Few-Shot Pattern Detection via Template Matching and Regression

Supplementary Material

In this supplementary material, we provide additional experimental results and analysis to support our method, including qualitative results.

8. Additional details

8.1. Detailed model architecture

We provide the details of the model architecture in Tab. 10. We design our model architecture to be as simple as possible, and there are only 6 learnable layers in total.

module	structure	# params.
backbone projection	linear(in=256, out=512)	0.13M
$\boldsymbol{F}_{\text{TM}}$ scaler	nn.Parameter	1
box regressor	conv(k=(3, 3), in=1024, out=1024)	9.44M
	LeakyReLU	0
	linear(in=1024, out=4)	4096
presence classifier	conv(k=(3, 3), in=1024, out=1024)	9.44M
	LeakyReLU	0
	linear(in=1024, out=1)	1024

Table 10. The total learnable layers in TMR. We exclude the feature backbone parameters that are frozen. The "k" in the table denotes the 2D convolution kernel size.

8.2. Template extraction details

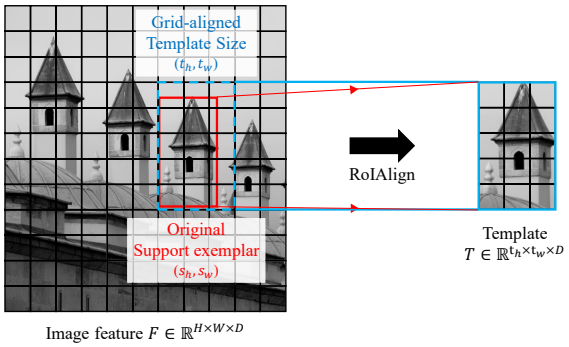


Figure 9. Using RoIAlign, the template $\boldsymbol{T}$ (blue box) is extracted by adaptively determining its size (t_h, t_w) (blue dashed box) to fully cover the support exemplar's region (s_h, s_w) (red box) on $\boldsymbol{F}$, which preserves spatial alignment between $\boldsymbol{F}$ and $\boldsymbol{T}$.

Prior methods typically use Global Average Pooling or RoIAlign to produce fixed-size prototypes. However, these approaches can lead to spatial misalignment between the feature map $\boldsymbol{F}$ and the template $\boldsymbol{T}$, which degrades template matching performance. As shown in Fig. 9, we address this issue by adaptively determining the template size (t_h, t_w) based on the support exemplar's region (s_h, s_w) on $\boldsymbol{F}$ (red box), rounding it up to the smallest grid-aligned region that fully contains the support exemplar's area (blue dashed box). Using this size, we apply RoIAlign to extract a spatially aligned template (blue box), enabling more precise and consistent template matching.

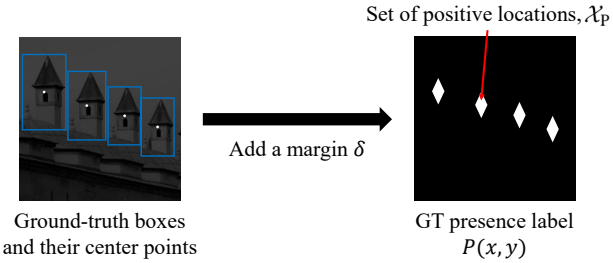


Figure 10. Definition of $\mathcal{X}_{\text{P}}$ using GT centers and margin δ.

8.3. Definition of the extended center point set

To avoid supervising the presence prediction with only a single pixel at the ground-truth center, we define $\mathcal{X}_{\text{P}}$ as a set of extended center point within a margin δ around each ground-truth center point $(x_{\text{c}}, y_{\text{c}})$.

