This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Learning to Count without Annotations

Lukas Knobel
University of Amsterdam
lukasknbl@gmail.com

Tengda Han*
University of Oxford
htd@robots.ox.ac.uk

Yuki M. Asano*
University of Amsterdam
y.m.asano@uva.nl

Abstract

While recent supervised methods for reference-based object counting continue to improve the performance on benchmark datasets, they have to rely on small datasets due to the cost associated with manually annotating dozens of objects in images. We propose UnCounTR, a model that can learn this task without requiring any manual annotations. To this end, we construct "Self-Collages", images with various pasted objects as training samples, that provide a rich learning signal covering arbitrary object types and counts. Our method builds on existing unsupervised representations and segmentation techniques to successfully demonstrate for the first time the ability of reference-based counting without manual supervision. Our experiments show that our method not only outperforms simple baselines and generic models such as FasterRCNN and DETR, but also matches the performance of supervised counting models in some domains.[1]

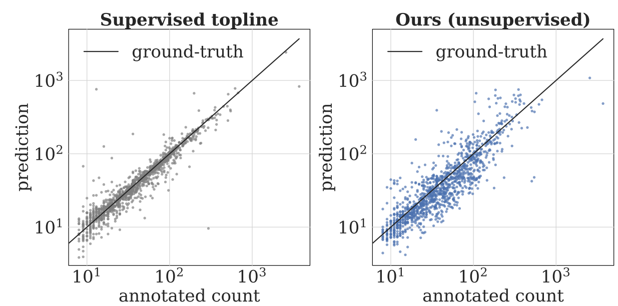


Figure 1. **CounTR *vs.* our proposed UnCounTRv2.** Our counting model, UnCounTRv2, is trained without any labels and manual counting annotations. It generalizes well from its up to 19 pseudo-counts encountered during training to the significantly higher numbers in the FSC-147 test set. This demonstrates that learning to count is possible without annotations.

1. Introduction

Cognitive neuroscientists speculate that visual counting, especially for a small number of objects, is a pre-attentive and parallel process [12, 55], which can help humans and animals make prompt decisions [43]. Accumulating evidence shows that infants and certain species of animals can differentiate between small numbers of items [11, 12, 41] and as young as 18-month-old infants have been shown to develop counting abilities [53]. These findings indicate that the ability of visual counting may emerge very early or even be inborn in humans and animals.

On the non-biological side, recent developments in computer vision have been tremendous. The state-of-the-art computer vision models can classify thousands of image classes [19, 27], detect various objects [66], or segment almost anything from an image [26]. Partially inspired by how babies learn to see the world [54], some of the recent well-performing models are trained with self-supervised learning methods, whereby a learning signal for neural networks is constructed without the need for manual annotations [15, 20]. The pretrained visual representations from such methods have demonstrated superior performances on various downstream visual tasks, like image classification and object detection [7, 20, 21]. Moreover, self-supervised learning signals have been shown to be sufficient for successfully learning image groupings [57, 59] and even object and semantic segmentations without any annotations [7, 62]. Motivated by these, we ask in this paper whether visual counting might also be solvable without relying on human annotations.

The current state-of-the-art visual counting methods, *e.g.* CounTR [31], typically adapt pretrained visual representations to the counting task by using a considerable size of human annotations. However, we conjecture that the existing visual representations are already *strong enough* to perform counting, even *without* any manual annotations.

In this paper, we design a straightforward self-supervised training scheme to teach the model 'how to count', by pasting a number of objects on a background image, to make a Self-Collage. Our experiments show that when construct-

*Equal senior contribution.
[1]Code: https://github.com/lukasknobel/SelfCollages

ing the Self-Collages carefully, this training method is effective enough to leverage the pretrained visual representation on the counting task, even approaching other methods that require manually annotated counting data. For the visual representation, we use the self-supervised pretrained DINO features [7], which have been shown to be useful and generalisable for a variety of visual tasks like segmenting objects [34, 67]. Note that the DINO model is also trained without manual annotations, thus our entire pipeline does not require annotated datasets.

To summarise, this paper focuses on the objective of *training a reference-based counting model without any manual annotation*. The following contributions are made: (i) We propose a simple yet effective data generation method to construct 'Self-Collages', which pastes objects onto an image and gets supervision signals for free. (ii) We leverage self-supervised pretrained visual features from DINO and develop UnCounTR, a transformer-based model architecture for counting. (iii) The experiments show that our method trained without manual annotations not only outperforms baselines and generic models like FasterRCNN and DETR, but also matches the performance of supervised counting models.

2. Related work

Counting with object classes. The class-specific counting methods are trained to count instances of a single class of interest, such as cars [23, 36] or people [28, 28, 32, 47, 48]. These methods require retraining to apply them to new object classes [36]. In addition, some works rely on class-specific assumptions such as the proportionality of features and counts [24], or the distribution [48] or shapes [56] of objects, which cannot be easily adapted.

By contrast, class-agnostic approaches are not designed with a specific object class in mind. Early work by Zhang et al. [63] proposes the salient object subitizing task, where the model is trained to count by classification of $\{0, 1, 2, 3, 4+\}$ salient objects regardless of their classes. Other reference-less methods like Hobley and Prisacariu [22] frame counting as a repetition-recognition task and aim to automatically identify the objects to be counted. An alternative approach of class-agnostic counting requires a prior of the object type to be counted in the form of reference images, also called 'exemplars', each containing a single instance of the desired class [14, 29, 33, 45, 50, 60, 61].

Counting with different methods. Categorised by the approach taken to obtain the final count, counting methods can be divided into classification, detection and regression-based methods.

Classification-based approaches predict a discrete count for a given image [63]. The classes either correspond to single numbers or potentially open-ended intervals. Thus, predictions are limited to the pre-defined counts and a generalisation to new ranges is by design not possible.

An alternative is detection-based methods [23]. By predicting bounding boxes for the counted instances and deriving a global count based on their number, these methods are unlike classification-based approaches not constrained to predefined count classes. While the bounding boxes can facilitate further analysis by explicitly showing the counted instances, the performance of detection-based approaches deteriorates in high-density settings [22].

