Real-Time YOLO-family Comparison For Blast Quality Estimation in the Open Pit Conditions

Mikhail Ronkin
IRIT-RTF
Ural Federal University
Yekaterinburg, Russia
m.v.ronkin@urfu.ru

Kirill Reshetnikov
IRIT-RTF
Ural Federal University
Yekaterinburg, Russia,
ki.reshetnikov@urfu.ru

2023 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT) | 979-8-3503-3605-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/USBEREIT58508.2023.10158813

***Abstract*—Conducting blasting in an open pit environment is an essential step in the technology of rock processing. The subsequent stages of transporting and processing the resulting rock fragments critically depend on the quality of the blast. As a result, the control of blast quality in real time is necessary during blasting in the open pit. This paper compares the quality of detection and segmentation of rocks obtained after blasting using the YOLO model family. A series of neural network training experiments were conducted and compared using the metrics precision, recall, average precision, and inference time per image. Based on experimental results, it is shown that the updated YOLOv8X model performs better for both detection and segmentation tasks. For the object detection task, YOLOv8X shows an average precision of 76%. This outperforms the YOLOv7X model by 1% on Average precision and the YOLOv5X model by 5%. In the segmentation task, the YOLOv8X model outperforms the YOLOv7X model by 5% on Average precision, and the YOLOv5X model by 7%.**

Keywords—object detection, instance segmentation, computer vision, deep learning, blast quality estimation, fragmentation, deep learning, yolo, rock fragmentation estimation, open pit

I. Introduction

The mining industry faces a number of significant challenges, including analyzing and estimating mining performance under open-pit conditions [1, 2, 3, 4]. Generally, open-pit mining performance (productivity) is determined by blasting quality, which is carried out at each stage. Under high-quality blasts, we assume blasts in which most of the obtained rock fragments are found in the lower part of their size distribution [1, 2]. At the processing factory, such fragments of rock can be easily transported and treated. Furthermore, rock fragments that have a relatively large surface area cannot be handled by standard manufacturing equipment or transported by standard vehicles. As a result, such rock chunks require additional treatment with special equipment. This procedure has a high cost and time consumption [1, 2]. Therefore, blast quality needs to be controlled in real time during blasting in the open pit (i.e., control on a real-time scale).

Based on the literature review, it can be shown that computer vision systems can be utilized to estimate blast quality [1, 5]. Some papers suggest the use of classical methods of instance segmentation, such as watershed-based algorithms [5]. While blast quality can be assessed by a value proportional to the square root of the rock fragment area distribution.

However, it is noted that such approaches provide high accuracy only for small-scale open pits and in a limited range of environmental conditions,i.e., sufficient light, daytime, dry weather and so on [1]. In paper [1] it was proposed to use deep learning-based neural networks to solve the problem of instance segmentation in the framework of the problem under study. The authors showed that this approach overcomes many of the previously mentioned drawbacks. Authors of [1] suggested using the Mask-RCNN architecture [6] for instance segmentation, a well-known architecture in the literature that has been successfully applied to similar investigations and other industrial tasks. Nonetheless, this architecture demands significant resources and does not offer an ideal trade-off between time consumption and accuracy, as reported in [7]. Obtaining Mask-RCNN-based results becomes less practically significant due to this circumstance. Maintaining a balance between low computational complexity and relatively high accuracy is a critical challenge for low-performing (low-cost) devices, such as industrial tablets or mobile phones used as measurement devices. This is particularly crucial in cases where it's necessary to adjust blasting in open pits in real-time [7].

It is noteworthy that the studies discussed in [1, 2] often used the square root of the rock fragment area as the base or equivalent size. This choice can be attributed to the comparison of their results with conventional software such as GoldSize [8], which is oriented toward estimating partition distributions by using absolute scales of sizes. However, this equivalent size may not be an accurate measure when dealing with oblong or round-shaped fragments, as it fails to represent the maximum dimension size of such fragments.

