Appendix: 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

This Appendix provides further insights into training counting methods in a self-supervised manner using Self-Collages. In Appendix A, we describe the implementation in more detail. In Appendix B, we provide further information about the Self-Collages including a discussion of the underlying assumptions. Finally, in Appendix C, we analyse UnCounTR's performance in different settings.

A. Further implementation details

A.1. Model architecture

Image encoder For the image encoder Φ, we employ a ViT-B/16 pretrained using the DINO approach [3] as the backbone. It consists of 12 transformer blocks with 12 heads each and uses fixed, sinusoidal position embeddings. The transformer operates on $d = 768$ dimensions and increases the hidden dimensions within the two-layer MLPs after each attention block by a factor of four to 3072 dimensions. Each linear layer in the MLP is followed by the GELU non-linearity. More information can be found in the original work from Caron et al. [3]. By default, we freeze all 86M parameters of the backbone.

Exemplar encoding To encode an exemplar $\boldsymbol{E}$ with fixed spatial dimensions $H' \times W'$ into a single feature vector $\mathbf{z} \in \mathbb{R}^d$, we first pass it through the backbone of the exemplar encoder Ψ to obtain a feature map $\mathbf{x}_\text{E} = \Psi(\boldsymbol{E}) \in \mathbb{R}^{h \times w \times d}$ where $\Psi = \Phi$. The final representation $\mathbf{z}$ is derived by computing the weighted sum of $\mathbf{x}_\text{E}$ across the spatial dimensions where the weight of each patch is determined by the attention in the final `CLS`-attention map of Ψ averaged over the heads.

Feature interaction module We follow the architecture proposed by Liu et al. [9] which uses 2 transformer blocks with 16 heads to modify the image embeddings by using self-attention. In addition to self-attention, each block utilises cross-attention where the keys and values are based on the encoded exemplars. Since this transformer operates on 512-dimensional feature vectors, $\mathbf{x}$ and $\mathbf{z}_j,\ j \in \{1, ..., E\}$ are projected to these dimensions using a linear layer. To give the feature interaction module direct access to positional information, fixed, sinusoidal position embeddings are added to the feature map. Similar to the image encoder, the MLPs increase the dimensionality four times to 2048. This results in 8.8M parameters.

Decoder The decoder is built based on 4 convolutional layers that upscale the patch-level features to the original resolution to obtain the final density map. It has a total of 3.0M parameters. All blocks contain a convolutional layer with 256 output channels and a kernel size of 3×3. Each of them is followed by group normalisation with 8 groups and a ReLU non-linearity. The last block has a final convolutional layer with 1×1 filters that reduce the number of channels to 1 to match the desired output format. After each block, the spatial resolution is doubled using bilinear interpolation. The result is a density map $\hat{\mathbf{y}} \in \mathbb{R}^{H \times W}$ with the same resolution as the input image.

A.2. Composer module details

We always place the images $\mathcal{I}_0$ of the target cluster c_0 on top of the non-target images $\mathcal{I}_i,\ i \in \{1, ..., n_c - 1\}$. This reduces the noise of Self-Collages since target objects can only be occluded by other target objects but never completely hidden by non-target images. Crucially, this also guarantees that exemplars are never covered by non-target objects which could alter

*Equal senior contribution.

the desired target cluster. While the resulting Self-Collages exhibit pasting artefacts (see Figure 3), previous work has shown that eliminating these artefacts by applying a blending method does not improve the performance on downstream tasks such as object detection [5] and instance segmentation [7, 16]. Hence, we focus on simple pasting to keep the complexity of the construction process low and rely on breaking correlations between the number of artefacts and the target count of an image by pasting a constant number of images as discussed in the paper. Algorithm 1 shows the pseudocode for the composer module.

If we use the "no-overlap" setup, which prevents objects are pasted on top of each other, and an image cannot be pasted without overlapping with an already copied image, the construction process is restarted with new random sizes and locations. To guarantee the termination of g, potential overlaps between images are ignored if the construction process fails 20 times.

Density map construction We construct the density map by placing unit density at the centre of each pasted target image $\boldsymbol{I} \in \mathcal{I}_0$. Following Djukic et al. [4], we apply a Gaussian filter to blur the resulting map. Its kernel size and standard deviation vary per image and are based on the average bounding box size divided by 8.

Algorithm 1 The composer module

Require: t_{min} ▷ minimum number of clusters
t_{max} ▷ maximum number of clusters
n_{min} ▷ minimum number of target objects
n_{max} ▷ maximum number of objects
K ▷ total number of clusters
E ▷ number of exemplars
$\mathcal{O}$ ▷ set of object images
$\mathcal{B}$ ▷ set of background images

 # create clusters and select cluster sizes
$\mathcal{C} \leftarrow$ k-means$_K[(d(\mathcal{O}))^{\text{CLS}}]$ ▷ cluster $\mathcal{O}$ using K clusters
$n_c \sim U[t_{\text{min}}, t_{\text{max}}]$ ▷ select the number of clusters
$n_0 = n \sim U[n_{\text{min}}, n_{\text{max}} - n_c + 1]$ ▷ select the number of objects in the target cluster
for i in range(1, $n_c - 1$) **do** ▷ select the number of objects in the other clusters

$$n_i \sim U\left[1, n_{\text{max}} - \overbrace{\sum_{j=0}^{i-1} n_j}^{\text{previous clusters}} - \underbrace{n_c + i + 1}_{\text{remaining clusters}}\right]$$

end for
$n_{n_c-1} \leftarrow n_{\text{max}} - \sum_{i=0}^{n_c-2} n_i$ ▷ set the number of objects in the final cluster

 # select images
for i in range(0, n_c) **do**
$c_i \leftarrow$ random_cluster($\mathcal{C} \setminus \bigcup_{j=0}^{i-1}\{c_j\}$) ▷ select a random, unique cluster as the i^{th}cluster
$\mathcal{I}_i \leftarrow$ select(n_i,c_i,$\mathcal{O}$) ▷ select n_i random images in cluster c_i from $\mathcal{O}$
end for
$\mathcal{I} \leftarrow \bigcup_{i=0}^{n_c-1} \mathcal{I}_i$
$\tilde{\boldsymbol{I}} \leftarrow$ select(1, $\mathcal{B}$) ▷ select a random background image from $\mathcal{B}$

