CounTR: Transformer-based Generalised Visual Counting

Chang Liu[1]
liuchang666@sjtu.edu.cn
Yujie Zhong[2]
jaszhong@hotmail.com
Andrew Zisserman[3]
az@robots.ox.ac.uk
Weidi Xie[1]
weidi@sjtu.edu.cn

[1] Coop. Medianet Innovation Center, Shanghai Jiao Tong University, China
[2] Meituan Inc., China
[3] Visual Geometry Group (VGG), Department of Engineering Science University of Oxford, UK

Abstract

In this paper, we consider the problem of generalised visual object counting, with the goal of developing a computational model for counting the number of objects from *arbitrary* semantic categories, using an *arbitrary* number of "exemplars", *i.e.* zero-shot or few-shot counting. To this end, we make the following four contributions: (1) We introduce a novel transformer-based architecture for generalised visual object counting, termed a Counting TRansformer (**CounTR**), which explicitly captures the similarity between image patches or with given exemplars using the attention mechanism; (2) We adopt a two-stage training regime, that first pre-trains the model with self-supervised learning, followed by supervised fine-tuning; (3) We propose a simple, scalable pipeline for synthesizing training images with a large number of instances or from different semantic categories, explicitly forcing the model to make use of the given exemplars; (4) We conduct thorough ablation studies on a large-scale counting benchmark, FSC-147, and demonstrate state-of-the-art performance on both zero and few-shot settings. Project page: https://verg-avesta.github.io/CounTR_Webpage/.

1 Introduction

Despite all its exceptional abilities, the human visual system is particularly weak in counting objects in an image. In fact, given a visual scene with a collection of objects, one can only make a rapid, accurate, and confident judgment if the number of items is below five, with an ability known as subitizing [14]. While for scenes with an increasing number of objects, the accuracy and confidence of the judgments tend to decrease dramatically. Until at some point, counting can only be accomplished by calculating estimates or enumerating the instances, which incurs low accuracy or tremendous time cost.

In this paper, our goal is to develop a **generalised visual object counting** system, that augments humans' ability for determining the number of objects in a visual scene. Specifically, generalised visual object counting refers to the problem of identifying the number of

salient objects of *arbitrary* semantic class in an image (*i.e.* open-world visual object counting) with *arbitrary* number of instance "exemplars" provided by the end user, to select the particular objects to be counted, *i.e.* from zero-shot to few-shot object counting. To this end, we propose a novel architecture that transforms the input image (with the few-shot annotations if any) into a density map, and the final count can be obtained by simply summing over the density map.

We take inspiration from Lu *et al.* [16] that self-similarity is a strong prior in visual object counting, and introduce a transformer-based architecture where the self-similarity prior can be explicitly captured by the built-in attention mechanisms, both among the input image patches and with the few-shot annotations (if any). We propose a two-stage training scheme, with the transformer-based image encoder being firstly pre-trained with self-supervision via masked image modeling [9], followed by supervised fine-tuning for the task at hand. We demonstrate that self-supervised pre-training can effectively learn the visual representation for counting, thus significantly improving the performance. Additionally, to tackle the long-tailed challenge in existing generalised visual object counting datasets, where the majority of images only contain a small number of objects, we propose a simple, yet scalable pipeline for synthesizing training images with a large number of instances, as a consequence, establishing reliable data sources for model training, to condition the user-provided instance exemplars.

To summarise, in this paper, we make four contributions: *First*, we introduce an architecture for generalised visual object counting based on a transformer, termed **CounTR** (pronounced counter). It exploits the attention mechanisms to explicitly capture the similarity between image patches, or with the few-shot instance exemplars provided by the end user; *Second*, we adopt a two-stage training regime (self-supervised pre-training, followed by supervised fine-tuning) and show its effectiveness for the task of visual counting; *Third*, we propose a simple yet scalable pipeline for synthesizing training images with a large number of instances, and demonstrate that it can significantly improve the performance on images containing a large number of object instances; *Fourth*, we conduct thorough ablation studies on large-scale counting benchmarks, including FSC-147 [20] and CARPK [11], and demonstrate state-of-the-art performance on both zero-shot and few-shot settings, improving over the previous best approach by a noticeable margin on the mean absolute error of the FSC-147 test set.

