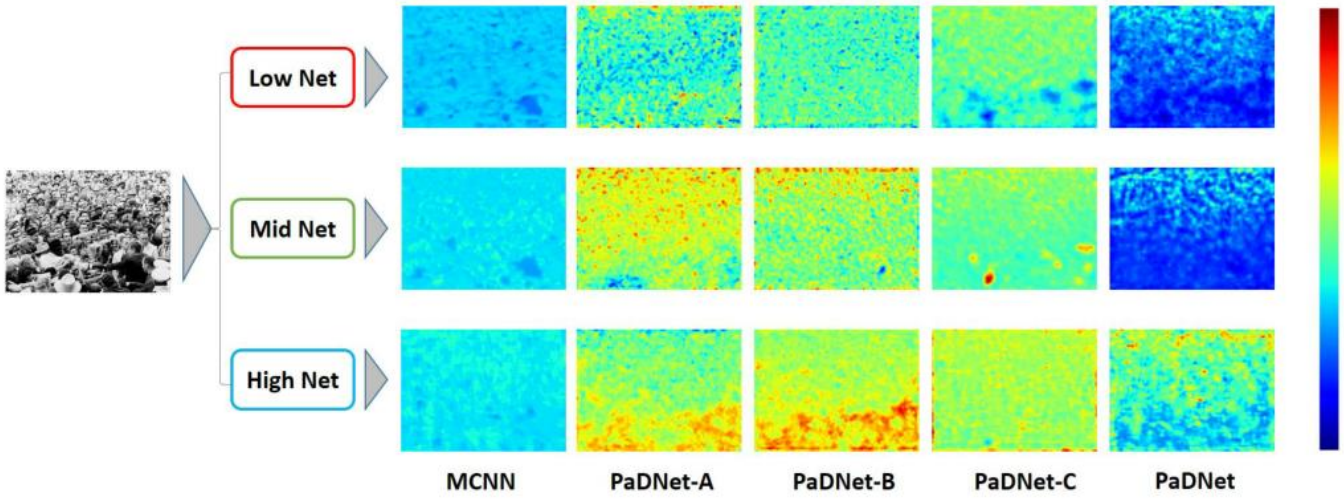

Figure 6: Visualization of feature maps $\boldsymbol{FM_1}$, $\boldsymbol{FM_2}$, and $\boldsymbol{FM_3}$ generated by the three sub-networks with different component of the PaDNet. (a) PaDNet-A represents directly training the PaDNet without pre-training and the Scale Reinforcement Net. (b) PaDNet-B represents training the PaDNet with only the Scale Reinforcement Net. (c) PaDNet-C represents training the PaDNet with only pre-training. The color in the rightest vertical strip indicates the degree of crowd density.

As can be seen from Figure 6, the top region of crowd image is dense, and the bottom region is sparse. The ideal result is that different sub-networks learn different scale features. Specifically, the Low-Net, Mid-Net and High-Net can identify sparse, medium-density and dense crowd. But for the feature maps generated by MCNN, there is little difference among three sub-networks. It performs against the original intention of the MCNN design for learning different features for each column. But in PaDNet-A whose architecture is similar to MCNN, it roughly learns different scale features for each column. This shows that our network is more superior at capturing fine-grained features than that of MCNN. The main difference between the two networks is that the sub-network and the Fusion Net of the PaDNet is deeper and wider than that of MCNN. Thus, it has a strong ability to extract features of the crowd. It is obvious that PaDNet-A are a good choice in estimating density. But PaDNet-A only randomly learns different scale features. The upper half region of Mid Net is denser than High Net, and High Net evaluates the lower region as crowded. These evaluations of PaDNet-A are non-ideal.

Compared with PaDNet-A, PaDNet-B adds the SRN. Due to the influence of the loss L_{cls}, the distinction of the three sub-networks is more obvious than PaDNet-A. The Low-Net of the PaDNet learned the sparse crowd feature, and the Mid-Net evaluates the top region of images as dense. There is an extremely dense region in the High-Net. Although the performance of PaDNet-B is better than that of PaDNet-A, the evaluation of High Net is still biased because the bottom crowd of the original image is sparse, but its evaluation is dense.

Compared with PaDNet-A, PaDNet-C divides the image patches into three categories: low-density, medium-density, and high-density datasets to pre-train the corresponding sub-networks for refining the ability of sub-networks in counting crowd at specific scale. As result, the Low-Net of PaDNet-C identifies the sparse region at bottom of the image and the previous incorrect estimation of High Net of the PaDNet-A that the bottom of crowd is dense is alleviated.

By incorporating the pre-training process of PaDNet-C, the SRN of the PaDNet-B and the architecture of PaDNet-A into PaDNet, finally the biased evaluation is completely eliminated and the feature maps of different networks have obvious distinctions. As expected, the Low-Net, Mid-Net and High-Net can identify sparse, medium-density and dense crowd. Because the PaDNet accurately learns different scale features for each column, it is natural that it achieves higher accuracy over state-of-the-art methods.

