DOI: 10.20535/1970.71(1).2026.361591
UDC 004.93

OBJECT DETECTION IMPROVEMENT WITH FREEZING LAYERS OF YOLOv11 AND YOLO26

1)M. S. Mamuta, 1)I. V. Kravchenko, 2)O. D. Mamuta
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine;
Institute of Physics, National Academy of Sciences of Ukraine, Kyiv, Ukraine
E-mail: rybalkomaryna@gmail.com; kravchenko.igor@lll.kpi.ua; mamuta@iop.kiev.ua

Object detection plays a crucial role in computer vision and many areas of modern life. An outstanding model for this task today is YOLO. However, for specific classes of objects it needs to be trained to distinguish them. For this purpose, transfer learning is used. One promising technique in transfer learning is layer freezing. However, this topic is underexplored, especially for the latest architectures of the YOLO series such as YOLOv11 and YOLO26 despite significant improvements in their architecture, namely replacement of C2f block with C3k2 and adding C2PSA block after SPPF. The question of object detection improvement with YOLOv11 and YOLO26 via layer freezing during transfer learning was addressed in the article. Experiments were conducted on two datasets: medium and small, which were automatically downloaded from the Roboflow API. The layer freezing strategy aimed to check detection improvement with freezing such key backbone layers as C3k2, C2PSA and SPPF and some layers of the neck in comparison with transfer learning without layer freezing in terms of such metrics as precision, recall, mAP@50 and mAP@50-95. All experiments were done in Google Colaboratory Pro environment with NVIDIA A100 GPU (40GB). Experiments were conducted for 50 epochs with an early stopping mechanism (patience 20), batch size 16, learning rate 0.01, optimizer MuSGD for YOLO26 and SGD for YOLOv11 and optimizer AdamW for both models. Results revealed that there is no optimal strategy, but rather empirically driven recommendations that are given in the article. For example, an effective strategy is to freeze layers of the backbone and stop freezing on C3k2 layer for both models with all optimizers. Some benefits are demonstrated by freezing the C2PSA for YOLO26. In the case of a small dataset, it is beneficial to stop freezing at the SPPF block namely for YOLO26 with optimizer MuSGD. Freezing higher layers of the neck did not show significant improvement, however, in some cases, it was beneficial. Results reveal that YOLOv11 demonstrates lower training time and per image inference latency and higher results in precision, recall, mAP@50 and mAP@50-95 metrics than YOLO26.

Keywords: *YOLO, object detection, transfer learning, computer vision.*

Introduction

One of the key tasks of computer vision is object detection [1]. Two-stage and one-stage detectors [2] can be used for this task, but the most popular today especially for real-time applications is model YOLO (You Only Look Once) [3, 4]. This model is unique in that it can process the whole image in one pass [5, 6] showing the best results in terms of speed and precision [7]. YOLO was initially trained on dataset COCO and shows high efficiency in detecting object belonging to this database [8]. For detecting new objects transfer learning technique is used where pretrained YOLO is trained to detect new objects [9, 10]. However, this process is not as simple as it seems especially in cases with imbalanced data and edge and low-power devices like IoT or embedded systems where even milliseconds make a difference. In such cases promising is method of freezing deep layers of the YOLO model [9].

Model YOLO first announced in 2016 [6] has evolved from YOLOv1 to YOLO26 released in January 2026 [11]. It consists of three main parts: backbone, neck and head. Backbone analyzes the raw image and extracts the most important features of the image. Then neck refines and strengthens these features and passes to the head. The head makes the final decision in terms of classification and regression. Although the model itself [12] and the method of fine tuning are extensively examined, the question of which layers of the model should be frozen in order to avoid overfitting and catastrophic forgetting is underexplored [13] especially for the latest YOLO architectures. For instance, in [14] backbone freezing is investigated in YOLOv5 for imbalanced data. In [15] it is studied in YOLOv8 showing interesting results for freezing intermediate layers between neck and head. However, there is no research of transfer learning with layers freezing for YOLOv11 not to mention YOLO26 despite vast improvements in architecture compared to predecessors [16].

YOLO26 architecture and YOLOv11 architecture are very similar as they both were designed by the same Ultralytics company. Both of them are advanced, next-generation extensions of the

YOLOv8 architecture. They were designed to improve efficiency, accuracy, and deployment speed. Architecture of YOLOv11 is shown in Fig. 1 [17].