Lastly, regression-based methods predict a single number and can be further divided into scalar and density-based approaches. Scalar methods directly map an input image to a single scalar count corresponding to the number of objects in the input [22]. Density-based methods on the contrary predict a density map for a given image and obtain the final count by integrating over it [9, 14, 32, 33, 48, 49]. Similar to detection-based approaches, these methods allow locating the counted instances which correspond to the local maxima of the density map but have the added benefit of performing well in high-density applications with overlapping objects [33]. The recent work CounTR [31] is a class-agnostic, density-based method which is trained to count both with and without exemplars using a transformer decoder structure with learnable special tokens.

Self-supervised learning. Self-supervised learning (SSL) has shown its effectiveness in many computer vision tasks. Essentially, SSL derives the supervision signal from the data itself rather than manual annotations. The supervision signal can be found from various "proxy tasks" like colorization [64], spatial ordering or impainting [15, 21, 37, 42], temporal ordering [18, 35], contrasting similar instances [10, 20, 39], clustering [3, 6], and from multiple modalities [1, 44]. Another line of SSL methods is knowledge distillation, where one smaller student model is trained to predict the output of the other larger teacher model [4, 8, 25]. BYOL [17] design two identical models but train one model to predict the moving average of the other as the supervision signal. Notably, DINO [7] is trained in a similar way but using Vision Transformers (ViTs) as the visual model [16], and obtains strong visual features. Simply thresholding the attention maps and using the resulting masks yields high-quality image segmentations. Follow-up works [34, 51, 67] demonstrate the semantic segmentation quality can be further improved by applying some lightweight training or post-processing of DINO features. For counting in SSL, Noroozi et al. [38] use feature "counting" as a proxy task to learn representations. In this work, we focus on counting itself and explore approaches to teach the model to count without manual supervision.

3. Method

We tackle the task of counting objects given some exemplar crops within an image. With an image dataset $\mathcal{D} =$

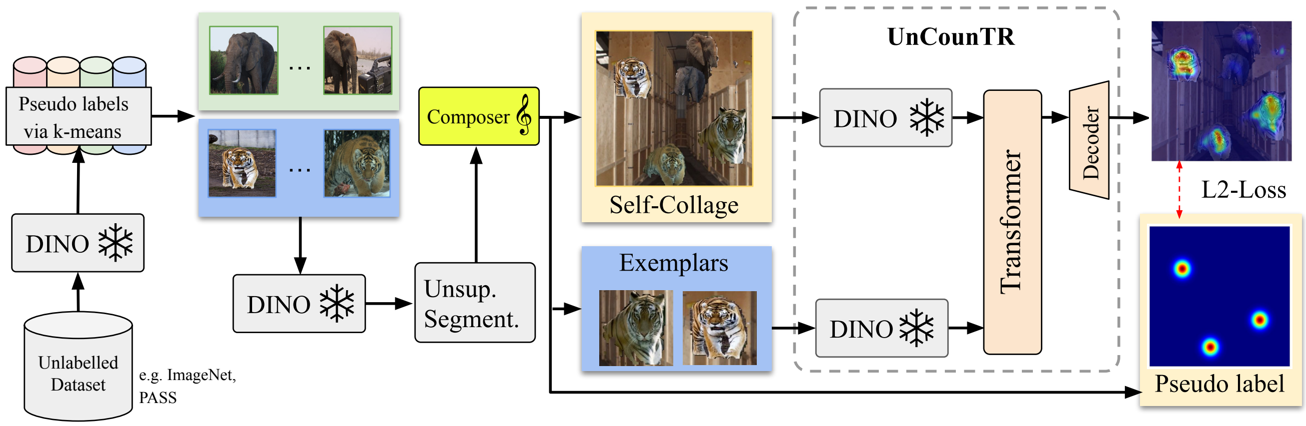


Figure 2. **Method overview.** Our method leverages the strong coherence of deep clusters to provide pseudo-labelled images which are used to construct a self-supervised counting task. The composer utilises self-supervised segmentations for pasting a set of objects onto a background image and UnCounTR is trained to count these when provided with unsupervised exemplars.

$\{(\boldsymbol{I}_1, \mathcal{S}_1, \mathbf{y}_1), \ldots, (\boldsymbol{I}_i, \mathcal{S}_i, \mathbf{y}_i)\}$, where $\boldsymbol{I}_i \in \mathbb{R}^{H \times W \times 3}$ denotes image i, $\mathcal{S}_i = \{\boldsymbol{E}_i{}^1, ..., \boldsymbol{E}_i{}^E\}$ represent the E visual exemplars of shape $\boldsymbol{E}_i{}^e \in \mathbb{R}^{H' \times W' \times 3}$, and $\mathbf{y}_i \in \mathbb{R}^{H \times W}$ is the ground-truth density map, the counting task can be written as:

$$\hat{\mathbf{y}}_i = f_\Theta(\boldsymbol{I}_i, \mathcal{S}_i) \tag{1}$$

Here, the predicted density map $\hat{\mathbf{y}}_i$ indicates the objects to be counted, as specified by the exemplars $\mathcal{S}_i$, such that $\sum_{kl} \hat{\mathbf{y}}_{i_{kl}}$ yields the overall count for the image $\boldsymbol{I}_i$. We are interested in training a neural network f parameterised by Θ to learn how to count based on the exemplars $\mathcal{S}_i$. For the supervised counting methods [31, 33], the network parameters Θ can be trained with the '(`prediction`, `ground-truth`)' pairs: $(\hat{\mathbf{y}}_i, \mathbf{y}_i)$. However, for self-supervised counting, the learning signal $\mathbf{y}_i$ is not obtained from manual annotations, but instead from the data itself.

In this section, we introduce two essential parts of our method: we start by presenting our data generation method for counting in Section 3.1, that is the construction of the tuple $(\boldsymbol{I}_i, \mathcal{S}_i, \mathbf{y}_i)$; then, we explain the UnCounTR model, *i.e.* f_Θ in Equation (1), in Section 3.2. An overview of our method is provided in Figure 2.

