Segmentation Assisted U-shaped Multi-scale Transformer for Crowd Counting

Yifei Qian[1]
yq1@st-andrews.ac.uk
Liangfei Zhang[2]
lz36@st-andrews.ac.uk
Xiaopeng Hong[3]
hongxiaopeng@ieee.org
Carl R.Donovan[1]
crd2@st-andrews.ac.uk
Ognjen Arandjelović[2]
oa7@st-andrews.ac.uk

[1] School of Mathematics and Statistics
University of St Andrews
Fife, UK

[2] School of Computer Science
University of St Andrews
Fife, UK

[3] Harbin Institute of Technology
Harbin, P.R.China

Abstract

Automated crowd counting has made remarkable progress recently in computer vision thanks to the development of CNNs. However, this application area has run into bottlenecks since CNNs, by their nature, are limited by locally attentive receptive fields and are incapable of modelling larger-scale dependencies. To address this problem, we introduce a multi-scale transformer-based crowd-counting network, termed Crowd U-Transformer (CUT) which extracts and aggregates semantic and spatial features from multiple levels. In this design, we use crowd segmentation as an attention module to gain fine-grained features. Also, we propose a loss function that better focuses on the counting performance in the foreground area. Experimental results on four widely used benchmarks are presented and our method shows state-of-the-art performances.

1 Introduction

Crowd counting, which refers to the automated counting of a multitude of people in an image or video, is a challenging task in the field of computer vision. Considering the ubiquity of digital imagery and increasing frequency of large gatherings of people, the analysis of crowded scenes has become highly practically important. Accurate estimation of their size would be beneficial in various applications, including urban planning, disease control, traffic surveillance, wildlife monitoring and disaster management [34].

However, the practical use of crowd counting is currently limited since the accuracy of automated counting falls short of many application requirements. The primary difficulties are *large scale variations of objects* and *cluttered scene layouts*. The mainstream methods rely on Convolutional Neural Networks (CNNs) to predict a density map for an input image, and then the total count is obtained by subsequent integration over that map. While

CNNs are equipped with the intrinsic inductive bias [52], and hence are good at modelling local feature structures - they are however disadvantaged by receptive field limitations and a poor ability to capture long-range dependencies, which have both proved important in crowd counting [14]. Researchers proposed some methods to mitigate these problems, such as the multi-scale mechanism [24, 55] and auxiliary task learning [29], but these do not fully resolve the problems.

Transformers have recently become an increasingly popular architecture in computer vision community. Its global self-attention mechanism allows capture of long-range dependencies and the mapping a global receptive field – exactly what CNNs lack. Therefore, a number of transformer based methods have shown promising performances in traditional vision tasks like classification [9, 49], detection [2] and segmentation [57].

The advantages of transformers complement the limitations of CNNs, hence in this paper, we propose a multi-scale transformer based network for improved automated crowd counting. Our architectural basis is formed by the pyramid vision transformer (PvT) [6, 49] and we follow the U-shaped design to aggregate features from different levels. Moreover, we use a *Segmentation As Attention Module (SAAM)* that leverages both the focus provided by the crowd segmentation task and the global self-attention from the transformer layers, to gain fine-grained features with abundant semantic information. Our work explores the potential of combining transformer structure with traditional pseudo-density maps for crowd counting tasks. The proposed model achieves the state-of-the-art results on several benchmarks including ShanghaiTech, UCF-QNRF, JHU-Crowd++ and NWPU-Crowd which demonstrates the efficacy of this combination.

The main contributions of this paper are summarized as follows:

- We propose a 'U-shaped' design [30] of multi-scale Transformer network for crowd counting, we refer to as Crowd U-Transformer (CUT). Our design effectively improves the model's performance in the presence of large scale variations of objects.
- We introduce an attention module, SAAM, which leverages the crowd segmentation results and a simple transformer block to extract fine-grained features in crowd regions.
- We design a new loss function for supervising the regression which provides a significant improvement on counting accuracy over previous attempts.

2 Related work

2.1 Crowd Counting

Modern crowd counting methods can be roughly categorized into three groups: detection-based [19, 22], direct regression-based [3, 4, 46] and density regression based approaches [13, 35]. To detect people in a crowd requires bounding box annotations, but such annotations are not only hard to obtain, but also imprecise due to the severe occlusion in crowd scenes. Therefore, the performance of such methods is limited. Also, without considering spatial annotations, the performance of direct regression is still unsatisfactory. The current mainstream approaches are based on density regression, which predict a pseudo-density map of an image and obtains the count by effectively summing the density values. Such approaches have been well integrated with the CNN framework and achieved significant performance gains. Various methods have been proposed to enhance the capabilities of CNNs for crowd counting, which includes multi-scale network design, auxiliary tasks and attention mechanism - whose

details are addressed in turn.

