UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Few-Shot Object Counting Using External Visual Prompts

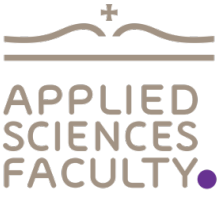
Author:
Anton BRAZHNYI

Supervisor:
Kostiantyn BOKHAN

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences



Lviv 2024

Declaration of Authorship

I, Anton BRAZHNYI, declare that this thesis titled, "Few-Shot Object Counting Using External Visual Prompts" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

Few-Shot Object Counting Using External Visual Prompts

by Anton BRAZHNYI

Abstract

Object counting is the task of estimating the number of specific objects present in an image. Similarly to other computer vision tasks, traditional object counting methods typically require a large training dataset and are not suited for counting novel classes. Class-agnostic object counting, which is generally divided into few-shot and zero-shot approaches, aims to count arbitrary object categories. Few-shot counting requires manually labeled image patches depicting the object of interest, which is impractical in real-world applications. Zero-shot counting is primarily focused on using text prompts to specify the object without relying on manual annotations. However, text descriptions can be ambiguous and may not precisely convey object characteristics such as shape, texture, or size. Visual exemplars such as image patches act as a more direct reference, which leads to better generalizability and accuracy. In this work, we plan to explore the possibility of counting arbitrary objects in a few-shot manner without having humans in the loop. In particular, we are interested in utilizing a set of support images, which can be prepared in advance for a given object category and later used for all the query images. This would allow to accurately count specific objects without the need for extensive annotation.

Contents

List of Figures

List of Tables

List of Abbreviations

SOTA	**S**tate-**O**f-**T**he-**A**rt
CNN	**C**onvolutional **N**eural **N**etwork
ViT	**V**ision **T**ransformer
MAE	**M**ean **A**bsolute **E**rror
RMSE	**R**oot **M**ean **S**quared **E**rror
ECA	**E**fficient **C**hannel **A**ttention
SE	**S**queeze-and-**E**xcitation
RGB	**R**ed, **G**reen, **B**lue

Chapter 1

Introduction

1.1 Background and Motivation

Object counting is a fundamental task in computer vision with diverse applications ranging from crowd monitoring and traffic analysis to biological cell counting. It is a challenging problem that requires reasoning about the number of object instances that are present in an image while also addressing object scale and appearance variations. Despite being more niche than other computer vision tasks such as object detection or segmentation, object counting has seen a rise in research in recent years.

Traditional object counting methods are typically class-specific, requiring extensive labeled datasets for each target category. This approach is not scalable to a large number of object categories due to the high cost and time involved in data annotation. Recent advancements have introduced class-agnostic object counting methods, which aim to count arbitrary objects at test time, significantly reducing the need for extensive labeled datasets. However, these methods primarily utilize patch exemplars or text prompts to specify the target objects, facing several challenges. Patch-based methods rely on annotated bounding boxes for exemplars, which are often impractical to obtain, while text-guided methods may suffer from ambiguities in visual-textual alignment.

The goal of this work is to address these limitations by using the "external visual prompts" for class-agnostic object counting. External visual prompts are images or visual elements that provide information about the object of interest but are not part of the query image. This approach allows the model to count objects from new categories with minimal manual annotation while preserving high accuracy. By utilizing external visual prompts, we aim to enhance the scalability and applicability of object counting methods, making them suitable for a wider range of scenarios. Our primary motivation is to tackle the practical challenges of deploying object counting methods in diverse applications.

1.2 Thesis Structure

This thesis is structured as follows:

- Chapter 1 outlines the motivation and the goal of this work.
- Chapter 2 reviews existing class-specific and class-agnostic object counting methods, including weakly supervised, few-shot, and zero-shot approaches.
- Chapter 3 introduces the benchmark datasets used in object counting, highlights issues with the class-agnostic datasets, and presents a refined dataset. It also describes the metrics used to evaluate model performance.

- Chapter 4 discusses the limitations of state-of-the-art methods and presents the proposed solution, including a novel model architecture and training procedure. It provides extensive evaluation results.
- Chapter 5 details the experimental setup, including backbone selection and an ablation study to evaluate the impact of different architectural choices.
- Chapter 6 summarizes our findings, discusses the limitations of the current approach, and outlines directions for future research.

Chapter 2

Related Work

2.1 Class-Specific Object Counting

Class-specific methods focus on predefined categories such as humans, animals, cells, or cars, which means they are limited to specific classes and require additional data annotation for new object categories. Initial approaches were based on detection, where the count is just the number of detected object instances. Despite being a straightforward method, counting by detection requires the model to learn a significant amount of possibly redundant information, including the precise location of object instances, which can be challenging when dealing with heavily occluded objects. Counting by regression, on the other hand, generally performs well in the presence of occlusions. Chan and Vasconcelos, 2009 proposed an effective way to regress high-dimensional low-level features to the count values. This approach is called "glancing" and is further explored by Chattopadhyay et al., 2017. The authors showed that directly predicting image level counts from the CNN representations outperforms the detection-based methods. While being easy to train and use, "glancing" is efficient only if the object count is small. To tackle this problem, Chattopadhyay et al., 2017 resort to "subitizing", a psychological term that means the ability to instantly recognize the number of objects in a small group (typically 1-4) without counting them one by one. Inspired by this concept, they proposed to divide the image into non-overlapping cells, use "glancing" in each cell, and use addition to get the total object count. Other regression-based methods mainly rely on generating the density map, which is then used to obtain the object count, usually by summing up the pixel values. In order to convert point-level annotations into a density map, these methods make use of a Gaussian kernel. This begs the question: how to choose the best kernel size, or is the Gaussian kernel even the optimal method to obtain a density map from point annotations? Extensive analysis of the impact of different density maps on the counting accuracy was conducted by Wan and Chan, 2019. The authors also proposed to learn a density map representation via an adaptive density map generator.

2.2 Weak-Supervised Object Counting

All the above methods require massive datasets with thousands or millions of annotated object instances to be trained on. Collecting an enormous amount of training data is not always practical or even possible. Weak-supervised methods, a subset of class-specific counting, are focused on reducing the required level of supervision. Laradji et al., 2018 presented a new architecture and a loss function to perform object counting and localization with point-level annotations only. Yang et al., 2020 introduced a method that does not rely on location supervision for training. The network

is trained to count by exploiting the relationship among the input images, specifically sorting them by their count values. Cholakkal et al., 2022 proposed a novel method for density map estimation with image-level supervision, which counts the number of instances within or beyond the subitizing range and does not require location information.

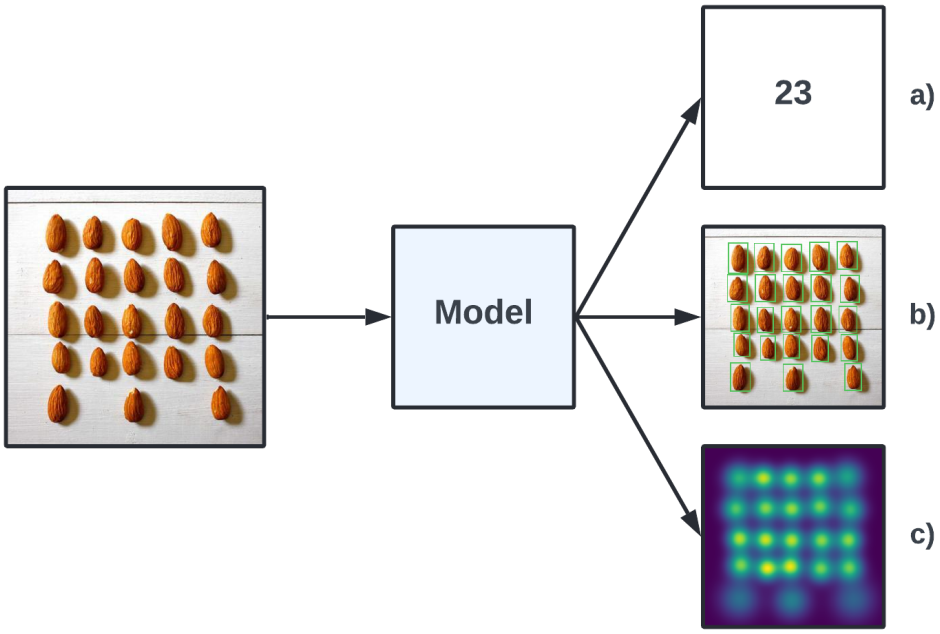


FIGURE 2.1: Different targets in object counting task. a) Glancing: directly regressing the number of objects. b) Detection-based: object count corresponds to the number of the detected instances. c) Density map prediction: object count is obtained by summing up the pixel values.

2.3 Few-Shot Object Counting

2.3.1 Patch-Based Methods

Few-shot object counting methods aim to count arbitrary objects given a few exemplar patches as inference-time guidance. The pioneering work by Lu, Xie, and Zisserman, 2018 reformulated the counting problem as the matching one and proposed a Generic Matching Network (GMN) architecture that can count in a class-agnostic manner simply by specifying a bounding box containing the object of interest. GMN features an explicit adapter module, which customizes the model to the target domain. This adaptation procedure, however, still requires hundreds of labeled examples. Ranjan et al., 2021 adopted a similar correlation matching approach and presented a novel model, FamNet, along with a test-time adaptation scheme that requires only a few bounding boxes around the object of interest. This adaptation scheme tunes the model to the provided exemplars with a few gradient descent updates.

Subsequent advancements can be categorized into two streams. The first one focuses on leveraging advanced visual architectures like vision transformers to improve feature representation. Liu et al., 2022 introduced a Counting Transformer (CounTR), a novel transformer-based architecture for class-agnostic object counting, which explicitly captures the similarity between image patches using the attention mechanism. Đukić et al., 2023 used a transformer to fuse the exemplar shape and appearance information with image features. Lin, Hong, and Wang, 2021 proposed LaoNet, an effective transformer-based network named for one-shot object counting, which achieves results comparable with few-shot methods while learning with a high convergence speed.

The second stream aims to enhance the exemplar matching process by explicitly modeling exemplar-image similarity or by further exploiting exemplar guidance. Shi et al., 2022 argued that a fixed inner product, which is used to compare exemplars with query features, may be insufficient in modeling class-agnostic similarity and introduced a learnable dynamic similarity metric. You et al., 2022 proposed a similarity-aware feature enhancement block, which encourages the model to inspect the query image by focusing more on the regions akin to the exemplars, leading to much clearer boundaries between different objects. Lin et al., 2022 designed a Scale-Prior Deformable Convolution Network (SPDCN) to extract features of objects with specific size and thus take advantage of scale information. The scale information is embedded into the deformable convolution so that its receptive field is adjusted automatically, and the extracted features correspond to the scale of the given exemplars. This design significantly increases the counting accuracy because objects of the same class typically have similar scale in an image.

A high-level architecture of a typical patch-based counting method is shown in Figure 2.2.

