ICCV #20

Few-Shot Pattern Detection via Template Matching and Regression

Anonymous ICCV submission

Paper ID 20

Abstract

We address the problem of few-shot pattern detection, which aims to detect all instances of a given pattern, typically represented by a few exemplars, from an input image. Although similar problems have been studied in few-shot object counting and detection (FSCD), previous methods and their benchmarks have narrowed patterns of interest to object categories and often fail to localize non-object patterns. In this work, we propose a simple yet effective detector based on template matching and regression, dubbed TMR. While previous FSCD methods typically represent given target exemplars as a spatially collapsed prototype, losing their spatial structure, we revisit and refine classic template matching and regression. It effectively preserves and leverages the spatial layout of exemplars through a minimalistic structure with only a few learnable convolutional or projection layers on top of a frozen backbone. We also introduce a new dataset, dubbed RPINE, which covers a wider range of patterns than existing object-centric datasets. Experiments on three benchmarks, RPINE, FSCD-147, FSCD-LVIS, demonstrate that our method outperforms recent state-of-the-art methods, showing an outstanding generalization ability on cross-dataset evaluation.

1. Introduction

Few-shot detection aims to identify target patterns with minimal labeled examples. While significant progress has been made in few-shot object detection [8, 9, 20, 72, 73], most existing methods remain object-centric, focusing primarily on identifying object-level patterns with relatively clear boundaries. However, many real-world applications require detecting arbitrary target patterns that extend beyond objects to include structural, geometric, or abstract patterns across diverse visual data. Despite recent progress based on deep neural networks, current methods still fall short in addressing these broader pattern detection tasks. Furthermore, the object-centric design of conventional few-shot detectors may lead to performance degradation when the target object lacks clear boundaries or when occlusion and deformation cause its boundaries to become indistinct.

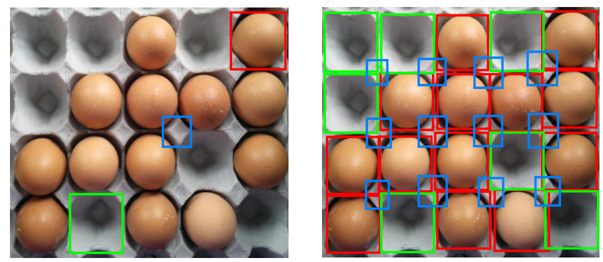


Figure 1. Few-shot pattern detection. Given a few exemplar for each target pattern (left), the task is to detect all matching instances of each pattern (right). This example include non-object patterns (*e.g.*, green and blue) as well as object patterns (*e.g.*, red).

The task of few-shot pattern detection is illustrated in Fig. 1. Recent related research topics for few-shot detection include few-shot counting and detection [51] and few-shot object detection [8]. Both aim to reduce the annotation cost of object detection [13, 36, 61, 68] by learning to detect all instances of given support exemplars. Consequently, many of these methods are heavily biased on object-centric benchmarks prior [7, 17, 34, 50]. In addition, many recent approaches [8, 19, 25, 56, 72, 73] represent the support exemplars as the spatially pooled vector, often named as a prototype [64]. While this pooling strategy is effective for detecting objects, it collapses the geometric properties, such as the shape and structure of the support exemplars. As a consequence, these methods tend to underperform when detecting non-object geometric patterns such as object parts or shape-intensive elements as shown in Fig. 3.

In this work, we propose a simple yet effective few-shot detector for arbitrary patterns, revisiting and refining the classic template matching strategy. The proposed method, dubbed template matching and regression (TMR), is designed to be aware of the structure and shape of given exemplars. Given an input image, TMR first extracts a feature map using a backbone network. It then crops a template feature from the support exemplar's bounding box using a template extraction technique based on RoIAlign [24]. This template is correlated with the image feature map to produce a template matching feature map. Using this correlation map, the model learns bounding box regression param-

FSCD-147

RPINE (Proposed dataset)

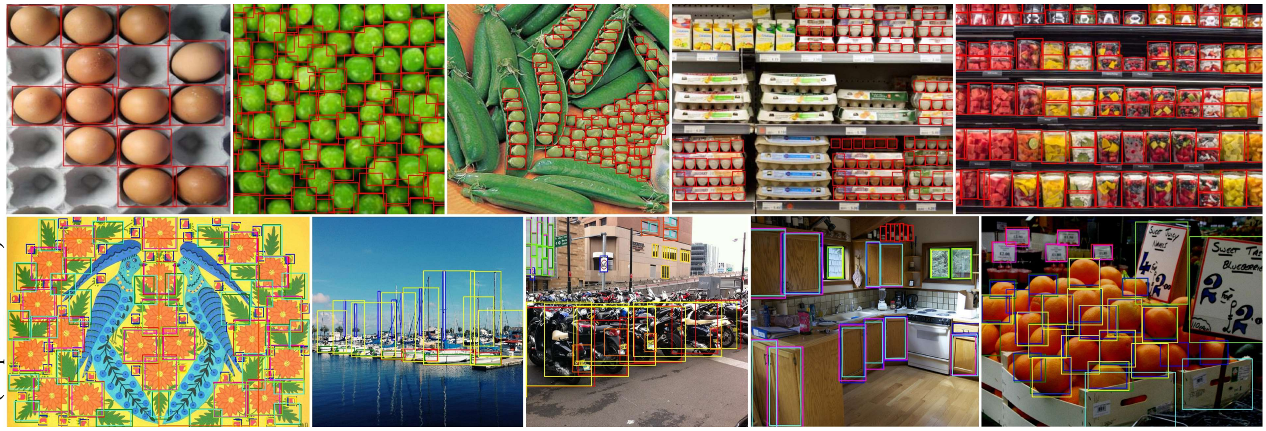
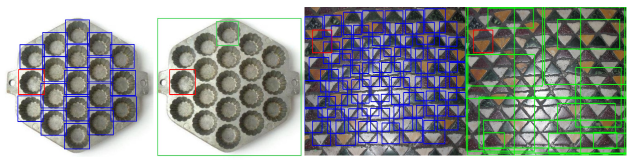


Figure 2. Annotation examples of FSCD-147 [51] and RPINE. FSCD-147 is annotated with the repetitive object-level patterns but disregards the repetition of non-object patterns such as the egg tray pattern under the eggs in the first image. RPINE is annotated with arbitrarily noticed repeated patterns which include non-object patterns and nameless parts of objects. Plus, FSCD-147 is annotated with a single pattern, while RPINE is annotated with all existing repetitions (marked with different colors) recognized by three different annotators.



red: exemplar blue: GT | red: exemplar green: predictions | red: exemplar blue: GT | red: exemplar green: predictions

Figure 3. Few-shot detectors trained with strong object prior are often biased to objects instances and struggle to detect non-objects.

eters to rectify the template box size adaptively. This process, termed template-conditioned regression, enables the model to handle support exemplars of varying sizes more effectively. Notably, TMR consists only of a few 3×3 and linear projections without any complicated modules such as cross-attention, commonly used in prior work [50, 56].

