REAL-TIME OBJECT DETECTION OF SIMPLE DRAWINGS USING YOLO11 ON CONSTRAINED DATASETS

LADISLAV HRIC[1], JAN BACIK[1], DANIELA PERDUKOVA[1]

[1]Dept. of Electrical Engineering and Mechatronics, FEI, Technical University of Kosice, Kosice, Slovak Republic

DOI: 10.17973/MMSJ.2025_03_2024118

ladislav.hric@tuke.sk

Object detection as a fundamental computer vision task has applications across different industries. One particular application is the detection and recognition of drawings, which often relies on convolutional neural networks. Developing a robust model for such an application typically requires a large dataset containing images of the target object from various perspectives and conditions. However, in many real-world cases, this is not possible, and the model must be trained on a small dataset. This paper addresses this challenging task by training a model for real-time detection of drawings using a constrained dataset. We leverage the state-of-the-art YOLO11 model as a foundation, known for its balance of speed and accuracy in real-time object detection. Different YOLO11 variants are trained under various configurations to evaluate their performance. The results of the study demonstrate the robustness and effectiveness of models for the real-time detection of drawings in cases when data resources are limited.

KEYWORDS

Object Detection, YOLO11, Small Datasets, Drawing Recognition, Deep Learning

1 INTRODUCTION

Object detection is a widely discussed topic due to its broad application in critical industries such as healthcare, security, manufacturing, and more. The demand for these algorithms has existed for a long time, but their applications have come to the forefront only in recent years. One of the main reasons is the rapid increase in computational power over time. Nowadays, powerful GPUs and CPUs provide sufficient computational power for a wide range of machine-learning applications [Kuric 2021, Kuric 2022, Hortobágyi 2021, Hu 2022].

A convolutional neural network is a type of deep learning neural network architecture extensively used in computer vision tasks because of its effectiveness in learning features from images. Such algorithms can recognise everything from simple lines and curves to complex real-world objects like people, dogs, and cats, essentially any patterns [Ji 2024, Komarskiy 2024, Narwaria 2024] the model is trained on. The quality and efficiency of neural networks depend on their structure. There are many versions that can be used as starting points [Bačík 2015, Hamzah 2024] for further applications and improvements. In the field of real-time object detections, the YOLO (You Only Look Once) models from Ultralytics are state-of-the-art, and known for their balance of speed and accuracy [Liu 2024]. The latest released version is YOLO11 which is suitable for many tasks. This paper focuses on the detection tasks supported by YOLO's five different variants: *n*, *s*, *m*, *l* and *x*. Each variant offers a unique balance between speed and accuracy to cover a wide range of applications. The *n* variant has the smallest (lightweight) architecture, while the *x* variant has the largest.

To build a robust model, it is crucial to focus on collecting a large amount of data for training. As mentioned in work [Sanchez 2021, Salman 2023], many real-world scenarios require working with limited data resources. One such example is the detection of rare diseases, where data scarcity necessitates the use of preprocessing and augmentation techniques to enhance the training process. Similar challenges were met in paper [Karthikeyan 2023] investigating the detection of spirals drawn on paper to detect and classify stages of Parkinson's disease. Drawing detection has a wide range of applications, with study [Luo 2022], which utilizes neural networks to recognize specific electrical symbols, being just one example among many.

In this study, we emulate such a situation and develop a model for the real-time detection trained on a small dataset. The dataset contains three basic drawings: cars, flowers and houses. Although these drawings are simple for humans to recognise, they still present a challenging task for computer vision due to their abstraction and the combination of different curves. Models capable of detecting basic drawings have potential in many applications.

The main contribution of this paper is the evaluation of YOLO11-based models for a real-time drawing detection application trained on a small dataset.

2 RELATED WORKS

This section examines related works which compare and use the YOLO models for the detection of drawings or simple features trained on small datasets.

A study [Karthikeyan 2023] proposes a system to detect and classify Parkinson's disease (PD) stages leveraging drawing models and the universal YOLO object recognition framework. This work collected data from 70 patients with mild PD, 70 patients with moderate PD, 70 patients with severe PD, and 70 healthy controls. The algorithm focuses on separating different stages of PD from the drawings of certain shapes. The work also compared the CNN model, the YOLOv8-n model, and the YOLOv8-x model, where the achieved accuracy of 94% had the model based on YOLOv8-x. The results of the study show the potential of using drawing patterns and the YOLO for the detection and classification of Parkinson's disease.