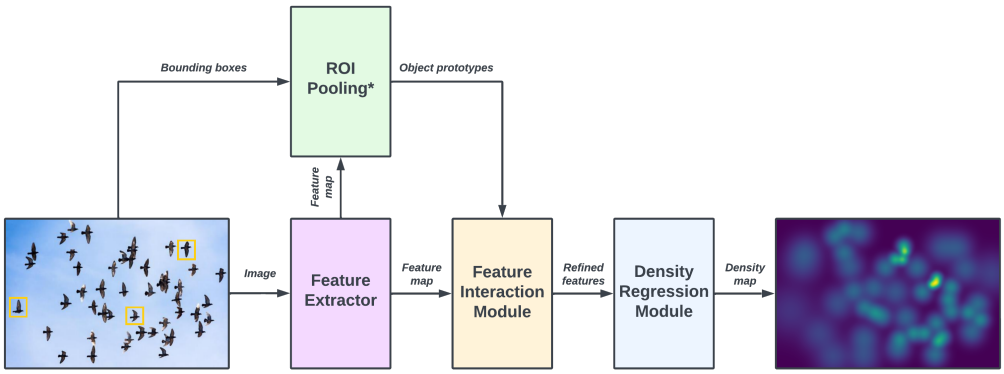


FIGURE 2.2: High-level architecture of few-shot patch-based methods. The query image with bounding boxes around the object of interest is passed as input. These bounding boxes, along with the image features, are passed to the ROI pooling (or some other similar layer) to extract object prototypes. The prototypes are then matched with the image features, typically using a correlation or attention mechanism. The resulting features are used to regress the density map. The count is obtained by summing up the pixel values in the density map.

2.3.2 Support-Based Methods

While the above methods require additional patch-level annotation depicting the object of interest as inputs, Yang et al., 2021 explored a different strategy. They presented a model, Class-agnostic Fewshot Object Counting Network (CFOCNet), that can count arbitrary objects provided a query image and a support set of object exemplars that are not a part of the query image. These exemplars are referred to as external. It is important to note that some patch-based methods, such as CounTR (Liu et al., 2022), have the capability to incorporate external exemplars into their input, even though they were not initially designed to do so. Jiang et al., 2023 proposed a similar strategy termed cross-image counting, which allows the combination of different reference and target images. Specifically, annotated exemplars from one image can be used to count objects of the same class in other images. Sokhandan et al., 2020 presented a class-agnostic counting architecture, where a reference pair consisting of an image and its corresponding target density map is used to provide information about the object of interest.

Although these methods are usually not distinguished as a separate category from other few-shot methods, we will classify them as support-based. Another way to think about this is to consider the origin of the information provided by the exemplars. Patch-based methods utilize internal exemplars, meaning that they are derived from within the query image itself or are annotated therein. Conversely, support-based methods employ external exemplars, which are independent of the query image. This distinction allows the model to count objects in new images without the need for direct annotation within those images.

2.4 Zero-Shot Object Counting

2.4.1 Reference-Less Methods

Reference-less counting has recently gained attention as a promising approach for class-agnostic counting without human annotation. Ranjan and Nguyen, 2023 proposed an exemplar-free counting approach, which works by identifying exemplars from the most frequent objects via a Repetitive Region Proposal Network (RepRPN). The work by Hobley et al. Hobley and Prisacariu, 2022 expanded the idea of exemplar-free counting and demonstrated that regression from vision transformer features without point-level supervision or reference images is competitive with methods that use reference images. Although reference-less methods do not require exemplars at test time, these methods simply count objects that belong to the category with the highest number of instances present in the image. As a result, they are not suitable for counting a specific class of interest and can be used only for images with a single predominant object class.

2.4.2 Text-Guided Methods

Recent advances in object counting are focused on utilizing multimodal models to use text prompts for specifying the object of interest. In particular, Xu et al., 2023 introduced the task of text-guided zero-shot object counting, where only the class name is needed in inference time. They proposed a two-stage method, where they first generate exemplar prototypes by using a text-conditioned variational autoencoder. Following that, these exemplar prototypes are passed into a regular few-shot object counter trained with exemplar supervision. Extending this idea Jiang, Liu,

and Chen, 2023 presented CLIP-Count, an end-to-end pipeline that employs text guidance to estimate density maps for objects in the open vocabulary in a zero-shot manner. By aligning text embedding with patch-level visual features, CLIP-Count fully utilizes the pretrained knowledge in CLIP (Radford et al., 2021). Recently, Kang et al., 2023 presented a way to substantially reduce the number of trainable parameters and thus the memory cost of using visual-language models for object counting task. Amini-Naieni et al., 2023 proposed CounTX, an open-world object counting model that accepts an image and an arbitrary object class description and directly uses these inputs to predict the object count. CounTX eliminates the need for an exemplar-based counting model and also accepts a more detailed specification of the target object to count rather than simply using a class name.

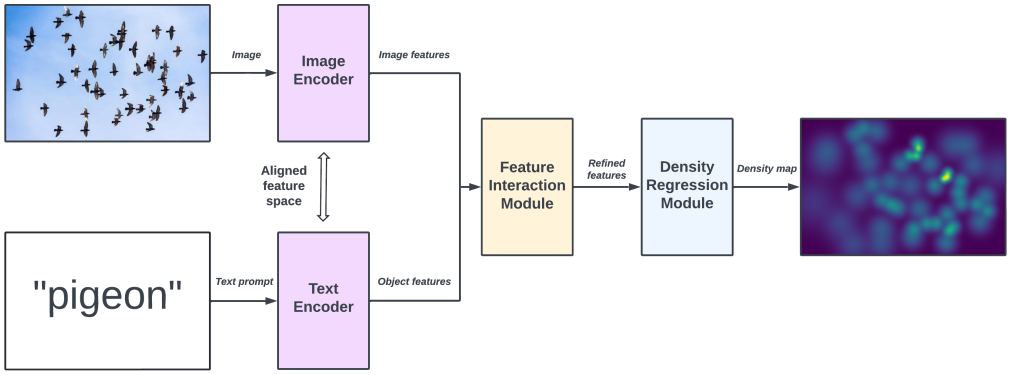


FIGURE 2.3: High-level text-guided zero-shot object counting architecture. A text prompt specifying the object is passed alongside the query image. A multimodal encoder provides aligned image and text embeddings. The text embeddings, representing the object of interest, are matched with the image features and processed in a manner similar to few-shot methods.

2.5 Summary

Originally, object counting focused on specific targets such as crowds, cells, animals, and cars. It involved training specialized networks with extensive labeled samples for each object class, which made scaling to a broader range of visual categories challenging. As a solution, weakly supervised approaches were developed. These methods reduce the reliance on detailed annotations by utilizing simpler forms of data labeling, such as point annotations or image-level counts, enabling easier scaling to new domains, though some supervision is still required. Reference-less methods emerged as a first step towards class-agnostic counting. However, they generally focus on the predominant object class in an image, limiting the ability to count specific objects.

Recently, significant progress has been made in counting arbitrary objects using human-annotated patch exemplars. Nonetheless, the dependency on manually annotated bounding boxes during inference can be impractical for real-world applications.

To address this, zero-shot object counting was introduced, minimizing the dependence on human labor. This approach employs natural language prompts to specify the object of interest. However, utilizing natural language to guide object counting introduces some challenges. Text prompts, unlike patch annotations, do

not provide explicit descriptions of the object, leading to intrinsic ambiguity. Additionally, effective semantic alignment between the textual and visual modalities presents considerable difficulties.

Support-based methods combine the ease of text prompts with the accuracy of patch-based methods. Similar to zero-shot text-guided methods, this approach also relies on external prompts, but instead of text, it utilizes visual information akin to patch-based methods. This strategy effectively mitigates the ambiguity associated with natural language prompts and provides a more direct and reliable way to specify the object of interest without the need for extensive additional annotation.

A visual representation of the different class-agnostic approaches can be seen in Figure 2.4.

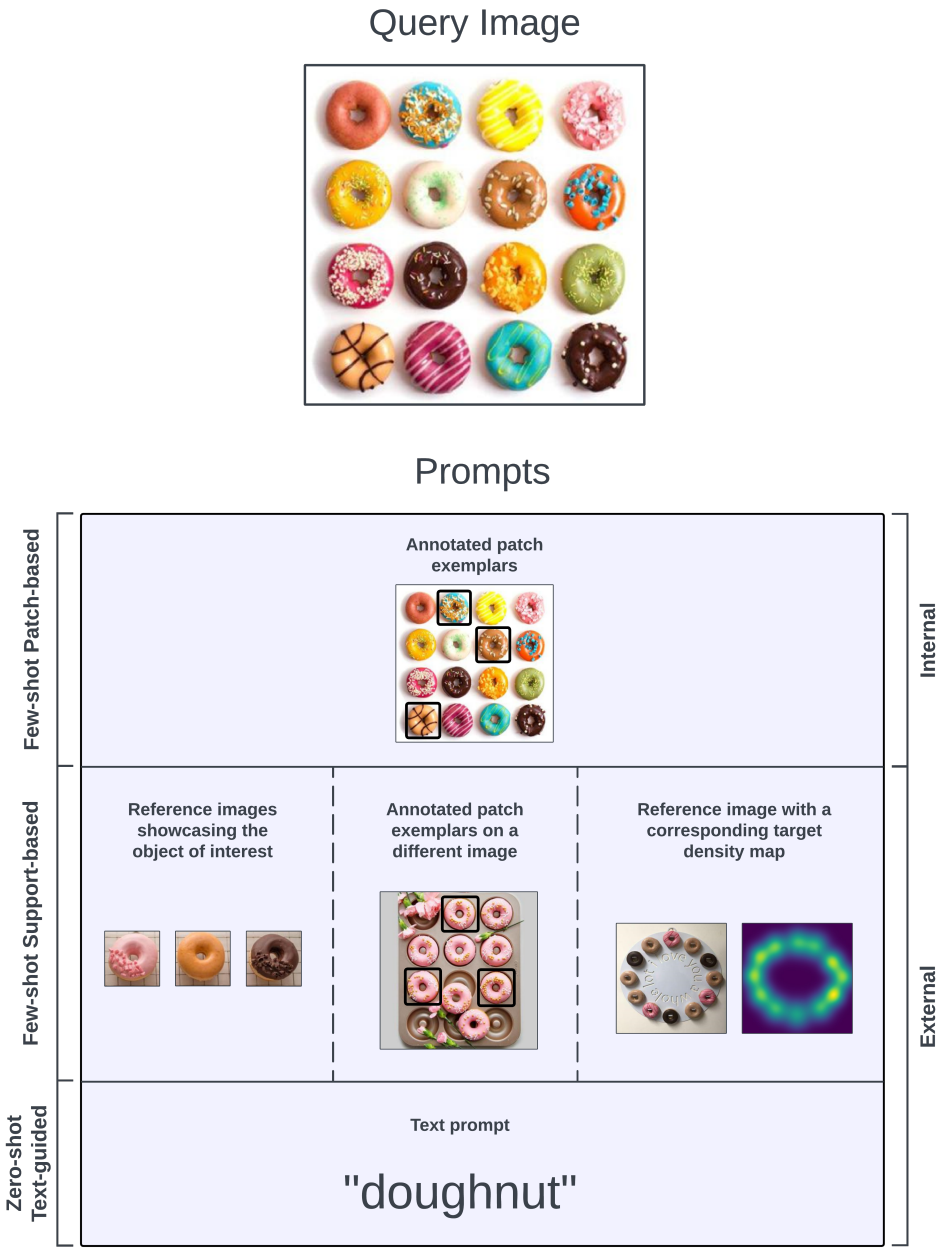


FIGURE 2.4: Class-agnostic object counting approaches.