Existing benchmarks (*e.g.*, FSCD-147 [50], FSCD-LVIS [50]) mainly target object-level patterns, limiting the evaluation of general pattern detection. To address this, we introduce a new dataset, Repeated Patterns IN Everywhere (RPINE), which covers diverse repeated patterns in the real world. RPINE contains images with varying degrees of objectness, from well-defined object-level patterns to non-object patterns, all annotated with bounding boxes via crowd-sourcing. Compared to FSCD datasets, RPINE provides broader coverage, including both non-object patterns and nameless parts of objects, as illustrated in Fig. 2.

TMR demonstrates strong performance in detecting repeated patterns, not only on RPINE but also on the FSCD benchmarks, FSCD-147 and FSCD-LVIS [51]. In particular, TMR is especially effective on RPINE, includes diverse patterns with minimal object priors. Notably, our simple architecture contributes to improved generalization across datasets. Our contribution is summarized as follows:

- We generalize the few-shot object counting and detection to a pattern detection task that does not assume objectness in either the target patterns or exemplars.
- We present a simple yet effective pattern detector by refining template matching, which efficiently detects coherent patterns guided by exemplars.
- We introduce a new densely annotated dataset, RPINE, which covers diverse repetitive patterns in the real world, ranging from object-level patterns to non-object patterns.
- TMR not only outperforms the state-of-the-art FSCD models on RPINE and FSCD-LVIS but also achieves strong cross-dataset generalization.

2. Related work

Few-shot object detection (FSOD) aims to detect objects of novel classes using only a few support images of novel classes. Existing methods can be roughly categorized into two groups: finetuning based [9, 16, 67, 71, 72] and meta-learning based [8, 18, 19, 21, 26, 73, 75] approaches. Despite significant progress, finetuning methods require retraining whenever new classes are added. This becomes more severe when low-level semantics repeat, as they are difficult to define as distinct classes, requiring the model to learn each exemplar separately. In contrast, meta-learning methods typically construct prototypes from support images and utilize the prototypes to classify the bounding boxes query images, avoiding fine-tuning through online adaptation. However, while spatially collapsed prototypes may effectively capture object-level repetition, they struggle with non-object patterns that require spatial understanding.

Few-shot counting (FSC) aims to count objects in an image given a few exemplars (typically 1 to 3) from the same image. Previous methods [32, 43, 59, 60, 74] have tackled this problem using density-map regression without bounding box prediction for individual instances. However, relying solely on density maps limits their ability to localize instances precisely. To mitigate this limitation, detection-based FSC, *i.e.*, FSCD methods, have been proposed, *e.g.*, Counting-DETR [51], SAM-C [45], DAVE [55], PseCo [25], and GeCo [56]. However, similar

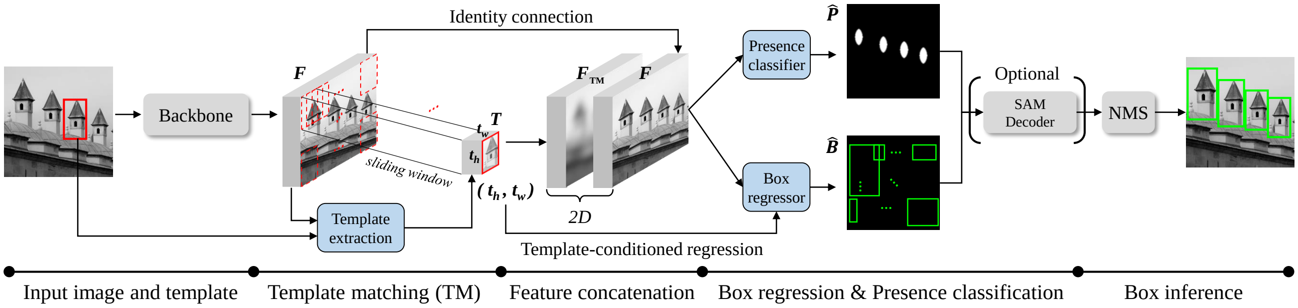


Figure 4. **Template Matching and Regression (TMR).** A template feature map is cropped from the image feature map and then correlated with the image feature map via channel-wise template matching. The TM feature map and the image feature map are concatenated. For each feature map point, the box regressor predicts the shifting & scaling parameters of the template box size, and the presence classifier scores the binary presence map. Both of them consist of a 3×3 convolution and a linear projection without any complicated layers.

to FSOD methods, these models typically generate a prototype by spatially averaging exemplar bounding boxes. This discards the spatial structure of the exemplars, potentially losing important cues and fine-grained details for accurate counting and detection. In contrast, our method preserves the spatial structure of exemplar boxes during matching, enabling more accurate counting and localization of both object and non-object patterns.

Template matching. Template matching [23, 28, 47–49, 69, 70] has been widely used from the beginning of computer vision and pattern recognition and also adopted in convolution-based neural detectors with additional regression [13, 36]. Given a 2D template, template matching identifies the matching region by sliding-window fashion.

Repetitive pattern detection. While humans effortlessly recognize repetitive patterns, detecting them remains a fundamental yet challenging problem in computer vision. Early research focused on geometrically constrained settings where patterns are nearly regular and aligned [15, 22, 33, 54]. Using this near-regularity, early methods assumed a global repetitive structure and discovered the repetition [23, 33, 40, 41, 53]. However, such strong assumptions about geometric regularity led to a new line of research that relaxed these constraints. Later methods [12, 37, 65] defined the smallest repeating unit, called texel [2, 30, 38] and identified all matching subparts based on identified texels. In our context, the texel corresponds to the given exemplars as a user interface to the system.

3. Few-shot pattern detection

Given an input image $\boldsymbol{I} \in \mathbb{R}^{H_0 \times W_0 \times 3}$, the goal of few-shot pattern detection is to predict matching patterns with a given set of support exemplars. With an abuse of notations, the model is given the set of support exemplar, $\mathcal{E} = \{e_1, \cdots, e_{N_s}\}$ and aims to predict the corresponding ground-truth bounding boxes of pattern $\mathcal{B} = \{b_1, \cdots, b_{N_p}\}$, where N_p and N_s denotes the number of ground-truth bounding boxes of the pattern and the number of exemplars (typically referred to as the "shot"), respectively. The input exemplars and the output patterns are both represented as bounding boxes parameterized by the center coordinates and the box size: $b_i, e_i \in \mathbb{R}^4$.

