Crowd counting model based on CNN and Transformer

Zhiyu Liu[†]*
Shanghai Jiao Tong University
Shanghai,China
lizhyuxi@sjtu.edu.cn

Yongqing Qu†
Chongqing University
Chongqing, China
20204050@cqu.edu.cn

[†]These authors contribute equally to this work

***Abstract*—With the improvement of industrialization and modernization, the frequent stampede accidents in various places caused by the increasing urban population during holidays have attracted people's attention. The excellent crowd counting algorithm can help calculate the distribution of the crowd density in the area, which is widely used in video surveillance and public security. Therefore, the research based on crowd counting model is of great significance. Methods: This paper proposes an innovative model CNN-Transformer network (CNnet), which combines the convolutional neural networks (CNN) which can adapt to different human head sizes with the Encoder of Transformer which has greater capability of capturing global feature. Then the density map generated by CTnet is integrated to get the number of people. Compared with the single CNN structure, the model achieves better accuracy on several datasets. According to the experimental results, the combination of CNN and Transformer can indeed make use of the advantages of both to achieve the improvement of the model. However because of the fact that the current series binding method is relatively simple and cannot combine the advantages of the two to the greatest extent, the experimental results of CTnet have not made a huge breakthrough, and future studies are expected to explore a better combination way.**

Keywords— crowd counting, CNN, Transformer

I. INTRODUCTION

Crowd counting aims to calculate the number, density or distribution of a crowd in a scene or video. With the advancement of modernization, the urban population increases rapidly. When the crowd is highly concentrated, if the management is not proper, the crowded crowd is prone to stampede accidents and other malignant social events.

The crowd counting system can help to carry out real-time statistics on the number, distribution, or density of people in relevant places, timely detect crowd crowding and abnormal behavior, and give early warning, so as to take measures to alleviate the occurrence of tragedy can be avoided.

In addition, the crowd-counting algorithm is also highly transferable. Other target counting fields, such as bacteria and cell counting and car counting on the road, can be regarded as extended applications of the crowd-counting algorithm.

Therefore, the research on crowd-counting algorithms is of important practical significance and application value.

The original crowd-counting techniques extracted pedestrian features through computer vision methods, and then directly estimated the number or distribution of crowds in the image or video through object detection or regression. However, this method cannot extract more detailed abstract features from the input, and the actual accuracy gradually fails to meet the required demand.

In recent years, because of the great improvement of deep learning technology, researchers tried to explore new crowd-counting methods from the angle of convolutional neural networks (CNN) and proposed classical crowd-counting such as MCNN [1]. Multi-column deep neural networks (MCNN) map the input to its crowd density map and then calculates the number of the crowd through integral. Due to the use of a multi-column structure, It corresponds to three kinds of nuclei with different receptive field sizes, so the model can adapt to the large variety of human head sizes, thus improving the accuracy of the model [1].

Although CNN has reached great accuracy in the task of crowd counting with its good local feature-capturing ability, its difficulty in capturing global features has also led researchers at home and abroad to consider new improvements. Self-attention module in natural language processing was introduced into the field of computer vision several years ago. The introduction of a vision transformer (ViT) [2] makes up for the defects of CNN in capturing global features. Crowd models based on Transformer have also been proposed successively, such as TransCrowd [3], CCTrans [4], etc.

However, few models combine the two models and try to further improve the performance of the model with their advantages.

Therefore, this paper proposes a combined model of transformer and CNN based on the existing classical model. Feature maps are extracted from the input graph lines through CNN and then introduced into the Transformer encoder. Finally, crowd counting is predicted through one or more fully connected layers. This network architecture can not only use CNN to extract image features but also use Transformer to realize global context modeling.

In this paper, the internal structure details of CNN and Transformer will be determined by experimental testing on the counting datasets of shanghai-tech and other mainstream datasets. In the end, this paper will also compare CNN +

Transformer model with classic models that only use CNN or Transformer.

II. Ease of Use Methodology

A. *Overview of our network architecture*

Our proposed crowd-counting method uses a multi-column convolutional neural network (MCNN) in conjunction with the vision transformer (VIT) model [1]. We attempt to leverage this network configuration, known as CTnet, to lessen the effect of significant fluctuations brought on by various camera angles and picture resolutions on crowd-counting performance. In the method fully supervised learning is used, which requires labels of each head in the training sample at the position level because generally, it is easier to get better performance compared with weak supervision [5].

