1. Ahmad F. Al-allaf [1], 2. Bashar H. Asker [2]

Northern Technical University, Engineering Technical College of Mosul, Department of Computer Engineering Technology, IRAQ (1)(2)

doi:10.15199/48.2022.11.11

Detecting the usage of a mobile phone during an online test using AI technology

hone. In this research, the YOLOV5 machine learning algorithm was used to detect using of the mobile phone during the electronic test. A custom dataset of phone images in different poses and orientations was created and then categorized using the Makesense webpage. The webcam on the examinee's computer captures real-time videos of the examinee and then analyzes them with the YOLOV5 algorithm. The maximum accuracy of real-time mobile phone usage detection was 92% and FAR 4%. For comparison and verification purposes, the Jetson Nano was used to detect phone usage for the same data set. The accuracy of detection with Jetson was up to 85% with 5% FAR. The two results were good and promising.

Streszczenie. *Jednym z niebezpieczeństw egzaminów elektronicznych jest oszustwo przy użyciu telefonu komórkowego. W badaniach wykorzystano algorytm uczenia maszynowego YOLOV5 do wykrywania użycia telefonu komórkowego podczas testu elektronicznego. Utworzono niestandardowy zestaw danych obrazów telefonu w różnych pozach i orientacjach, a następnie skategoryzowano go za pomocą strony internetowej Makesense. Kamera internetowa na komputerze osoby badanej przechwytuje wideo osoby badanej w czasie rzeczywistym, a następnie analizuje je za pomocą algorytmu YOLOV5. Maksymalna dokładność wykrywania użycia telefonu komórkowego w czasie rzeczywistym wyniosła 92%, a FAR 4%. Do celów porównawczych i weryfikacji wykorzystano Jetson Nano do wykrywania użycia telefonu dla tego samego zestawu danych. Dokładność wykrywania za pomocą Jetson wynosiła do 85% przy 5% FAR. Oba wyniki były dobre i obiecujące. (**Wykrywanie użycia telefonu komórkowego podczas testu online z wykorzystaniem technologii AI**)*

Keywords: YOLOv5, cell phone detection, Cheating in the online exam, Jetson nano
Słowa kluczowe: telefon kpmórkowy, egzamin online

Introduction

Object detection recognizes and locates each object in picture, then surrounded by bounding box [1]. In recent years, many unique object detection algorithms have been proposed to detect elements from photos, promoting the vast development of deep learning. In general, there are two methods of detecting elements from images: one-stage and two-stage. The two-stage detection approaches identify regions of interest (ROI) from input images before performing bounding box regression and classification in these ROIs.

In the single-stage detection methods, the object detection process is taken as a regression problem and does the localization and classification at the same stage. Single-stage detection systems, on the other hand, are known for their fast detection speed and low accuracy. Therefore, it is used in many ways and has achieved good results. While the two-stage detection methods have good detection accuracy but lower real-time performance [2].

You Only Look Once (YOLO) model is one of the most rapid and modern models for discovering embedded objects in photos and videos. YOLO stands for You Only Look Once, and the initial model was introduced in 2016, followed by YOLOv2 (2017), YOLOv3 (2018), and YOLOv4 (2020). YOLOv5 is the next member of the YOLO family released in 2020 by Ultranytics a few days after YOLOv4 [3]. The camera may be used as an image sensor to identify or recognize things, replacing the role of the human sense of sight. A camera sensor's target object can be processed or transformed into signal of information. Image sensors or cameras can be processed using a variety of techniques [4].

In the current paper, the YOLOv5 model was used to discover attempts to use cellphones while conducting electronic tests. A webcam installed on the examinee student's laptop was used. YOLOv5 will receive frames from the webcam and use them to detect mobile phones in real time. System trained on a large set of images consisting of (1169) images with (2070) labels, some taken from the Internet and others captured by camera. The images are converted to dimensions (640x640) according to YOLOV5 requirements.