4. Template matching and regression (TMR)

For clarity, we primarily focus on the one-shot, single-scale pattern detection setting, where a single support exemplar is given and a single-resolution image feature map is used. However, TMR can be effortlessly extended to few-shot and multi-scale scenarios, as described in Sec.4.3 and Sec.9.

The overall architecture TMR is illustrated in Fig. 4. The input image $\boldsymbol{I}$ is first encoded by a backbone such as ViT [27] to extract a feature map $\boldsymbol{F} \in \mathbb{R}^{H \times W \times D}$. The template feature $\boldsymbol{T} \in \mathbb{R}^{t_h \times t_w \times D}$ is then cropped by the bounding box size of support exemplar using ROIAlign [24] on the image feature map. In RoIAlign, unlike previous fixed size pooling methods [51, 56], the model adaptively determine the size of $\boldsymbol{T}$ to fit the corresponding size on $\boldsymbol{F}$ by rounding up the size of the support exemplars on $\boldsymbol{F}$ as described in Sec. 8.2. This preserves the spatial alignment between $\boldsymbol{T}$ and $\boldsymbol{F}$ with translation. The image feature map $\boldsymbol{F}$ and template feature $\boldsymbol{T}$ are correlated by template matching (Sec. 4.1), which outputs the template-matching feature $\boldsymbol{F}_{\text{TM}}$. The concatenation of the feature maps $\boldsymbol{F}$, $\boldsymbol{F}_{\text{TM}}$ is fed to the subsequent box prediction module, which consists of a pattern box regressor and a pattern presence classifier. The pattern box regressor predicts the localization bounding boxes: $\hat{\boldsymbol{B}}$, which is parameterized by scaling and shifting factors of the given template size. The pattern presence classifier predicts the presence score of pattern: $\hat{\boldsymbol{P}}$. A box proposal is then generated on each feature map point based on the combination of $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{P}}$ (Sec. 4.2). At inference, bounding boxes with low presence scores are removed by Non-Maximum Suppression (NMS). SAM decoder [27] can be optionally applied for box refinement before NMS.

4.1. Template matching (TM)

We are motivated to encourage detecting non-object patterns given a support exemplar. Non-object patterns often lack high-level semantics yet exhibit low-level structural features, thus, preserving the spatial structure of the support exemplar is crucial. Inspired by traditional template matching [28, 47, 69], we compute the matching score between the image and the support exemplar to detect the locations of the pattern.

Specifically, TM cross-correlates the feature map $\boldsymbol{F}$ and the template feature $\boldsymbol{T}$ by centering $\boldsymbol{T}$ at each (x, y) position in $\boldsymbol{F}$ to obtain the TM feature $\boldsymbol{F}_{\text{TM}}$. The TM feature map then encodes the spatial structure of the template and is obtained as follows:

$$\boldsymbol{F}_{\text{TM}}(x,y) = \frac{1}{t_w t_h}\sum_{x',y'}\boldsymbol{F}(x{+}x'{-}\lfloor\frac{t_w}{2}\rfloor, y{+}y'{-}\lfloor\frac{t_h}{2}\rfloor)\boldsymbol{T}(x',y'), \tag{1}$$

where $\lfloor\cdot\rfloor$ denotes the floor operation used for centering $\boldsymbol{T}$ at each (x, y), and (x', y') ranges within the template coordinates: $\in [0, t_w) \times [0, t_h)$. Note that Eq. 1 computes the correlation by channel-wise multiplication, resulting in $\boldsymbol{F}_{\text{TM}} \in \mathbb{R}^{H\times W\times D}$. Finally, the TM feature $\boldsymbol{F}_{\text{TM}}$ is concatenated with the feature map $\boldsymbol{F}$: $\boldsymbol{F}_{\text{P}} = [\boldsymbol{F}_{\text{TM}}; \boldsymbol{F}] \in \mathbb{R}^{H\times W\times 2D}$. The concatenated feature map $\boldsymbol{F}_{\text{P}}$ is fed to the subsequent box prediction module.

4.2. Box regression

Our box prediction module consists of *a pattern box regressor* and *a pattern presence classifier* following the architecture of the anchor-free detection methods [68].

The pattern box regressor g_{B} consists of a 3×3 convolutional layer followed by a linear layer and predicts the four localization parameters $(\Delta x, \Delta y, \alpha_w, \alpha_h)$ for each feature map point. Unlike the previous work [51, 56] that directly regresses absolute box parameters, our method performs *template-conditioned* regression such that the predicted bounding box regression is parameterized by the template size. Since targets vary in size from small to large, leveraging the template size as a prior and dynamically adjusting it allows the model to better adapt to varying object scales, leading to improved localization accuracy and robustness compared to directly predicting bounding boxes. A predicted bounding box at a feature point (x, y) shifts and scales the template size (t_w, t_h) such as:

$$(x + t_w\Delta x, y + t_h\Delta y, e^{\alpha_w}t_w, e^{\alpha_h}t_h). \tag{2}$$

The pattern presence classifier g_{P} consists of a linear layer and predicts presence scores, which represent the confidence of the predicted bounding boxes at each feature map point. The aforementioned procedure of the box regressor g_{B} and the presence classifier g_{P} is summarized as:

$$\hat{\boldsymbol{B}} = g_{\text{B}}(\boldsymbol{F}_{\text{P}}), \quad \hat{\boldsymbol{B}} \in \mathbb{R}^{H\times W\times 4}, \tag{3}$$

$$\hat{\boldsymbol{P}} = \sigma(g_{\text{P}}(\boldsymbol{F}_{\text{P}})), \quad \hat{\boldsymbol{P}} \in \mathbb{R}^{H\times W\times 1}, \tag{4}$$

where σ denotes the sigmoid function.

4.3. Inference

At inference, we first remove bounding boxes whose presence score is below a threshold τ to filter out low-confidence predictions. Afterward, we can optionally apply box localization refinement via the SAM decoder [27] as often adopted by the FSCD work [25, 45, 56]. Exactly following [25, 56], we input the predicted box coordinates into the SAM prompt encoder to obtain the prompt feature. The obtained prompt feature and the image feature extracted from the SAM backbone are fed to the SAM decoder that slightly refines input box coordinates. To obtain the final box prediction, $\hat{\mathcal{B}}$, we apply NMS on the bounding boxes.

TMR is easily extended for few-shot inference without re-training. When multiple support exemplars are given, we perform the above process for each exemplar individually and then aggregate the results before applying NMS.