In our previous paper, we proposed considering the blast quality estimation task as an object detection problem. The benefits of this approach were shown in the trade-off between high accuracy and time-consumption of model work [9]. The Yolov5X model was proposed as the most efficient both for accuracy and inference time in 2021. YOLO architectures recently released a built-in implementation of an instance segmentation solution, as well as several updated architecture versions [10, 11, 12].

The goal of this work is to evaluate the YOLO family models by analyzing the balance between their accuracy and inference time in the modes of object detection and instance segmentation on images of the open-pit work places, obtained after blasting.

II. YOLO family review

The Mask R-CNN network architecture with ResNet-101-FPN as the backbone is the usual solution to segment instances in the research area (see [1]). Previous results show the benefits of using YOLOv5x architectures for object detection [9]. The utilization of the YOLO family of models can be justified based on the fact that the YOLO architecture was created with the aim of achieving an optimal balance between accuracy and resource consumption.

This research was supported by the Russian Science Foundation and Government of Sverdlovsk region, Joint Grant No 22-21-20051, https://rscf.ru/en/project/22-21-20051/.

Authorized licensed use limited to: b-on: UNIVERSIDADE DO PORTO. Downloaded on July 15,2024 at 12:17:05 UTC from IEEE Xplore. Restrictions apply.

Furthermore, these architectures have gained widespread application in computer vision tasks within the industry, even on low-performance (inexpensive) devices [13, 14]. Additionally, the YOLO family exhibits top-tier performance on standard benchmarks such as MS COCO [15].

The You Look Only Ones (YOLO) family was firstly designed as successful attempt to provide real-time one-stage object detection task solution. It is based on the idea that the input image is divided into cells of equal size, that correspond with one output vector each. Such a vector can include one or more anchors with bounding box size predictions, class and object determination scores. The output cell with object can be determined using non-maximum suppression algorithm [16]. The next progress in this field was made through experiments with architecture and training features both for backbone and head (with neck) parts. The YOLOv2 uses only convolutional layers except for the last one and a region proposal network as anchor generator [17]. It was proposed that YOLOv3 would use a feature-pyramidal based neck part and several modifications [18]. The YOLO v4 includes a lot of specified training tricks («Bag of Fribies») and architecture improvements («Bag of Specials»). The YOLOv5 was created in PyTorch [10] using some small improvements from V4 and redesigned. The 4,5 version became an industry standard in the object detection field. In designing version 3-5, many other works in the field of real-time object detection were considered. Among other YOLO models, YOLOv7 is proposed as a modification of YOLOv4 [11]. In this architecture, the authors propose the new block E-ELAN (Extended efficient layer aggregation networks) as a modification of the CSP block. This block was chosen as a trade-off between fast computation and easy scaling. Block based on the idea of group shuffle-net convolution with different ways of processing each group. Also the original trick of self-label assignment made in the head part. The current brand-new YOLOv8 architecture [12] includes new convolution block based on the CSP idea but with flexible scaling factor. Also, the anchor-free head part is applied. This idea was proven in the YOLO v6 version [19]. The anchor-free approach allowing to significantly decrease computational complexity of the models without sufficient loss of accuracy. The YOLO v5,7,8 architectures includes also built-in implementation of an optional instance segmentation head part [10, 11, 12].

III. Experiments with models

A. Datbase description

The dataset includes 220 images of the open-pits after blasting. All data are collected in the Bazhenovskoye field (Russia, Sverdlovsk obl., Asbeset). The applied camera system descriptions were given in the papers [3]. The majority of the dataset was captured during five experiments held in autumn and winter under varying weather conditions (sunny, cloudy, snow), with each experiment including 3-5 open-pit processing areas selected by the geological service of the field processing company. Multiple images were taken of different parts of each open-pit area, with each image containing roughly 20-80 fragments of varying sizes. The collected data was labeled for instance segmentation (and object detection) using the CVAT software [20], with typical labeled images shown in Figure 1. The database is available by the link https://data.mendeley.com/v1/datasets/pfdbfpfygh/draft?preview=1. The data was divided into 174 train images, 14 validation images and 72 images in the test part. Also the 134 images of rock chunks on the conveyer belt was added to the train part. The data description can be found in the paper [9].

