Spatially-Aware Class-Agnostic Object Counting

Md Tanvir Hossain
u7363536

Amanda Kau
u6747529

Robert Wijaya
u7177316

Abstract

In an effort to mimic the human counting ability, we address the issue of generalised object counting from a single input image. We focus on a zero-shot method that does not require reference images or annotations of any kind. Motivated by several state-of-the-art methods which fail to produce reliable counts when faced with more structurally complex objects, we build upon the Counting TRansformer (CounTR) [6] architecture to address this limitation. Specifically, we employ a self-supervised knowledge distillation model, DINOv2, as an image encoder to improve the feature representation of input images. To preserve the spatial relationships inherent within the input images, we incorporate positional embeddings into the visual feature representation before being passed to ConvNet based decoder. Experimental results on the class-agnostic counting dataset, FSC-147, show that our method is comparable to established methods and is effective in counting objects with more complicated structures. Code is available at https://github.com/wijayarobert/spatially-aware-counting.

1. Introduction

Object counting in computer vision aims to accurately quantify any particular object in an image, and is applicable to logistics, surveillance, urban planning and more. Specialised counters such as ones for crowd-counting [16] and vehicles [8] exist. These specialised counters can effectively count large quantities of objects but are limited to specific object classes, a limitation that the human counting ability does not possess. Other general counters can count arbitrary object classes, but require additional text prompts or exemplars. Exemplars are particularly inefficient as they require humans to manually produce annotations like bounding boxes to isolate the desired object to count.

To this end, Liu et al. [6] developed Counting TRansformer (CounTR), a transformer-based model that can count any arbitrary class of objects in an effort to mimic our human counting ability. The authors were motivated by findings from Lu et al. in [7], where the object counting problem was posed as a matching problem for image patches, and self-similarity was an found to be an important element for counting. Attention mechanisms were built into CounTR's architecture as a result, which seemed to give CounTR an edge over several state-of-the-art (SOTA) models.

Figure 1: For the given image (left) with a ground truth of 149 objects, our model produced a density map (right) that was summed to predict an object count of 149.

However, CounTR, like many other models, failed where object structural information was key. For instance, the counts of sunglasses would be double the true count as each lens would be treated as a separate object. These cases necessitated test-time augmentation, which involved performing transformations to the query image to obtain several augmented images, performing counting on each image, then returning the average count as the final one. This introduced additional steps into the pipeline and increased computational demands.

We postulate that this limitation could be attributed to the visual encoder in CounTR [6] that failed to capture adequate structural information. As the visual encoder is the very first step in the pipeline, failing to capture key information in the image embedding at this stage limits downstream processes, resulting in inaccurate counts regardless of the counting efficiency of following stages in the model. We address this limitation by employing the self-supervised DINOv2 [9] model as the image encoder for two main reasons. Firstly, the model contains a substantial number of parameters that could enrich the image feature vectors. Secondly, DINOv2 has been shown to produce features that are suitable for various computer vision tasks without the need for finetuning, hence it is appropriate for our zero-shot counting task.

Our main contributions are as follows:

- Taking inspiration from CounTR [6] and building upon code from CounTX [1], which is a text-specified count-

ing model with a similar architecture as CounTR, we replace the existing image encoder with the DINOv2 [9] encoder.

- We combine the image features with positional embeddings as inputs to the model, and utilise a CNN-based decoder to produce a density map, which is summed to produce a predicted count.
- We evaluate our model on the FSC-147 [14] dataset and provide comparisons with state-of-the-art models.
- We demonstrate the effectiveness of our model in counting more structurally complex objects.

2. Related Work

2.1. Class-specific Object Counting

Object counting can be classified into two broad categories, class-specific and class-agnostic counting. The former counts specific classes, such as people [16] or cars [8]. Counting methods can be generally classified into one of two types [15]. The first type are detection-based methods, which make use of object detectors which define bounding boxes around objects and count the number of boxes. The second type are regression-based methods that predict a density map for the given image and produce a final count by summing up the map's pixel values. Class-specific methods cannot be extended to arbitrary classes and cannot count multiple types of objects that might appear in one image.

2.2. Class-agnostic Object Counting

Some earlier class-agnostic methods are the General Matching Network (GMN) [7] and Few-shot Adaptation and Matching Network (FamNet) [14] that require annotated exemplars, which may be difficult or impractical to obtain. They also require lots of training data. In the case of GMN [7], to avoid having to gather annotated exemplars, Lu et al. turned to the ILSVRC object detection video dataset, which contained approximately 4500 videos which totalled to over 1M annotated frames. However, the nature of video data is that many frames would be highly similar and contain a limited number of possible object classes.