3.1. Constructing Self-Collages

A key component of self-supervised training is the construction of a supervision signal without any manual annotations. In this paper, we generate training samples by pasting different images on top of a background. This is a well-known technique [2, 65], which next to its conceptual simplicity provides great flexibility regarding the number and type of pasted images. While other works combine annotated training images to enrich the training set [22, 31], we use this idea to construct the *whole* training set including unsupervised proxy labels, yielding self-supervised collages, or Self-Collages for short. Their generation process is described by a composer module g, which forms a distribution $g(\mathcal{O}, \mathcal{B}) = p(\tilde{\boldsymbol{I}}, \mathcal{S}, \mathbf{y} \mid \mathcal{O}, \mathcal{B})$ of constructed images $\tilde{\boldsymbol{I}} \in \mathbb{R}^{H \times W \times 3}$ along with unsupervised exemplars $\mathcal{S}$ and labels $\mathbf{y}$. The process is based on two sets of unlabelled images $\mathcal{O}$ and $\mathcal{B}$ for object and background images. $\mathcal{S} = \{\boldsymbol{E}\}^E$, $\boldsymbol{E} \in \mathbb{R}^{H' \times W' \times 3}$ is a set of $E \in \mathbb{N}$ exemplars and $\mathbf{y} \in \mathbb{R}^{H \times W}$ corresponds to the density map of $\tilde{\boldsymbol{I}}$.

The composer module g first randomly selects the number of distinct object categories $n_c \sim U[t_{\text{min}}, t_{\text{max}}]$ the first of which is taken as the target cluster. t_{min} and t_{max} control the minimum and maximum number of categories in a Self-Collage. To reduce the risk of overfitting to construction artefacts, we always construct images with n_{max} objects and change the associated number of target objects $\sum_{ij} \mathbf{y}_{ij}$ solely by altering the number of objects in the target cluster. This way, the number of pasted objects and, therefore, the number of artefacts is independent of the target count. The number of target objects $n_0 = n \sim U[n_{\text{min}}, n_{\text{max}} - n_c + 1]$ has an upper-bound lower than n_{max} to guarantee that there is at least one object of each of the n_c types. For all other clusters, the number of objects is drawn from a uniform distribution of points on the $n_c - 1$ dimensional polytope with $L1$-size of $n_{\text{max}} - n$ ensuring that the total number of objects equals n_{max}. Further details, including the pseudocode for the composer module, are shown in the Appendix.

Unsupervised categories. We obtain unsupervised object categories by first extracting feature representations for all samples in $\mathcal{O}$ using a pretrained DINO ViT-B/16 backbone [7] d and subsequently running k-means with K clusters:

$$c(\boldsymbol{I}) = \text{k-means}_K[(d(\mathcal{O}))^{\texttt{CLS}}](\boldsymbol{I}), \tag{2}$$

where $c(\boldsymbol{I})$ is the unsupervised category for image $\boldsymbol{I}$ constructed using the final `CLS`-embedding of d. For each of the n_c clusters, a random, unique cluster c_i, $i \in [0, n_c - 1]$ is chosen from all K clusters where c_0 is the target cluster.

Image selection. Next, random sets of images $\mathcal{I}_i \subset \mathcal{O}$ are picked from their unsupervised categories c_i, such that $|\mathcal{I}_i| = n_i$ and $c(\boldsymbol{I}) = c_i \ \forall \boldsymbol{I} \in \mathcal{I}_i$. We denote the union of these sets as $\mathcal{I} = \bigcup_{i=0}^{n_c-1} \mathcal{I}_i$. In addition, we sample one image $\boldsymbol{I}_\text{b}$ from another dataset $\mathcal{B}$, which is assumed to not contain salient objects to serve as the background image.

3.1.1 Construction strategy

Here we detail the Self-Collage construction. First, the background image $\boldsymbol{I}_\text{b}$ is reshaped to the final dimensions $H \times W$ and used as a canvas on which the modified images $\mathcal{I}$ are pasted. To mimic natural images that typically contain similarly sized objects, we first randomly pick a mean object size $d_\text{mean} \sim U[d_\text{min}, d_\text{max}]$. Subsequently, the target size of the pasted objects is drawn independently for each $\boldsymbol{I} \in \mathcal{I}, \boldsymbol{I} \in \mathbb{R}^{d_\text{h} \times d_\text{w} \times 3}$ from a uniform distribution $d_\text{paste} \sim U[(1-\sigma) \cdot d_\text{mean}, (1+\sigma) \cdot d_\text{mean}]$ where $\sigma \in (0, 1)$ controls the diversity of objects sizes in an image. After resizing $\boldsymbol{I}$ to $\boldsymbol{I}_\text{r} \in \mathbb{R}^{d_\text{paste} \times d_\text{paste} \times 3}$, the image is pasted to a random location on $\boldsymbol{I}_\text{b}$. This location is either a position where previously no object has been pasted, or any location in the constructed image, potentially leading to overlapping images. By default, we will use the latter.

Segmented pasting. Since pasting the whole image $\boldsymbol{I}_\text{r}$ might violate the assumption of having a single object and results in artefacts by also pasting the background of $\boldsymbol{I}_\text{r}$, we introduce an alternative using self-supervised segmentations. This method, which we will use by default, uses an unsupervised segmentation method [51] to obtain a noisy foreground segmentation $\mathbf{s} \in [0, 1]^{d_\text{h} \times d_\text{w}}$ for $\boldsymbol{I}$. Instead of pasting the whole image, the segmentation $\mathbf{s}$ is used to only copy its foreground. Additionally, having access to $\mathbf{s}$, this method can directly control the size of the pasted objects rather than the pasted images. To do that, we first extract the object in $\boldsymbol{I}$ by computing the Hadamard product $\boldsymbol{I}_\text{object} = \text{cut}\,(\boldsymbol{I} \circ \mathbf{s})$ where the operation "cut" removes rows and columns that are completely masked out. In the next step, $\boldsymbol{I}_\text{object} \in \mathbb{R}^{h_\text{object} \times w_\text{object} \times 3}$ is resized so that its maximum dimension equals d_paste before pasting it onto $\boldsymbol{I}_\text{b}$.

Exemplar selection. To construct the exemplars used for training, we exploit the information about how the sample was constructed using g: The set of E exemplars $\mathcal{S}$ is simply obtained by filtering for pasted objects that belong to the target cluster c_0 and subsequently sampling E randomly. Then, for each of them, a crop of $\tilde{\boldsymbol{I}}$ is taken as exemplar after resizing its spatial dimensions to $H' \times W'$.