a) b)

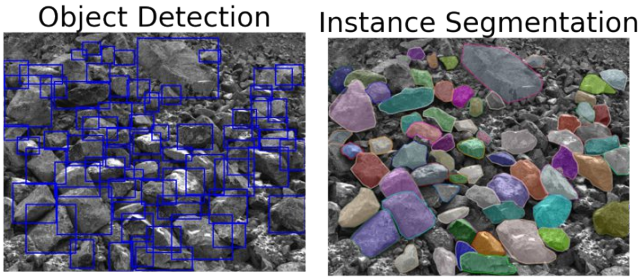


Fig. 1. Typical examples of labeled images for object detection on the left and instance segmentation on the right.

B. Experiments results

Experiments with neural network models for object detection and instance segmentation were carried out. To train and test the models, we utilized a server machine with an 18.04.1 Ubuntu system, and a single Tesla V100-SXM2 32GB graphics card running CUDA 11.8. Training for all models was performed for 100 epochs by SGD method, with initial LR 0.01 and batch value equal to 4. Images in all experiments were scaled to 512×512 pixels. During training, dynamic augmentations with default parameters provided in the respective packages for YOLOv5, YOLOv7, and YOLOv8 were used for all models. Model weights were pre-trained on the COCO dataset [21]. Images of 512×512 pixels without any processing were used during testing.

The Fast R-CNN model was applied as a baseline solution to the detection problem. Detectron2 was used to train the Fast R-CNN model with the backbone rcnn_R_101_FPN_3x [22]. Additionally, Mask R-CNN model with backbone rcnn_R_101_FPN_3x is employed for segmentation. The parameters used for training were learning rate $5 \cdot 10^{-3}$, number of epochs 3000, batch size 4, image size was changed to 800×800. Augmentation parameters used for training: random extent with scale in range [0.7,1.3] and shift 0.1, random brightness with intensity in range [0.8, 1.8], random contrast in range [0.6, 1.3], random saturation in range [0.8, 1.4], random rotation by an angle in the range [-90,90], random lighting with scale 0.7.

The models were evaluated using metrics such as Precision, Recall, Average Precision (AP), as well as inference time per image.

- Precision is the ratio of the area obtained as a result of crossing the bounding boxes (bboxes) of the real object location and the predicted one in the area of the predicted bounding boxes. The metric takes into account false positives, in our case incorrect detection of rock chunks.
- Recall is the ratio of the area obtained by crossing the bounding boxes of the real and predicted object location in the area of the bounding boxes of the real object location. In other words the recall score measures the fraction of positive results that were predicted correctly.
- AP (average precision) is the popular metric in the object detection task. To calculate the metric, the Precision-Recall relation must be given. Then AP is caclulated as the value of the area under the obtained curve. In other words the average precision generalizes the PR curve to a single scalar value. Average precision is high when both Precision and Recall are high, and low when either is low.

Table I contains measures values on the test data for models under investigation in the object detection problem solution. The table I results show that the highest accuracy is provided by YOLOv8 models. The YOLOv7x is comparable with YOLOv8l in detection accuracy, while being the fastest in image processing speed. Both models outperform their

Authorized licensed use limited to: b-on: UNIVERSIDADE DO PORTO. Downloaded on July 15,2024 at 12:17:05 UTC from IEEE Xplore. Restrictions apply.

competitors by 2-3 ms. The YOLOv5 models show the lowest accuracy, losing 5% in average. Fast R-CNN show results for test data AP@.5 62.67%, AP@0.95 41%, inference time for one image 102.1 ms.

TABLE I. RESULTS FOR OBJECT DETECTION TASK

Model	Precision, %	Recall,%	AP@.5, %	AP@0.95, %	Time, ms
YOLOv8x	71.3	71.2	76.0	54.4	18.3
YOLOv8l	72.1	69.5	76.8	54.9	17.9
YOLOv5x	69.1	71.6	72.2	49.9	20.8
YOLOv5l	70.2	68.7	72.3	49.1	17.6
YOLOv7x	68.7	73.8	76.3	53.7	15.9
Fast R-CNN			62.7	41.0	102.1

The result of the predicted bounding boxes and ground truth labels are presented in Fig. 2 a-f for different models. From the visual analysis, we can note that the YOLOv7x model detects small objects better compared to YOLOv5x. Fast R-CNN model detects small stone fragments poorly.