4.4. Learning objective

The train loss is composed of $\mathcal{L}_{\text{P}}$, which penalizes the presence score, and $\mathcal{L}_{\text{B}}$, which penalizes the bounding box regression. To calculate the presence loss $\mathcal{L}_{\text{P}}$, we first define the set of coordinates (x, y) that correspond to the ground-truth (GT) locations of the pattern. Specifically, a location (x, y) is set to a GT location if it falls within a margin δ around the center of any GT bounding box of the pattern, as the centers of GT bounding boxes are not strictly localized to a single point:

$$\mathcal{X}_{\text{P}} = \{(x,y) \mid \exists(x_p,y_p) \subset \mathcal{B},\ |x{-}x_p| \leq \delta,\ |y{-}y_p| \leq \delta\}, \tag{5}$$

where (x_p, y_p) is the center point of the GT bounding box in $\mathcal{B}$. Finally, the GT presence label $P(x, y)$ is defined as:

$$P(x,y) = \begin{cases} 1, & \text{if } (x,y) \in \mathcal{X}_{\text{P}}, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where $(x, y) \in [0, W) \times [0, H)$.

To calculate $\mathcal{L}_{\text{B}}$, we match the GT bounding box for (x, y), where the corresponding GT bounding box at (x, y), denoted as $B(x, y)$, is assigned only if $P(x, y) = 1$. With these sets, we can naturally define the classification loss $\mathcal{L}_{\text{P}}$ using the binary cross-entropy function (BCELoss) over all possible coordinates, and the bounding box loss $\mathcal{L}_{\text{B}}$ using the generalized IoU (gIoU) loss [62] over the regression GT bounding box in $\mathcal{X}_{\text{P}}$:

$$\mathcal{L}_{\text{P}} = \sum_{(x,y)}^{\forall(x,y)} \text{BCELoss}(\hat{\boldsymbol{P}}(x,y), \boldsymbol{P}(x,y)), \tag{7}$$

$$\mathcal{L}_{\text{B}} = \sum_{(x,y)}^{(x,y)\in\mathcal{X}_{\text{P}}} \text{gIoULoss}(\hat{\boldsymbol{B}}(x,y), \boldsymbol{B}(x,y)). \tag{8}$$

The overall training loss is the sum of them: $\mathcal{L} = \mathcal{L}_{\text{P}} + \mathcal{L}_{\text{B}}$.

dataset	pattern tiling	repetition	object bias	multi-pattern annotations per image
Wallpaper [39]	near regular [41]	high	low	
Pascal [6], COCO [35]	-	low	high	✓(multi-category)
Wikiart [63], Frieze [10], Counting bench [52]	arbitrary	mid	high	
FSCD-147 [51], FSCD-LVIS [51]	arbitrary	high	high	
RPINE	arbitrary	high	low to high	✓

Table 1. Benchmark dataset comparison of related work. RPINE covers pattern or object repetitions in the real world. We classify 'pattern-tiling' as *near-regular* following [41], *i.e*., elements with an almost periodic lattice with minor variations in shape, color or lighting, and as *arbitrary* otherwise. The 'repetition' is classified based on the average number of repeated same class instances on an image: *low* if ≤ 5, *mid* if ≤ 20, and *high* otherwise. The 'object-bias' is *high* if the dataset annotations correspond to a predefined class set, and *low* otherwise.

Figure 5. TMR predictions (green) with different exemplars (red) from RPINE, which is the only dataset containing multiple patterns for each image among FSCD datasets. GTs are annotated with blue boxes.

5. Proposed dataset: RPINE

Existing benchmarks (*e.g*., FSCD-147 [50], FSCD-LVIS [50]) focus on object-level patterns, limiting their use for general pattern detection. Therefore, we introduce a new pattern dataset, RPINE: Repeated Patterns In Everywhere.

We collect images from the repetitive pattern detection literature and annotate them to contain various repetitive patterns in the wild. Images are collected from FSC-147 [51], FSCD-LVIS [51], Countbench [52], Wikiart [63], Frieze [46], and Wallpaper [39]. Tab. 1 shows where RPINE stands among related datasets. The dataset contains 4,362 images, divided into 3,925 and 435 for training and testing, respectively. As shown in Fig. 5, RPINE is annotated with multiple patterns per image. RPINE is a suitable evaluation benchmark for multi-pattern detection within an image.

In the real world, a pattern cannot be rigorously defined. We thus define the following criteria inspired by the definition of the translation symmetry [66] to minimize the subjective variance among different annotators.

- Number of patterns: max 3 different patterns are annotated per image if the image exhibits multiple patterns.
- Number of pattern instances: has no upperbound.
- Minimum Size: the width and height of a pattern instance must be at least 3% of the shorter side of the image.
- Appearance variance: visually similar patterns with different scales/colors/semantics/rotation angles are annotated as the same pattern.
- Reflection: visually similar patterns but reflection symmetric patterns are annotated in different patterns.
- Occlusion: if visually similar patterns are occluded from each other, the visible parts are annotated.

We ask annotators to carefully draw bounding boxes on the recognized patterns by the above instructions as consistently as they can. Despite of the instruction details, pattern are not pixel-perfectly defined across annotators. Therefore, we assign three individual annotators per image and include all the annotated patterns as ground truth.

6. Experiments

6.1. Dataset and metrics

We evaluate our model on RPINE as well as on the two standard FSCD benchmarks: FSCD-147 [51] and FSCD-LVIS [51]. FSCD-147 contains a total of 6,135 images, with 3,659 for training, 1,286 for validation, and 1,190 for testing. FSCD-LVIS seen-split contains a total of 6,195 images, with 4,000 for training, 1,181 for validation, and 1,014 for testing, covering 372 object categories. FSCD-LVIS unseen-split contains a total of 6,201 images, with 3,959 for training and 2,242 for testing where test-time object categories are not observed during training.

Following the evaluation protocol [51, 56], we report Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for counting. For detection, we report average precision with IoU thresholds of 0.5 and 0.75, denoted as AP50 and AP75, respectively, along with the averaged AP over IoU thresholds from 0.5 to 0.95 in increments of 0.05.

6.2. Implementation details

We use the pre-trained SAM-ViT/H [27] of the patch size 16 as the backbone and set it frozen during training, which returns the $64 \times 64 \times 256$ feature map. Due to the input patchification encapsulated in the ViT backbones [5], we find the receptive field of the raw feature map too coarse to detect small instances. We thus bilinearly interpolate the feature map resolution from 64×64 to 128×128 to produce a higher-resolution correlation map, which enables denser predictions and leads to better performance. (Tab. 11)

The channel dimension is expanded to 512 via a learnable linear projection. TMR is trained with the learning rate of 10^{-4} with AdamW [42] and the batch size of 16 on four Nvidia RTX 3090 GPUs with 24 GB VRAM for 24 hours. The box presence threshold τ for NMS set to 0.4, 0.3 for RPINE and FSCD-147, respectively.