The study [Wiratchawa 2022] compared YOLOv4, YOLOv5 and EfficientDet models for detecting prostate cancer lesions using a small training dataset. According to the results, EfficientDet-D5 achieved the highest performance with 45.4% precision, 52.6% recall, and 52.6% accuracy (TPR). However, the study also showed that YOLOv4 and YOLOv5 achieved higher precision when trained with additional data from a public dataset.

Another work [Nawae 2023] compares various YOLO models for sperm and impurity detection. The models were trained on the original small dataset, which consisted of 256 images, as well as on a dataset leveraging the proposed augmentation. Comparing YOLOv5, YOLOv6, YOLOv7 and YOLOv8s, results show that YOLOv8s achieved the highest accuracy on the augmented dataset. The study [Kumar 2024] also employs a YOLO-based model trained on a small number of images for brain tumour detection. Training conducted on 400 images achieved 94.55% precision, 90.18% accuracy, and 94.23% specificity.

An important aspect of image preprocessing and data augmentation is their value when working with small datasets. The study [Xu 2023] examines the influence of different types of augmentations on a trained model. This study demonstrates promising results when using the mosaic data enhancement method.

3 METHODOLOGY

This section describes the creation of the dataset, followed by image preprocessing, the extension of the original dataset using augmentation techniques, and concludes with model training and evaluation.

3.1 Data collection

In this work, we use a custom dataset composed of simple drawings categorized into three classes: cars, flowers, and houses (Fig. 1). These drawings represent combinations of curves that the model must recognize to accurately detect and classify them. For each class, 45 training images, 25 validation images, and 30 test images were collected, forming the core of a small dataset. Various preprocessing and augmentation techniques were applied to these images to expand the dataset.

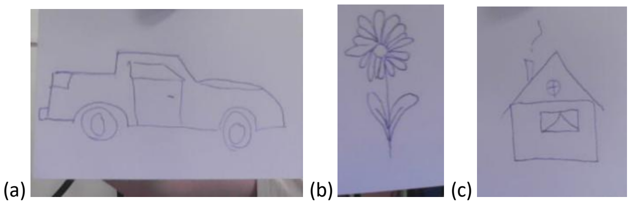

Figure 1. Dataset classes: (a) cars, (b) flowers, (c) houses

3.2 Image preprocessing

Image preprocessing is the process of manipulating input data to speed up model inference and training. The choice of specific methods depends on the application. The most well-known techniques of image preprocessing include resizing, orientation adjustment, and colour correction, among others. Unlike augmentation, this process does not expand the original dataset. In this work, resizing was the first preprocessing technique applied. This is a crucial step in computer vision that helps models learn faster on smaller images. Also, plenty of deep learning model architectures require images to be the same size. Therefore, this step is often necessary because collected images vary in size. In this work, all images were stretched to 640x640.

As the second step, the auto-orientation was applied. This technique ensured all images in the dataset had the same orientation. The model trained on properly oriented pictures, avoiding confusion caused by different orientations.

Edge detection is crucial for object detection. Contrast adjustment, as a preprocessing step, enhances edges, making it easier for neural networks to understand them. Therefore, it was chosen as the next preprocessing step.

Subsequently, all images in the dataset were converted to grayscale. This simplifies the training process because the model can focus on shapes and omit colour dependencies. Since our dataset includes only drawings, and the goal is to recognize patterns, colour information is useless in our case. Grayscale conversion is the final preprocessing step, and all applied steps are illustrated in Fig. 2.

