Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence
Canadian AI 2026
25–29 May 2026, Vancouver, BC

Enhancing Thermal Image Object Detection using Spatial Edge-aware Attention and Self-supervision Pretext

Gaeul Han[†*], Thangarajah Akilan[‡*]
[†] Department of Electrical & Computer Engineering, Lakehead University
[‡] Department of Software Engineering, Lakehead University

Abstract

Thermal cameras offer robust sensing for object detection in low-visibility driving conditions, but thermal images often suffer lower resolution and weaker object boundaries than RGB imagery. This paper presents *SEA-YOLO-E* (Spatial Edge Attention YOLO-E), an enhanced single-modality thermal object detector that integrates a SEA mechanism and semi-supervised learning to overcome these challenges. First, we introduce the SEA-YOLO architecture, which embeds an Edge Extractor and a novel SEA module into a YOLOv8 backbone to emphasize object boundaries and improve detection accuracy in thermal domains. Bases on it, we extend SEA-YOLO with a semi-supervised learning paradigm: a self-supervised rotation prediction pretext task leverages unlabeled infrared images to learn general feature representations, and synthetic thermal data mitigates class imbalance in training. The proposed two-phase training (self-supervised pretraining followed by supervised fine-tuning) significantly boosts detection performance. Experiments on multiple thermal driving datasets demonstrate that SEA-YOLO-E achieves state-of-the-art results, with improvements of up to 9–12% in mAP over existing detectors. Notably, our edge-enhanced attention and rotation-pretrained model outperforms recent multi-modal RGB-thermal detectors while using only thermal input. The model is available at https://github.com/GaeulHan0930/SEA-YOLO-E.

Keywords: Attention mechanism, autonomous driving, computer vision, deep learning, object detection, self-supervised learning.

1. Introduction

Autonomous vehicles must reliably detect objects under all visibility conditions, including darkness, fog, and glare. Adverse weather and low-light scenarios degrade traditional RGB camera performance and have been linked to increased accident risk [1]. Thermal infrared cameras, which capture heat radiation, can offer a complementary modality for robust perception [2], [3]. Prior works have explored thermal imaging to enhance object detection in adverse conditions [4]. However, thermal sensors typically produce lower-resolution images with indistinct object edges, leading to missed detections of small or distant objects. Recent research has shown that combining thermal and visible spectra can improve detection performance. Many multispectral detectors fuse RGB and thermal inputs to leverage both modalities; however, such approaches increase system complexity and cost, motivating advances in single-modality thermal detection.

In this work, we aim to boost thermal-image object detection by addressing the edge information loss and data scarcity issues. First, we propose SEA-YOLO, which incorporates an edge extractor module and a spatial attention mechanism to enhance thermal object features; this method compensates for the weak boundary in thermal images. Second, to further improve performance while resolving labeled data scarcity, we adopt a self-supervised rotation prediction pretext on unlabeled raw thermal images to pretrain the model. In addition, we generated synthetic thermal samples to augment underrepresented object classes, improving data balance. The resulting enhanced model, SEA-YOLO-E, is able to detect objects in thermal imagery with higher accuracy and robustness.

[*] ghan2@lakeheadu.ca, takilan@lakeheadu.ca

We validated our approach on multiple thermal datasets, including various conditions. SEA-YOLO-E achieves significant accuracy gains and exceeds the performance of state-of-the-art detectors, despite using only thermal input. In summary, the contributions of this paper include: (i) a novel edge-aware attention architecture for thermal object detection, (ii) addressing the limited labeled dataset issue using self-supervised pretraining and synthetic data, and (iii) comprehensive experiments demonstrating improved detection performance and robustness across different datasets. The rest of this paper is organized as follows: Section 2 investigates existing solutions, Section 3 introduces the proposed method, Section 4 analyzes experimental results, and Section 5 concludes the work with future directions.

2. Related Work

2.1. Thermal and Multispectral Object Detection

Thermal infrared cameras have been explored in the intelligent transportation domain to complement or replace RGB sensors under poor visibility. Early efforts focused on applying existing detectors to thermal images, but faced challenges with lower resolution and contrast. Agrawal and Subramanian [2] showed thermal imaging can enhance detection in foggy conditions, while others leveraged thermal for nighttime pedestrian detection.

To boost accuracy, several works fuse thermal and visible modalities. For example, El Ahmar et al. [5] and Zhang et al. [6] propose RGB-thermal feature fusion frameworks that significantly improve robustness in difficult environments. Jang et al. introduced CAMDet [7], a condition-adaptive multispectral detector using style translation to blend modalities. These multimodal approaches achieve strong results but at the expense of additional sensors and increased model complexity.

Within thermal-only object detection, recent research works have focused on network improvements tailored to infrared characteristics. Li et al. [8] developed a high-performance thermal IR detection framework with centralized feature regulation, and Aboalia et al. [9] explored deep learning models specifically for infrared multi-object detection. To address the challenge of small object detection in thermal UAV imagery, Han et al. proposed a lightweight YOLO-based model LRDS-YOLO [10]. Our work shares a similar goal of improving thermal-object detector accuracy, but we take a distinct approach by incorporating explicit edge-awareness and semi-supervised learning.

2.2. Attention Mechanisms and Self-Supervision

Attention modules have been widely adopted to improve feature representation in deep detectors. Spatial and channel attention mechanisms can adaptively weight important features, as seen in Convolutional Block Attention Module (CBAM) [11–13] and related modules. However, most prior works are developed for RGB images. In thermal images, object edges and shapes are often blurred; thus, injecting boundary-focused attention is a promising idea. Our SEA module shares the concept of spatial attention, but uniquely leverages an external edge map to guide the attention mask.

Self-supervised learning has emerged as an effective way to pre-train models on unlabeled data [14]. Contrastive learning [15, 16] and predictive tasks allow networks to learn useful features without manual labels. For detection tasks, semi-supervised frameworks such as SoftTeacher [17] and Unbiased Teacher [18] have shown that leveraging unlabeled images can improve performance when annotations are limited. Gidaris et al. [19] introduced a simple yet powerful self-supervised task: predicting the rotation angle of an image. This rotation prediction forces the model to understand object geometry and orientation, yielding representations beneficial for downstream tasks. Recently, some works have started integrating self-supervision into object detectors. For instance, Kotthapalli et al. [20] applied

contrastive pretraining for label-efficient detection. In this work, we exploit rotation prediction to pretrain our model, which helps it learn generalizable features from unlabeled data and improve its detection performance.

