Crowd Density Estimation Based on Multi-Column Hybrid Convolutional Network

Linlin Guo* and Weimin Zhou
School of Mechanical and Electrical Engineering and Automation, Shanghai University, 99 Shangda Road, Baoshan District, Shanghai, China
*Email: linlinguo@shu.edu.cn

Abstract. This paper studies an accurate counting model for dealing with highly crowded people, multi-column hybrid convolutional neural network model. The model is mainly composed of three parts. The first part uses the first ten layers of VGG-16 convolutional network for image feature extraction. The middle layer is a dilated convolution with three rows of "jaggy" dilation rates, and each row uses the Resnet-block connection method, which is used primarily to perceive human head features of different sizes. Compared with a variety of image up-sampling ways, in the third part of the model, this paper tries to use a combination of bilinear interpolation and convolution to up-sample image features, and research shows that this method effectively reduces the model error. In this experiment, the average absolute error (MAE), mean square error (MSE) and average relative error (MRE) are used as evaluation indicators, and experiments on the ShanghaiTech dataset prove that the network works well.

1. Introduction
There are two traditional methods of counting people. One is based on the detection method [1-3], which uses wavelet HOG, edge and other features extracted from the whole body of pedestrians to train a classifier to detect people density. However, as the population density increases, this method, which is more suitable for sparse populations, is gradually replaced by partial body detection. Part-based detection of the body is to address the problem of occlusion between people. It mainly detects the partial structure of the body. Compared with the detection based on the whole, the effect is slightly improved. Another traditional method is based on regression [4,5], learning a feature to the number of people mapping. Although it can address the problem of occlusion to a certain extent, it makes use of the feature of the whole image but neglects the spatial information of the image.

Deep learning methods can extract high-level features conveniently and efficiently, and has been used by more and more scholars in crowd density research.

In 2016, Zhang et al. proposed the MCNN architecture [6], who were the first to use multi-column convolution to extract image features to accommodate different head sizes. Based on MCNN, Sindagi et al. [7] combined the global and local context information of images to predict population density. In 2018, CSRnet [8] was proposed, which divided the network into front and back ends to generate a population distribution density map with high density. The idea adopted by ic-CNN [9] is to gradually refine the obtained density map from a low-resolution density map to a high-resolution density map. The SANet [10] used a module similar to the Inception architecture [11], in which convolutions with different kernel sizes were used in each convolution layer. Finally, the density map corresponding to the size of the original image was obtained by deconvolution.
The previous work has done well, but with the deepening of the convolution layer, the premature convergence of the model caused by the disappearance of gradient is still a great factor affecting the prediction results. This paper makes the following improvements to this problem. Firstly, the first ten layers convolution of VGG-16 is used to extract features. Secondly, three columns residual network is used to realize feature reuse, which effectively avoids the phenomenon of gradient disappearing. Finally, density regression modules are constructed to output predicted density map.

2. Crowd Density Estimation Based on Multi-Column Hybrid Convolutional Network

2.1. Density Map Generation Algorithm

If the position of a labelled point is $x_i$, we expressed the point as the function $\delta(x - x_i)$. For a crowd image marked with N heads, it can be expressed as the follows:

$$H(x) = \sum_{i=0}^{N} \delta(x - x_i)$$  \hspace{1cm} (1)

The Gaussian function is used to convolve the above formula, and change the position marked as the human head into the density function of the area. In a real scene, especially when the crowd density is high, the position of each $x_i$ is not independent, and the scale of the pixels and the surrounding samples are inconsistent. Therefore, to accurately estimate the population density function, perspective transformation needs to be considered. Assuming that the population is uniformly distributed, the average distance between the position $x_i$ and the nearest $k$ neighbours is reasonably estimated with geometric distortion. That is, the parameter $\sigma$ is determined according to the distance between the $k$ heads in the image.

For each head $x_i$, give the head distance of $k$ neighbours $\{x^i_1, x^i_2, ..., x^i_m\}$, calculate the average distance $\bar{d}_i = \frac{1}{m} \sum_{j=1}^{m} d^i_j$. The location of $x_i$ in the image corresponds to an area, and the population density in this area is similar to $d_i$ in proportion, convolution is performed using an adaptive Gaussian kernel whose variance $\sigma_i$ is variable and $\bar{d}_i$ proportional.

$$F(x) = \sum_{i=0}^{N} \delta(x - x_i) \cdot G_{\sigma_i}(x)$$  \hspace{1cm} (2)

where $\sigma_i(x) = \beta \bar{d}_i$, in this experiment $\beta = 0.3$. Figure 1 shows the real density map obtained by the above algorithm.