Note that the price to pay for such performances is that data preprocessing is sort of complex since we have to use different density-level datasets to pre-train the corresponding sub-networks. Furthermore, its computational cost is pretty high when training PaDNet which is deeper and wider than other networks. For example, it takes about 5 hours to train the PaDNet on ShanghaiTech Part_A dataset with 4 NVIDIA GTX 1080TI GUPs. But in the prediction phase, it only costs 0.11 seconds on average for a image with 1 NVIDIA GTX 1080 TI GUP so that PaDNet can be applied in the real-time scene for crowd counting.

Conclusion

We proposed a novel end-to-end model, the PaDNet, based on the multi-column architecture for pan-density crowd counting. By combining different density level image patches to pre-train corresponding sub-networks and the Scale Reinforcement Net, the PaDNet accurately learns different scale features. Experiments indicate that PadNet attains the lowest predictive errors and higher robustness

for pan-density crowd counting when compared with other state-of-the-art algorithms. To our knowledge, this is the first work for pan-density crowd counting. We will explore a simplify network architecture for pan-density crowd counting in the future.

References

Cao, X.; Wang, Z.; Zhao, Y.; and Su, F. 2018. Scale aggregation network for accurate and efficient crowd counting. In *The European Conference on Computer Vision*.

Chan, A. B., and Vasconcelos, N. 2009. Bayesian poisson regression for crowd counting. In *2009 IEEE International Conference on Computer Vision*, 545–551.

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

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, 886–893.

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; and Ramanan, D. 2010. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(9):1627–1645.

Fu, M.; Xu, P.; Li, X.; Liu, Q.; Ye, M.; and Zhu, C. 2015. Fast crowd density estimation with convolutional neural networks. *Engineering Applications of Artificial Intelligence* 43:81–88.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(9):1904–1916.

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

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

Leibe, B.; Seemann, E.; and Schiele, B. 2005. Pedestrian detection in crowded scenes. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, 878–885.

Lempitsky, V. S., and Zisserman, A. 2010. Learning to count objects in images. In *International Conference on Neural Information Processing Systems*, 1324–1332.

Li, Y.; Zhang, X.; and Chen, D. 2018. CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 1091–1100.

Lin, S.-F.; Chen, J.-Y.; and Chao, H.-X. 2001. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 31(6):645–654.

Liu, J.; Gao, C.; Meng, D.; and Hauptmann, A. G. 2018. DecideNet: Counting varying density crowds through attention guided detection and density estimation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 5197–5206.

Ranjan, V.; Le, H.; and Hoai, M. 2018. Iterative crowd counting. In *The European Conference on Computer Vision*.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 779–788.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6):1137–1149.

Ryan, D.; Denman, S.; Fookes, C.; and Sridharan, S. 2009. Crowd counting using multiple local features. In *2009 Digital Image Computing: Techniques and Applications*, 81–88.

Sam, D. B.; Surya, S.; and Babu, R. V. 2017. Switching convolutional neural network for crowd counting. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 4031–4039.

Shen, Z.; Xu, Y.; Ni, B.; Wang, M.; Hu, J.; and Yang, X. 2018. Crowd counting via adversarial cross-scale consistency pursuit. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 5245–5254.

Shi, Z.; Zhang, L.; Liu, Y.; Cao, X.; Ye, Y.; Cheng, M.-M.; and Zheng, G. 2018. Crowd counting with deep negative correlation learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 5382–5390.

Sindagi, V. A., and Patel, V. M. 2017. Generating high-quality crowd density maps using contextual pyramid cnns. In *2017 IEEE International Conference on Computer Vision*, 1879–1888.

Tuzel, O.; Porikli, F.; and Meer, P. 2008. Pedestrian detection via classification on Riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(10):1713–1727.

Wang, M., and Wang, X. 2011. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *2011 IEEE Conference on Computer Vision and Pattern Recognition*, 3401–3408.

Wang, C.; Zhang, H.; Yang, L.; Liu, S.; and Cao, X. 2015. Deep people counting in extremely dense crowds. In *ACM International Conference on Multimedia*, 1299–1302.

Wu, B., and Nevatia, R. 2005. Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. In *Tenth IEEE International Conference on Computer Vision*, volume 1, 90–97.

Wu, B., and Nevatia, R. 2007. Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *International Journal of Computer Vision* 75(2):247–266.

Zeng, L.; Xu, X.; Cai, B.; Qiu, S.; and Zhang, T. 2017. Multi-scale convolutional neural networks for crowd count-

ing. In *2017 IEEE International Conference on Image Processing*, 465–469.

Zhang, C.; Li, H.; Wang, X.; and Yang, X. 2015. Cross-scene crowd counting via deep convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, 833–841.

Zhang, Y.; Zhou, D.; Chen, S.; Gao, S.; and Ma, Y. 2016. Single-image crowd counting via multi-column convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 589–597.