Related works

In [5], Nico Hahn et al. use thermal network cameras and transfer learning to YOLOv5 to see if the front cabin and the back are appropriate features for heavy goods truck detection. The findings obtained by the researchers indicate that the trained algorithm can reliably recognize the front cabin of heavy goods trucks, whereas recognizing the rear cabin appears to be more difficult, especially when the vehicle is parked far away from the camera. When winter weather causes difficult images with a lot of overlaps and cut-offs, the system does a better job of distinguishing large freight vehicles by looking at their front and back.

Zhe Huanga [6] et al., suggested using an improved SSD algorithm method named CSP-SSD (Cross Stage Partial SSD) to detect low-resolution components in a mobile phone image. Modifications made by the researchers to the SSD algorithm include network topology optimization, use of gradient flow information, deconvolution and multi-scale transformation. The researchers claimed that the results demonstrated that utilizing SSD300-VGG16, DSSD321-ResNet, and SSD300-VGG16 for a 300*300 picture, and new model-VGG16, the mean average precision (mAP) of algorithms achieved 72.4, 66.1, and 78.9 %, respectively.

In [7], Poonam Rajput et al., to identify cell phone use, they employed deep-learning algorithms such as the single shot multiBox detector (SSD) and the faster region-based convolution neural network (Faster-RCNN). They utilized the State Farm Distracted Driver Detection dataset from Kaggle, this has a total of (10238) images at a resolution of (640x480), and then created the IITH-dataset on cellphone phone usage (IITH-DMU), which has (618) images with a resolution of (6000x4000). They claimed that the AP at 0.5IoU with SSD is 98.97% and 98.84% with Faster-RCNN. And the percentages are 92.6 for SSD and 95.92 for Faster-RCNN on the IITH-DMU dataset.

Bin Yan [8] et al., used the YOLOv5 algorithm to determine the catchable and non-catchable apples in the apple tree image. The researchers made various improvements to the YOLOv5 algorithm's structure, including optimizing the BottleneckCSP module, introducing the SE module, which belongs to the visual attention mechanism network, defining feature map correlation fusions, and optimizing the initial bounding of original

network's box size. The results according to the authors showed that the recognition retrieval, accuracy, mAP and F1 are 91.48%, 83.83%, 86.75% and 87.49%, respectively.

In [9], Abhinu C G et al., proposed a system for Multiple Object Tracking using YOLO V5 that is able to track a variety of objects that have been trained. The system was trained using (200) photos as a dataset for a class of object key, and then, after (200) epochs, the dataset created a yolo model with (95.39)% mAP, which was utilized for detecting. Object detecting had an accuracy prediction is (20-90)% when custom model was employed, depending on the quality of pictures and object appearance in the image.

YOLO-V5

Due to its high performance and minimal time requirements, YOLOV5 deep convolution neural models are used for object detection. YOLO algorithm released in 2015, makes explode in object detection. It reformulates object detection that works in a single neural network. YOLO has been upgraded to five editions. The fifth version of YOLO is released by (Glenn Jocher), who is not the original inventor of YOLO. However, in terms of accuracy and speed, YOLOV5 exceeds YOLOV4 [10].

YOLOv5 can detect and classify 80 classes like a person, bicycle, car, motorcycle, airplane … etc. There are several model configuration files and versions of the object detector to choose from. The proposed system employs YOLOv5S (small). The others (YOLOv5M (medium), YOLOv5L (Large) and YOLOv5X, (Xlarge) grow in size in order, as shown in Figure 1. As network size grows, performance will also improve, albeit at the cost of longer processing times. Therefore, larger models are used only for complex problems with large data sets [11].