Further research into exemplar-free counting yielded models such as RepRPN [13], although it could only count dominant object classes. Other exemplar-free counting methods turned toward additional text input in place of exemplars. Xu et al. [15] used a generative variational autoencoder (VAE) to create exemplars from text prompts before selecting class-relevant patches via K-nearest neighbours to generate a final count. CLIP-Count [5] made use of the vision-language model (VLM) CLIP [12] to generate text and image embeddings, which were sent through their Hierarchical Text-patch Interaction Module, before the density map was regressed using a decoder based on convolutional neural networks (CNN). CounTX [1] adopted a similar approach to CLIP-Count by employing CLIP [12], but fed the text and image embeddings into a Feature Interaction Module that contained cross-attention mechanisms. Still, these methods require text prompts that although less demanding than annotated exemplars, require user interaction that prevents automatic counting.

2.3. Image Representation in Object Counting

A popular option amongst counting models that employ VLMs is CLIP [12] due to the semantic nature of its embeddings and zero-shot capabilities. As CLIP consists of a jointly trained image encoder and text encoder, it can pair image and text extremely well. Nevertheless, CLIP-based models have been seen to lack spatial awareness necessary for object counting due to CLIP features not capturing sufficient structural information during pre-training [1]. In fact, CLIP cannot accurately return images with the correct number of objects requested for within a text prompt. Paiss et al. [10] observed that CLIP focused on parts of its input containing nouns, but ignored other parts like numbers. In a separate paper, Paiss et al. [11] proposed that CLIP's limitations were due to unspecified object quantities in the captions of CLIP's training data and lack of focus on quantity during training.

Other non-text approaches recognise counting as a repetition-recognition problem. Methods that make use of CNNs are less effective due to localised receptive fields that prevent the model from understanding the global context [4]. Architectures that contain attention mechanisms like ViT do not suffer from this and are more effective at counting. CounTR [6] employed a ViT-based image encoder that divided the image into patches and included additional positional encoding. Hobley et al. [4] made use of a self-supervised method, RCC, which employed a 'teacher' and a 'student' network, by utilising the pre-trained DINO [2] model and directly regressing for count predictions. More recently, DINOv2 [9], a self-supervised ViT model with 1B parameters has been released which produces better all-purpose features for downstream tasks than the SOTA. DINOv2 provides a variety of image encoders pre-trained on large datasets that do not require finetuning and have been shown to allow for better performance. Following from [2], DINOv2 also employs a student-teacher network that is elaborated on in Section 3.1.1.

3. Method

In this paper, we address the challenging problem of visual object counting without reference images or arbitrary exemplars provided by the users.

Overview. Our objective is to train a generalized visual object counter capable of performing effectively on a test

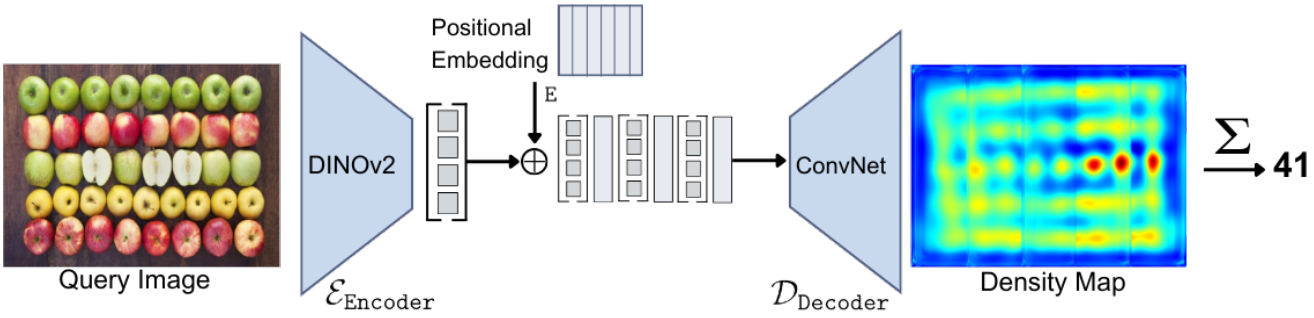


Figure 2: Detailed overview of the proposed model architecture: Our approach utilizes a self-supervised visual encoder, specifically DINOv2, to process the query image. The extracted image features are then summed with the positional embeddings which serves as the query vectors. Subsequently, we use a CNN-based decoder to decode these vectors and up-scales them to generate the corresponding density map. Finally, the total object count is determined by integrating the values across the density map.

