UCLouvain

epl

École polytechnique de Louvain

Agnostic object counting with deep learning models

Author: **Jordan Hanotiaux**
Supervisor: **Christophe De Vleeschouwer**
Readers: **Christophe De Vleeschouwer, Maxime Istasse, Benoît Macq**
Academic year 2025–2026
Master [120] in Computer Science and Engineering

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Pr. De Vleeschouwer, for his time, availability, and valuable guidance throughout this year, which has been beneficial both academically and personally for the future.

I would also like to thank Mr. Istasse for the advice provided during previous projects, which helped me gain confidence and develop myself further.

I am also grateful to Pr. Macq for agreeing to take the time to serve as a reviewer of this thesis and for evaluating this work.

A special thank you goes to my father for believing in me despite my difficult beginnings and for his constant support.

Finally, I would like to thank my partner, my friends, and my family for their support throughout this long and meaningful stage of my life, which is now coming to an end.

Code

Code is available here Github Repository.

AI Disclaimer

Artificial intelligence has been used for translation and text rephrasing purposes, using ChatGPT. No ideas, or experimental results were generated by AI tools.

Contents

1 Introduction **6**
- 1.1 Context . . . 6
- 1.2 Research Question . . . 6
- 1.3 Structure . . . 6

2 Foundations and Prior Work **7**
- 2.1 Related work . . . 7
  - 2.1.1 Class-agnostic counting methods . . . 7
  - 2.1.2 Few-Shot Object Counting . . . 8
  - 2.1.3 Foundational Vision Models (ViT, DINOv2) . . . 9
- 2.2 Background & Preliminaries . . . 11
  - 2.2.1 CountingDINO . . . 11
  - 2.2.2 Graph-Cut . . . 12

3 Methodology **15**
- 3.1 Overview . . . 15
- 3.2 DINO feature extraction . . . 15
- 3.3 Similarity Map . . . 16
- 3.4 Convolutional Map . . . 18
- 3.5 Density map . . . 19
- 3.6 Post processing . . . 20
- 3.7 LM-Fallback . . . 22

4 Experiments and Results **23**
- 4.1 Evaluation Protocol . . . 23
  - 4.1.1 Datasets . . . 23
  - 4.1.2 Metrics . . . 25
  - 4.1.3 comparaison with SOTA . . . 25
- 4.2 Quantitative Results . . . 25
  - 4.2.1 Results on the FSC-147 dataset . . . 25
  - 4.2.2 Results on the FSCindu dataset . . . 27
- 4.3 Qualitative Results . . . 29
- 4.4 Failure Cases . . . 31
- 4.5 Ablation studies . . . 34
  - 4.5.1 Impact of Resolution . . . 35
  - 4.5.2 Impact of Graph-Cut Optimization . . . 37
  - 4.5.3 Impact of LM Fallback . . . 38
  - 4.5.4 Impact of Number of Templates . . . 39
  - 4.5.5 Impact of small templates exemplars . . . 40
  - 4.5.6 Inference Time . . . 41

5 Discussion **43**
- 5.1 Methodological Improvements over CountingDINO . . . 43
  - 5.1.1 Resolution Handling . . . 43
  - 5.1.2 Convolution Map Construction . . . 44
  - 5.1.3 Scale-Aware Template Normalization . . . 44
  - 5.1.4 Interpretability and Error Analysis . . . 45

5.2 Contributions & Takeaways 45
5.3 Limitations & Future Work 47

6 Conclusion **48**

Chapter 1

Introduction

1.1 Context

Object counting in images is an important computer vision task with numerous applications, including quality control in production lines, traffic monitoring, agriculture, and medical analysis, among others. While it may appear straightforward for humans to count a small number of objects in an image, this task becomes significantly more complex as the number of instances increases and when occlusions, scale and shape variations are present.

Over the years, with the development of deep learning models, performance on this task has steadily improved. Recent approaches are now capable of robustly estimating previously challenging scenarios, providing valuable support by automating this task and reducing human error. However, despite these advances, counting across diverse categories using a single robust and generalizable model remains a challenging problem.

1.2 Research Question

Although some methods are able to generalize and count multiple object categories with strong performance, several limitations remain, particularly due to object variability in images, the need for large annotated datasets with thousands of annotations, and the restriction of many models to specific categories.

This work is guided by the following research question:

> To what extent is it possible to design a method capable of counting any type of object based solely on the visual representation of an image, while remaining robust to scale variations and occlusions, and without relying on any annotated dataset for training?

1.3 Structure

This thesis is structured as follows.

First, the problem of class-agnostic object counting is introduced, along with an overview of existing methods and the ways in which these models differ, both in terms of architecture and training strategies.

Next, our proposed method is presented, with a detailed component-by-component explanation.

This is followed by an analysis of the obtained results, a comparison with the state of the art, and an ablation study highlighting the impact of the different modules introduced in the method.

Finally, a discussion is provided, summarizing the differences with the closest related method, the main contributions and key takeaways of this work, as well as its limitations and possible future directions.

Chapter 2

Foundations and Prior Work

2.1 Related work

2.1.1 Class-agnostic counting methods

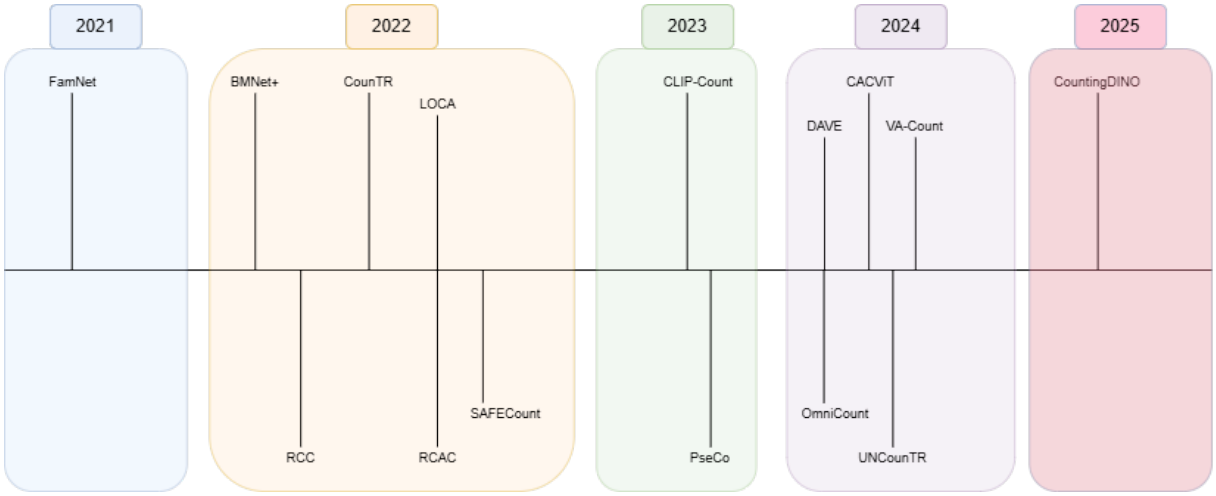


Figure 2.1: **Class-agnostic methods over time.** Publication date of notable CAC methods over time.

Class-agnostic counting methods are designed to count objects irrespective of their category and offer several advantages over traditional category-specific approaches. Unlike the latter, they do not rely on dedicated annotated datasets for each object category, which are often costly and time-consuming to collect and annotate. As a result, class-agnostic methods are inherently more flexible and can be readily applied to unseen object categories without requiring the training of multiple specialized models. Owing to these advantages, they have attracted increasing attention in recent years, leading to the development of a wide range of approaches, as illustrated in Fig. 2.1.

One of the earliest foundation models is the Generic Matching Network (GMN) [15]. This model formulates the counting task as a matching problem between an exemplar, provided as a bounding box, and the full image. It aims to learn a density map through a matching module, which is then used to estimate the number of objects either by integrating the map or by counting local maxima when objects are well separated. GMN is trained on annotated video data, which provides multiple views of the same object under varying conditions, facilitating the learning of the matching function. However, when applied to new categories, the model requires fine-tuning through an adaptation module to adjust the feature representations to the new domain. This requirement deviates from the class-agnostic paradigm. Without such adaptation, it has been shown that GMN performs poorly [22].

Nevertheless, GMN laid the foundation for subsequent work. One of the most well-known methods in class-agnostic counting, FamNet [22], builds upon the ideas introduced by GMN. It reformulates the problem as a few-shot regression task and proposes an architecture capable of generalizing counting to

arbitrary object categories based on a small number of input examples, leveraging the self-similarity property of images. FamNet also addresses the issue of dataset availability for training. While previous methods relied on multiple datasets, each dedicated to a single class, FamNet introduced a new dataset, FSC-147, which has become a benchmark dataset for class-agnostic object counting.

Subsequent models have extended the work of FamNet. The problem has been explored and improved along multiple directions. Zero-shot counting methods have also been proposed like VA-Count [27] or Clip-Count [10], although the best-performing models remain predominantly few-shot approaches [4]. Some methods address also intra-class variability, like RCAC [8], such as differences in shape, color, or size, while others focus also on improving model architectures, as CACViT [26].

2.1.2 Few-Shot Object Counting

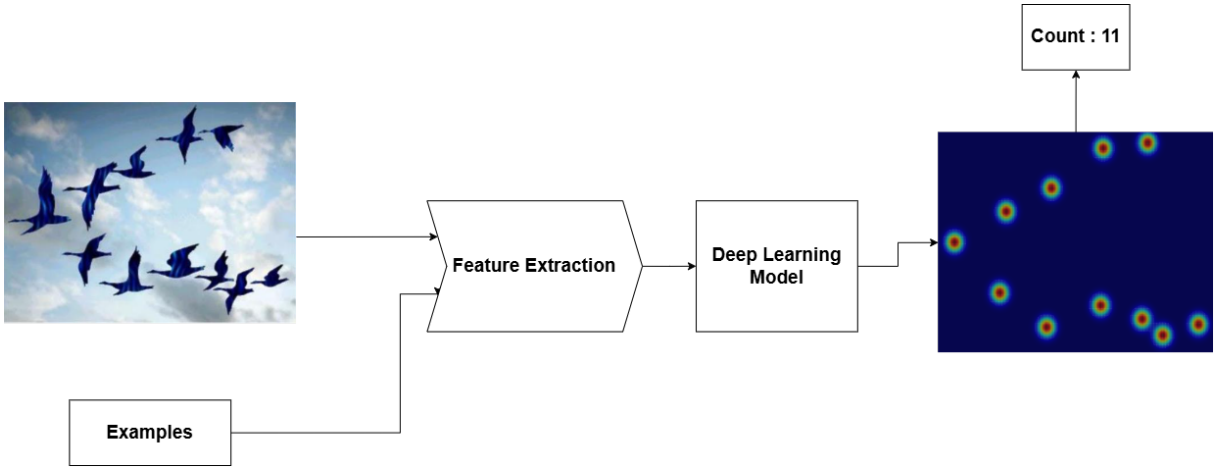


Figure 2.2: **Baseline of a few-shot agnostic object counting model.** Given a query image and a few support exemplars, the model extracts visual features using an encoder and generates a density map through a trainable deep learning module. Based on this density map, the final object count is predicted.

Many methods addressing class-agnostic counting rely on a few-shot counting paradigm. These approaches take as input an image containing objects, from which one or several instances of the target object are selected using points or bounding boxes. Typically, one, two, three, or more examples of the object can be provided. Increasing the number of examples generally improves performance but also increases the cost of dataset construction or the time required from the user to provide annotations. As illustrated in Fig. 2.2, which represents a baseline few-shot counting method, both the input image and the provided examples are encoded into an embedding space. These features are then used as input to a deep learning module that predicts a density map. This density map is subsequently used either to estimate the object count through an additional module or by directly integrating the values over the map.

Most existing methods build upon this baseline with various improvements. They can be distinguished in several ways. First, they differ in how the image and examples are encoded into the embedding space. Some models rely on convolutional neural networks [8, 20, 22, 23], while more recent approaches employ vision transformers [10, 14, 18, 26, 27]. Second, methods vary in how exemplar templates are represented: some directly encode the templates through a vision encoder [14, 23, 26], whereas others extract features from the image feature map itself [8, 18, 22]. Furthermore, models differ in how the density map is predicted and how it is subsequently used. They also vary in their training strategies: some approaches jointly train both the vision encoder and the density prediction module [6, 8, 14, 20, 22, 23, 26], while others train only one of these components [12, 16]. Certain methods, such as CountingDINO [18], rely on pre-trained vision transformers trained using self-supervised learning and adopt a training-free prediction strategy.

A key challenge for these methods lies in intra-class variability among the provided exemplars. Indeed, instances of the same object may significantly differ in size, shape, or color. As highlighted by RCAC,

this variability can substantially degrade the performance of models such as FamNet making it a central issue in the design of robust counting methods.

2.1.3 Foundational Vision Models (ViT, DINOv2)

Although traditional models rely on convolutional neural networks (CNNs) to encode images and exemplars, recent years have seen the emergence of a different type of encoder: vision transformers. Several recent models, such as CACViT [26], CounTR [14], UNCounTR [12], and CountingDINO [18], exploit this architecture and achieve state-of-the-art performance.

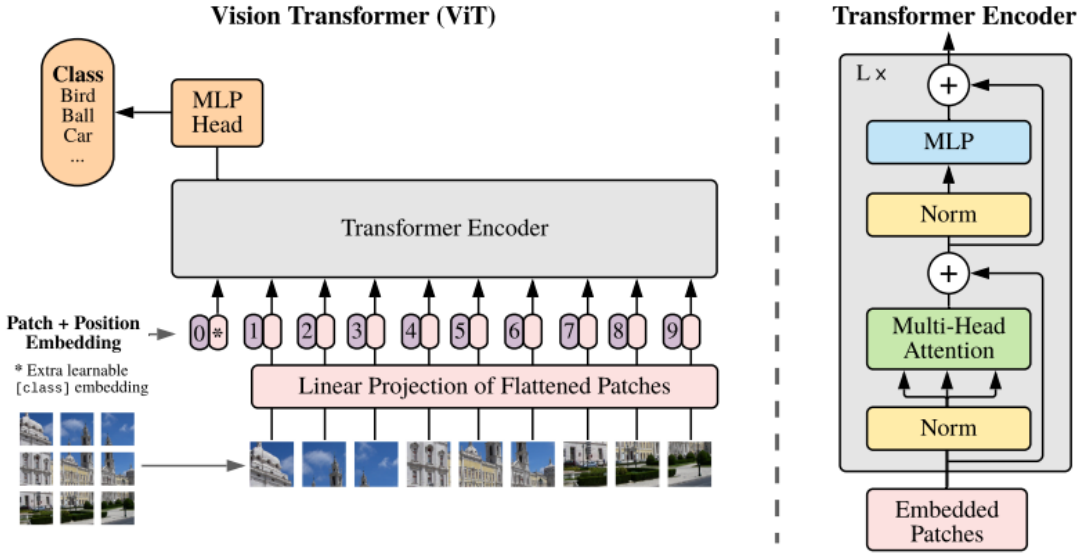


Figure 2.3: **Baseline architecture of a Vision Transformer (ViT). Source: [7]**
The input image is divided into fixed-size patches, and each patch is linearly projected into an embedding vector. An additional learnable embedding token is prepended to represent the global image representation for classification. Positional embeddings are also added to preserve spatial information, and the resulting sequence is fed into the Transformer encoder.

Originally introduced in 2021 by Dosovitskiy et al., [7] vision transformers leverage the self-attention mechanism widely used in natural language processing [25]. Images are treated as sequences of patches (typically of size 14×14 pixels), which are then processed by a standard transformer architecture (see Fig. 2.3). This design enables the model to learn correlations between patches while also extracting local features at the patch level, leading to general-purpose visual representations.

Among vision transformers, the DINO family [17, 24] of models adopts a self-supervised learning approach, eliminating the need for large annotated datasets by learning directly from raw images. Training is based on a student-teacher framework. The student model is optimized during training, while the teacher model serves as a stable reference. Different augmented views of the same image (e.g., cropping, inpainting, rotation) are generated and passed through both models: the teacher processes the original image, while the student processes the transformed views. The objective is for the student to produce feature representations that are as close as possible to those of the teacher. While the student is updated through gradient-based optimization, the teacher is updated using an exponential moving average (EMA) of the student parameters, allowing it to evolve more smoothly and remain stable throughout training. This joint evolution results in improved representations.

Self-supervised training follows an intra-image learning strategy. First, the DINO loss (see Fig. 2.4) captures global image information through the classification token. Two different crops of the same image are fed into the two models, and the objective is to align the resulting classification tokens. Second, the iBOT loss (see Fig. 2.5) focuses on local details by masking certain regions of the student input and requiring the model to predict the representations of the masked patches. Finally, the KoLeo loss acts

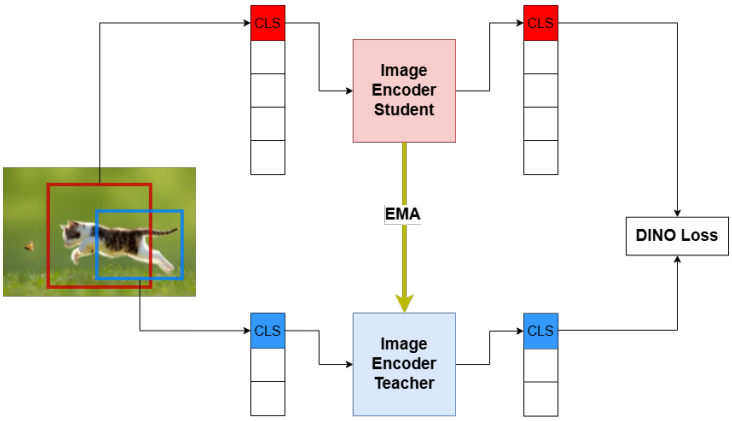


Figure 2.4: **Overview of the DINO self-supervised learning loss.** Two augmented crops of the same image are first divided into patches, linearly projected into embedding vectors, and augmented with a classification token [CLS] before being processed by the student and teacher encoders. The resulting representations are then compared using the DINO loss based on the classification token, which captures the global representation of the image. The teacher network parameters are then updated using an Exponential Moving Average (EMA) of the student network parameters.

as a regularization term by encouraging a more uniform distribution of embeddings in the latent space, thereby preventing representation collapse and improving diversity.

The combination of these three loss functions enables the learning of both global and local representations while ensuring a well-structured feature space.

More recent work, particularly on DINOv3 [24], has introduced new self-supervised learning strategies and scaling improvements to further enhance model performance.

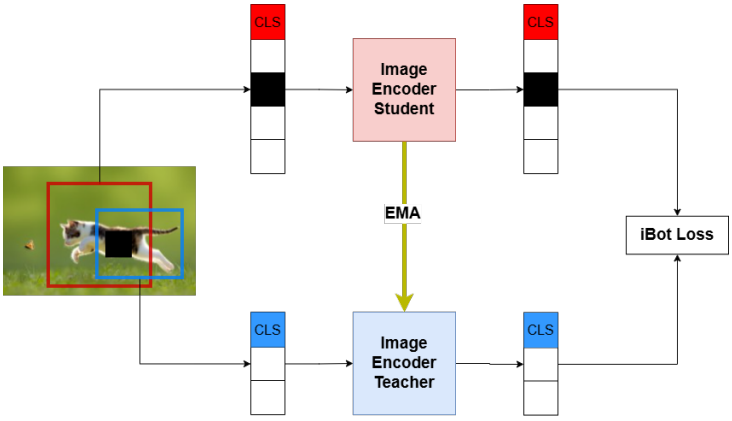


Figure 2.5: **Overview of the iBOT self-supervised learning loss.** Two crops of the same image are divided into patches and linearly projected into embedding vectors. In the student branch, the embedding vector corresponding to a selected shared patch is masked before being processed by the encoder. After encoding, the model predicts the representation of the masked patch, and the loss is computed by comparing the student prediction with the corresponding teacher patch representation. The objective is to reconstruct patch embeddings that are as similar as possible to those produced by the teacher network and so learn local details at patch level. The teacher network parameters are then updated using an Exponential Moving Average (EMA) of the student network parameters.

We can observe in Fig. 2.6 the robustness and effectiveness of the features learned by the DINOv2 model, where the similarity between objects of the same instance is high, while a clear discrimination is maintained between objects and regions that do not belong to the corresponding category.