YOLOv11 got significant improvement due to replacement of C2f block with Cross Stage Partial with kernel size 2 block (C3k2) and utilizing new Cross Stage Partial with Spatial Attention (C2PSA) block after Spatial Pyramid Pooling – Fast (SPPF) block (Fig. 1).

C3k2 block improves real-time object detection due to flexibility in feature extraction [16]. C2PSA realizes attention mechanism which enables concentration on key region in the image, enhancing detection of small and occluded objects.

In YOLO26 SPPF block includes a shortcut connection, which improves gradient flow and stabilizes optimization.

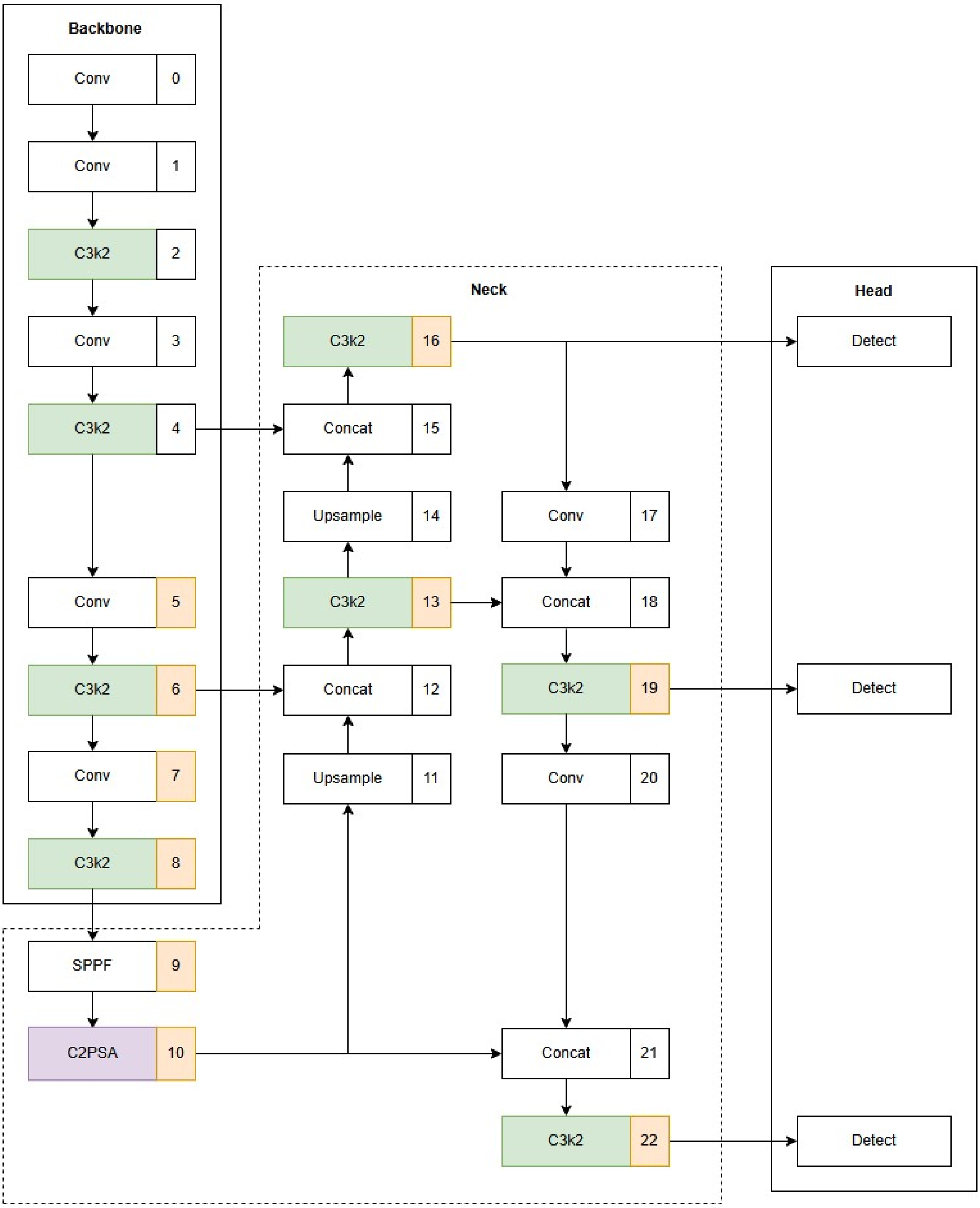

Fig. 1. YOLOv11 architecture [17]

Another difference is in the C3k2 block before the detect block that is used in combination with attention mechanism in the PSABlock module. Box regression is used directly instead of Distributed Focal Loss (DFL) block. This simplifies training and enables more confident predictions. YOLO26 eliminates the need for Non-Maximum Suppression (NMS) by producing final detections directly from the network, avoiding overlapping predictions and post-processing, thus introducing End-to-End NMS-free inference. New optimizer MuSGD is used for better convergence and training stability. YOLO26 uses score-based prediction selection instead of IoU-based filtering [11].