Multi-scale network design: This kind of approach is focused on improving CNNs' capabilities in handling large scale variations of crowds. One popular design is the multi-column network [1, 31, 55], which attempts to extract features of crowds at different scales by exploiting multi-columns of stacked CNNs with different receptive fields. Many works [24] adopt a single column network with pyramid design which aggregates features from different levels of the backbone network to obtain scale adaption. SASNet [38] uses a feature pyramid network to learn the relevancy across scales and feature levels. Its final prediction is a weighted average of individual predictions from different levels. TEDnet [12] hierarchically aggregates features from different levels which helps it learn multi-scale representations. Moreover, multi-scale blobs [32, 54] are frequently inserted in the single column neural networks to increase receptive field flexibility.

Auxiliary tasks: Object localization [10, 18, 50] and crowd segmentation [11, 29] are introduced into CNN models as a means of better capturing global semantic information. RAZNet [18] acquires better spatial information by adding a localization branch along with a zooming mechanism. ASNet [11] utilizes crowd segmentation to distinguish regions at different density levels, and applies corresponding scaling factors to different regions to improve counting accuracy.

Attention mechanism: The attention mechanism has been widely incorporated in crowd counting models [18, 27, 29] since it enables the model to focus on important regions and to some extent helps address scale and density variations. SDANet [27] reduces the impact of the background on counting performance by leveraging an attention mask generated from low-level features. CFANet [29] exploits a from-coarse-to-fine attention mechanism to better fuse features from different levels while focusing on the crowd regions.

More recent works [23, 25, 45] focus on designing novel loss function to directly have point annotations for supervision. Although significant progress has been achieved under CNN framework, the limited receptive field of CNN has also constrained the development of crowd counting.

2.2 Vision transformer

The transformer was originally designed for Natural Language Processing (NLP). With a global attention mechanism, the transformer is able to model long range dependencies and has been widely used in this field [2, 8, 42]. Inspired by the success of transformer in NLP tasks, Dosovitskiy *et al*. [9] proposed the first vision transformer (ViT), achieving promising results on image classification tasks. Since then, several vision transformer structures have been proposed for different downstream vision tasks such as classification [6, 49, 53], detection [2], and segmentation [57]. Currently, the study of applying transformer structures in crowd counting is still in an initial stage. The earliest study [15] performs weakly supervised learning to directly regress the count of an image with ViT. Later, Sun *et al*. [40] designed the first point-supervised crowd counting model which utilizes T2T-ViT [53] as a backbone to extract features. At about the same time, CCTrans [41] was proposed, which adopts Twins-SVT [6] as the backbone structure and designs a multi-scale blob to better handle scale and density variances. Both of these methods use point maps for supervision and adopt optimal transport loss [45]. A major disadvantage of these two works is they require an image to be cut into patches during inference stage to match the image size used in training, which can result in redundant counts at the boundary. Our work resolves this problem and fills in

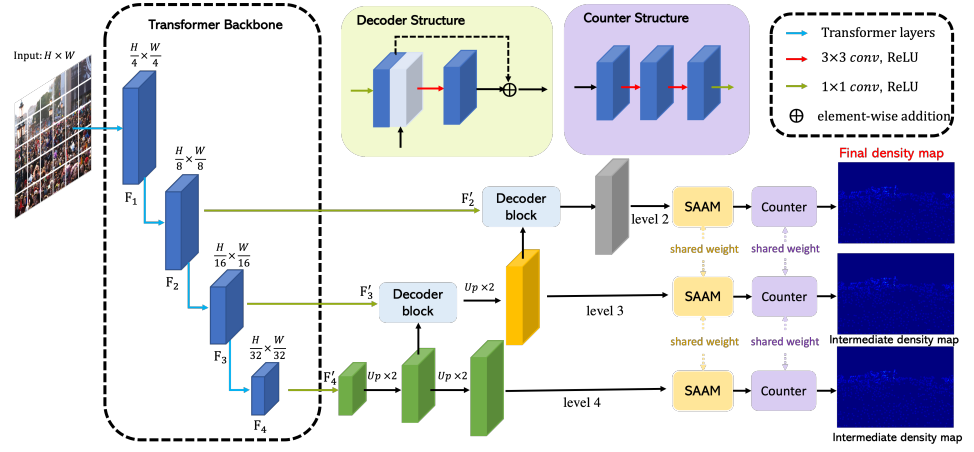


Figure 1: The pipeline of the proposed CUT. We have Twins-PCPVT [6] as the backbone network to extract multi-scale features. Feature maps from different levels are gradually aggregated under supervision. The final density map is predicted from the last fused feature maps.