set without any exemplars. The training dataset is defined as follows:

$$D_{train} = (X_1, y_1), ..., (X_N, y_N) \tag{1}$$

where $X_i \in R^{H \times W \times 3}$ represents the input image, and $y_i \in R^{H \times W \times 1}$ is a binary spatial density map. This map features 1's at the central location of the objects, denoting their presence, and 0's elsewhere where objects are absent. The total object count can be ascertained by performing a spatial summation over the density map. We aim to develop a visual object counter with a generalized approach, designed to work on a test set, even when provided with no exemplars, denoted as follows:

$$D_{test} = (X_{N+1}), ..., (X_M) \tag{2}$$

We employ the DINOv2 [9] vision foundation model to execute self-supervised knowledge distillation, enhancing the model's ability to understand the feature space within the images. To preserve the spatial relationships inherent within the input images, we incorporate positional embeddings into the visual feature representations. The feature vectors, which now embody both the distilled knowledge and spatial context, are subsequently processed through a ConvNet-based decoder. This transformation yields a density map that is subsequently refined by a linear regression layer, resulting in a single-channel heatmap.

3.1. Architecture

In this section, we outline the architecture of our proposed model, which can be viewed in Figure 2. We use the DINOv2 visual encoder, denoted as $\mathcal{E}_{\texttt{Encoder}}$, that converts the input image $\mathcal{X}_i$ into a feature map, $\mathcal{F}_{\texttt{DINO}}$. To this feature map, positional embeddings, denoted as $\texttt{E}$, are applied, yielding a spatial feature representation $\mathcal{F}_{\texttt{E}}$. A ConvNet-based decoder $\mathcal{D}_{\texttt{Decoder}}$, consist of four upsampling stages decode $\mathcal{F}_{\texttt{E}}$ into one-channel heatmap y_i with the original image size that signifies the density of objects within the image. The overall computational sequence is represented in Equation 3.

$$y_i = \mathcal{D}_{\texttt{Decoder}}(\texttt{E} + (\mathcal{E}_{\texttt{Encoder}}(\mathcal{X}_i))) \tag{3}$$

3.1.1 Visual Encoder

To learn a comprehensive and informative feature space without relying on labels, we employ a self-supervised knowledge distillation model as outlined in [9]. The concept of knowledge distillation, introduced in [3], involves training a student network, denoted as g_{θ_s}, to mimic the output of a specified teacher network g_{θ_t}. Both networks are characterized by their respective parameters, θ_s and θ_t. Given an input image x, both networks yield probability distributions across K dimensions, represented by P_s for the student and P_t for the teacher. The probability P is derived by applying the softmax function to the output of the network g. In [2], this process specify as follows,

$$P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^{K} \exp(g_{\theta_s}(x)^{(k)}/\tau_s}} \tag{4}$$

with $\tau_s > 0$ that controls the sharpness of the output distribution. According to [2], with a predetermined teacher network g_{θ_t}, we aim to align the distributions by minimizing the cross-entropy loss between P_t and P_s, detailed as follows.

$$\min_{\theta_s} \sum_{x \in \{x_1^g, x_2^g\}} \sum_{\substack{x' \in V \\ x' \neq x}} H(P_t(x), P_s(x')) \tag{5}$$

where $H(a, b) = -a \log b$. More precisely, from a provided image, we produce a set of V comprising various views. This set includes two global views x_1^g and x_2^g, along with multiple local views at a reduced resolution. This configuration is termed the foundational parametrization of DINO. In our approach, for an input image represented by $\mathcal{X}_i$, our method

generates a feature map $\mathcal{F}_{\texttt{DINO}}$ as described in Equation 6 below.

$$\mathcal{F}_{\texttt{DINO}} = \mathcal{E}_{\texttt{Encoder}}(\mathcal{X}_i) \in \mathbb{R}^{M \times D} \tag{6}$$

This feature map is the result of encoding process $\mathcal{E}_{\texttt{Encoder}}$, transforming the input image into $M \times D$ dimensions.