2 Related Work

Visual object counting. In the literature, object counting approaches can generally be cast into two categories: detection-based counting [3, 5, 11] or regression-based counting [1, 2, 4, 13, 15, 17, 22]. The former relies on a visual object detector that can localize object instances in an image. This method, however, requires training individual detectors for different objects, and the detection problem remains challenging if only a small number of annotations are given. The latter avoids solving the hard detection problem, instead, methods are designed to learn either a mapping from global image features to a scalar (number of objects), or a mapping from dense image features to a density map, achieving better results on counting overlapping instances. However, previous methods from both lines (detection, regression) have only been able to count objects of one particular class (*e.g.* cars, cells).

Class-agnostic object counting. Recently, class-agnostic few-shot counting [16, 20, 23] has witnessed a rise in research interest in the community. Unlike the class-specific models that could only count objects of specific classes like cars, cells, or people, class-agnostic

counting aims to count the objects in an image based on a few given "exemplar" instances; thus it is also referred to as 'few-shot counting'. Generally speaking, class-agnostic few-shot counting models need to mine the commonalities between the instances of different classes of objects during training. In [16], the authors propose a generic matching network (GMN), which regresses the density map by computing the similarity between the CNN features from image and exemplar shots; FamNet [20] utilizes feature correlation for prediction and uses an adaptation loss to update the model's parameters at test time; SAFECount [23] uses a support feature to enhance the query feature, making the extracted features more refined and then regresses to obtain density maps. In a very recent work [10], the authors exploit a pre-trained DINO [18] model and a lightweight regression head to count without exemplars. In our approach, we also use a transformer-based architecture, however, it is trained from scratch, and augmented with the ability to count the objects given *any shot*.

3 Methods

In this paper, we consider the challenging problem of generalised visual object counting, where the goal is to count the salient objects of an **arbitrary** semantic class in an image, *i.e.* open-world visual object counting, with **arbitrary** number of exemplars provided by the end user, *i.e.* from zero-shot to few-shot object counting.

Overview. Given a training set, $\mathcal{D}_{\text{train}} = \{(\mathcal{X}_1, \mathcal{S}_1, y_1), \ldots, (\mathcal{X}_N, \mathcal{S}_N, y_N)\}$, where $\mathcal{X}_i \in \mathbb{R}^{H \times W \times 3}$ denotes the input image, $\mathcal{S}_i = \{b_i\}^K$ denotes the box coordinates ($b_i^k \in \mathbb{R}^4$) for a total of $K \in \{0, 1, 2, 3...\}$ given exemplars, *i.e.* zero-shot or few-shot counting, $y_i \in \mathbb{R}^{H \times W \times 1}$ refers to a binary spatial density map, with 1's at the objects' center location, indicating their existence, and 0's at other locations without the objects; the object count can thus be computed by spatially summing over the density map. Our goal here is to train a generalised visual object counter that can successfully operate on a test set, given zero or few exemplars, *i.e.* $\mathcal{D}_{\text{test}} = \{(\mathcal{X}_{N+1}, \mathcal{S}_{N+1}), \ldots, (\mathcal{X}_M, \mathcal{S}_M)\}$. **Note that**, the semantic categories for objects in the training set ($\mathcal{C}_{\text{train}}$) and testing set ($\mathcal{C}_{\text{test}}$) are disjoint, *i.e.* $\mathcal{C}_{\text{train}} \cap \mathcal{C}_{\text{test}} = \emptyset$.

To achieve this goal, we introduce a novel transformer-based architecture, called the Counting TRansformer (**CounTR**). Specifically, the attention mechanism in the transformer enable it to explicitly compare visual features between any other spatial locations and with exemplars, which are provided by the end user in the few-shot scenario. In Section 3.2, we further introduce a two-stage training regime, in which the model is firstly pre-trained with self-supervision via masked image reconstruction (MAE), followed by fine-tuning on the downstream counting task. To the best of our knowledge, this is the first work to show the effectiveness of self-supervised pre-training for generalised visual object counting. Additionally, in Section 3.3, we propose a novel and scalable *mosaic* pipeline for synthesizing training images, as a way to resolve the challenge of long-tailed distribution (*i.e.* images with a large number of instances tend to be less frequent) in the existing object counting dataset. In Section 3.4, we will introduce our test-time normalisation method including test-time cropping.