A location (x, y) is considered positive if it falls within this margin region around the center of any ground-truth bounding box. As illustrated in Fig. 10, $\mathcal{X}_{\text{P}}$ is defined as follows:

$$\mathcal{X}_{\text{P}} = \left\{ (x, y) \;\middle|\; \forall (x_{\text{c}}, y_{\text{c}}, w, h) \in \mathcal{B},\ \frac{|x_{\text{c}} - x|}{w} + \frac{|y_{\text{c}} - y|}{h} \leq \delta \right\}. \tag{7}$$

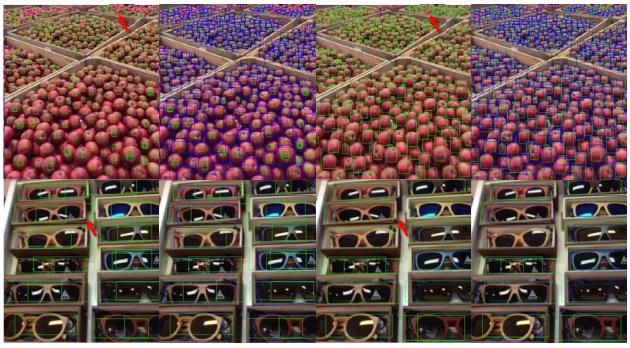
Here, $\mathcal{B}$ denotes the set of ground-truth boxes; $(x_{\text{c}}, y_{\text{c}})$ represents the center coordinates, and (w, h) the width and height of each ground-truth box. The resulting shape of $\mathcal{X}_{\text{P}}$ forms a rhombus centered at each ground-truth location. We fix $\delta = 0.33$ in all experiments.

8.4. Dataset details

FSCD-147 extends FSC-147 [56] dataset, which includes only dot annotations for objects, to include bounding box annotations. It covers 147 object categories with additional bounding box annotation. **FSCD-LVIS** includes more complex scenes with multiple object classes, each containing multiple instances, compared to FSCD-147, where each image has a relatively simple scene. Regardless, FSCD-LVIS still uses a single pattern per image unlike RPINE that is annotated with multiple existing pattern classes.

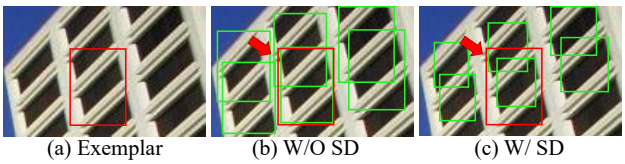
9. Additional experimental details

Qualitative analysis of prototype matching failures. Fig. 11 shows failure cases of the prototype matching



prototype matching $[\boldsymbol{F}; \boldsymbol{F}_{\text{PM}}]$ template matching $[\boldsymbol{F}; \boldsymbol{F}_{\text{TM}}]$

Figure 11. Noticeable failure cases of prototype matching. Prototypes collapse the geometric layout of the exemplars, being especially vulnerable for localizing instances among dense repetition or patterns including sub-patterns.



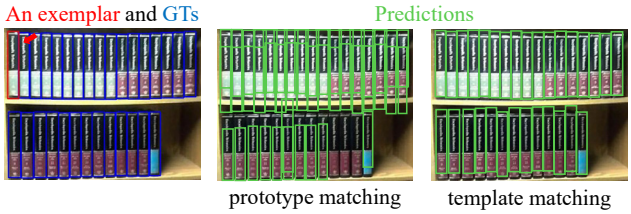
(a) Exemplar (b) W/O SD (c) W/ SD

Figure 12. Additional failure case of SAM decoder.

method compared to our template matching approach. Fig. 13 provides a detailed comparison in terms of box regression (a), matching feature maps (b), and presence scores (c). In both Fig. 11 and Fig. 13 (a), prototype-matching models often produces inaccurate bounding boxes that fail to tightly enclose the target pattern.

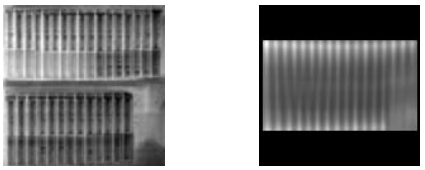
Fig. 13 (b) and (c) further illustrate the differences in feature maps and predicted presence scores, respectively. Prototype feature maps ($\boldsymbol{F}_{\text{PM}}$) highlight regions with similar semantics (e.g., edges or colored bends of a book) while ignoring the exemplar's spatial structure, making it difficult to localize the center of the target pattern. In contrast, template matching feature maps ($\boldsymbol{F}_{\text{TM}}$) preserve spatial structure and clearly emphasize the central region of the target pattern, enabling more precise localization. Consistent with this, the predicted presence score maps in Fig. 13 (c) show that prototype-based scores often activate spatially misaligned but semantically related regions, while template-based scores focus accurately on the true target center. These observations highlight the limitations of prototype matching in capturing spatial structure and demonstrate the effectiveness of template matching for precise localization.