Materials and methods

Ultralytics gives certain recommendations for transfer learning with YOLO26 namely for fine-tuning [18].

Taking into account these recommendations and model architecture the following strategy was formed: preserve weights of C2PSA block for YOLOv11 and YOLO26. Freeze first 10 layers for this purpose. Check whether freezing of SPPF layer gives better results (freeze first 9 layers). Make sure that freezing of C3k2 layers and layers in-between them inside backbone give some benefits (freeze first 8, 7, 6 and then 5 layers) and decide where is better to stop freezing. Freeze some layers of the neck, namely 13, 16, 19 and 22 layer and check whether it gives any improvement. Layers which were frozen are indicated in Fig. 1.

In model YOLOv11 the last layer 'model.23.dfl.conv.weight' was frozen automatically for every training.

Version nano was chosen as the lightest and fastest one.

Datasets were automatically downloaded from the Roboflow API. First dataset consists 1000 images [19], second – 140 [20].

Experiments were conducted for 50 epochs with an early stopping mechanism (patience 20), batch size 16, learning rate 0.01, optimizer MuSGD for YOLO26 and SGD for YOLOv11 and AdamW for both models were used. All training and evaluation procedures were executed on Google Colaboratory Pro platform utilizing NVIDIA A100 GPU (40GB).

Model settings and hyperparameters are summarized in Table 1.

Table 1. Model settings and hyperparameters

Category	Parameter	Value
Basic	Epochs	50
	Batch size	16
	Image size	640×640 pixels
	Early Stopping Patience	20 epochs
Optimization	Optimizer	AdamW / SGD / MuSGD
	Initial learning rate (lr0)	0.01
	Final learning rate (lrf)	0.01
	Momentum	0.937
System	GPU	NVIDIA A100 (40GB)

Model performance was estimated with standard metrics as precision, recall, mAP@50 and mAP@50-95. Precision shows true positives among all positive samples. Recall demonstrates true positives among all positives in the particular class. mAP@50 is a measure of mean average precision at a 50% IoU. mAP@50-95 provides more precise estimation across a range from 50 % to 95 % IoU.

Computational efficiency was estimated through per-image inference latency in milliseconds and training time. Computational complexity was evaluated through trainable parameter count and GFLOPs.

Experiments with learning rates 0.01 and 0.001 have shown the same results in many cases.

Results and discussion

Results of training YOLOv11 and YOLO26 with optimizer AdamW on dataset 1 are shown in Table 2.

Results of training YOLOv11 with optimizer SGD and YOLO26 with optimizer MuSGD on dataset 1 are shown in Table 3.

Although both models, YOLOv11 and YOLO26, have good results without freezing layers during learning, there is still room for improvement especially in combination of such metrics as mAP@50-95 and recall. For YOLOv11 with optimizer AdamW the best result relative to no-freezing training gives freezing the first 6 layers for all metrics. In case of YOLOv11 with optimizer SGD the best result is obtained for all metrics by freezing the first 5 layers. YOLOv11 (optimizer AdamW, 6 layers frozen) relative to YOLOv11 (optimizer SGD, 5 layer frozen) gives higher recall sharing the same mAP@50 value and slightly lower precision.

For YOLO26 with optimizer AdamW the best result relative to no-freezing training gives freezing the first 10 layers as recommended by Ultralytics [10] for such metrics as recall and mAP@50-95.

Table 2. Results of training YOLOv11 and YOLO26 with optimizer AdamW on dataset 1