Chapter 3

Datasets and Metrics

3.1 Class-Specific Object Counting Datasets

3.1.1 CARPK

The Car Parking Lot Dataset (Hsieh, Lin, and Hsu, 2017) consists of 89,777 cars in high-resolution images captured by drones over various parking lots. Unlike other parking lot datasets, CARPK is the first and largest to support object counting. Each image in the dataset is annotated with bounding boxes around the cars.

3.1.2 ShanghaiTech

The ShanghaiTech dataset (Zhang et al., 2016) is a large-scale crowd counting dataset with 1,198 annotated images, containing a total of 330,165 people with their head centers marked. The dataset is divided into two parts: Part A includes images randomly crawled from the Internet, while Part B comprises images taken from busy streets in metropolitan Shanghai. The significant variation in crowd density between these subsets makes accurate crowd estimation more challenging than with most existing datasets.

3.2 Class-Agnostic Object Counting Datasets

3.2.1 FSC-147

Introduced by Ranjan et al., 2021, the FSC-147 dataset establishes a benchmark for class-agnostic few-shot counting tasks. It consists of 6135 images that span 147 diverse categories, including items like kitchen utensils, office supplies, vehicles, and animals. The number of objects per image in this dataset ranges from 7 to 3731, with an average of 56 objects per image. Each image is annotated with dots marking the approximate center of each object instance. Additionally, three randomly selected object instances are designated as exemplars, each annotated with axis-aligned bounding boxes. The dataset is partitioned into training, validation, and test sets, with no overlap in object categories — 89 classes are allocated for training, while 29 are reserved for each of the validation and test sets.

3.2.2 FSCD-LVIS

Despite FSC-147 containing images with numerous objects, the scenes are relatively simple. In FSC-147, the class of the target object is presented with such clarity that identifying which class of objects to count is straightforward, eliminating the need for providing specific exemplars. To address this limitation for real-world deployment of few-shot counting and detection methods, Nguyen et al., 2022 introduced

FIGURE 3.1: Sample images from FSC-147 and FSCD-LVIS datasets.

FSCD-LVIS. This new dataset features more complex scenes with multiple object classes and instances as can be seen in Figure 3.1. Without providing the exemplars for the target class, one cannot definitely guess which the target class is. The dataset contains 6195 images and 372 classes. Unlike FSC-147, FSCD-LVIS includes box annotations for all objects, with three objects randomly selected as exemplars.

3.2.3 CA-44

Jiang et al., 2023 introduced a new benchmark for class-agnostic object counting, named CA-44, which includes 30,085 images sourced from 44 distinct datasets collected via Roboflow (Dwyer et al., 2024). The CA-44 benchmark primarily features images with small and densely packed objects. These characteristics reflect the common attributes of scenes in the object counting task.

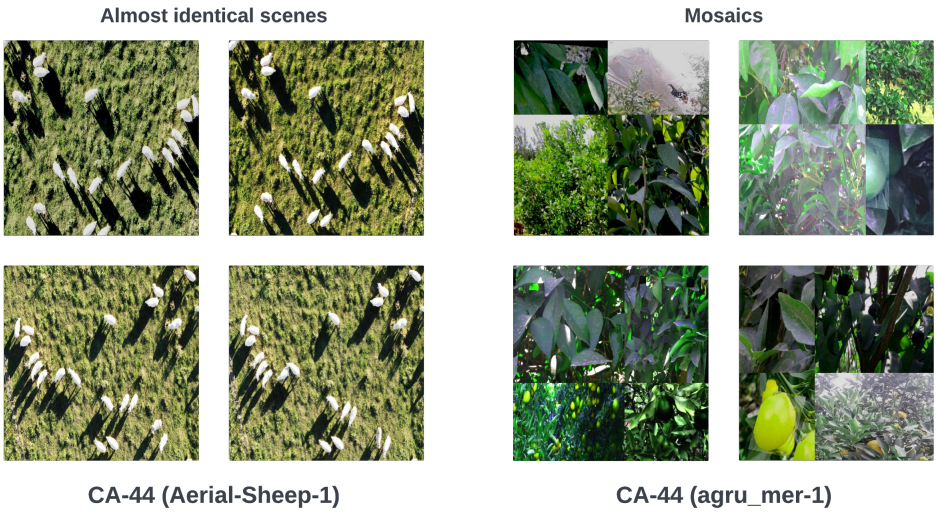


FIGURE 3.2: Sample images highlighting data redundancy in CA-44 dataset.

3.2.4 Summary

The current benchmark dataset, FSC-147, provides only simple scenes that are not representative of the real world. FSCD-LVIS, on the other hand, offers images containing more than one object class with much more complex scenes. Nevertheless, these two datasets are relatively small and do not cover enough domains. Although CA-44 has a significantly larger number of images compared to the FSC-147 and FSCD-LVIS datasets, it contains many nearly identical images. For example, aerial views of sheep flocks comprise more than 10% of the dataset. Additionally, as shown in Figure 3.2, a substantial portion of the dataset consists of mosaics, which means that some image parts can be duplicated.

3.3 Unified Dataset

Each class-agnostic dataset we examined presents its own challenges, including the issue of duplicate images, with some instances occurring across both training and testing subsets. This duplication undermines the foundational purpose of having separate splits for model evaluation. To remove the problems associated with the existing datasets, we refined and combined them into one comprehensive dataset. In addition to the three datasets previously mentioned, we incorporated additional images from Roboflow (Dwyer et al., 2024), specifically targeting scenes with more than 10 object instances to ensure relevance to dense-object counting scenarios.

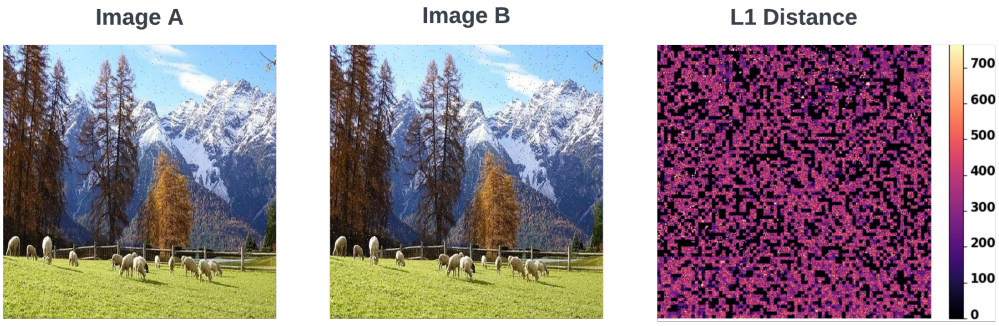


FIGURE 3.3: Duplicate images detected by perceptual hashing. Image A and Image B have minor variations, such as slight changes in brightness and compression artifacts. The L1 distance heatmap highlights the differences between the two images.

After aggregating the images from all sources, our initial step was to eliminate duplicates. We employed perceptual hashing to identify and remove such images. This method, in contrast to direct pixel comparison, better withstands minor variations like compression artifacts, changes in brightness or contrast, and slight cropping, as shown in Figure 3.3. We detected numerous duplicates both within and across the different datasets. Moreover, to make the dataset more balanced, we limited the number of images each class can have, addressing an issue especially prevalent in the CA-44 dataset.

We also standardized and unified similar object class names to enhance consistency and generalizability across the dataset. For example, classes labeled as "mandarin_orange", "orange_(fruit)", and "oranges" in the original datasets were unified under the "orange" class.

Our unified dataset contains 29,819 images, spanning 466 diverse object classes ranging from kitchenware and sports equipment to vehicles and animals, with a

Dataset	Images	Classes	Instances
FSC-147	6,135	147	335,025
FSCD-LVIS	6,195	372	193,148
CA-44	30,085	79	1,171,061
Roboflow	984	11	62,267
Unified	29,819	466	1,263,251

TABLE 3.1: Comparison between the proposed and the existing few-shot object counting datasets. Roboflow is a set of images we have additionally collected.

total count of 1,263,251 objects. A detailed comparison with the original datasets is provided in Table 3.1. We structured the dataset into training, validation, and test splits, ensuring no overlap in object classes among them— 320 classes in the training set, 73 in validation, and 73 in testing.

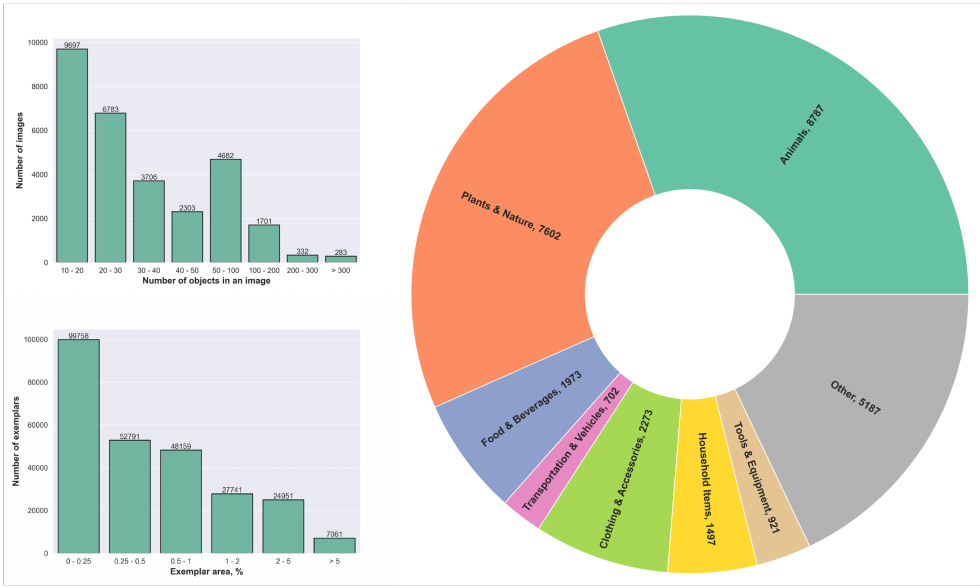


FIGURE 3.4: Statistics of the unified dataset. It covers a diverse range of visual categories, including animals, plants, food, vehicles, clothing, tools, equipment, and people.