Failure cases of TMR. We analyze the failure cases of TMR, as shown in Fig. 14. In the first row of Fig. 14, TMR often fails to detect highly crowded patterns when the support exemplar is extremely small. In the second row, the model struggles with highly textured patterns that exhibit large variations in texture appearance. These examples suggest potential directions for improvement, such as incorporating multi-scale representations with much higher resolution features and designing pattern-specific backbones to reduce noise caused by texture variation.



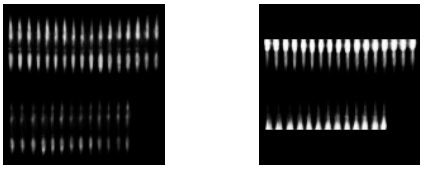
prototype matching template matching

(a) Qualitative comparison: prototype matching vs. template matching



prototype matching template matching

(b) Visualization of the matching feature maps ($\boldsymbol{F}_{\text{PM}}$ vs. $\boldsymbol{F}_{\text{TM}}$)



prototype matching template matching

(c) Visualization of the predicted presence scores ($\hat{\boldsymbol{P}}$)

Figure 13. Comparison between prototype matching and template matching.

Additional analysis of SAM decoder's (SD) failures. To support our main analysis in Sec.6.4, we provide additional qualitative example highlighting the limitations of the SAM decoder (SD) in handling non-object patterns. As shown in Fig.12, although the exemplar includes both the window and decoration around the window sills, the refined prediction closely aligns with the black window frames, missing the broader structure present in the exemplar. This edge-sensitive behavior of SD is consistent with the findings in the Sec.7.2 of the SAM paper [25], which reports that SD produces high-recall edge maps even without explicit edge supervision.

Qualitative results on FSCD-147, FSCD-LVIS and RPINE. We provide additional qualitative results on the FSCD-147, FSCD-LVIS and RPINE dataset in Figs. 17, Figs. 16 and 18 to show the model's effectiveness. As shown in Fig. 17, our method effectively detects the given exemplar. For instance, in the first row, TMR successfully identifies all instances of the given exemplar, whereas other state-of-the-art models either fail to detect some instances or produce false positives. Furthermore, TMR successfully detects non-object patterns, as shown in the penultimate row of Fig. 18.

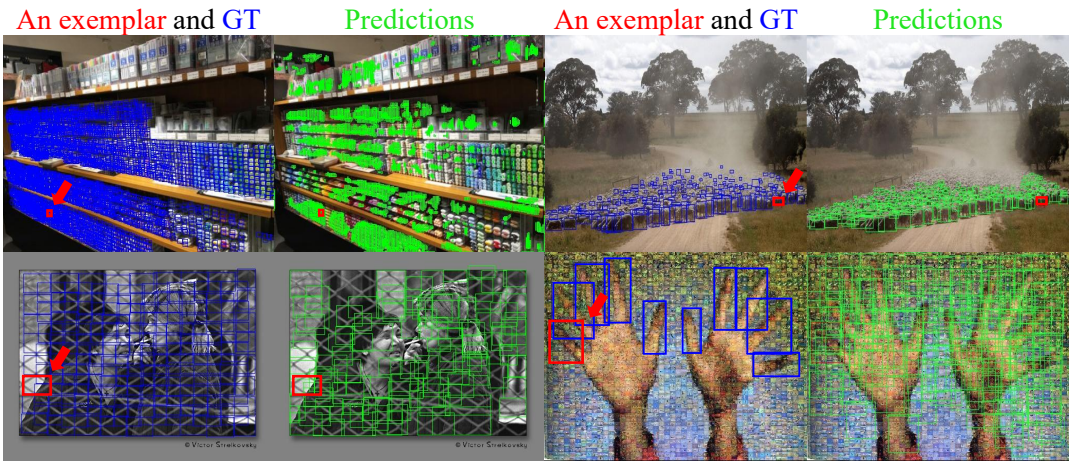


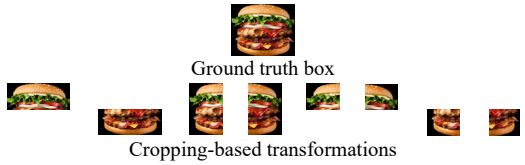
Figure 14. Qualitative analysis of failure cases in TMR.