In addition, related work like [1] has indicated that rather than the output crowd's total size, its density map can preserve more information and has better performance for inputs with widely fluctuating perspective effects [1]. For these reasons, our model receives images as input and outputs the crowd density maps. Then we compare it to the ground truth density maps to calculate the loss. As Figure 1 shows, CTnet has two modules: the MCNN module (MCM) and the Vision Transformer module (VITM).

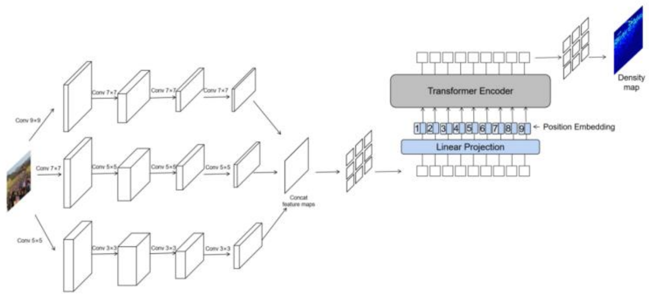
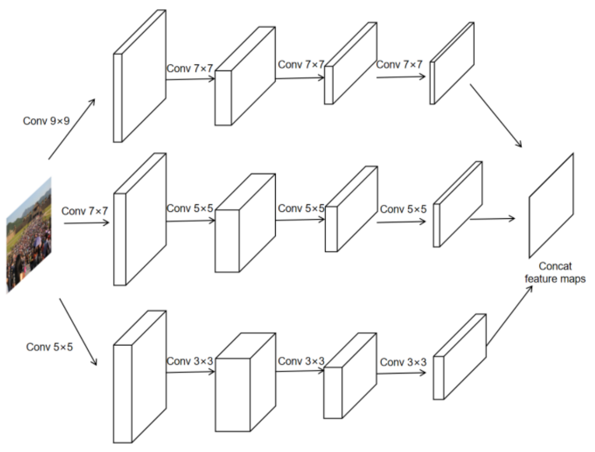


Fig. 1. Structure of CTnet.

(Photo credit: original)

B. *MCNN module*

The first part of our proposed method involves using MCNN architecture to extract features from the input images. MCNN is composed of multiple columns, each with the same network structure (i.e., conv-pooling-conv-pooling) but with different receptive field sizes, to capture features at different scales [1]. Specifically, we set three columns with receptive field sizes (small, medium, and large). Each column consists of four convolutional layers. Figure 2 illustrates the structure of MCM, which comes from the original MCNN paper [1].



Structure of MCNN[1].

The column(small) contains the filters 5*5, 3*3, 3*3 and 3*3. The column(large) contains the filters 7*7, 5*5, 5*5 and 5*5. The column(large) contains the filters 9*9, 7*7, 7*7 and 7*7. When an input image is given, it is sent to these three columns of the first convolutional layer, processed respectively and simultaneously with the filters 9*9, 7*7, and 5*5. After each convolutional layer, instead of Dropout, BN is done for the regularization. BN was a method introduced by Ioffe and Szegedy [6], proved to have the ability to train the model with saturating nonlinearities and to be more tolerant to increased training rates than Dropout. As deep learning has advanced, BN has gradually replaced Dropout in modern convolutional architectures. We prefer it for the following two reasons:

The regularization effect of Dropout in convolutional layers is limited, for the reason that the convolution layer has fewer training parameters, compared with the fully connected layer. The seventh reference [7] demonstrates the advantages of using dropout after batch normalization layers when the batch size is large (256 samples or more) and the dropout rate is low (0.125). (similar to the findings in [8]). Also, it makes the assumption that dropout failed in [6] because a tiny batch size was used during testing. In our work, the batch size is only 2, so dropout is not used. After BN, the ReLU activation function is done, replacing all the negative values with zero. Between two convolutional layers, 2*2 max pooling is used.

Then the image matrix is passed through the next convolutional layer for further feature extraction, with the filters 7*7, 5*5, and 3*3 in different columns. This combination of the convolutional layer, BN layer, and ReLU function is performed 3 more times, the first two with pooling layers and the last one without. At last, we obtain three feature maps with the same size but different resolutions, then we concatenate them along the channel axis to obtain a combined feature map, using learnable weights. To feed the feature maps obtained by the MCNN into the ViT, we do not perform the Fully connected layer (or 1*1 convolutional layer), instead, we directly pass the input of the fourth convolutional layer into the ViT module.