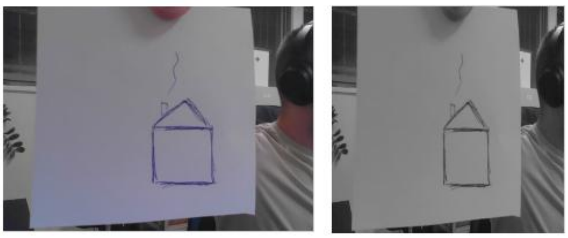

Figure 2. Image preprocessing: on the left is original picture; on the right is picture after applying resizing, auto-orientation, contrast adjustment, and grayscale conversion

3.3 Image augmentation

Image augmentation is the manipulation of existing images to extend the original dataset. It is done by creating different versions of the original pictures, which helps the model generalize better and avoid overfitting. This is beneficial in the real world, where it may be difficult to obtain images of an object from different angles, distances, lighting conditions, or other scenarios. This is a crucial step in our case, as our dataset consists of a small number of pictures. In this work, several augmentation techniques were selected.

Rotation is an important augmentation step in cases where the input picture is not fixed and can be rotated. An example of such a case is object detection used in mobile phones. In our case, pictures on the model's input can be rotated, so this step is important. In this work, the augmentation step generates new pictures with rotations between -10° and +10°. This rotation is demonstrated in Fig. 3.

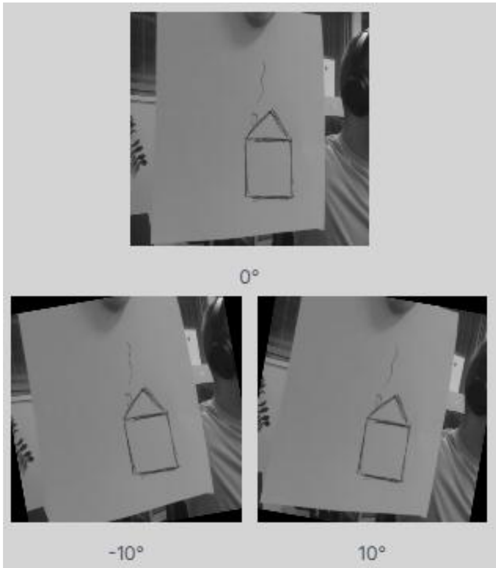

Figure 3. Rotation augmentation: upper is original image with 0° rotation; below are augmented images with rotations of ±10°

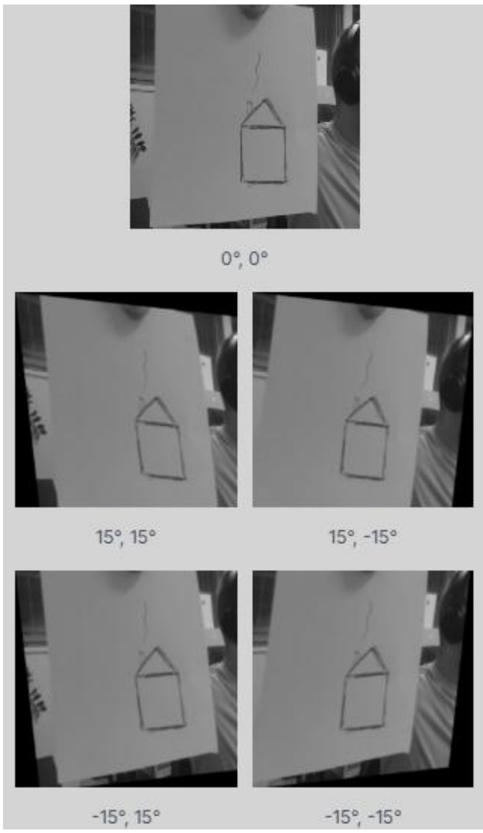

Figure 4. Shear adjustment: original image with 0° shear, followed by combinations of ±15° horizontal and vertical shear

Shear adjustment was chosen to modify the perspective of the image, which helps the model understand changes in pitch and yaw. We set the shear adjustment in a range from ±15° horizontally and ±15° vertically. The shear adjustment is shown in Fig. 4.

In many cases, even if we have the possibility to collect pictures of a target object from different angles, it may still be hard to change the lighting conditions. Therefore, extending the dataset with pictures that have adjusted brightness helps the model become more robust in various lighting conditions. In our case, we set the brightness adjustment between -25% and +25%. This adjustment is demonstrated in Fig. 5.