 # compose the image and pseudo ground-truth
B $\leftarrow$ [ ] ▷ initialise an empty list for the object bounding boxes
for $\boldsymbol{I}$ in $\mathcal{I}$ **do** ▷ iterate over all object images
$s \leftarrow$ get_random_size($\boldsymbol{I}$, $\mathcal{I}$) ▷ get a random size, which is correlated for all images in $\mathcal{I}$
$p \leftarrow$ get_random_position(s) ▷ get a random position for the current image
$\boldsymbol{I}_r \leftarrow$ resize($\boldsymbol{I}$, s) ▷ resize $\boldsymbol{I}$, if using segmentations, this involves cutting the object
$\tilde{\boldsymbol{I}} \leftarrow$ paste($\tilde{\boldsymbol{I}}$, $\boldsymbol{I}_{\text{r}}$, p) ▷ paste $\boldsymbol{I}_{\text{r}}$ into $\tilde{\boldsymbol{I}}$ at position p, if $\boldsymbol{I}$ is a target object, place it on top
if $\boldsymbol{I} \in \mathcal{I}_0$ **then**
b $\leftarrow$box(p, s) ▷ create the bounding box of the current object if $\boldsymbol{I}$ is a target object
B.append(**b**) ▷ append the bounding box to the list of all object boxes
end if
end for
$\mathcal{S} \leftarrow$ crop_exemplars($\tilde{\boldsymbol{I}}$, **B**, E) ▷ create E exemplar crops
y $\leftarrow$ create_density_map(**B**) ▷ create the density map
return $\tilde{\boldsymbol{I}}$, $\mathcal{S}$, **y**

A.3. Connected components baseline

We evaluate the connected components baseline on the FSC-147 training set to find the best values for its three thresholds p_{att}, n_{head}, and p_{size}. To this end, we run an exhaustive grid search by testing all combinations of the following settings:

$$p_{\text{att}} \in \{0.1, 0.2, ..., 0.9, 0.95, 0.99\}, \tag{1}$$

$$n_{\text{head}} \in \{1, ..., 12\}, \tag{2}$$

$$p_{\text{size}} \in \{0, 0.01, 0.02, 0.05, 0.1, 0.2\} \tag{3}$$

This results in 792 configurations, where we pick the setup with the lowest MAE on the whole FSC-147 training set which results in the following thresholds: $p_{\text{att}} = 0.7$, $n_{\text{head}} = 10$, and $p_{\text{size}} = 0$.

A.4. Inference details

We follow Liu et al. [9] in our evaluation procedure. Specifically, we resize the inference image to a height of 384 keeping the aspect ratio fixed and scan the resulting image with a window of size 384×384 and a stride of 128 pixels. The density maps of the overlapping regions are averaged when aggregating the count of the entire image. If the image contains very small objects, defined as at least one exemplar with a width and height of less than 10 pixels, the image is divided into a 3×3 grid. Each of the 9 tiles is then resized to a height of 384 pixels and processed independently. The prediction for the original image is obtained by combining the individual predictions. Additionally, we apply the same test-time normalisation: We normalise the predicted count by the average sum of the density map areas that correspond to the exemplars if it exceeds a threshold of 1.8.

A.5. Self-supervised semantic counting details

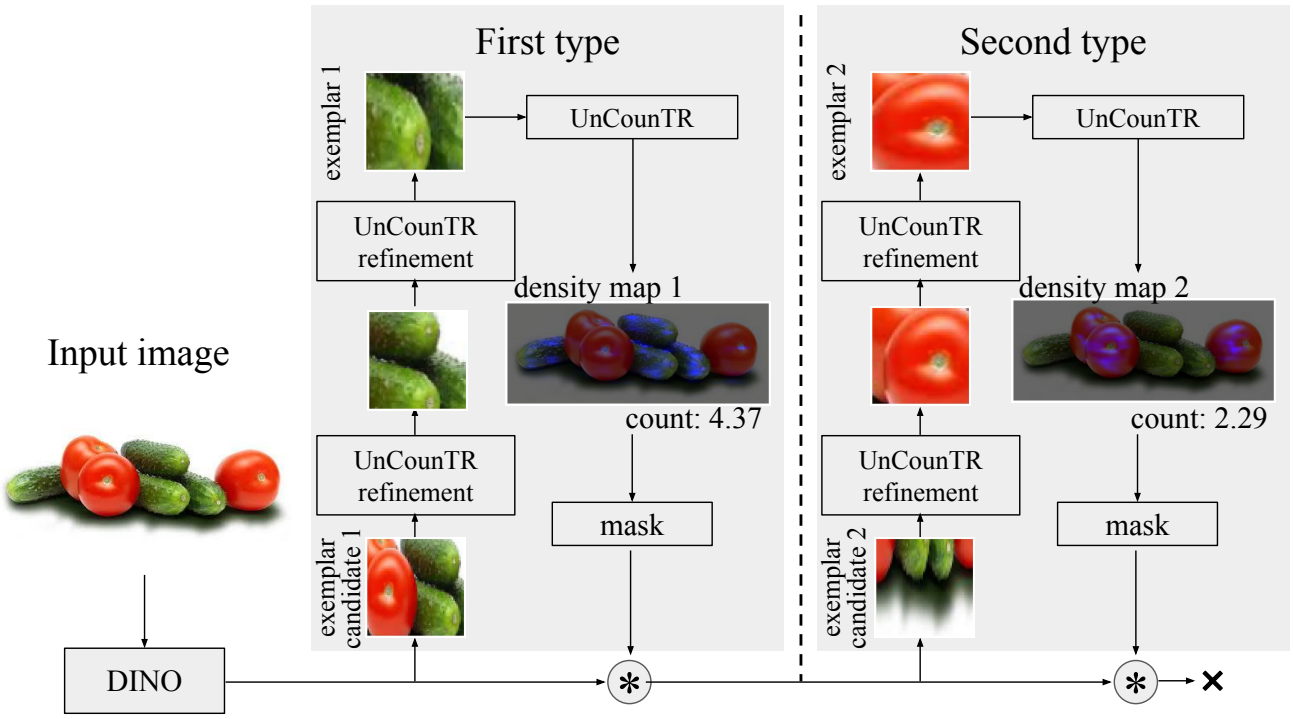


Figure 1. **Self-supervised semantic counting.** To predict the number of objects without any prior, the model uses its DINO backbone to get initial exemplar candidates, which it subsequently refines and uses to predict density maps for the discovered object types.

Problem overview In the self-supervised semantic counting setup, the model automatically identifies the number of object types and appropriate exemplars to predict the density map for each category. Figure 1 illustrates this process.