3. Methodology

Our approach, SEA-YOLO-E, improves thermal object detection through two main innovations: a spatial edge-aware attention architecture SEA-YOLO, as shown in Fig. 1, and a semi-supervised enhancement via self-supervised pretraining and synthetic data. We first describe the base SEA-YOLO network, then detail the semi-supervised extension. An overview of the entire training pipeline is depicted in Fig. 2.

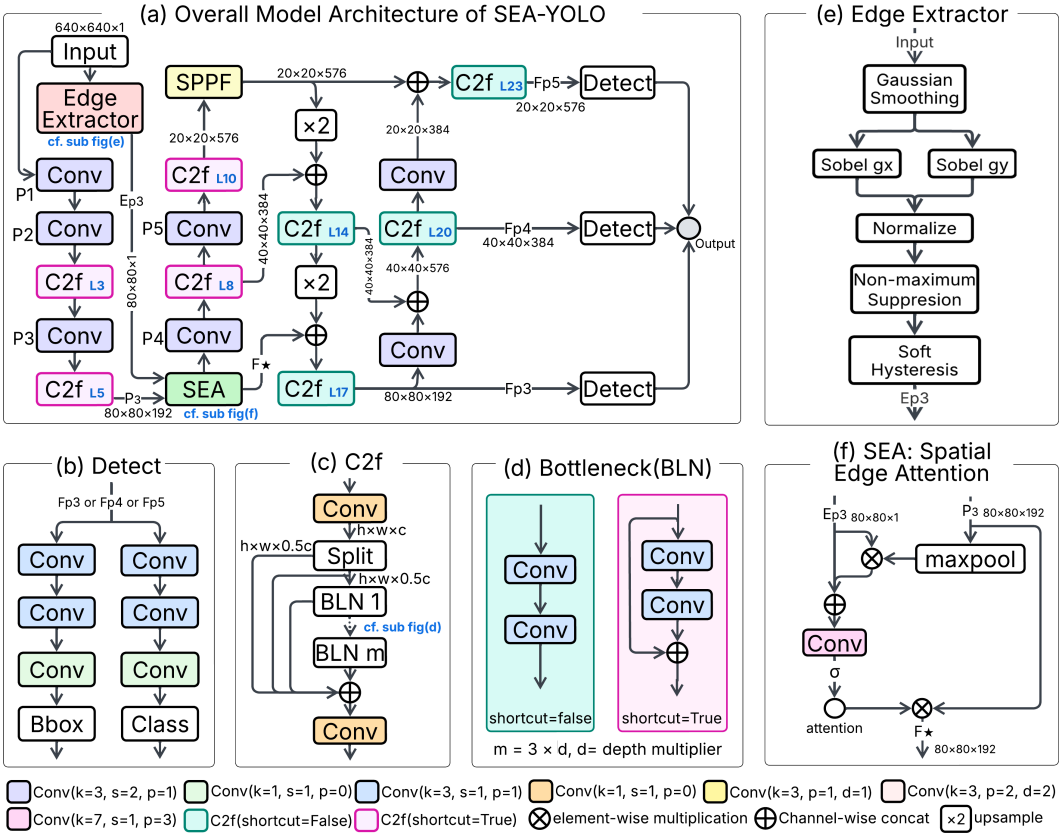

Figure 1. Overall SEA-YOLO architecture with its constituents. Note: P1 to P5 - feature maps with different spatial dimensions and channels, and **L#** - layer index.

3.1. SEA-YOLO with Spatial Edge Attention

SEA-YOLO is built upon the YOLOv8 backbone, optimized to exclusively use thermal input. We chose YOLOv8 for its favorable trade-off between accuracy and speed on our task after ablation tests with different object detectors to choose the optimal model. The key novelty in SEA-YOLO is the introduction of two modules into the backbone (Fig. 1): an Edge-Extractor and a SEA mechanism block.

3.1.1. Backbone and Multi-level Feature Learning

The backbone network processes a thermal image of size 640×640 through a series of convolutional and downsampling layers to produce a hierarchy of feature maps. We denote

by $P3 \in \mathbb{R}^{C\times H\times W}$ an intermediate feature map from a mid-level layer (with $H = W = 80$ at $1/8$ scale of input, and $C = 192$ channels in YOLOv8m). This feature level is chosen for inserting our SEA module, based on the ablation test to decide the optimal configuration. The backbone layers then produce feature maps in different levels and feed into the detection head.

3.1.2. **The Edge-Extractor Module**

Thermal images often exhibit weak intensity contrast and blurred object boundaries; to provide the detector with an explicit boundary cue, we compute a Canny-style edge response from the raw thermal input and make it available to the subsequent pipeline. The Edge-Extractor module (Fig. 1 (e)) follows the classical sequence of (i) Gaussian smoothing, (ii) gradient estimation, (iii) non-maximum suppression, and (iv) hysteresis thresholding, implemented in a differentiable manner.

Given an input thermal image $I \in \mathbb{R}^{1\times H\times W}$, we first apply Gaussian smoothing to suppress sensor noise: $I_s = I * G_\sigma$, where $G_\sigma \in \mathbb{R}^{5\times 5}$ is a normalized Gaussian kernel with parameter σ ($k = 5, p = 2$). From the smoothed image, we estimate horizontal and vertical intensity derivatives using Sobel filters K_x and K_y: $G_x = I_s * K_x, G_y = I_s * K_y$, where $K_x, K_y \in \mathbb{R}^{3\times 3}$ are fixed Sobel kernels ($k = 3, p = 1$). After this, we compute the gradient magnitude map: $M = (G_x^2 + G_y^2 + \epsilon)^{0.5}$, with a small ϵ for numerical stability, and normalize it to a comparable range: $\hat{M} = M/(\max(M) + \epsilon)$. To obtain accurate edges, we perform Canny-style non-maximum suppression (NMS) on $\hat{M}$ along the local gradient direction. The gradient orientation is computed as $\theta = \text{atan2}(G_y, G_x + \epsilon)$, and quantized into four principal directions $0°, 45°, 90°, 135°$. For each pixel, we compare $\hat{M}$ with its two neighbors along the quantized direction and keep it only if it is a local maximum. Denoting the NMS operator by $\mathcal{N}(\cdot)$, we obtain a refined magnitude map $\hat{M}_{\text{nms}} = \mathcal{N}(\hat{M}; G_x, G_y)$.