Originally, each dataset featured its own approach to generating the target, resulting in variations in the density maps across different datasets. To establish consistency in the ground truth targets, we adopted the density map generation method outlined by Zhang et al., 2016. Given the typically congested scenes characteristic of object counting tasks, we utilized a geometry-adaptive kernel defined as:

$$f(x) = \sum_{i=1}^{N} \delta(x - x_i) \times G_{\sigma_i}(x), where\ \sigma_i = \alpha \bar{d}_i$$

Here, for each object centroid x_i in the ground truth δ, d_i represents the average distance of k nearest neighbors. To generate the density map, $\delta(x - x_i)$ is convolved with a Gaussian kernel $G_{\sigma_i}(x)$, where σ_i is the the standard deviation. In our case, we used $\alpha = 0.3$ and $k = 3$. By applying a Gaussian kernel to each object's centroid,

we obtained ground truth targets that accurately reflect the spatial distribution of the objects.

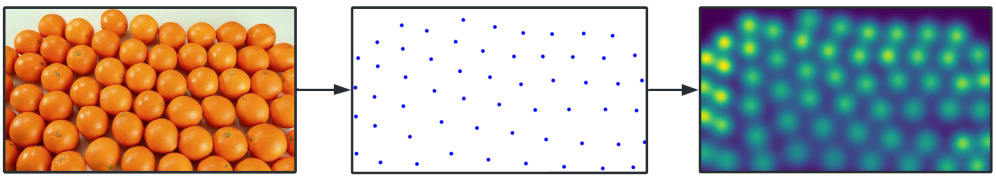


FIGURE 3.5: Original image, object centroids, and the corresponding density map obtained by convolving geometry-adaptive Gaussian kernel.

3.4 Metrics

Object counting methods are usually evaluated using two widely recognized statistical metrics: Mean Average Error (MAE) and Root Mean Squared Error (RMSE). These metrics are defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|C_i - C_i^*|$$

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

Here, N is the total number of images in the evaluation set, C_i is the ground truth count for the i-th image, and C_i^* is the predicted count. The MAE measures the average absolute deviation between the predicted counts and the actual counts across all observations, offering a straightforward and intuitive assessment of predictive accuracy. The RMSE, on the other hand, calculates the average of the squares of these deviations, weighting larger errors more heavily. This characteristic makes RMSE especially sensitive to outliers, thus providing a more conservative measure of model performance. In essence, while MAE reflects the general accuracy of the predictions, RMSE indicates their robustness.

Chapter 4

Approach

4.1 Limitations of State-Of-The-Art

We are interested in an object counting method that can easily scale to handle a large number of novel visual categories while maintaining the highest possible accuracy. Few-shot patch-based methods assume the availability of accurate bounding boxes for extracting patch exemplars during both training and inference, a requirement that is often impractical in real-world applications. While both few-shot support-based and zero-shot text-guided methods offer practical solutions to the challenges posed by manual annotation, there are notable advantages associated with the former. Visual exemplars in the support set allow for a more direct comparison between the exemplars and the objects in query images than text prompts can achieve. Moreover, the reliance on pretrained vision-language models for text-guided counting may introduce biases, as these models might not adequately capture the nuances of specific visual scenarios.

Scheme	Method	MAE	RMSE
Patch-based	FamNet (Ranjan et al., 2021)	22.56	101.54
	CounTR (Liu et al., 2022)	11.95	91.23
	LOCA (Đukić et al., 2023)	**10.97**	**56.97**
Reference-less	RepRPN-Counter (Ranjan and Nguyen, 2023)	26.66	129.11
	RCC (Hobley and Prisacariu, 2022)	17.12	104.53
	LOCA (Đukić et al., 2023)	**16.22**	**103.96**
Text-guided	CLIP-Count (Jiang, Liu, and Chen, 2023)	17.78	106.62
	VLCounter (Kang et al., 2023)	**17.05**	**106.16**

TABLE 4.1: Quantitative comparison of the state-of-the-art methods on the FSC147 test set.

As can be seen in Table 4.1, text-guided methods generally underperform in terms of counting accuracy compared to patch-based methods. Moreover, they have similar or even inferior results compared to reference-less methods, highlighting difficulties in leveraging textual representations effectively. This performance discrepancy could partly be attributed to the simplistic nature of the FSC-147 dataset, which allows reference-less methods to easily identify the objects to count.

Given these observations, we believe that support-based methods should be a better choice in terms of balancing accuracy and ease of use, as they combine the strengths of both visual and external exemplars. However, there was not much research done in this direction. To the best of our knowledge, only a few works have explored this approach extensively. Nevertheless, they lack comprehensive technical details or evaluations on the FSC-147 dataset, making direct comparisons to other class-agnostic methods challenging. Although some patch-based few-shot methods

allow the use of reference images depicting the object of interest, this practice can result in biased object counts. The primary reason is that those methods were explicitly trained with query image patches as exemplars, which have a distribution similar to other objects in the query image. In contrast, external exemplars may originate from varied sources, environments, and conditions, potentially leading to biased results.

4.2 Proposed Solution

To address the limitations identified in the current state-of-the-art for object counting, we propose developing a novel model capable of integrating various forms of external visual prompts as input. Each such prompt can contain visual exemplars, which provide information about the object of interest but are not part of the query image. We aim to count arbitrary objects based on visual guidance without manual annotation.

Formally, the model is designed to accept a query image $Q \in \mathbb{R}^{H \times W \times 3}$ and can utilize one or more of the following types of external visual prompts:

- Prompt $\mathcal{E}$ (Reference Object Image): Contains one or more images $E \in \mathbb{R}^{\acute{H} \times \acute{W} \times 3}$ depicting the object of interest, each potentially providing different instances or views. Having multiple images is useful in scenarios where the object of interest might appear in different forms, orientations, or conditions.
- Prompt $\mathcal{C}$ (Cross-Image Counting): Contains one or more images $C \in \mathbb{R}^{H \times W \times 3}$ with several bounding boxes $B_C \in R^{b \times 4}$ per image that highlight the object of interest. Each such pair can showcase the object of interest in different scenes or amongst different surrounding objects.
- Prompt $\mathcal{K}$ (Reference Counting Result): Contains one or more pairs, each consisting of an image $K \in \mathbb{R}^{H \times W \times 3}$ and a corresponding target density map $D_K \in \mathbb{R}^{H \times W}$. Each pair represents different counting contexts or different object densities. For example, one pair might show a sparse arrangement of objects, while another might depict a denser grouping.

The combination of these prompts can vary, providing flexibility in how much contextual information is available. The objective is to estimate a density map $D_Q \in \mathbb{R}^{H \times W}$ for the object specified by the prompts. The estimated object count could be calculated by summing the values across the density map $N = \text{SUM}(D_Q)$. This approach eliminates the need to annotate each query image, significantly reducing labor and time costs while enhancing scalability and flexibility.

Moreover, we plan to evaluate the model using both newly proposed and existing benchmark datasets. This would allow us to directly compare the effectiveness of our support-based method against patch-based and text-guided methods. Through this approach, we aim to provide a more adaptable and efficient solution for class-agnostic object counting that accommodates a wide range of applications and minimizes reliance on labor-intensive manual annotations.

4.3 Architecture

To implement our proposed solution, we introduce a novel transformer-based architecture. This model incorporates a feature extraction module specifically designed to derive representations from both the query image and the visual exemplars. Taking inspiration from LOCA (Đukić et al., 2023) and CounTR (Liu et al., 2022), these representations are subsequently processed by a feature interaction module. This module refines the exemplars by incorporating contextual information from the query image features. Following this, the enhanced exemplars are depthwise convolved with the image features to generate a response map. This map is then passed to the density prediction module, which further refines and upscales it to produce the final density map. Each component of this architecture is explained in detail in the following sections.

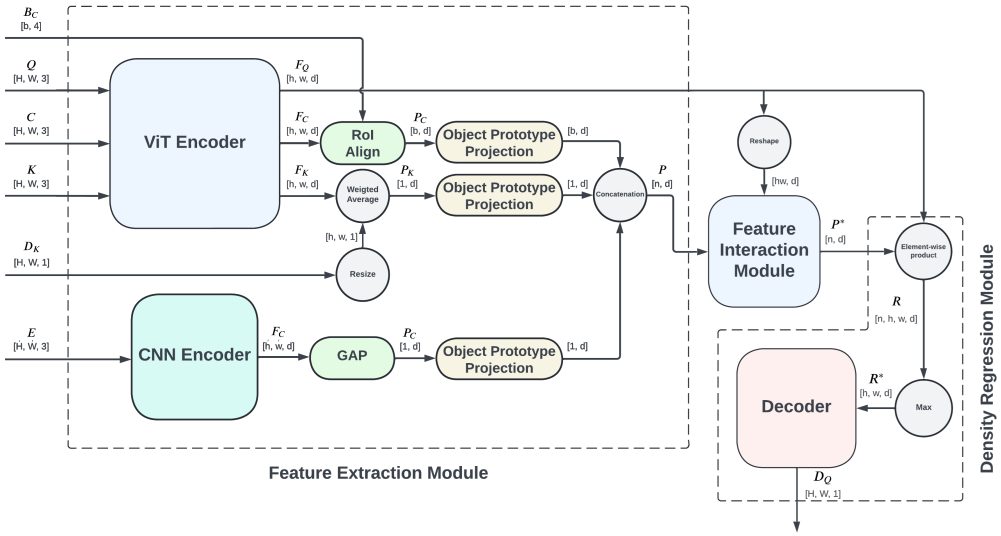


FIGURE 4.1: High-level architecture of the proposed model.

4.3.1 Feature Extraction Module

The feature extraction module is specifically designed to efficiently handle all types of external prompts, extracting rich representations of both image and exemplar prototypes. It consists of two primary encoders: a Vision Transformer (ViT) and a lightweight Convolutional Neural Network (CNN).

The ViT architecture excels in generating high-dimensional feature maps that comprehensively capture relational information across the entire image. For our purposes, we employ EfficientViT (Cai et al., 2022), a state-of-the-art transformer architecture optimized for high-resolution dense prediction tasks. It incorporates a multi-scale attention module that enables global receptive field capabilities and multi-scale learning. These features are accomplished with minimal hardware demands, making it ideal for our task. This encoder is used for processing images Q, C, and K, which typically contain complex scenes with multiple objects and detailed backgrounds. After feeding the images to the encoder, we obtain the corresponding feature maps $F_Q, F_C, F_K \in \mathbb{R}^{h \times w \times d}$.