3.1.2 Density map construction

To train our counting model, we construct an unsupervised density map $\mathbf{y}$ for each training image $\tilde{\boldsymbol{I}}$. It needs to have the following two properties: i) it must sum up to the overall count of objects that we are counting and ii) it must have high values at object locations. To this end, we create $\mathbf{y}$ as a simple density map of Gaussian blobs as done in supervised density-based counting methods [14, 31]. For this, we use the bounding box for each pasted target image $\boldsymbol{I} \in \mathcal{I}_0$ and place Gaussian density at the centre and normalise it to one.

3.2. UnCounTR

Model architecture. UnCounTR's architecture is inspired by CounTR [31]. To map an input image $\boldsymbol{I}$ and its exemplars $\mathcal{S}$ to a density map $\hat{\mathbf{y}}$, the model consists of four modules: image encoder Φ, exemplar encoder Ψ, feature interaction module f_fim, and decoder f_dec. An overview of this architecture can be seen in Figure 2. The image encoder Φ encodes an image $\boldsymbol{I}$ into a feature map $\mathbf{x} = \Phi(\boldsymbol{I}) \in \mathbb{R}^{h \times w \times d}$ where h, w, d denote the height, width, and channel depth. Similarly, each exemplar $\boldsymbol{E} \in \mathcal{S}$ is projected to a single feature vector $\mathbf{z} \in \mathbb{R}^d$ by taking a weighted average of the grid features. Instead of training a CNN for an exemplar encoder as CounTR does, we choose $\Psi = \Phi$ to be the frozen DINO visual encoder weights. Reusing these weights, the exemplar and image features are in the same feature space. The feature interaction module (FIM) f_fim enriches the feature map $\mathbf{x}$ with information from the encoded exemplars $\mathbf{z}_j,\ j \in \{1, ..., E\}$ with a transformer decoder structure. Finally, the decoder f_dec takes the resulting patch-level feature map of the FIM as input and upsamples it with 4 convolutional blocks, ending up with a density map of the same resolution as the input image. Please refer to the Appendix for the full architectural details.

UnCounTR supervision. UnCounTR is trained using the Mean Squared Error between model prediction and pseudo labels. Given a Self-Collage $\tilde{\boldsymbol{I}}$, exemplars $\mathcal{S}$, and density map $\mathbf{y}$, the loss $\mathcal{L}$ for an individual sample is computed using the following equation where $f_\Theta(\tilde{\boldsymbol{I}}, \mathcal{S})_{ij}$ is UnCounTR's spatially-dense output at location (i, j):

$$\mathcal{L} = \frac{1}{H \cdot W} \sum_{ij} (\mathbf{y}_{ij} - f_\Theta(\tilde{\boldsymbol{I}}, \mathcal{S})_{ij})^2 \qquad (3)$$

4. Experiments

4.1. Implementation Details

Datasets. To construct Self-Collages, we use **ImageNet-1k** [13] and **SUN397** [58]. ImageNet-1k contains 1.2M mostly object-centric images spanning 1000 object categories. SUN397 contains 109K images for 397 scene categories like 'cliff' or 'corridor'. Note that the object or scene category information is never used in our method. We assume that images from ImageNet-1k contain a single salient object and images from SUN397 contain no salient objects to serve as sets $\mathcal{O}$ and $\mathcal{B}$. Based on this, g randomly selects images from SUN397 as background images and picks objects from ImageNet-1k. Although Imagenet-

1k and SUN397 contain on average 3 and 17 objects respectively [30], these assumptions are still reasonable for our data construction. Additional ablations show that even explicitly violating the assumption of having no salient objects in $\mathcal{B}$ by using $\mathcal{O} = \mathcal{B}$ only have a small effect on performance, indicating the robustness of our method against violations (see Appendix). An example of a Self-Collage with 5 objects can be seen in Figure 2. Examples of both datasets and a more detailed discussion of these assumptions are included in the Appendix.

To evaluate the counting ability, we use the standard **FSC-147** dataset [45], which contains 6135 images covering 147 object categories, with counts ranging from 7 to 3731 and an average count of 56 objects per image. For each image, the dataset provides at least three randomly chosen object instances with annotated bounding boxes as exemplars. To analyse the counting ability in detailed count ranges, we partition the FSC-147 test set into three parts of roughly 400 images each, resulting in *FSC-147-*{*low,medium,high*} subsets, each covering object counts from 8-16, 17-40 and 41-3701 respectively. Unless otherwise stated, we evaluate using 3 exemplars.

Additionally, we also use the **CARPK** [23] and the Multi-Salient-Object (**MSO**) dataset [63] for evaluation. CARPK consists of 459 images of parking lots. Each image contains between 2 and 188 cars, with an average of 103. MSO contains 1224 images covering 5 categories: {0,1,2,3,4+} salient objects with bounding box annotations. This dataset is largely imbalanced as 338 images contain zero salient objects and 20 images contain at least 4. For evaluation, we removed all samples with 0 counts, split the 4+ category into exact counts, and chose only one annotated object as exemplar.

Construction details. We configure the composer module to construct training samples using objects of $K = 10,000$ different clusters. To always have objects of a target and a non-target cluster in each image, we set $t_{\text{min}} = t_{\text{max}} = 2$. Since the minimum number of target objects in an image limits the maximum number of exemplars available during training, we set $n_{\text{min}} = 3$, the maximum is $n_{\text{max}} = 20$. Finally, we choose $d_{\text{min}} = 15$, $d_{\text{max}} = 70$, and $\sigma = 0.3$ to obtain diverse training images with objects of different sizes as confirmed by visual inspection of the generated Self-Collages (see the Appendix).

Training and inference details. During training, the images are cropped and resized to 224×224 pixels. The exemplars are resized to 64×64 pixels. For each batch, we randomly draw the number of exemplars E to be between 1 and 3. We follow previous work [31] by scaling the loss, in our case by a factor of 3,000, and randomly dropping 20% of the non-object pixels in the density map to decrease the imbalance of object and non-object pixels. By default, we use an AdamW optimizer and a cosine-decay learning rate schedule with linear warmup and a maximum learning rate of 5×10^{-4}. We use a batch size of 128 images. Each model is trained on an Nvidia A100 GPU for 50 epochs of $10,000$ Self-Collages each, which takes about 4 hours. At inference, we use a sliding window approach similar to Liu et al. [31], see the Appendix for details. For evaluation metrics, we report Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) following previous work [14, 31]. We also report Kendall's τ coefficient, which is a rank correlation coefficient between the sorted ground-truth counts and the predicted counts.