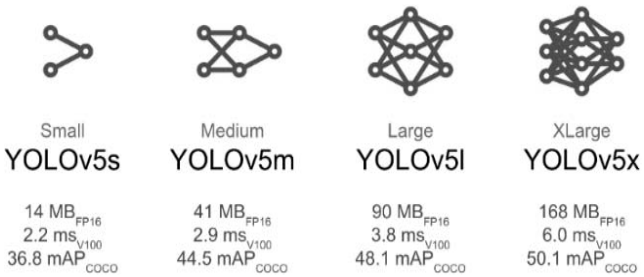

Fig. 1: YOLOV5 types

As we know object detection is challenging because it has to draw a bounding box around each detected object in the image. A cell phone is a special object that varies in shape, size, thickness, ratio (length: width), and color. We will use the accuracy, recall, and mean accuracy (mAP) object detector performance metrics to evaluate performance. These metrics are described below.

$$\boldsymbol{precision} = \frac{\boldsymbol{FT}}{\boldsymbol{FT+FF}} \quad \boldsymbol{recall} = \frac{\boldsymbol{FP}}{\boldsymbol{FP+TP}} \quad \boldsymbol{mAP} = \frac{\boldsymbol{1}}{\boldsymbol{N}} \sum_{\boldsymbol{N}}^{\boldsymbol{i=1}} \boldsymbol{APi}$$

where: FT (Frame True): is the detection true when the image contains an object.
FF (False Frame): An error is detected when YOLO-V5 marks an object that the image does not contain.

TP (True Positive): YOLOV5 doesn't detect objects but is in the image [12-14]. Cheat images will be saved in a folder and sent to the exam management center during the exam. Figure (2) shows YOLOV5 architecture, it is made up of an input module, a backbone network for feature extraction, neck network for cross-scale feature fusion, and prediction network for mobile phone detection [15]. To improve the training process, YOLOV5 employs rectified linear unit (ReLU), and dropout, a regularization approach that protects the model from errors occurring in learning [16].

Specifications and organization of proposed system:

For the system to function properly, a number of conditions must be met and are as follows:
1. Examinee did not turn off the camera during the exam.
2. Examinee should provide good lighting in the room, and distance between him and webcam should be about 50 cm.
3. A computer that has minimum specifications such as (4th gen. Intel Core-I5 processor, 12 GB RAM, SSD HDD and Graphics card (4600) internal.
4. Set the video camera to (30 FPS).

Proposed system consists of number of units. Fig.3 shows a flowchart of system units and relationship with each other.