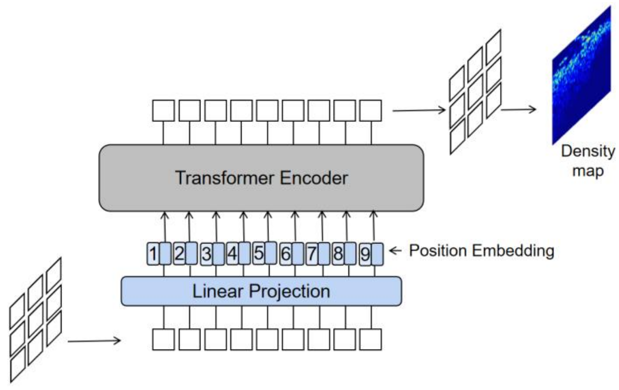
C. *Vision Transformer module*

MCNN, the architecture along with feature fusion approaches to a regress density map, have been proven to be effective in solving the huge scale variations induced in dense crowds. However, since the CNN structure itself can only obtain local features and is not capable to obtain the connection between features and global context information, this paper considers introducing the Transformer structure after the MCNN structure to help the model obtain global information better and improve model performance.

In 2020, the Google team proposed Vision Transformer (ViT) and then succeeded to use it for image classification. Although it is not the first paper to apply a transformer to visual tasks, its model is simple with good effect. The bigger the model is, the better performance it achieves. Since then, the transformer has become a milestone in the application of CV and triggered subsequent relevant research.

Therefore, this paper uses the architecture of ViT for reference and forms a series mode with MCNN to jointly extract human head features. Figure 3 shows how ViT works in CTnet.

Fig. 2. Structure of TRM.



(Photo credit: original)

In terms of the original ViT structure, this paper makes the following adjustments:

(1) The input was changed from the original map to the output of the previous MCNN.

(2) According to the requirements of crowd counting tasks, the classification part of the ViT structure was removed, and the original sequence was directly output through the Encoder structure. Finally, the final output density map was spliced.

The Transformer application part of our model is distributed next. In the Transformer part of the model, the output of MCNN is first divided into multiple patches (16x16), and then each patch is projected into Transformer as a vector of fixed length. The succeeding encoder functions exactly the same as the original Transformer did. Figure 4 illustrates the structure of the Encoder.

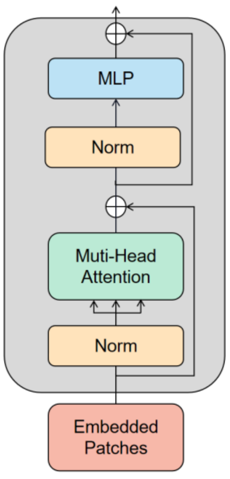


Fig. 3. Structure of Encoder[2].

Following the above flowchart, a Transformer block can be divided as follow:

(1) Patch embedding: The picture is divided into a patch of fixed size. The size of the patch in this paper is 16x16. For example, if the input size is 256x192 and the image is divided into patches of fixed size, and the patch size is 16x16, each image will generate 256x192/16x16=192 patches, that is, the input sequence length is 192. Using patch embedding, we turned a visual problem into a seq2seq problem.

(2) Transformer encoder: Input the patch sequence obtained in the previous step into the encoder. Layer Normalization's output dimensions remain unchanged. If there is only one head, the dimension of qkv is the same as that of the input. If there are n heads, the dimension of q k v is equal to that of the input divided by n, and there are n groups of q k v. Finally, the output of n groups of q k v is spliced together, and the output dimension is still the same as that of the input. Finally, after a Layer Normalization, the dimensions remain the same.

(3) Merge into the final predicted density map: Concatenate the sequence of the previous step into the final output.

So far, this is a demonstration of the principle of the Transformer block of our model. After passing the Transformer structure, the final output size is still the same as the intermediate output of MCNN, so as to ensure that the size mismatch will not occur in the loss function.