Baselines. To verify the effectiveness of our method, we introduce a series of baselines to compare with. (1) **Average** baseline: the prediction is always the average count of the FSC-147 training set, which is 49.96 objects. (2) **Connected components** baseline: for this, we use the final-layer attention values from the `CLS`-token of a pretrained DINO ViT-B/8 model. To derive a final count, we first threshold the attention map of each head to keep p_{att} percent of the attention mask. Subsequently, we consider each patch to where at least n_{head} attention heads attend to belong to an object. The number of connected components in the resulting feature map which cover more than p_{size} percent of the feature map is taken as prediction. To this end, we perform a grid search with almost 800 configurations of different value combinations for the three thresholds p_{att}, n_{head}, and p_{size} on the FSC-147 training set and select the best configuration based on the MAE. The specific values tested can be found in the Appendix. (3) **FasterRCNN** baseline: we run the strong image detection toolbox FasterRCNN [46] with a score threshold of 0.5 on the image, which predicts a number of object bounding boxes. Then the object count is obtained by parsing the detection results by taking the total number of detected bounding boxes. Just like the connected components baseline, this model is applied to images resized to 384×384 pixels. (4) **DETR** baseline: similar to FasterRCNN, we evaluate the detection model DETR [5] on the counting task. Here, we resize the images to 800×800 pixels to better match DETR's evaluation protocol.

4.2. Comparison against baselines

In Table 1, we compare against multiple baselines on the three splits of FSC-147. As the connected components cannot leverage the additional information of the few-shot samples provided, we try to make the comparison as fair as possible, by testing almost 800 threshold parameters on the FSC-147 training set. While this yields a strong baseline, we find that our method of learning with Self-Collages more than halves the MAE on the *low* split, despite using a similar visual backbone. Next, we compare against the FasterRCNN and DETR object detectors which, unlike UnCounTR, are trained in a supervised fashion. Even though DETR outperforms UnCounTR in terms of RMSE on FSC-

Method	FSC-147 *low*			FSC-147 *medium*			FSC-147 *high*		
	MAE↓	RMSE↓	$\tau\uparrow$	MAE↓	RMSE↓	$\tau\uparrow$	MAE↓	RMSE↓	$\tau\uparrow$
Average	37.71	37.79	-	22.74	23.69	-	68.88	213.08	-
Conn. Comp.	14.71	18.59	0.14	14.19	17.90	0.16	69.77	210.54	0.17
FasterRCNN	7.06	8.46	-0.03	19.87	22.25	-0.04	109.12	230.35	-0.06
DETR	6.92	**8.20**	0.07	19.33	21.56	-0.08	109.34	**162.88**	-0.07
UnCounTR (ours)	**5.60**	10.13	**0.27**	**9.48**	**12.73**	**0.34**	**67.17**	189.76	**0.26**
σ(5 runs)	±0.48	±0.84	±0.02	±0.19	±0.33	±0.02	±1.03	±1.38	±0.01

Table 1. **Comparison to baselines.** Evaluation on different FSC-147 test subsets. "Conn. Comp." refers to connected components.

		FSC-147		FSC-147 *low*	
Frozen Φ	Frozen Ψ	MAE↓	RMSE↓	MAE↓	RMSE↓
✗	✗	37.64	**126.91**	7.99	13.37
✓	✗	36.94	130.22	9.10	14.69
✓	✓	**35.77**	130.34	**5.60**	**10.13**

(a) Keeping both image encoder Φ and exemplar encoder Ψ frozen works best.

		FSC-147		FSC-147 *low*	
Segm.	Overlap.	MAE↓	RMSE↓	MAE↓	RMSE↓
✓	✗	36.47	**128.97**	8.84	14.54
✗	✓	37.46	136.03	6.33	11.55
✓	✓	**35.77**	130.34	**5.60**	**10.13**

(b) Using both segmented objects and allowing for overlapping objects works well.

	FSC-147		FSC-147 *low*	
$n_{\max}$	MAE↓	RMSE↓	MAE↓	RMSE↓
50	**30.17**	**115.42**	6.65	13.60
20	35.77	130.34	**5.60**	**10.13**

(c) The number of pasted objects correlates with performance.

	FSC-147		FSC-147 *low*	
#Exemplars	MAE↓	RMSE↓	MAE↓	RMSE↓
0 - 3	36.28	131.44	6.23	10.33
1 - 3	**35.77**	**130.34**	**5.60**	**10.13**

(d) Training our method strictly in the 1-3 exemplar setting works best.

Table 2. **Ablations.** We ablate various components of our model architecture and Self-Collage construction method. Default settings are highlighted in grey.

		FSC-147		FSC-147 *low*	
Architecture	Pretraining	MAE↓	RMSE↓	MAE↓	RMSE↓
ViT-B/8	Leopart [67]	36.04	126.75	5.72	10.67
ViT-B/8	DINO [7]	38.16	132.09	6.10	9.79
ViT-B/16	DINO [7]	35.77	130.34	5.60	10.13

Table 3. **Combining Self-Collages with different backbones.** We can apply our method to any recent state-of-the-art pretrained network and achieve strong performances.

Methods	Val		Test	
	MAE↓	RMSE↓	MAE↓	RMSE↓
CounTR [31]	13.13	49.83	11.95	91.23
LOCA [14]	**10.24**	**32.56**	**10.79**	**56.97**
UnCounTR (ours)	36.93	106.61	35.77	130.34
σ(5 runs)	±0.49	±1.46	±0.60	±0.94

Table 4. **Comparison to supervised models.** We evaluate on the val and test split of FSC-147.

147 *low* and *high*, we find that our method still outperforms all baselines on 7 out of 9 measures. This is despite the additional advantages such as access to the FSC-147 training distribution for these baselines. The gap between most baselines and UnCounTR is the smallest on the *high* split, suggesting limits to its generalisation which we will explore further in the Appendix. There, we also provide a qualitative analysis of some failure cases of FasterRCNN.

4.3. Ablation Study

We analyse the various components of our method in Tables 2a to 2d.