Processing the input image We first obtain the feature map of the input image $\boldsymbol{I}$ resized to 384×384 pixels using the image encoder: $\mathbf{x} = \Phi(\boldsymbol{I}) \in \mathbb{R}^{h \times w \times d}$. In addition, we extract the final `CLS`-attention map $\mathbf{a} \in \mathbb{R}^{h \times w}$ where we take the average across the attention heads and only consider the self-attention for the $h \cdot w$ patches ignoring the `CLS`-token itself.

Proposing an exemplar candidate After processing the input image, the model determines an exemplar candidate to predict the density map $\hat{\mathbf{y}}^{(t)}$ of the first type $t = 1$. To achieve this, we blur $\mathbf{a}$ by applying a Gaussian kernel of size 3 with sigma 1.5 resulting in $\mathbf{a}^{(1)}$ and identify the position $(y^{(1)}_{\text{max}}, x^{(1)}_{\text{max}}) = \arg\max_{(y,x)} \mathbf{a}^{(1)}_{y,x}$ of the patch with the maximum attention. Subsequently, we compute a binary feature map $\mathbf{b}^{(1)} = \mathbf{a}^{(1)} > 0.5 \cdot \max \mathbf{a}^{(1)}$ and obtain the connected component which includes the position $(y^{(1)}_{\text{max}}, x^{(1)}_{\text{max}})$. The crop of $\boldsymbol{I}$ corresponding to this component is taken as the exemplar candidate $\boldsymbol{E}^{(1)}_{\text{candidate}}$.

Refining the exemplar Using $\boldsymbol{I}$ and $\boldsymbol{E}^{(1)}_{\text{candidate}}$, we predict a density map

$$\hat{\mathbf{y}}^{(1)}_{\text{candidate}} = f_\Theta(\boldsymbol{I}, \{\boldsymbol{E}^{(1)}_{\text{candidate}}\}), \tag{4}$$

where we can skip the image encoder Φ by reusing the feature map $\mathbf{x}$. $\hat{\mathbf{y}}^{(1)}_{\text{candidate}}$ is used to obtain a refined exemplar $\boldsymbol{E}^{(1)}_{\text{refined}}$. To this end, we binarise the density map $\hat{\mathbf{y}}^{(1)}_{\text{candidate}}$ by selecting values greater than 20% of its maximum and receive the second largest connected component, where we assume the largest component represents the background and the second largest corresponds to an object of the first type. We take a crop $\boldsymbol{E}^{(1)}_{\text{crop}}$ of $\boldsymbol{I}$ at the location of this component. Since $\boldsymbol{E}^{(1)}_{\text{crop}}$ might contain adjacent objects of different categories, we construct $\boldsymbol{E}^{(1)}_{\text{refined}}$ by taking another center crop of $\boldsymbol{E}^{(1)}_{\text{crop}}$ reducing each dimension by a factor of 0.3. The refinement step can be repeated multiple times by using $\boldsymbol{E}^{(1)}_{\text{refined}}$ as the new candidate in Equation (4). In practice, we use two refinement iterations.

Predicting the count We employ test-time normalisation and take the density map $\hat{\mathbf{y}}^{(1)}$ obtained using Equation (4) and replacing $\boldsymbol{E}^{(1)}_{\text{candidate}}$ with $\boldsymbol{E}^{(1)}_{\text{refined}}$ as final prediction for the first type.

Counting multiple categories To obtain predictions for more categories, we repeat these steps starting with a new maximum $(y^{(2)}_{\text{max}}, x^{(2)}_{\text{max}})$ after masking out the patches in $\mathbf{a}^{(1)}$ which correspond to the current category resulting in a new attention map $\mathbf{a}^{(2)}$. We identify these patches using two heuristics: First, we mask out patches where $\hat{\mathbf{y}}^{(1)}_{\text{candidate}}$ or $\hat{\mathbf{y}}^{(1)}_{\text{refined}}$, resized to match the dimensions of $\mathbf{a}^{(1)}$, predict a value higher than or equal to 0.5. Second, we set the attention to 0 in an area of 5×5 patches centred around $(y^{(1)}_{\text{max}}, x^{(1)}_{\text{max}})$ to prevent $(y^{(2)}_{\text{max}}, x^{(2)}_{\text{max}}) = (y^{(1)}_{\text{max}}, x^{(1)}_{\text{max}})$. Since every object should only be counted once and to facilitate the knowledge of previous iterations, we subtract the sum of the density maps of previous iterations, from the current prediction:

$$\hat{\mathbf{y}}^{(t)} = \max\left(f_\Theta(\boldsymbol{I}, \{\boldsymbol{E}^{(t)}_{\text{candidate}}\}) - \sum_{t'=1}^{t-1} \max\left(\hat{\mathbf{y}}^{(t')}, \mathbf{0}\right), \mathbf{0}\right) \tag{5}$$

This procedure keeps detecting exemplars and making predictions for new categories t until the maximum remaining attention value is less than 20% of the original maximum at which point we assume that all salient object types have been detected.

Evaluating the importance of the refinement steps Especially the first refinement step is crucial to obtain meaningful exemplars as illustrated in Figure 1. While the initial candidates, which are only based on the DINO backbone, successfully highlight the salient objects, they fail to focus on a single object type. A single refinement step based on UnCounTR solves this issue which indicates the importance of UnCounTR's self-supervised training for this task. The second refinement step has a more subtle impact on the exemplar quality by reducing the number of objects in each exemplar. Based on these self-supervised exemplars, UnCounTR produces counts close to the true number of objects.

Limitation While these results are promising, they are only a qualitative exploration and are intended to highlight a potential avenue for future work. The creation of an evaluation dataset for the semantic counting task as well as the development of a metric to measure exemplar quality are required for a more thorough evaluation of this use case.

B. More details of Self-Collages

B.1. Underlying assumptions for Self-Collages

In the paper, we describe the construction of Self-Collages with ImageNet-1k and SUN397 dataset. Our underlying assumptions are twofold:

1. Images in the SUN397 dataset do not contain objects to serve as the background for our Self-Collages.
2. Images in the ImageNet-1k dataset feature a single salient object to obtain correct pseudo labels.