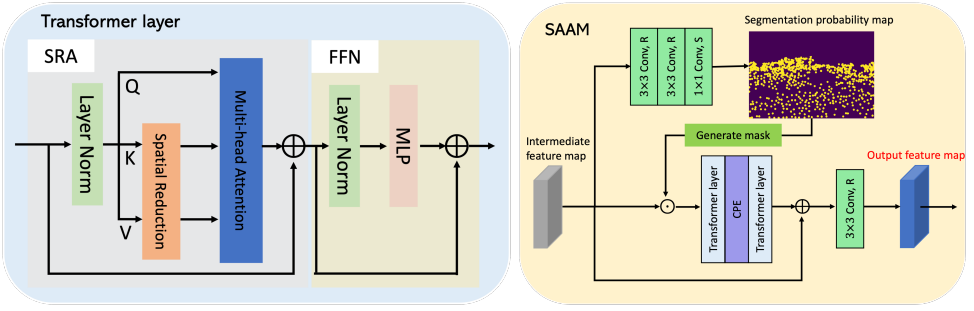
the gap of having density maps as the learning target under the transformer framework and shows the protential of transformers in crowd counting.

3 Proposed method

The overall architecture of our CUT is presented in Figure 1. An input image is first fed into the backbone Twins-PCPVT to extract multi-level features. Inspired by the U-shape design, we progressively fuse feature maps from different levels to recover the subsampling effected information loss caused by the backbone. Supervision is performed on each level. Crowd segmentation is introduced into the framework to guide the regression. The density map predicted from the last fused feature map is used as the final estimate. During the inference stage, our method only needs to scale the height and width of the input image to a minimum divisible by 32, which makes it much easier to use.

3.1 U-shaped multi-scale transformer architecture

To facilitate the modelling of long-term dependencies, the backbone of our network is formed by Twins-PCPVT [6] which extracts multi-scale features. This is an improved version of the pyramid vision transformer [49] which replaces absolute position encoding with conditional position encoding (CPE). The latter is more suitable for crowd counting since it enables the model to accept input images of varying sizes. The model itself has four stages and each stage has a similar architecture. At the start of each stage, the input image/feature map is divided into non-overlapping patches and the output is fed into a convolutional projection, which is then flattened for passing through the transformer layers. CPE is added after the first transformer layer of each stage. Our CPE is generated as follows: the flattened tokens are first reshaped back to a 2D feature map and then the result is fed into a single convolutional layer. Finally, the generated CPE is added on to the corresponding token. The output feature maps F_1, F_2, F_3 and F_4 from four stages are $\frac{1}{4}, \frac{1}{8}, \frac{1}{16}$ and $\frac{1}{32}$ of the original height and



(a) (b)

Figure 2: (a) The structure of the basic transformer layer in our model, proposed in PvT [49], which contains a SRA and FFN unit. $\oplus$ denotes element-wise addition. (b) The structure of the proposed Segmentation as attention module. In convolutional blocks, R is short for ReLU function while S is Sigmoid function. $\odot$ denotes element-wise multiplication.

width of the input, respectively. We fuse feature maps F_4, F_3 and F_2 in turn with CNN-based decoders to equip the final feature map with rich semantic and spatial information.

Transformer layer: The transformer layer of PcPvT is composed of the spatial-reduction attention (SRA) and the fully connected feed forward network (FFN). Differing from the multi-head self-attention, which the original transformer applied [42], SRA reduces the spatial scale of K (key) and V (value) with a convolutional operation which largely lowers the computational complexity. We present this structure in Figure 2(a). The output of a transformer layer can be written as follows:

$$T_i^{'} = SRA(T_{i-1}) + T_{i-1} \qquad i = 1, 2, ..., t_N, \tag{1}$$

$$T_i = FFN(T_i^{'}) + T_i^{'} \qquad i = 1, 2, ..., t_N, \tag{2}$$

where t_N is the number of transformer layers in a stage and T_i is the output of layer i.

Multi-level feature aggregation: The feature maps extracted by the transformer backbone are multi-scale. High-level feature maps typically contain more semantic information than low-level feature maps; the content of the latter comprising finer detail and local appearance information. To obtain a feature map having both rich semantic and spatial information, we follow the popular 'U-shaped' design and gradually fuse features from different stages.