We apply a soft hysteresis mechanism with low and high thresholds (τ_ℓ, τ_h) to emphasize reliable boundaries while retaining informative weak edges: $S = \mathbb{I}(\hat{M}\text{nms} \geq \tau_h), W = \mathbb{I}(\hat{M}\text{nms} \geq \tau_\ell)$, where $\mathbb{I}(\cdot)$ is the indicator function. Instead of hard connectivity-based hysteresis, we form a soft edge response $E_{\text{full}} = S + \alpha \cdot (W - S)$, with $\alpha = 0.5$ in our implementation, which preserves strong edges while partially retaining weak candidates. The resulting $E_{\text{full}} \in \mathbb{R}^{1\times H\times W}$ is then resized via bilinear interpolation to match the spatial resolution of the P3 feature scale: $E_{P3} = \text{Resize}\times\frac{1}{8}(E_{\text{full}}) \in \mathbb{R}^{1\times \frac{H}{8}\times \frac{W}{8}}$. This E_{P3} provides an aligned boundary feature at the P3 scale, enabling the detector to see object contours even under low-contrast thermal conditions.

3.1.3. **The SEA Module**

The SEA module (Fig. 1-(f)) enhances the intermediate feature map P_3 using both semantic cues and edge information. To begin, a preliminary spatial attention map $M \in \mathbb{R}^{1\times 80\times 80}$ is derived by applying channel-wise max pooling over all $C = 192$ channels of P_3, i.e., $M = \text{Max}(P_3)$. This attention map is then scaled element-wise by $(1 + E_{P3})$ to amplify edge-dense regions. The resulting map is concatenated with the edge map E_{P3}, producing $Z = \text{Cat}(M \otimes (1 + E_{P3}), E_{P3}) \in \mathbb{R}^{2\times 80\times 80}$, where $\otimes$ denotes element-wise multiplication. To generate a refined attention mask, Z is passed through a 7×7 convolutional layer followed by a sigmoid activation, yielding $A = \sigma(\text{Conv}(Z)) \in (0, 1)^{1\times 80\times 80}$. Finally, this mask is applied to the original feature map as $F^\star = P_3 \otimes A$, resulting in an edge-aware refined feature $F^\star \in \mathbb{R}^{192\times 80\times 80}$. Through this mechanism, spatial locations and object edges are emphasized, while less informative regions are suppressed. The output $F^\star$ is forwarded through the detection head and used to attain the final output.

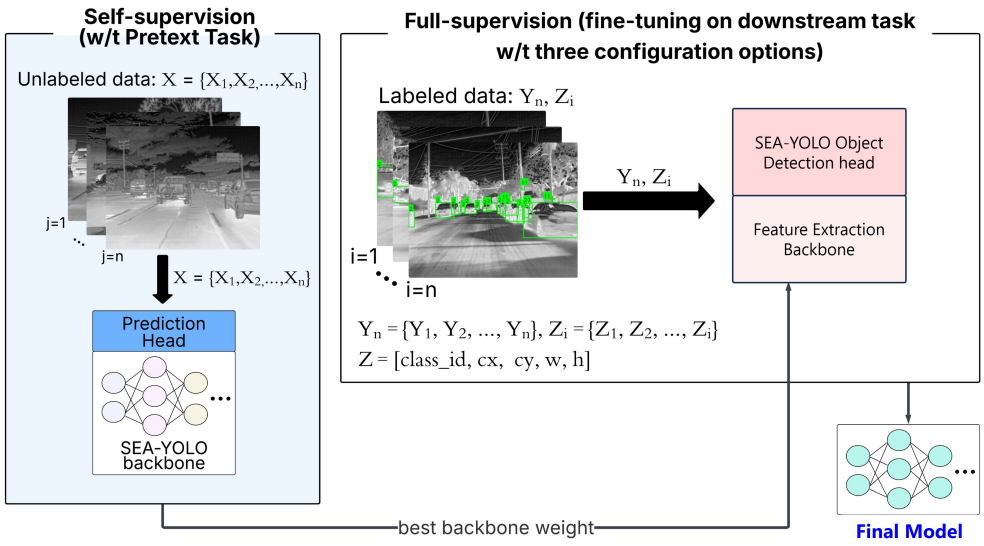

Figure 2. SEA-YOLO-E training pipeline, which consists of two stages: self-supervision and supervised fine-tuning. Unlabeled thermal images X are used in the rotation prediction task (Stage 1 - Section 3.2.1), after which the pretrained backbone is integrated into the detection model for supervised training on labeled data Y (Stage 2 - Section 3.2.2).

3.2. **Semi-supervised Extension (SEA-YOLO-E)**

Building upon SEA-YOLO, we introduce a semi-supervised learning framework to further enhance detection performance without additional annotated data. This extension has two components, as depicted in Fig. 2: (1) self-supervised pretraining via a rotation prediction task on unlabeled thermal images, and (2) supervised fine-tuning on the target detection dataset (details shown in Fig. 3). We refer to the final model as SEA-YOLO-E.

3.2.1. **Rotation Prediction Pretext Task**

Manual labeling is labor-intensive and costly. To exploit unlabeled data, we adopted the rotation prediction task proposed by Gidaris *et al.* as a self-supervised pretext. The intuition is that by training the network to recognize image rotations, we encourage it to learn about object shapes, orientations, and context, which are useful for detection.

For pretraining, as shown in Fig. 3, we collect a set of unlabeled thermal images X. Each image is randomly rotated by one of four angles $\{0°, 90°, 180°, 270°\}$; the rotation angle index (0,1,2,3) serves as a synthetic label r for the image. We then train the backbone network of SEA-YOLO to predict r correctly via a 4-way classification loss. Specifically, the feature map $F^{\star}$ from Layer 6 in the SEA-YOLO backbone is pooled into a feature vector, which is fed into a small rotation classifier to output logits for the 4 rotation classes. We optimize this network on the unlabeled set X using a cross-entropy loss. This pretext task makes the model learn general features. This helps the backbone encode semantically meaningful and orientation-invariant features, which provide a strong initialization for detection. After this self-supervision, pretrained SEA-YOLO backbone weights will be used to initialize the detector for fine-tuning, as shown in Fig. 2.