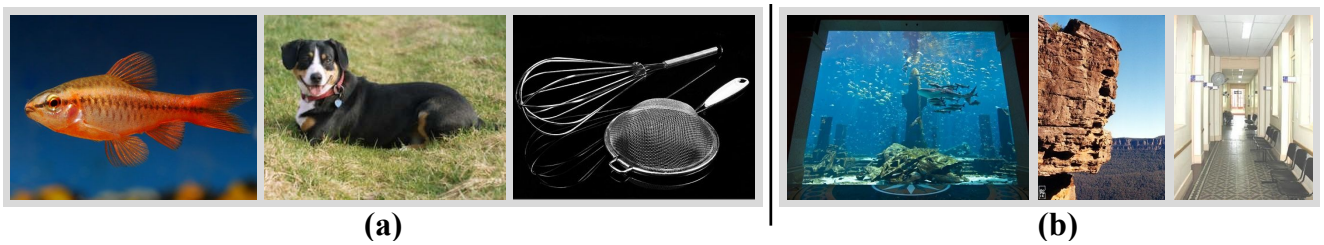


Figure 2. **ImageNet-1k and SUN397 images.** **(a)** Example images from the ImageNet-1k dataset [11]. **(b)** Example images from the SUN397 dataset [13]. While the figures in the SUN397 dataset may contain multiple objects, there is no clearly salient object.

Figure 2 shows three samples from ImageNet-1k and SUN397. We can see that even though images in the latter do contain objects, see *e.g.* the fish in the aquarium, they are usually not salient. Hence, the first assumption is still reasonable for constructing Self-Collages. We acknowledge that this assumption has its limitations and introduces noises for Self-Collages, *e.g.* salient objects exist in some images from SUN397. In Appendix C.3, we consider variants of the default setup to investigate the robustness of our method against violations of this assumption.

The second assumption is crucial to derive a strong supervision signal from unlabelled images. While some ImageNet-1k images contain multiple salient objects, see *e.g.* the right-most image in Subfigure **a**, the final performance of our method shows that the model is able to learn the task even with this noisy supervision. Some techniques such as filtering images based on their segmentation masks, might be able to further improve the supervision signal. We consider these methods as future works.

B.2. Examples of Self-Collages

Figure 3 shows different examples of Self-Collages. While the whole pipeline does not rely on any human supervision, the Self-Collages contain diverse types that align with human concepts from objects like bicyles (Subfigure **a**) to animals such as beetles (Subfigure **c**). Likewise, the unsupervised segmentation method proposed by Shin et al. [12] successfully creates masks for the different instances. The correlated sizes lead to similarly sized objects in each Self-Collage with some samples containing primarily small instances (Subfigures **a** and **d**) and others having mainly bigger objects (Subfigures **b** and **c**). Adding to the diversity of the constructed samples, some Self-Collages show significantly overlapping objects (Subfigure **b**) while others have clearly separated entities (Subfigure **d**). Since the total number of pasted objects is constant, the amount of pasting artefacts in each image does not provide any information about the target count. The density maps indicate the position and number of target objects, overlapping instances result in peaks of higher magnitude (Subfigure **b** and **c**). By computing the parameters of the Gaussian filter used to construct the density maps based on the average bounding box size, the area covered by density mass correlates with the object size in the image.

B.3. Collages in other works

The idea of deriving a supervision signal from unlabelled images by artificially adding objects to background images, the underlying idea behind Self-Collages, is also used in other domains such as object discovery and segmentation.

Arandjelović and Zisserman [1] propose a generative adversarial network, called copy-pasting GAN, to solve this task in an unsupervised manner. They train a generator to predict object segmentations by using these masks to copy objects into background images. The learning signal is obtained by jointly training a discriminator to differentiate between fake, *i.e.* images containing copied objects, and real images. A generator that produces better masks results in more realistic fake images and has therefore higher chances of fooling the discriminator. During inference, the generator can be used to predict instance masks in images. By contrast, our composer module g is only used during training to construct samples. Hence, we

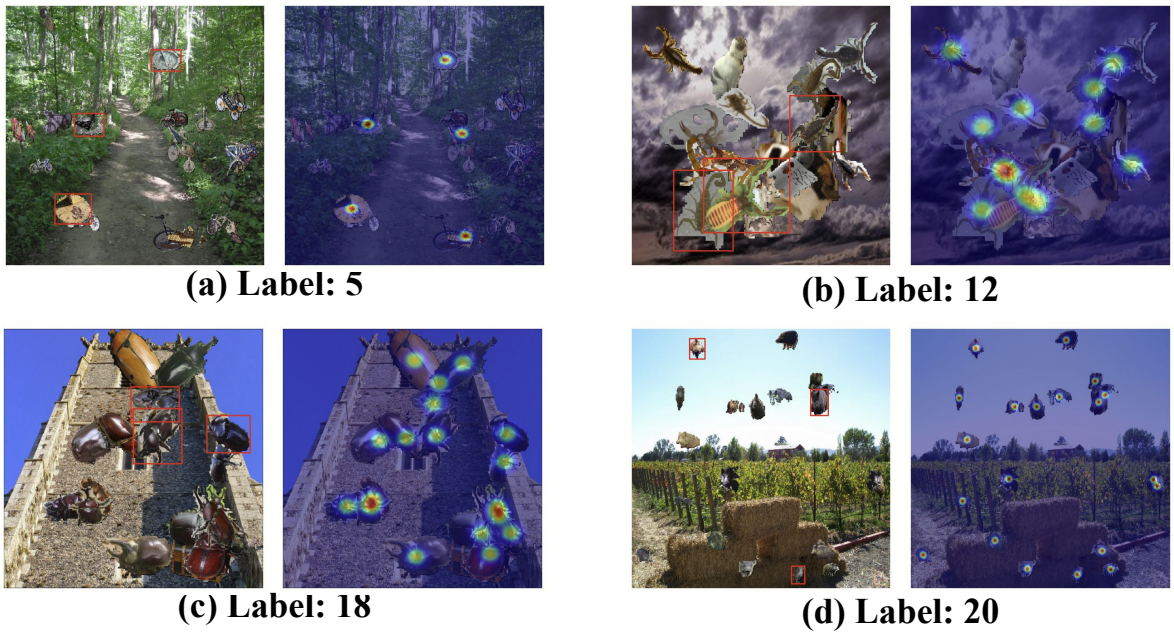


Figure 3. **Example Self-Collages.** The Subfigures show Self-Collages for different counts. The red boxes indicate the exemplars and the heatmaps show the pseudo ground-truth density maps.

do not need an adversarial setup and, without the need to update g during training, the composer module does not have to be differentiable.