Model	Optimizer AdamW							
	Precision	Recall	mAP@50	mAP@50-95	Per image inference latency, ms	Training time, hours	Trainable parameter count	GFLOPs
YOLOv11n, no freezing	**0.969**	**0.943**	**0.975**	**0.789**	0.9	0.091	2,582,347	6.3
YOLOv11n, 22	0.904	0.865	0.926	0.68	0.7	0.064	2,582,347	6.3
YOLOv11n, 19	0.968	0.929	0.957	0.737	1.4	0.066	2,582,347	6.3
YOLOv11n, 16	0.974	0.94	0.973	0.784	0.8	0.070	2,582,347	6.3
YOLOv11n, 13	0.972	0.944	0.969	0.795	0.6	0.071	2,582,347	6.3
YOLOv11n, 10	0.98	0.954	0.987	0.806	0.6	0.074	2,582,347	6.3
YOLOv11n, 9	0.977	0.959	0.98	0.799	0.6	0.077	2,582,347	6.3
YOLOv11, 8	0.983	0.944	0.978	0.791	0.9	0.076	2,582,347	6.3
YOLOv11, 7	0.961	0.964	0.977	0.796	1.4	0.077	2,582,347	6.3
YOLOv11, 6	**0.974**	**0.964**	**0.981**	**0.806**	0.6	0.078	2,582,347	6.3
YOLOv11, 5	0.969	0.952	0.972	0.787	0.6	0.082	2,582,347	6.3
YOLO26n, no freezing	**0.959**	**0.938**	**0.984**	**0.799**	1.6	0.118	2,375,031	5.2
YOLO26n, 22	0.739	0.761	0.825	0.575	1.4	0.090	2,375,031	5.2
YOLO26n, 19	0.916	0.919	0.965	0.73	1.5	0.095	2,375,031	5.2
YOLO26n, 16	0.974	0.914	0.976	0.769	1.5	0.098	2,375,031	5.2
YOLO26n, 13	0.905	0.916	0.968	0.792	0.8	0.101	2,375,031	5.2
YOLO26n, 10	0.952	**0.954**	0.984	**0.8**	1.5	0.102	2,375,031	5.2
YOLO26n, 9	0.95	0.929	0.967	0.795	2.2	0.107	2,375,031	5.2
YOLO26, 8	0.938	0.923	0.96	0.787	1.5	0.106	2,375,031	5.2
YOLO26, 7	0.98	0.934	0.981	0.797	2.3	0.108	2,375,031	5.2
YOLO26, 6	0.961	0.954	0.976	0.792	0.7	0.109	2,375,031	5.2
YOLO26, 5	0.974	0.943	0.978	0.79	2.2	0.111	2,375,031	5.2

Table 3. Results of training YOLOv11 with optimizer SGD and YOLO26 with optimizer MuSGD on dataset 1

Model	Optimizer SGD / MuSGD							
	Precision	Recall	mAP@50	mAP@50-95	Per image inference latency, ms	Training time, hours	Trainable parameter count	GFLOPs
YOLOv11n, no freezing	**0.988**	**0.944**	**0.975**	**0.796**	0.6	0.087	2,582,347	6.3
YOLOv11n, 22	0.888	0.889	0.93	0.673	0.6	0.062	2,582,347	6.3
YOLOv11n, 19	0.974	0.914	0.956	0.735	0.6	0.066	2,582,347	6.3
YOLOv11n, 16	0.979	0.934	0.959	0.767	0.6	0.067	2,582,347	6.3
YOLOv11n, 13	0.989	0.954	0.977	0.794	0.6	0.070	2,582,347	6.3
YOLOv11n, 10	0.993	0.954	0.979	0.8	0.6	0.071	2,582,347	6.3
YOLOv11n, 9	0.996	0.939	0.981	0.797	0.7	0.074	2,582,347	6.3
YOLOv11, 8	0.993	0.954	0.98	0.789	0.7	0.075	2,582,347	6.3
YOLOv11, 7	0.964	0.98	0.979	0.789	0.9	0.078	2,582,347	6.3
YOLOv11, 6	0.989	0.944	0.975	0.799	0.6	0.078	2,582,347	6.3
YOLOv11, 5	**0.989**	**0.95**	**0.981**	**0.801**	0.7	0.079	2,582,347	6.3
YOLO26n, no freezing	**0.952**	**0.913**	**0.96**	**0.797**	1.7	0.155	2,375,031	5.2
YOLO26n, 22	0.786	0.726	0.827	0.579	1.6	0.103	2,375,031	5.2
YOLO26n, 19	0.955	0.87	0.967	0.72	0.7	0.110	2,375,031	5.2
YOLO26n, 16	0.951	0.919	0.973	0.764	0.7	0.116	2,375,031	5.2
YOLO26n, 13	0.966	0.934	0.976	0.792	2.2	0.124	2,375,031	5.2
YOLO26n, 10	0.968	0.923	0.972	0.783	1.4	0.128	2,375,031	5.2