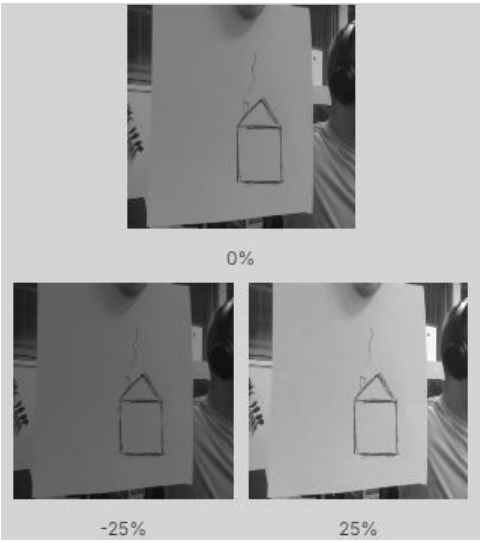

Figure 5. Brightness adjustment: original image with normal brightness, followed by examples with -25% and +25% brightness adjustments

Adding noise to images is beneficial due to its impact on reducing overfitting. In this step, random pixels are converted to completely black or white, as shown in Fig. 6. Our dataset used this augmentation step by changing up to 1.85% of the pixels.

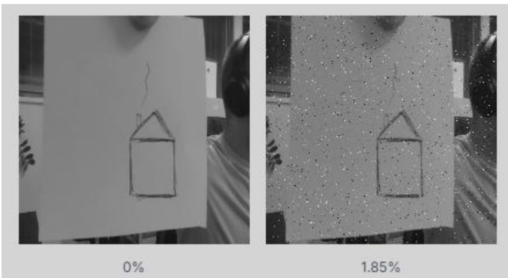

Figure 6. Noise addition: original image followed by an example with up to 1.85% of pixels converted to black or white

3.4 *Final dataset overview*

The final dataset, after applying preprocessing and augmentation, consists of 441 images in the training set and 108 images in the validation set. To evaluate the trained model on images that it did not see during the training process, we also collected 30 images of cars, 30 images of flowers, and 30 images of houses.

3.5 *Model training*

YOLO is a state-of-the-art model [Redmon 2016], considering both speed and accuracy. The latest iteration in the Ultralytics YOLO series of real-time object detectors is YOLO11 [Khanam 2024]. It was released in 2024 and offers higher accuracy and speed than its previous versions. In this study, we propose different models based on YOLO11 and compare their performance. The output model should be suitable for real-time detection of drawings, considering both speed and accuracy.

The YOLO11 series offers models specialized for different computer vision tasks. In this paper, we focus on the detection task and therefore choose from the n, s, m, l and x model variants. First, we prepared models based on all variants and compared their efficiency. Afterwards, we trained the best model with different batch sizes.

The batch size is a key concept in machine learning. It determines the number of training examples used in one iteration of model training. It influences the efficiency, speed of training, and model performance. Smaller batch sizes can speed up the learning process and lower the chance of overfitting, however, they may result in more unstable convergence. Larger batches offer smoother convergence and leverage parallel computation power, however, they require more memory.

3.6 *Metrics*

To evaluate the quality of our model’s outputs, it's essential to use the appropriate performance metrics. One key metric is precision, which assesses the model's ability to avoid false positives. Precision can be calculated as:

$$precision = \frac{TP}{TP + FP} \quad (1)$$

where TP is the number of true positive predictions and FP is the number of false positive predictions.

Another important metric is recall, which measures the model's ability to correctly identify all instances of a given class.

$$recall = \frac{TP}{TP + FN} \quad (2)$$

where FN is the number of false negatives (missed objects).

The mAP50-95 metric is commonly used in object detection tasks and provides a more comprehensive evaluation of a model’s performance. It extends the standard mAP by assessing the model at multiple levels of IoU (Intersection over Union), ranging from 0.5 to 0.95.

4 EXPERIMENTAL RESULTS

The results of the output models based on different variants of the YOLO11 model are shown in Tab. 1. All models were trained with a batch size of 16 for 100 epochs, which is the default setup for the training process in the YOLO framework. The models were tested on the test set from the dataset. Based on the defined metrics, the n variant proved to be the best-performing model on this small dataset.