Keeping backbone frozen works best. In Table 2a, we evaluate the effect of unfreezing the last two backbone blocks shared by Φ and Ψ. In addition, we test training a CNN-based encoder Ψ from scratch similar to Liu et al. [31]. We find that keeping both frozen, *i.e.* $\Phi = \Psi =$ const. works best, as it discourages the visual features to adapt to potential artefacts in our Self-Collages.

Benefit of self-supervised exemplars. In Table 2d we evaluate the effect of including the zero-shot counting task (*i.e.* counting all salient objects) as done in other works [14, 31]. However, we find that including this task even with only a 25% chance leads to a lower performance. This is likely due to the zero-shot task overfitting our Self-Collages and finding short-cuts, such as pasting artefacts, as we relax the constraint introduced in Section 3.1.1 and vary the number of pasted objects to match the desired target. We therefore train our model with 1-3 exemplars and show how we can still conduct semantic zero-shot counting in Section 4.7.

Maximum number of pasted objects. Next, we evaluate the effect of varying $n_{\max}$, the maximum number of objects pasted, as shown in Table 2c. Pasting up to 50 objects yields overall better performance on the full FSC-147 test dataset which contains on average 66 objects. While this shows that the construction of Self-Collages can be successfully scaled to higher counts, we find that pasting with 20 objects already achieves good performance with shortened

Method	FSC-147 *low* (8-16 objects)				FSC-147 *medium* (17-40 objects)				FSC-147 *high* (41-3701 objects)			
	MAE↓	RMSE↓	Δ	$\tau\uparrow$	MAE↓	RMSE↓	Δ	$\tau\uparrow$	MAE↓	RMSE↓	Δ	$\tau\uparrow$
CounTR	6.58	48.11		**0.42**	**4.48**	**10.02**		**0.60**	**20.50**	**126.95**		**0.80**
UnCounTR (ours)	**5.60**	**10.13**	-37.9	0.27	9.48	12.73	+2.7	0.34	67.17	189.76	+62.8	0.26
σ(5 runs)	±0.48	±0.84		±0.02	±0.19	±0.33		±0.02	±1.03	±1.38		±0.01

Table 5. **Comparison to CounTR.** Evaluation results on the FSC-147 *low* and *medium* test subsets.

Method	MSO		
	MAE↓	RMSE↓	$\tau\uparrow$
Conn. Comp. Baseline	10.67	12.88	0.12
CounTR	2.34	8.12	0.36
UnCounTR (ours)	1.07	2.32	0.45
σ(5 runs)	±0.06	±0.25	±0.01

Table 6. **Evaluation on the MSO [63] dataset.**

Method	CARPK		
	MAE↓	RMSE↓	$\tau\uparrow$
Conn. Comp. Baseline	49.62	54.50	0.56
CounTR	19.62	29.70	0.57
UnCounTR (ours)	30.35	35.67	0.54
σ(5 runs)	±2.26	±2.36	±0.06

Table 7. **Evaluation on the CARPK [23] dataset.**

construction times and so use this setting.

Segmented pasting works best. From Table 2b, We find that the best construction strategy involves pasting the self-supervised segmentations, regardless of their overlap with other objects. We simply store the segmentations and combine these to create diverse Self-Collages at training time.

Compatible with various pretrained backbones. In Table 3, we show the effect of using different frozen weights for the visual encoder. Across ViT-B models with varying patch sizes and pretrainings, from Leopart's [67] spatially dense to the original DINO's [7] image-level pretraining, we find similar performances across our evaluations. This shows that our method is compatible with various architectures and pretrainings and will likely benefit from the continued progress in self-supervised representation learning which is supported by our findings based on DINOv2 [40] presented in the Appendix. We do not find any benefit in going from patch size 16 to 8 (similar to Siméoni et al. [52]) so we use the faster DINO ViT-B/16 model for the rest of the paper.

4.4. Benchmark comparison

Next, we compare with previous methods on three datasets: FSC-147 [45], MSO [63], and CARPK [23]. Table 4 shows the results on the validation and test split of the FSC-147 dataset. In contrast to our approach, which investigates the feasibility of learning to count without any reliance on supervised data, these methods heavily depend on manual annotations. Hence, the comparison is certainly not fair: (1) our method does not use manually annotated counting data, also has never seen any FSC-147 training images, (2) our training samples based on Self-Collages only cover 3-19 target objects, whereas the full test set of FSC-147 has 66 objects on average. Evaluating our method on the full set of FSC-147 is actually evaluating the transferring ability as discussed in more detail in the Appendix.

We choose CounTR [31] for further analysis because of its similar architecture to UnCounTR. To fairly compare with this method, we trisect the FSC-147 test set as described in Section 4.1. Table 5 shows the evaluation results on the three partitions. It is remarkable that our method outperforms supervised CounTR on the *low* partition, especially on the RMSE metric (10.13 vs 48.11). Also, our method is not far from CounTR's performance on the *medium* partition (*e.g.* RMSE 12.7 vs 10.02), showing a reasonable transferring ability since the model is trained with up to 19 object counts only.

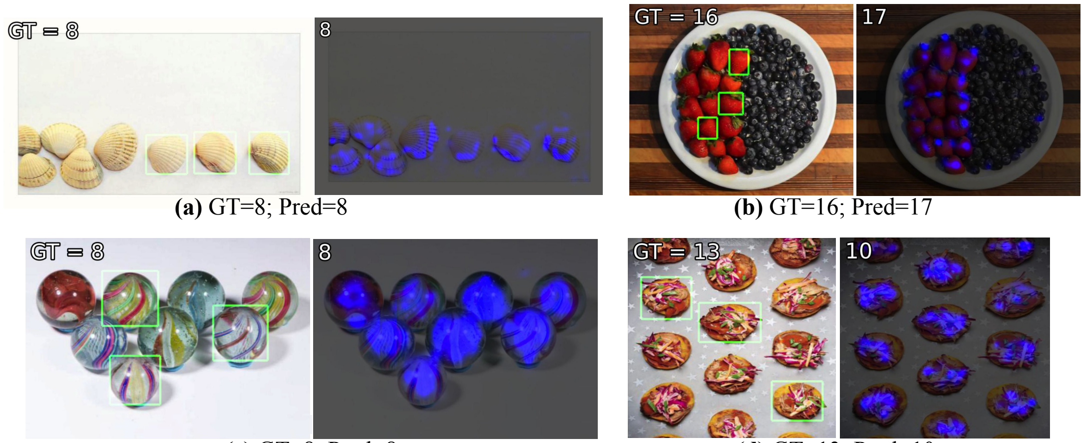
Additionally, Table 6 compares our method with the connected components baseline and CounTR on a subset of MSO as described in Section 4.1. The results show our self-supervised method outperforms the supervised CounTR model by a large margin on the lower counts tasks. In Table 7, the performance on the CARPK dataset is shown. We use the same CounTR model finetuned on the FSC-147 dataset as before to compare the generalisation across datasets. While the MAE of CounTR almost doubles compared to the results on the FSC-147 test set, UnCounTR's performance is more stable, improving by 15%.