Type	Scales			AP	AP50	AP75
	32^2	64^2	128^2			
Single-Scale		✓		27.49	56.15	23.25
			✓	33.59	64.05	30.52
Multi-Scales	✓		✓	34.03	64.86	31.66
		✓	✓	34.78	66.71	31.51
	✓	✓	✓	**35.41**	**66.88**	**32.52**

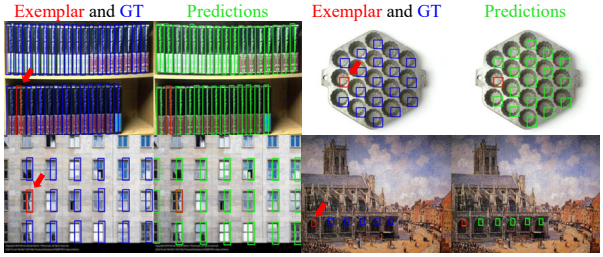
Table 11. Multi-scales experiment of TMR on RPINE. The gray-shaded row indicates the default single-scale configuration described in Sec. 4, where only a single feature scale is used. For the single-scale setting, upscaling the feature map from 64×64 to 128×128 improves performance, as the resulting higher-resolution correlation map enables denser predictions.

Multi-scale extension. TMR can be naturally extended to multi-scale prediction by incorporating a multi-scale architecture, such as ViTDet [29]. Following a similar approach to the few-shot extension in Sec. 4.3, we first extract feature maps at multiple scales and independently apply the prediction process to each scale. The resulting predictions are then aggregated and filtered using Non-Maximum Suppression (NMS) to remove duplicates across scales. As shown in Tab. 11, leveraging multi-scale features yields further performance improvements, demonstrating the benefit of scale-aware detection in capturing pattern instances of varying sizes. For a fair comparison with prior state-of-the-art methods [23, 52], we use the single-scale setting in all experiments, except for the multi-scale experiment reported in Tab. 11.

RPINE-edgeless. To evaluate the model under the minimal assumptions of the object-level edge prior, we augment the RPINE dataset via bounding box transformation. Given a ground-truth bounding box, we apply eight types of cropping-based transformations: left half, right half, top half, bottom half, top-left corner, top-right corner, bottom-left corner, and bottom-right corner, as illustrated in Fig. 15 (a). Based on this, we construct the RPINE-edgeless dataset



(a) Example of cropping-based transformations



(b) Qualitative results of TMR w/o SD on RPINE-edgeless dataset

Figure 15. RPINE-edgeless examples.

Method	SD	AP(↑)	AP50(↑)	AP75(↑)
GeCo [52]	✓	12.53	30.96	8.55
TMR (ours)		**33.25**	**67.63**	**28.22**
TMR (ours)	✓	**17.99**	**46.31**	**10.89**

Table 12. One-shot pattern detection results on the RPINE-edgeless dataset.