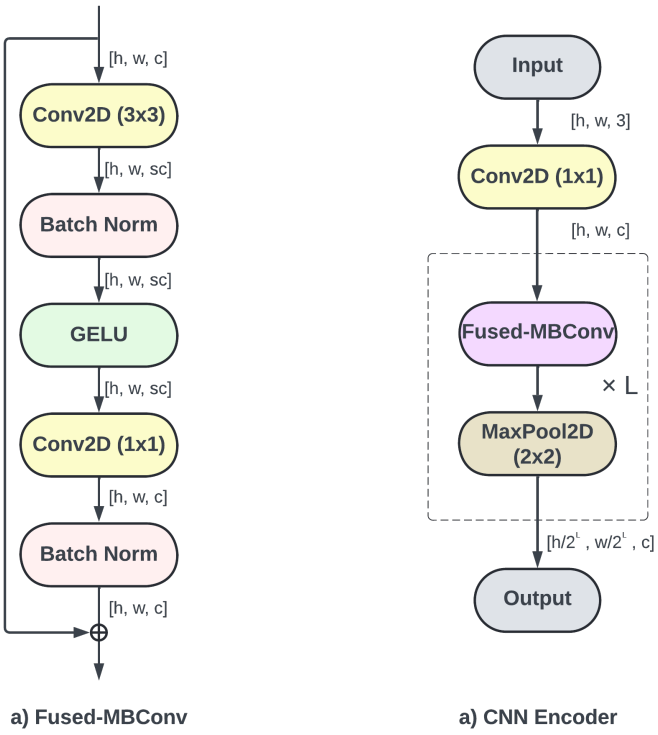


FIGURE 4.2: a). Fused-MBConv block. We use a variant without the Squeeze-and-Excitation block in the middle. b) CNN Encoder. The encoder starts with a 1×1 convolution that projects the input image into a higher dimensional space. This is followed by L repetitions of the Fused-MBConv block, each succeeded by a 2×2 MaxPooling layer.

Conversely, the CNN encoder is used for processing the image E, which depicts a single object of interest. This encoder is optimized to efficiently extract object features from these simpler images, focusing on key characteristics without the additional computational overhead. It consists of a pointwise convolution that projects the RGB image into a high-dimensional space and a number of Fused-MBConv (Gupta and Tan, n.d.) blocks followed by a max pooling operation as shown in Figure 4.2. The resulting feature map $F_E \in \mathbb{R}^{\acute{h} \times \acute{w} \times d}$ captures the key characteristics of the object of interest.

The extracted image features are then used to obtain object prototypes represented by high-dimensional vectors (embeddings). Specifically, CNN features F_E go through the Global Average Pooling layer (Lin, Chen, and Yan, 2013) resulting in the $P_E \in R^d$ vector. F_C with the corresponding bounding boxes B_C are passed to the RoI Align layer (He et al., 2017) with a single bin leading to the $P_C \in \mathbb{R}^{b \times d}$ tensor. For F_K, the features are element-wise multiplied with the density map D_K to compute a weighted average, represented by the formula $P_K = \frac{\sum(F_K \odot D_K)}{\sum D_K}$, where $\odot$ denotes element-wise multiplication, producing the vector $P_K \in \mathbb{R}^d$.

Given that the object prototypes are derived from various types of prompts, they inherently carry features with varying levels of abstraction and focus. To address this variability, our architecture incorporates an additional projection module. This module is designed to map the object prototypes from different sources into a unified semantic space. The projection module consists of a fully connected layer coupled with an activation function, followed by an Efficient Channel Attention (Wang et

al., 2019) layer. This layer dynamically adjusts the importance of each channel in the feature vector, thereby enhancing the most relevant features and suppressing the less useful ones. The detailed scheme of this projection module can be seen in Figure 4.3.

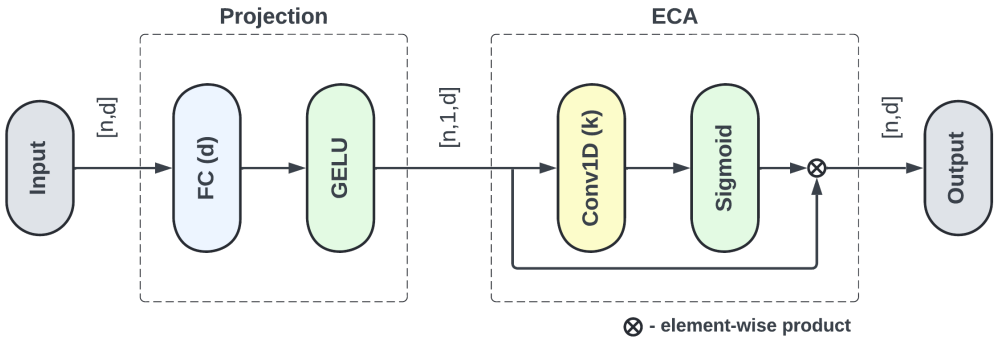


FIGURE 4.3: Object prototype projection module. Here, n represents the number of prototypes, and d is the dimension of each prototype. The input is first projected using a fully connected layer, followed by GELU activation. The Efficient Channel Attention (ECA) then generates channel-specific weights by applying a 1D convolution that treats each channel as an individual element in a sequence, focusing on local inter-channel relationships within a fixed-size window. These weights are normalized through a sigmoid function and element-wise multiplied with the projected vector.

4.3.2 Feature Interaction Module

The Feature Interaction Module is designed to fuse the information from the query image features with the object prototypes, enhancing the model's ability to generalize and specifically tune the exemplars to the contextual features of the query image. It generalizes and specifically tunes the exemplars to the image features by alternating self- and cross-attention layers. Self-attention is employed on the object prototypes, enabling the model to consolidate information by analyzing the interrelationships among them. Subsequently, cross-attention is utilized to provide the interaction between the prototypes and the image features. Here, the image features serve as both the key and value in the attention mechanism, with the object prototypes acting as the query. This setup enables the model to map and align prototype features directly against the corresponding features in the query image, effectively tuning the prototypes to be more contextually relevant to the specific image being processed.

The module leverages a series of modified transformer encoder layers, which comprise multi-head attention (Vaswani et al., 2017) followed by a feedforward network. Its detailed diagram is presented in Image 4.4. Given tensors $Q \in \mathbb{R}^{n\times d}$, $K, V \in \mathbb{R}^{m\times d}$ as input, this layer performs the following transformations:

$$Q^{'} = Q + \text{Drop}((\text{MHA}(\text{LN}(Q), K, V)))$$

$$Q^{''} = \text{Drop}(\text{GELU}(\text{LN}(Q^{'})W_1 + b_1))$$

$$Q^{'''} = \text{Drop}(Q^{''}W_2 + b_2)$$

$$Q^{*} = Q^{'} + Q^{'''}$$

Here, LN represents layer normalization (Ba, Kiros, and Hinton, 2016), Drop denotes the dropout operation (Srivastava et al., 2014), and Q^* is the output.

The combined object prototypes tensor $P = P_0$ is formed by stacking P_E, P_C and P_K. When no prompts are provided, the model operates in a reference-less mode, utilizing a single learned vector of tokens as a generalized object prototype, which serves as P. The prototypes, along with the reshaped query features $F_Q \in \mathbb{R}^{hw \times d}$, pass through the feature interaction layers in the following sequence, where $l = 1...L$ indexes the layers within the module.

$$P_l^* = \text{TransformerLayer}_l^{self}(Q = P_{l-1}, K = P_{l-1}, V = P_{l-1})$$

$$P_l = \text{TransformerLayer}_l^{cross}(Q = P_l^*, K = F_Q, V = F_Q)$$

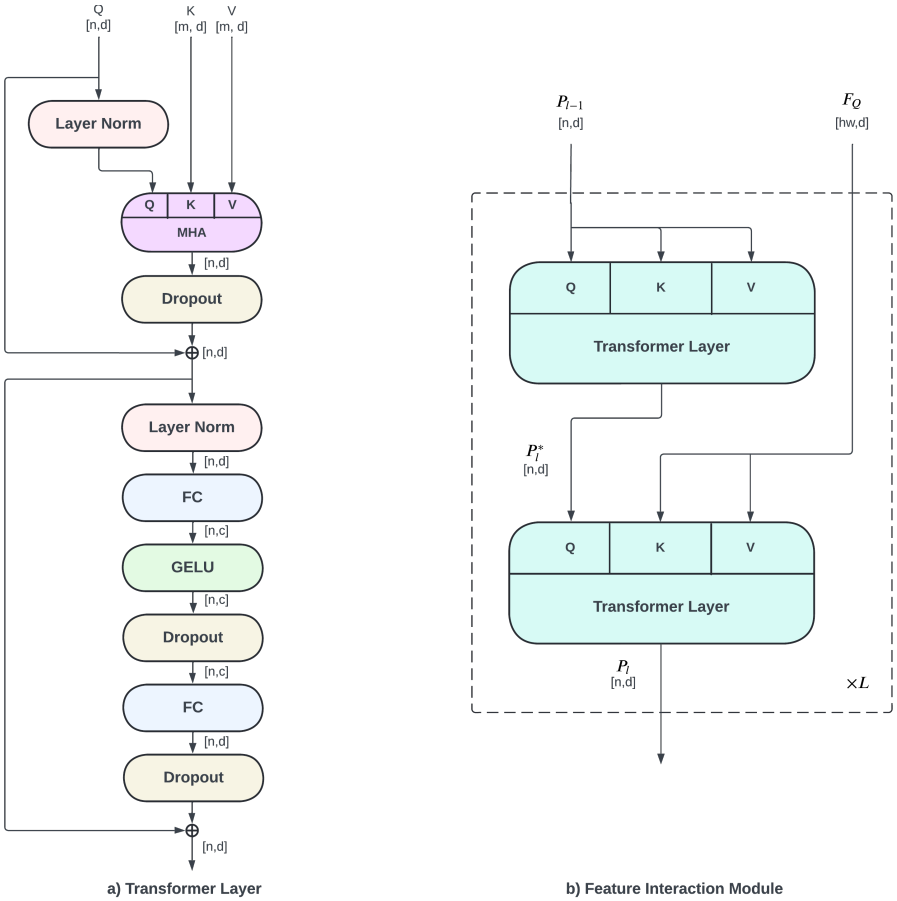


FIGURE 4.4: a) The transformer layer incorporates a multi-head attention followed by a feedforward network. b) Feature interaction module comprises L blocks of self- and cross-attention transformer layers.

4.3.3 Density Regression Module

The refined object prototypes $P^* = P_L$ are depth-wise correlated with the image features F_Q. Each prototype thus generates a multi-channel similarity tensor. The individual n prototypes are then combined through a per-channel, per-pixel max operation, forming a joint response tensor R^* with the same dimensions as F_Q. This

response tensor is then processed by a decoder. The decoder consists of three up-sampling blocks, each comprising a convolution layer followed by a 2× bilinear interpolation. Following the final up-sampling block, a fully connected layer is employed as the density regressor. This layer outputs a one-channel density map D_Q, spatially corresponding to the input query image, which represents the estimated distribution of objects within the image. The total count of objects is then estimated by summing the values across this density map, expressed as $N = \text{SUM}(D_Q)$.