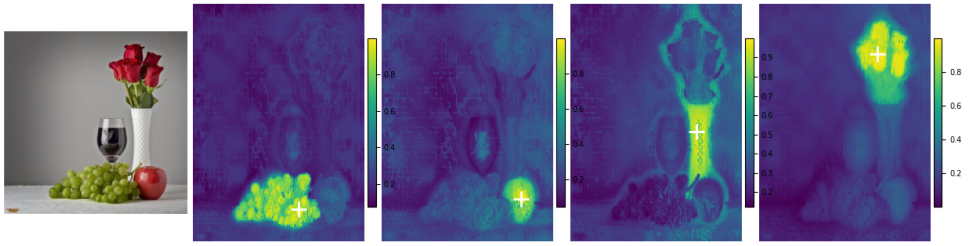


Figure 2.6: **High-resolution dense features.** Cosine similarity map between several patches (marked with a white cross) and the entire image (with a resolution of $2520 * 2520$). This results in a map where high-intensity regions correspond to areas most similar to the selected patch.

2.2 Background & Preliminaries

2.2.1 CountingDINO

CountingDINO [18] was the first method, introduced in 2025, to propose a fully training-free pipeline for class-agnostic object counting based on a self-supervised, vision-only backbone, namely DINOv2 [17]. This approach is particularly interesting because, unlike more traditional methods, it does not require training on large annotated datasets containing thousands of dot annotations and bounding boxes. Such requirements considerably limit the scalability of existing models and may also hinder their ability to generalize across different scenarios.

Instead, CountingDINO relies on a backbone trained through self-supervised learning, thereby avoiding dependence on extremely large annotated datasets such as ImageNet [5]. Furthermore, by removing the training stage associated with density map prediction, the method provides a more flexible and generalizable framework for object counting across previously unseen categories.

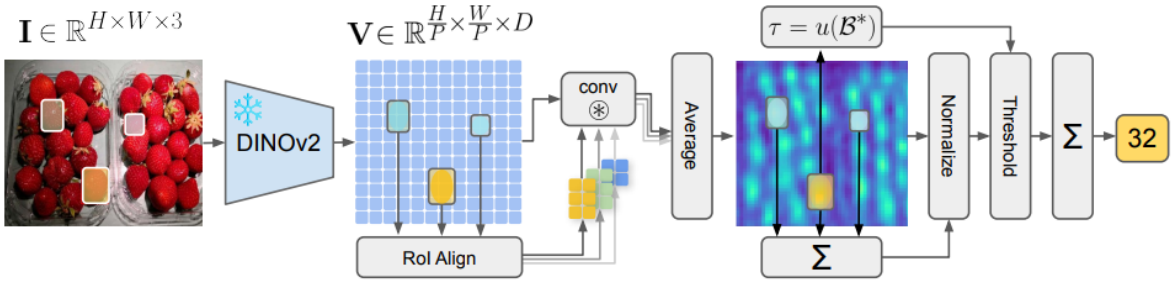


Figure 2.7: **Overview of the CountingDINO pipeline. Source : [18]**

As illustrated in Fig. 2.7, the method relies on extracting a feature map from the image using DINOv2, followed by a ROI alignment of the features in order to perform a convolution between each template and the feature map. A post-processing stage is then applied, where the convolution maps obtained from the different templates are aggregated by averaging the three resulting maps.

A normalization coefficient is subsequently computed according to the following principle. Since each provided bounding box is known to contain an instance of the target object, the values of the convolution map within these regions are summed, and the average of the three obtained sums is used as the normalization coefficient.

Finally, before estimating the object count by summing all pixel values of the final density map, a thresholding step is applied. This step relies on the assumption that the contribution of the weights within a template is spatially uniform. Considering the largest provided template, the threshold is therefore defined as the inverse of the template area expressed in number of patches.

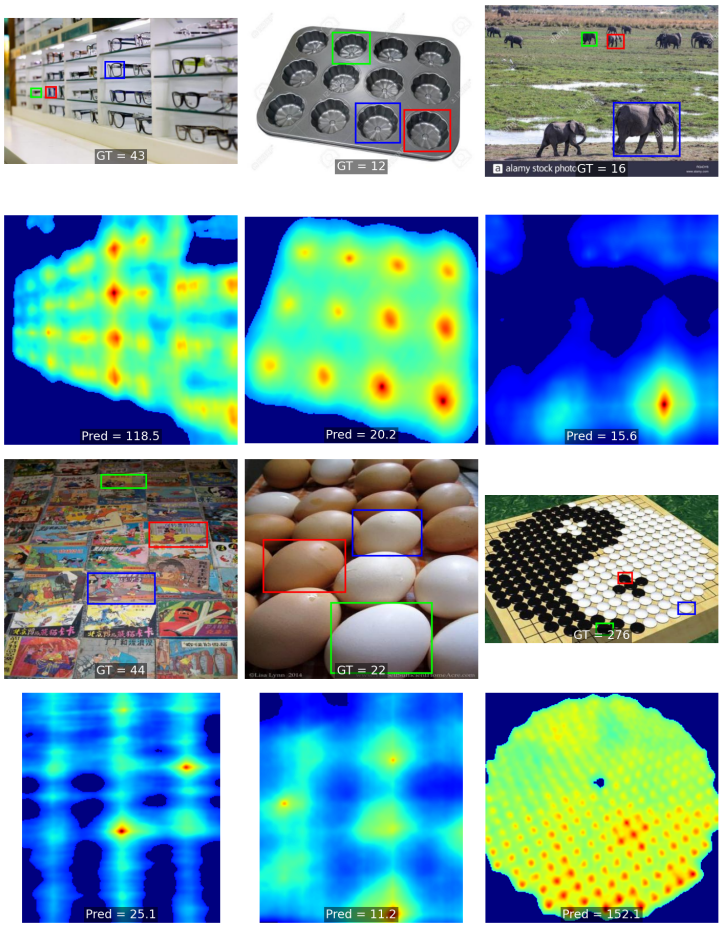


Figure 2.8: **Examples of CountingDINO results on the FSC-147 dataset.** The input image and provided templates are shown at the top, while the resulting density map along with the final count prediction is displayed at the bottom.

A closer examination of the results obtained by CountingDINO in Fig. 2.8 shows that, although the method is able to provide an approximation of the number of objects, it remains relatively distant from the ground truth in several similar cases. It is also observed that the final density map is difficult to interpret visually for a human observer; for instance, in the case of eggs, it is not straightforward to infer the position of each individual egg solely from the map. In some examples, a correct peak appears at each object location, but a significant amount of surrounding noise is also present.

CountingDINO has opened a promising direction by introducing a method relying exclusively on features extracted from a self-supervised vision-only backbone. However, these features may potentially be further refined in order to obtain more visually consistent and relevant results.

2.2.2 Graph-Cut

In computer vision, a recurring problem consists in assigning a label to each pixel of an image. For instance, in segmentation, a pixel is labeled depending on whether it belongs to an object or not. In object counting, a similar formulation can be considered: given similarity maps and provided exemplars, each pixel can be assigned either to the background or to the most visually similar exemplar. To address this problem, Boykov proposed *Fast Approximate Energy Minimization via Graph Cuts* in 2001 [3], a method that assigns labels by minimizing an energy function while enforcing spatial smoothness in label transitions between neighboring pixels, thereby reducing noise in the labeling process.

Indeed, an object may be visually similar to the provided exemplars; however, a labeling based solely on similarity between the given exemplars can introduce significant noise, leading to large variations of labels across pixels. The approach of Boykov mitigates this issue while preserving sharp transitions in regions with strong discontinuities, such as object boundaries.

The labeling problem is formulated as an energy minimization problem of the form:

$$E(L) = \sum_{p \in P} D_p(L_p) + \sum_{(p,q) \in N} V_{pq}(L_p, L_q),$$

where L denotes the labeling of all pixels, L_p the label assigned to pixel p, P the set of image pixels, and N a neighborhood system defined over adjacent pixels.

The term $D_p(L_p)$ is the data term, representing the cost of assigning a label to pixel p. The term $V_{pq}(L_p, L_q)$ is the smoothness term, which encourages spatial consistency by penalizing label changes between neighboring pixels. Direct minimization of this energy is generally NP-hard. Boykov therefore proposed an approximate optimization algorithm based on graph cuts, in particular the α-expansion algorithm. At each iteration, a label α is selected, and each pixel is allowed either to keep its current label or to switch to α, thereby reducing the multi-label problem to a sequence of binary optimization steps.

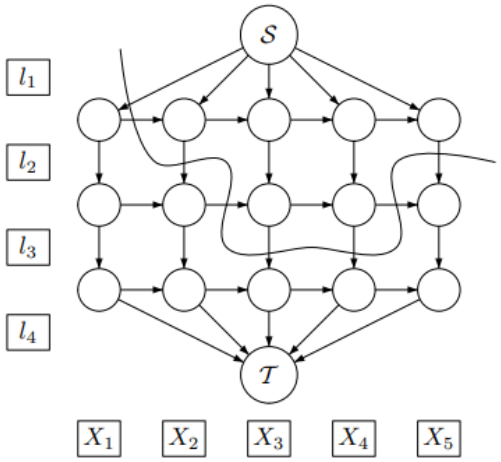


Figure 2.9: **Graph cut formulation for multi-label energy minimization. Source: [19]** Each column represents a pixel (variable) connected to a set of possible labels. A valid solution is obtained by cutting the graph such that each column is separated at a specific label level; the position of the cut determines the assigned label for that pixel. In this example, the resulting labeling is given by: $X1 = l1$, $X2 = l2$, $X3 = l3$, $X4 = l3$, $X5 = l2$.

The problem is then formulated as finding a minimum cut in a graph. Each pixel corresponds to a node, and edges encode either data costs or smoothness costs. The former connect nodes to terminal nodes, while the latter connect neighboring nodes as represented in Fig. 2.9. The optimal cut partitions the graph into regions that are coherent in terms of labels, where each pixel is assigned the label of the segment to which it belongs. An example of label refinement via α-expansion optimization is shown in Fig. 2.10. This formulation enables robust and spatially coherent label assignment, making it particularly well suited for refining noisy similarity maps in vision tasks such as segmentation or object counting.

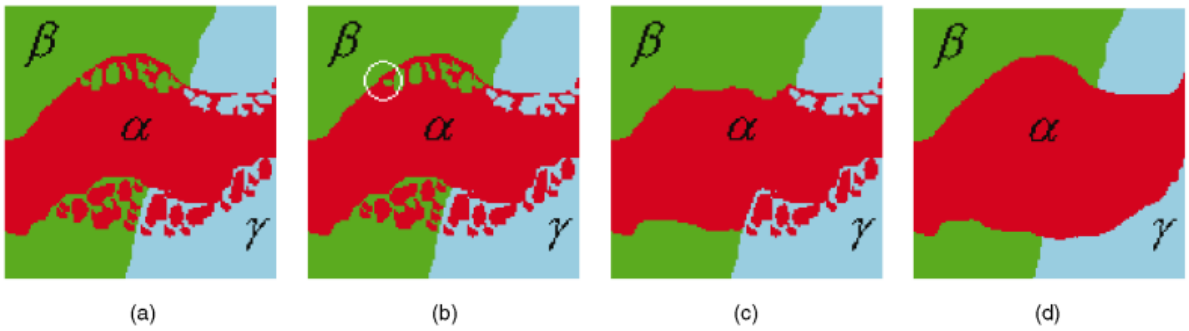


Figure 2.10: **Label refinement using graph-cut optimization. Source: [3]**
Illustration of graph-cut optimization for label refinement. (a) Initial labeling, typically noisy or coarse. (b) Effect of a local update on a single pixel assignment. (c) Example of a strong move allowing multiple pixels to change labels jointly, leading to larger coherent regions. (d) Final labeling after optimization, showing improved spatial consistency and smoother region boundaries.

Chapter 3

Methodology

3.1 Overview

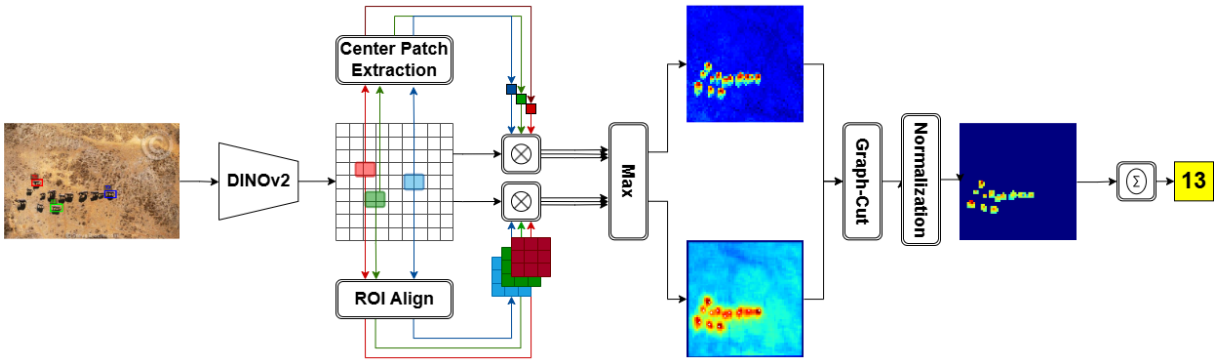


Figure 3.1: **Overview of the proposed pipeline.** Given an image and N templates, DINOv2 features are extracted. Central patches are used to compute similarity maps, while ROI-aligned template features serve as convolutional kernels to produce convolutional maps. The resulting maps are aggregated and post-processed into a density map for object counting.

Our objective is to accurately estimate the number of occurrences of a given object in an image based on a limited set of examples of that object. These examples are provided as template regions directly extracted from the input image. The model must then generalize from these templates in order to count all similar occurrences present in the image, without requiring a dedicated annotated dataset for training. This task is particularly challenging due to potential variations in object appearance, such as changes in scale, shape, and color. To address these challenges, our approach aims to build a robust and generalizable method based on the representation of the image.

To achieve this, we rely on representations extracted by DINOv2 [17], a pre-trained Vision Transformer model that generates image representations in the form of features. Based on these representations, we propose a multi-step processing pipeline designed to effectively leverage these features. The objective of this pipeline is to produce a density map, i.e., a map indicating the probability of object presence at each spatial location in the image. Integrating this map then allows an estimation of the total number of objects.

An overview of the proposed pipeline is shown in Fig. 3.1. The following sections describe each component of this approach in detail.

3.2 DINO feature extraction

Let the input image be $I \in \mathbb{R}^{H \times W \times 3}$, where H and W denote the image height and width, respectively. The image is fed into the DINOv2 model $f(\cdot)$. The output $f(I)$ produces a dense feature map $F \in$

$\mathbb{R}^{H_{\text{new}} \times W_{\text{new}} \times D}$, where $H_{\text{new}} = \frac{H}{\text{patch\_size}}$, $W_{\text{new}} = \frac{W}{\text{patch\_size}}$ and D corresponds to the dimensionality of the feature embeddings.
The input image must be compatible with the model's patch size so that DINOv2 can divide it into patches of size patch_size × patch_size, here the patch_size is equal to 14. Each image is first resized to a resolution of 840×840 pixels at the beginning of the process. However, if one of the provided exemplars is considered too small (i.e., with an area of less than 9 patches in the feature map), the resolution is increased to 1680×1680 in order to obtain a higher-resolution representation and finer object features. Each element of F therefore represents a feature vector associated with a specific patch of the input image.

3.3 Similarity Map

The strength of DINO-based models lies in their ability to produce robust feature representations, capturing high-level properties such as object parts and scene geometry, regardless of the image domain [17, 24]. By analyzing the similarity between the central patch of a given template and all other patches in the feature map F, we observe in Fig. 3.2 a high similarity with objects of the same type, despite variations in size, shape, or color. However, color differences appear to be less discriminative compared to other visual attributes.

Moreover, these similarity patterns reveal an implicit segmentation of the objects of interest, which we leverage in our approach. For each template t_i, we extract its central patch representation x_i, where i indexes the N exemplar templates provided. We then compute the cosine similarity between x_i and all patches in F, resulting in N similarity maps $\text{SimMap}_i \in \mathbb{R}^{H_{\text{new}} \times W_{\text{new}}}$ where each entry encodes the similarity between a spatial patch and the corresponding template center.

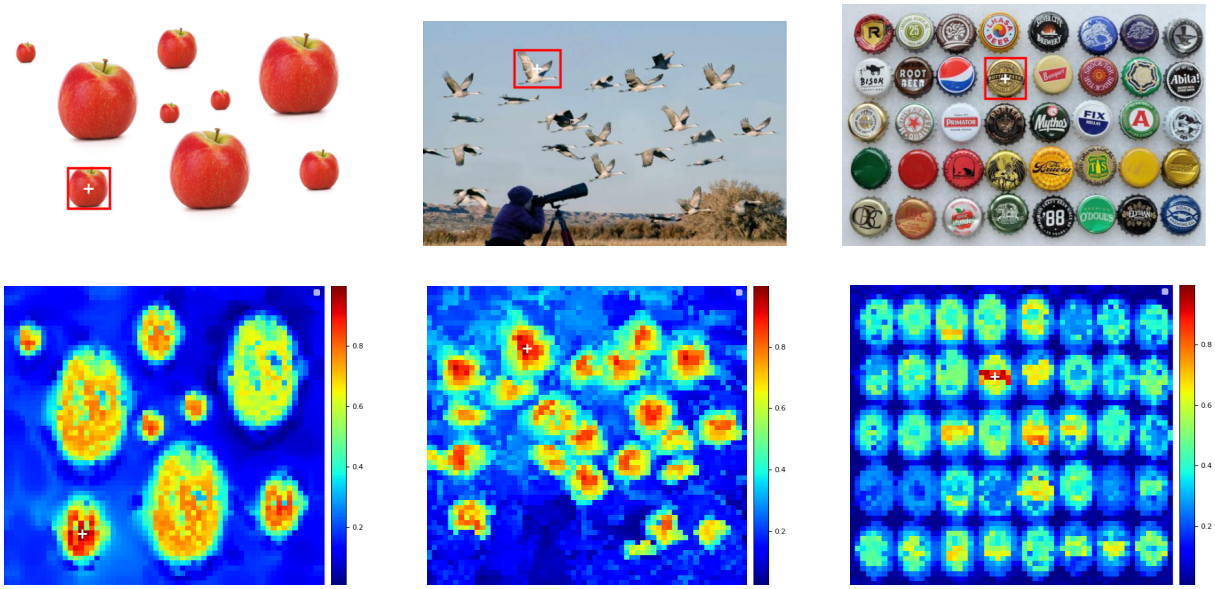


Figure 3.2: **Illustration of cosine similarity maps.** The top row presents the original images with the given exemple (template), while the bottom row shows the corresponding cosine similarity maps computed by comparing each patch to the central patch of the selected region.

As shown in Fig. 3.3, each template provides complementary information regarding similarity across the image. In particular, smaller objects tend to exhibit higher similarity with templates of similar scale, while larger objects respond more strongly to templates matching their size. To aggregate this information, we construct a final similarity map $S \in \mathbb{R}^{H_{\text{new}} \times W_{\text{new}}}$ defined as

$$S_{ij} = \max_n \left(\text{SimMap}_n(i, j) \right).$$

This operation allows us to retain the strongest similarity response at each spatial location, effectively combining information from templates of different scales and shapes.