3.2.2. **Supervised Fine-tuning**

The second stage fine-tunes the pretrained SEA-YOLO model on the labeled target dataset for object detection. Specifically, we integrate this pretrained backbone with learned

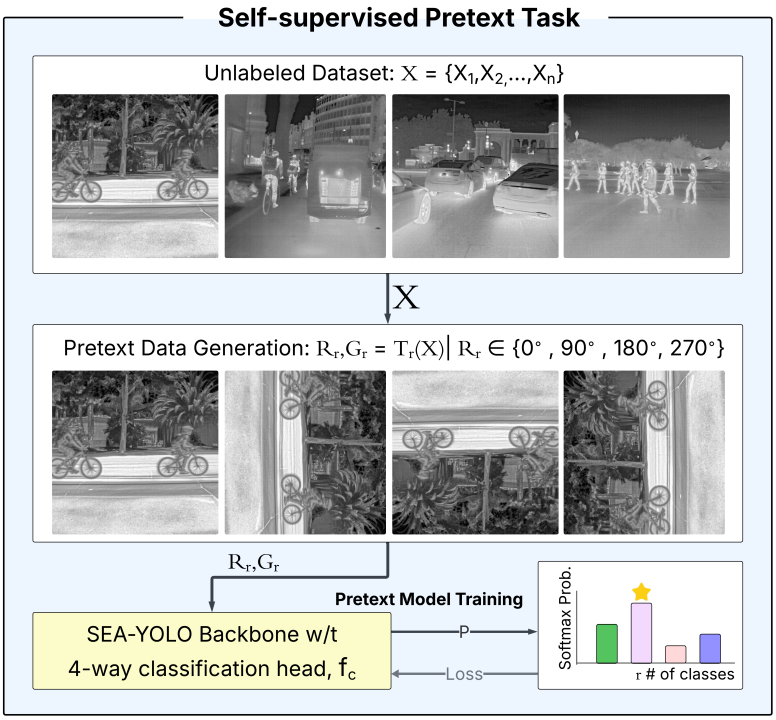

Figure 3. Illustration of the rotation prediction self-supervised task. Unlabeled infrared images are rotated by 0, 90, 180, or 270 degrees, and the network predicts the rotation angle by using softmax probability.

weights with the full YOLO detection head. We then train the detector in a supervised manner using the ground truth bounding boxes and class labels of the training set Y.

For a thorough analysis, we considered three different fine-tuning strategies:

a) **SEA-YOLO-E (Entire model fine-tuning):** We allow all layers (backbone + detection head) to be trained on the labeled data. This gives the model full flexibility to adjust, at the cost of more training time and potential overfitting if the dataset is small.

b) **SEA-YOLO-P (Partial fine-tuning):** We freeze the middle backbone layers and only fine-tune the early part of the backbone (L0 to L6), which includes our novel Edge Extractor (L0) and SEA module (L6) and the head. In our experiments, we fine-tuned L0 to L6 and head (L12 to L24). This reduces training time and retains more of the self-supervised features.

c) **SEA-YOLO-H (Head-only fine-tuning):** We freeze the entire backbone (all pretrained weights) and only train the detection head L12 to L24.

We empirically evaluated these configurations as shown in Table 2. Through this ablation study, we found that the entire model fine-tuning (SEA-YOLO-E) achieved the highest accuracy; thus, we considered the SEA-YOLO-E configuration for the final comparative analysis.

3.3. Synthetic Data Augmentation

Among the datasets used in this work, FLIR-aligned and HIT-UAV exhibited severe inter-class imbalance. To mitigate the significant inter-class imbalance and empirically evaluate the impact of reducing this inter-class gap, we employed a synthetic data generation strategy. Specifically, minority-class objects were extracted as individual patches, while background

images were curated separately for both training and validation sets to maintain data split integrity. Leveraging `ChatGPT-4o`, we created synthetic images by compositing the cropped object patches onto the selected backgrounds, avoiding duplicates and generating accurate annotation files for each synthetic instance. These generated samples were then incorporated into the respective training and validation datasets.

For FLIR-aligned, a total of 1474 and 136 synthetic images for training and validation of the `bicycle` class, and 550 and 110 images of the `person` class were respectively created to decrease the gap among classes. For HIT-UAV, a total of 300 synthetic images for training and 30 for the validation set, which include multiple `OtherVehicle` class objects. These numbers were chosen as a result of empirical experiments (see Appendix A), and tables of quantitative comparison in Section 4.4 show the effect of synthetic data on model performance.

4. Experimental Setup and Performance Analysis

4.1. Datasets

We evaluate our approach on three thermal image benchmarks: FLIR-aligned, M3FD, and HIT-UAV, which represent diverse scenarios. Details are shown in the Table 1.

Dataset	Type	Resolut.	N_c	Train	Val	Test
FLIR-aligned [21]	IR+RGB	640×512	4	3,717	412	1,013
M3FD [22]	IR+RGB	1280×1024	6	2,905	863	432
HIT-UAV [23]	IR	640×512	5	2,008	287	571

Note: IR – infrared; IR+RGB – paired thermal and RGB; N_c – number of categories.

Table 1. Key attributes of the benchmark thermal datasets

4.2. Evaluation Metrics

Mean average precision (mAP) is adopted as the primary evaluation metric. For each object class i, the Average Precision (AP) is defined as the area under the corresponding precision–recall curve and is computed as $AP_i = \int_0^1 P_i(r)\,dr$, and the mean Average Precision is defined as $mAP = \frac{1}{N_c}\sum_{i=1}^{N_c} AP_i$, where $P_i(r)$ represents the precision as a function of recall r, and N_c denotes the total number of object classes. The mean Average Precision (mAP) is obtained by averaging the AP values across all classes. In this work, we report the average of mAP50 and mAP50 : 95 as an additional composite metric.

4.3. Training Strategy and Environment

Training was performed using the `AdamW` optimizer with an initial learning rate of 0.005 and 0.01. The loss comprised the CIoU (Complete IoU) and DFL (Distribution Focal Loss) for bounding box regression, and BCE (Binary Cross Entropy) for classification and objectness prediction. The batch size was set between 8 and 16 samples, and training converged within 50–100 epochs. Model development was conducted using Python 3.11.5, using Pytorch 2.7.1. Training and evaluation were performed on the Nibi cluster of the Digital Research Alliance of Canada, utilizing an NVIDIA Tesla H100 GPU.