4.5. Qualitative results and limitations

Figure 3 shows a few qualitative results to demonstrate our method's effectiveness and its limitations. The model is able to correctly predict the number of objects for clearly separated and slightly overlapping instances (*e.g.* Subfigures **a** and **c**). The model also successfully identifies the object type of interest, *e.g.* in Subfigure **b** the density map correctly highlights the strawberries rather than blueberries. One limitation of our model is images where the number of objects is ambiguous. For example, in Subfigure **d**, the prediction missed a few burgers which are possibly the ones partially shown on the edge. However, it is also difficult for humans to decide which of those objects should be counted.

4.6. Improving upon UnCounTR

Having established UnCounTR and compared it to several methods, we now explore three ways to further improve its



(a) GT=8; Pred=8 **(b)** GT=16; Pred=17 **(c)** GT=8; Pred=8 **(d)** GT=13; Pred=10

Figure 3. **Qualitative examples of UnCounTR's predictions.** We show predictions on four images from the FSC-147 test set, the green boxes represent the exemplars. Our predicted count is the sum of the density map rounded to the nearest integer.

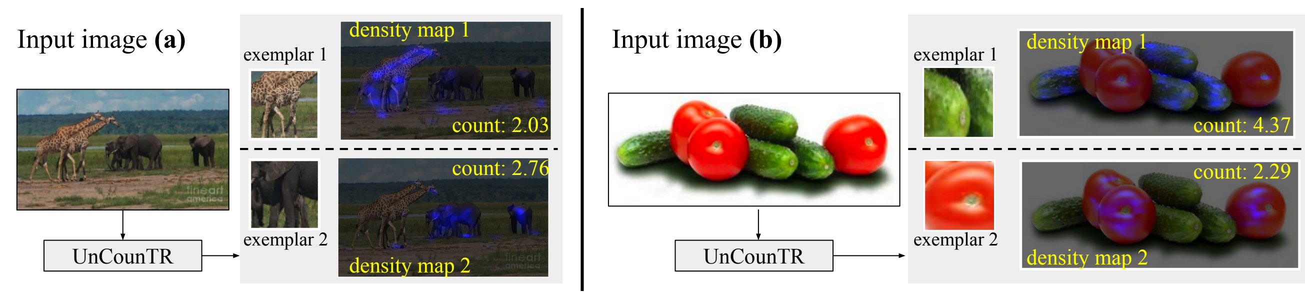


Figure 4. **Self-supervised semantic counting.** In this setting, the model proposes the exemplars by itself and then performs reference-based counting.

performance: (1) Integrating the DINOv2 [40] backbone improves the MAE particularly on the FSC-147 *low* set by 27% (2) Using cluster similarity to create more challenging Self-Collages reduces the MAE on FSC-147 to 31.06 (3) Refining high-count predictions further improves performance yielding the final **UnCounTRv2** model. It achieves an MAE of 28.67 on FSC-147 and 3.88 on FSC-147 *low*. For a comprehensive discussion, please refer to the Appendix. See Figure 1 for a visual comparison with CounTR.

4.7. Self-supervised semantic counting

In this last section, we explore the potential of UnCounTR for more advanced counting tasks. In particular, we test whether our model can not only unsupervisedly count different kinds of objects in an image, but also determine the categories by itself, a scenario we refer to as *semantic* counting. To this end, we use a simple pipeline that picks an area surrounding the maximum in the input image's `CLS`-attention map as the first input and refines it to obtain an exemplar. Next, UnCounTR predicts the number of objects in the image based on the self-constructed exemplar. Finally, the locations that have been detected using this procedure are removed from the initial attention map and if the remaining attention values are high enough, the process is repeated to count objects of another type. We provide the details of this method in the Appendix. In Figure 4, we demonstrate results on two difficult real images.

5. Conclusion

In this work, we have introduced UnCounTR, showcasing, for the first time, the capability of reference-based counting without relying on human supervision. To this end, our method constructs Self-Collages, a simple unsupervised way of generating proxy learning signals from unlabeled data. Our results demonstrate that by utilising an off-the-shelf unsupervisedly pretrained visual encoder, we can learn counting models that can even outperform strong baselines such as DETR and achieve similar performances to dedicated counting models such as CounTR on CARPK, MSO, and various splits of FSC-147. Finally, we have shown that our model can unsupervisedly identify multiple exemplars in an image and count them, something no supervised model can yet do. This work opens a wide space for extending unsupervised visual understanding beyond simple image-level representations to more complex tasks previously out of reach, such as scene graphs or unsupervised semantic instance segmentations.

References

[1] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. *NeurIPS*, 33:25–37, 2020. 2

[2] Relja Arandjelović and Andrew Zisserman. Object discovery with a copy-pasting gan. *arXiv preprint arXiv:1905.11369*, 2019. 3

[3] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *ICLR*, 2020. 2

[4] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 5

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018. 2

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 1, 2, 3, 6, 7

[8] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *NeurIPS*, 30, 2017. 2

[9] Hao Chen, Yangzhun Zhou, Jun Li, Xiu-Shen Wei, and Liang Xiao. Self-supervised multi-category counting networks for automatic check-out. *IEEE Transactions on Image Processing*, 31:3004–3016, 2022. 2

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 2

[11] Hank Davis and Rachelle Pérusse. Numerical competence in animals: Definitional issues, current evidence, and a new research agenda. *Behavioral and Brain Sciences*, 11(4):561–579, 1988. 1

[12] Stanislas Dehaene. *The number sense: How the mind creates mathematics*. Oxford University Press, 2011. 1