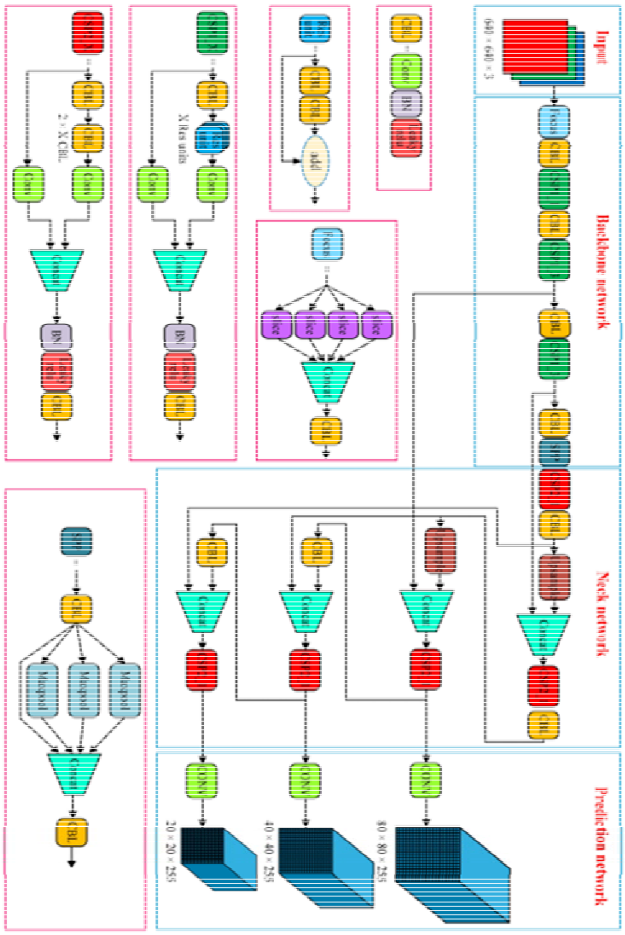

Fig. 2. Architecture of YOLOv5

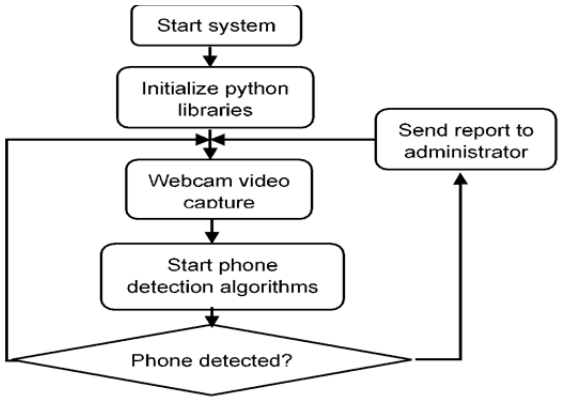

Fig. 3. Organization of system modules

Implemented system

OpenCV2 and NumPy libraries used with 1920x1080 30 frames per second (FPS) webcam used on laptops.
In computer vision and deep learning software, OpenCV is used. It contains several libraries that can be used in object

detection. NumPy contains data formats for matrices and multidimensional arrays [17].

Data set

Publicly available data sets that can be accessed via the Internet are inconsistent with the objectives of the study. Because YOLO-V5 requires annotated images in (.txt) format, the photos in the datasets are modest in size, and even huge dataset images have low resolution due to the minimal number of images in the datasets or low resolution. We were unable to obtain any data from the Internet that contained images with stickers (.txt) requested by YOLOV5. After a thorough search, no acceptable dataset for the proposed study was found. As a result, a custom dataset was constructed by picking a number of web pictures (1,169) of various phone kinds and quality levels. Additional photos taken with a personal camera from various perspectives and distances have been added to this collection.

Software used in the proposed system

A set of programs were used in this project, as follows:

1- IDLE version (3.10.0): Python text editor.

2- makesense.ai: A web website used to classify photos and is open-source. It's simple to set up, downloads data sets online, and speeds up data set preparation, making it suitable for deep learning applications [18].

3- Roboflow: Free public computer vision data sets are available in a variety of formats. Roboflow used to make discussion steps for the dataset like (Flipping, 90° rotating, Gray scaling and Brightness) [19]. Converting the data set to YOLOv5 format. We have a dataset of (2070) labels in (1169) images ready for use.

Jetson Nano

The NVIDIA development kit, it is tiny but powerful computer that allows us run numerous neural networks for image classification, object identification, segmentation, and voice processing in parallel [20] (See figure 4 [21]).

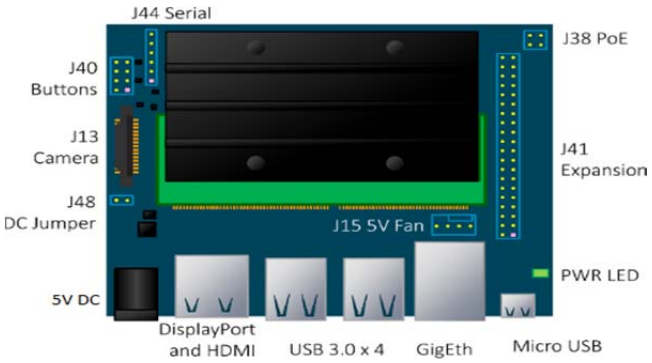

Fig. 4. Jetson Nano Kit

```
parser.add_argument('--weights', type=str, default=ROOT /'yolov5s.pt')
parser.add_argument('--data', type=str, default=ROOT /'data/c_data.yaml')
parser.add_argument('--epochs', type=int, default=350)
parser.add_argument('--batch-size', type=int, default=12')
parser.add_argument('--imgsz','--img','--img-size',type=int,default=640')
parser.add_argument('--device', default='cpu')
```