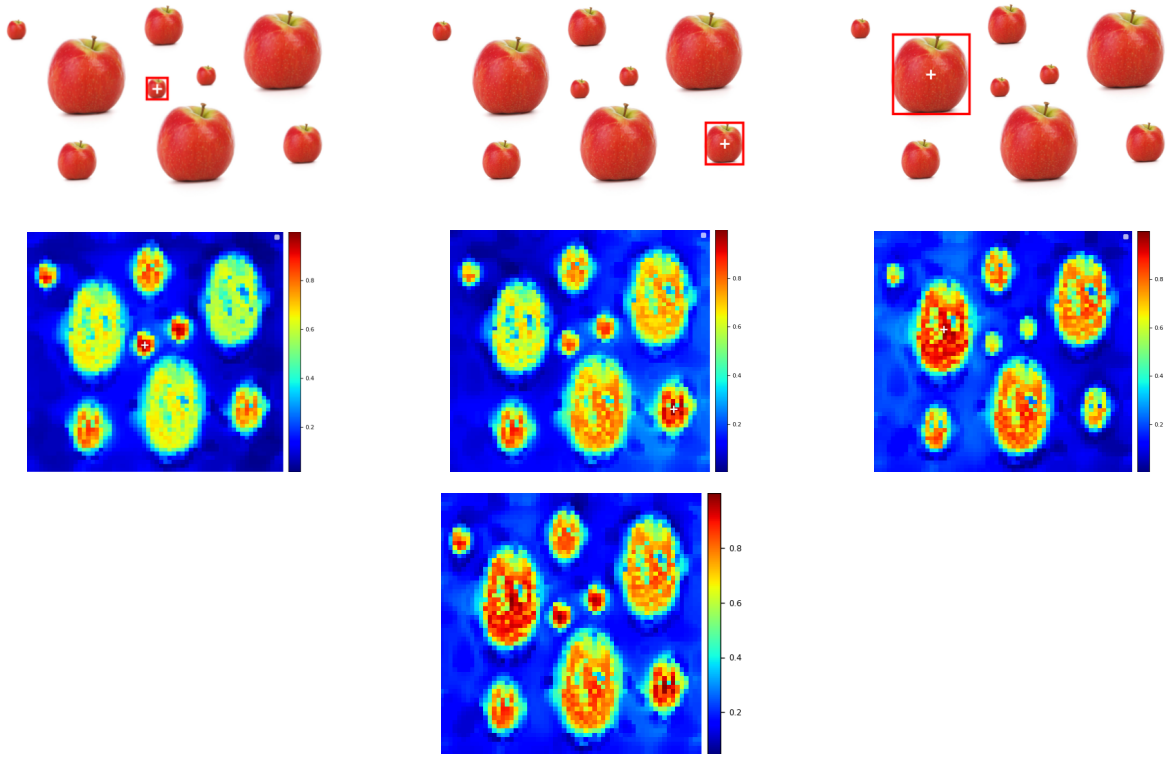


Figure 3.3: **Comparison of cosine similarity maps obtained using different templates.** The first row shows the original image with three regions of interest (templates). The second row presents the corresponding cosine similarity maps computed for each template by comparing each patch to the central patch of the selected region. The third row shows the final aggregated similarity map.

However, as illustrated in Fig. 3.4, each object is not always best represented by the most visually similar template. A closer analysis shows that similarity scores across the different maps are often very close for a given object, making it difficult to clearly associate a patch with a specific template. This ambiguity becomes problematic in the context of density-based counting, where each pixel contributes to the final object count.

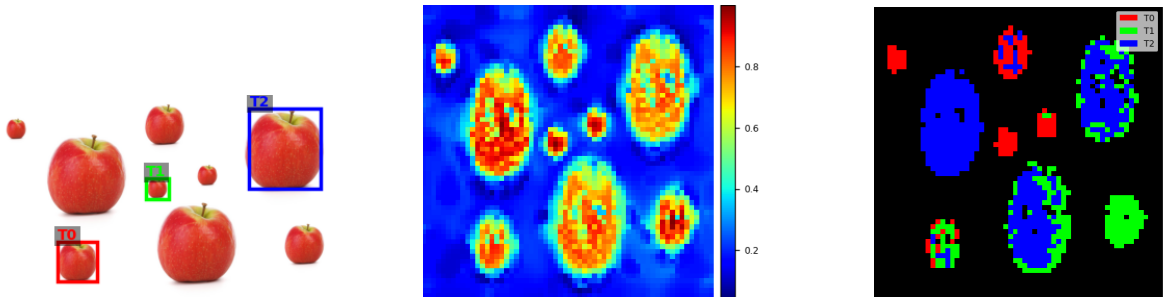


Figure 3.4: **Template matching using cosine similarity.** From left to right: original image with template, similarity map, and thresholded color map (cosine similarity ≥ 0.5). In the color map, each pixel is assigned the color of the template that achieves the highest cosine similarity score at that location. The results show that dominant responses are not necessarily associated with templates of matching size.

In such approaches, a density map assigns a weight to each spatial location, and the total count is obtained by integrating over this map. Ideally, objects of different sizes should contribute equally to the final count. However, without additional information, smaller objects may contribute less than larger ones due to their reduced spatial extent.

Therefore, it is desirable to distinguish objects based on their scale and assign appropriate weights

to ensure consistent contributions across different object sizes. While the similarity map provides valuable information—such as implicit object segmentation, background discrimination, and robust similarity scores despite variations in appearance—it lacks the necessary cues to properly normalize object contributions.

In summary, although the similarity map captures essential structural and semantic information, additional mechanisms are required to assign accurate weights to each object and achieve robustness to scale variations.

3.4 Convolutional Map

CountingDINO [18], an unsupervised and training-free class-agnostic object counting method, relies on a convolutional map that acts as a heatmap, indicating the likelihood of object presence at each spatial location. This map provides complementary information to similarity-based approaches by emphasizing spatial activation patterns associated with the target objects.

In this stage, we construct a set of convolutional maps that measure the similarity between the global feature representation and each template. For each template t_i, defined by a bounding box b_i, we first extract a corresponding feature representation using ROI alignment:

$$R_i = \text{ROIAlign}(F, b_i) \in \mathbb{R}^{k_i \times k_i \times C},$$

where k_i is the predefined output size representing the template's size. This operation preserves spatial alignment by sampling features from the feature map F.

Each extracted feature R_i is then used as a convolutional kernel applied over the feature map F. Formally, we compute:

$$\text{ConvMap}_i = \text{Conv2D}(F, R_i) \in \mathbb{R}^{(H_{\text{new}}-k_i+1)\times(W_{\text{new}}-k_i+1)},$$

where the convolution operation measures the similarity between the template representation and all spatial locations in F.

This process is repeated for all N templates, resulting in a set of convolutional maps $\{\text{ConvMap}_i\}_{i=1}^{N}$. Each map highlights regions that exhibit strong correspondence with a particular template in terms of appearance and structure.

These convolution maps can be interpreted as spatial response maps, where higher values indicate a stronger match between the local feature representation and the template. As illustrated in Fig. 3.5, they provide coherent and localized activations over the objects of interest.

Following the same principle as for the similarity maps, we construct an aggregated convolution map $C \in \mathbb{R}^{H_{\text{new}} \times W_{\text{new}}}$. Since each ConvMap_i has a spatial support that depends on the size of its corresponding template, the resulting maps may have different effective receptive fields. To ensure spatial alignment, we resize each convolution map to match the original feature map resolution. This is achieved by applying zero-padding, which allows us to preserve the localization of high-activation regions with respect to the original image. We then define the aggregated convolution map as:

$$C_{ij} = \max_n \left(\text{ConvMap}_n^{\text{resized}}(i, j) \right).$$

By analyzing the color map of C in Fig. 3.6, we observe that regions of high intensity are well aligned with the most similar templates. In addition to this aggregated map, we extract local maxima from C using a peak detection method (e.g., `peak_local_max`). We then filter these maxima by applying a threshold on their intensity values, retaining only the most significant responses. This set of filtered local maxima allows us not only to localize regions of high activation in the convolution map, but also to associate each region with the template that produced the strongest response.

We therefore obtain two maps, S and C, which both contain rich information derived from the feature map F. These maps provide an estimate of the likelihood of object presence at each spatial location.

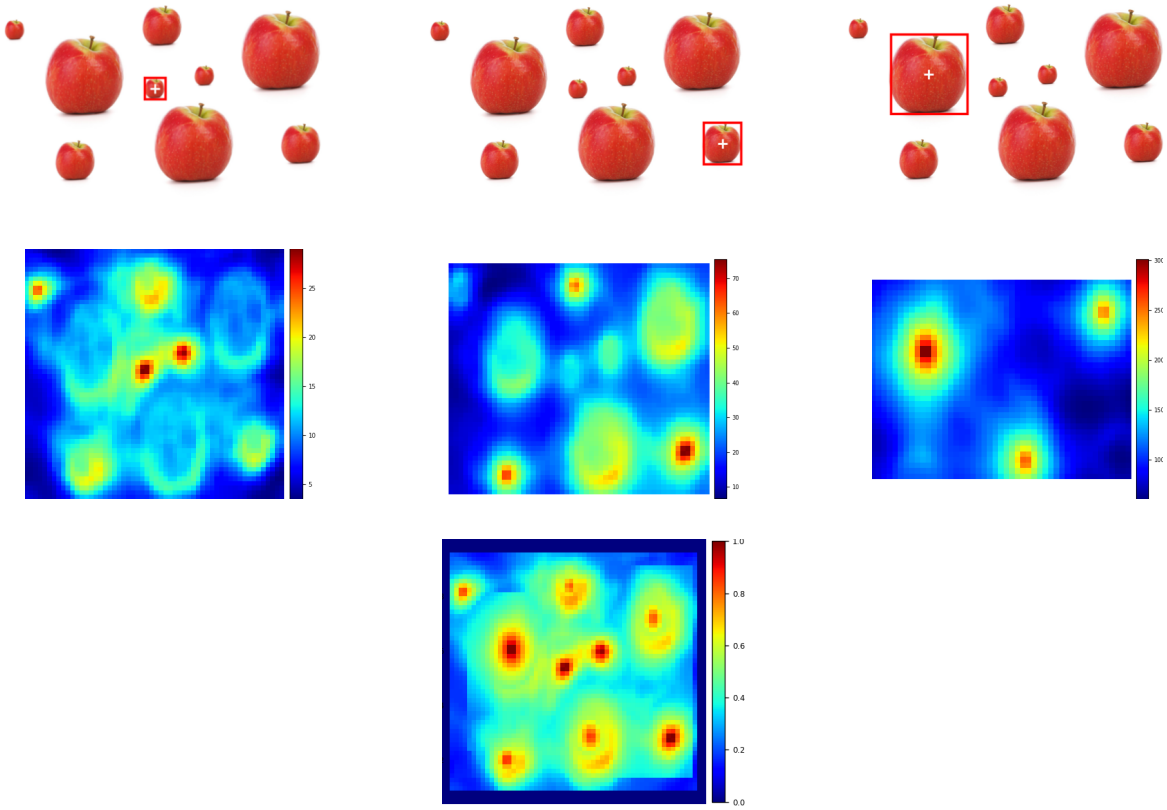


Figure 3.5: **Feature-space template matching and aggregation.** The first row shows the input images along with the selected template. The second row presents the corresponding convolution response maps. The third row shows the aggregated response map obtained by combining the individual convolution maps (e.g.,via max pooling).

The similarity map S offers a good implicit segmentation of the objects, while the convolutional map C provides information about which template best matches each region. The next step is thus to perform a post-processing of these two maps in order to construct the final density map.

3.5 Density map

Both information maps S and C have to be leveraged. The final objective is to associate a normalization coefficient specific to each template. Intuitively, each template should contribute equally to the counting process regardless of the object scale. Therefore, a dedicated normalization coefficient is required for each template to compensate for scale-dependent variations in the similarity responses. Consequently, the construction of the density map is formulated as a labeling problem [3], where each object is assigned to the template it is most similar to, in order to ensure a consistent association between objects and their closest template representations.

The goal is for each apple in the colormap to be best represented by the label of the corresponding template in terms of size, while also correctly identifying the background. This therefore corresponds to an $N + 1$ labeling problem, involving the N templates and the background.

The graph-cut framework is formulated as the minimization of the following energy function:

$$E(L) = \sum_{p \in \mathcal{P}} D_p(L_p) + \sum_{(p,q) \in \mathcal{N}} V(L_p, L_q), \quad L_p \in \mathcal{L} = \{0, 1, \ldots, N\},$$

where $\mathcal{P}$ denotes the set of pixels, $\mathcal{N}$ the set of neighboring pixel pairs, and $\mathcal{L}$ the label space composed of N object templates and one background label.

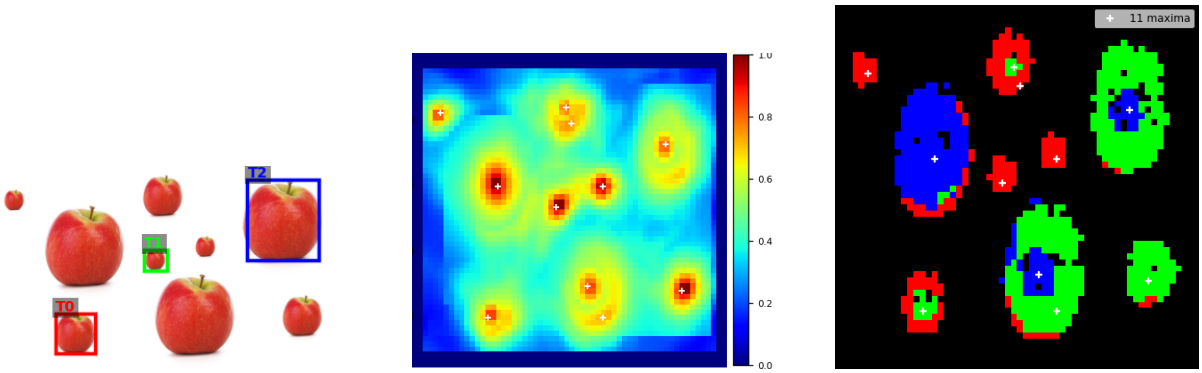


Figure 3.6: **Template matching via convolution.** From left to right: original image with template, convolution response map, and thresholded color map (response ≥ 0.5). In the color map, each pixel is assigned the color of the template that achieves the highest cosine similarity score at that location. Local maxima of the convolution map are indicated with a white cross. The locations of maximum response are consistent with the reference template, indicating effective detection of similar structures.

The data term for object labels is defined as:

$$D_p(l) = \big(1 - s_l(p)\big) + \alpha_{\text{dist}} \min_{m \in \mathcal{M}_l} \|p - m\|_2, \quad \forall l \in \{0, \ldots, N-1\},$$

where $s_l(p)$ denotes the similarity score between pixel p and template l, $\mathcal{M}_l$ represents the set of local maxima associated with template l, and the coefficient α_{dist} is a normalization factor used to scale the Euclidean distance term to the range $[0, 1]$.

For the background label, the data term is defined as:

$$D_p(\text{BG}) = \big(1 - \text{bg\_prob}(p)\big) + \alpha_{\text{dist}} \min_{m \in \mathcal{M}_{\text{all}}} \|p - m\|_2 + \gamma\big(1 - \text{conv\_map}(p)\big),$$

where $\text{bg\_prob}(p)$ is the background probability, $\mathcal{M}_{\text{all}}$ denotes the union of all local maxima, and $\text{conv\_map}(p)$ is the score obtain on the convolution map for the pixel p. The parameter γ controls the contribution of this additional constraint.

The smoothness (regularization) term is defined as:

$$V(L_p, L_q) = \begin{cases} 0 & \text{if } L_p = L_q, \\ \lambda & \text{otherwise,} \end{cases} \quad \forall (p, q) \in \mathcal{N},$$

where λ controls the penalty for label discontinuities between neighboring pixels.

As shown in Fig. 3.7, background detection is highly accurate, eliminating the need for thresholding to identify apples in the color map thanks to the introduction of an explicit background label. In addition, the regularization term promotes spatial smoothness between labels, effectively reducing noise and producing coherent regions. The data term further ensures that the correct label is consistently assigned to each apple, resulting in robust segmentation performance.

3.6 Post processing

Once the density map has been correctly labeled, we exploit the available prior information. We consider N different templates, each assumed to correspond to a single object. For each template, we compute a normalization coefficient by summing the pixel values of the cosine similarity map within the regions assigned to that template. Before summation, the background pixels are set to zero using the label map obtained via graph cut segmentation. This procedure yields a distinct normalization coefficient for each template.

Next, we iterate over all pixels of the similarity map and normalize each pixel value by the coefficient associated with its corresponding label in the graph cut segmentation. For greater accuracy, if

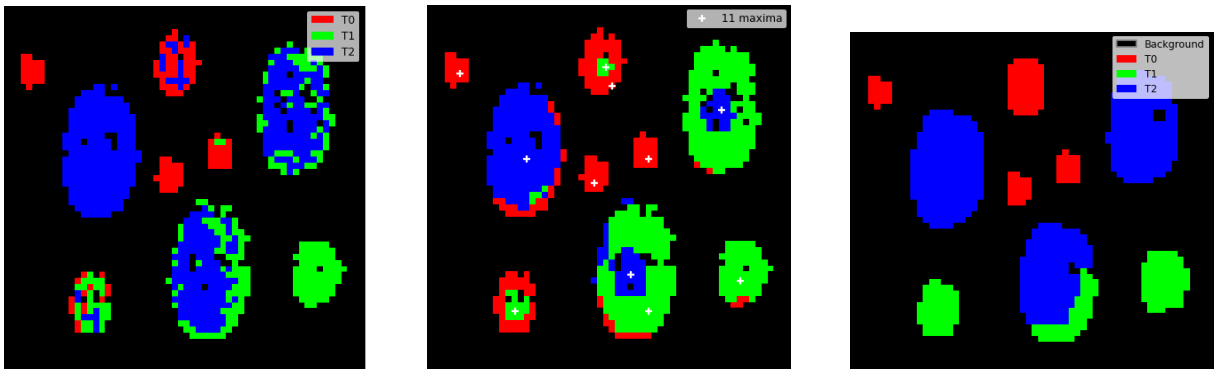


Figure 3.7: **Comparison of threshold-based and graph-cut approaches for labeling.** The first two maps are obtained by applying a threshold (≥ 0.5) to cosine similarity and convolution responses, respectively. The final result is computed via graph cut energy minimization without thresholding, yielding a more globally consistent and spatially coherent segmentation.

all templates are of similar sizes, we simply take the average of the sums as the coefficient. Finally, all normalized pixel contributions are summed to obtain the final object count prediction.

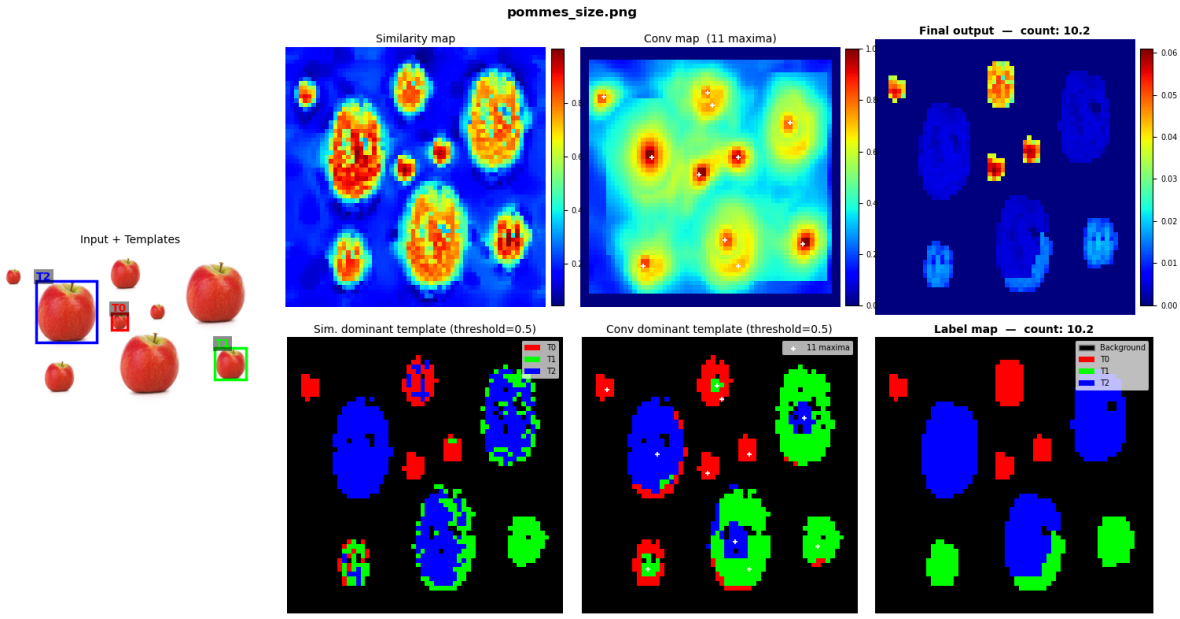


Figure 3.8: **Overview of the complete pipeline for template-based density estimation and counting.** The first row illustrates the progression from input image with templates, to cosine similarity-based matching map, convolution-based responses map, and final density map. The second row shows the corresponding thresholded color maps (threshold ≥ 0.5) and the final labeling obtained via graph cut optimization.