To boost the performance of instance segmentation methods, Zhao et al. [16] propose the X-Paste framework. It leverages recent multi-modal models to either generate images for different object classes from scratch or filter web-crawled images. After creating instance masks and filtering these objects, the resulting images are pasted into background images. Generated samples can then be used in isolation or combined with annotated samples to train instance segmentation methods. Compared to this work, our goal is not to train instance segmentation but counting methods where the correct number of objects in the training samples is more important than the quality of the individual segmentations. Partly due to this, our method is conceptually simpler than X-Paste and does not require, for example, multiple filtering steps.

C. Further results

In this section, we further analyse UnCounTR's performance in out-of-domain settings (see Appendix C.1) and explore ways to improve UnCounTR's default setup based on recent advances in self-supervised representation learning (see Appendix C.2). We then evaluate our method under different data distributions in Appendix C.3. We conclude this section with a qualitative comparison between UnCounTR and FasterRCNN (see Appendix C.4) and show additional qualitative results (see Appendix C.5).

C.1. Generalisation to out-of-domain count distributions

In this section, we examine UnCounTR's performance on the different FSC-147 subsets, as presented in the paper, to investigate its generalisation capabilities to new count distributions. We consider the results on the three subsets *low*, *medium*, and *high* with an average of 12, 27, and 117 objects per image respectively. Since UnCounTR is trained with Self-Collages of 11 target objects on average, the count distribution of FSC-147 *low* can be seen as in-distribution while the *medium* and *high* subsets are increasingly more out-of-distribution (OOD).

Unsurprisingly, the model performs worse on subsets whose count distributions deviate more from the training set. Looking at FSC-147 *medium*, the RMSE changes only slightly considering the significant increase in the number of objects compared to *low*. When moving to the *high* subset whose count distribution differs significantly from the training set, the error increases substantially.

These trends can also be seen in Figure 4. Figure 4a shows that the model's predictions are distributed around the ground-

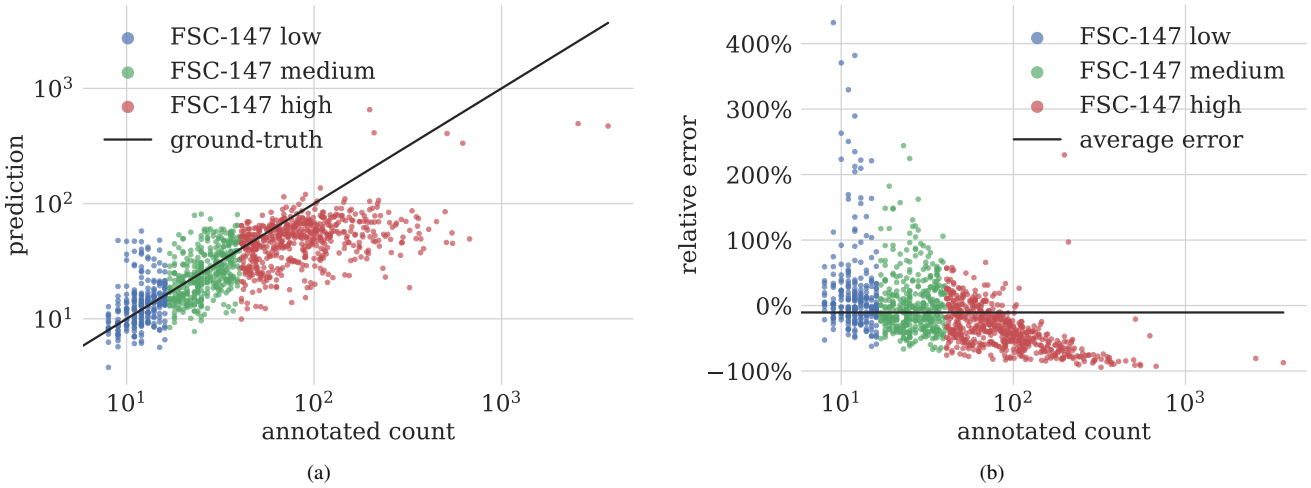


Figure 4. **UnCounTR's predictions.** Figure 4a compares the model's predictions on the different FSC-147 test subsets with the ground-truth. Figure 4b visualises the relative error for these predictions.

Method	FSC-147 *low*			FSC-147 *medium*			FSC-147 *high*		
	MAE↓	RMSE↓	τ ↑	MAE↓	RMSE↓	τ ↑	MAE↓	RMSE↓	τ ↑
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. **UnCounTR's performance on FSC-147.** Evaluation on different FSC-147 test subsets as described in the paper.

			FSC-147			FSC-147 *low*		
Backbone	similarity	refinement	MAE↓	RMSE↓	τ ↑	MAE↓	RMSE↓	τ ↑
DINOv2	✗	✗	34.35	132.02	0.63	4.10	7.34	0.32
DINOv2	✓	✗	31.06	119.35	0.67	3.98	7.58	**0.33**
DINOv2	✓	✓	**28.67**	**118.40**	**0.71**	**3.88**	**7.03**	**0.33**
DINO	✗	✗	35.77	130.34	0.57	5.60	10.13	0.27

Table 2. **Improving UnCounTR's performance.** We explore several variations of UnCounTR's default setup, highlighted in grey, to further close the performance gap to its supervised counterparts.

truth for FSC-147 *low* and *medium*. On the *high* subset, the model tends to underestimate the number of objects in the images. The relative error becomes increasingly negative as these counts increase (see Figure 4b) illustrating the limitations of generalising to OOD count ranges.

While this is expected, we can assume that a model that learned a robust notion of numerosity gives higher count estimates for images containing more objects even in these OOD settings. This can be quantified using the rank correlation coefficient Kendall's τ. Interestingly, the correlation coefficient on FSC-147 *low* and *high* is almost the same, the highest value for τ is achieved on the *medium* subset (see Appendix C.1). This suggests that while UnCounTR's count predictions become less accurate for out-of-distribution samples, the model's concept of numerosity as learned from Self-Collages generalises well to much higher count ranges.