3.1 Architecture

Here, we introduce the proposed Counting TRansformer (**CounTR**), as shown in Figure 1. The input image ($\mathcal{X}_i$), and user-provided exemplars ($\mathcal{S}_i^k, \forall k \in \{0, 1, 2, 3\}$) are fed as input and

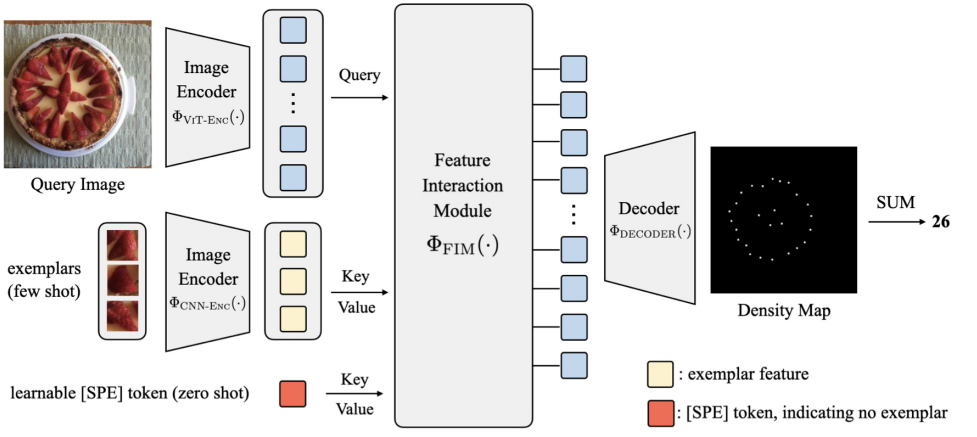


Figure 1. Architecture detail for CounTR. The query image and exemplars are encoded by separate visual encoders. The image features are then fed into the feature interaction module as query vectors, and the exemplar features are fed as key and value vectors. When there is no instance exemplar provided, a learnable [SPE] token is used as key and value instead. The outputs are up-sampled in the decoder and finally, we get the corresponding density map. The object count can be obtained by summing the density map. **Note that**, given the different exemplars with diversity, the model should ideally understand the invariance (shape, color, scale, texture). For example, if the three given exemplars are all of the same color, the model should only count the objects of that color, otherwise, count all instances of the same semantic.

mapped to a density heatmap, where the object count can be obtained by simply summing over it:

$$y_i = \Phi_{\text{DECODER}}(\Phi_{\text{FIM}}(\Phi_{\text{ViT-Enc}}(\mathcal{X}_i), \Phi_{\text{CNN-Enc}}(\mathcal{S}_i^k))),\ \forall k \in \{0,1,...,K\} \tag{1}$$

In the following sections, we will detail the three building components, namely, visual encoder ($\Phi_{\text{ViT-Enc}}(\cdot)$ and $\Phi_{\text{CNN-Enc}}(\cdot)$), feature interaction module (*i.e.* FIM, $\Phi_{\text{FIM}}(\cdot)$), and visual decoder ($\Phi_{\text{DECODER}}(\cdot)$).

3.1.1 Visual Encoder

The visual encoder is composed of two components, serving two purposes: *first*, an encoder based on a Vision Transformer (ViT) [8] for processing the input image that maps it into a high-dimensional feature map; *second*, an encoder to compute the visual features for the exemplars, if there are any. Specifically, as for ViT, the input image is broken into patches with a size of 16×16 pixels and projected to tokens by a shared MLP. To indicate the order of each token in the sequence, positional encoding is added, ending up with M 'tokens'. They are further passed through a series of transformer encoder layers, in our model, 12 layers are used. We do not include the [CLS] token in the sequence, and the output from the ViT encoder is a sequence of D-dim vectors :

$$\mathcal{F}_{\text{ViT}} = \Phi_{\text{ViT-Enc}}(\mathcal{X}_i) \in \mathbb{R}^{M \times D} \tag{2}$$

for more details, we refer the readers to the original ViT paper.