(a) training parameters

```
train: ../datasets/coco128/images/train2017
val: ../datasets/coco128/images/train2017
nc: 1
names: ['cell-phone']
```

(b) yaml file contents for one class

```
all     1169     2070     0.987     0.99     0.995  0.934
350 epochs completed in 190.143 hours.
Optimizer stripped from runs\train\exp\weights\last.pt, 14.3MB
Optimizer stripped from runs\train\exp\weights\best.pt, 14.3MB
```

(c) training accuracy and all training time

Fig. 5. Training parameters of mobile detection

In this research, Jetson Nano was used as another way to implement phone usage detection algorithm for the purpose of comparing with using laptop in terms of accuracy of the results compared to size of two systems and cost of each two systems, which is less than 100$ for Jetson Nano. One of important features of Jetson is its reliance on graphics processing unit (GPU), which is very efficient in processing.

Results and discussions:

Training parameters were: yolov5s.pt default weights, central processing unit (CPU), custom dataset, 350 epochs, batch size (12), image size (640x640) (Figure 5a) and yaml file (Figure 5b), which contain settings for mobile detection.

Training time ≅190 hours, 80% from images for training and 20% for validation with default architecture. YOLOV5 takes about 190 hours to complete 350 epochs with a dataset including 1169 photos, and the model's (mAP) is (99)% with just (150) epochs (Figure 5c). This demonstrates that, even without any optimization strategies, the YOLOv5 model is not only quick but also accurate.

Figure 6 shows the accuracy of the trained system, where X-axis represents the number of epochs (350) epochs, and Y-axis represents mAP0.5 (0.995) and mAP:0.95 (0.934).

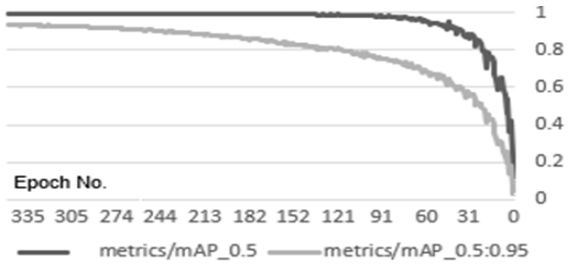

Fig. 6. Metrics mAP0.5 and mAP:0.95

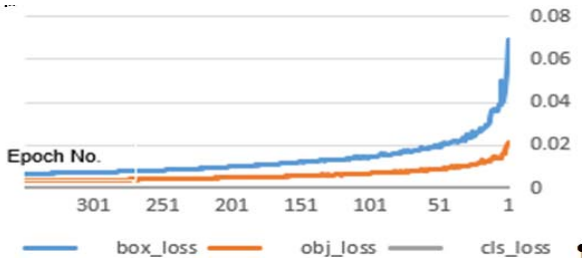

Fig. 7. Box, object and classification loss for training

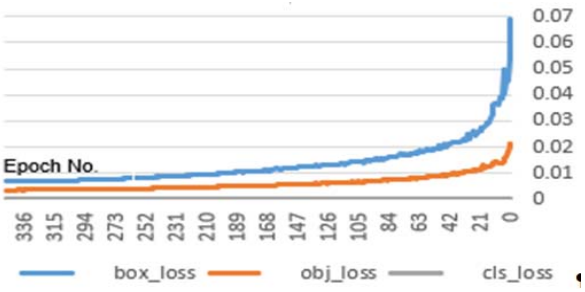

Fig. 8. Box, object and classification loss for validation

Train losses and validation losses shown in Figures 7 & 8 respectively, divided into three categories, Firstly, box loss which measures how well the algorithm detects the center of an object and how the bounding box surrounds it. The final value in epoch 350 (X-axis) was 0.0068, secondly, object loss which is a measure of the likelihood that an item exists in given (ROI). If the objectivity is high, image window is likely to have an object in it. Final value in epoch 350 was 0.0036. Finally, classification loss indicates how effectively

the algorithm predicts the proper class of an item. It is always zero because we train yolov5 on one object.