3.1.2 Positional Embedding

The model needs to understand where each token comes from in the original image. We use positional embeddings $\texttt{E}$ to help maintain the spatial relationships that were present in the original 2D image. The spatial structure of the image is encoded in these embeddings and is used by the model to understand the relative or absolute positions of the tokens before feeding them into the upsampling CNN layers. We add this positional embedding to the visual features to produce spatial feature representation $\mathcal{F}_{\texttt{E}}$, represented as follows.

$$\mathcal{F}_{\texttt{E}} = \texttt{E} + \mathcal{F}_{\texttt{DINO}} \tag{7}$$

The positional embeddings $\texttt{E}$ can provide an initial encoding of spatial features that can be further processed by convolutional layers. While CNNs can extract and leverage spatial hierarchies due to their nature, the positional embeddings can provide a starting point that encodes absolute or relative positions within the image.

3.1.3 Decoder

The spatial feature representation, which is the output after adding visual encoder features with positional embeddings are subsequently transformed into 2D feature maps, matching the input image's original resolution. We then utilize a progressive up-sampling design, where the vector sequence is reconfigured into a dense feature map before being refined by a decoder based on ConvNet, denoted as $\mathcal{D}_{\texttt{Decoder}}$. This process can be represented as the following.

$$y_i = \mathcal{D}_{\texttt{Decoder}}(\mathcal{F}_{\texttt{E}}) \in \mathbb{R}^{H \times W \times 1} \tag{8}$$

Our ConvNet consists of four up-sampling stages, each comprising a convolutional layer followed by a bilinear interpolation. Concluding this sequence of up-sampling, we employ a linear layer to serve as the density regressor, which yields a single-channel heatmap y_i indicative of object density.

4. Experiment & Evaluation

4.1. Dataset & Metrics

Dataset. The FSC-147 [14] dataset is widely used in class-agnostic counting, and contains 6135 images across 147 object classes. The number of objects per image ranges from 7 to 3731, and averages at 56. FSC-147 provides a corresponding binary density map y_i for each training image X_i in D_{train}.

Metrics. We use the standard Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to evaluate the performance of the model.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|C_i - C_i^{GT}|, \tag{9}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(C_i - C_i^{GT})^2} \tag{10}$$

Here, N refers to the number of images in the test set while C_i and C_i^{GT} refer to the predicted count and ground truth of the i-th image. The count C_i is obtained from summing the values across the density map y_i.

4.2. Implementation Details

Training. The input images are resized to a dimension of 256 x 256 pixels. Prior to model input, the images are transformed using center-cropping (224 x 224 pixels), horizontal and vertical flips, random rotations within 30 degrees, and finally, normalization. We employ the AdamW optimization method with a batch size set of 8 and a learning rate of 6.25e-6. We train our model with the RTX 3070 Ti Mobile GPU. After training for 200 epochs, the model exhibiting the lowest MAE on the validation set is chosen.

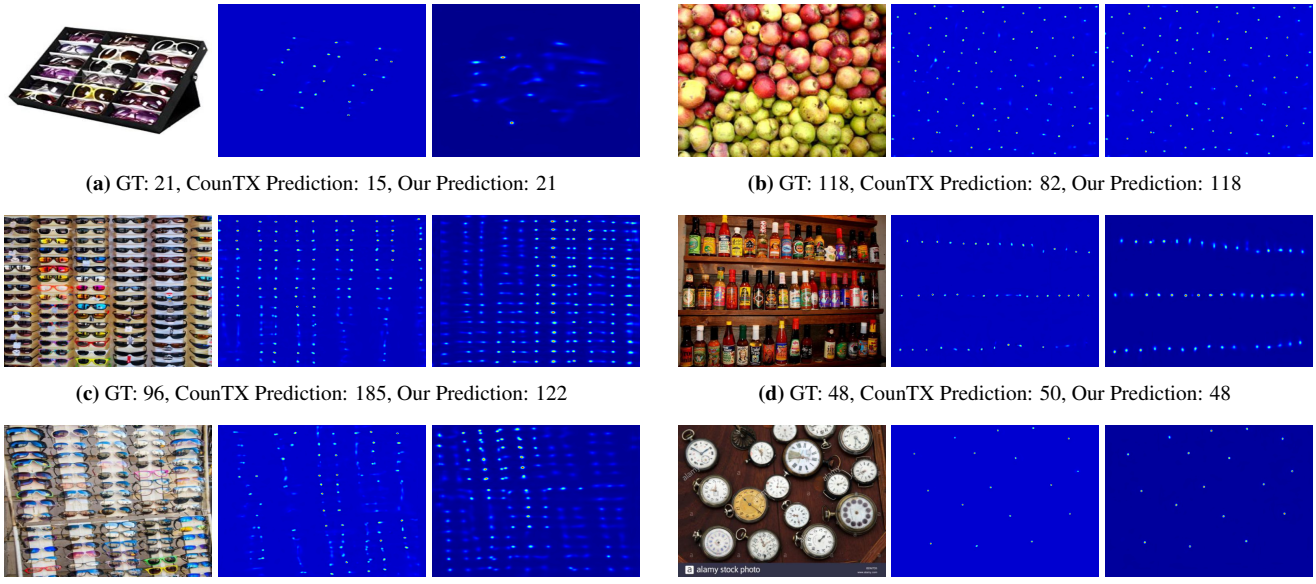
Inference. During inference time, we follow the same procedure as in [6]. Specifically, a square sliding window is adopted to scan over the image, using a stride of 128 pixels. Similar to the training phase, each image is resized to 256 x 256 pixels and subsequently normalized prior to being input into the model. For handling overlapping regions in the density map, we apply the approach described in [6].

