TABLE I
PERFORMANCE INDICATORS USED

Indicator	Formula	Section
Precision (P)	$\frac{\text{TP}}{\text{TP}+\text{FP}}$	III,IV,V
Recall–sensitivity (R)	$\frac{\text{TP}}{\text{TP}+\text{FN}}$	III,IV,V
Specificity	$\frac{\text{TN}}{\text{TN}+\text{FP}}$	IV
Accuracy	$\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$	IV,VII
F1 score	$\frac{2\times P\times R}{P+R}$	IV
Mean average precision (mAP)	$\frac{1}{N}\sum_{k=1}^{N}\text{AP}_k$	III
Intersection over union (IoU)	$\frac{\lvert A\cap B\rvert}{\lvert A\cup B\rvert}$	III

TABLE II
RESULTS FOR TESTING SSD, DETR, YOLOV5, YOLOV9, AND YOLOV10, WITH IOU $= 0.5$

	model		P	R	$\text{mAP}_{0.5}$	$\text{mAP}_{0.5}^{0.95}$
$C = 0.3$	SSD	**ALL**	65.5%	26.8%	57.4%	25.1%
		ONLY	73.1%	26.8%	59.8%	26.3%
	DETR	**ALL**	81.8%	12.7%	50.9%	24.1%
		ONLY	81.8%	12.7%	54.4%	25.8%
	YOLOV5-X	**ALL**	53.5%	54.8%	44.4%	29.0%
		ONLY	95.1%	54.8%	52.0%	34.3%
	YOLOV9-T	**ALL**	71.0%	61.0%	67.0%	37.0%
		ONLY	95.0%	61.0%	78.0%	44.0%
	YOLOV9-E	**ALL**	57.50%	52.0%	51.0%	27.0%
		ONLY	89.0%	52.0%	69.0%	37.0%
	YOLOV10-N	**ALL**	74.0%	60.0%	50.0%	23.0%
		ONLY	90.0%	60.0%	71.0%	33.0%
	YOLOV10-X	**ALL**	50.0%	64.0%	41.0%	19.0%
		ONLY	79.0%	64.0%	72.0%	32.0%
$C = 0.1$	SSD	**ALL**	62.5%	28.2%	56.7%	24.0%
		ONLY	71.4%	28.2%	60.6%	25.9%
	DETR	**ALL**	73.3%	14.1%	54.4%	25.8%
		ONLY	73.3%	14.1%	57.2%	26.8%
	YOLOV5-X	**ALL**	39.5%	62.0%	48.7%	31.0%
		ONLY	92.4%	62.0%	60.6%	39.9%
	YOLOV9-T	**ALL**	71.0%	66.0%	67.0%	36.0%
		ONLY	94.0%	66.0%	81.0%	45.0%
	YOLOV9-E	**ALL**	57.0%	62.0%	50.0%	26.0%
		ONLY	88.0%	62.0%	74.0%	38.0%
	YOLOV10-N	**ALL**	74.0%	60.0%	50.0%	23.0%
		ONLY	90.0%	60.0%	71.0%	33.0%
	YOLOV10-X	**ALL**	50.0%	64.0%	41.0%	19.0%
		ONLY	79.0%	64.0%	72.0%	32.0%

of objects in images and is more accurate than many other detectors [28], [31], it tends to lose specimens (i.e., high value of FN) and also to have a not negligible number of FP. When testing SSD on ALL and ONLY, its precision score remains stable. The number of FP slightly increases in the ALL experiment, but not dramatically.

The DETR leverages the transformer architecture, originally designed for natural language processing tasks. It has emerged as a novel NN for object detection, comprising four main components: backbone, encoder, decoder, and prediction heads [32]. In our experiments, we employed ResNet-50 as the backbone, using a CNN to learn a 2-D representation of the input image. The model flattens this representation and supplements it with positional encoding before feeding it into a transformer encoder. Subsequently, the encoded image data pass through an encoder–decoder structure and is then directed to the prediction heads. These prediction heads, based on feed-forward networks, predict either a detection class and bounding box, or a NOOBJECT class (i.e., background). We utilized the PyTorch implementation of DETR in our application, and the testing results are presented in Table II. As indicated, the recall is poor, resulting in a high number of FN. However, since it achieves the same P for ALL and ONLY, it follows model robust against empty patches.

YOLOV5 stands out for its lightweight and fast computation capabilities, demanding less computational power compared with other current state-of-the-art algorithms while maintaining comparable performance [16]. Although YOLOV5 is offered in several model sizes, in the following, we rely solely on the eXtra-large model (X) since it is the most powerful configuration to find fine-grain objects inside a frame due to its architecture. YOLOV5-X has been trained on our custom dataset by fine-tuning pretrained weights [33] on an NVIDIA RTX 3060 OC. The YOLOV5-X results are depicted in Table II. In principle, its precision on ONLY is considerably high, while it balances precision and recall on ALL.

YOLOV9 [34] introduces some innovation with respect to its predecessor. Specifically, to improve accuracy, it introduces programmable gradient information (PGI) and the generalized efficient layer aggregation network (GELAN). PGI prevents data loss and ensures accurate gradient updates, whereas GELAN optimizes lightweight models with gradient path planning. We trained YOLOV9-T and YOLOV9-E using transfer learning. Table II gives that both models achieve notable performance. YOLOV9-T demonstrates a solid robustness against background-only patches, overcoming YOLOV9-E in all the metrics. Despite the previous observation, YOLOV9-E depicts significant detection ability. Notably, YOLOV9-T showcases simultaneously a high $\text{mAP}_{0.5}$ and $\text{mAP}_{0.5}^{0.95}$, setting interesting confidences and IoU values, respectively.

We extend our analysis to the latest release of YOLO family, namely, YOLOV10 [35]. The newest architecture offers a range of model scales, but we decided to rely on YOLOV10-N and YOLOV10-X. One of the main improvements is the introduction of *consistent dual assignments*, which replaces nonmaximum suppression. Moreover, the developers introduced partial self-attention to boost the detection ability without any burdening on inference speed. As reported in Table II, YOLOV10 confirms its ability to recognize objects also in challenging environments showcasing interesting performance for both the models. Once again, despite YOLOV10-N is the lightest version, it outperforms YOLOV10-X demonstrating also a fair robustness against background-only patches. On the other hand, YOLOV10-X gains limited knowledge on BMSB detection suggesting potential overfitting on training data.

Comparing all the NNs, SSD and DETR have performed almost the same obtaining a higher precision and a lower recall. SDD tends to miss fewer BMSB inside the frame than DETR, detecting at least the $\approx 27\%$ of the total occurrences in contrast with the poor $\approx 13\%$ reached by DETR. Both SSD and DETR revealed a significant robustness against background-only pixels because their performance is almost the same in the two sets, i.e., ALL and ONLY. Moreover, DETR raised less FP than SSD,