YOLO26n, 9	0.967	0.934	0.981	0.793	0.7	0.136	2,375,031	5.2
YOLO26, 8	0.932	**0.968**	**0.983**	**0.798**	1.5	0.137	2,375,031	5.2
YOLO26, 7	0.954	0.934	0.972	0.786	1.9	0.140	2,375,031	5.2
YOLO26, 6	0.951	0.949	0.985	0.797	1.6	0.145	2,375,031	5.2
YOLO26, 5	0.955	0.954	0.968	0.789	1.6	0.143	2,375,031	5.2

However, for YOLO26 with optimizer MuSGD the best result is obtained for metrics recall, mAP@50 and mAP@50-95 by freezing the first 8 layers, but there is a decrease in precision. YOLO26 (optimizer MuSGD, 8 layers frozen) relative to YOLO26 (optimizer AdamW, 10 layers frozen) gives higher recall.

So, the best results with no-freezing technique demonstrates YOLOv11 with optimizer SGD in terms of quality and speed. Freezing layers benefits the most YOLOv11 with optimizer AdamW and YOLO26 with optimizer MuSGD. In addition, freezing first 6 layers of YOLOv11 with optimizer AdamW gives the best results among all others in terms of the combination of three metrics: recall, mAP@50 and mAP@50-95.

Results of training YOLOv11 and YOLO26 with optimizer AdamW on dataset 2 are shown in Table 4.

Table 4. Results of training YOLOv11 and YOLO26 with optimizer AdamW on dataset 2

Model	**Optimizer AdamW**							
	Precision	**Recall**	**mAP@50**	**mAP@50-95**	**Per image inference latency, ms**	**Training time, hours**	**Trainable parameter count**	**GFLOPs**
YOLOv11n, no freezing	**0.993**	**0.947**	**0.99**	**0.757**	0.6	0.023	2,582,347	6.3
YOLOv11n, 22	0.949	1	0.993	0.757	0.6	0.017	2,582,347	6.3
YOLOv11n, 19	0.863	0.993	0.944	0.713	0.6	0.017	2,582,347	6.3
YOLOv11n, 16	0.949	1	0.99	0.71	0.6	0.018	2,582,347	6.3
YOLOv11n, 13	1	0.995	0.995	0.709	0.7	0.018	2,582,347	6.3
YOLOv11n, 10	0.999	1	0.995	0.754	0.6	0.019	2,582,347	6.3
YOLOv11n, 9	**0.995**	**1**	**0.995**	**0.76**	0.9	0.020	2,582,347	6.3
YOLOv11, 8	**1**	**0.995**	**0.995**	0.757	0.7	0.020	2,582,347	6.3
YOLOv11, 7	1	0.986	0.995	0.745	0.7	0.021	2,582,347	6.3
YOLOv11, 6	**0.998**	**1**	**0.995**	0.75	0.7	0.021	2,582,347	6.3
YOLOv11, 5	0.947	0.948	0.993	0.745	0.6	0.021	2,582,347	6.3
YOLO26n, no freezing	**0.863**	**0.895**	**0.934**	**0.709**	0.8	0.029	2,375,031	5.2
YOLO26n, 22	1	0.676	0.862	0.739	1.1	0.022	2,375,031	5.2
YOLO26n, 19	0.795	0.818	0.867	0.656	1.1	0.024	2,375,031	5.2
YOLO26n, 16	0.763	0.846	0.884	0.701	0.7	0.025	2,375,031	5.2
YOLO26n, 13	0.828	0.842	0.909	0.737	0.8	0.026	2,375,031	5.2
YOLO26n, 10	0.852	**0.947**	**0.962**	**0.726**			2,375,031	5.2
YOLO26n, 9	0.774	0.947	0.923	0.724	0.7	0.026	2,375,031	5.2
YOLO26, 8	0.837	0.895	0.936	0.779	0.8	0.027	2,375,031	5.2
YOLO26, 7	**0.944**	0.881	0.937	**0.728**	0.7	0.027	2,375,031	5.2
YOLO26, 6	0.775	0.947	0.929	0.743	0.7	0.028	2,375,031	5.2
YOLO26, 5	0.937	0.78	0.94	0.757	0.8	0.028	2,375,031	5.2
YOLO26, 4	0.942	0.857	0.946	0.763	0.7	0.028	2,375,031	5.2

Results of training YOLOv11 with optimizer SGD and YOLO26 with optimizer MuSGD on dataset 2 are shown in Table 5.