As shown in Fig. 3.8, the predicted value of 10.2 is consistent with the observed density map. Indeed, 8 out of 9 apples are correctly assigned to their corresponding labels, which theoretically corresponds to 8 objects. One apple (a medium-sized one) is assigned to the label of a small apple. Since the scale ratio between these two categories is approximately 2, a medium-sized apple corresponds to two small apples. As a result, this instance contributes two objects instead of one, leading to a final estimate of approximately 10 apples.

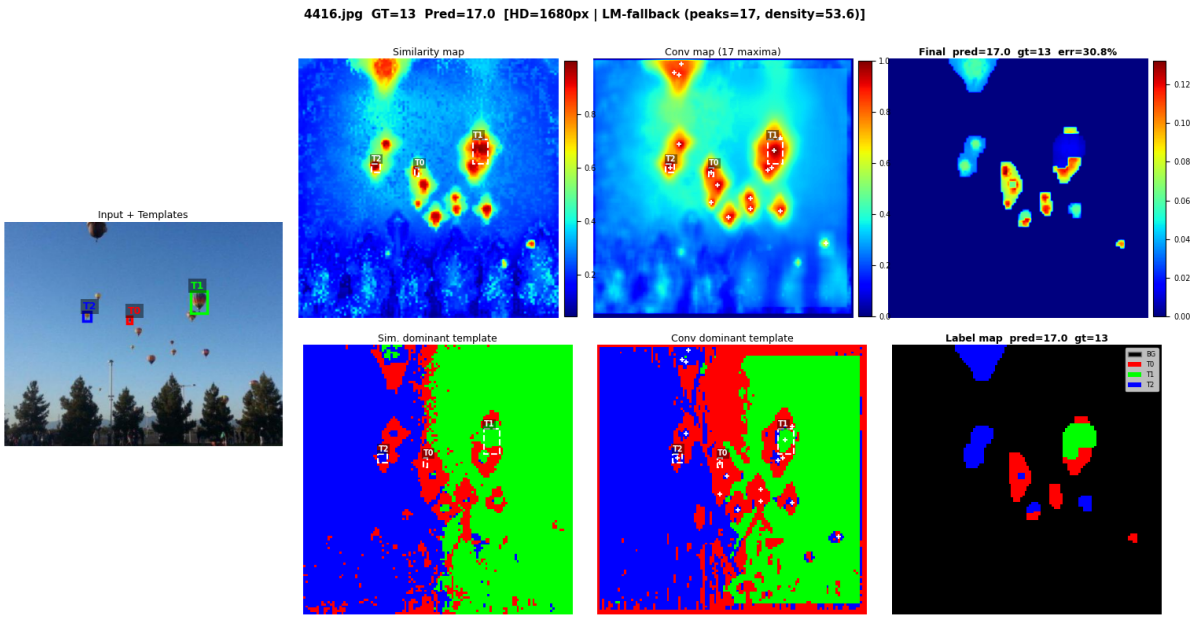


Figure 3.9: **LM-Fallback usage example**. From left to right: the input image with exemplar templates, the similarity map, the convolution map with detected local maxima (crosses), and the final density prediction after Graph-Cut refinement. Each time presented with their respective color map

3.7 LM-Fallback

A limitation arises when there is a large variation in template sizes combined with the presence of very small templates. As illustrated in Fig. 3.9, templates T_0 and T_2 have areas smaller than 196 pixels. Consequently, after the image is divided into patches, these templates are entirely contained within a single patch. During the construction of the convolution map, the convolution operation involving such small templates becomes significantly less restrictive than the convolution using a normally sized template such as T_1. As a result, these small templates produce both high similarity scores and high probabilities of presence at many object locations throughout the image. During the graph-cut stage, objects are therefore predominantly associated with these small templates. Because of their corresponding coefficients are small, the resulting values in the density map become excessively large, leading to an overestimated prediction of 53.6 balloons. To address this issue, the number of local maxima is used directly as the final prediction in such cases. When the image is processed at high resolution, the convolution map alone provides a reliable estimation of both object presence and localization through its local maxima.

Chapter 4

Experiments and Results

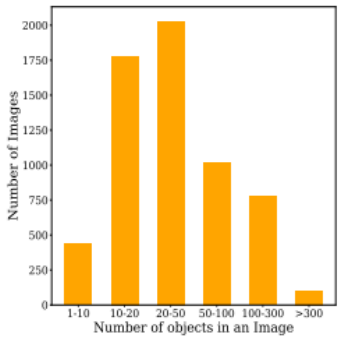
4.1 Evaluation Protocol

4.1.1 Datasets

FSC-147



(a) Image categories and number of images for each category in our dataset.



(b) Number of images in several ranges of object count.

Dataset	Images	Categories	Annotation type: Dot	Annotation type: Bounding Box
UCF CC 50 [15]	50	1	✓	✗
Shanghaitech [55]	1198	1	✓	✗
UCF QNRF [16]	1535	1	✓	✗
NWPU [49]	5109	1	✓	✗
JHU Crowd [43]	4372	1	✓	✓
CARPK [14]	1448	1	✓	✓
Proposed	6135	147	✓	✓

(c) Comparison with popular counting datasets.

Figure 4.1: **Statistics of the FSC-147 datasets. Source:** [22] Presentation of the categories and the number of images included in each of them, the number of objects and comparison with other object counting datasets.

Originally, Viresh Ranjan et al. [22] introduced the FSC-147 dataset in response to the lack of datasets specifically suited for class-agnostic object counting. Most existing datasets focused on a single object category, such as cars, animals, cells, or people, while datasets containing multiple categories were not necessarily appropriate for evaluating class-agnostic counting methods. In particular, they often lacked images exhibiting strong variations in object shape or appearance, significant occlusions, or a sufficiently large number of instances. Indeed, object counting aims to automate complex counting tasks, whereas humans can easily count objects when only a few instances are present in an image.

The dataset was constructed from images collected from web sources such as Flickr, Google, and Bing. The authors first selected a set of diverse object categories and then retrieved corresponding images while applying filtering criteria to ensure high image quality, limited appearance variation, and a minimum number of object instances per image (at least seven). The resulting dataset contains 6,135 images distributed across 147 object categories, with an average of 56 instances per image, as illustrated in Fig. 4.1. Representative examples of images from the FSC-147 dataset are shown in Fig. 4.2.

To suit the object counting task, each image is provided with dot annotations for every object instance used to generate ground-truth density maps and compute the counting loss during training, along with three exemplar bounding boxes used for few-shot object counting. The dataset is divided into three

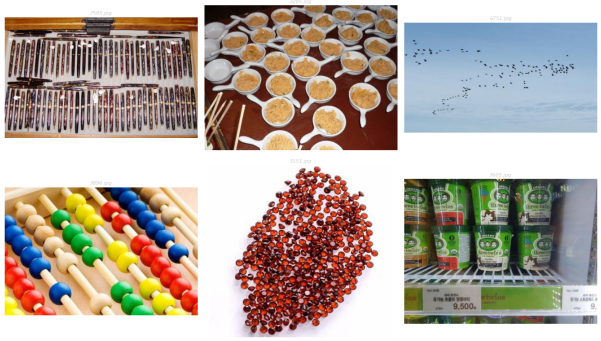


Figure 4.2: **Examples from the FSC-147 dataset.**

splits: 89 object categories for training and 29 categories each for validation and testing. This split ensures that, at every stage of the evaluation process, the model is tested on unseen object categories, thereby preserving the class-agnostic setting. Since its introduction, FSC-147 has become a fundamental benchmark for class-agnostic object counting and a standard basis for comparing models [6, 8–10, 12–14, 16, 18, 20, 22, 23].

FSCindu

The FSC-147 dataset is primarily composed of everyday objects, such as fruits, animals, or cosmetic products. However, it is also relevant to evaluate counting methods on datasets with a more industrial setting, where images are acquired in a more uniform manner and objects are more clearly distinguishable. The FSCIndu dataset was introduced by Nicolas Bourez [2] for this purpose. It consists of 591 images distributed across 51 categories.

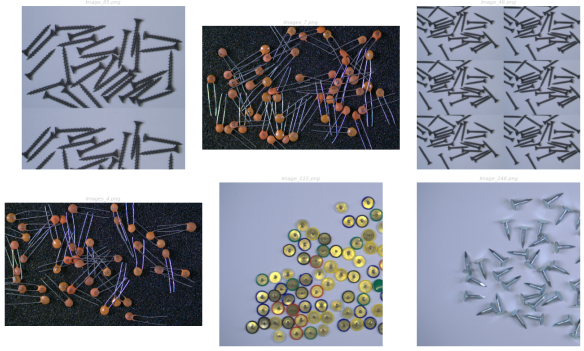


Figure 4.3: **Examples of the FSC-indu dataset.**

First, a subset of images from the FSC-147 dataset was selected (297 images from 46 categories). These images correspond to scenes where objects are clearly distinguishable. In addition, high-resolution images (1920×1920) of nuts, pins, screws, nails, and capacitors are included, together with augmented versions of these images. This industrial subset contains 294 images across 5 categories, as illustrated in Fig. 4.3.

Images from this dataset are therefore technically easier to analyze due to the reduced noise and simpler backgrounds. However, they may still present challenges for counting methods because of aligned object arrangements, high object density, and the presence of very small and repetitive objects, which require finer precision.

4.1.2 Metrics

We primarily rely on the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) to compare our method with state-of-the-art approaches. The MAE is defined as the main evaluation metric commonly used for comparing counting models [6, 8–10, 12–14, 16, 18, 20, 22, 23], while RMSE provides additional insight into the robustness of the predictions under larger errors. The MAE is defined as:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_i - y_i| \,,$$

where $\hat{y}_i$ and y_i denote the predicted and ground-truth object counts for image i, respectively.

The RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}.$$

In addition to these standard metrics, the method was further analyzed across different density ranges in order to evaluate their impact on the results. For each density interval, the MAE and the median NAE were computed, where the Median Normalized Absolute Error (MedNAE) corresponds to the median of the Normalized Absolute Error (NAE) over the dataset. The NAE for a single image is defined as:

$$\mathrm{NAE}_i = \frac{|y_i - \hat{y}_i|}{y_i},$$

where y_i and $\hat{y}_i$ denote respectively the ground truth and predicted counts for image i. The MedNAE is then computed as:

$$\mathrm{MedNAE} = \mathrm{median}(\mathrm{NAE}_i).$$

4.1.3 comparaison with SOTA

To quantitatively evaluate our approach, we adopt the experimental settings of CountingDINO [18] and Nicolas Bourez [2]. CountingDINO reports results on several supervised state-of-the-art methods, including FamNet [22], RCAC [8], BMNet+ [23], CounTR [14], LOCA [6], PseCo [9], CACViT [26], and DAVE [20]. It also includes training-free methods such as TFPOC [13] and OmniCount [16], which, similarly to our approach, do not rely on a deep learning network for density map prediction, although they are based on supervised backbones for feature extraction. In addition, results for UnCounTR and UnCounTRv2 [12] are reported; these methods are unsupervised and based on DINO models, but they are not training-free as they train an additional network to predict the density map. We also compare our method with CountingDINO which employs a DINOv2 ViT-L/14 backbone with registers, comprising 300M parameters, whereas we use a ViT-B/14 with registers with 86M parameters [17]. Regarding the FSCIndu dataset, we restrict our comparison to previously mentioned supervised class-agnostic counting methods as well as CountingDINO.

4.2 Quantitative Results

4.2.1 Results on the FSC-147 dataset

The results of our method on the FSC-147 dataset, together with those of competing approaches, are presented in Table 4.1. Although our method is both unsupervised, relying directly on a pre-trained unsupervised vision transformer without additional supervision, and training-free, requiring no deep learning module for density map prediction, it achieves highly competitive results compared to state-of-the-art methods.

First, our approach outperforms the previous state-of-the-art unsupervised and training-free method, CountingDINO. Although the overall results remain relatively close, a significant improvement can be observed in terms of RMSE on the test split of the dataset. This indicates that our method tends to produce smaller errors and fewer outliers than CountingDINO.

Compared to unsupervised but non-training-free approaches, such as UnCounTR, which relies on a trained CNN to predict a density map from the feature map and encoded exemplars, our method substantially improves both MAE and RMSE on the validation and test splits. These results demonstrate

Method	Unsup.	Training-Free	Validation		Test	
			MAE ↓	RMSE ↓	MAE ↓	RMSE ↓
FamNet (CVPR '21) [22]	✗	✗	23.75	69.07↑	22.08↑	99.54↑
RCAC (ECCV '22) [8]	✗	✗	20.54	60.78↑	20.21	81.86↑
BMNet+ (CVPR '22) [23]	✗	✗	15.74	58.53	14.62	91.83↑
CounTR (BMCV '22) [14]	✗	✗	13.13	49.83	11.95	91.23↑
LOCA (ICCV '23) [6]	✗	✗	10.24	32.56	10.79	56.97↑
PseCo (CVPR '24) [9]	✗	✗	15.31	68.34↑	13.05	112.86↑
CACViT (AAAI '24) [26]	✗	✗	10.63	37.95	9.13	48.96↑
DAVE (CVPR '24) [20]	✗	✗	8.91	28.08	8.66	32.36
TFPOC (WACV '24) [13]	✗	✓	–	–	19.95	132.16↑
OmniCount (AAAI '25) [16]	✗	✓	–	–	18.63	112.98↑
UnCounTR (CVPR '24) [12]	✓	✗	36.93↑	106.61↑	35.77↑	130.34↑
UnCounTRv2 (CVPR '24) [12]	✓	✗	–	–	28.67↑	118.40↑
CountingDINO (WACV '26) [18]	✓	✓	42.29	87.87	30.05	90.3
CountingDINOv2 (WACV '26) [18]	✓	✓	25.48	57.38	20.93	71.37
Our method	✓	✓	**24.8**	**59.27**	**20.91**	**47.9**

Table 4.1: **Comparison with state-of-the-art methods on the FSC-147 dataset.** Methods are grouped according to whether they are unsupervised and training-free. The best-performing method within each partition is underlined, while the best unsupervised and training-free method is highlighted in **bold**. Metrics from methods that are supervised and/or not training-free and still perform worse than our approach are marked with ↑.

that highly effective counting performance can be achieved directly from frozen vision representations without requiring task-specific training for density prediction.

Regarding supervised methods such as Omnicount and TFPOC, which rely on supervised backbones such as CLIP [21] or SAM [11], our method achieves slightly lower performance in terms of MAE but significantly reduces the RMSE. This suggests that, although supervised methods may slightly improve average accuracy, our approach better handles difficult cases such as occlusions or ambiguous exemplars, leading to more stable predictions.

Finally, when compared to more traditional supervised methods specifically trained for object counting, our approach remains highly competitive. In particular, it challenges FamNet, one of the foundational and simplest object counting models, and achieves lower RMSE values, further demonstrating its ability to handle the most difficult cases. In terms of MAE, our results approach those of FamNet and RCAC, showing that, despite relying on an unsupervised backbone and a training-free framework, our method can effectively compete with supervised approaches, highlighting the effectiveness of leveraging pretrained vision transformers as strong feature extractors for training-free counting.

Let us now analyze Fig. 4.4. First, a certain consistency can be observed across the histograms regardless of density. All histograms exhibit a similar shape, with a tendency toward higher values on the right side of the curve. This indicates a degree of stability in the method, which largely avoids catastrophic errors exceeding 100% and generally produces predictions within a moderate error range.

When examining the evolution of MedNAE across density intervals, we observe a gradual increase up to the 50–100 interval, from 18% in the 7–10 range to 32.4% in the 50–100 range, followed by a decrease to 27.5% and 22.6% for the 100–300 and >300 ranges, respectively. This trend is consistent with the increasing difficulty of the counting task. The subsequent decrease may reflect the model's ability to smooth predictions and better approximate high-density images, likely due to more regular density maps with fewer variations between instances.

By comparing MAE and MedAE, we observe that MAE is consistently higher than MedAE, with larger gaps in high-density ranges (100–300), where MAE reaches 50.76 versus a MedAE of 40.39, and in the >300 range, where MAE is 154.73 compared to a MedAE of 86.68. This discrepancy is attributable to the presence of certain outlier images that degrade the overall performance.

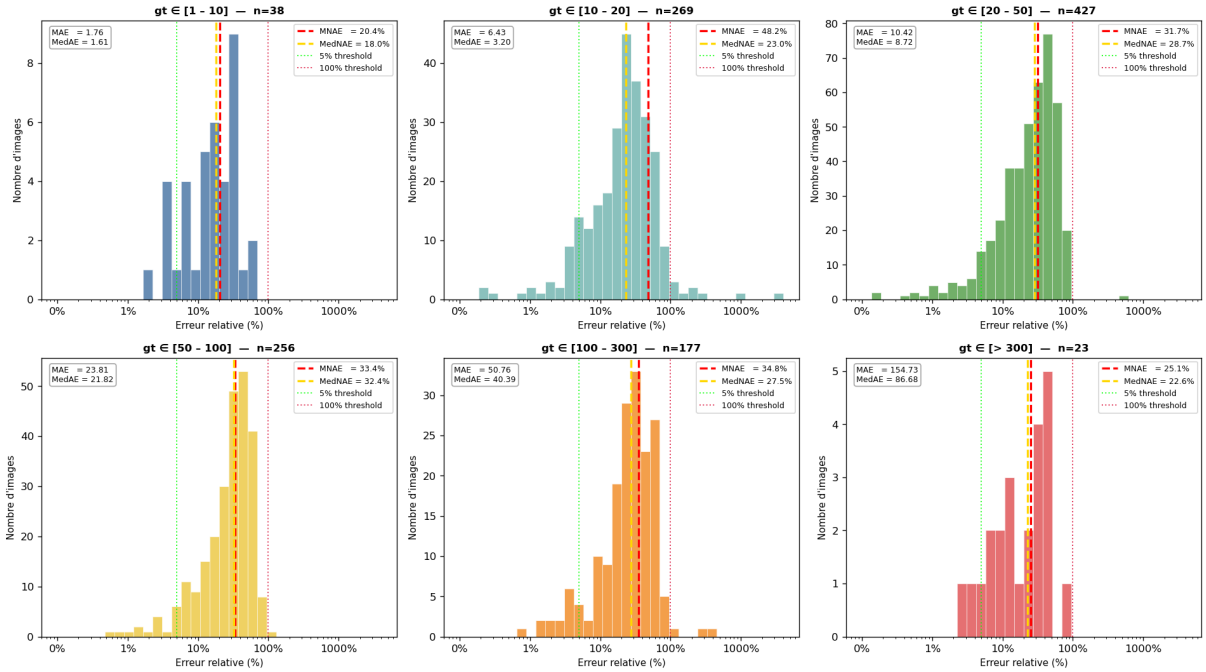


Figure 4.4: **Distribution of relative prediction errors per density object counts on the FSC-147 dataset's test split**. Images are split by ground-truth count. Each histogram uses a log-scaled x-axis; dashed lines mark the mean (MNAE) and median (MedNAE) relative errors, with absolute MAE and MedAE reported in the inset.

In low-density ranges (7–10), the MedNAE of 18% may appear high; however, it should be noted that in such cases, an error of a single object already corresponds to approximately 20% NAE. It is also important to highlight that MAE and MedAE remain almost identical (1.76 and 1.61), indicating the absence of outliers and an average error of 1.76 objects, which remains relatively precise. In contrast, the 10–20 range appears more problematic, exhibiting the highest MedNAE (48.2%) and the presence of outliers exceeding 1000% error, which may be due to poorly defined examples, as discussed in subsequent sections.

Overall, outliers remain relatively rare across the dataset, which is indicative of the robustness of the method. They are absent in extreme density ranges such as 7–10 and >300, and are mainly concentrated in the 10–20 range, suggesting the existence of a subset of images within this interval that is particularly challenging for the model.

In conclusion, the method appears stable and consistent in its predictions, achieving low errors in both low- and medium-density settings, while also being capable of smoothing and accurately approximating high-density images. The consistently lower median error compared to the mean further confirms the presence of a limited number of outliers in the dataset.