C.2. Improving upon UnCounTR

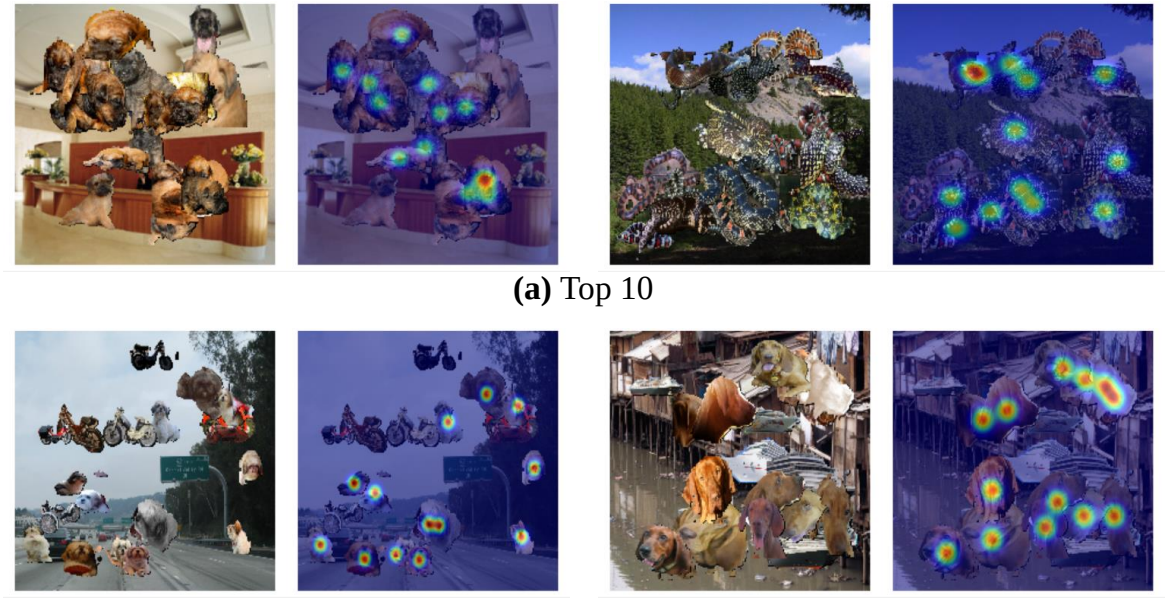
In the paper, we introduce UnCounTRv2 which integrates three modifications compared to the default setup of UnCounTR based on the very recent DINOv2 [10] backbone. Building on this change, we investigate two further modifications: exploiting cluster similarity and refining the model's predictions. Table 2 shows the results for the different variations.

C.2.1 Updating the backbone

First, we update the DINO backbone [3] with the newer DINOv2 [10]. More specifically, we employ the ViT-B model with a patch size of 14. Due to the different patch size, we change the resolution of the exemplars slightly from 64×64 to 70×70. In addition, we update the sliding window's dimension to 392×392 during inference.

The results in Table 2 show that updating the backbone improves the overall performance on the whole FSC-147 dataset, with a small increase in RMSE by 1.3% being the only exception. In particular, this change seems to be beneficial for images with lower counts, the corresponding metrics improve by 19-28%. This demonstrates UnCounTR's ability to take advantage of recent advances in self-supervised representation learning as discussed in the paper.

C.2.2 Cluster similarity



(a) Top 10

(b) Bottom 1000

Figure 5. **Constructing Self-Collages based on cluster similarity.** The similarity between clusters can be used when constructing Self-Collages, the pseudo ground-truth on the right of each image indicates the objects of the target cluster. Subfigure **a** shows Self-Collages where the non-target cluster was randomly chosen among the 10 clusters most similar to the target cluster. Due to the high similarity, objects in the target and non-target clusters are not easily distinguishable. By contrast, the examples in Subfigure **b** use the 1000 most different clusters. The two types in each image are visually very distinct.

Since multiple object types are pasted to generate our training images, we next exploit information about their similarity. We hypothesise that increasing the difficulty of the counting task during training by selecting non-target clusters that are similar to the target clusters could facilitate the learning process. This idea resembles the use of hard negatives in contrastive learning which has been shown to improve the robustness of learned representations [6, 14, 15]. Unlike in supervised setups with manually annotated classes, information about the similarity of object clusters is readily available in our self-supervised setup and can be computed simply as the negative Euclidean distance between the cluster centres. Figure 5 shows the effect of cluster similarity on the constructed Self-Collages.

In Table 3, we compare different ways of exploiting the similarity information. It can be seen that selecting non-target clusters that are similar to the target cluster significantly improves the performance. Importantly, picking non-target clusters that are too similar to the target cluster severely harms the training process. Considering the qualitative samples in Figure 5, very similar object types are almost visually indistinguishable even for humans. Hence, we use the 100 clusters most similar to the target cluster excluding the top 10. Unlike the updated backbone, this change improves the performance especially on images with higher counts as seen in Table 2.

	FSC-147		
similarity range	MAE↓	RMSE↓	τ ↑
✗	34.35	132.02	0.63
top 10	36.67	130.30	0.59
10-100	**31.06**	**119.35**	**0.67**
bottom 1000	34.71	131.12	0.63

Table 3. **Using cluster similarities to construct Self-Collages.** We use cluster similarities in the composer module g to select non-target clusters during Self-Collage construction. Top X describes the setup where we pick the non-target cluster among the X clusters that are most similar to the target cluster. Likewise, the clusters are chosen from the set of the X most dissimilar clusters in the bottom X setting. If we specify a range X-Y, the set of possible non-target clusters is equal to top Y without the elements in top X. All setups use DINOv2 as backbone.

C.2.3 Prediction refinement

Lastly, we modify our evaluation protocol to mitigate the count distribution shift between training and the FSC-147 test set as discussed in Appendix C.1. To this end, we employ a refinement strategy which aims in particular at improving the predictions for images with high object counts. First, we obtain a prediction using the default inference setup described in Appendix A.4. Then, if the model predicts more than 50 objects, we utilise the same setup as employed for small objects where we split the image into 9 tiles and predict the counts for each of them independently before aggregating the final prediction (see Appendix A.4).

Table 2 shows that employing this evaluation protocol further improves the performance resulting in an MAE of 28.67 on FSC-147. Combining all three modifications reduces the MAE of UnCounTRv2 compared to the default UnCounTR setup by 20% on FSC-147 and 31% on FSC-147 *low* which narrows the gap to the supervised counterparts and highlights the potential of unsupervised counting based on Self-Collages.

C.3. Self-Collages based on different datasets

In Table 4, we ablate the datasets used for object ($\mathcal{O}$) and background images ($\mathcal{B}$) to investigate the behaviour of our method under different data distributions. Figure 6 shows training samples for the ablated setups. We first describe the different datasets in Appendix C.3.1, followed by the ablation results in Appendix C.3.2.

C.3.1 Dataset ablations

ImageNet-1k-only To simplify the construction process, we experiment with using the ImageNet-1k dataset for both, $\mathcal{O}$ and $\mathcal{B}$. To make sure that the counting task is not affected by the background, we exclude all images in the two clusters used for the object images before randomly selecting the background image.