Specifically, we first unify the channels of the output feature maps from the backbone to 256 with convolutional projections, noting that only feature maps from the last three stages described previously are used here. For a feature map $F_i^{'}$ and the feature map in its previous stage $F_{i-1}^{'}$, the expression of the fusion procedure can be written as follows:

$$F^{i,i-1} = f([UP(F_i'), F_{i-1}^{'}]) + F_{i-1}^{'}, \tag{3}$$

where $f(\cdot)$ is a 3×3 convolutional operation with ReLU as activation function, $[\cdot]$ denotes the concatenation layer and $UP(\cdot)$ represents up-sampling by a ratio of 2. We use the skip connection here to preserve spatial information better.

3.2 Segmentation as attention module

The overall structure of SAAM is shown in Figure 2(b). We introduce crowd segmentation to gain extra focus for an image. The process is detailed here. For a given intermediate

feature map $F \in \mathbb{R}^{C\times W\times H}$, we first feed it into three consecutive convolutional layers to get a probability map. Note, the labels (S_{gt}) for segmentation task is generated from the down-sampled ground-truth density map (D_{gt}).

$$S_{gt} = \mathbb{1}(D_{gt} > \varepsilon), \tag{4}$$

where $\mathbb{1}(\cdot)$ is the indicator function and ε is a threshold which is set as 1e-3 here. The predicted probability map is then converted to a binary mask and applied on the intermediate feature maps. The result is fed into a simple transformer block, followed by a skip connection with F.

Previous methods [26, 29, 33] which also add segmentation as auxiliary task, tend to directly assign the predicted probability as weight to according pixels. However, such an approach is highly dependent on the quality of the segmentation result since it posits that the probability is proportional to the density values. Meanwhile, the segmentation results may be inaccurate, especially when using the generated density maps for supervision [43] as the precise boundary of foreground and background is normally unclear. Moreover, the point annotations themselves are sometimes inaccurate. Bad segmentation results on boundary pixels can result in small values on the feature maps of corresponding position, thus affecting the density predictions. Our method can alleviate this problem somewhat – we are not scaling the values in 'low-probability' regions but instead leveraging transformer layers to model the relationships between these crowd regions, which in turn get better fine-grain features.

3.3 Loss design

Supervisions are provided on all three levels and for each level l, we have loss function L^l, that can be written as:

$$L^l = L^l_S + L^l_R \qquad l = 2, 3 \text{ and } 4, \tag{5}$$

where L_S and L_R supervise the segmentation and the regression tasks respectively. In order to focus learning on misclassified samples, L_S adopts pixel-level focal loss [17] and is defined as follows:

$$L_S = -\sum_{i\in S_{gt}} l_i(1-p_i)^\gamma \log(p_i) + (1-l_i)p_i^\gamma \log(1-p_i), \tag{6}$$

where i denotes the pixel within S_{gt}, and l_i and p_i represent the actual label of the pixel and the predicted probability of that pixel being foreground, respectively; γ is the modulating factor.

We use L_R to supervise the generation of the density map. Previous density map-based methods often adopt Euclidean loss for this purpose [14, 20, 55]. However, such choice has two weaknesses: (1) it assumes pixel-wise independent which ignores the local relation in the density map, and (2) it results in excessively smooth predictions [28] as it ignores the common imbalance between low-density and high-density distributions. To alleviate this problem, we propose the following loss function, which emphasises the importance of counting accuracy over dense regions:

$$L_R = SL(D_p \odot S_{gt}, D_{gt} \odot S_{gt}) + \lambda \cdot L_{TV}(D_p, D_{gt}), \tag{7}$$

where SL represents the structural loss [29], D_p indicates the predicted density map, $\odot$ is the element-wise multiplication, λ is the tunable hyper-parameter, and L_{TV} is the total variation loss [45]. SL leverages SSIM [51] index and the pooling operation to ensure consistency of

local relations and the counting accuracy, respectively. By masking the low-density regions in SL, the dense regions can be approximated well since the imbalance problem is somewhat alleviated. Here, we employ a 3×3 Gaussian kernel to calculate SSIM index. L_{TV} is adopted here to provide supervision on low-density regions. λ is set as 0.01 in our experiments.

The overall loss function is a combination of L^l from three levels:

$$L_{total} = \sum_{l=2}^{4} L_S^l + \alpha \sum_{l=3}^{4} L_R^l + L_R^2. \tag{8}$$

α is a factor used to lower the weight of loss from intermediate levels and its value is set to 0.5.

4 Experiments

In this section, we first introduce the datasets that are used in the experiments and the corresponding experimental settings, followed by an ablation study. Finally, we compare our results with the state-of-the-art methods.