4.3.1. Finetuning Strategy

We examined how different fine-tuning strategies work in this work and which is optimal. Table 2 summarizes the performance of SEA-YOLO under three fine-tuning modes: SEA-YOLO-E, SEA-YOLO-H, and SEA-YOLO-P as described earlier in section 3.2.2. All

Configurations	FLIR-aligned			HIT-UAV		
	mAP_{50} ↑	mAP↑	Train(m)↓	mAP_{50} ↑	mAP↑	Train(m)↓
SEA-YOLO*	77.0	43.9	60.5	80.1	54.9	67.5
SEA-YOLO-E	**82.1**	47.2	36.3	**83.3**	**59.3**	46.1
SEA-YOLO-H	82.0	47.4	37.8	83.2	57.5	**41**
SEA-YOLO-P	81.5	**48.3**	**36.2**	82.5	58.1	45.4

Note: SEA-YOLO* - no self-supervision; SEA-YOLO-P, SEA-YOLO-H, and SEA-YOLO-E denote self-supervision followed by partial, head-only, and entire model supervision, respectively. (see Section 3.2.2); Train (m) - downstream finetuning time in minutes; **bold** - the best; and underline - the second best results.

Table 2. Comparative analysis of downstream finetuning configurations with SEA-YOLO across FLIR-aligned and HIT-UAV benchmark datasets

Model	Input		3-class (%) ↑			Complexity ↓	
	IR	RGB	mAP_{50}	mAP	Avg	#Param (M)	GFLOPs
EFETN (2025) [24]	✓		73.2	37.7	55.5	-	-
EFETN (2025) [24]	✓	✓	75.6	38.8	57.2	-	-
SSD (2025) [24]	✓		65.5	29.6	47.6	-	-
YOLOv3 (2025) [24]	✓		73.6	36.8	55.2	63	157.3
YOLOv5 (2025) [24]	✓		72.8	36.5	54.7	46.5	109.1
YOLOv9 (2025) [24]	✓		73.6	38.1	55.9	20	76.5
RT-DETR (2024) [25]	✓	✓	70.1	34.3	50.2	42	136
CAMDet (2025) (V2T) [26]	✓	✓	75.4	31.5	53.5	-	-
CAMDet (2025) (T2V) [26]	✓	✓	76.3	31.8	54.1	-	-
Ours w/o self-supervision							
SEA-YOLO	✓		76.3	42.4	59.4	25.8	78.8
SEA-YOLO*	✓		77.0	43.9	60.5	25.8	78.8
Ours w/t self-supervision							
SEA-YOLO-E*† ◐	✓		79.4	46.3	62.9	25.8	78.8
SEA-YOLO-E*† ◔	✓		79.9	**47.5**	63.7	25.8	78.8
SEA-YOLO-E*† ●	✓		**82.1**	47.2	**64.7**	25.8	78.8

Note: † - integration of semi-supervision; * - use of synthetic data in supervised-learning; ◐ - use of 50% of train set; ◔ - use of 70% of train set; ● - use of whole train set; Avg - Average of mAP_{50} & mAP; #Param. (M) - number of trainable parameters in million; ↑ - highest is best; ↓ - lowest is best; **bold** – best results; underline – second-best results.

Table 3. Analysis on FLIR-aligned using different finetuning configurations & policies.

cases use the rotation-pretrained backbone weights. We found that SEA-YOLO-E achieves notably higher mAP than SEA-YOLO* on FLIR-aligned and updating the entire model (SEA-YOLO-E) gives the highest accuracy. Based on this, we conducted a comparative analysis with SEA-YOLO-E with three different data policies: using i) 50% of the train set ◐, ii) 70% of the train set ◔ , and iii) the entire train set ● to thoroughly analyze how self-supervision pretraining positively affects the model performance even with a smaller labeled dataset.

4.4. **Quantitative Analysis**

In this section, we present the test results of SEA-YOLO-E in comparison to other methods on the three datasets. All benchmarks include real-world low-visibility scenarios through which we can prove the robustness of the proposed model in challenging situations.

Model	Input		6-class (%)↑			Complexity↓	
	IR	RGB	mAP_{50}	mAP	Avg	#Param	GFLOPs
YOLOv5 IR (2025) [4]	✓		57.2	34.9	46.1		
YOLOv5 RGB (2025) [4]		✓	60.2	36.1	48.2		
DIVFusion (2025) [4]	✓	✓	60.8	37.1	48.9		
PSFusion (2025) [4]	✓	✓	61.1	38.0	49.6		
AUIF (2025) [4]	✓	✓	**62.0**	38.3	50.2	NFL	NFL
CDDF (2025) [4]	✓	✓	61.9	38.6	50.3		
U2Fusion (2025) [4]	✓	✓	61.9	38.7	50.3		
TarDAL (2025) [4]	✓	✓	61.9	39.1	50.5		
Ours w/o self-supervision							
SEA-YOLO	✓		60.5	39.7	50.1	25.8	78.8
Ours w/t self-supervision							
SEA-YOLO-E† ◐	✓		57.8	37.0	47.4	25.8	78.8
SEA-YOLO-E† ◔	✓		60.2	38.6	50.4	25.8	78.8
SEA-YOLO-E† ●	✓		61.6	**40.5**	**51.1**	25.8	78.8

Note: † – integration of semi-supervision; * – use of synthetic data in supervised learning; ◐ – use of 50% of train set; ◔ – use of 70% of train set; ● – use of whole train set; Avg – Average of mAP_{50} & mAP; #Param. (M) – number of trainable parameters (million); ↑ – higher is better; ↓ – lower is better; **bold** – best results; underline – second-best results, NFL - Not found in the literature.

Table 4. Analysis on M3FD dataset using different finetuning configurations & policies

4.4.1. **FLIR-aligned Dataset**

Table 3 shows the model performance on FLIR-aligned. SEA-YOLO-E ● achieves the highest performance with 82.1% mAP_{50} and 47.2% mAP. Compared to YOLOv9, SEA-YOLO-E ● shows improvements of +8.5% (mAP_{50}) and +9.1% (mAP). Compared to CAMDet (T2V), it yields improvements of +5.8% (mAP_{50}) and +15.4% (mAP). Compared to SEA-YOLO without self-supervision, SEA-YOLO-E improves by +5.8% (mAP_{50}) and +4.8% (mAP). Most meaningful find is that SEA-YOLO-E ◐ and SEA-YOLO-E ◔ achieve higher mAP and mAP50 than any other existing state-of-art models even though they use only 50% and 70% of labeled FLIR-aligned train set. This confirms the effectiveness of self-supervision pretraining in the labeled data scarcity condition.