Method	SD	MAE(↓)	RMSE(↓)	AP(↑)	AP50(↑)	AP75(↑)
C-DETR [51]		9.58	21.24	13.88	32.20	10.22
SAM-C [45]	✓	18.77	37.14	18.80	34.04	18.74
PseCo [25]	✓	48.20	88.16	23.18	44.54	21.24
GeCo [56]	✓	9.57	17.07	23.33	45.93	21.19
TMR (ours)		**8.45**	19.87	**33.59**	**64.05**	**30.52**
TMR (ours)	✓	**8.30**	19.40	**29.66**	**58.94**	**25.41**

Table 2. One-shot pattern counting and detection results on the RPINE dataset. SD denotes box refinement with the SAM decoder. All the models are trained by the official code.

Method	Seen		Unseen	
	AP(↑)	AP50(↑)	AP(↑)	AP50(↑)
FSDetView-PB [72]	2.72	7.57	1.03	2.89
AttRPN-PB [8]	4.08	11.15	3.15	7.87
C-DETR [51]	4.92	14.49	3.85	11.28
DAVE [55]	6.75	22.51	4.12	14.16
PseCo [25]	22.37	42.56	-	-
GeCo [56]	-	-	11.47	24.49
TMR (ours)	**27.49**	**48.48**	**22.71**	**39.68**

Table 3. Three-shot counting detection-based methods on the FSCD-LVIS seen and unseen split.

			AP		AP50	
Train	Test	cross-eval	GeCo	TMR	GeCo	TMR
F-147	F-147		**43.42**	44.43	**75.06**	73.83
	F-LVIS$_{seen}$	✓	13.96	**21.25**	25.87	**37.18**
	RPINE	✓	19.47	**26.21**	38.69	**52.01**
RPINE	F-147	✓	36.99	**41.39**	60.38	**69.19**
	F-LVIS$_{seen}$	✓	10.01	**20.92**	17.44	**37.87**
	RPINE		23.33	**29.66**	45.93	**58.94**

Table 4. Cross-dataset comparison of GeCo [56] and TMR, where F-147, F-LVIS indicate FSCD-147 and FSCD-LVIS.

Method	One-shot		Three-shot	
	AP(↑)	AP50(↑)	AP(↑)	AP50(↑)
GeCo [56]	32.71	69.95	32.49	70.51
TMR (ours)	**36.01**	**71.19**	**38.57**	**72.61**

Table 5. Comparison without optional SAM decoder on FSCD147

6.3. Comparison with state-of-the-art methods

RPINE. To demonstrate the effectiveness of our method in counting and detecting non-object patterns, we evaluate it on RPINE. In Tab. 2, TMR surpasses the previous FSCD methods with a large margin. As the previous FSCD methods rely on prototypes for matching, they tend to struggle with non-object patterns that require an understanding of spatial details rather than semantics. In contrast, TMR effectively detects non-object patterns by incorporating spatial details in the template matching process. Note that RPINE is the only dataset equipped with multiple pattern annotations for each image in FSCD. Figure 5 demonstrates the bounding box predictions with different exemplars, where TMR predicts bounding boxes adaptively to the given exemplars. Figure 7 shows the qualitative comparisons with other FSCD methods where TMR accurately localize target patterns.

FSCD-LVIS and FSCD-147. We compare TMR with existing methods that are dedicated to FSCD under the FSCD setting on FSCD-LVIS and FSCD-147. The results in Tab. 3 demonstrate that TMR significantly outperforms previous state-of-the-art approaches. Notably, when evaluated on the unseen split, where test-time object categories are not observed during training, TMR surpasses prior methods. This suggests that TMR is less biased to object semantics during training potentially because TMR leverages the exemplar's spatial structure that generalizes across different categories. In addition to FSCD-LVIS, Tab. 6 compares methods on FSCD-147, where TMR performs on par with the previous methods on both one-shot and three-shot settings.

6.4. Analyses and ablation study

TMR learns with less semantic object bias and generalizes well across datasets. We compare TMR and GeCo [56] in the cross-dataset scenarios by evaluating the trained models on different datasets that are unseen during training. As shown in Tab. 4, *TMR presents overwhelming performances, showing its strong generalization ability.* Specifically, when GeCo is trained on FSCD-147, its performance drops significantly when evaluated on different datasets compared to when tested on FSCD-147 itself. GeCo, like previous FSCD methods, relies on prototypes for both counting and detection. We also suspect this lower generalization ability than ours originates from prototype matching, which is prone to overfitting to the training object semantics. GeCo struggles when evaluated on datasets with different object semantics. In contrast, TMR utilizes the structural information for matching instead of relying on semantic-intensive prototypes and generalizes more effectively on unseen datasets.

SAM decoder is biased to edges. Table 2 shows the negative impact of the optional box refinement using the SAM decoder [27] on RPINE. Figure 6 visualizes two representative examples when the SAM decoder (SD) degrades performance. We observe that SD tends to align box predictions with the nearest edge, which is expected given that SAM is a segmentation model. The edge-sensitive prediction is particularly harmful for non-typical objects, such as objects that are loosely bounded by edges or patterns with many confusing edges. The SD refinement is seemingly good at edge detection for typical object exemplars with clear boundaries, and this is why the existing FSCD methods [25, 45, 56] benefit from adopting SD for box refinement. However, arbitrary patterns are not necessarily bounded by clear edges. Hence, edge-driven refinement may be even harmful as verified on RPINE. We emphasize that SD is an additional box post-processor, and we option-

Method	SAM decoder	One-shot				Three-shot			
		MAE(↓)	RMSE(↓)	AP(↑)	AP50(↑)	MAE(↓)	RMSE(↓)	AP(↑)	AP50(↑)
FSDetView-PB [72]		-	-	-	-	37.54	147.07	13.41	32.99
AttRPN-PB [8]		-	-	-	-	32.42	141.55	20.97	37.19
C-DETR† [51]		16.99	125.22	19.14	47.63	16.79	123.56	22.66	50.57
SAM-C† [45]	✓	33.17	141.77	35.09	56.02	27.97	131.24	27.99	49.17
PseCo [25]	✓	14.86	118.64	41.63	70.87	13.05	112.86	42.98	73.33
DAVE [55]		<u>11.54</u>	86.62	19.46	55.27	<u>10.45</u>	74.51	26.81	62.82
GeCo [56]	✓	**8.10**	<u>60.16</u>	<u>43.11</u>	**74.31**	**7.91**	<u>54.28</u>	<u>43.42</u>	**75.06**
TMR (ours)	✓	11.63	**57.46**	**43.15**	<u>71.55</u>	13.78	**51.87**	**44.43**	<u>73.83</u>

Table 6. Comparison with few-shot detection and counting methods on FSCD-147. The one-shot performance of the models with † are evaluated using the official code.