4.2.2 Results on the FSCindu dataset

The comparison of state-of-the-art methods on the FSCIndu dataset is reported in Table 4.2. Our method achieves an MAE of 14.64, outperforming CountingDINO, which obtains an MAE of 18.47. This highlights the ability of our model to better exploit the features provided by DINO, as well as its robustness and improved accuracy in the final density map estimation on more industrial and therefore less complex images.

In comparison with supervised CAC methods such as FamNet, BMNet+, and PseCo, our approach achieves superior results, which is particularly encouraging given that it does not rely on supervised training for object counting. This suggests that competitive performance can be obtained with a fully training-free approach, opening perspectives for leveraging features produced by vision transformers that learn directly from images without annotations.

More recent state-of-the-art models, such as CounTR, CACViT, and LOCA, achieve better performance than our method, with MAE values decreasing to 6.74.

Method	Unsup.	Training-Free	Test MAE ↓
FamNet (CVPR '21) [22]	✗	✗	18.62↑
BMNet+ (CVPR '22) [23]	✗	✗	17.35↑
PseCo (CVPR '24) [9]	✗	✗	15.37↑
CounTR (BMCV '22) [14]	✗	✗	12.22
CACViT (AAAI '24) [26]	✗	✗	8.46
LOCA (ICCV '23) [6]	✗	✗	6.74
CountingDINOv2 (WACV '26) [18]	✓	✓	18.47↑
Our method	✓	✓	**14.64**

Table 4.2: **Comparison with state-of-the-art methods on the FSCindu dataset.** The best supervised method is underlined, while the best unsupervised and training-free method is highlighted in **bold**. Metrics from supervised methods that still perform worse than our approach are marked with ↑.

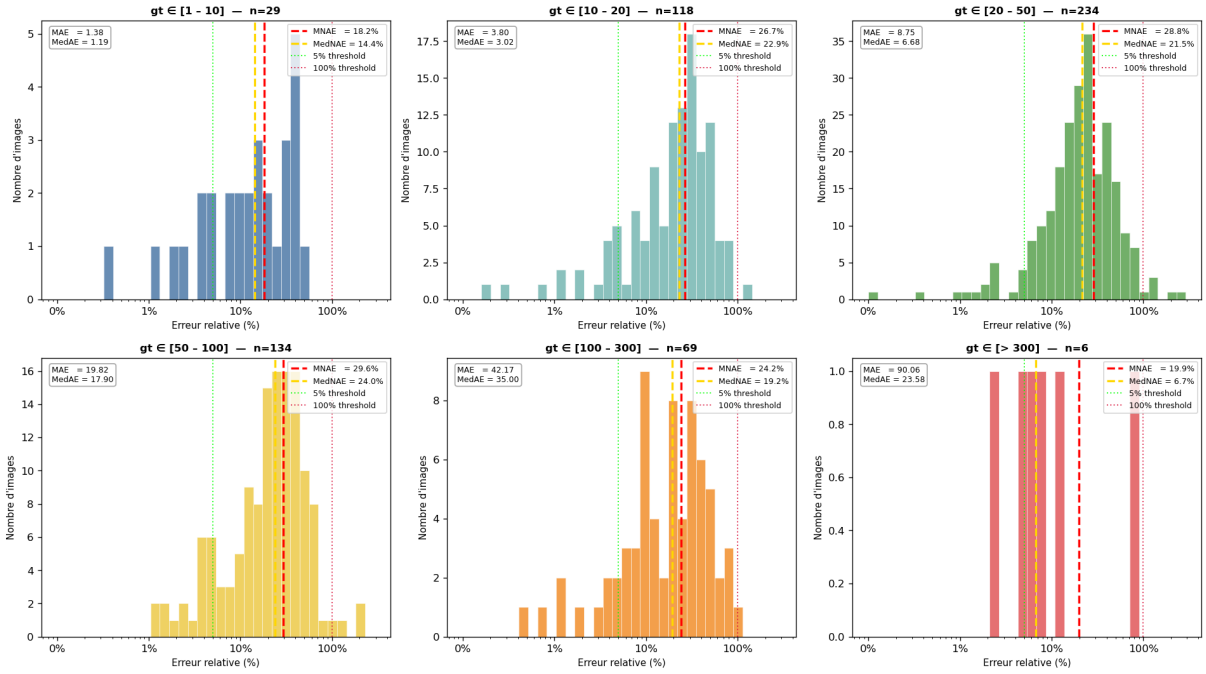


Figure 4.5: **Distribution of relative prediction errors per density object counts on the FSCindu dataset**. Images are split by ground-truth count. Each histogram uses a log-scaled x-axis; dashed lines mark the mean (MNAE) and median (MedNAE) relative errors, with absolute MAE and MedAE reported in the inset.

By observing Fig. 4.5, we note that the histograms remain consistent and similar across density ranges, indicating a degree of homogeneity in the results regardless of the density interval.

First, when comparing the MedNAE across density bins, the method performs better overall compared to the results on the FSC-147 dataset, particularly in the intermediate ranges (20–50 and 50–100). Unlike previous results, where a monotonic increase in MedNAE was observed up to a certain density range followed by a decrease, this dataset exhibits relatively stable MedNAE variations (between 14.4% and 24%). This suggests a more homogeneous dataset in terms of difficulty. Indeed, industrial-type images contain less complex backgrounds and fewer objects with appearances no similar to the target instances. A clear improvement is also observed for high-density cases (>100).

Regarding the gap between MAE and MedAE, it is significantly smaller than in the previous dataset. For instance, we obtain MAE and MedAE values of 1.38 and 1.19 for densities in the 1–10 range, 3.8 and 3.02 for 10–20, 8.75 and 6.68 for 20–50, and 19.82 and 17.90 for 50–100. This reflects consistent and robust performance without the presence of strong outliers or highly erroneous predictions. For higher densities, however, a larger gap is observed, indicating the presence of cases where the prediction deviates significantly from the ground truth.

Overall, this dataset yields more favorable results, which can be attributed to the nature of industrial images, reducing the number of outliers. Performance improves across all density ranges, and the method consistently demonstrates an ability to smooth and accurately approximate predictions in high-density scenarios (>100).

4.3 Qualitative Results

Regarding the qualitative results, Fig. 4.6, which presents successful cases, highlights three main patterns of favorable predictions.

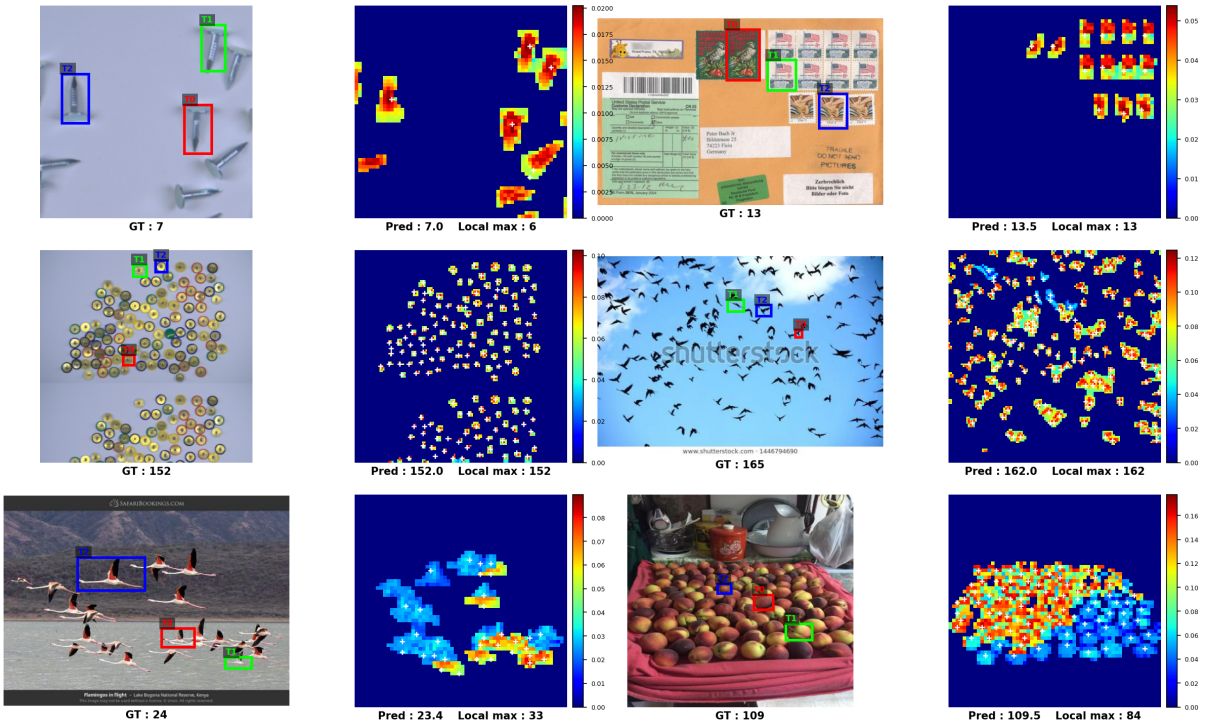


Figure 4.6: **Qualitative results (successful cases) on the FSC-147 and FSCindu datasets.** On the left, the input images are shown along with the provided exemplar templates and the ground truth (GT). On the right, the corresponding predicted density maps are displayed, together with the final count estimation. Local maxima are highlighted with crosses on the density map, indicating the predicted object locations.

Regular Object Layouts. In the first row, the method achieves nearly perfect predictions for objects that are well aligned, free of occlusions, and of medium size within the image, despite variations in size between the provided templates.

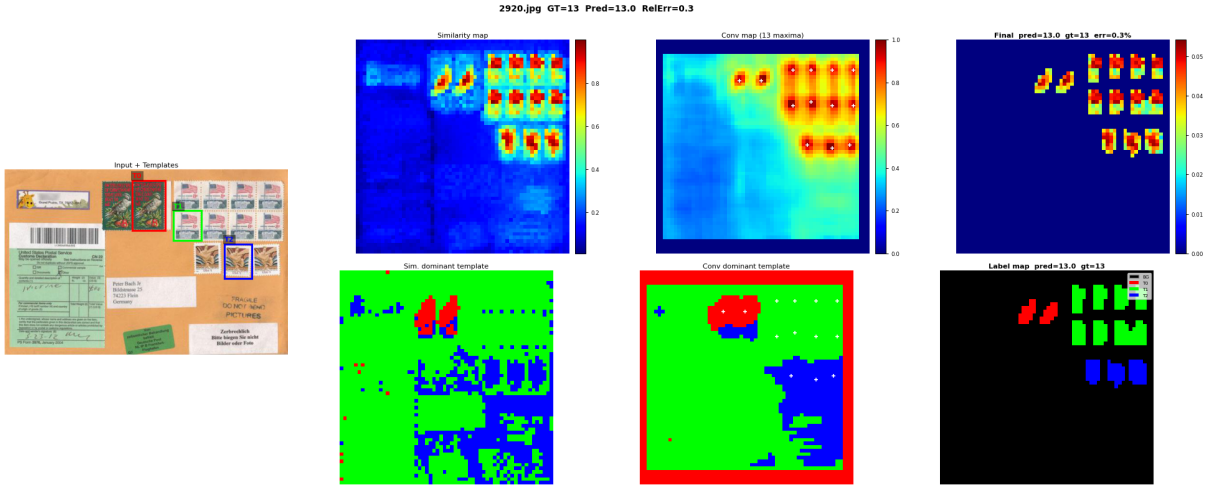


Figure 4.7: **Qualitative visualization on the FSC-147 dataset (stamp example).** From left to right: the input image with exemplar templates, the similarity map with dominant template assignments, the convolution map with detected local maxima (crosses), and the final density prediction after Graph-Cut refinement.

Fig. 4.7 presents a closer examination of the stamp image results. First, the similarity map shows highly accurate object segmentation directly from the extracted features, with strong similarities concentrated exclusively on the stamps. Then, using the convolution map and its local maxima, the graph-cut method successfully identifies both the background and the correspondence between each object and the most visually similar template. Consequently, the coefficient computed for each template is consistently assigned to objects of matching size and shape, leading to more accurate predictions.

Handling Small Templates and Scale Imbalance. The second row of Fig. 4.6 illustrates cases where both a small template, causing the image to be processed in HD mode, and large variations between templates are present. In these situations, the LM-Fallback strategy is used for prediction, preventing overestimation caused by the dominance of the smallest template.

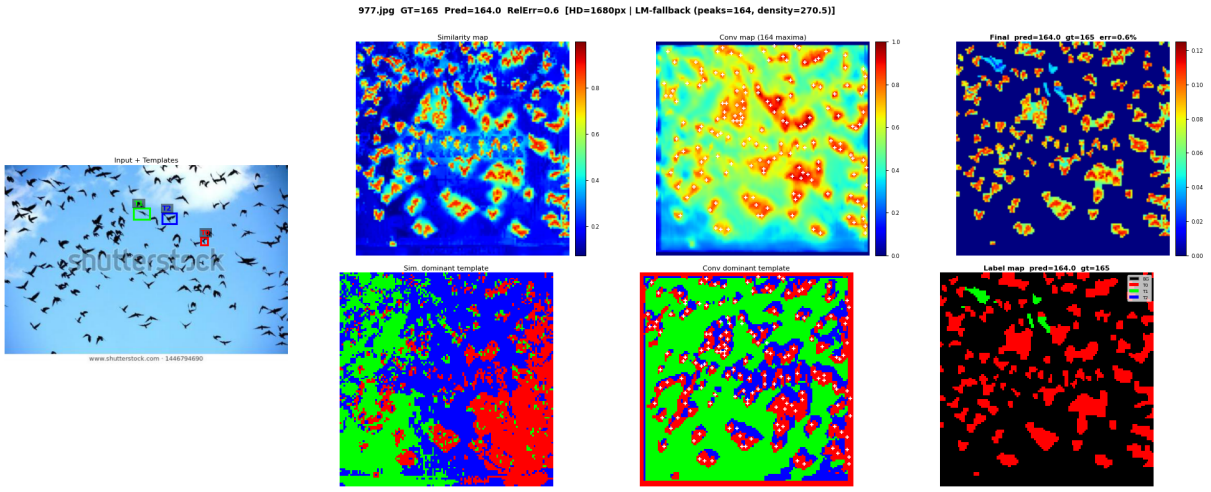


Figure 4.8: **Qualitative visualization on the FSC-147 dataset (bird example).** From left to right: the input image with exemplar templates, the similarity map with dominant template assignments, the convolution map with detected local maxima (crosses), and the final density prediction after Graph-Cut refinement.

Fig. 4.8 provides a more detailed analysis of a case relying on the LM-Fallback strategy. Although the similarity map correctly segments the birds, one of the provided templates is significantly smaller

in comparaison with the original image resolution (the red template). As a consequence, this template dominates the convolution map whenever a local maximum is detected. Indeed, the smaller template is less restrictive than the others, making the convolution response stronger at the location of other birds. This effect can also be observed in the graph-cut label assignment, where the small template dominates almost the entire map, resulting in excessively high values across the final density map and an initial prediction of 270 birds instead of the ground-truth value of 165. Fortunately, the LM-Fallback strategy introduced to address such situations enables a highly accurate estimation of 164 objects using the number of local maxima.

Shape and Scale Variations. Finally, the last row in Fig. 4.6 presents more complex images, involving variations in both object size and shape. Nevertheless, the model successfully assigns the correct labels to each object, allowing the density map to assign larger values to smaller objects than to larger ones. As a result, the final prediction remains nearly perfect.

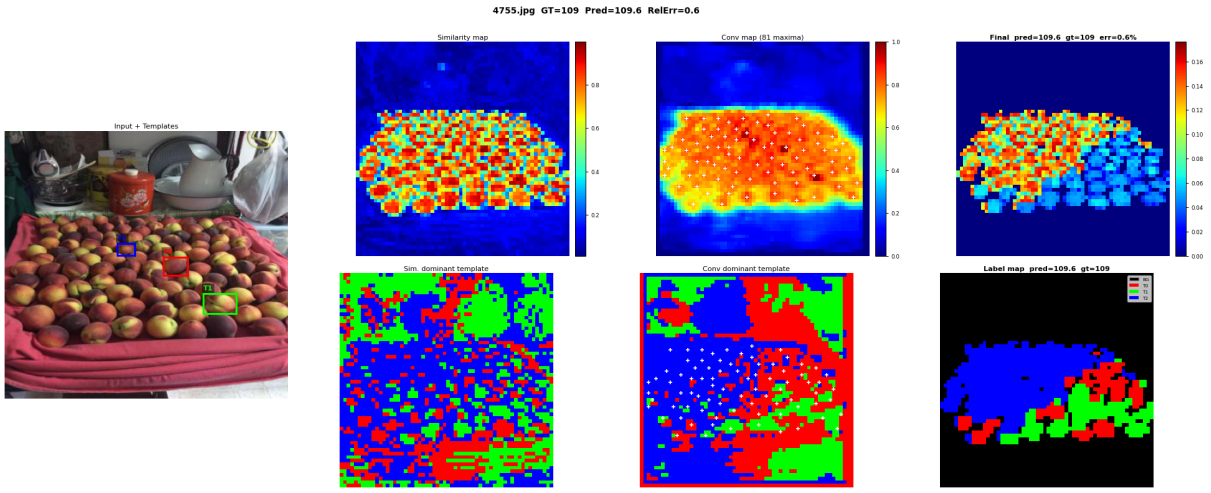


Figure 4.9: **Qualitative visualization on the FSC-147 dataset (apple example).** From left to right: the input image with exemplar templates, the similarity map with dominant template assignments, the convolution map with detected local maxima (crosses), and the final density prediction after Graph-Cut refinement.

Finally, Fig. 4.9 illustrates a more challenging example involving variations in object size and shape, as well as partial occlusions. Once again, the similarity map effectively discriminates between regions belonging to the target category and the background directly from the extracted features. However, significant variations in template size can be observed. In particular, apples located in the background appear visually smaller than those in the foreground. Providing representative templates for each scale allows the graph-cut-based density map construction to correctly distinguish smaller apples in the background (blue labels) from larger apples in the foreground (red and green labels). This behavior is also visible after map normalization, where larger apples are associated with lower values than smaller apples, enabling the integral over each apple, regardless of its size, to approximate a value close to one.

4.4 Failure Cases

We now examine the failure cases of our method. Once again, several recurring patterns leading to inaccurate predictions can be identified.

Too Small Templates. Fig. 4.10 presents example where the provided templates are extremely small compared to the original image resolution. In these images, the template areas are respectively 54, 45, and 54 pixels. Considering that the image is divided into patches of size 14×14 pixels within the vision transformer, with each patch associated with a distinct embedding, the selected object regions occupy

only a very small portion of their corresponding patches as observed on the figure representing the patch in which the object is located. Consequently, a large part of each patch contains background information (e.g., the ground surrounding the sheep). This results in similarity and density maps that correctly identify the object regions globally but also introduce substantial noise due to the strong similarities between the background surrounding the objects and the objects themselves. Adding a significant number of instances to the final prediction.

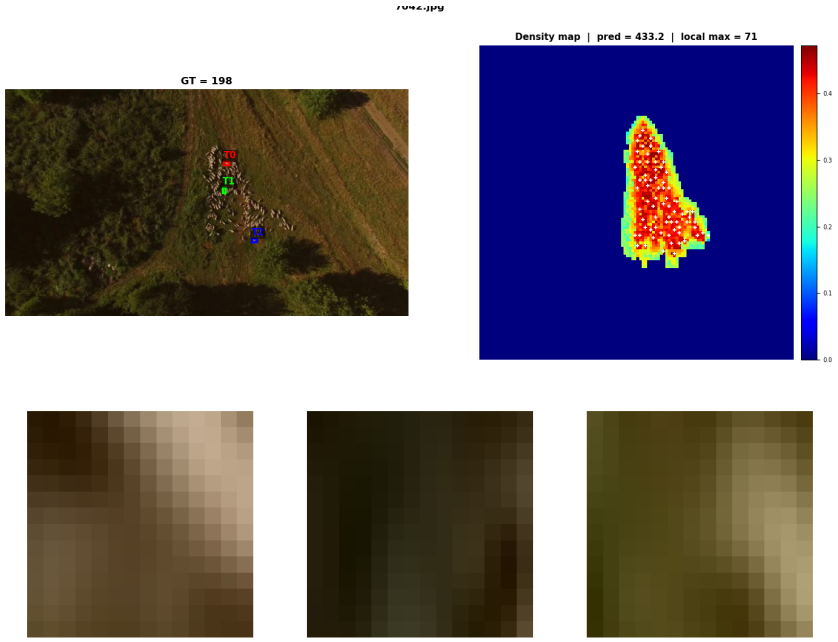


Figure 4.10: **Failure case analysis on small object templates.** The first row shows the input image with exemplar templates and ground truth (GT), followed by the predicted density map with local maxima (crosses) and the final count prediction. The second row illustrates the central patches of each template. Since the objects are very small, each sheep is fully contained within a single central patch. Brighter pixels correspond to sheep regions, while darker pixels represent the background.