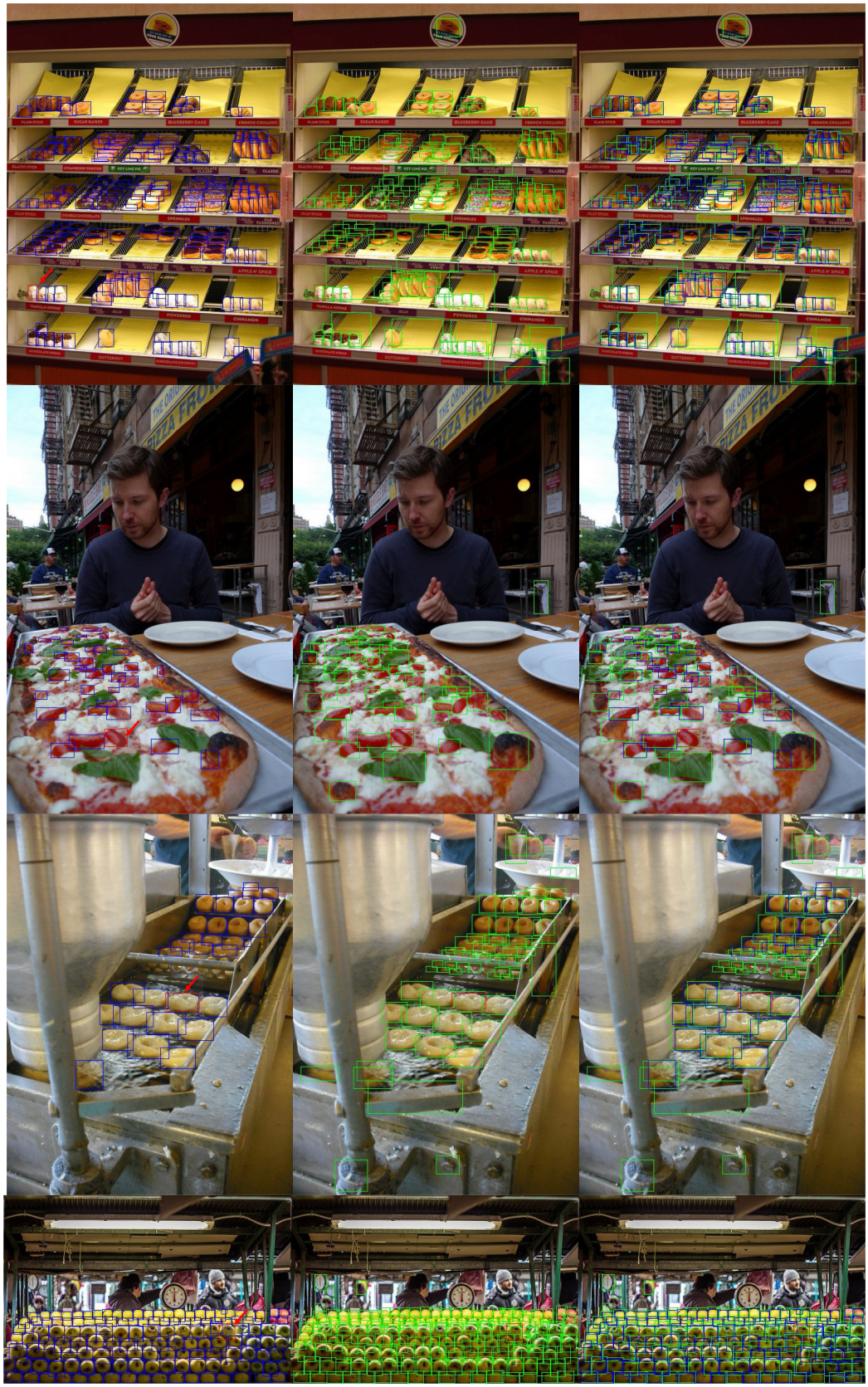
(Fig. 15 (b)), which contains 13,772 training samples and 1,402 validation samples. As shown in Tab. 12, we compare TMR with GeCo, and TMR demonstrates strong performance. However, we also observe that when using SD, the performance drops significantly, which aligns with our claim in Sec. 6.4.

10. Future work

Rotation invariance. Although TMR effectively handles scale variations, achieving rotation invariance remains a challenging problem, as observed in prior approaches as well [23, 52]. This limitation could be further mitigated by incorporating rotation-invariant data augmentation or

adopting rotation-equivariant architectures [4, 54].

Applications. The proposed template-matching based detection framework is potentially useful for detecting low-semantic, user-defined patterns. One interdisciplinary application is flow cytometry [13, 53], which analyzes the physical and chemical characteristics of cell or particle populations [3, 41], such as in cell counting tasks. Since repetitive patterns are a fundamental component of many natural and artificial structures [10, 27], the framework could be extended to broader applications in real-world settings, such as agricultural or industrial vision. A thorough investigation of these directions is beyond the scope of this study and is left for future work.



An exemplar and GT TMR (ours) TMR (ours) and GT

Figure 16. Additional qualitative results on the FSCD-LVIS dataset.

An exemplar and GT | TMR (ours) | SAM-C (Ma *et al.*) | PseCo (Huang *et al.*) | GeCo (Pelhan *et al.*)

Figure 17. Additional qualitative results on the FSCD-147 dataset.

An exemplar and GT | TMR (ours) | SAM-C (Ma *et al.*) | PseCo (Huang *et al.*) | GeCo (Pelhan *et al.*)

Figure 18. Additional qualitative results on the RPINE dataset.