Variant	Precision	Recall	mAP50-95
n	**0.997**	**1**	**0.842**
s	0.997	1	0.833
m	0.986	1	0.834
l	0.996	1	0.816
x	0.978	1	0.817

Table 1. Performance comparison of models using different YOLO11 variants.

The YOLO framework provides useful charts from the training process that can be used to analyze the training process and its effectiveness. In this paper, we focused on three metrics: precision, recall, and mAP50-95, which are commonly used to assess the performance of object detection models.

The changes in the model’s precision during training are shown in Fig. 7, which shows significant precision increase during the early training epochs. The precision stabilizes around 0.9 with a few fluctuations. The best epoch reached a precision of 0.997, which suggests that the YOLO model effectively minimized false positives.

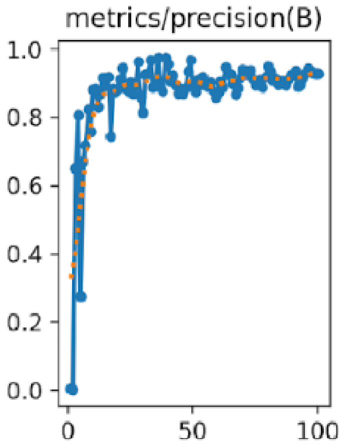

Figure 7. Change in precision metric of the YOLO11 n variant model during training across epochs.

Fig. 8 shows a rapid increase in the ability to correctly identify all instances of a given class according to the recall metric. After a few epochs, the recall stabilized near 1.0 without significant changes.

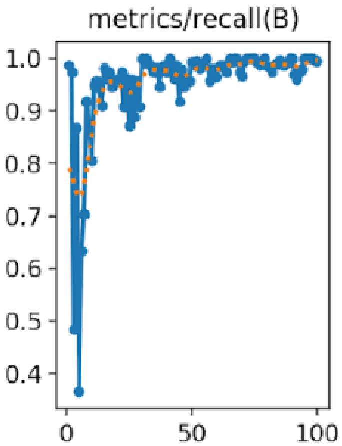

Figure 8. Change in recall metric of the YOLO11 n variant model during training across epochs.

The last evaluated mAP50-95 metric steadily increased throughout the training as shown in Fig. 9. The best results reached values around 0.8, highlighting the model's ability to provide accurate detections across varying levels of Intersection over Union (IoU) thresholds.

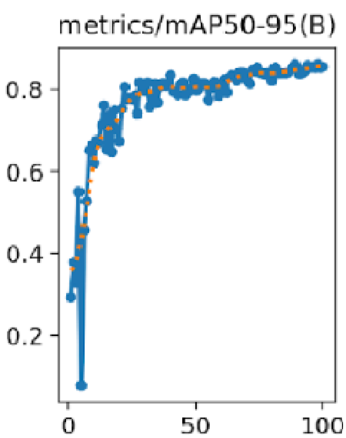

Figure 9. Change in mAP50-95 metric of the YOLO11 n variant model during training across epochs.

Based on the defined metrics, the n variant proved to be the best-performing model on this small dataset. Fig. 7 shows the precision, recall, and mAP50-95 values after every training epoch.

Tab. 2 further displays models based on the n variant of YOLO11 but with modified training setups. Here, we adjusted the batch size parameter, starting with a pre-trained model using a batch size of 16. We then increased and decreased the batch size to attempt to improve model quality. As seen in Tab. 2, the best results were achieved with a model trained with a batch size of 16. This model achieved a precision of 0.986, a recall of 0.989, and 0.836 in the mAP50-95 matric on the test dataset.

Batch size	Precision	Recall	mAP50-95
14	0.997	1	0.829
15	0.994	1	0.816
16	**0.997**	**1**	**0.842**
17	0.997	0.99	0.835
18	0.997	1	0.836
19	0.997	1	0.825
20	0.997	1	0.825

Table 2. Performance comparison of YOLO11 n variant models with varying batch sizes.