It is worth mentioning that early stopping mechanism was activated in a few cases for this dataset and obvious data outliers were excluded. For example, for no-freezing training YOLO26 with optimizer MuSGD such values were obtained with early stopping mechanism: precision – 0.00333, recall – 1, mAP@50 – 0.775, mAP@50-95 – 0.643. Additional training was made for all 50 epochs and result was placed in table 5.

For YOLOv11 with optimizer AdamW the best result relative to no-freezing training gives freezing of

6, 8 or 9 first layers for almost all metrics. In case of YOLOv11 with optimizer SGD the best result is obtained for all metrics by freezing 6 first layers. YOLOv11 (optimizer AdamW, 8/9 layer frozen) relative to YOLOv11 (optimizer SGD, 6 layer frozen) gives higher precision but loses in mAP@50-95 value.

For YOLO26 with optimizer AdamW results are very diverse and there is no best variant. For example, freezing of 7 layers gives rise in precision and mAP@50-95, but at the same time loss in recall. Freezing of 10 layers benefits such metrics as recall, mAP@50 and mAP@50-95, but gains decay in precision. For YOLO26 with optimizer MuSGD the best result is obtained for all metrics with freezing 9 first layers. YOLO26 (optimizer MuSGD) relative to YOLO26 (optimizer AdamW) gives more stable results and better overall performance.

Table 5. Results of training YOLOv11 with optimizer SGD and YOLO26 with optimizer MuSGD on dataset 2

Model	**Optimizer SGD / MuSGD**							
	Precision	**Recall**	**mAP@50**	**mAP@50-95**	**Per image inference latency, ms**	**Training time, hours**	**Trainable parameter count**	**GFLOPs**
YOLOv11n, no freezing	**0.95**	**0.998**	**0.99**	**0.782**	0.6	0.022	2,582,347	6.3
YOLOv11n, 22	0.995	1	0.995	0.771	0.6	0.016	2,582,347	6.3
YOLOv11n, 19	1	0.619	0.869	0.68	0.7	0.013	2,582,347	6.3
YOLOv11n, 16	0.944	0.894	0.952	0.696	0.6	0.017	2,582,347	6.3
YOLOv11n, 13	1	0.999	0.995	0.77	0.7	0.018	2,582,347	6.3
YOLOv11n, 10	0.991	1	0.995	0.761	0.6	0.018	2,582,347	6.3
YOLOv11n, 9	1	0.99	0.995	0.764	0.6	0.017	2,582,347	6.3
YOLOv11, 8	1	0.771	0.857	0.732	0.8	0.015	2,582,347	6.3
YOLOv11, 7	1	0.999	0.995	0.761	0.7	0.020	2,582,347	6.3
YOLOv11, 6	**0.988**	**1**	**0.995**	**0.801**	0.6	0.020	2,582,347	6.3
YOLOv11, 5	1	0.997	0.995	0.788			2,582,347	6.3
YOLO26n, no freezing	**0.893**	**0.881**	**0.949**	**0.714**	0.8	0.023	2,375,031	5.2
YOLO26n, 22	0.854	0.926	0.961	0.734	0.7	0.025	2,375,031	5.2
YOLO26n, 19	0.943	0.874	0.966	0.753	0.8	0.027	2,375,031	5.2
YOLO26n, 16	1	0.103	0.825	0.64	0.7	0.020	2,375,031	5.2
YOLO26n, 13	0.76	0.895	0.913	0.742	0.7	0.029	2,375,031	5.2
YOLO26n, 10	0.751	0.947	0.934	0.734	0.7	0.030	2,375,031	5.2
YOLO26n, 9	**0.941**	**0.895**	**0.951**	**0.746**	0.8	0.031	2,375,031	5.2
YOLO26, 8	0.941	0.836	0.961	0.752	0.8	0.031	2,375,031	5.2
YOLO26, 7	0.944	0.885	0.966	0.783	0.8	0.032	2,375,031	5.2
YOLO26, 6	0.818	0.945	0.941	0.744	0.7	0.032	2,375,031	5.2
YOLO26, 5	0.89	0.842	0.953	0.751	0.7	0.033	2,375,031	5.2

Conclusions

YOLOv11 demonstrates better overall performance despite greater GFLOPs and number of trainable parameters, especially in training with optimizer AdamW on medium dataset and with optimizer SGD on small dataset.

The best freezing strategy is to freeze layers of the backbone and stop freezing at the C3k2 layer. In training on medium dataset for YOLOv11 with optimizer AdamW it is 6th layer, for YOLO26 with optimizer MuSGD it is 8th layer. In training on small dataset for YOLOv11 with optimizer AdamW it is 6th or 8th layer and for YOLOv11 with optimizer SGD it is 6th layer.