Accuracy, recall and mAP all increased quickly before plateauing after roughly 200 epochs. The validation data's box, object, and classification loss likewise exhibited a fast drop until about epoch 200 where these loss values consider good results and are close to zero.

Furthermore, YOLOv5 gives two weight results (best.pt and last.pt); Weights of the epoch with the best accuracy is (best.pt). Weights at the last epoch are (last.pt). Both files are about 14MB in size, making them extremely easy to integrate in artificial intelligence (AI) systems.

The model produces exceptionally amazing predicted results, as demonstrated in Figure 9. As previously stated, the model forecasts an item based on its likelihood; if the probability is lower than a specific threshold (our is 0.35), YOLOV5 concludes that the object is not a cellphone. Some cell phones may have probability less than our threshold. It will not expected to be a cell phone due to a low prediction probability around the threshold.

Then using file (detect.py) to detect phones with (our weights file (best.pt), CPU, default confidence threshold =0.25, default intersection over union (IOU) thresholds= 0.45 and image size= 640) on webcam; accuracy changes from (40 to 92) % in 0.328 second, as shown in Figure 10. False detection about (5%) because phones have various shapes, textures and colors. It has good results due to HD pictures and a large number of pictures. On Jetson nano, also using file (detect.py) to detect phones with same parameters above, same weights, image size= 640 but using GPU on webcam, we obtained (30-85) % accuracy in 0.422 second, which is a good result for a low-cost unit. We find that PC uses CPU and Jetson uses GPU, with specifications mentioned in this paper, almost the same results are obtained. This indicates that Jetson can be used as a good platform for AI rather than expensive computers.

Fig. 9. Training results of mobile detection

```
(weights=ROOT / 'best.pt',  # model.pt path(s)
 source=ROOT / '0',  # file/dir/URL/glob, 0 for web
 imgsz=640,  # inference size (pixels)
 conf_thres=0.25,  # confidence threshold
 iou_thres=0.45,  # NMS IOU threshold
 max_det=1000,  # maximum detections per image
 device='cpu',  # cuda device, i.e. 0 or 0,1,2,3 or
```

Fig. 10. Detection parameters for cell phone detection

Conclusion

Remote exams are currently necessary for several reasons, the most important of which is the Corona pandemic, and the most prominent challenges facing these exams are cheating via mobile phones. In this research, a system was designed to detect the use of mobile phones during the performance of electronic tests using one of the most important and modern algorithms for (AI), the YOLO5s algorithm. Good results were obtained with accuracy (40-92%) in 0.328 seconds using a laptop that uses CPU. System also implemented on Jetson Nano developer kit that uses GPU and accuracy was (30-85%) in 0.422 seconds. In both cases, processing was done in real time and it is necessary to notify the test supervisor of the presence of cheating using the phone. Because the architecture of yolov5x is significantly greater than that of yolov5s (weights in yolov5x are 168MB, whereas weights in yolov5s are 14MB), it is feasible to acquire much higher detection accuracy, but there is a delay in detecting speed.