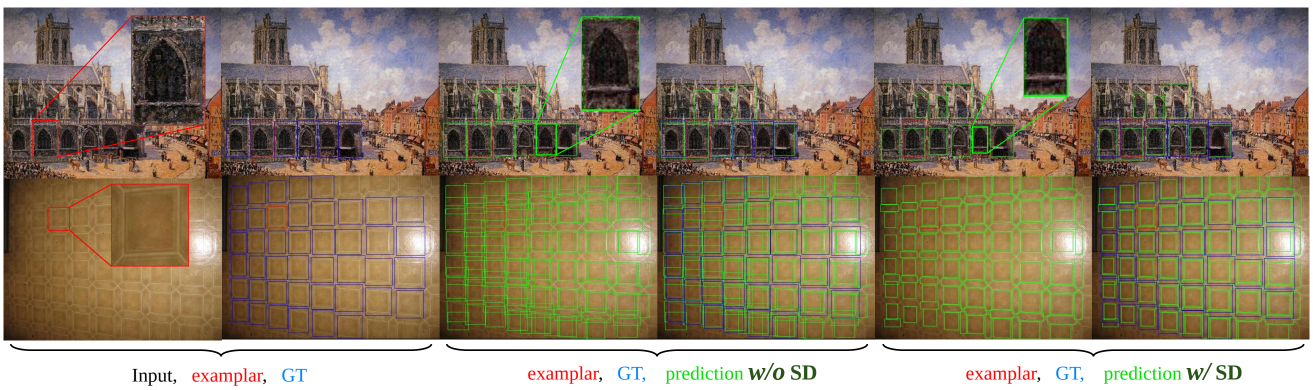


Figure 6. TMR without *vs.* with SAM decoder [27] (*w/o* SD *vs.* *w/* SD) for box refinement. SAM decoder tends to blindly attach the prediction box to the closest edges. The edge-sensitive prediction is harmful for non-objects, objects that have loose boundaries, or patterns with many confusing edges. The exemplar in the first image loosely contains an arched window, but the SAM decoder shrinks the box toward the window boundaries.

	feature $\boldsymbol{F}_{\text{P}}$	RPINE		FSCD-147	
		AP	AP50	AP	AP50
(a)	$\boldsymbol{F}$	11.44	24.02	20.95	55.59
(b)	$\boldsymbol{F}_{\text{TM}}$	32.55	57.43	31.96	63.03
(c)	$\boldsymbol{F} \oplus \boldsymbol{F}_{\text{TM-cos}}$	29.74	57.85	24.73	60.29
(d)	$\boldsymbol{F} \oplus \boldsymbol{F}_{\text{PM}}$	20.94	47.96	28.91	66.93
(e)	$\boldsymbol{F} \oplus \boldsymbol{F}_{\text{TM}}$	**33.59**	**64.05**	**36.01**	**71.19**

Table 7. Effect of template matching features (Sec. 4.1). $\boldsymbol{F}_{\text{PM}}$ is the correlation with average-pooled prototype. $\boldsymbol{F}_{\text{TM-cos}}$ is the cosine similarity of the template and image feature.

	box regression		RPINE		FSCD-147	
	shift	scale	AP	AP50	AP	AP50
(a)	$(0,0)$	$(0,0)$	26.20	55.20	22.81	57.30
(b)	$(\Delta x, \Delta y)$	$(e^{\alpha_w}, e^{\alpha_h})$	23.88	51.25	17.01	55.14
(c)	$(\Delta x, \Delta y)$	$(e^{\alpha_w} t_w, e^{\alpha_h} t_h)$	31.29	59.52	35.08	70.23
(d)	$(t_w \Delta x, t_h \Delta y)$	$(e^{\alpha_w} t_w, e^{\alpha_h} t_h)$	**33.59**	**64.05**	**36.01**	**71.19**

Table 8. Effect of box regression methods (Sec. 4.2)

ally add SD to TMR to compare with the SD-based state of the arts. Table 5 compares models by taking out SD, where TMR shows powerful performance.

Effectiveness of template matching features. We verify the effectiveness of 2D template matching, which preserves the spatial structure of exemplar bounding boxes for correlation. We experiment with different input feature $\boldsymbol{F}_{\text{P}}$ to the box regression module. Table 7 compares our final model (e) and its variants. The lower bound model (a) observes zero information on the exemplar. Model (b) shows that correlating with 2D templates is greatly beneficial. The final model (e) concatenates the template-matching and the image feature, providing the correlation and appearance information for pattern matching regression. The models (c, d) replace the variants of the template matching feature $\boldsymbol{F}_{\text{TM}}$ with something else. Model (c) replaces the channel-wise correlation of (e) to cosine similarity, *i.e.*, $\boldsymbol{F}_{\text{TM-cos}}$, showing that retaining the channel dimension helps. Model (d) first performs average-pooling of the template and uses the prototype matching feature, $\boldsymbol{F}_{\text{PM}}$, for correlation. Note that this average-pooled feature is often used in previous FSC methods [25, 56]. The result shows that preserving the spatial structure of exemplar bounding boxes is more effective. The representative failure cases are shown in Fig. 9 in supp. material. The pooled prototype feature loses the geometric layout of the template and struggle to detect patterns where geometric clues are crucial. Comparing (d) and (e) verifies our hypothesis that prototype-based matching relies on the object priors; Model (d) drops more significantly on RPINE, which is less object-centric than FSCD-147.

Effectiveness of template-conditioned regression. Table 8 compares different attempts on the pattern box re-

Figure 7. Qualitative comparison with the state-of-the-art models on RPINE (the first two images) and FSCD-147 (the last two images). More visualization examples on three datasets are included in the supplementary materials Figs 14-13.