![Figure 1. Crowd density map. Left: original image. Right: ground truth.](image)

2.2. Image Processing and Data Enhancement

Due to the mixing of colour images and grayscale images in the ShanghaiTech dataset, and to avoid the impact of sample imbalance, this experiment converts each image of the dataset into grayscale images before model training. The ShanghaiTech dataset has two parts, A and B. There are 300 images in Part A and 400 images in Part B. Since each image in the dataset has different crowd
density, this paper will reduce each image to 9 pieces before training, and the corresponding density map is also reduced to 9 pieces for training.

2.3. Network Model

Figure 2. Multi-column hybrid convolutional network structure diagram.

Figure 2 shows the structure diagram of the multi-column hybrid convolution network. The network consists of three parts, and the generated density map is 1/4 of the original image size. Figure 3 shows the first part of the network, feature extraction module.

Figure 3. Feature extraction module.

A large convolution kernel will make the computational complexity increase dramatically. As shown in Figure 4, this paper uses three column 3 * 3 dilated convolutions in the middle part of the network to adaptively select the characteristics of different receptive field, and obtain the same effect of large convolution kernel under the same receptive field.
In this paper, the maximum pooling of vgg-16 is carried out four times. Therefore, before the output density map of the network, the image features should be sampled to four times of the convolution, so as to ensure the dimension matching of the features in the training process. As shown in Table 1, two bilinear interpolations with a scale-factor of 2 are used instead of a linear interpolation with a scale-factor of 4 to reduce the interference of an up-sampling with a multiple of 4 on the extracted image features.

Table 1. Density regression module.

| Layer type | Kernel number | Kernel size | Stride | Activation function |
|------------|---------------|-------------|--------|---------------------|
| Conv       | 192           | 3*3         | 1      | ReLU                |
| Up-sampling| 192           |             |        |                     |
| Conv       | 96            | 3*3         | 1      | ReLU                |
| Up-sampling| 96            |             |        |                     |
| Conv       | 48            | 3*3         | 1      | ReLU                |
| Conv       | 1             | 1*1         | 1      | ReLU                |

In addition, in the entire model, four times of maximum pooling and two up-sampling with scale-factor of 2 are used, so the final output predicted density map is a quarter of the original image. The original image is resized to an integer multiple of 4 to ensure the matching of parameter dimensions, and the real density map is resized to 1/4 of the original.

3. Dilated Convolution and Bilinear Interpolation

3.1. Dilated Convolution

Dilated convolution is to inject holes into the standard convolution kernel, which can expand the receptive field. In general, in the process of image feature extraction, small targets are more easily reflected in the feature map which is closer to the front. Therefore, we use dilation convolution to extract the target after a series of convolutions, which increases the receptive field and appropriately reduces the parameters. The schematic diagram of hole convolution is shown in Figure 5.
rate = 1 rate = 2 rate = 3

**Figure 5.** Dilated convolution. The dilation rates from left to right are 1, 2 and 3.

Dilated convolution also has some problems, because of the existence of the convolution kernel interval, it means that not all the inputs participate in the calculation, and the overall characteristic map reflects a kind of discontinuity of the convolution centre point. As shown in Figure 6, since the three dilation rates are consistent, the calculation centre of the convolution will show a network-like outward expansion, and some points will not become the centre of the calculation. In this paper, we set the dilation rate of the continuously arranged dilated convolution to "jaggy", which is shown in Figure 7.

**Figure 6.** Three continuous dilated convolutions with dilation rate = 2. The dark mark is the convolution centre involved in the calculation, and the thickness of the colour indicates times.

**Figure 7.** Dilated convolution with dilation rate "jaggy". The dilation rates of three consecutive dilated convolutions from left to right are 1, 2, and 3, respectively.

3.2. **Bilinear Interpolation**

In deep learning, bilinear interpolation and transposed convolution are commonly used in up-sampling. The difference between the two is that bilinear interpolation does not require learning parameters, while transposed convolution requires learning parameters. And transposed convolution is prone to uneven overlap. In principle, neural networks can avoid this situation by carefully learning the weights, but in practice, it is difficult for neural networks to completely avoid this situation. There are two reasonable methods to solve the situation of the “chessboard effect” resulted from transposed
convolution. The first one ensures that the size of the convolution kernel of transposed convolution can be divisible by the stride to avoid the overlap problem. But the stride and kernel size of the transposed convolution are 3 and 2 respectively in this paper. Obviously, the first method is not feasible. The second method is to use interpolation up-sampling and convolution to convert the image from low resolution to high resolution. Figure 8 shows the schematic diagram of bilinear interpolation.