5 CONCLUSIONS

Models based on the state-of-the-art YOLO11 architectures using convolutional neural networks demonstrate their effectiveness in real-time detection tasks, such as detecting simple drawings with high precision. These models can be applied in various fields, including autonomous vehicles for detecting road signs, augmented reality applications for recognizing sketches in interactive experiences, interactive whiteboards for education, and digital art platforms. These types of models can be effectively trained on restricted datasets using appropriate preprocessing and augmentation techniques. This work proposed various model variants trained on our custom dataset, which was collected with the intention of emulating challenging situations where models are needed for the detection of rare or limited sources of drawings. The first set of models differ in architecture, and we choose the best one as a candidate for further improvement. This model was trained on different setups with varying batch sizes. The best model reached a precision of 0.986, a recall of 0.989, and 0.836 in the mAP50-95. These results gave us confidence that models based on YOLO11 can work effectively in these types of applications. Future improvements should focus on enhancing the dataset, exploring different pre-processing techniques or augmentation types, and improving the architecture of the neural network for this type of application, which would yield interesting results.

ACKNOWLEDGMENTS

This work was supported by the Slovak Research and Development Agency under the Contract no. APVV-19-0210 and project VEGA 1/0363/23.

REFERENCES

[Bacik 2015] Bacik, J., Perdukova, D., Fedor, P. Design of Fuzzy Controller for Hexacopter Position Control. In: 4th Computer Science On-line Conference (CSOC): Artificial Intelligence Perspectives and Applications, Vol. 347, p. 193-202. DOI: 10.1007/978-3-319-18476-0_20.

[Kuric 2021] Kuric, I., Klarak, J., Saga, M., Cisar, M., Hajducik, A., Wiecek, D. Analysis of the Possibilities of Tire-Defect Inspection Based on Unsupervised Learning and Deep Learning. In: Sensors, 2021, Vol. 21, No. 21, p. 7073. DOI: 10.3390/s21217073.

[Kuric 2022] Kuric, I., Klackova, I., Domnina, K., Stenchlak, V., Saga, M., Jr. Implementation of Predictive Models in Industrial Machines with Proposed Automatic Adaptation Algorithm. In: Applied Sciences, 2022, Vol. 12, No. 4, p. 1853. DOI: 10.3390/app12041853.

[Hortobagyi 2021] Hortobagyi, Á., Pivarciova, E., Koleda, P. Holographic Interferometry for Measuring the Effect of Thermal Modification on Wood Thermal Properties. In: Applied Sciences, 2021, Vol. 11, No. 6, p. 2516. DOI: 10.3390/app11062516.

[Hu 2022] Hu, J., Zhang, H., Sfarra, S., Pivarciova, E., Yao, Y., Duan, Y., Ibarra-Castanedo, C., Tian, G., Maldague, X. Autonomous dynamic line-scan continuous-wave terahertz non-destructive inspection system combined with unsupervised exposure fusion. In: NDT & E International, 2022, Vol. 132, p. 102705. DOI: 10.1016/j.ndteint.2022.102705.

[Hamzah 2024] Hamzah, R., et al. Comparing modified Yolo V5 and Faster Regional Convolutional Neural Network performance for Recycle Waste Classification. In: 2024 IEEE Int. Conf. on Automatic Control and Intelligent Systems (I2CACIS), 2024, pp. 415-419. DOI: 10.1109/I2CACIS61270.2024.10649835.

[Ji 2024] Ji, Y., et al. Lightweight Road Damage Detection Algorithm based on the Improved YOLO Model. In: 2024 5th Int. Conf. on Artif. Intelligence and Electromech. Automation (AIEA), 2024, pp. 832-835. DOI: 10.1109/AIEA62095.2024.10692408.

[Karthikeyan 2023] Karthikeyan, B., Futina, J.J., Abinaya, P.P., Priya, R.S. Parkinson's Disease Detection and Classification of Stages from Drawing Patterns Using Deep Learning. In: 2023 Int. Conf. on Bio Signals, Images, and Instrumentation (ICBSII), 2023, pp. 1-7. DOI: 10.1109/ICBSII58188.2023.10181055.

[Khanam 2024] Khanam, R. and Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. arXiv preprint arXiv:2410.17725, 2024.