For few-shot counting, we use the exemplar encoder to extract the visual representation. It exploits a lightweight ConvNet architecture (4 convolutional layers, followed by a global

average pooling), that maps the given exemplars (resized to the same resolution) into vectors,

$$\mathcal{F}_{\text{CNN}} = \Phi_{\text{CNN-ENC}}(\mathcal{S}_i^k) \in \mathbb{R}^{K \times D} \tag{3}$$

Note that, under the zero-shot scenario with no exemplar given, we adopt a learnable [SPE] token as the substitute to provide cues for the model.

3.1.2 Feature Interaction Module

Here, we introduce the proposed feature interaction module (FIM), for fusing information from both encoders. Specifically, the FIM is constructed with a series of standard transformer decoder layers, where the image features act as the Query, and two different linear projections (by MLPs) of the exemplar features (or learnable special token), are treated as the Value and Key. With this design, the output from the FIM remains the same dimensions as the image features ($\mathcal{F}_{\text{VIT}}$), throughout the interaction procedure:

$$\mathcal{F}_{\text{FIM}} = \Phi_{\text{FIM}}(\mathcal{F}_{\text{VIT}},\, W^{\text{k}} \cdot \mathcal{F}_{\text{CNN}},\, W^{\text{v}} \cdot \mathcal{F}_{\text{CNN}}) \in \mathbb{R}^{M \times D} \tag{4}$$

Conceptually, such a transformer architecture perfectly reflects the self-similarity prior to the counting problem, as observed by Lu *et al.* [16]. In particular, the self-attention mechanism in the transformer decoder enables it to measure the self-similarity between regions of the input image, while the cross-attention between Query and Value allows it to compare image regions with the **arbitrary** given shots, incorporating users' input for more customised specification on the objects of interest, or simply learning to ignore the ConvNet branch when encountering the learnable [SPE] token.

3.1.3 Decoder

At this stage, the outputs from the feature interaction module are further reshaped back to 2D feature maps and restored to the original resolution as the input image. We adopt a progressive up-sampling design, where the vector sequence is first reshaped to a dense feature map and then processed by a ConvNet-based decoder. Specifically, we use 4 up-sampling blocks, each of which consists of a convolution layer and a $2\times$ bilinear interpolation. After the last up-sampling, we adopt a linear layer as the density regressor, which outputs a one-channel density heatmap:

$$y_i = \Phi_{\text{DECODER}}(\mathcal{F}_{\text{FIM}}) \in \mathbb{R}^{H \times W \times 1} \tag{5}$$

3.2 Two-stage Training Scheme

In images, the visual signals are usually highly redundant, *e.g.* pixels within local regions are spatially coherent. This property is even more obvious in the counting problem, as the objects often tend to appear multiple times in a similar form. Based on this observation, we employ MAE self-supervised learning to pre-train the visual encoder ($\Phi_{\text{VIT-ENC}}(\cdot)$). Specifically, we adopt the recent idea from Masked Autoencoders (MAE), to train the model by image reconstruction with only partial observations.

(a) Type A: using four images. (b) Type B: using one image.

Figure 2. The mosaic pipeline for synthesizing training images. (1) stands for crop and scale, and (2) stands for collage and blending. Type A uses four different images to improve background diversity and Type B uses only one image to increase the number of objects contained in an image. White highlights are the dot annotation density map after Gaussian filtering for visualization.

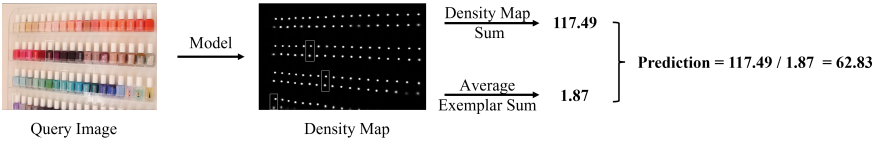
Self-supervised Pre-training with MAE. In detail, we first divide the image into regular non-overlapping patches, and only sample a subset of the patches (50% in our case) as input to the ViT encoders. The computed features are further passed through a lightweight decoder, consisting of several transformer decoder layers, where the combination of learnable mask tokens and positional encoding is used as `Query` to reconstruct the input image from only observed patches. The training loss is simply defined as the Mean Squared Error (MSE) between the reconstructed image and the input image in pixel space.