D. *Implementation details of model*

1) Loss function

We use MSE (Mean Squared Error) L2 loss to evaluate the performance. It is defined as:

$$L(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\|D(X_i;\theta) - D_i\|_2^2 \qquad (1)$$

where L represents the loss parameterized by learnable parameters θ. N is the number of training images. D_i is the ground truth density map of input X_i and $D(X_i;\theta)$ is the corresponding estimation.

In the formula of MSE, we use 2N instead of N, because it is much easier to calculate the derivative of the loss function later on. We have compared MSE with RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error).

In our work, the total number of people is not large, so the squared difference between to be too large; in other words, there is no clear difference between MSE and RMSE (Root mean squared error). While compared with MAE L1 loss, MSE L2 loss was proved in the ninth reference [9] to have the following advantages:

MSE or RMSE can better illustrate the error distribution. MSE is typically better at highlighting differences in model performance since it gives larger weight to the outliers and adverse situations. Set error = true value – predicted value. If error > 1, then MSE will further increase the error. MSE avoid the use of absolute value, which is seldom used and very undesirable in many mathematical calculations, because it could be challenging to determine the gradient or sensitivity of the MAE with regard to particular model parameters.

2) Ground truth density map

In our work, density maps of the dataset are used to train the model, so the method used to convert images labeled with head positions into density maps of people is important. The geometry-adaptive kernel technique is typically used for datasets with highly congested scenes. If a point is labeled in position x_i, it can be represented as a delta function $\delta(x - x_i)$. Gneralized in this way, an image with N points labeled is $\sum_{i=1}^{N} \delta(x - x_i)$.

In order to make density function continuous, $\delta(x - x_i)$ is convolved by G_{μ_i,σ_i} (Gaussian kernel), with parameter μ_i (standard deviation) and σ_i (kernel size). The Gaussian kernel size is a variance σi proportional to $\overline{d_i}$, which represent the average distance from x_i to its k closest neighbor.

$$F(x) = \sum_{i=1}^{N} \delta(x - x_i) * G_{\mu_i,\sigma_i}, with\ \sigma_i = \beta\overline{d_i} \qquad (2)$$

where F(x) represents the density map. N is the number of annotated heads. In accordance with [1], we configure k=3, beta=0.3 and μi= 15 constant. We also fix σi (standard deviation) for the low-density datasets.

3) Training details

We train our proposed method on the Shanghai-Tech dataset and evaluated its performance on several datasets. Adam optimizer is used for training, with a learning rate of 1e-5 and momentum of 0.9. All the procedures are conducted on a computing system equipped with a GeForce GTX 1660Ti GPU and a RAM of 16 GB with a 64-bit operating system. We write the codes in Python 3.8 version, using Windows 10 Home as the operating system.

III. EXPERIMENT

To evaluate the effectiveness, we assess our CTnet on two datasets: ShanghaiTech and UCF_CC_50. And then we compared it with original MCNN.

A. Dataset

1) Shanghai-tech

There are two components in Shanghaitech dataset [1]: Part_A and Part_B. Part A contains 241,677 annotated points with a range from 33 to 3,139 individuals in each image. There are 182 testing images and 300 training images in part A, and the resolution varies greatly. Part B has 716 annotated images totaling 88,488 individuals with a range from 9 to 578 people per image. It consists of 316 testing images and 400 training images. Every image has a 768 x 1024 resolution.

2) UCF_CC_50

There are 50 images in the UCF CC 50 collection [10] with various crowd densities. It has 63,974 head annotations, and the resolution varies. The average head count per image is 1279, with head counts ranging from 94 to 4543.

B. Evaluation metric

In keeping with previous studies, the eleven and twelve reference [11, 12], we use MAE and MSE metrics to evaluate our method:

$$MAE = \frac{1}{N}\sum_{1}^{N}|z_i - \widehat{z_i}| \qquad (3)$$

$$MSE = \sqrt{\frac{1}{N}\sum_{1}^{N}(z_i - \widehat{z_i})^2} \qquad (4)$$

where z_i, $\widehat{z_i}$ represent the ground truth and estimated count of the i-th input, respectively. N is the number of test images. Generally, MSE and MAE represent the robustness and accuracy of the estimations, respectively.

C. Comparisons with original MCNN

The outcomes of our method in comparison to the original MCNN are displayed in Table 1. On both two datasets, our proposed method achieved better performance.