Method	Trainable params	Total params	FLOPS
PseCo [25]	56.99M	0.70B	5.08T
GeCo [56]	7.98M	0.65B	4.72T
TMR (ours)	19.01M	0.66B	3.04T

Table 9. Comparison on the computational complexity

gression and validates the effectiveness of our template-conditioned box regression method. As shown in (b), simply performing box regression without considering the template size results in poor localization. Notably, it even performs worse than (a), which directly uses the template bounding box without any regression. These results demonstrate that incorporating the template size in both shift and width/height scaling is crucial for accurate localization. Thus, when we incorporated the template size in predicting the width and height, there was a significant performance gain shown in (c). Additionally, using the template size for scaling further improved performance shown in (d), demonstrating the effectiveness of template-conditioned regression in detecting the given pattern.

Comparison on computational complexity. Table 9 compares the complexity of TMR with state-of-the-art FSCD methods, demonstrating its efficiency and effectiveness. Compared to PseCo [25], which introduces a large number of trainable parameters and FLOPs, TMR introduces only about 19M trainable parameters, which is detailed in Tab. 10. Although TMR has more trainable parameters than GeCo [56], its total parameter count remains comparable. Thanks to the simple architecture, TMR is significantly efficient in terms of FLOPs (3.04T), which is notably lower than PseCo (5.08T) and GeCo (4.72T). This reduction in computational cost not only enhances training efficiency but also results in faster inference. In contrast, the high FLOPs of PseCo and GeCo contribute to longer inference, making them less practical for real-time applications.

7. Discussion

Conclusion. We have proposed a simple template-matching based method for few-shot pattern detection. We also introduce a new dataset with bounding box annotations that covers various patterns around the world across non-object patterns to typical objects from various image domains from nature and human-made products.

Discussion. We observed that the widely used SAM decoder improves our performance by 7% AP points on FSCD-147 by leveraging object-level edge priors, but it degrades performance on RPINE, which contains more general patterns. This suggests a promising future direction that combines the powerful pre-trained knowledge of foundation models with detection mechanisms less dependent on object-level edge priors. In parallel, exploring a lightweight, pattern-specific architecture that efficiently captures fine-grained repetitive structures without relying on object semantics could further improve generalization and efficiency.

References

[1] Rossella Aversa, Mohammad Hadi Modarres, Stefano Cozzini, Regina Ciancio, and Alberto Chiusole. The first annotated set of scanning electron microscopy images for nanoscience. Scientific data, 5(1):1–10, 2018. 2, 3

[2] Yunliang Cai and George Baciu. Translation symmetry detection: A repetitive pattern analysis approach. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 223–228, 2013. 3

[3] Cheng Chen, Wei Wang, John A Ozolek, and Gustavo K Rohde. A flexible and robust approach for segmenting cell nuclei from 2d microscopy images using supervised learning and template matching. Cytometry Part A, 83(5):495–507, 2013. 3

[4] Taco S Cohen and Max Welling. Steerable cnns. arXiv preprint arXiv:1612.08498, 2016. 3

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020. 5

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88:303–338, 2010. 5

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. IJCV, 88:303–338, 2010. 1

[8] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4013–4022, 2020. 1, 2, 6, 7

[9] Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4527–4536, 2021. 1, 2

[10] Christopher Funk, Seungkyu Lee, Martin R Oswald, Stavros Tsogkas, Wei Shen, Andrea Cohen, Sven Dickinson, and Yanxi Liu. 2017 iccv challenge: Detecting symmetry in the wild. In Proceedings of the IEEE international conference on computer vision workshops, pages 1692–1701, 2017. 5

[11] Hiroyasu Furukawa, Kyle E Cordova, Michael O'Keeffe, and Omar M Yaghi. The chemistry and applications of metal-organic frameworks. Science, 341(6149):1230444, 2013. 3

[12] Jizhou Gao, Yin Hu, Jinze Liu, and Ruigang Yang. Unsupervised learning of high-order structural semantics from images. In ICCV, pages 2122–2129. IEEE, 2009. 3

[13] Ross Girshick. Fast r-cnn. In ICCV, pages 1440–1448, 2015. 1, 3

[14] Alice L Givan. Flow cytometry: an introduction. Flow cytometry protocols, pages 1–29, 2011. 3

[15] Branko Grünbaum and Geoffrey Colin Shephard. Tilings and patterns. Courier Dover Publications, 1987. 3

[16] Karim Guirguis, Johannes Meier, George Eskandar, Matthias Kayser, Bin Yang, and Jürgen Beyerer. Niff: Alleviating forgetting in generalized few-shot object detection via neural instance feature forging. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 24193–24202, 2023. 2

[17] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In CVPR, pages 5356–5364, 2019. 1

[18] Guangxing Han, Yicheng He, Shiyuan Huang, Jiawei Ma, and Shih-Fu Chang. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3263–3272, 2021. 2

[19] Guangxing Han, Shiyuan Huang, Jiawei Ma, Yicheng He, and Shih-Fu Chang. Meta faster r-cnn: Towards accurate few-shot object detection with attentive feature alignment. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 780–789, 2022. 1, 2

[20] Guangxing Han, Shiyuan Huang, Jiawei Ma, Yicheng He, and Shih-Fu Chang. Meta faster r-cnn: Towards accurate few-shot object detection with attentive feature alignment. In AAAI, pages 780–789, 2022. 1

[21] Guangxing Han, Jiawei Ma, Shiyuan Huang, Long Chen, and Shih-Fu Chang. Few-shot object detection with fully cross-transformer. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 5321–5330, 2022. 2

[22] Junwei Han, Stephen J McKenna, and Ruixuan Wang. Regular texture analysis as statistical model selection. In ECCV, pages 242–255. Springer, 2008. 3

[23] James Hays, Marius Leordeanu, Alexei A Efros, and Yanxi Liu. Discovering texture regularity as a higher-order correspondence problem. In Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II 9, pages 522–535. Springer, 2006. 3

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In ICCV, pages 2961–2969, 2017. 1, 3

[25] Zhizhong Huang, Mingliang Dai, Yi Zhang, Junping Zhang, and Hongming Shan. Point segment and count: A generalized framework for object counting. In CVPR, 2024. 1, 2, 4, 6, 7, 8, 3

[26] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In Proceedings of the IEEE/CVF international conference on computer vision, pages 8420–8429, 2019. 2

[27] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In ICCV, 2023. 3, 4, 5, 6, 7, 2

[28] Simon Korman, Daniel Reichman, Gilad Tsur, and Shai Avidan. Fast-match: Fast affine template matching. In CVPR, pages 2331–2338, 2013. 3, 4

[29] Minji Lee, Hyunju Noh, and Youngsuk Kim. Diamidocarbene-derived palladium and nickel–sulfur

clusters. Chemical Communications, 60(94):13867–13870, 2024. 3

[30] Thomas Leung and Jitendra Malik. Detecting, localizing and grouping repeated scene elements from an image. In ECCV. Springer, 1996. 3