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 4

[14] Nikola Djukic, Alan Lukezic, Vitjan Zavrtanik, and Matej Kristan. A low-shot object counting network with iterative prototype adaptation. *arXiv preprint arXiv:2211.08217*, 2022. 2, 4, 5, 6

[15] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015. 1, 2

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33:21271–21284, 2020. 2

[18] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCV Workshops*, pages 0–0, 2019. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 1, 2

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 1, 2

[22] Michael Hobley and Victor Prisacariu. Learning to count anything: Reference-less class-agnostic counting with weak supervision. *arXiv preprint arXiv:2205.10203*, 2022. 2, 3

[23] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H Hsu. Drone-based object counting by spatially regularized regional proposal network. In *ICCV*, pages 4145–4153, 2017. 2, 5, 7

[24] Takayuki Katsuki, Tetsuro Morimura, and Tsuyoshi Idé. Unsupervised object counting without object recognition. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3627–3632. IEEE, 2016. 2

[25] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*, 2016. 2

[26] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[28] Dingkang Liang, Jiahao Xie, Zhikang Zou, Xiaoqing Ye, Wei Xu, and Xiang Bai. Crowdclip: Unsupervised crowd counting via vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2893–2903, 2023. 2

[29] Hui Lin, Xiaopeng Hong, and Yabin Wang. Object counting: You only need to look at one. *arXiv preprint arXiv:2112.05993*, 2021. 2

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 5

[31] Chang Liu, Yujie Zhong, Andrew Zisserman, and Weidi Xie. Countr: Transformer-based generalised visual counting. *BMVC*, 2022. 1, 2, 3, 4, 5, 6, 7

[32] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *CVPR*, pages 7661–7669, 2018. 2

[33] Erika Lu, Weidi Xie, and Andrew Zisserman. Class-agnostic counting. In *ACCV*, pages 669–684. Springer, 2018. 2, 3

[34] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *CVPR*, pages 8364–8375, 2022. 2

[35] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, pages 527–544. Springer, 2016. 2

[36] 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 *ECCV*, pages 785–800. Springer, 2016. 2

[37] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84. Springer, 2016. 2

[38] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, pages 5898–5906, 2017. 2

[39] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[40] 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. 7, 8

[41] Mario Pahl, Aung Si, and Shaowu Zhang. Numerical cognition in bees and other insects. *Frontiers in psychology*, 4: 162, 2013. 1

[42] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 2

[43] Manuela Piazza and Stanislas Dehaene. From number neurons to mental arithmetic: The cognitive neuroscience of number sense. *The cognitive neurosciences, 3rd edition, ed. MS Gazzaniga*, pages 865–77, 2004. 1

[44] 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 *ICML*, pages 8748–8763. PMLR, 2021. 2

[45] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *CVPR*, pages 3394–3403, 2021. 2, 5, 7

[46] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015. 5

[47] Deepak Babu Sam, Neeraj N Sajjan, Himanshu Maurya, and R Venkatesh Babu. Almost unsupervised learning for dense crowd counting. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8868–8875, 2019. 2

[48] Deepak Babu Sam, Abhinav Agarwalla, Jimmy Joseph, Vishwanath A Sindagi, R Venkatesh Babu, and Vishal M Patel. Completely self-supervised crowd counting via distribution matching. *arXiv preprint arXiv:2009.06420*, 2020. 2

[49] Zenglin Shi, Pascal Mettes, and Cees GM Snoek. Counting with focus for free. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4200–4209, 2019. 2

[50] Zenglin Shi, Ying Sun, and Mengmi Zhang. Training-free object counting with prompts. *arXiv preprint arXiv:2307.00038*, 2023. 2

[51] Gyungin Shin, Samuel Albanie, and Weidi Xie. Unsupervised salient object detection with spectral cluster voting. In *CVPR*, pages 3971–3980, 2022. 2, 4

[52] Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *BMVC*, 2021. 7

[53] Virginia Slaughter, Shoji Itakura, Aya Kutsuki, and Michael Siegal. Learning to count begins in infancy: Evidence from 18 month olds' visual preferences. *Proceedings of the Royal Society B: Biological Sciences*, 278(1720):2979–2984, 2011. 1

[54] Linda Smith and Michael Gasser. The development of embodied cognition: Six lessons from babies. *Artificial life*, 11 (1-2):13–29, 2005. 1

[55] Lana M Trick and Zenon W Pylyshyn. Why are small and large numbers enumerated differently? a limited-capacity preattentive stage in vision. *Psychological review*, 101(1): 80, 1994. 1

[56] Jordan R Ubbens, Tewodros W Ayalew, Steve Shirtliffe, Anique Josuttes, Curtis Pozniak, and Ian Stavness. Autocount: Unsupervised segmentation and counting of organs in field images. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 391–399. Springer, 2020. 2

[57] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *ECCV*, 2020. 1

[58] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492. IEEE, 2010. 4

[59] Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. Clusterfit: Improving generalization of visual representations. In *CVPR*, pages 6509–6518, 2020. 1

[60] Shuo-Diao Yang, Hung-Ting Su, Winston H Hsu, and Wen-Chin Chen. Class-agnostic few-shot object counting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 870–878, 2021. 2

[61] Zhiyuan You, Kai Yang, Wenhan Luo, Xin Lu, Lei Cui, and Xinyi Le. Few-shot object counting with similarity-aware feature enhancement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6315–6324, 2023. 2

[62] Andrii Zadaianchuk, Matthaeus Kleindessner, Yi Zhu, Francesco Locatello, and Thomas Brox. Unsupervised semantic segmentation with self-supervised object-centric representations. In *ICLR*, 2023. 1

[63] Jianming Zhang, Shugao Ma, Mehrnoosh Sameki, Stan Sclaroff, Margrit Betke, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Salient object subitizing. In *CVPR*, pages 4045–4054, 2015. 2, 5, 7

[64] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pages 649–666. Springer, 2016. 2

[65] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, et al. X-paste: Revisit copy-paste at scale with clip and stablediffusion. *arXiv preprint arXiv:2212.03863*, 2022. 3

[66] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, pages 350–368. Springer, 2022. 1

[67] Adrian Ziegler and Yuki M Asano. Self-supervised learning of object parts for semantic segmentation. In *CVPR*, pages 14502–14511, 2022. 2, 6, 7