4.1 Datasets

To demonstrate the effectiveness of our method, we conduct extensive experiments on four largest crowd counting datasets: ShanghaiTech [56], UCF-QNRF [10], JHU-Crowd++ [36] and NWPU-Crowd [43]. We succinctly summarize these:

ShanghaiTech A contains 482 images, collected from internet. 300 images are used for training and the remaining 182 images are used for testing. The number of annotations in an image varies from 33 to 3139.

UCF-QNRF consists of 1535 high resolution images with around 1.25 million annotations. The large diversity in densities and scenes makes it challenging. The training set has 1201 images and the remaining 304 images are in the testing set.

JHU-Crowd++ is a large scale crowd counting dataset that includes 4372 images. 2272 images are divided into the training set, 500 images are in the validation set and the remaining 1600 images are used for testing. A considerable number of images in this dataset are with adverse weather conditions and illumination variations.

NWPU-Crowd is the current largest crowd counting dataset which consists of 5109 high-resolution images with over 2.13 million annotations. The number of people in an image ranges from 0 to 20033. Moreover, negative samples are introduced in this dataset which refers to images without people or has similar texture as crowd scenes. 3109 images are used for training, 500 images are for validation and the remaining are for testing.

4.2 Experimental settings

Implement details: Following [14], we generate the ground-truth density map by using geometry-adaptive kernels. The transformer backbone is initialized with the official Twins-PCPVT-Large which has been pretrained on ImageNet. The drop path rate is set to 0.45 for the backbone. We only adopt random horizontal flipping and random cropping as image augmentation techniques. In addition, we limit the longer side of each image within 1920 pixels in all datasets. Note, this image resolution is the lowest among previous transformer-based methods. AdamW is used to optimize our model. Other settings vary with datasets and are detailed in Table 1.

Table 1: The detailed training settings for different datasets.

Dataset	learning rate	batch size	cropping size	γ
ShanghaiTech A	1e-5	4	256 ×256	4
UCF-QNRF	1e-4	8	512 ×512	2
JHU-Crowd++	1e-5	8	512 ×512	1
NWPU-Crowd	1e-5	8	512 ×512	2

Evaluation metrics: To evaluate the performance of our method, we adopt two commonly used metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE).

4.3 Ablation studies

We conduct extensive ablation studies on ShanghaiTech A (SHA) and UCF-QNRF to analyze the contribution of each component in our method.

Table 2: The Ablation study on multi-level supervision (MLS).

	SHA		UCF-QNRF	
	MAE	MSE	MAE	MSE
w/o MLS	54.0	88.1	79.1	138.1
Ours	**51.9**	**79.1**	**78.4**	**135.6**

Effectiveness of multi-level supervision: We perform an experiment to investigate the necessity of intermediate-level supervision by removing supervisions on both levels. The result is presented in Table.2. With supervisions on mid-level density maps, MAE and MSE are improved by 3.8% and 10.2% on SHA, 0.9% and 1.8% on UCF-QNRF, respectively. Therefore, mid-level supervisions could enhance the robustness of intermediate feature maps and help with the final density regression, especially for small datasets.

Table 3: Ablation study on SAAM.

	SHA		UCF-QNRF	
	MAE	MSE	MAE	MSE
w/o SAAM	54.2	85.9	82.1	144.2
PAW [29]	53.6	82.5	80.6	146.2
Ours	**51.9**	**79.1**	**78.4**	**135.6**

Table 4: Ablation study on loss function.

	SHA		UCF-QNRF	
	MAE	MSE	MAE	MSE
BSL [29]	56.5	90.3	87.4	144.3
Ours	**51.9**	**79.1**	**78.4**	**135.6**

Effectiveness of SAAM: We first perform an experiment where SAAM is totally removed from the model. Then, to further demonstrate its effectiveness over traditional 'probability as weight' (PAW) approach, we perform another experiment where probability map is directly applied on the intermediate feature maps. As shown in Table 3, solid improvement on MAE and MSE is observed on both datasets by adding SAAM. Moreover, our method is better than the previous PAW approach. Specifically, MAE is reduced by 3.2% and 2.7% on SHA and UCF-QNRF, respectively. MSE is also reduced by 4.1% and 7.3%, respectively.

Effectiveness of loss function: We compare our loss design with Rong and Li's *BSL* [29] by substituting L_R at each level with their settings. The result is shown in Table 4 which shows that our method effects a significant improvement of counting accuracy. In particular, MAE and MSE are reduced by 8.1% and 12.4% on SHA, 10.3% and 6.0% on UCF-QNRF.