![Bilinear interpolation diagram](image)

Figure 8. Bilinear interpolation diagram. Principle: $Q_{12}, Q_{22}, Q_{11}, Q_{21}$ are four known points. The point to be interpolated is point $P$. First, interpolate the two points $R_1, R_2$ in the x-axis direction, and then interpolate point $P$ according to $R_1, R_2$.

4. Experiment

4.1. Loss Function Definition
The loss function of error backpropagation is defined as the follows:

$$L(\theta) = \frac{1}{2N} \sum_{i=1}^{N} \| F(X_i; \theta) - F_i \|^2$$  \hspace{1cm} (3)

$\theta$ is the network learning parameter. The density map output of the i-th image is $F(X_i; \theta)$. The label density map of the i-th image is $F_i$. The total number of training images is N.

4.2. Evaluation Index
This paper uses MAE and MSE as evaluation indicators. Using MRE (mean relative error) to emphasize the importance of prediction results to each image, and to reflect the accuracy of the prediction intuitively. These indicators are defined as follows:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |x_i - x_i'|$$  \hspace{1cm} (4)

$$\text{MSE} = \left( \frac{1}{N} \sum_{i=1}^{N} |x_i - x_i'|^2 \right)^{1/2}$$  \hspace{1cm} (5)

$$\text{MRE} = \frac{1}{N} \sum_{i=1}^{N} \frac{|x_i - x_i'|}{x_i}$$  \hspace{1cm} (6)

The amount of test images is N. The predicted number and true amount of people of the i-th image is $x_i'$ and $x_i$ respectively.

4.3. Experimental Design and Result Analysis

| Name       | Parameter                                      |
|------------|-----------------------------------------------|
| System     | LINUX64 Ubuntu16.04                           |
| Frame      | Pytorch                                       |

Table 2. Experimental environment.
Language | Python
--- | ---
CPU | Intel® Core™ i5-8300H CPU @2.30GHz × 8
GPU | GeForce GTX 1060/PCle/SSE2
RAM | 16GB

The environment of this experiment is shown in Table 2. The dataset used in this experiment is ShanghaiTech dataset, which was proposed by Zhang et al. [6] in 2016.

### Table 3. ShanghaiTech dataset experimental results.

|         | Part_A |          | Part_B |          |
|---------|--------|----------|--------|----------|
|         | MAE    | MSE      | MRE    | MAE      | MSE      | MRE    |
| Zhang et al. [12] | 181.8  | 277.7    |        | 32.0     | 49.8     |        |
| MCNN [6]    | 110.2  | 173.5    | 0.379  | 26.4     | 41.3     | 0.242  |
| Marsden et al. [13] | 126.5  | 173.5    | 0.379  | 23.8     | 33.1     | 0.242  |
| Sindagi et al. [7] | 101.3  | 152.4    | 0.379  | 20.0     | 31.1     | 0.242  |
| Peng et al. [14] | 97.3   | 147.8    | 0.379  | 25.5     | 40.2     | 0.242  |
| This paper  | 89.60  | 142.04   | 0.256  | 18.25    | 41.32    | 0.221  |

The model proposed in this paper performs well on Part_A and Part_B in Table 3. Compared with MCNN [6], the average relative error MRE is improved by 32.5% and 8.7% respectively, the MAE is improved by 18.7% and 30.9% respectively, and the MRE is improved by 18.1% in Part_A. The experimental results on the ShanghaiTech dataset are as follows. Figure 9 shows the experimental results on part_A where the crowd density is relatively crowded, and Figure 10 shows the results on part_B where the crowd density is relatively sparse.

**Figure 9.** Experimental results of Part_A. Left column: original image. Middle column: predicted density map. Right column: true density map.
Figure 10. Experimental results of Part_B. Left column: original image. Middle column: predicted density map. Right column: true density map.

5. Conclusion

This paper is inspired by the crowd density estimation SCLnet [15] network architecture, and improved on this basis. The network consists of three parts, feature extraction module, improved residual module and density regression module. The original images of the network can be input in various sizes to ensure the integrity of the images. In the part of the improved residual module, feature fusion is carried out to realize feature reuse, which improves the problem of serious loss of image features due to the depth of the model. At the back end of the model, up-sampling is performed to restore the image size. This paper verifies the effectiveness of this network through experiments. Compared with other networks, the error has been improved, and this model has robust scalability and is easy to train. However, this model is still insufficient in transfer learning capabilities. Therefore, subsequent research work will expand the dataset in a targeted manner, combine the datasets with sparse population density and dense population density, and introduce transfer learning to continue to carry out the network structure optimization.