LM-Fallback With Too Small Templates. Fig. 4.11 illustrates an example relying on the LM-Fallback strategy because of large variations in template size combined with the presence of a very small template. The template areas in this example are respectively 1472, 176, and 4158 pixels, corresponding approximately to 7, 1, and 21 patches in the encoded feature map. Once again, the patch associated with the smallest template contains background information, producing a similarity map with significant noise around each bird. However, the main issue arises from the convolution map construction. Since the convolution operation is performed between the template feature map and the image feature map, the convolution involving the smallest template, consisting of a single patch, becomes far less restrictive. As a consequence, a large number of local maxima are detected, especially because this patch exhibits high similarity with patches containing substantial background regions around the birds. This explains why the method detects 70 local maxima instead of the 21 birds actually present in the image.

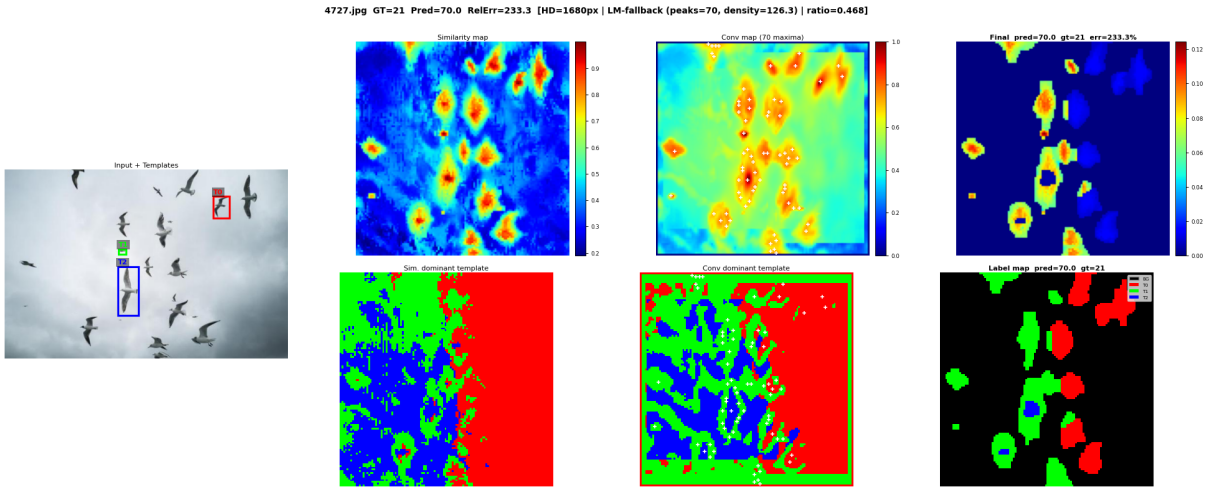


Figure 4.11: **Failure case on the FSC-147 dataset (bird example).** From left to right: the input image with exemplar templates, the similarity map with dominant template assignments, the convolution map with detected local maxima (crosses), and the final density prediction after Graph-Cut refinement.

Incorrect Templates Provided. Fig. 4.12 presents an example where unsuitable templates are provided. The objective is to count horses; however, the selected templates only contain horse heads rather than complete horse instances. The issue arises because the features extracted by DINOv2 remain highly similar for patches representing either the head or the body of a horse. As a result, the similarity map contains substantial noise since body regions are also strongly activated. If complete horse bodies are selected as templates, as shown in the example below the figure, a similar density map is obtained, but the resulting prediction becomes significantly closer to the expected count.

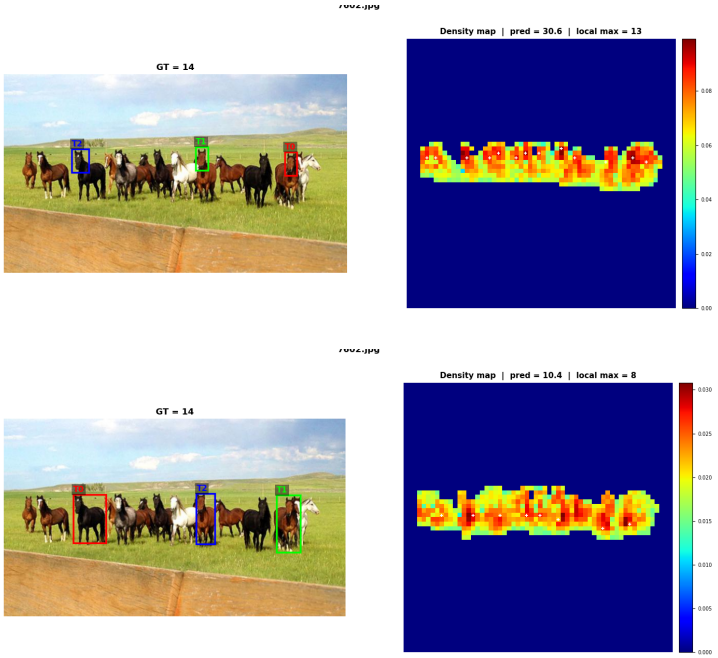


Figure 4.12: **Failure case analysis on incorrect templates provided.** The first row shows the input image with exemplar templates and ground truth (GT), followed by the predicted density map with local maxima (crosses) and the final count prediction. The second row illustrates input image with better exemplar templates (all the horse) and ground truth (GT), followed by the predicted density map with local maxima (crosses) and the better final count prediction.

High Object Density. Fig. 4.13 illustrates cases involving a very high object density together with severe occlusions. Although the resulting density maps remain qualitatively accurate, the method is only

able to provide coarse approximations and struggles to recover the exact object counts.

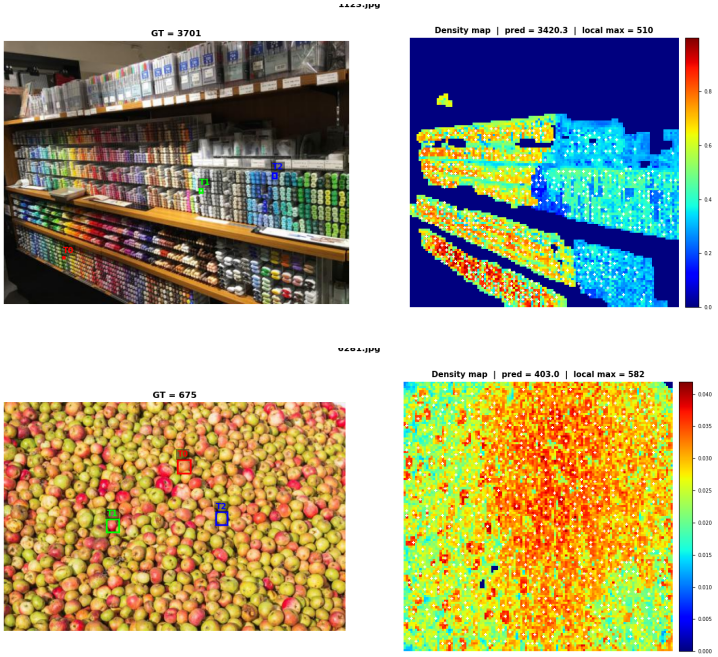


Figure 4.13: **Failure cases analysis on high object density.** On the left, the input image with exemplar templates and ground truth (GT), followed by the predicted density map with local maxima (crosses) and the final count prediction.

Incorrect Problem Definition. Finally, Fig. 4.14 highlights another failure pattern. The objective is to count the number of LEGO pieces in the image; however, the provided exemplars correspond only to individual LEGO studs rather than complete pieces. Consequently, the method estimates the number of LEGO studs instead of the number of LEGO pieces.

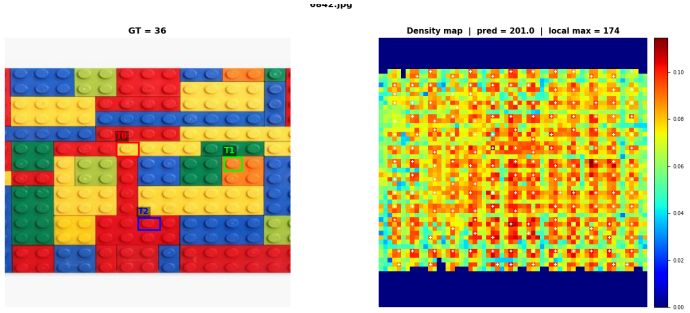


Figure 4.14: **Failure case analysis on incorrect problem definition.** On the left, the input image with exemplar templates and ground truth (GT), followed by the predicted density map with local maxima (crosses) and the final count prediction.

4.5 Ablation studies

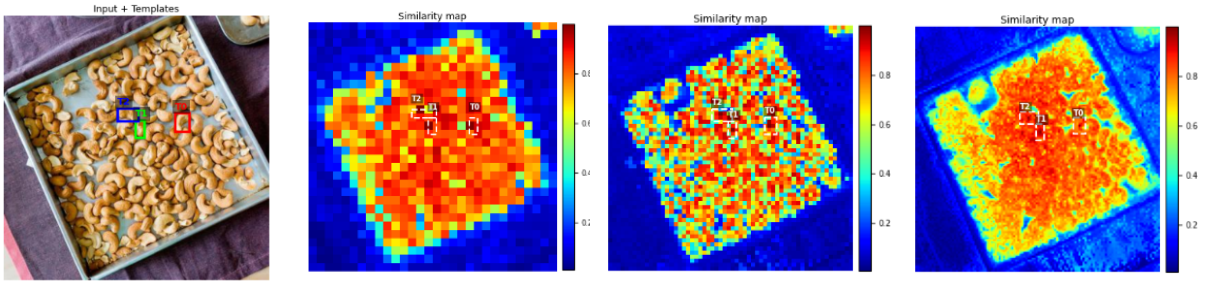
This section investigates the impact of the different design choices and modules involved in our method. We first analyze the influence of the input image resolution, followed by the contribution of the graph-cut procedure for obtaining a correctly labeled density map. The effect of the proposed LM-Fallback strategy is then examined. Since the number of provided exemplars is also a key parameter in few-shot counting, its impact is evaluated as well. In addition, we study the behavior of the method on a dataset that is more suited to our approach. Finally, an analysis of the inference time is presented.

Res	LM	GC	MAE ↓	RMSE ↓
Mix	✓	✓	**20.91**	**47.91**
	✓	✗	23.01	49.82
	✗	✓	21.90	49.25
	✗	✗	24.18	51.84
420	✓	✓	30.61	125.69
	✓	✗	31.74	126.17
	✗	✓	**26.72**	**120.59**
	✗	✗	27.67	121.08
840	✓	✓	24.21	88.81
	✓	✗	26.45	92.66
	✗	✓	**23.57**	**87.64**
	✗	✗	25.79	91.85
1680	✓	✓	**22.88**	**49.23**
	✓	✗	25.49	51.72
	✗	✓	23.87	50.54
	✗	✗	26.66	53.67

Table 4.3: Ablation study on the FSC-147 test split. **GC** = Graph-Cut, **LM** = LM-Fallback, **Res** = input resolution. Best results per resolution group are **bolded**.

4.5.1 Impact of Resolution

In our experiments on image resolution, we varied the resizing resolution applied to the input image before feeding it into the DINOv2 backbone. The baseline resolution of 840×840 pixels, originally used in CountingDINO, was adopted as the reference configuration. To study the impact of resizing, additional experiments were conducted using images resized to half this resolution (420×420) and to twice this resolution (1680×1680). Finally, a mixed-resolution configuration was evaluated, combining medium and high resolutions depending on the size of the provided templates.



(a) 420×420 (b) 840×840 (c) 1680×1680

Figure 4.15: **Effect of input resolution on similarity map quality and final counting prediction.** Starting from a template-image pair, similarity maps are computed at different input resolutions. The predicted counts are 37 (420×420), 99 (840×840), and 175 (1680×1680), compared to the ground truth (157).

The results presented in Table 4.3 clearly demonstrate the impact of image resolution on the evaluation metrics. Across all configurations, MAE decreases as the resolution increases, while the improvement is even more pronounced for RMSE. This behavior can be explained by the patch-based structure of DINOv2. The resized image is divided into 14×14 patches, each of which is mapped to a vector representation in the embedding space by the vision transformer. As illustrated in Fig. 4.18, lower resolutions produce fewer patches during image decomposition. Consequently, each patch must encode a larger amount of visual information, leading to less precise representations of fine details but stronger

global representations. Since our method heavily relies on similarity maps, accurately distinguishing regions belonging to the target object from the background is essential. Less precise feature representations therefore result in poorer similarity maps and degraded counting performance. Conversely, higher-resolution images are divided into a larger number of patches, each capturing more localized information. This enables the generation of more discriminative similarity maps, improving the distinction between object and non-object regions.

The best results are consistently obtained with the mixed-resolution configuration. After analysis, the largest differences between the pure 1680 configuration and the mixed-resolution strategy were found in high-density images, a representative example of which is shown in Fig. 4.16.

As shown in the figure, both resolutions produce relatively similar similarity maps. The main difference arises in the convolution map. Because the higher resolution results in templates spanning a larger number of patches, the matching process between the template and the feature map becomes more restrictive, leading to a reduction in the number and intensity of activation peaks within the convolution map.

This reduction in activations subsequently affects the labeling stage performed through the graph-cut procedure. Owing to the construction of the energy function, a patch is assigned to a template primarily according to its proximity to regions of strong activation. As a consequence, weaker and less numerous activation peaks lead to a less accurate density map and, ultimately, a less accurate counting prediction.

These observations indicate that high-resolution processing must be used carefully. Excessively increasing the resolution also increases the complexity of the feature map and the relationships encoded within it. An appropriate balance between global and local representations must therefore be maintained.

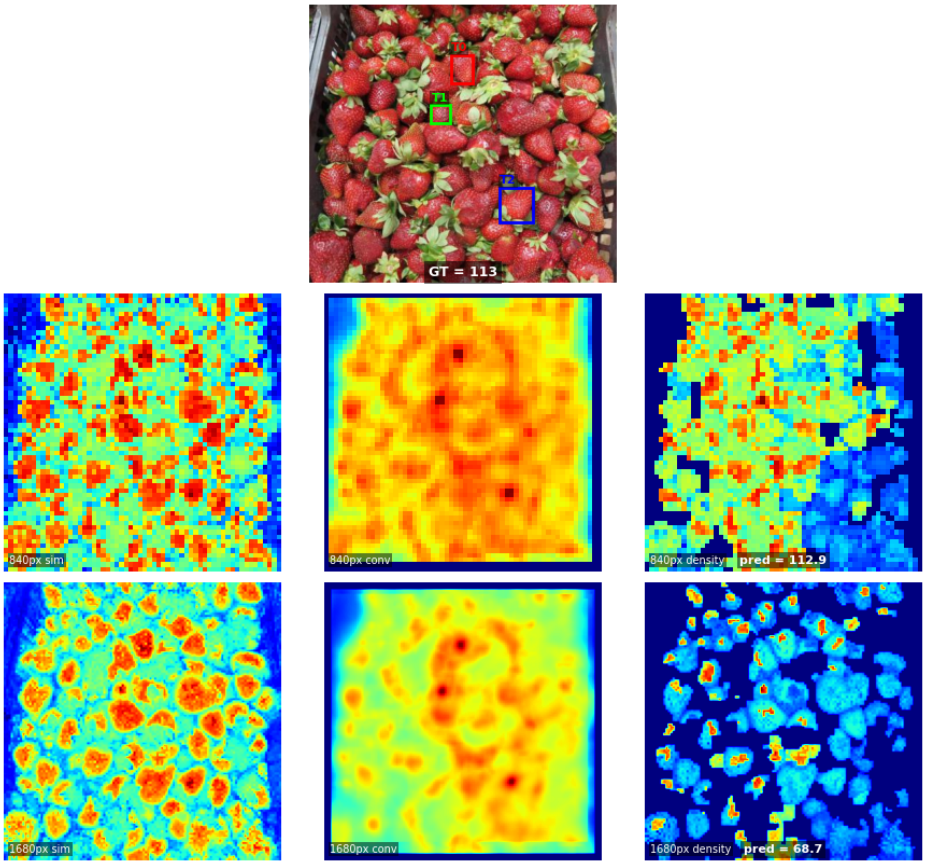


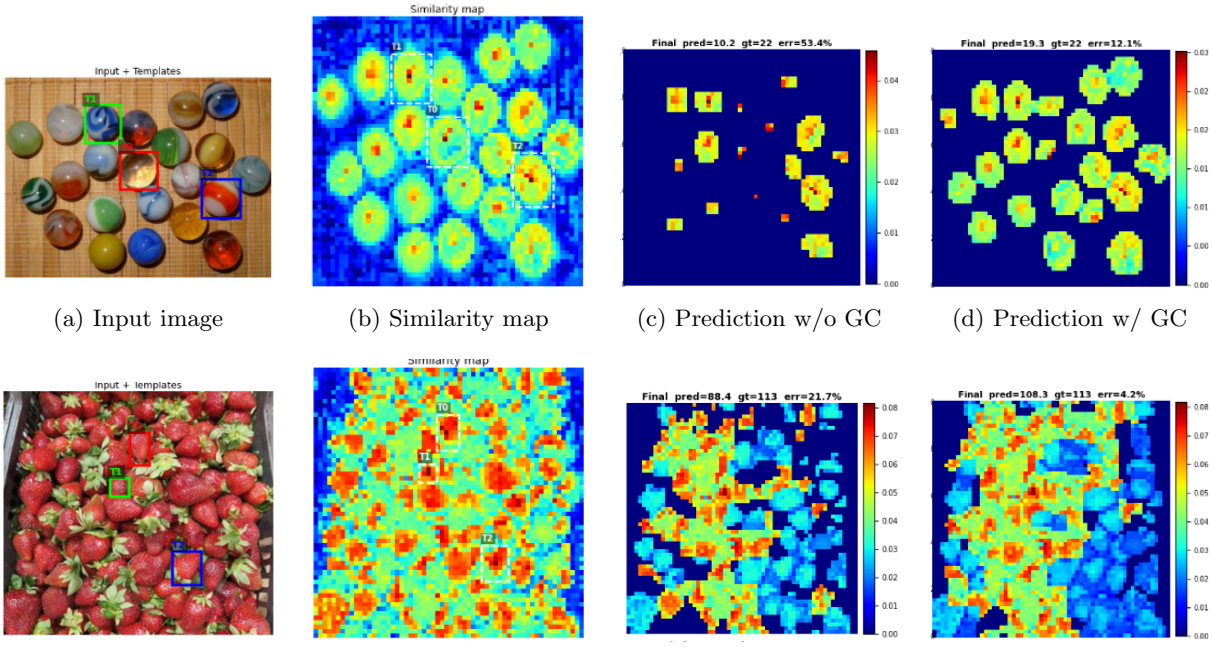
Figure 4.16: **Qualitative comparison of an image at different resolutions** Each row shows the result of the input image at different resolution, respectively $840 * 840$ and $1680 * 1680$ pixels.

This explain why very high resolution, despite appearing intuitively more informative, can become counterproductive in high-density scenes containing medium-sized objects. High resolution increases the strictness of the matching criteria, contributing to systematic error of counting.

In contrast, high resolution remains highly beneficial for small objects. At standard resolution, a small object may fit entirely within a single patch, or at most two or three patches, providing insufficient visual information for self-attention to construct a discriminative representation. At 1680, however, the same object occupies a sufficient number of patches for each one to capture meaningful and characteristic information, such as contours, textures, or local context. This provides self-attention with richer visual cues, enabling the extraction of more specific and discriminative features. The similarity map therefore benefits from improved spatial precision, with activations that are more accurate and better localized, while the extracted templates become more representative of the actual object appearance.