[Komarskiy 2024] Komarskiy, A.A., Ponomarev, A.V., Krinitzin, V.V. Application of YOLO Algorithm for Segmentation and Classification of Minerals in CT Slices Obtained by Dual-and Multi-Energy CT. In: 2024 Int. Conf. on Industrial Engineering, Applications and Manuf. (ICIEAM), 2024, pp. 880-885. DOI: 10.1109/ICIEAM60818.2024.10553944.

[Kumar 2024] Kumar, A. and Das, P.K. Efficient Detection of Brain Tumor Using Deep Learning with Small Dataset. In: 2024 3rd Int. Conf. on Artificial Intelligence for IoT (AIIoT), 2024, pp. 1-5. DOI: 10.1109/AIIoT58432.2024.10574723.

[Liu 2024] Liu, A., Liu, Y., Kifah, S. Deep convolutional neural network for enhancing traffic sign recognition developed on Yolo V5. In: 2024 Int. Conf. on Advancements in Smart, Secure and Intelligent Computing (ASSIC), 2024, pp. 1-6. DOI: 10.1109/ASSIC60049.2024.10508025.

[Luo 2022] Luo, Y., Yu, T., Zheng, J., Ding, Y. Design of engineering drawing recognition system based on Yolo V4. In: 2022 IEEE 6th Information Technology and Mechatronics Engineering Conf. (ITOEC), 2022, Vol. 6, pp. 1221-1225. DOI: 10.1109/ITOEC53115.2022.9734453.

[Narwaria 2024] Narwaria, R.P., et al. Smart Object Detection Using ESP32-CAM Based on YOLO Algorithm. In: 2024 2nd Int. Conf. on Intelligent Cyber Physical Systems and IoT (ICoICI), 2024, pp. 817-820. DOI: 10.1109/ICoICI62503.2024.10696374.

[Nawae 2023] Nawae, M., et al. A Comparative Study of YOLO Models for Sperm and Impurity Detection Based on Proposed Augmentation in Small Dataset. In: 2023 15th Int. Conf. on Information Technology and Electrical Engineering (ICITEE), 2023, pp. 305-310. DOI: 10.1109/ICITEE59582.2023.10317746.

[Redmon 2016] Redmon, J. You only look once: Unified, real-time object detection. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 2016. DOI: 10.1109/CVPR.2016.91.

[Salman 2023] Salman, N.A. and Hasson, S.T. A Prediction Approach for Small Healthcare Dataset. In: 2023 8th Int. Conf. on Smart and Sustainable Technologies (SpliTech), 2023, pp. 1-5. DOI: 10.23919/SpliTech58164.2023.10193552.

[Sanchez 2021] Sanchez, K., et al. Subspace-based domain adaptation using similarity constraints for pneumonia diagnosis within a small chest x-ray image dataset. In: 2021 IEEE 18th Int. Symp. on Biomedical Imaging (ISBI), 2021, pp. 1232-1235. DOI: 10.1109/ISBI48211.2021.9434173.

[Wiratchawa 2022] Wiratchawa, K., et al. Training Deep CNN's to Detect Prostate Cancer Lesion with Small Training Data. In: 2022 37th Int. Technical Conf. on Circuits/Systems, Computers and Communications (ITC-CSCC), 2022, pp. 227-230. DOI: 10.1109/ITC-CSCC55581.2022.9895044.

[Xu 2023] Xu, Z., Pan, S., Ma, X. A Pedestrian Detection Method Based on Small Sample Data Set. In: 2023 IEEE Int. Conf. on Image Processing and Computer Applications (ICIPCA), 2023, pp. 669-674. DOI: 10.1109/ICIPCA59209.2023.10257732.

CONTACTS:

Ing. Ladislav Hric; Ing. Jan Bacik, PhD.
Technical university of Kosice, Faculty of Electrical Engineering and Informatics, Dept. of Electrotechnics and Mechatronics
Letna 9, Kosice, 042 00, Slovak Republic
+421 55 602 2251, ladislav.hric@tuke.sk; jan.bacik.2@tuke.sk

prof. Ing. Daniela Perdukova, PhD.
Technical university of Kosice, Faculty of Electrical Engineering and Informatics, Dept. of Electrotechnics and Mechatronics
Letna 9, Kosice, 042 00, Slovak republic
+421 55 602 2256, daniela.perdukova@tuke.sk