Supervised Fine-tuning. After the pre-training, we initialise the image encoder with the weights of the pre-trained ViT, and fine-tune our proposed architecture on generalised object counting. In detail, our model takes the original image $\mathcal{X}_i$ and K exemplars $\mathcal{S}_i = \{b_i\}^K$ from $\mathcal{D}_{\text{train}}$ as input and outputs the density map $\hat{y}_i \in \mathbb{R}^{H\times W\times 1}$ corresponding to the original image $\mathcal{X}_i$. The statistical number of salient objects in the image $C_i \in \mathbb{R}$ can be obtained by summing the discrete density map $\hat{y}_i$. We use the mean square error per pixel to evaluate the difference between the predicted density map $\hat{y}_i$ and the ground truth density map y_i. The ground truth density maps are generated based on the dot annotations: $\mathcal{L}(\hat{y}_i, y_i) = \frac{1}{HW}\sum||y_i - \hat{y}_i||_2^2$.

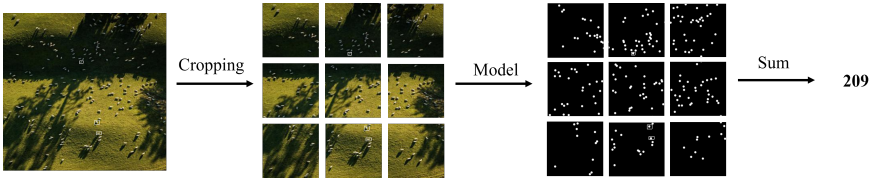
3.3 Scalable Mosaicing

In this section, we introduce a scalable *mosaic* pipeline for synthesizing training images, in order to tackle the long-tailed problem (*i.e.* very few images contain a large number of instances) in existing counting datasets. We observe that existing datasets for generalised object counting are highly biased towards a small number of objects. For example, in the FSC-147 dataset, only 6 out of 3659 images in the train set contain more than 1000 objects. This is potentially due to the costly procedure for providing manual annotation. In the following, we elaborate on the two steps of the proposed mosaic training data generation, namely, collage and blending (as shown in Figure 2). Note that, we also notice one concurrent work [10] uses a similar idea.

Collage. Here, we first crop a random-sized square area from the image and scale it to a uniform size, *e.g.* a quarter of the size of the original image. After repeating the region cropping multiple times, we collage the cropped regions together and update the corresponding density map. It comes in two different forms: using only one image or four different images. If we only use one image, we can increase the number of objects contained in the image, which helps a lot with tackling the long-tail problem. If we use four different images, we can significantly improve the training images' background diversity and enhance the model's ability to distinguish between different classes of objects. To fully use these two advantages, we make the following settings. If the number of objects contained in the image is more than



(a) Test-time Normalisation.



(b) Test-time Cropping.

Figure 3. The test-time normalisation process visualisation. In test-time normalisation, if the average sum of the exemplar positions in the density map is over 1.8, the sum of the density map will be divided by this average to become the final prediction. In test-time cropping, if at least one exemplar's side length is smaller than 10 pixels, the image will be cropped into 9 pieces and the model will process these 9 images separately. The final prediction will be the sum of the results of these 9 images.

a threshold, we use the same image to collage; if not, we use four different images. Note that if four different images are used, we could only use the few-shot setting for inference, otherwise the model will not know which object to count. If we use the same image, the mosaiced image can be used to train the few-shot setting and zero-shot setting.

Blending. Simply cropping and collaging does not synthesize perfect images, as there remain sharp artifacts between the boundaries. To resolve these artifacts, we exploit blending at the junction of the images. In practise, we crop the image with a slightly larger size than a quarter of the original image size, such that we can leave a particular space at the border for α-channel blending. We use a random α-channel border width, which makes the image's composition more realistic. **Note that**, we only blend the original image instead of the density map, to maintain the form of dot annotation (only 0 and 1). Since there are few objects inside the blending border and the mosaic using one image is only applied to images with a very large number of objects, the error caused by blending is almost negligible.