The 1680 resolution shifts the equilibrium toward smaller objects, which explains why the mixed-resolution configuration, by combining both resolutions, is able to effectively cover a broader range of object sizes and densities.

4.5.2 Impact of Graph-Cut Optimization



(a) Input image (b) Similarity map (c) Prediction w/o GC (d) Prediction w/ GC

(e) Input image (f) Similarity map (g) Prediction w/o GC (h) Prediction w/ GC

Figure 4.17: **Impact of graph-cut refinement on object localization.** For each example, we show (from left to right) the input image, the corresponding similarity map, the prediction obtained without graph-cut optimization, and the prediction obtained after graph-cut refinement. The graph-cut framework enforces spatial consistency between neighboring patches, leading to smoother object regions and more coherent label assignments.

By examining the different configurations without graph cut (GC) in Table 4.3, it can be observed that removing GC consistently degrades the results across all resolutions. Moreover, this degradation becomes more pronounced as the resolution increases. This behavior is expected, since higher resolutions produce more detailed and precise similarity maps, which can be more effectively exploited by the graph-cut procedure to refine the final representations.

An analysis of the examples presented in Fig. 4.17 further illustrates the impact of removing GC. In the image containing marbles, the contribution of GC to the final density map is particularly evident. Each marble receives a strong activation at its center, mainly due to the central patch capturing the light reflection. Consequently, without GC, the resulting label map covers only a limited portion of each

object, or in some cases fails to cover it entirely. With GC, the density map becomes more spatially extended and more consistent with the actual object shapes. The graph-cut procedure propagates labels into neighboring regions with lower activations but strong spatial coherence, thereby recovering areas that the original density map would otherwise neglect.

A similar phenomenon can be observed in the strawberry image. Without GC, the density map contains activation regions that are well localized around the strawberries, but the connecting regions between neighboring strawberries remain under-activated. These lower-activation areas are better handled with GC, which again extends the density map to produce a representation that is more faithful to the actual object regions.

In conclusion, the graph-cut procedure plays a significant role in the construction of the density map. First, it aims to assign labels to object regions more accurately, thereby reducing the sensitivity of the final estimation to variations in object size. Furthermore, it mitigates the limitations of the similarity map, whose construction often produces a strong activation peak at the center of an object, followed by surrounding regions exhibiting lower activations due to variations in color, shape, or partial occlusions, despite still belonging to the target object. The graph-cut formulation addresses this issue by enforcing spatial consistency in the labeling process. As a result, object regions are represented more coherently, while areas located far from activation peaks and exhibiting low similarity to the provided templates are more likely to be assigned to the background. This leads to a smoother and more reliable density map that better reflects the underlying object structure.

4.5.3 Impact of LM Fallback

Resolution	**Config GC**	**MAE without LM**	**MAE with LM**	**Gain**
Mix	✓	21.90	20.91	−0.99
Mix	✗	24.18	23.01	−1.17
1680	✓	23.87	22.88	−0.99
1680	✗	26.66	25.49	−1.17
840	✓	23.57	24.21	+0.64
840	✗	25.79	26.45	+0.66
420	✓	26.72	30.61	+3.89
420	✗	27.67	31.74	+4.07

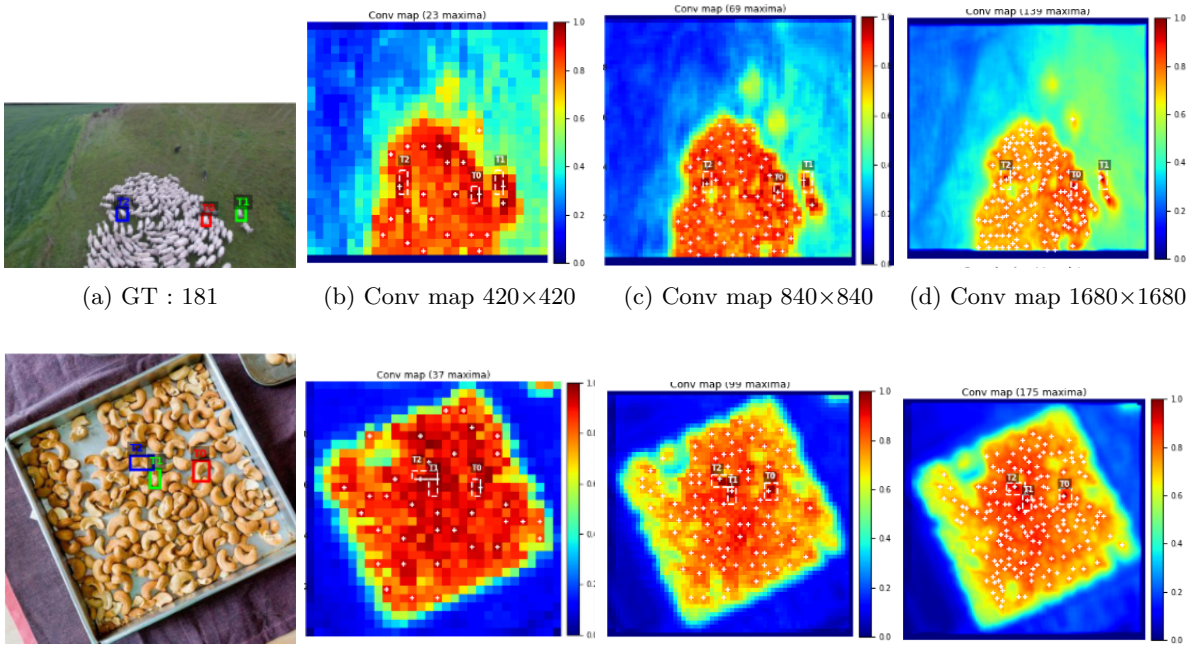
Table 4.4: Gain apporté par le LM selon la résolution

As a reminder, the LM-Fallback mechanism is activated when at least one of the provided templates is extremely small and when a large variation exists between the sizes of the exemplars. Template size is evaluated in an absolute manner by comparing the number of patches covered by each exemplar when the image is resized to the baseline resolution of 840×840, which serves as the reference resolution in our method. Consequently, the activation criterion remains identical regardless of the resizing resolution, meaning that the same images trigger the mechanism independently of the selected resolution. Once activated, the final estimation is obtained from the number of local maxima detected in the convolution map, preventing the graph-cut procedure from being dominated by a very small template that would otherwise lead to an explosion in the predicted count.

Since the set of images activating the LM-Fallback remains unchanged across resolutions, the differences observed in Table 4.4 are entirely due to the quality of the estimations produced by the method rather than to differences in the activation process itself.

We observe in Fig. 4.18 the impact of resolution on the LM-Fallback mechanism. At low resolution, due to the reduced granularity of patch representations, the regions corresponding to sheep or nuts are correctly identified but exhibit less variation in intensity, since each patch aggregates multiple parts of the objects. In contrast, at high resolution, with finer details, intensity peaks become more distinct due to improved representation granularity, enabling better discrimination between objects.

The impact of the LM-Fallback is also important for addressing a limitation of the method in the presence of an excessively small template, which is distributed over only a small number of patches and



(a) GT : 181 (b) Conv map 420×420 (c) Conv map 840×840 (d) Conv map 1680×1680

(e) GT : 157 (f) Conv map 420×420 (g) Conv map 840×840 (h) Conv map 1680×1680

Figure 4.18: **Impact of image resolution on convolution map quality.** We observe that increasing the resolution progressively improves the sharpness and localization quality of the predicted activation maps, leading to more precise object responses.

exhibits strong variations compared to the other templates, as illustrated in Fig. 4.19. Indeed, the baseline method produces a density map dominated by the blue template (the smallest one), which spans only two patches. In this case, the LM-Fallback mechanism allows the method to rely solely on the convolution map, providing a more reliable estimate of the number of objects.

The results in the table also reveal that graph cut (GC) and LM-Fallback operate independently. Regardless of the resolution, the presence or absence of GC produces relatively similar gains, confirming that the subsets of images affected by the two modules are largely disjoint.

In conclusion, the LM-Fallback mechanism relies on the assumption that each object corresponds to a distinct activation peak in the convolution map. This assumption becomes invalid in high-density scenes at low resolution, where multiple objects may merge within the same patch, leading to smoother activation maps.

4.5.4 Impact of Number of Templates

# Templates	MAE ↓	RMSE ↓
1	30.22	63.16
2	23.28	51.74
3	**20.91**	**47.91**

Table 4.5: Impact of the number of template examples on counting performance, evaluated on the FSC-147 test split with the full pipeline (LM-Fallback + Graph-Cut, mixed resolution).

The number of templates has a significant impact on the obtained results, as shown in Table 4.5. This parameter plays a central role in few-shot counting, and it is therefore expected that increasing the number of provided exemplars improves the overall performance.

It can be observed that using only a single exemplar leads to results that remain far from the best-

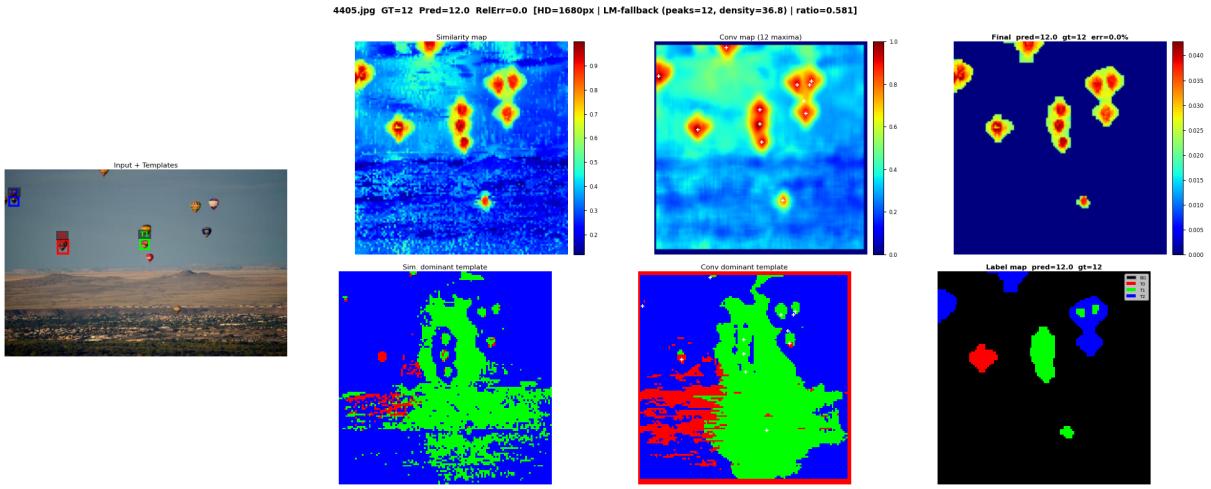


Figure 4.19: **Example of LM-Fallback activation.** The LM-Fallback method correctly predicts 12 objects, matching the 12 balloons present in the image. In contrast, the original method produces an overestimation, with a prediction of 36.8.

performing configuration, with an increase of more than 30 MAE. Several factors may explain this behavior. A very small template may assign a normalization coefficient proportional to its size to much larger objects, resulting in severe overestimation of the predictions. Conversely, a large template, associated with a lower normalization coefficient, may be applied to smaller objects, leading to underestimation of the instance count.

In scenarios where objects exhibit similar sizes, appearance variations can also become a major source of error. A single template does not provide a similarity map as informative as the one obtained from multiple exemplars contributing jointly to the representation. Similarly, the convolution between a single template and the other object instances may produce fewer activation peaks in the convolution map, resulting in a less precise graph-cut estimation.

Increasing the number of templates therefore improves the results by allowing each exemplar, regardless of the magnitude of its variations, to contribute complementary information to both the similarity and convolution maps.

4.5.5 Impact of small templates exemplars

The failure case analysis presented previously highlighted a recurring difficulty related to excessively small templates.

Table 4.6 reports the results of our method after removing images containing such small templates, which require the use of HD resolution and the LM-Fallback mechanism. A clear improvement in MAE is observed on both split of the FSC-147 dataset, decreasing from 20.91 to 15.23 on the test split and from 24.8 to 15.48 on the validation split. Beyond this improvement, the results also indicate a similar MAE across both datasets, closely matching those obtained on the FSCIndu dataset, which contains fewer images. It is important to recall that these two splits include entirely different object categories. This suggests good generalization of the method, as it produces consistent results regardless of the evaluated categories. The RMSE also decreases on both splits of the dataset, indicating a reduction in large-scale errors in the predictions.

Fig. 4.20 further supports this observation by illustrating the distribution of errors with respect to object density. A near absence of outliers is observed across all density ranges, with the MAE remaining consistently close to the MedAE. Moreover, the MNAE increases approximately linearly from 20.9% for low-density images to 32.1% for high-density images, reflecting the increasing difficulty of the task as density grows.

Overall, the method appears stable, yielding consistent results across datasets and without significant outliers when the provided templates have a reasonable size relative to the image.

	Validation		Test		FSCIndu	
Method	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓
Our method	24.80	59.27	20.91	47.90	14.64	31.13
Our method w/o HD images	**15.48**	**28.07**	**15.23**	**24.30**	**12.41**	**20.94**

Table 4.6: **Impact of excluding images requiring HD processing.** The second setting evaluates only images requiring more than 3 patches per side for an input resolution of 840×840. This corresponds to 1025 evaluated images out of 1286 for the validation split, 1028 evaluated images out of 1190 for the test split, and 551 evaluated images out of 590 on the FSCindu dataset.

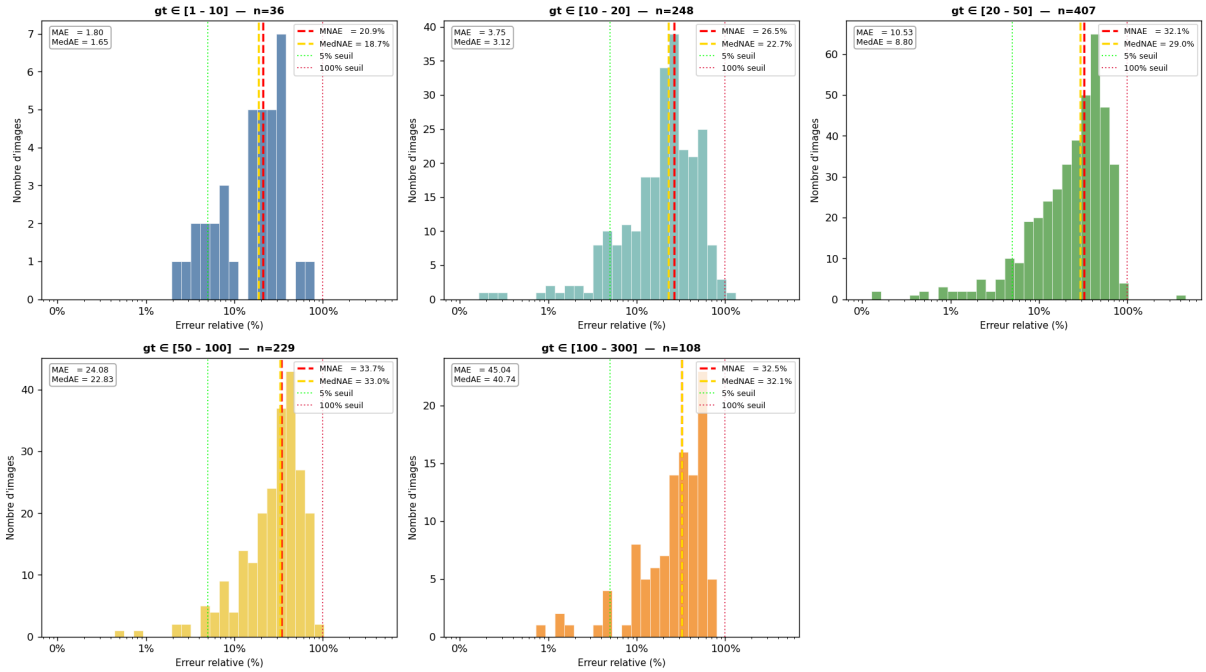


Figure 4.20: **Distribution of relative prediction errors per density object counts on the FSC-147 dataset's test split without images needing HD resolution**. Images are split by ground-truth count. Each histogram uses a log-scaled x-axis; dashed lines mark the mean (MNAE) and median (MedNAE) relative errors, with absolute MAE and MedAE reported in the inset.

4.5.6 Inference Time

Fig. 4.21 highlights significant differences in inference time across the evaluated methods. A clear variation is observed within our approach depending on the input resolution. Increasing the resolution from 840×840 to 1680×1680 leads to a substantial increase in computational cost, from 0.165 seconds to 1.66 seconds per iteration. This increase can be explained by the quadratic growth in the number of patches processed by the Vision Transformer, which directly affects the complexity of the attention mechanism and the size of the resulting feature representations.

When compared to other methods, our approach remains competitive with recent transformer-based architectures such as CountGD [1], CACViT [26], and CounTR [14], while being slower than lightweight CNN-based approaches such as LOCA [6], BMNet+ [23], and FamNet [22]. This difference is mainly due to the additional computational overhead introduced by our pipeline, which includes dense feature extraction, similarity map computation, convolution map construction, and graph-cut optimization.

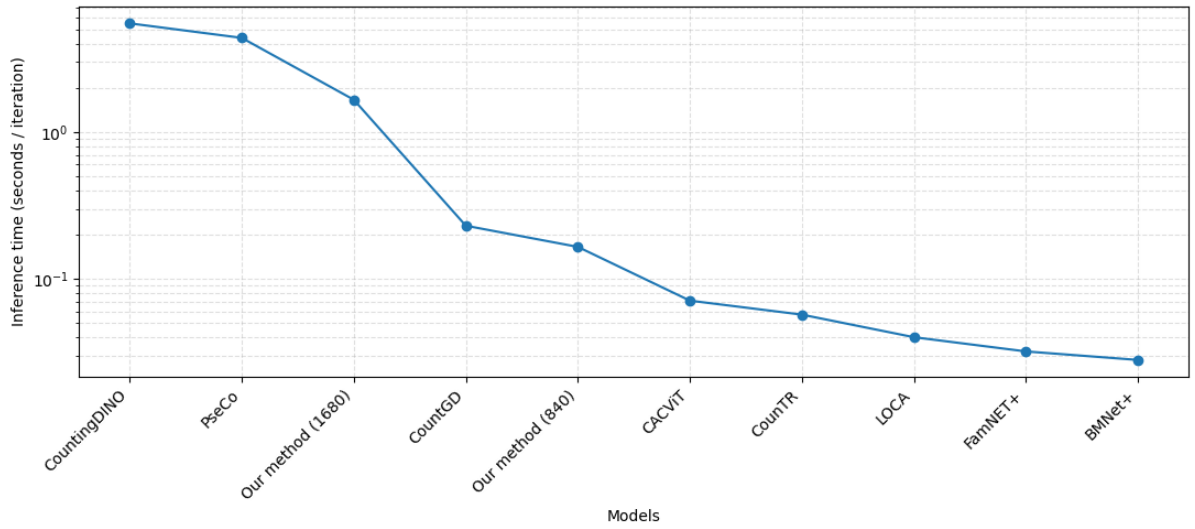


Figure 4.21: **Inference time comparison across different object counting methods. Source : [2]**

Finally, compared to CountingDINO, our method is significantly more efficient. This improvement is mainly explained by the fact that CountingDINO relies on a grid-based strategy where the image is divided into 16 sub-regions, each independently resized and processed through the backbone. This leads to multiple forward passes and a substantial increase in inference time. In contrast, our approach preserves a global representation of the image and avoids redundant computations, resulting in a more efficient inference pipeline.

Chapter 5

Discussion

This section provides a discussion of several aspects of the proposed method. First, although our approach is inspired by CountingDINO [18] and achieves improved performance compared to that method, it is important to analyze the actual differences between the two approaches and to identify the specific contributions introduced by our framework. We then discuss the main findings and key takeaways of this work, highlighting the contributions brought by the proposed method. Finally, we examine its current limitations and outline potential directions for future improvements.