On Shanghai-Tech Part A dataset, our method achieved an MAE of 102.5 and an MSE of 156.3, while on Part B they are respectively 21.2 and 32.5. On UCF_CC_50 dataset, it achieved an MAE of 102.5 and an MSE of 156.3.

TABLE I. COMPARATIVE RESULTS

Method	Publications	ShanghaiTechA		ShanghaiTechB		UCF_CC_50	
		MAE	MSE	MAE	MSE	MAE	MSE
MCNN [1]	CVPR 16	110.2	173.2	26.4	41.3	377.6	509.1
Ctnet(ours)	——	102.5	156.3	21.2	32.5	362.7	467.6

MCNN uses three columns with different filter sizes to make the features adaptive to different head sizes. However, its receptive fields are still fixed and not learnable, which means it cannot be changed and learned depending on different scenes. In this respect, our model, adding self-attention mechanisms to MCNN, is more generalizable, which may be one of the reasons for its better performance on both datasets.

IV. Conclusion

In this paper, we propose an innovative crowd-counting model, CTnet, for fully supervised crowd counting in images by combining CNN with Transformer. As far as we know, CNN, as a model structure that can effectively extract local human head features, is widely used in the field of crowd counting. The analysis shows that the attention mechanism is promising in global feature capture. Therefore, we adopt the multi-perception wild CNN model that can adapt to different human head sizes, combined in series with Transformer Encoder which is more efficient. In this way, the advantages of both can be combined effectively to achieve the improvement of model performance.

A large number of experiments on challenging datasets show that CTnet achieves better counting performance than the convolutional neural network alone. There are still some deficiencies in the current research. The combination of CNN and Transform was simple in series, and other more complex combinations were not attempted. In the future, we plan to use the architecture interspersed with self-attention in CNN for fully supervised counting tasks and extend it to downstream counting tasks based on the dynamic video.

References

[1] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao and Yi Ma, "Single-image crowd counting via multi-column convolutional neural network," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.589-597, 2016.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” arXiv.org, 03-Jun-2021.

[3] Tian, Ye, et al. “CCTrans: Simplifying and Improving Crowd Counting with Transformer.” ArXiv.org, 29 Sept. 2021, https://arxiv.org/abs/2109.14483.

[4] D. Liang, X. Chen, W. Xu, Y. Zhou, and X. Bai, “TransCrowd: Weakly-supervised crowd counting with Transformers,” arXiv.org, 08-Sep-2022. [Online]. Available: https://arxiv.org/abs/2104.09116. [Accessed: 08-Mar-2023].

[5] Yinjie Lei, Yan Liu, Pingping Zhang and Lingqiao Liu, "Towards using count-level weak supervision for crowd counting," Pattern Recognition, vol.109, 2021.

[6] Ioffe S, Sergey and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," International conference on machine learning. pmlr, pp.448-456, 2015.

[7] Ping Luo, Xinjiang Wang, Wenqi Shao and Zhanglin Peng, "Towards understanding regularization in batch normalization," arXiv preprint arXiv:1809.00846, 2018.

[8] Xiang Li, Shuo Chen, Xiaolin Hu and Jian Yang, "Understanding the disharmony between dropout and batch normalization by variance shift," Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp.2682-2690, 2019.

[9] Tianfeng Chai and Roland R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature," Geoscientific model development 7.3, pp.1247-1250, 2014.

[10] Haroon Idrees, Imran Saleemi, Cody Seibert and Mubarak Shah, "Multi-source multi-scale counting in extremely dense crowd images," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.2547-2554, 2013.

[11] Tota, Karunya, and Haroon Idrees, "Counting in dense crowds using deep features," Proc.CRCV, pp.1-4, 2015.

[12] Y. Lyu, Z. Yang, H. Liang, B. Zhang, M. Ge, R. Liu, Z. Zhang, and H. Yang, “Artificial Intelligence (AI) assisted Fatigue Fracture Recognition based on morphing and fully convolutional networks,” Fatigue &amp; Fracture of Engineering Materials &amp; Structures, vol. 45, pp. 1690–1702, 2021.

[13] Cong Zhang, Hongsheng Li, Xiaogang Wang and Xiaokang Yang, "Cross-scene crowd counting via deep convolutional neural Networks, " Proceedings of the IEEE conference on computer vision and pattern recognition, pp.833-841, 2015.