Table 5: Performance comparison with the state of the art models on ShanghaiTech A, UCF-QNRF, JHU-Crowd++ and NWPU. The best and the second best performance are shown in **bold** and underlined, respectively.

Model	Label level	SHA		UCF-QNRF		JHU-Crowd++		NWPU	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCNN[53] (CVPR16)	density map	110.2	173.2	277	426	188.9	483.4	232.5	714.6
CSRNet[17] (CVPR18)	density map	68.2	115.0	-	-	121.3	387.8	121.3	387.8
SANet[8] (ECCV18)	density map	67.0	104.5	-	-	91.1	320.4	-	-
CAN[21] (CVPR19)	density map	62.3	100.0	107.0	183.0	100.1	314.0	106.3	386.5
SFCN[47] (CVPR19)	density map	64.8	107.5	102.0	171.4	77.5	297.6	-	-
BL[25] (ICCV19)	point map	62.8	101.8	88.7	154.8	75.0	299.9	105.4	454.2
ASNet[10] (CVPR20)	density map	57.8	90.1	91.6	159.7	-	-	-	-
AMRNet[20] (ECCV20)	density map	61.5	98.3	86.6	152.2	-	-	-	-
DPN-IPSM[28] (ACMMM20)	point map	58.1	91.7	84.7	147.2	-	-	-	-
DM-Count[45] (NIPS20)	point map	59.7	95.7	85.6	148.3	-	-	88.4	388.6
UOT[23] (AAAI21)	point map	58.1	95.9	84.3	142.3	60.5	252.7	87.8	387.5
S3[16] (IJCAI21)	point map	57.0	96.0	80.6	139.8	59.4	244.0	-	-
GL[32] (CVPR21)	point map	61.3	95.4	84.3	147.5	59.9	259.5	79.3	346.1
D2CNet[6] (IEEE-TIP21)	density map	57.2	93.0	81.7	137.9	73.7	292.5	85.5	361.5
CFANet[26] (WACV21)	density map	56.1	89.6	89.0	152.3	-	-	-	-
SASNet[39] (AAAI21)	density map	53.6	88.4	85.2	147.3	-	-	-	-
P2PNet[37] (ICCV21)	point map	52.7	85.1	85.3	154.5	-	-	72.6	331.6
BCCT[40] (arXiv21)	point map	53.1	82.2	83.8	143.4	54.8	**208.5**	82.0	366.9
CCTrans[41] (arXiv21)	point map	52.3	84.9	82.8	142.3	-	-	**69.3**	**299.4**
CUT(ours)	density map	**51.9**	**79.1**	**78.4**	**135.6**	**54.3**	229.1	**69.3**	304.0

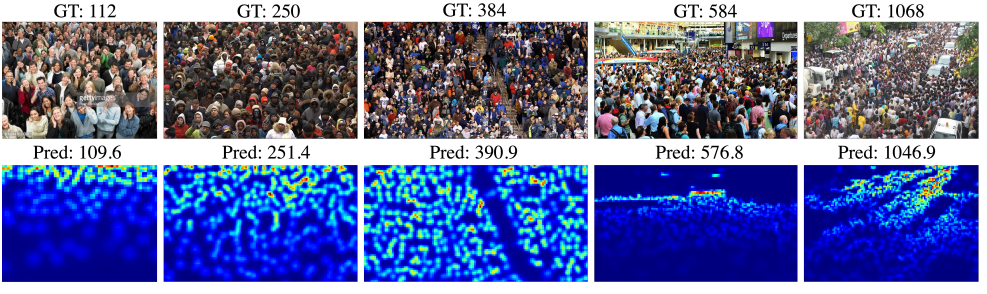


Figure 3: Visualizations of CUT on ShanghaiTech A. The first row are the input images and their corresponding estimated density maps are given in second row. 'GT' represents ground-truth count while 'Pred' means predicted count.

4.4 Comparisons with the state of the art

We evaluate the performance of our method on the aforementioned four datasets and compare our method's performance with the current state-of-the-art crowd counting models. The results are shown in Table 5. Visualizations of our CUT on SHA are shown in Figure 3.

Overall, our method establishes the new state-of-the-art overall and convincingly outperforms the current best density map-based method on all four benchmarks. On SHA, our method achieves better results than any reported in the existing literature. On UCF-QNRF, compared with the second best method CCTrans [41], CUT reduces MAE and MSE by 4.4 and 6.7, respectively. CUT also performs well on large-scale datasets. We achieve comparable performance of the current state-of-the-art work BCCT [40] and CCTrans [41] on JHU-Crowd++ and NWPU-Crowd, respectively.