In some cases, it is beneficial to stop freezing on SPPF block namely for YOLO26 with optimizer MuSGD in training on small dataset. Freezing higher layers of the neck demonstrates some model performance improvement on medium dataset namely for YOLOv11 with optimizer SGD and YOLO26 with optimizer MuSGD, however not as beneficial as freezing layers of the backbone or SPPF layer.

YOLOv11 demonstrates lower training time and per image inference latency and higher results in precision, recall, mAP@50 and mAP@50-95 metrics than YOLO26.

Further improvement is envisioned in combining layer freezing with hyperparameter optimization.

References

[1] Katsushi Ikeuchi. Computer Vision. Springer, 2021, 1406. DOI: 10.1007/978-3-030-63416-2

[2] M. A. Trigka, E. Dritsas, "Comprehensive Survey of Machine Learning Techniques and Models for Object Detection", *Sensors*, p. 214, 2025. DOI: 10.3390/s25010214

[3] M. K. Jahan, F. I. Bhuiyan, A. Amin et al, "Enhancing the YOLOv8 model for realtime object detection to ensure online platform safety", *Scientific Reports*, vol. 15, p. 21167, 2025. DOI: 0.1038/s41598-025-08413-4

[4] M. L. Ali, Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection", *Computers*, vol. 13(12), p. 336, 2024. DOI: 10.3390/computers13120336

[5] J. Redmon, S. Divvala, S. Girshick, A. Farhadi, "You only look once: Unified, real-time object detection", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016. DOI: 10.1109/CVPR.2016.91

[6] A. Yılmaz, Y. Yurtay, N. Yurtay, "YOLO: Development of a Real-Time Object Detection Model for the Detection of Secretly Cultivated Plants", *Applied Sciences*, vol. 15, p. 2718, 2025. DOI: 0.3390/app15052718

[7] H. V. Nguyen, J. H. Bae, Y. E. Lee, H. S. Lee, K. R. Kwon, "Comparison of Pre-Trained YOLO Models on Steel Surface Defects Detector Based on Transfer Learning with GPU-Based Embedded Devices", *Sensors*, vol. 22(24), p. 9926. DOI: 10.3390/s22249926

[8] T. Y. Lin, M. Maire, S. Belongie et al, "Microsoft COCO: Common Objects in Context", Computer Vision – ECCV 2014. 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, vol. 8693, pp. 740–755, Springer Cham, 2014. DOI: 10.1007/978-3-319-10602-1_48

[9] G. S. R. Machiraju, K. A. Kumari and S. K. Sharif, "Object Detection and Tracking for Community Surveillance using Transfer Learning", in *6th International Conference on Inventive Computation Technologies* (ICICT), Coimbatore, India, pp. 1035-1042, 2021. DOI: 10.1109/ICICT50816.2021.9358698

[10] E. Pandilova, M. Petrov, V. Spasev, I. Dimitrovski, I. Kitanovski, "Transfer Learning with Yolo for Object Detection in Remote Sensing", ICT Innovations 2024. TechConvergence: AI, Business, and Startup Synergy. ICT Innovations 2024. Communications in Computer and Information Science, vol. 2436, pp. 121–135, 2024, Springer Cham. DOI: 10.1007/978-3-031-86162-8_9

[11] P. Hidayatullah, R. Tubagus, "YOLO26: A Comprehensive Architecture Overview and Key Improvements. DOI: 10.48550/arXiv.2602.14582.

[12] L. T. Ramos, A. D. Sappa, "A Decade of You Only Look Once (YOLO) for Object Detection: A Review", *IEEE Access*, vol. 13, pp. 192747-192794, 2025. DOI: 10.1109/ACCESS.2025.3630988

[13] A. D. Dobrzycki, A. M. Bernardos, J. R. Casar, "An Analysis of Layer-Freezing Strategies for Enhanced Transfer Learning in YOLO Architectures", *Mathematics*, vol. 13. p. 2539, 2025. DOI: 10.3390/math13152539

[14] A. N. Yuma. Rafi, M. Yusuf, "Improving Vehicle Detection in Challenging Datasets: YOLOv5s and Frozen Layers Analysis", *International Journal of Informatics and Computation*, vol. 5, no. 2. pp. 31–45, 2023. DOI: 10.35842/ijicom.v5i2.64