[31] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection, 2022. 2

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

[33] Hsin-Chih Lin, Ling-Ling Wang, and Shi-Nine Yang. Extracting periodicity of a regular texture based on autocorrelation functions. Pattern recognition letters, 18(5):433–443, 1997. 3

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In ECCV, pages 740–755. Springer, 2014. 1

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13, pages 740–755. Springer, 2014. 5

[36] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In CVPR, pages 2117–2125, 2017. 1, 3

[37] Wen-Chieh Lin and Yanxi Liu. Tracking dynamic near-regular texture under occlusion and rapid movements. In ECCV, pages 44–55. Springer, 2006. 3

[38] Jingchen Liu and Yanxi Liu. Grasp recurring patterns from a single view. In CVPR, pages 2003–2010, 2013. 3

[39] Jingchen Liu, George Slota, Gang Zheng, Zhaohui Wu, Minwoo Park, Seungkyu Lee, Ingmar Rauschert, and Yanxi Liu. Symmetry detection from realworld images competition 2013: Summary and results. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 200–205, 2013. 5

[40] Siying Liu, Tian-Tsong Ng, Kalyan Sunkavalli, Minh N Do, Eli Shechtman, and Nathan Carr. Patchmatch-based automatic lattice detection for near-regular textures. In Proceedings of the IEEE international conference on computer vision, pages 181–189, 2015. 3

[41] Yanxi Liu, Wen-Chieh Lin, and James Hays. Near-regular texture analysis and manipulation. ACM Transactions on Graphics (ToG), 23(3):368–376, 2004. 3, 5

[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 5

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

[44] Bronwyn A Lucas, Benjamin A Himes, Liang Xue, Timothy Grant, Julia Mahamid, and Nikolaus Grigorieff. Locating macromolecular assemblies in cells by 2d template matching with cistem. Elife, 10:e68946, 2021. 3

[45] Zhiheng Ma, Xiaopeng Hong, and Qinnan Shangguan. Can sam count anything? an empirical study on sam counting, 2023. 2, 4, 6, 7

[46] Massimiliano Mancini, Elisa Ricci, Barbara Caputo, and Samuel Rota Bulo. Adding new tasks to a single network with weight transformations using binary masks. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, pages 0–0, 2018. 5

[47] Giovanni Marola. Using symmetry for detecting and locating objects in a picture. Computer Vision, Graphics, and Image Processing, 46(2):179–195, 1989. 3, 4

[48] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. Image and vision computing, 22(10):761–767, 2004.

[49] Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio. Example-based object detection in images by components. IEEE TPAMI, 23(4):349–361, 2001. 3

[50] Thanh Nguyen, Chau Pham, Khoi Nguyen, and Minh Hoai. Few-shot object counting and detection. In ECCV. Springer, 2022. 1, 2, 5

[51] Thanh Nguyen, Chau Pham, Khoi Nguyen, and Minh Hoai. Few-shot object counting and detection. In European Conference on Computer Vision, pages 348–365. Springer, 2022. 1, 2, 3, 4, 5, 6, 7

[52] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In ICCV, pages 3170–3180, 2023. 5

[53] Minwoo Park, Kyle Brocklehurst, Robert T Collins, and Yanxi Liu. Deformed lattice detection in real-world images using mean-shift belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(10):1804–1816, 2009. 3

[54] Mark Pauly, Niloy J Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J Guibas. Discovering structural regularity in 3d geometry. In ACM SIGGRAPH 2008 papers, pages 1–11, 2008. 3

[55] 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, pages 23293–23302, 2024. 2, 6, 7

[56] Jer Pelhan, Alan Lukezic, Vitjan Zavrtanik, and Matej Kristan. A novel unified architecture for low-shot counting by detection and segmentation. Adv. Neural Inform. Process. Syst., 2025. 1, 2, 3, 4, 5, 6, 7, 8

[57] Julien Picot, Coralie L Guerin, Caroline Le Van Kim, and Chantal M Boulanger. Flow cytometry: retrospective, fundamentals and recent instrumentation. Cytotechnology, 64: 109–130, 2012. 3

[58] Muhammad Farrukh Qureshi, Zohaib Mushtaq, Muhammad Zia Ur Rehman, and Ernest Nlandu Kamavuako. E2cnn: An efficient concatenated cnn for classification of surface emg extracted from upper limb. IEEE Sensors Journal, 23(8): 8989–8996, 2023. 3

[59] Viresh Ranjan and Minh Hoai. Vicinal counting networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4221–4230, 2022. 2

[60] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3394–3403, 2021. 2, 1

[61] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In CVPR, pages 779–788, 2016. 1

[62] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 658–666, 2019. 4

[63] Babak Saleh and Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. arXiv preprint arXiv:1505.00855, 2015. 5

[64] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. NeurIPS, 30, 2017. 1

[65] Luciano Spinello, Rudolph Triebel, Dizan Vasquez, Kai O Arras, and Roland Siegwart. Exploiting repetitive object patterns for model compression and completion. In ECCV, pages 296–309. Springer, 2010. 3

[66] V.J. Stenger. Timeless Reality: Symetry, Simplicity, and Multiple Universes. Prometheus, 2009. 5

[67] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 7352–7362, 2021. 2

[68] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In ICCV, pages 9627–9636, 2019. 1, 4

[69] Du-Ming Tsai and Cheng-Huei Chiang. Rotation-invariant pattern matching using wavelet decomposition. Pattern Recognition Letters, 23(1-3):191–201, 2002. 3, 4

[70] Andreas Turina, Tinne Tuytelaars, and Luc Van Gool. Efficient grouping under perspective skew. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, pages I–I. IEEE, 2001. 3

[71] Xin Wang, Thomas Huang, Joseph Gonzalez, Trevor Darrell, and Fisher Yu. Frustratingly simple few-shot object detection. In International Conference on Machine Learning, pages 9919–9928. PMLR, 2020. 2

[72] Yang Xiao, Vincent Lepetit, and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. IEEE transactions on pattern analysis and machine intelligence, 45(3):3090–3106, 2022. 1, 2, 6, 7

[73] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In Proceedings of the IEEE/CVF international conference on computer vision, pages 9577–9586, 2019. 1, 2

[74] Zhiyuan You, Kai Yang, Wenhan Luo, Xin Lu, Lei Cui, and Xinyi Le. Iterative correlation-based feature refinement for few-shot counting. arXiv preprint arXiv:2201.08959, 4, 2022. 2

[75] Xinyu Zhang, Yuhan Liu, Yuting Wang, and Abdeslam Boularias. Detect everything with few examples. arXiv preprint arXiv:2309.12969, 2023. 2