5 Conclusion

In this paper, we proposed a crowd counting method, CUT, which utilises a transformer backbone for feature extraction. Using a U-shape design, CUT progressively aggregates multi-level features and recovers resolution. We used a segmentation based attention module 'SAAM' to further obtain fine-grained features. We designed a new loss function which shows a major performance improvement over the previous structural loss. Extensive experiments have shown our model achieves the state of the art performance on several popular crowd counting benchmarks.

Acknowledgement

This work is supported by the Fundamental Research Funds for the Central Universities (AUGA5710011522). The authors would like to thank the China Scholarship Council – University of St Andrews Scholarships (No.201908060250) funds L. Zhang for her PhD.

References

[1] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *ECCV*, 2018.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing.

[3] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. Feature mining for localised crowd counting. In *Proceedings of the British Machine Vision Conference*, pages 21.1–21.11. BMVA Press, 2012. ISBN 1-901725-46-4. doi: http://dx.doi.org/10.5244/C.26.21.

[4] Ke Chen, Shaogang Gong, Tao Xiang, and Chen Change Loy. Cumulative attribute space for age and crowd density estimation. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2467–2474, 2013. doi: 10.1109/CVPR.2013.319.

[5] Jian Cheng, Haipeng Xiong, Zhiguo Cao, and Hao Lu. Decoupled two-stage crowd counting and beyond. *IEEE Transactions on Image Processing*, 30:2862–2875, 2021. doi: 10.1109/TIP.2021.3055631.

[6] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *NeurIPS 2021*, 2021.

[7] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988.

Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1285. URL https://doi.org/10.18653/v1/p19-1285.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net, 2021.

[10] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 544–559, Cham, 2018. Springer International Publishing.

[11] Xiaoheng Jiang, Li Zhang, Mingliang Xu, Tianzhu Zhang, Pei Lv, Bing Zhou, Xin Yang, and Yanwei Pang. Attention scaling for crowd counting. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4705–4714, 2020. doi: 10.1109/CVPR42600.2020.00476.

[12] Xiaolong Jiang, Zehao Xiao, Baochang Zhang, Xiantong Zhen, Xianbin Cao, David Doermann, and Ling Shao. Crowd counting and density estimation by trellis encoder-decoder networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6126–6135, 2019. doi: 10.1109/CVPR.2019.00629.

[13] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, page 1324–1332, Red Hook, NY, USA, 2010. Curran Associates Inc.

[14] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1100, 2018.

[15] Dingkang Liang, Xiwu Chen, Wei Xu, Yu Zhou, and Xiang Bai. Transcrowd: weakly-supervised crowd counting with transformers. *Science China Information Sciences*, 65 (6):1–14, 2022.

[16] Hui Lin, Xiaopeng Hong, Zhiheng Ma, Xing Wei, Yunfeng Qiu, Yaowei Wang, and Yihong Gong. Direct measure matching for crowd counting. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 837–844. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/116. URL https://doi.org/10.24963/ijcai.2021/116.

[17] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society. doi: 10.1109/ICCV.2017.324. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.324.

[18] Chenchen Liu, Xinyu Weng, and Yadong Mu. Recurrent attentive zooming for joint crowd counting and precise localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[19] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G. Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2018. doi: 10.1109/CVPR.2018.00545.

[20] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5094–5103, 2019. doi: 10.1109/CVPR.2019.00524.

[21] Xiyang Liu, Jie Yang, Wenrui Ding, Tieqiang Wang, Zhijin Wang, and Junjun Xiong. Adaptive mixture regression network with local counting map for crowd counting. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 241–257, Cham, 2020. Springer International Publishing.

[22] Yuting Liu, Miaojing Shi, Qijun Zhao, and Xiaofang Wang. Point in, box out: Beyond counting persons in crowds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6462–6471, 2019. doi: 10.1109/CVPR.2019.00663.

[23] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6141–6150, 2019. doi: 10.1109/ICCV.2019.00624.

[24] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. *Learning Scales from Points: A Scale-Aware Probabilistic Model for Crowd Counting*, page 220–228. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450379885. URL https://doi.org/10.1145/3394171.3413642.

[25] Zhiheng Ma, Xing Wei, Xiaopeng Hong, Hui Lin, Yunfeng Qiu, and Yihong Gong. Learning to count via unbalanced optimal transport. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(3):2319–2327, May 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16332.

[26] Yanda Meng, Joshua Bridge, Meng Wei, Yitian Zhao, Yihong Qiao, Xiaoyun Yang, Xiaowei Huang, and Yalin Zheng. Counting with adaptive auxiliary learning, 2022. URL https://arxiv.org/abs/2203.04061.