4.3.4 Implementation Details

We resize the input images Q, C, and K to the $H = W = 512$ pixels. The image E, which is processed by the CNN encoder, is resized to the $\acute{H} = \acute{W} = 64$ pixels. For the ViT encoder, we use the EfficientViT-SAM-L0 variation, which outputs a feature map with 256 channels and has a spatial reduction of 8. This is why the decoder in the density regression module has 3 up-sampling blocks with 2× interpolation. Similarly, the CNN encoder also has 3 blocks, leading to the same 8× spatial reduction and producing the 8×8 tensor with 256 channels. The learned tokens for the reference-less mode are initialized from a normal distribution.

The feature interaction module has 3 blocks, resulting in 6 alternating self- and cross-attention transformer layers. The MHA inside the transformer layer consists of 8 attention heads with a hidden dimension of 256, while the feedforward network has the hidden dimension $c = 1024$. Dropout is applied with a probability of 0.1.

4.4 Training

4.4.1 Procedure

The proposed and benchmark datasets do not provide a fixed set of external exemplars for each query image. Consequently, our training procedure is designed to dynamically generate these exemplars rather than relying on a predefined set. For each query image, we randomly select a subset of images from the dataset that share the same object class but are distinct from the query image. We then use this subset to generate the prompts. This approach ensures the exemplars are both relevant and varied, enhancing the model's robustness by preventing overfitting to specific exemplar instances.

The dataset already includes annotations for exemplar bounding boxes and density maps, which simplifies the process of generating prompts. Specifically, prompts $\mathcal{C}$ and $\mathcal{K}$ are directly derived without additional processing steps. The images for prompt $\mathcal{E}$ are produced by cropping around the object of interest using the exemplar bounding boxes. Note that $\mathcal{C}$, $\mathcal{K}$, $\mathcal{E}$ are each sourced from different entries within the dataset, ensuring that they feature distinct exemplars from different images.

4.4.2 Loss Function

Our model is trained using the normalized L2 loss between the predicted density map D_Q and the ground-truth map D. It is defined as:

$$\mathcal{L} = \frac{1}{C} \|D - D_Q\|_2^2$$

Here, C represents the number of objects in the batch. This normalization ensures that the loss emphasizes errors in images containing many objects, which often present the most challenging scenarios due to high local object densities.

4.4.3 Augmentation

To enhance the robustness of our model, we apply standard data augmentation techniques to both the query image and the images used for prompts. These techniques include horizontal flipping and color jitter, which help the model generalize better across varying visual conditions.

Furthermore, we implement additional augmentations targeting the objects within these images. Specifically, we employ a random perspective transformation and apply color jitter individually to each object. This means that each object receives a unique degree of augmentation. While object counting tasks typically assume that objects within the same image should have a similar appearance, this additional layer of augmentation is crucial for addressing challenges associated with using external exemplars. Since these exemplars can significantly differ from the objects in the query image in terms of appearance, such augmentations ensure that our model can reliably generalize from them despite their visual discrepancies.

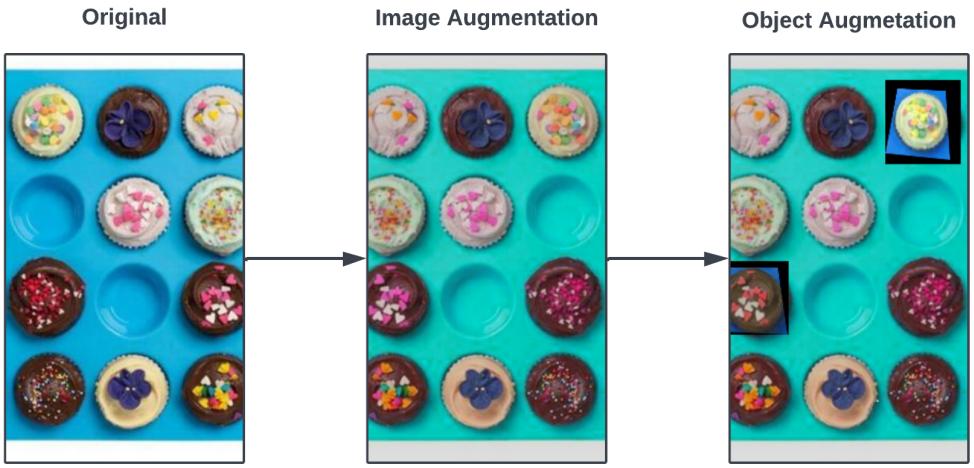


FIGURE 4.5: Augmentation example.

4.4.4 Details

The parameters of the ViT encoder are frozen, while all other model parameters are trained over 50 epochs using the AdamW (Loshchilov and Hutter, 2017) optimizer with a weight decay of 10^{-3}. The learning rate is set to 10^{-4} for the first 30 epochs and reduced to 10^{-5} for the remaining 20. We employ gradient clipping with a maximum norm of 0.01 to stabilize training. The model is implemented in Pytorch (Paszke et al., 2019) and is trained on a single A100 GPU, using a batch size of 8 in mixed precision mode, with the entire training process taking approximately 60 hours.

The types of prompts used are selected randomly for each batch. Given the three types of prompts available $\mathcal{C}$, $\mathcal{K}$, $\mathcal{E}$, there are seven possible combinations of these prompts that can be utilized, plus an additional reference-less mode where

no prompts are used. Each combination offers a different context and level of detail about the target object, varying the information provided to the model. When a particular type of prompt is not used in a batch, the specific model components or weights that process this prompt do not receive training updates in that iteration. By not consistently training on the same prompt types, the model may better learn to handle varied input configurations and adapt to different kinds of information about the objects. It encourages the model to not rely excessively on any single prompt type, which can be beneficial in scenarios where the availability of certain types of data may vary.

During both training and evaluation, the prompt $\mathcal{E}$ comprises 3 images, each cropped from the same original image to provide multiple views of the object. The prompt $\mathcal{C}$ includes 1 image that contains 3 bounding boxes, each highlighting an object of interest within the scene. Meanwhile, prompt $\mathcal{K}$ consists of 1 image accompanied by a corresponding density map, providing context on the spatial distribution of objects. To get a fair and consistent evaluation, we fix the random seed during the evaluation phase. This ensures that the same set of prompts is used for each evaluation run, making the results comparable and reproducible.

4.5 Evaluation

4.5.1 Quantitative Results

To assess the performance of our model, we compute the Mean Average Error (MAE) and the Root Mean Squared Error (RMSE) based on the predicted and actual object counts. Specifically, we test our model in three modes: patch-based, support-based, and reference-less. For our dataset, we trained the state-of-the-art methods according to the procedures described in their original papers. For the FSC-147 dataset, we used the reported metrics. Additionally, we trained the CounTR model using reference object images as exemplars without changing the architecture.

In the patch-based mode, our model utilizes patch exemplars located within the query image itself, denoted as prompt $\mathcal{P}$. For this mode, the prototype extraction process follows the same method as used for prompt $\mathcal{C}$. In the reference-less mode, instead of using object-specific prototypes, we employ learned tokens that represent generic object features.

On the proposed dataset, our model is compared with FamNet, LOCA, and CounTR. In the few-shot patch-based mode on the validation set, our model achieves performance comparable to LOCA, albeit with a slightly higher RMSE. On the test set, it shows a slight underperformance, with a 7.67% and 21.93% reduction in MAE and RMSE, respectively, compared to the best-performing model. This discrepancy is attributed to the fact that other methods leverage additional techniques tailored to the patch-based scenario, such as integrating information from bounding box coordinates or implementing post-hoc error correction routines. In the reference-less mode, our model once again matches LOCA, achieving state-of-the-art results. It performs slightly better on the validation set but exhibits worse performance on the test set. For the support-based mode, we assess all possible combinations of prompts. We use the same number of exemplars as during training. When utilizing all three types of prompts ($\mathcal{C}$, $\mathcal{K}$, and $\mathcal{E}$), our model surpasses the best patch-based method on the validation set and achieves similar results on the test set. When each prompt type is used separately, the model achieves its best performance with the $\mathcal{C}$ prompt, having 13.91 MAE and 6.2 RMSE. Utilizing the $\mathcal{K}$ prompt results in a 15.7% and 9.36% increase in MAE and RMSE, respectively, compared to the $\mathcal{C}$ prompt

Scheme	Method	Prompts				Val Set		Test Set	
		$\mathcal{P}$	$\mathcal{C}$	$\mathcal{E}$	$\mathcal{K}$	MAE	RMSE	MAE	RMSE
Patch-based	FamNet	+	-	-	-	27.61	65.51	32.52	79.78
	CounTR	+	-	-	-	15.34	38.23	15.22	40.04
	LOCA	+	-	-	-	**13.11**	**30.56**	**14.91**	**34.42**
	Ours	+	-	-	-	13.6	34.23	16.16	44.09
Support-based	CounTR	-	-	+	-	17.23	48.11	18.58	52.11
	Ours	-	+	+	+	**12.78**	**30.09**	15.3	**39.42**
	Ours	-	+	+	-	12.92	31.22	**15.1**	39.95
	Ours	-	+	-	+	13.72	35.96	16.01	40.23
	Ours	-	-	+	+	14.59	38.77	15.99	46.34
	Ours	-	+	-	-	13.91	35.66	16.2	44.51
	Ours	-	-	+	-	15.7	40.13	17.04	49.98
	Ours	-	-	-	+	16.5	45.78	18.8	55.02
Reference-less	LOCA	-	-	-	-	19.04	59.2	**21.13**	**63.06**
	Ours	-	-	-	-	**18.55**	**54.98**	22.24	65.72

TABLE 4.2: Comparison with state-of-the-art on the proposed dataset.

on the validation set. This performance drop can be attributed to the fact that $\mathcal{K}$ contains only one exemplar, placing the model in a one-shot setting. Relative to CounTR, our model shows a 10-15% improvement on the same inputs. The results are summarized in Table 4.2.

On the FSC-147 dataset, we observe similar results, presented in Table 4.3. Our model matches the state-of-the-art in the reference-less mode. In the patch-based mode it performs on par with CounTR. In the support-based mode our model outperforms CounTR and achieves the results close to the patch-based. However, the gap between patch-based and support-based is larger here than on the proposed dataset. This is probably due to the smaller dataset size, so our model underfits and does not generalize enough.

Scheme	Method	Prompts				Val Set		Test Set	
		$\mathcal{P}$	$\mathcal{C}$	$\mathcal{E}$	$\mathcal{K}$	MAE	RMSE	MAE	RMSE
Patch-based	FamNet	+	-	-	-	24.32	70.94	22.56	101.54
	CounTR	+	-	-	-	13.13	49.83	11.95	91.23
	LOCA	+	-	-	-	**10.23**	**32.56**	**10.97**	**56.97**
	Ours	+	-	-	-	13.45	39.37	11.82	66.13
Support-based	CounTR	-	-	+	-	14.87	55.42	13.58	92.37
	Ours	-	+	+	+	13.64	46.22	**12.39**	**77.81**
	Ours	-	+	+	-	**13.51**	**44.34**	12.91	83.75
	Ours	-	+	-	+	14.22	51.67	14.17	89.46
	Ours	-	-	+	+	14.59	49.41	13.93	82.07
	Ours	-	+	-	-	14.05	57.08	13.78	84.19
	Ours	-	-	+	-	15.01	58.13	14.51	93.82
	Ours	-	-	-	+	16.33	67.52	15.05	105.86
Reference-less	LOCA	-	-	-	-	17.43	**54.96**	16.22	103.96
	Ours	-	-	-	-	**17.01**	72.84	**16.17**	**102.02**

TABLE 4.3: Comparison with state-of-the-art on the FSC-147 dataset.