4.4.2. **M3FD Dataset**

Referring to Table 4, it is found that SEA-YOLO-E ● achieves 61.6% mAP_{50} and 40.5% mAP. Compared to the baseline YOLOv5 IR, our model improves by +4.4% (mAP_{50}) and +5.6% (mAP). SEA-YOLO-E achieves the best performances wrt mAP and the average of mAP50 and mAP with +3.4% mAP than DIVFusion and +1.4% mAP than TarDAL even though it uses thermal-only modality while others use thermal+RGB modality.

4.4.3. **HIT-UAV Dataset**

Table 5 shows results on HIT-UAV. For the 5-class setting, SEA-YOLO-E ● achieves 83.3% mAP_{50} and 59.3% mAP. LRDS-YOLO shows the highest mAP_{50} at 84.5%, but SEA-YOLO-E achieves the highest mAP. Compared to SEA-YOLO without self-supervision and synthetic data, which achieved 77.6% mAP_{50} and 52.1% mAP, SEA-YOLO-E improves by +5.7% (mAP_{50}) and +7.2% (mAP). In the 4-class setting, SEA-YOLO-E ● records 87.6% mAP_{50} and 62.5% mAP. Compared to DINO which shows 85.4% mAP_{50} and 52.9% mAP, SEA-YOLO-E leads by +2.2% (mAP_{50}) and +9.6% (mAP). Also, compared to SEA-YOLO (84.2% mAP_{50}, 56.9% mAP), the gains are +3.4% mAP50 and +5.6% mAP.

Model	Input		5-class (%)↑			4-class (%)↑			Complexity↓	
	IR	Syn	mAP_{50}	mAP	Avg	mAP_{50}	mAP	Avg	#Param. (M)	GFLOPs
LRDS-YOLO (2025) [27]	✓		**84.5**	54.2	69.4				NFL	NFL
YOLOv5 (2025) [27]	✓		75.4	48.1	61.8				21.2	109.1
YOLOv6 (2025) [27]	✓		78.3	49.7	64.0				34.3	85.8
YOLOv9 (2025) [27] (Baseline)	✓		80.8	52.1	66.5				20.0	76.3
YOLOv10 (2025) [27]	✓		77.7	47.3	62.5				24.0	74.0
YOLOv11 (2025) [27]	✓		82.3	52.4	67.4				25.6	94.0
RT-DETR (2025) [27]	✓		78.8	52.2	65.5				42.0	136.0
SimCLR (2025) [28]	✓	NFL	NFL	NFL	NFL	80.4	49.7	65.1	NFL	NFL
DeepClusterV2 (2025) [28]	✓					81.2	50.0	65.6		
DINO (2025) [28]	✓					85.4	52.9	69.2		
MAE (2025) [28]	✓					83.7	51.8	67.8		
Ours w/o self-supervision										
SEA-YOLO	✓		77.6	52.1	64.9	84.2	56.9	70.9	25.8	78.8
SEA-YOLO*	✓	✓	80.1	54.9	67.5	85.2	58.1	71.7	25.8	78.8
Ours w/t self-supervision										
SEA-YOLO-E*† ◐	✓		80.8	56.2	68.5	84.5	58.3	71.4	25.8	78.8
SEA-YOLO-E*† ◔	✓		75.8	52.3	64.1	79.9	55.5	67.7	25.8	78.8
SEA-YOLO-E*† ●	✓	✓	83.3	**59.3**	**71.3**	**87.6**	**62.5**	**75.1**	25.8	78.8

Note: † - integration with semi-supervision; * - use of synthetic data in supervised-learning; ◐ - use of 50% of train set; ◔ - use of 70% of train set; ● - use of whole train set; Avg - Average of mAP_{50} & mAP; #Param. (M) - number of trainable parameters in million; ↑ - highest is best; ↓ - lowest is best; **bold** – best results; underline – second-best results; NFL - Not found in the literature.

Table 5. Analysis on HIT-UAV dataset w/t different finetuning configurations & policies

4.5. **Qualitative Analysis**

Fig. 4 shows the success scenarios of highly occluded objects and distant + occluded objects detection by improved SEA-YOLO with pretraining task in this chapter. In this work, with integration of self-supervision pretraining, SEA-YOLO-E learns meaningful generalizable features from unlabeled data, and attains great performance and better occluded object detection performance in return. As a result, as shown from the 1st row of the 3rd column of Fig. 4, the proposed model successfully detected a very small `person` instance which is highly occluded by a tree. In addition, as shown from the 2nd row of the same column, a very distant `person` instance, even hardly noticeable without a ground-truth box, is successfully detected as well.

5. **Conclusion**

We presented SEA-YOLO-E, a novel thermal object detection model that integrates spatial edge-aware attention and self-supervised learning to tackle the challenges ofthe previous limitation of thermal-based object detection and the issue of high dependency on labeled datasets. Our approach enhances a strong one-stage detector with the Edge Extractor and SEA module that leverages the extracted edges to enhance object features. In addition, we introduced a semi-supervised training paradigm: first pretraining the model on unlabeled thermal images via rotation prediction to learn general feature representations, and then fine-tuning on labeled data with strategic synthetic augmentation. On multiple benchmarks, SEA-YOLO-E achieved state-of-the-art performance with less labeled data than benchmarks, demonstrating the effectiveness of edge-guided attention and self-supervised feature learning.

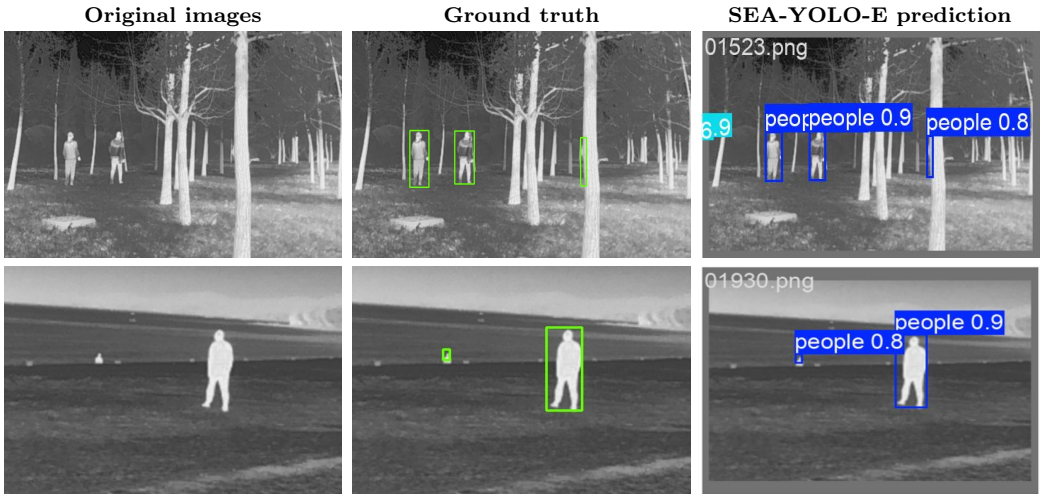

Figure 4. Object detection success scenario from M3FD dataset (2nd row: Occluded object detection success scenario - the SEA-YOLO-E with self-supervision pretraining succeeds to detect `people` instances in a highly occluded case. 3rd row: Distant object detection success scenario - the SEA-YOLO-E with self-supervision pretraining succeeds to detect very small `people` instances in a distant case.