3.4 Test-time Normalisation

For few-shot counting, we have introduced a test-time normalisation strategy to calibrate the output density map. Specifically, at inference time, we exploit the prior knowledge that the object count at the exemplar position should exactly be 1.0, any prediction deviation can thus be calibrated by dividing the density map by the current predicted count at the exemplar position. We take this approach because due to the ambiguity of the bounding boxes, the model sometimes chooses the smallest self-similarity unit of an object to count, rather than the entire object, as shown in Figure 3 (a). Therefore, if the average sum of the density map area corresponding to the bounding boxes exceeds a threshold, such as 1.8, we will exploit this test-time normalisation approach.

Additionally, for images with tiny objects (one exemplar with a side length shorter than 10 pixels), we adopt a sliding window prediction, by dividing the image equally into nine

pieces and scaling them to their original size, to be individually processed by our model. The total number of objects is the sum of the individual count results of the nine images.

4 Experiments

Here, we start by briefly introducing the few-shot counting benchmark, FSC-147 dataset, and the evaluation metrics. In Section 4.2, we describe the implementation details of our model and the design of test-time normalisation; In Section 4.3, we compare our model's performance with other counting models and demonstrate state-of-the-art performance on both zero-shot and few-shot settings; In Section 4.4, we conduct a series of ablation studies to demonstrate the effectiveness of the two-stage training and the image mosaicing.

4.1 Datasets and Metrics

Datasets. We experiment on FSC-147 [20], which is a multi-class few-shot object counting dataset containing 6135 images. Each image's number of counted objects varies widely, ranging from 7 to 3731, and the average is 56. The dataset also provides three randomly selected object instances annotated by bounding boxes as exemplars in each image. The training set has 89 object categories, while the validation and test sets both have 29 disjoint categories, making FSC-147 an open-set object counting dataset.

Metrics. We use two standard metrics to measure the performance of our model, namely, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

$$MAE = \frac{1}{N_I}\sum_{i=1}^{N_I}|C_i - C_i^{GT}|, \qquad RMSE = \sqrt{\frac{1}{N_I}\sum_{i=1}^{N_I}(C_i - C_i^{GT})^2} \tag{6}$$

Here, N_I is the total number of testing images, and C_i and C_i^{GT} are the predicted number and ground truth of the i^{th} image.

4.2 Implementation

4.2.1 Training Details

In this section, we give the details of our proposed two-stage training procedure. That is, first pre-train the ViT encoder with MAE [9], and then fine-tune the whole model on supervised object counting.

MAE Pre-training. As input, the image is of size 384×384, which is first split into patches of size 16×16, and projected into 576 vectors. Our visual encoder uses 12 transformer encoder blocks with a hidden dimension of 768, and the number of heads in the multi-head self-attention layer is 12. The decoder uses 8 transformer layers with a hidden dimension of 512. As input for pre-training ViT with MAE, we randomly drop 50% of the visual tokens, and task the model to reconstruct the masked patches with pixel-wise mean square error. During pre-training, we chose a batch size of 16 and trained on the FSC-147 for 300 epochs with a learning rate of 5×10^{-6}.

Fine-tuning stage. The feature interaction module uses 2 transformer decoder layers with a hidden dimension of 512. The ConvNet encoder exploits 4 convolutional layers and a global average pooling layer to extract exemplar features with 512 dimensions. The image decoder uses 4 up-sampling layers with a hidden dimension of 256. For optimisation, we minimise the mean square error between the model's prediction and the ground truth density map, which is generated with Gaussians centered on each object. We scale the loss by a factor of 60, and randomly drop 20% non-object pixels, to alleviate the sample imbalance issue. We use AdamW as the optimiser. Our model is trained on the FSC-147 training set with a learning rate of 1×10^{-5} and a batch size of 8. Our model is trained and tested on NVIDIA GeForce RTX 3090.