We also examine our model's performance based on the type of the prompt used

and the size of the objects within those prompts. For prompts $\mathcal{E}$ and $\mathcal{C}$, we calculate the size as the average area of their respective exemplar bounding boxes. For prompt K, we compute the average area of all objects within the image. Objects are categorized based on the area: small (less than 0.5% of the image), medium (between 0.5% and 5%), and large (greater than 5%). As indicated in Table 4.4, the performance varies significantly with the size of the object and the type of prompt. Cross-image counting using prompt $\mathcal{C}$ is most effective with large object exemplars, whereas the use of reference object images (prompt $\mathcal{E}$) performs best with small exemplars. This variation can be attributed to the influence of surrounding context in the feature maps. For small objects, this context can disproportionately affect the feature representation, leading to biased embeddings. In contrast, reference object images typically contain minimal contextual information beyond the object itself.

Prompt	Average Object Size		
	Small	Medium	Large
$\mathcal{C}$	15.93	13.05	12.55
$\mathcal{E}$	14.02	16.01	15.47
$\mathcal{K}$	15.57	17.48	14.59

TABLE 4.4: MAE by prompt type and object size category on the validation set of the proposed dataset.

Furthermore, we assess the cross-dataset generalization capabilities of our model following the methodology described by Ranjan et al., 2021. Specifically, we train our model on the FSC-147 dataset and subsequently evaluate it on the CARPK dataset. This latter dataset comprises aerial images of parking lots used for car counting, which presents a context considerably different from that of FSC-147. To avoid any overlap in object classes between the training and testing datasets, we ensure that car images are excluded from the FSC-147 training set. According to the approach used by Ranjan et al., 2021, we select twelve exemplars from the CARPK training set, which serve as reference object images, constituting prompt E. The performance of our model on this dataset, as reported in Table 4.5, demonstrates state-of-the-art cross-dataset generalization. This result underscores that current few-shot methods are generally not designed nor trained to work effectively with external prompts.

Method	MAE	RMSE
FamNet	28.84	44.47
LOCA	9.97	12.51
Ours	**8.62**	**10.84**

TABLE 4.5: Cross-dataset generalization comparison on the CARPK dataset.

4.5.2 Qualitative Results

Figure 4.6 presents the qualitative results of our model applied to the cross-image counting task using prompt $\mathcal{C}$. In the displayed example, we compare the predicted density maps for the same query image using various exemplars. Despite noticeable differences in appearance between the objects in the query image and those in the prompts, the model accurately estimates counts that are very close to the actual numbers. The predicted density maps visually align well with the ground truth, demonstrating the model's precision in spatially localizing objects within the scene.

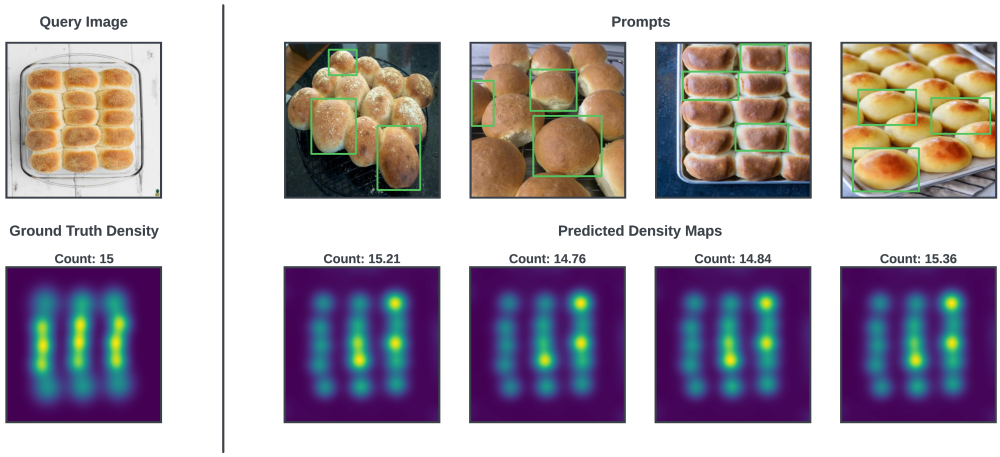


FIGURE 4.6: Qualitative results in the cross-image counting task.

4.5.3 Complexity

As illustrated in Table 4.6, our proposed architecture has nearly the same total number of parameters as LOCA and half as many trainable parameters, while its computational complexity is comparable to other state-of-the-art methods. Despite incorporating the Vision Transformer (ViT) as a backbone, which is typically resource-intensive, the overall complexity is not largely affected. It's important to note that during training, the ViT encoder processes additional images from the $\mathcal{C}$ and $\mathcal{K}$ prompts, significantly increasing computational demands. However, during inference, the complexity is reduced as the features from the prompts can be precomputed and used directly. This optimization also applies to the CNN encoder and the object prototype projection modules, which may not be utilized during inference, further reducing computational load.

Method	GFLOPs	Number of parameters	
		Trainable	Total
FamNet	55	760K	26M
CounTR	91	99M	100M
LOCA	80	11M	37M
Ours	102	6M	36M

TABLE 4.6: Computational complexity and the number of parameters.

Chapter 5

Experiments

5.1 Backbone Selection

Selecting an appropriate backbone is crucial for optimizing the performance of our model. Prior to finalizing the model's architecture and initiating full-scale training, we conducted an analysis of various pretrained backbones. We specifically evaluated four different options:

1. **EfficientNet-50 SwAV**. SwAV (Caron et al., 2020) is a self-supervised learning algorithm that generates rich features that are well-suited for transfer learning on downstream tasks. We tested the CNN model EfficientNet-50 (Tan and Le, 2019) with SwAV weights, which is also employed as the backbone in LOCA.
2. **EfficientViT-L0 SAM**. EfficientViT (Cai et al., 2022) is a series of lightweight, high-resolution Vision Transformer models. We tested the L0 variant with an 8×8 patch size, pretrained for as backbone in the Segment Anything Model (Kirillov et al., 2023), a state-of-the-art zero-shot segmentation model. We believe the features learned for zero-shot segmentation are highly effective for our few-shot object counting task.
3. **ViT-L DINOv2**. DINOv2 (Oquab et al., 2023) is a family of models trained in a self-supervised manner, which produce robust visual features and achieve state-of-the-art results in downstream tasks. We tested a distilled ViT-L variant with a 14×14 patch size.
4. **ViT-H I-JEPA**. I-JEPA (Assran et al., 2023) is another self-supervised approach that excels in learning highly semantic image features. It achieves results comparable to DINOv2 but is more effective at capturing low-level image details. We tested a ViT-H model with a 16×16 patch size.

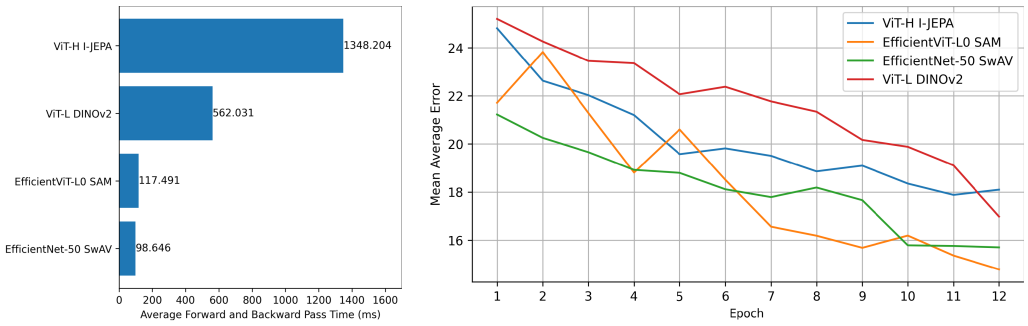


FIGURE 5.1: Evaluation of different backbones.

We used these models as backbones in our architecture and performed training on a small subset of our proposed dataset for a fixed number of steps. The overall architecture did not differ much from the final variant, only missing the object prototype projection modules. Additionally, we adjusted the number of blocks in

the density regression module to match each backbone's spatial reduction — three blocks for EfficientViT and EfficientNet, and four blocks for ViT models, with the last block in the DINOv2 setup featuring an upsampling factor of 1.75.

The training was performed on 10% of the dataset for 12 epochs without the validation and test sets, meaning we only analyzed the behavior on the training set. We analyzed the convergence speed and execution time of each backbone. As depicted in Figure 5.1, all backbones delivered comparable MAE scores, with EfficientViT and EfficientNet showing slightly better results. However, the ViT-L and ViT-H models exhibited up to 13.5× longer execution times.

Given our time and computational constraints, we opted not to proceed with the regular ViT transformer models. The final decision was between EfficientNet and EfficientViT, both offering similar performance. We ultimately selected EfficientViT due to its multi-scale feature capabilities.

Our approach to selecting a backbone has both strengths and limitations. Testing different backbones using a consistent, scaled-down version of the architecture allows for a more rapid and resource-efficient comparison of their impacts on the model's performance. This is crucial when resources are limited or when a quick decision is needed. However, this method does not assess how changes in the backbone might interact with different architectural adjustments, such as projection modules or feature interaction components.

5.2 Ablation Study

We finally analyze the architectural design choices and examine the influence of the input resolution. The experiments are performed on the proposed dataset using all three external prompt types. The model is trained on an A40 GPU with a batch size of 4, maintaining the constant learning rate of 3×10^{-5} for 50 epochs.

We first examine the significance of maintaining a high input image resolution. Upon reducing the resolution from 512×512 to 448×448 pixels, we observe a 5% decrease in performance. Further reduction in resolution to 384×384 pixels results in an 11% drop in MAE and a 15% drop in RMSE compared to the baseline. The results are presented in Table 5.1.

Resolution	384	448	512
MAE	14.56	13.77	13.11
RMSE	39.29	35.86	34.15

TABLE 5.1: Impact of the input image resolution.

Next, we evaluate the importance of the object prototype projection modules. We test three alternative configurations: one without any projection module, one with a projection module that excludes the Efficient Channel Attention (Wang et al., 2019) layer, and one where the ECA is replaced with a Squeeze-and-Excitation (Hu et al., 2017) block.

	No projection	Only FC	ECA	SE
MAE	13.78	13.53	13.11	12.98
RMSE	37.02	35.34	34.15	34.54

TABLE 5.2: Ablation of the object prototype projection module.