[15] Y. Feng, D. A. Sumiyoshi, "Application of Three Transfer Learning Methods Based on YOLOv8 for Object Recognition in Architectural Drawings", Proceedings of the ASim Conference 2024: 5th Asia Conference of IBPSA, Osaka, Japan, 8–10 November 2024, vol. 5, pp. 443–450, 2024. DOI: 10.69357/asim2024.1256

[16] R. Khanam, M. Hussain, YOLOv11: An Overview of the Key Architectural Enhancements. DOI: 10.48550/arXiv.2410.17725. *Режим доступу*: https://arxiv.org/abs/2410.17725

[17] Ultralytics YOLO26 object detection model with P3/8 - P5/32 outputs. [Online]. Available: https://github.com/ultralytics/ultralytics/blob/main/ultralytics/cfg/models/26/yolo26.yaml

[18] YOLO26 Training Recipe. [Online]. Available: https://docs.ultralytics.com/guides/yolo26-training-recipe/#start-simple

[19] Drone detection dataset. [Online]. Available: https://universe.roboflow.com/testing-pde8o/drone-detection-hp47t/dataset/1

[20] Fixed Combined Data v5. [Online]. Available: https://universe.roboflow.com/mariam-aljaberi/target-detection-shahed/dataset/3

[1)]М. С. Мамута, [1)]І. В. Кравченко, [2)]О. Д. Мамута
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
Інститут фізики, Національна академія наук України, Київ, Україна

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВИЯВЛЕННЯ ОБ'ЄКТІВ МОДЕЛЯМИ YOLOV11 ТА YOLO26 ВНАСЛІДОК ЗАМОРОЖУВАННЯ ШАРІВ

Виявлення об'єктів відіграє ключову роль для систем комп'ютерного зору. Однією з провідних моделей для розв'язання цієї задачі на сьогодні є модель YOLO. Для коректного виявлення специфічних класів об'єктів модель YOLO потребує додаткового навчання. З цією метою застосовується метод перенесення навчання. Перспективним підходом у його межах є заморожування шарів. Однак це питання залишається недостатньо дослідженим, особливо для новітніх архітектур серії YOLO, таких як YOLOv11 та YOLO26, попри суттєві вдосконалення, зокрема заміну блоку C2f на C3k2 та додавання блоку C2PSA після SPPF. У статті розглянуто проблему підвищення ефективності виявлення об'єктів моделями YOLOv11 та YOLO26 шляхом застосування заморожування шарів під час перенесення навчання. Експерименти проводилися з використанням двох наборів даних – середнього та малого розміру, які було автоматично завантажено через платформу Roboflow. Стратегія заморожування передбачала аналіз впливу заморожування ключових шарів основної частини таких як C3k2, C2PSA, SPPF, а також окремих шарів шиї, порівняно з підходом без заморожування. Оцінювання здійснювалося за метриками точності, повноти, mAP@50 та mAP@50–95. Усі експерименти виконано в середовищі Google Colaboratory Pro з використанням графічного процесора NVIDIA A100 (40 ГБ пам'яті). Навчання тривало 50 епох із застосуванням механізму ранньої зупинки у разі відсутності покращення протягом 20 епох, розмір пакета становив 16 зразків, швидкість навчання – 0,01. Для моделі YOLO26 використовувався оптимізатор MuSGD, для YOLOv11 – SGD, а також оптимізатор AdamW для обох моделей. Отримані результати свідчать про відсутність універсальної оптимальної стратегії заморожування шарів; натомість сформульовано емпірично обґрунтовані рекомендації. Зокрема, доцільно заморожувати шари основної частини із зупинкою на блоці C3k2 для обох моделей незалежно від вибору оптимізатора. Для YOLO26 додаткові переваги демонструє заморожування до блоку C2PSA включно. У випадку малого набору даних ефективною є стратегія заморожування до блоку SPPF включно, зокрема для YOLO26 з оптимізатором MuSGD. Заморожування шарів шиї вищого порядку не надає суттєвого покращення ефективності, хоча в окремих випадках може підвищувати значення окремих метрик. Також встановлено, що модель YOLOv11, незважаючи на більшу кількість GFLOPs і параметрів тренування, демонструє менший час навчання та нижчу затримку інференсу одного зображення. Крім того, YOLOv11 забезпечує вищі значення метрик точності, повноти, mAP@50 та mAP@50–95 порівняно з YOLO26 для обох наборів даних.

Ключові слова: YOLO, виявлення об'єктів, перенесення навчання, комп'ютерний зір.

Надійшла до редакції
22 березня 2026 року

Рецензовано
14 квітня 2026 року