4.2.2 Inference Details

At inference time, we adopt sliding windows for images of different resolutions, with the model processing a portion of an image with a fixed-size square window as used in training, and gradually moving forward with a stride of 128 pixels. The density map for overlapped regions is simply computed by averaging the predictions.

4.3 Comparison to state-of-the-art

We evaluate the proposed CounTR model on the FSC-147 dataset and compare it against existing approaches. As shown in Table 1, CounTR has demonstrated new state-of-the-art on both zero-shot and few-shot counting, outperforming the previous methods significantly. For results on Val-COCO, Test-COCO [20], and CARPK [10], we refer the readers to the appendix.

Methods	Year	Backbone	# Shots	Val		Test	
				MAE	RMSE	MAE	RMSE
RepRPN-C [19]	Arxiv2022	ConvNets	0	31.69	100.31	28.32	128.76
RCC [11]	Arxiv2022	Pre-trained ViT	0	20.39	64.62	21.64	103.47
CounTR (ours)	2022	ViT	0	17.40	70.33	14.12	108.01
FR [23]	ICCV2019	ConvNets	3	45.45	112.53	41.64	141.04
FSOD [2]	CVPR2020	ConvNets	3	36.36	115.00	32.53	140.65
P-GMN [16]	ACCV2018	ConvNets	3	60.56	137.78	62.69	159.67
GMN [16]	ACCV2018	ConvNets	3	29.66	89.81	26.52	124.57
MAML [8]	ICML2017	ConvNets	3	25.54	79.44	24.90	112.68
FamNet [20]	CVPR2021	ConvNets	3	23.75	69.07	22.08	99.54
BMNet+ [21]	CVPR2022	ConvNets	3	15.74	58.53	14.62	91.83
CounTR (ours)	2022	ViT	3	**13.13**	**49.83**	**11.95**	**91.23**

Table 1. Comparison with state-of-the-art on the FSC-147 dataset. P-GMN stands for Pre-trained GMN. RepRPN-C stands for RepRPN-Counter. RCC stands for reference-less class-agnostic counting with weak supervision.

4.4 Ablation Study

In this section, we have conducted thorough ablation studies to demonstrate the effectiveness of the proposed ideas. As shown in Table 2, we can make the following observations: (1) **Data augmentation:** While comparing the Model-A, we include image-wise data augmentation in Model-B training, including Gaussian noise, Gaussian blur, horizontal flip, color jittering, and geometric transformation. As indicated by the result, Model B slightly

outperforms Model A on both validation and test set, suggesting that these augmentation methods can indeed be useful to the model to a certain extent. (2) **Self-supervised pre-training:** In Model-C, we introduce the self-supervised pre-training for warming up the ViT encoder. Compared with Model B which directly fine-tunes the ViT encoder (pre-trained on ImageNet) on the FSC-147 training set, Model C has improved all results on both validation and test sets significantly. (3) **Effectiveness of mosaic:** With the help of the mosaic method, Model-D has shown further performance improvements, demonstrating its effectiveness for resolving the challenge from the long-tailed challenge, by introducing images with a large number of object instances, and object distractors from different semantic categories. (4) **Test-time normalisation:** In Model-E, we experiment with test-time normalisation for the few-shot counting scenario, where the output prediction is calibrated by the given exemplar shot. On both validation and test set, test-time normalisation has demonstrated significant performance boosts. (5) **On shot number:** In Model E, as the number of given shots increases, *i.e.* 1, 2, or 3, we observe only tiny differences in the final performance, showing the robustness of CounTR for visual object counting under *any* shots.

Model	Augmentation	Selfsup	Mosaic	TT-Norm.	# Shots	Val		Test	
						MAE	RMSE	MAE	RMSE
A0	✗	✗	✗	✗	0	24.84	86.33	21.06	130.04
A1	✗	✗	✗	✗	3	24.68	85.89	20.98	129.58
B0	✓	✗	✗	✗	0	23.80	81.53	21.14	131.27
B1	✓	✗	✗	✗	3	23.67	81.40	20.93	130.75
C0	✓	✓	✗	✗	0	18.30	72.21	16.20	114.30
C1	✓	✓	✗	✗	3	18.19	71.47	16.05	113.11
D0	✓	✓	✓	✗	0	18.07	71.84	14.71	106.87
D1	✓	✓	✓	✗	3	17.40	70.33	14.12	108.01
E1	✓	✓	✓	✓	1	13.15	49.72	12.06	90.01
E2	✓	✓	✓	✓	2	13.19	49.73	12.02	90.82
E3	✓	✓	✓	✓	3	**13.13**	**49.83**	**11.95**	**91.23**
E3 (no 7171.jpg)	✓	✓	✓	✓	3	13.13	49.83	11.22	87.68