This work shows that even without RGB modality, a thermal-only single modality detector can be significantly improved by novel network design and self-supervised learning techniques. For future research, our framework could be extended by exploring other self-supervised tasks and thermal benchmarks and reducing model complexity.

Acknowledgements

We acknowledge that this research was enabled in part by support provided by the Digital Research Alliance of Canada (https://alliancecan.ca/en).

References

[1] Federal Highway Administration. *How Do Weather Events Affect Roads?* Accessed: 2025-11-02. 2025. URL: https://ops.fhwa.dot.gov/weather/roadimpact.htm.

[2] K. Agrawal and A. Subramanian. "Enhancing Object Detection in Adverse Conditions using Thermal Imaging". In: *arXiv preprint arXiv:1909.13551* (2019).

[3] W. El Ahmar, Y. Massoud, D. Kolhatkar, H. AlGhamdi, M. Alja'afreh, R. Hammoud, and R. Laganière. "Enhanced Thermal-RGB Fusion for Robust Object Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2023.

[4] X. Zhang, X. Zhang, J. Wang, J. Ying, Z. Sheng, H. Yu, C. Li, and H.-L. Shen. "TFDet: Target-Aware Fusion for RGB-T Pedestrian Detection". In: *IEEE Transactions on Neural Networks and Learning Systems* 36.7 (2025), pp. 13276–13290. DOI: 10.1109/TNNLS.2024.3443455.

[5] A. El Ahmar et al. "Multispectral object detection using RGB–thermal fusion". In: *IEEE Transactions on Intelligent Transportation Systems* (2023).

[6] L. Zhang et al. "Robust RGB-T object detection via cross-modal feature fusion". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.

[7] Y. Jang and S. Kim. "CAMDet: Condition-adaptive multimodal detection via style translation". In: *Pattern Recognition* (2023).

[8] X. Li and Y. Wang. "Centralized feature regulation for thermal infrared object detection". In: *Infrared Physics & Technology* (2022).

[9] M. Aboalia and I. Abdel-Qader. "Deep learning-based infrared multi-object detection". In: *Sensors* 21.7 (2021), p. 2463.

[10] J. Han et al. "LRDS-YOLO: Lightweight YOLO-based detector for long-range small infrared objects". In: *Sensors* (2024).

[11] S. Woo, J. Park, J.-Y. Lee, and I. Kweon. "Cbam: Convolutional block attention module". In: *ECCV (30th European Conference on Computer Vision)*. 2018, pp. 3–19.

[12] H. Sundaralingam, T. Suresh, and T. Akilan. "SegAttnDetec: A Segmentation-Aware Attention-Based Object Detector". In: *Procedia Computer Science* 260 (2025), pp. 914–922.

[13] H. Sundaralingam, T. Suresh, T. Akilan, and S. B. Ahmed. "Dilated Strip-Wise Spatial Feature Pyramid: An Efficient Network for Object Detection". In: *2025 IEEE 34th International Symposium on Industrial Electronics (ISIE)*. IEEE. 2025, pp. 1–6.

[14] T. Akilan, N. Jahan, and W. Zhang. "Self-supervised learning for image segmentation: a comprehensive survey". In: *arXiv preprint arXiv:2505.13584* (2025).

[15] K. He et al. "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[16] T. Chen et al. "A simple framework for contrastive learning of visual representations". In: *Proceedings of the International Conference on Machine Learning*. 2020.

[17] M. Xu et al. "End-to-end semi-supervised object detection with soft teacher". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.

[18] Y. Liu et al. "Unbiased teacher for semi-supervised object detection". In: *Proceedings of the International Conference on Learning Representations*. 2021.

[19] S. Gidaris, P. Singh, and N. Komodakis. "Unsupervised representation learning by predicting image rotations". In: *Proceedings of the International Conference on Learning Representations*. 2018.

[20] S. Kotthapalli et al. "Self-supervised pretraining for YOLO-based object detection". In: *arXiv preprint arXiv:2206.01234* (2022).

[21] Kaggle. *FLIR_aligned*. https://www.kaggle.com/datasets/s111mple/flir-aligned. Dataset page on Kaggle. Kaggle, 2023.

[22] J. Liu, X. Fan, Z. Huang, G. Wu, R. Liu, W. Zhong, and Z. Luo. *Target-aware Dual Adversarial Learning and a Multi-scenario Multi-Modality Benchmark to Fuse Infrared and Visible for Object Detection*. arXiv:2203.16220. 2022.

[23] J. Suo, T. Wang, X. Zhang, H. Chen, W. Zhou, and W. Shi. "HIT-UAV: A high-altitude infrared thermal dataset for Unmanned Aerial Vehicle-based object detection". In: *Scientific Data* 10.1 (2023). DOI: 10.1038/s41597-023-02066-6.

[24] G. Zhao, J. Zhu, Q. Jiang, S. Feng, and Z. Wang. "Edge Feature Enhanced Transformer Network for RGB and infrared image fusion based object detection". In: *Infrared Physics & Technology* 147 (2025), p. 105824. ISSN: 1350-4495. DOI: https://doi.org/10.1016/j.infrared.2025.105824. URL: https://www.sciencedirect.com/science/article/pii/S1350449525001173.

[25] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen. "Detrs beat yolos on real-time object detection". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 16965–16974.

[26] J. Jang, J. Lee, and J. Paik. "CAMDet: Condition-Adaptive Multispectral Object Detection Using a Visible-Thermal Translation Model". In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2025.

[27] Y. Han, C. Wang, H. Luo, et al. "LRDS-YOLO enhances small object detection in UAV aerial images with a lightweight and efficient design". In: *Scientific Reports* 15 (2025), p. 22627. DOI: 10.1038/s41598-025-07021-6.