ImageNet-1k Top-2 As a simpler version of the default setup, we filter the ImageNet-1k dataset to only use the images in the two biggest clusters, which we call ImageNet-1k Top-2. Looking at Subfigure **b** in Figure 6, these clusters seem to correspond to two different types of birds. This reduces the number of images from 1,281,167 to 1,266. Because this subset only features two clusters, every Self-Collage contains the same object types.

Synthetic dataset We construct a synthetic dataset based on simple shapes. To this end, we combine three types of shapes, `squares`, `circles`, and `triangles`, and four different colours, `red`, `green`, `blue`, and `yellow`, to obtain a total of 12 possible object types. After randomly picking two different types, we select a random background colour amongst the colours not used for the objects.

Noise We use the StyleGAN-Oriented dataset proposed by Baradad Jurjo et al. [2], which is based on a randomly initialised StyleGANv2 [8], as noise dataset to draw background images from. In total, it contains 1.3M synthetic images. We refer to the original work [2] for more details.

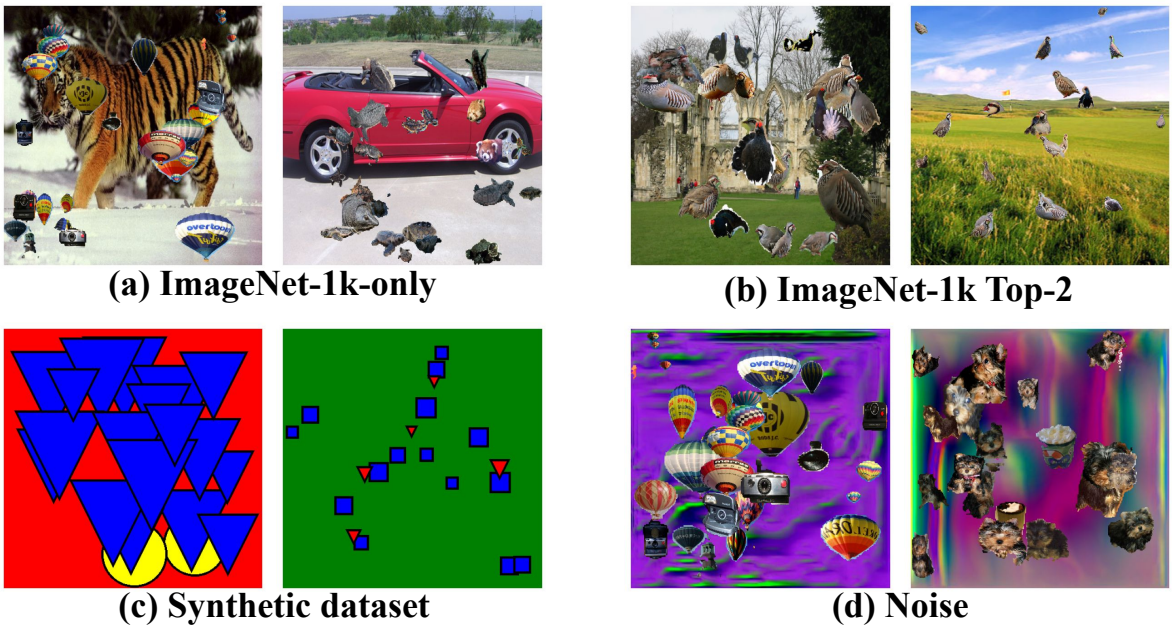


Figure 6. **Dataset ablations.** We modify the training dataset by using different datasets for $\mathcal{O}$ and $\mathcal{B}$ to investigate the effect on the final counting performance.

		FSC-147			FSC-147 *low*		
$\mathcal{O}$	$\mathcal{B}$	MAE↓	RMSE↓	τ ↑	MAE↓	RMSE↓	τ ↑
ImageNet-1k		35.68	130.10	0.56	6.78	12.43	0.25
ImageNet-1k Top-2	SUN397	39.47	132.37	0.41	12.08	17.84	0.16
Synthetic dataset		45.16	144.60	0.26	7.03	11.04	0.23
ImageNet-1k	Noise	**34.43**	**128.73**	**0.60**	5.94	10.72	0.26
ImageNet-1k	SUN397	35.77	130.34	0.57	**5.60**	**10.13**	**0.27**

Table 4. **Using different datasets to construct Self-Collages.** We vary the datasets used by the composer module g to obtain object ($\mathcal{O}$) and background images ($\mathcal{B}$) when constructing Self-Collages. The default setting is highlighted in grey.

C.3.2 Dataset ablation results

Image diversity improves generalisability It can be seen, that gradually decreasing the image diversity, by using the same dataset for $\mathcal{O}$ and $\mathcal{B}$, using only a very small ImageNet-1k subset for $\mathcal{O}$, or using a fully synthetic dataset, harms the performance on the FSC-147 dataset.

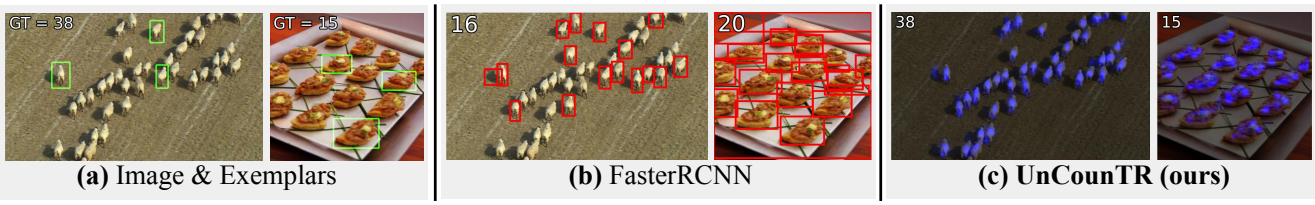
Simple objects are sufficient for low counts only While using the fully synthetic dataset yields the worst results on FSC-147, the performance on the FSC-147 *low* subset is comparable to the setup using only ImageNet-1k. This indicates the importance of more realistic objects for the generalisability to higher counts. At the same time, synthetic objects seem to be sufficient for learning to predict the number of objects in images with few instances.

Synthetic but diverse backgrounds perform similarly to real images When replacing SUN397 with a noise dataset, the performance on the whole FSC-147 dataset improves while being slightly worse on the FSC-147 *low* subset.