Table 2. Ablation study. We observe that one image in the test set (image id:7171) has significant annotation error (see supp. material), result without it has also been reported. **Selfsup**: refers to the proposed two-stage training regime. **TT-Norm**: denotes test-time normalisation

5 Conclusion

In this work, we aim at the generalised visual object counting problem of counting the number of objects from *arbitrary* semantic categories using an *arbitrary* number of "exemplars". We propose a novel transformer-based architecture for it, termed **CounTR**. It is first pre-trained with self-supervised learning, and followed by supervised fine-tuning. We also propose a simple, scalable pipeline for synthesizing training images that can explicitly force the model to make use of the given "exemplars". Our model achieves state-of-the-art performance on both zero-shot and few-shot settings.

Acknowledgements. AZ is supported by EPSRC Programme Grant VisualAI EP/T028572/1, and a Royal Society Research Professorship RP\R1\191132. We thank Xiaoman Zhang and Chaoyi Wu for proof-reading.

References

[1] Carlos Arteta, Victor Lempitsky, J Alison Noble, and Andrew Zisserman. Interactive object counting. In *European conference on computer vision*, 2014.

[2] Carlos Arteta, Victor Lempitsky, and Andrew Zisserman. Counting in the wild. In *European conference on computer vision*, 2016.

[3] Olga Barinova, Victor Lempitsky, and Pushmeet Kholi. On detection of multiple object instances using hough transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[4] Siu-Yeung Cho, Tommy WS Chow, and Chi-Tat Leung. A neural-based crowd estimation by hybrid global learning algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 1999.

[5] Chaitanya Desai, Deva Ramanan, and Charless C Fowlkes. Discriminative models for multi-class object layout. *International journal of computer vision*, 2011.

[6] 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 *Proceedings of the International Conference on Learning Representations*, 2021.

[7] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 2017.

[9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[10] Michael Hobley and Victor Prisacariu. Learning to count anything: Reference-less class-agnostic counting with weak supervision. *arXiv preprint arXiv:2205.10203*, 2022.

[11] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H Hsu. Drone-based object counting by spatially regularized regional proposal network. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[12] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[13] Dan Kong, Douglas Gray, and Hai Tao. A viewpoint invariant approach for crowd counting. In *18th International Conference on Pattern Recognition (ICPR'06)*, 2006.

[14] Kaufman E. L., Lord M. W., Reese T. W., and Volkmann J. The discrimination of visual number. *The American Journal of Psychology*, 1949.

[15] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. *Advances in neural information processing systems*, 2010.

[16] Erika Lu, Weidi Xie, and Andrew Zisserman. Class-agnostic counting. In *Asian Conference on Computer Vision*, 2018.

[17] Aparecido Nilceu Marana, SA Velastin, LF Costa, and RA Lotufo. Estimation of crowd density using image processing. 1997.

[18] Caron Mathilde, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision*, 2021.

[19] Viresh Ranjan and Minh Hoai. Exemplar free class agnostic counting. *arXiv preprint arXiv:2205.14212*, 2022.

[20] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[21] Min Shi, Hao Lu, Chen Feng, Chengxin Liu, and Zhi-Guo Cao. Represent, compare, and learn: A similarity-aware framework for class-agnostic counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[22] Weidi Xie, J Alison Noble, and Andrew Zisserman. Microscopy cell counting and detection with fully convolutional regression networks. *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, 2018.

[23] Zhiyuan You, Yujun Shen, Kai Yang, Wenhan Luo, Xin Lu, Lei Cui, and Xinyi Le. Few-shot object counting with similarity-aware feature enhancement. *arXiv preprint arXiv:2201.08959*, 2022.