As shown in Table 5.2, removing the projection layer or excluding the ECA leads to the 5% and 3% performance drop, respectively. The SE block achieves results almost identical to those with the ECA, having a slightly better MAE. However, the ECA is still preferred for being more lightweight.

Additionally, we explore the impact of varying the number of blocks in the feature interaction module, where each block consists of one self-attention and one cross-attention transformer layer. Results, displayed in Table 5.3, indicate that using three blocks provides the optimal balance between performance and model complexity.

# Blocks	1	3	5
MAE	15.01	13.11	12.99
RMSE	43.72	34.15	36.35

TABLE 5.3: Ablation of the feature interaction module.

Chapter 6

Conclusions

6.1 Discussion

This work addresses the challenge of class-agnostic object counting. We conducted a comprehensive analysis and comparison of current state-of-the-art methods. Existing approaches primarily leverage either patch exemplars or text prompts to specify the target objects, each one presenting its own set of challenges. Our research focuses on few-shot object counting using external visual prompts. This approach requires the model to count objects from new categories in an image, aided by a few exemplars not originally part of that image.

We have developed a novel network designed to efficiently utilize various types of external prompts. The model incorporates a multi-scale feature extractor and a feature interaction module that effectively manages both intra-relations and inter-relations among the features. Additionally, we address the limitations of current class-agnostic benchmark datasets by introducing a unified and refined dataset that better meets the needs of this domain.

Our experiments demonstrate that our model achieves results comparable to existing few-shot patch-based methods. Notably, on the CARPK dataset, our model exhibits state-of-the-art performance in cross-dataset generalization, underscoring its robustness and the effectiveness of our approach in real-world scenarios.

6.2 Limitations and Future Work

While our model demonstrates strong performance across various settings, it is not without its limitations. Currently, the effectiveness of the model heavily relies on the quality and relevance of the external prompts provided, which can vary significantly across different datasets and scenarios. Future work could explore various strategies or adaptive mechanisms for exemplar selection to enhance robustness and accuracy. Additionally, while the computational requirements of our model are not excessively high, they could still limit deployment in resource-constrained environments. Optimizing the model to reduce computational demands without sacrificing performance would be a beneficial direction.

Further, we envision adapting the proposed approach to few-shot object detection. This adaptation would require enhancements in the model's ability to not only recognize and count but also precisely localize objects within an image. It would involve more sophisticated handling of spatial relationships and potentially integrating additional features from exemplars.

Bibliography

Amini-Naieni, Niki et al. (2023). "Open-world Text-specified Object Counting". In: *ArXiv* abs/2306.01851. URL: https://api.semanticscholar.org/CorpusID:259075464.

Assran, Mahmoud et al. (2023). "Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. URL: https://api.semanticscholar.org/CorpusID:255999752.

Ba, Jimmy, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). "Layer Normalization". In: *ArXiv* abs/1607.06450. URL: https://api.semanticscholar.org/CorpusID:8236317.

Cai, Han et al. (2022). "EfficientViT: Multi-Scale Linear Attention for High-Resolution Dense Prediction". In: URL: https://api.semanticscholar.org/CorpusID:262824134.

Caron, Mathilde et al. (2020). "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments". In: *ArXiv* abs/2006.09882. URL: https://api.semanticscholar.org/CorpusID:219721240.

Chan, Antoni B and Nuno Vasconcelos (Sept. 2009). "Bayesian Poisson regression for crowd counting". In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE. DOI: 10.1109/iccv.2009.5459191. URL: http://dx.doi.org/10.1109/ICCV.2009.5459191.

Chattopadhyay, Prithvijit et al. (2017). "Counting Everyday Objects in Everyday Scenes". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4428–4437. DOI: 10.1109/CVPR.2017.471.

Cholakkal, Hisham et al. (2022). "Towards Partial Supervision for Generic Object Counting in Natural Scenes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.3, pp. 1604–1622. DOI: 10.1109/TPAMI.2020.3021025.

Dwyer, B. et al. (2024). *Computer vision tools for developers and enterprises*. URL: https://roboflow.com/.

Gupta, Suyog and Mingxing Tan (n.d.). *EfficientNet-EDGETPU: Creating accelerator-optimized neural networks with AutoML*. URL: https://research.google/blog/efficientnet-edgetpu-creating-accelerator-optimized-neural-networks-with-automl/.

He, Kaiming et al. (2017). "Mask R-CNN". In: URL: https://api.semanticscholar.org/CorpusID:54465873.

Hobley, Michael A. and Victor Adrian Prisacariu (2022). "Learning to Count Anything: Reference-less Class-agnostic Counting with Weak Supervision". In: *ArXiv* abs/2205.10203. URL: https://api.semanticscholar.org/CorpusID:248965349.

Hsieh, Meng-Ru, Yen-Liang Lin, and Winston H. Hsu (2017). "Drone-Based Object Counting by Spatially Regularized Regional Proposal Network". In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4165–4173. URL: https://api.semanticscholar.org/CorpusID:11908837.

Hu, Jie et al. (2017). "Squeeze-and-Excitation Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141. URL: https://api.semanticscholar.org/CorpusID:140309863.

Jiang, Qing et al. (2023). *T-Rex: Counting by Visual Prompting*. DOI: 10.48550/ARXIV.2311.13596. URL: https://arxiv.org/abs/2311.13596.

Jiang, Ruixia, Lin Liu, and Changan Chen (2023). "CLIP-Count: Towards Text-Guided Zero-Shot Object Counting". In: *Proceedings of the 31st ACM International Conference on Multimedia*. URL: https://api.semanticscholar.org/CorpusID:258676543.

Kang, Seunggu et al. (2023). "VLCounter: Text-aware Visual Representation for Zero-Shot Object Counting". In: *AAAI Conference on Artificial Intelligence*. URL: https://api.semanticscholar.org/CorpusID:266573057.

Kirillov, Alexander et al. (2023). "Segment Anything". In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3992–4003. URL: https://api.semanticscholar.org/CorpusID:257952310.

Laradji, Issam H. et al. (2018). "Where Are the Blobs: Counting by Localization with Point Supervision". In: *Lecture Notes in Computer Science*. Springer International Publishing, 560–576. ISBN: 9783030012168. DOI: 10.1007/978-3-030-01216-8_34. URL: http://dx.doi.org/10.1007/978-3-030-01216-8_34.

Lin, Hui, Xiaopeng Hong, and Yabin Wang (2021). "Object Counting: You Only Need to Look at One". In: *ArXiv* abs/2112.05993. URL: https://api.semanticscholar.org/CorpusID:245124135.

Lin, Min, Qiang Chen, and Shuicheng Yan (2013). "Network In Network". In: *CoRR* abs/1312.4400. URL: https://api.semanticscholar.org/CorpusID:16636683.

Lin, Wei et al. (2022). "Scale-Prior Deformable Convolution for Exemplar-Guided Class-Agnostic Counting". In: *British Machine Vision Conference*. URL: https://api.semanticscholar.org/CorpusID:256903360.

Liu, Chang et al. (2022). "CounTR: Transformer-based Generalised Visual Counting". In: *British Machine Vision Conference*.

Loshchilov, Ilya and Frank Hutter (2017). "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. URL: https://api.semanticscholar.org/CorpusID:53592270.

Lu, E., W. Xie, and A. Zisserman (2018). "Class-agnostic Counting". In: *Asian Conference on Computer Vision*.

Nguyen, Thanh Thoi et al. (2022). "Few-shot Object Counting and Detection". In: *ArXiv* abs/2207.10988. URL: https://api.semanticscholar.org/CorpusID:251018420.

Oquab, Maxime et al. (2023). "DINOv2: Learning Robust Visual Features without Supervision". In: *ArXiv* abs/2304.07193. URL: https://api.semanticscholar.org/CorpusID:258170077.

Paszke, Adam et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *ArXiv* abs/1912.01703. URL: https://api.semanticscholar.org/CorpusID:202786778.

Radford, Alec et al. (2021). "Learning Transferable Visual Models From Natural Language Supervision". In: *International Conference on Machine Learning*. URL: https://api.semanticscholar.org/CorpusID:231591445.

Ranjan, V. et al. (2021). "Learning To Count Everything". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, pp. 3393–3402. DOI: 10.1109/CVPR46437.2021.00340. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.00340.

Ranjan, Viresh and Minh Nguyen (Mar. 2023). "Exemplar Free Class Agnostic Counting". In: pp. 71–87. ISBN: 978-3-031-26315-6. DOI: 10.1007/978-3-031-26316-3_5.

Shi, Min et al. (June 2022). "Represent, Compare, and Learn: A Similarity-Aware Framework for Class-Agnostic Counting". In: pp. 9519–9528. DOI: 10.1109/CVPR52688.2022.00931.

Sokhandan, Negin et al. (2020). "A Few-Shot Sequential Approach for Object Counting". In: *ArXiv* abs/2007.01899. URL: https://api.semanticscholar.org/CorpusID:220363833.

Srivastava, Nitish et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *J. Mach. Learn. Res.* 15, pp. 1929–1958. URL: https://api.semanticscholar.org/CorpusID:6844431.

Tan, Mingxing and Quoc V. Le (2019). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *ArXiv* abs/1905.11946. URL: https://api.semanticscholar.org/CorpusID:167217261.

Đukić, Nikola et al. (2023). "A Low-Shot Object Counting Network With Iterative Prototype Adaptation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18872–18881.

Vaswani, Ashish et al. (2017). "Attention is All you Need". In: *Neural Information Processing Systems*. URL: https://api.semanticscholar.org/CorpusID:13756489.

Wan, Jia and Antoni Chan (2019). "Adaptive Density Map Generation for Crowd Counting". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1130–1139. DOI: 10.1109/ICCV.2019.00122.

Wang, Qilong et al. (2019). "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11531–11539. URL: https://api.semanticscholar.org/CorpusID:203902337.

Xu, Jingyi et al. (2023). "Zero-Shot Object Counting". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15548–15557. URL: https://api.semanticscholar.org/CorpusID:257353801.

Yang, Shuo-Diao et al. (2021). "Class-agnostic Few-shot Object Counting". In: *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 869–877. DOI: 10.1109/WACV48630.2021.00091.

Yang, Yifan et al. (2020). "Weakly-Supervised Crowd Counting Learns from Sorting Rather Than Locations". In: *Lecture Notes in Computer Science*. Springer International Publishing, 1–17. ISBN: 9783030585983. DOI: 10.1007/978-3-030-58598-3_1. URL: http://dx.doi.org/10.1007/978-3-030-58598-3_1.

You, Zhiyuan et al. (2022). "Few-shot Object Counting with Similarity-Aware Feature Enhancement". In: *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6304–6313. URL: https://api.semanticscholar.org/CorpusID:247315397.

Zhang, Yingying et al. (2016). "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 589–597. URL: https://api.semanticscholar.org/CorpusID:4545310.