4.3. Quantitative Results

	Val Set		Test Set	
Method	MAE	RMSE	MAE	RMSE
GMN [7]	39.02	106.06	37.86	141.39
FamNet [14]	32.15	98.75	32.37	131.46
CounTX* [1]	17.70	63.61	15.73	106.88
CounTR [6]	18.07	71.84	14.71	106.87
RCC [4]	17.49	58.81	17.12	104.53
Ours	**19.11**	**65.58**	**19.06**	**118.45**

Table 1: Comparison of (zero-shot) methods on the FSC-147 dataset. * indicates CounTX was not zero-shot, but had object classes specified using text of class names.

The evaluation results for our method on FSC-147 [14] are compared against other existing methods in Table 1. While not outperforming methods like CounTR [6] and RCC



(a) GT: 21, CounTX Prediction: 15, Our Prediction: 21

(b) GT: 118, CounTX Prediction: 82, Our Prediction: 118

(c) GT: 96, CounTX Prediction: 185, Our Prediction: 122

(d) GT: 48, CounTX Prediction: 50, Our Prediction: 48

(e) GT: 79, CounTX Prediction: 143, Our Prediction: 83

(f) GT: 17, CounTX Prediction: 15, Our Prediction: 17

Figure 3: Qualitative results of our model on FSC-147. Each sub-figure consists of the original image (left), CounTX's density map (middle), and our model's density map (right). The predicted counts from both models and ground truth (GT) counts have also been provided.

[4], our model performs reasonably well across both validation and test sets. The most notable results are observed in the qualitative results in the next Section 4.4, where our model demonstrates the ability to count more complex objects that CounTR fails to.

4.4. Qualitative Results

Some qualitative results can be found in Figure 3, where the predicted counts from our model and from CounTX [1] have also been provided for comparison. It can be observed that the density map produced by our model is unlike the density map produced by CounTX that resembles point-level annotations.

Interestingly, the DINOv2 [9] image encoder employed in our model appears to capture structural cues that CounTX's image encoder was unable to. Revisiting the example of glasses in Figure 4, CounTX would count each lens as a single object as is evident from its resultant density map. Similarly, CounTR [6] displayed similar behaviour to CounTX and would consistently double count. As detailed in [6], test-time augmentation was required to correct the predicted count. However, our model was able to predict a similar count to the ground truth without any such augmentation, possibly due to the encoder being able to encapsulate the structure of glasses within the image features and detect repetitions of this structure that corresponded to multiple glasses.

4.5. Exploration of Model Limitations

Our model is unfortunately limited in its ability to count objects that are not distinct, such as in the case of a Lego piece in Figure 5. Possibly due to the large number of objects in close proximity and without distinct borders, the resulting density map shows that there are no clear distinctions between objects. Several images on which the model performed badly were identified and are depicted in Figure 6.

In fact, when these four images were excluded from their respective image sets, the model's performance improved significantly, as shown in Table 2.

	Val Set		Test Set	
Our Model	MAE	RMSE	MAE	RMSE
Without exclusions	17.66	60.77	17.49	114.99
Excluding 4 images	15.64	48.12	13.14	33.05

Table 2: C.