[27] Yunqi Miao, Zijia Lin, Guiguang Ding, and Jungong Han. Shallow feature based dense attention network for crowd counting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:11765–11772, 04 2020. doi: 10.1609/aaai.v34i07.6848.

[28] Davide Modolo, Bing Shuai, Rahul Rama Varior, and Joseph Tighe. Understanding the impact of mistakes on background regions in crowd counting. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1649–1658, 2021. doi: 10.1109/WACV48630.2021.00169.

[29] Liangzi Rong and Chunping Li. Coarse- and fine-grained attention network with background-aware loss for crowd density map estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3675–3684, January 2021.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

[31] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. Switching convolutional neural network for crowd counting. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4031–4039, 2017. doi: 10.1109/CVPR.2017.429.

[32] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. Crowd counting with deep negative correlation learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5382–5390, 2018. doi: 10.1109/CVPR.2018.00564.

[33] Zenglin Shi, Pascal Mettes, and Cees Snoek. Counting with focus for free. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4199–4208, 2019. doi: 10.1109/ICCV.2019.00430.

[34] Julio Cezar Silveira Jacques Junior, Soraia Raupp Musse, and Claudio Rosito Jung. Crowd analysis using computer vision techniques. *IEEE Signal Processing Magazine*, 27(5):66–77, 2010. doi: 10.1109/MSP.2010.937394.

[35] Vishwanath A. Sindagi and Vishal M. Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1879–1888, 2017. doi: 10.1109/ICCV.2017.206.

[36] Vishwanath A Sindagi, Rajeev Yasarla, and Vishal M Patel. Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. *Technical Report*, 2020.

[37] Qingyu Song, Changan Wang, Zhengkai Jiang, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yang Wu. Rethinking counting and localization in crowds: A purely point-based framework. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3345–3354, 2021. doi: 10.1109/ICCV48922.2021.00335.

[38] Qingyu Song, Changan Wang, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Jian Wu, and Jiayi Ma. To choose or to fuse? scale selection for crowd counting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(3):2576–2583, May 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16360.

[39] Qingyu Song, Changan Wang, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Jian Wu, and Jiayi Ma. To choose or to fuse? scale selection for crowd counting. *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, 2021.

[40] Guolei Sun, Yun Liu, Thomas Probst, Danda Pani Paudel, Nikola Popovic, and Luc Van Gool. Boosting crowd counting with transformers, 2021. URL https://arxiv.org/abs/2105.10926.

[41] Ye Tian, Xiangxiang Chu, and Hongpeng Wang. Cctrans: Simplifying and improving crowd counting with transformer, 2021. URL https://arxiv.org/abs/2109.14483.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[43] Jia Wan and Antoni Chan. Adaptive density map generation for crowd counting. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1130–1139, 2019. doi: 10.1109/ICCV.2019.00122.

[44] Jia Wan, Ziquan Liu, and Antoni B. Chan. A generalized loss function for crowd counting and localization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1974–1983, 2021. doi: 10.1109/CVPR46437.2021.00201.

[45] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai. Distribution matching for crowd counting. In *Advances in Neural Information Processing Systems*, 2020.

[46] Chuan Wang, Hua Zhang, Liang Yang, Si Liu, and Xiaochun Cao. Deep people counting in extremely dense crowds. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, page 1299–1302, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450334594. doi: 10.1145/2733373.2806337. URL https://doi.org/10.1145/2733373.2806337.

[47] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8190–8199, 2019. doi: 10.1109/CVPR.2019.00839.

[48] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting and localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. doi: 10.1109/TPAMI.2020.3013269.

[49] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 568–578, October 2021.

[50] Yi Wang, Junhui Hou, Xinyu Hou, and Lap-Pui Chau. A self-training approach for point-supervised object detection and counting in crowds. *IEEE Transactions on Image Processing*, 30:2876–2887, 2021. doi: 10.1109/tip.2021.3055632. URL https://doi.org/10.1109%2Ftip.2021.3055632.

[51] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

[52] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems*, 34, 2021.

[53] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis E.H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 558–567, October 2021.

[54] Lingke Zeng, Xiangmin Xu, Bolun Cai, Suo Qiu, and Tong Zhang. Multi-scale convolutional neural networks for crowd counting. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 465–469, 2017. doi: 10.1109/ICIP.2017.8296324.

[55] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, 2016. doi: 10.1109/CVPR.2016.70.

[56] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, 2016. doi: 10.1109/CVPR.2016.70.

[57] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021.