References

[1] Dollár P, Wojek C, Schiele B and Perona P 2012 Pedestrian detection: an evaluation of the state-of-the-art IEEE Transactions on Pattern Analysis and Machine Intelligence 34 743–61

[2] Enzweiler M and Gavrila D M 2009 Monocular pedestrian detection: survey and experiments IEEE Transactions on Pattern Analysis and Machine Intelligence 31 2179–95

[3] Tuzel O, Porikli F and Meer P 2008 Pedestrian detection via classification on Riemannian manifolds IEEE Transactions on Pattern Analysis and Machine Intelligence 30 1713–27

[4] Ma J and Kockelman K 2006 Bayesian multivariate poisson regression for models of injury count, by severity Transportation Research Record: Journal of the Transportation Research Board 1950 24–34

[5] Saleem M S, Khan M J, Khurshid K and Hanif M S 2019 Crowd density estimation in still images using multiple local features and boosting regression ensemble Neural Computing and Application 2019 1-10

[6] Zhang Y, Zhou D, Chen S, Gao S and Ma Y 2016 Single-image crowd counting via multi-column convolutional neural network 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Las Vegas, NV, USA, 6/27/2016 - 6/30/2016) ([Place of publication not identified]: IEEE) pp 589–97

[7] Sindagi V A and Patel V M 2017 Generating high-quality crowd density maps using contextual pyramid CNNs 2017 IEEE International Conference on Computer Vision: ICCV 2017:
proceedings: 22-29 October 2017, Venice, Italy 2017 IEEE International Conference on Computer Vision (ICCV) (Venice, 10/22/2017 - 10/29/2017) (Piscataway, NJ: IEEE) pp 1879–88

[8] Li Y, Zhang X and Chen D 2018 CSRNet: dilated convolutional neural networks for understanding the highly congested scenes 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition: Proceedings: 18-22 June 2018, Salt Lake City, Utah 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Salt Lake City, UT, 6/18/2018 - 6/23/2018)

[9] Ranjan V, Le H and Hoai M 2018 Iterative crowd counting Computer Vision - ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018: proceedings / Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, Yair Weiss (eds.) (LNCS sublibrary: SL6 - Image Processing, Computer Vision, Pattern Recognition, and Graphics 11205-11220) ed V Ferrari (Cham, Switzerland: Springer) pp 278–93

[10] Cao X, Wang Z, Zhao Y and Su F 2018 Scale aggregation network for accurate and efficient crowd counting Computer Vision - ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018: proceedings / Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, Yair Weiss (eds.) (LNCS sublibrary: SL6 - Image Processing, Computer Vision, Pattern Recognition, and Graphics 11205-11220) ed V Ferrari (Cham, Switzerland: Springer) pp 757–73

[11] Szegedy C, Ioffe S, Vanhoucke V and et al 2016 Inception-v4, inception-resnet and the impact of residual connections on learning 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) arXiv:1602.07261v2

[12] Zhang C, Li H, Wang X and Yang X 2015 - 2015 Cross-scene crowd counting via deep convolutional neural networks 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Boston, MA, USA, 6/7/2015 - 6/12/2015) (IEEE) pp 833–41

[13] Marsden M, McGuinness K, Little S and E. O’Connor N 2017 Fully convolutional crowd counting on highly congested scenes VISAPP: Porto, Portugal, February 27-March 1, 2017 International Conference on Computer Vision Theory and Applications (Porto, Portugal, 2/27/2017 - 3/1/2017) (VISIGRAPP 2017 proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications : Porto, Portugal, February 27-March 1, 2017 / sponsored by INSTICC - Institute for Systems and Technologies of Information, Control and Communication ; volume 4) ed F Imai et al ([Setúbal]: SCITEPRESS - Science and Technology Publications, Lda) pp 27–33

[14] Peng S, Fang Z, Gao Y and et al 2018 Crowd counting based on multi-scale full convolutional network feature fusion Journal of Wuhan University (Science Edition) 2018 249-254

[15] Shunzhou W, Yao L, Tianfei Z, Huijun D, Lihua L and Lin Zhang 2020 SCLNet: spatial context learning network for congested crowd counting Neurocomputing 2020 404