5.1 Methodological Improvements over CountingDINO

Beyond a simple improvement in quantitative results across different datasets, our method addresses certain inconsistencies present in CountingDINO, while also introducing an alternative way of leveraging the features produced by the Vision Transformer.

5.1.1 Resolution Handling

CountingDINO also had to address the challenge posed by images containing very small templates. By default, the method resizes each image to a resolution of 840×840. However, because images with extremely small templates remained problematic, an alternative strategy was introduced.

Specifically, the image is divided into a 4×4 grid, resulting in 16 equally sized regions. Each region is then independently resized to 840×840 and processed by the Vision Transformer. This procedure produces 16 feature maps, which are subsequently concatenated to reconstruct a single large feature map.

This strategy, however, introduces an important limitation. Since each region is processed independently, the self-attention mechanism operates only within that region and can no longer capture interactions between patches located in different parts of the image. As a consequence, long-range relationships across the image are weakened. Fig. 5.1 illustrates this issue through two cosine similarity maps computed between a selected patch and the remainder of the image. The first map is obtained from the original image, whereas the second corresponds to the image after the splitting and reconstruction procedure used by CountingDINO. The resulting patch similarities are no longer as consistent as in the original representation, as can be observed in the lower part of the image where the penguins are no longer correctly identified. Moreover, a grid-like pattern can be observed in the second map, which may, in some cases, also propagate into the final output of CountingDINO. Furthermore, this division strategy is substantially more expensive in terms of inference time, as it requires repeated resizing operations, multiple feature extraction passes, and the reconstruction of the final feature map.

To address this limitation, our method adopts a different strategy. When a template occupies only a few patches in the image, the entire image is resized to 1680×1680, providing a finer representation of the scene and more detailed representations of the templates.

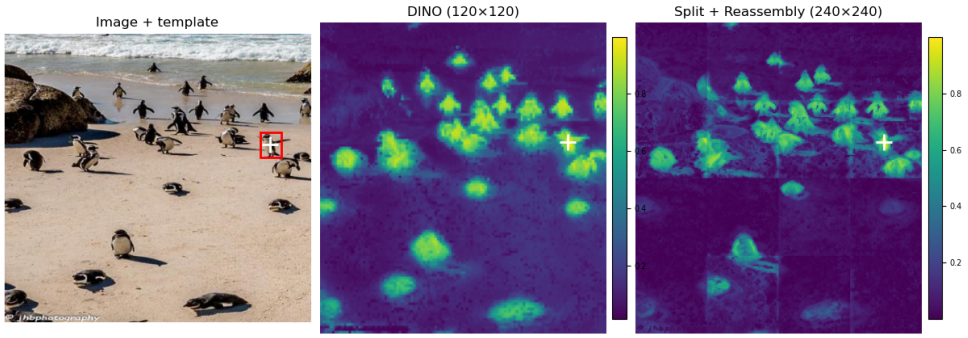


Figure 5.1: **Comparison of cosine similarity maps computed with a ViT-based encoder using a single forward pass versus a split-and-reassemble strategy.** The left figure shows the input image with the selected template and the location of the central patch used as query. The middle one presents the similarity map obtained from the original feature map with a resolution of $1680 * 1680$ corresponding to a $120 * 120$ feature map. The right figure corresponds to the similarity map obtained after dividing the image into 16 regions, processing each region independently through the ViT, and concatenating the resulting feature maps with a shape of $240 * 240$.

5.1.2 Convolution Map Construction

Another difference lies in the construction of the convolution map. In CountingDINO, the convolution map is computed for each template and then resized to a common resolution space through interpolation prior to aggregation. While this procedure ensures that all maps are aligned to the same dimensional space, it may also introduce distortions in the information, potentially shifting or adding noise in regions of high activation. This can affect both the precision and the localization of high-intensity regions, particularly when there are significant differences in template sizes.

As illustrated in Fig. 5.2, in the apple example, the three convolution maps are shown together with the aggregated map obtained either through interpolation or zero-padding. In contrast, our method addresses this inconsistency by applying zero-padding to resize all maps, thereby preserving the precision and coherence of the resulting map.

5.1.3 Scale-Aware Template Normalization

CountingDINO aims to exploit prior information available at inference time. Since each region defined by a bounding box corresponds to an object instance, the method computes the sum of the density values within each annotated region and uses the average of these sums as a normalization coefficient.

However, a limitation arises when significant differences exist between template sizes. In such cases, the sums computed within the different templates can vary substantially, leading to estimation errors. This phenomenon is illustrated in Fig. 5.3 with the seemingly simple task of counting keyboard keys. Both our method and CountingDINO produce a similar density map; the difference lies in the weights assigned to each patch. Examining the sums of the density values within the three templates yields values of 10.23, 13.54, and 21.04 for the largest template. Following the CountingDINO strategy, the normalization coefficient is computed as the inverse of their average, i.e., 1/14.66, which results in a significant underestimation of the true count.

This behavior is explained by the large number of keys whose size is close to that of the smallest template. Assuming that keys of similar size should contribute approximately the same value as the smallest template (10.23), normalizing the density map using a coefficient derived from 14.66 leads to an underestimation of these instances by approximately 69%.

In contrast, our method leverages the graph-cut procedure to assign each key the normalization coefficient associated with the template that is most similar in size. Consequently, keys corresponding to the smaller template receive a lower normalization coefficient, while larger keys are associated with

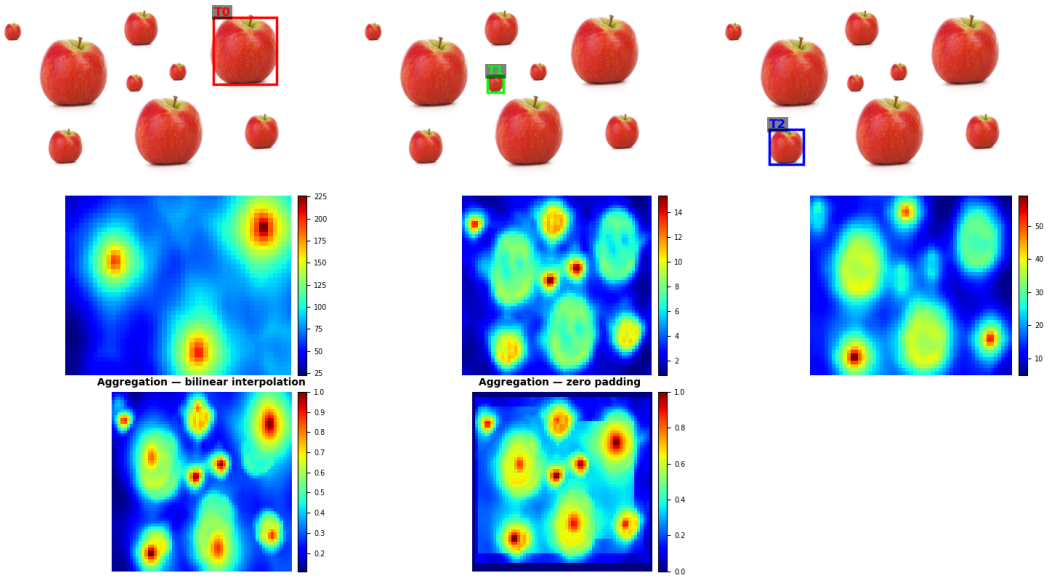


Figure 5.2: **Visualization of template-wise convolution maps and their aggregation strategies.** The first row shows the original input image with the three selected templates highlighted by bounding boxes. The second row displays the corresponding convolution maps computed independently for each template, illustrating their respective spatial response patterns. The third row presents the final aggregated convolution map obtained using two different strategies: bilinear interpolation as in CountingDINO, and zero-padding as proposed in our method. This comparison highlights the impact of the aggregation strategy on spatial alignment and the preservation of high-activation regions.

higher coefficients, preventing both underestimation and overestimation across different object scales. This results in a prediction of 63 objects, substantially closer to the ground-truth count of 70 than the prediction of 32 obtained by CountingDINO.

5.1.4 Interpretability and Error Analysis

Another advantage of the proposed method lies in the interpretability of its output. While CountingDINO produces a density map obtained through the aggregation of multiple convolution maps, the resulting representation can be difficult to interpret visually. High-density regions often appear diffuse, making it challenging to determine which areas effectively contribute to the final count and to identify the origin of potential counting errors.

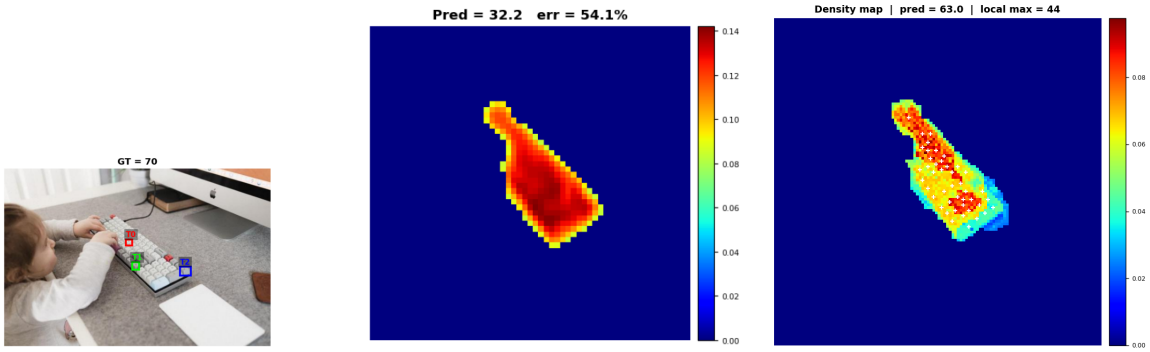
In contrast, the density map produced by our method is directly derived from similarity-based assignments and graph-cut optimization. As a result, object activations remain spatially coherent and can be more easily associated with individual object instances. This makes the final prediction more transparent and facilitates visual inspection by a human operator.

Fig. 5.4 illustrates this difference. While the output of CountingDINO contains broad activation regions whose contribution to the final count is difficult to assess, our density map exhibits more localized and interpretable responses. Consequently, false positives, missed detections, or activation leakage can be identified more rapidly, simplifying error analysis and debugging.

5.2 Contributions & Takeaways

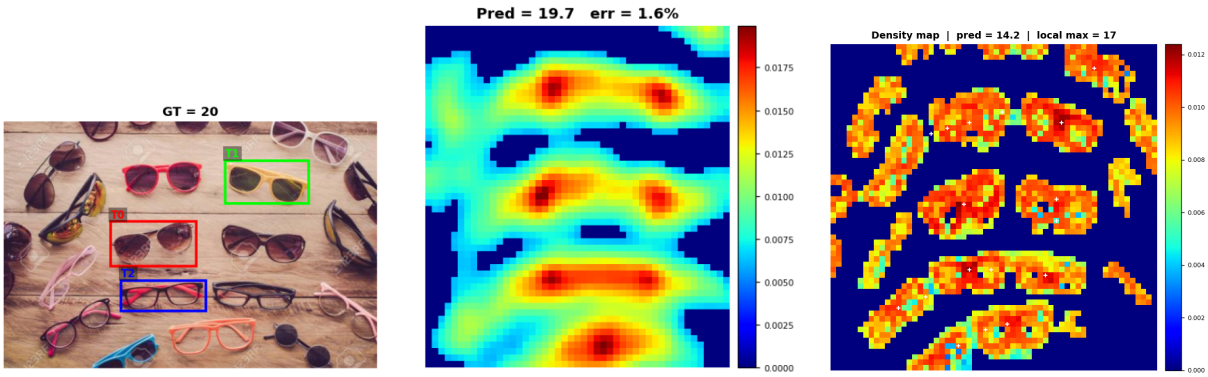
If one had to retain the key contributions and insights of this work, they can be summarized as follows.

First, we highlight the strong representational capacity of self-supervised vision features. In particular, features extracted from models such as DINOv2 [17] encode rich semantic structures that can be leveraged without any task-specific supervision. This is important because it demonstrates that strong priors for object localization already exist in foundation models, which significantly reduces the need for



(a) Input image + templates (b) CountingDINO result (c) Our method result

Figure 5.3: **Comparison of final prediction outputs by our method and CountingDINO with templates of different sizes.** The figure shows the input image with selected templates, the output obtained using CountingDINO, and the result produced by the proposed method. The aggregated template responses (sum of template weights) are respectively 10.23, 13.54 and 21.04.



(a) Input image (b) CountingDINO density map (c) Our method density map

Figure 5.4: **Comparison of the final density maps produced by CountingDINO and our method.** While CountingDINO generates broad and diffuse activation regions, our method produces more localized and spatially coherent responses. This results in a density map that is easier to interpret visually and facilitates the identification of false positives, missed detections, and other failure cases during error analysis.

costly annotated datasets and enables more scalable class-agnostic counting approaches.

Second, we demonstrate the importance of handling scale and size variability in object counting. To this end, we leverage a graph-cut based optimization framework, which enforces spatial consistency and refines label assignments. This is crucial because real-world scenes contain objects with highly variable sizes, and ignoring this variability leads to unstable predictions. Our formulation therefore improves robustness and makes the counting process more reliable across diverse visual conditions.

Finally, we emphasize the robustness of the proposed approach on well-chosen image settings, where it exhibits stable and consistent performance. This is particularly relevant because it shows that, beyond average performance gains, the method behaves predictably in favorable configurations, an important property for real-world deployment where reliability and consistency are often as critical as accuracy.

5.3 Limitations & Future Work

Regarding the limitations of the proposed method, we observe that the main failure cases occur in images containing excessively small templates. This issue can be partially mitigated by the LM-Fallback mechanism; however, it cannot be fully addressed within the original formulation of the method due to its inherent design constraints.

We have observed, however, that on images with well-sized templates, the errors become significantly lower. In this context, our model could be used in industrial settings as a supporting tool, with a visual verification of the consistency of the feature map and/or the LM-Fallback when it is activated.

Future improvements could focus on several directions. First, a better tuning of hyperparameters may further improve performance, such as the threshold used for local maxima detection or the energy function in the graph-cut optimization, which could be refined to enhance segmentation quality. Second, template augmentation strategies could be explored, inspired by approaches such as RCAC [8], where templates are resized or slightly perturbed to improve robustness. Such augmentations could help strengthen object activation in similarity maps and increase the reliability of detected regions.

Chapter 6

Conclusion

The objective of this work was to investigate the extent to which visual representations extracted from a Vision Transformer can be exploited for class-agnostic object counting. More specifically, the goal was to design a robust counting method capable of handling object variations, scale changes, and occlusions without relying on annotated datasets or any task-specific training procedure.

To address this objective, we first reviewed the existing literature on class-agnostic object counting and analyzed the main factors that differentiate current approaches, including their architectures, training strategies, and feature extraction mechanisms. This analysis shows an interest in methods that reduce or eliminate the dependence on supervised training and large annotated datasets, culminating in recent training-free approaches such as CountingDINO [18], which leverage self-supervised visual representations for object counting.

Building upon these observations, and particularly on the capabilities of DINOv2 [17], we proposed a fully training-free object counting pipeline based on a self-supervised vision-only backbone. The proposed method exploits the feature representations of both the input image and the provided exemplars to construct similarity and convolution maps, each capturing complementary information about object presence and appearance. To effectively exploit these representations while remaining robust to object scale variations, a graph-cut-based spatial refinement strategy was introduced for density map construction. Furthermore, to address failure cases caused by extremely small templates, an LM-Fallback mechanism was proposed, relying on the detection of local maxima within the convolution map to estimate the final object count.

Experimental evaluations conducted on the FSC-147 [22] and FSCIndu [2] datasets demonstrated that the proposed approach achieves competitive performance with respect to several fully supervised methods while outperforming the previous state-of-the-art unsupervised and training-free approach. In addition to these quantitative results, extensive ablation studies highlighted the contribution of the different components of the method, including graph-cut refinement, the LM-Fallback mechanism, and the image resolution strategy. A detailed analysis of both successful and failure cases further provided insights into the strengths and limitations of the proposed framework.

Additional experiments performed on subsets excluding images containing extremely small templates revealed a stable behavior across object categories and datasets, suggesting a strong capacity for generalization. These results indicate that the proposed method could serve as a practical support tool in industrial scenarios, where human validation may be incorporated when particularly challenging examples are encountered.

Overall, this work demonstrates that competitive class-agnostic object counting can be achieved solely from self-supervised visual representations, without any task-specific training or density map prediction network. These findings further highlight the potential of self-supervised Vision Transformers as a foundation for future training-free object counting methods.

Bibliography

[1] Niki Amini-Naieni, Tengda Han, and Andrew Zisserman. "Countgd: Multi-modal open-world counting". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 48810–48837.

[2] Nicolas Bourez. "Agnostic Object Counting Using Deep Learning for Manufacturing Applications". Supervisor: Christophe De Vleeschouwer; Readers: Christophe De Vleeschouwer, Sébastien Jodogne, Antoine Lejeune. Master's thesis. Louvain-la-Neuve, Belgium: École polytechnique de Louvain, 2025.

[3] Yuri Boykov, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts". In: *IEEE Transactions on pattern analysis and machine intelligence* 23.11 (2001), pp. 1222–1239.

[4] Luca Ciampi et al. "A survey on class-agnostic counting: Advancements from reference-based to open-world text-guided approaches". In: *Computer Vision and Image Understanding* (2026), p. 104703.

[5] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[6] Nikola Djukic et al. "A low-shot object counting network with iterative prototype adaptation". In: *arXiv preprint arXiv:2211.08217* (2022).

[7] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[8] Shenjian Gong et al. "Class-agnostic object counting robust to intraclass diversity". In: *European Conference on Computer Vision*. Springer. 2022, pp. 388–403.

[9] Zhizhong Huang et al. "Point segment and count: A generalized framework for object counting". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 17067–17076.

[10] Ruixiang Jiang, Lingbo Liu, and Changwen Chen. "Clip-count: Towards text-guided zero-shot object counting". In: *Proceedings of the 31st ACM International Conference on Multimedia*. 2023, pp. 4535–4545.

[11] Alexander Kirillov et al. "Segment anything". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 4015–4026.

[12] Lukas Knobel, Tengda Han, and Yuki M Asano. "Learning to count without annotations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 22924–22934.

[13] Wei Lin and Antoni B Chan. "A fixed-point approach to unified prompt-based counting". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 4. 2024, pp. 3468–3476.

[14] Chang Liu et al. "Countr: Transformer-based generalised visual counting". In: *arXiv preprint arXiv:2208.13721* (2022).

[15] Erika Lu, Weidi Xie, and Andrew Zisserman. "Class-Agnostic Counting". In: *CoRR* abs/1811.00472 (2018). arXiv: 1811.00472. URL: http://arxiv.org/abs/1811.00472.

[16] Anindya Mondal et al. "Omnicount: Multi-label object counting with semantic-geometric priors". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 39. 18. 2025, pp. 19537–19545.

[17] Maxime Oquab et al. "Dinov2: Learning robust visual features without supervision". In: *arXiv preprint arXiv:2304.07193* (2023).

[18] Giacomo Pacini et al. “Countingdino: A training-free pipeline for class-agnostic counting using unsupervised backbones”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2026, pp. 806–815.

[19] M Pechaud. “Introduction aux graphcuts en vision par ordinateur”. In: *ENS Paris* 212 (2007).

[20] Jer Pelhan, Vitjan Zavrtanik, Matej Kristan, et al. “Dave-a detect-and-verify paradigm for low-shot counting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 23293–23302.

[21] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *International conference on machine learning*. PmLR. 2021, pp. 8748–8763.

[22] Viresh Ranjan et al. “Learning To Count Everything”. In: *CoRR* abs/2104.08391 (2021). arXiv: 2104.08391. URL: https://arxiv.org/abs/2104.08391.

[23] Min Shi et al. “Represent, compare, and learn: A similarity-aware framework for class-agnostic counting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9529–9538.

[24] Oriane Siméoni et al. “Dinov3”. In: *arXiv preprint arXiv:2508.10104* (2025).

[25] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[26] Zhicheng Wang et al. “Vision transformer off-the-shelf: A surprising baseline for few-shot class-agnostic counting”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 6. 2024, pp. 5832–5840.

[27] Huilin Zhu et al. “Zero-shot object counting with good exemplars”. In: *European conference on computer vision*. Springer. 2024, pp. 368–385.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**