5. Conclusion

In this work, we trained a generalised visual object counter that performs without given exemplars. Taking inspiration from the CounTR [6] architecture and building upon CounTX [1], we proposed a separate visual encoder using the self-supervised model, DINOv2 [9], to obtain more informative features. The feature vectors were decoded using

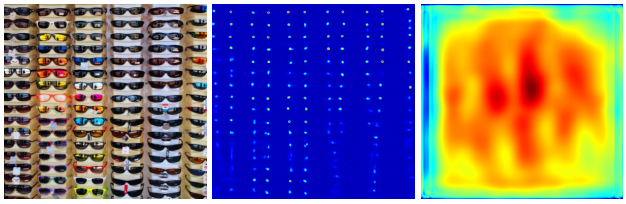


Figure 4: GT: 96, CounTX Prediction: 185, Our Prediction: 98
For the image given (left), CounTX's density map (middle) and our model's density map (right).

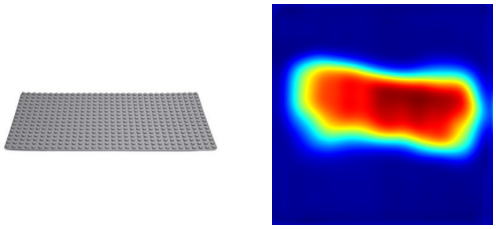
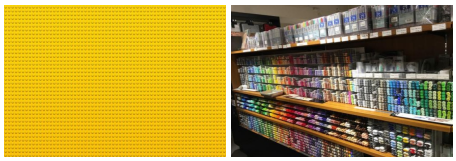
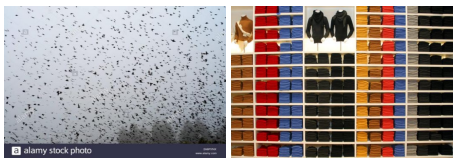


Figure 5: GT: 512, Our Prediction: 400
Failure case. Our model is limited in its ability to count objects that are not distinct. For the image given (left), the density map produced by our model (right).



(a) Images from the test set.



(b) Images from the validation set.

Figure 6: Examples of images where our model performed badly due to the number of objects and lack of distinction between them.

a ConvNet-based decoder to a density map, which was integrated to produce a predicted object count. To evaluate our model against existing methods, we used the standard counting dataset, FSC-147, and showed that our model achieves comparable results to other zero-shot methods. Most notably, the model was found to be more effective than CounTR at detecting and counting more structurally complex objects.

References

[1] Niki Amini-Naieni, Kiana Amini-Naieni, Tengda Han, and Andrew Zisserman. Open-world text-specified object counting. 2023. 1, 2, 4, 5

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2, 3

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 3

[4] Michael Hobley and Victor Prisacariu. Learning to count anything: Reference-less class-agnostic counting with weak supervision. *arXiv preprint arXiv:2205.10203*, 2022. 2, 4, 5

[5] Ruixiang Jiang, Lingbo Liu, and Changwen Chen. Clip-count: Towards text-guided zero-shot object counting. *arXiv preprint arXiv:2305.07304*, 2023. 2

[6] Chang Liu, Yujie Zhong, Andrew Zisserman, and Weidi Xie. Countr: Transformer-based generalised visual counting. *arXiv preprint arXiv:2208.13721*, 2022. 1, 2, 4, 5

[7] Erika Lu, Weidi Xie, and Andrew Zisserman. Class-agnostic counting. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 669–684. Springer, 2019. 1, 2, 4

[8] T Nathan Mundhenk, Goran Konjevod, Wesam A Sakla, and Kofi Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 785–800. Springer, 2016. 1, 2

[9] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 3, 5

[10] Roni Paiss, Hila Chefer, and Lior Wolf. No token left behind: Explainability-aided image classification and generation. In *European Conference on Computer Vision*, pages 334–350. Springer, 2022. 2

[11] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. *arXiv preprint arXiv:2302.12066*, 2023. 2

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[13] Viresh Ranjan and Minh Hoai Nguyen. Exemplar free class agnostic counting. In *Proceedings of the Asian Conference on Computer Vision*, pages 3121–3137, 2022. 2

[14] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3394–3403, 2021. 2, 4

[15] Jingyi Xu, Hieu Le, Vu Nguyen, Viresh Ranjan, and Dimitris Samaras. Zero-shot object counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15548–15557, 2023. 2

[16] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2