[28] S. Konstantakos, J. Cani, I. Mademlis, D. I. Chalkiadaki, Y. M. Asano, E. Gavves, and G. T. Papadopoulos. "Self-supervised visual learning in the low-data regime: A comparative evaluation". In: *Neurocomputing* 620 (Mar. 2025), p. 129199. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2024.129199. URL: http://dx.doi.org/10.1016/j.neucom.2024.129199.

Appendix A. Synthetic Data Generation

A.1. Use of Generative AI

Fig. 5 illustrates how we prompted a generative AI model to generate synthetic datasets while maintaining a strict train/validation split and precise annotations. Step#3 to 5 were fulfilled by using `ChatGPT-4o`, whereas the other steps were manually done. Fig. 6 shows a few synthetically generated samples using the strategy described above.

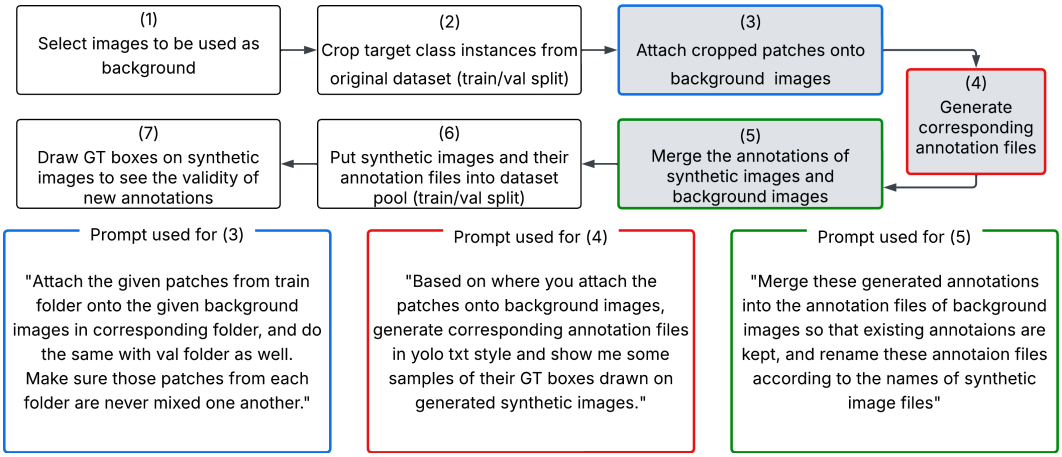

Figure 5. Synthetic data generation process. ☐ - done manually, ■ - done by GenAI with user prompts. ☐ - prompt for step #3, ☐ – prompt for step #4, ☐ – prompt for step #5.

Examples of Cropped patches of `bicycle` instance & Three distinct backgrounds

A few annotated synthetic samples generated targeting the `bicycle` instance in FLIR-aligned dataset

Figure 6. An example of synthetic samples with their target instance (cropped patch), and the used backgrounds from FLIR dataset.

A.2. Empirical Experiments

As our primary research scope did not include an in-depth study of synthetic data for deep learning training, we instead tried to address the severe class imbalance found across the

datasets while improving detection accuracy during the course of our research. Therefore, we conducted empirical experiments to decide the number of synthetic samples generated for each dataset instead of excessively focusing on methodology of synthetic data itself. Initial empirical tests shown in Table 6 included only `bicycle` synthetic samples for FLIR-aligned, while the other ablation tests on the FLIR-aligned dataset included synthetic data of `bicycle` and `person` classes, as shown in Table 7, which was adopted for this work.

Table 6 shows the results of the initial empirical experiments mentioned in Section 3.3. At the 1st iteration, the proposed model using synthetic samples of `OtherVehicle` class acquired improved detection performance on HIT-UAV, compared to the original performance without integration of synthetic data; therefore, we stopped the iteration for the said dataset and used the samples generated for the first round. On the other hand, the proposed model did not attain better detection results with synthetic data on the FLIR-aligned dataset in the first round. We generated more synthetic samples of `bicycle` class on FLIR-aligned dataset using the same prompts and process as illustrated in Fig. 5, and conducted the test for the second iteration; it showed the better result at the second iteration, so we stopped the experiment there and integrated the synthetic `bicycle` samples generated at the second round into our further tests.

Table 7 illustrates the results of an additional empirical ablation study. In this stage, we generated synthetic samples of the `person` class in the FLIR-aligned dataset to further address inter-class imbalance and incorporated them into our dataset. With integration of both `bicycle` and `person` synthetic images, we achieved a final detection performance of 77.0% mAP_{50} and 43.9% mAP, as shown in Tables 3 and 7. Following this ablation study, we proceeded with further experiments on semi-supervised learning using this final synthetic dataset.

Iteration	**FLIR-aligned**			**HIT-UAV**		
	# of samples	mAP50 ↑	mAP ↑	# of samples	mAP50 ↑	mAP ↑
w/o synthetic	N/A	76.3%	42.4%	N/A	77.6%	52.1%
1st	Train: 1,000 instances Val: 100 instances	75.4%	42.5%	Train: 600 instances Val: 60 instances	**80.1%**	**54.9%**
2nd	Train: 7,370 instances Val: 680 instances	**76.3%**	**43.9%**	N/A	N/A	N/A

Note: FLIR-aligned - synthetic samples of `bicycle` class; HIT-UAV - synthetic samples of `OtherVehicle` class; w/o synthetic - the test conducted without any synthetic dataset used; # of samples - the number of synthetic image instances (target class objects) and corresponding annotation files; mAP50 & mAP - detection results of the proposed model with and without the integration of synthetic dataset. ↑ - highest is best, ↓ - lowest is best. **bold** – best results.

Table 6. Initial empirical experiments results to decide the volume of synthetic datasets.

Model/ Configuration	**Use of Synthetic Data w/t:**		**Performance** (%)↑		
	`bicycle` class	`person` class	mAP_{50}	mAP	Avg
SEA-YOLO			76.3	42.4	59.4
SEA-YOLO*	✓		76.3	**43.9**	<u>60.1</u>
SEA-YOLO*	✓	✓	**77.0**	**43.9**	**60.5**

Note: `bicycle` class - 7,370 train/680 val instances, `person` class - 2,750 train/550 val instances used; ✓ – included; * - use of synthetic data; ↑ – higher is better; **bold** – best results; <u>underline</u> – second-best results.

Table 7. Ablation study on FLIR-aligned dataset to determine the optimal integration of edge filter and synthetic data