Self-Collages are robust against violations of the background assumption Comparing the setup which uses the ImageNet-1k dataset for $\mathcal{O}$ and $\mathcal{B}$ to the default setting, we can see that even by explicitly violating the assumption that there are no

salient objects in $\mathcal{B}$, the performance is not significantly affected. However, by using a noise dataset as $\mathcal{B}$ where the assumption holds, we can further improve the performance on FSC-147. This indicates that while our method is robust to violations of the aforementioned assumption, using artificial datasets where the assumption is true, can be beneficial.

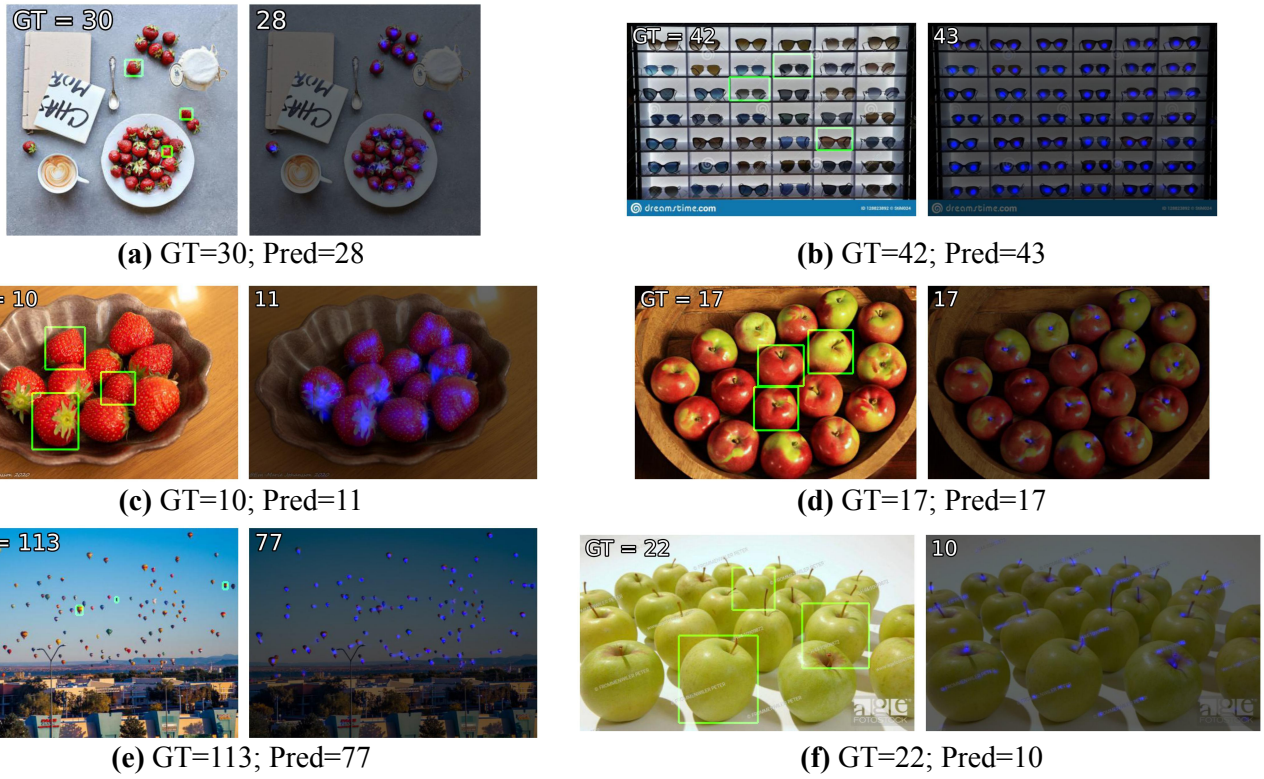
C.4. Comparing FasterRCNN and UnCounTR



(a) Image & Exemplars **(b)** FasterRCNN **(c) UnCounTR (ours)**

Figure 7. **When UnCounTR works better than FasterRCNN.** Given two images with exemplars **(a)** we compare the output predictions of FasterRCNN **(b)** and of our model **(c)**. We find that FasterRCNN either misses most objects in high-density settings or detects non-target instances which is because the model cannot utilise any prior knowledge in the form of exemplars.

C.5. Qualitative results



(a) GT=30; Pred=28 **(b)** GT=42; Pred=43

(c) GT=10; Pred=11 **(d)** GT=17; Pred=17

(e) GT=113; Pred=77 **(f)** GT=22; Pred=10

Figure 8. **Qualitative UnCounTR results.** We show UnCounTR's predictions on six different images from the FSC-147 test set. The green boxes represent the exemplars and the number in the top-left of each image indicate the ground-truth and predicted count for each sample. Our prediction is the sum of the heatmap rounded to the nearest integer.

Figure 8 visualises predictions of our model UnCounTR on the FSC-147 dataset. The model predicts good count estimates even for images with more than twice as many objects as the maximum number of objects seen during training (Subfigure **b**). In general, the model successfully identifies the object type of interest and focuses on the corresponding instances even if they only make up a small part of the entire image (Subfigure **a**). However, UnCounTR still misses some instances in these settings.

For very high counts (Subfigure **e**) and images with artefacts such as watermarks (Subfigure **f**), the model sometimes fails to predict a count close to the ground-truth.

References

[1] Relja Arandjelović and Andrew Zisserman. Object discovery with a copy-pasting gan. *arXiv preprint arXiv:1905.11369*, 2019. 5

[2] Manel Baradad Jurjo, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise. *Advances in Neural Information Processing Systems*, 34:2556–2569, 2021. 9

[3] 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, 8

[4] 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

[5] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 364–380, 2018. 2

[6] Songwei Ge, Shlok Mishra, Chun-Liang Li, Haohan Wang, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. *Advances in Neural Information Processing Systems*, 34:27356–27368, 2021. 8

[7] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2918–2928, 2021. 2

[8] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 9

[9] Chang Liu, Yujie Zhong, Andrew Zisserman, and Weidi Xie. Countr: Transformer-based generalised visual counting. *BMVC*, 2022. 1, 3

[10] 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

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252, 2015. 5

[12] Gyungin Shin, Samuel Albanie, and Weidi Xie. Unsupervised salient object detection with spectral cluster voting. In *CVPR*, pages 3971–3980, 2022. 5

[13] 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. 5

[14] Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. Hard negative examples are hard, but useful. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 126–142. Springer, 2020. 8

[15] Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Re. Correct-n-contrast: a contrastive approach for improving robustness to spurious correlations. In *International Conference on Machine Learning*, pages 26484–26516. PMLR, 2022. 8

[16] 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. 2, 6