***Authors**: Dr. Ahmad F. Al-allaf, Northern Technical University, Engineering Technical College of Mosul, Department of Computer Engineering Technology, Iraq, E-mail: ahmed.faleh@ntu.edu.iq; Bashar H. Asker, Northern Technical University, Engineering Technical College of Mosul, Department of Computer Engineering Technology, Iraq, E-mail: bashar.hasan@ntu.edu.iq.*

REFERENCES

[1] Najwan W., Nawal A. and Mohammed G., Automatic Detection of Underage Troopers from LiveVideos Based on Deep Learning, *Przegląd Elektrotechniczny*, 97(2021), No.9, 85-88.
[2] Zhu L., Geng X., Li Z. and Liu C., Improving YOLOv5 with Attention Mechanism for Detecting Boulders from Planetary Images. *Remote Sens*. 13(2021), No.3776, 1-19.
[3] Nelson J. and Solawetz J., Responding to the Controversy about YOLOv5, 12/JUN/2020, Available at: https://blog.roboflow.com/yolov4-versus-yolov5.
[4] Syahri M. and Budi P., Automatics Detect and Shooter Robot Based on Object Detection Using Camera, *Przegląd Elektrotechniczny*, 98(2022), No.1, 50-54.
[5] Kasper M., Hahn N., Berger S., Sebulonsen T., Myrland Ø., and Kummervold P., Short Communication: Detecting Heavy Goods Vehicles in Rest Areas in Winter Conditions Using YOLOv5, *Algorithms*, 14(2021), No. 114, 1-11.
[6] Zhe H., Zhenyu Y., Yue M., Chao F. and Anying C., Mobile Phone Component Object Detection Algorithm Based on Improved SSD, *10th International Conference of Information and Communication Technology (ICICT)*, 183 (2021), 107–114.
[7] Poonam R., Subhrajit N. and Sparsh M., Detecting Usage of Mobile Phones Using Deep Learning Technique, *International Conference on Smart Objects and Technologies for Social Good (GoodTechs '20)*, 2020.
[8] Yan B., Fan P., Lei X., Liu Z. and Yang F., A Real-Time Apple Targets Detection Method for Picking Robot Based on Improved YOLOv5. *Remote Sens*. ,13(2021), No.1619, 1-23.
[9] Abhinu C., Aswin P, Kiran K., Bonymol B. and Viji K., Multiple Object Tracking Using Deep Learning with YOLOV5, *International Journal of Engineering Research & Technology (IJERT)*, 9(2021), No.13, 47-51.
[10] Do Thuan, Evolution of Yolo Algorithm and Yolov5: The State-of-the-Art Object Detection Algorithm, Bachelor's Thesis, Information Technology, Oulu University of Applied Sciences, 2021, 1-61.
[11] Ildar I., YOLOv4 vs YOLOv5, 30/JUN/2020, available at: https://medium.com/deelvin-machine-learning/yolov4-vs-yolov5-db1e0ac7962b.
[12] Xu R., Lin H., Lu K., Cao L. and Liu Y., A Forest Fire Detection System Based on Ensemble Learning. *Forests,* 12(2021), No.217, 1-17.
[13] Song Q., Li S., Bai Q., Yang J., Zhang X., Li Z. and Duan Z., Object Detection Method for Grasping Robot Based on Improved YOLOv5, *Micromachines*, 12(2021), No.1273., 1-18.
[14] Maciej G. and Stanisław O., Classical Versus Deep Learning Methods for Anomaly Detection in ECG Using Wavelet Transformation, *Przegląd Elektrotechniczny*, 97(2021), No.6, 72-76.
[15] Lei F., Tang F. and Li S., Underwater Target Detection Algorithm Based on Improved YOLOv5, *Journal of Marine Science and Engineering*, 10(2022), No.310, 1-19.
[16] Souha A. and Zied L., Visual Emotion Sensing Using Convolutional Neural Network, *Przegląd Elektrotechniczny*, 98(2022), No.3, 89-92.
[17] Vinay S., Face Mask Detection Using YOLOv5 for COVID19, M.Sc. thesis, San Marcos, California state university, 2020,1-24.
[18] Makesense, https://www.makesense.ai/.
[19] Roboflow company, https://roboflow.com/.
[20] NVIDIA company, developer.nvidia.com/embedded/jetson-nano.
[21] NVIDIA company, tx-team.de/images/JetsonNano-overview.