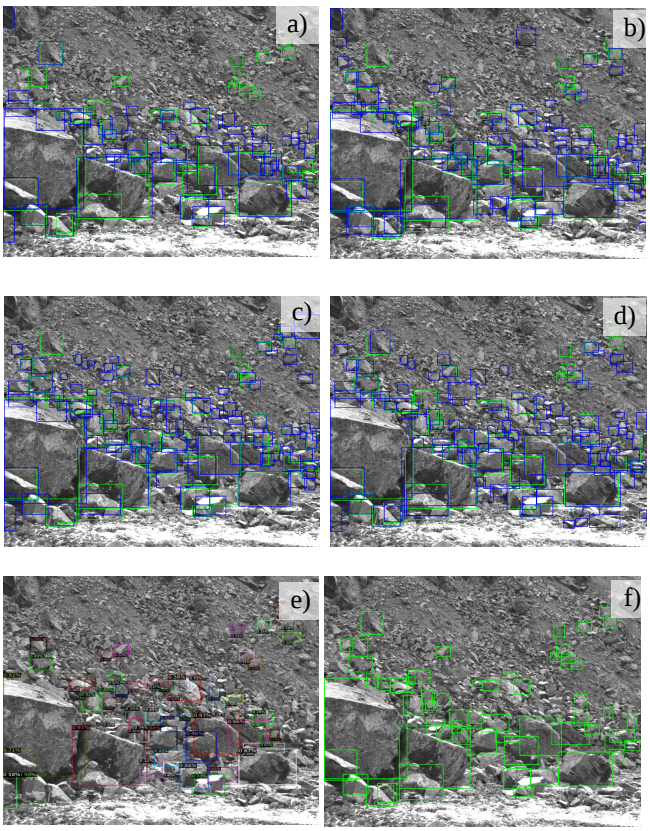


Fig 2. Typical examples of bunging boxes prediction for different models: a)YOLOv5x, b)YOLOv7x, c)YOLOv8l, d)YOLOv8x, e) Fast R-CNN, f) Ground truth labels. Green colored boxes are ground truth and blue boxes are obtained one.

Fig. 3 shows example of image with the minimal AP value for the case of YOLOv8x model (AP is 25%). The obtained boxes behavior can be explained as follows. There is a part of the stone out of focus on the left side of the image. However, the part in the upper right corner was not detected. The model detect three small unmarked stones in the upper left. There is no network response to the marked stone at the bottom of the image. This stone is not the central object in the image and part of this stone is covered by another stone at the bottom right.

Table II contains measures values on the test data for models under investigation in the instance segmentation problem. The table II includes precision, recall and average precision metrics all for IOU 0.5; average precision for IOU 0.5, 0.55...0.95 and also the inference time. The table II shows that the highest accuracy is demonstrated by the model YOLOv8x. The YOLOv5x model loses by 8% in comparison. The YOLOv7x has the highest speed with an average image processing time of 16.5 ms. Mask R-CNN show results for test data AP@0.5 52.3%, AP@0.95 32.1%, inference time for one image 87.65 ms. Mask R-CNN score is lower than YOLO. This may be due to the fact that YOLO uses modern technique like mosaic augmentation, exponential moving average teacher, etc.

TABLE II. RESULTS FOR INSTANCE SEGMENTATION TASK

Model	Precision, %	Recall,%	AP@.5, %	AP@0.95, %	Time, ms
YOLOv8x	67.7	70.4	73.3	46.4	20
YOLOv8l	69.6	65.8	71.6	44.6	17.8
YOLOv5x	68.1	59.8	63.9	38.8	19.1
YOLOv5l	67.5	58.4	61.0	36.2	20.7
YOLOv7x	69.1	65.3	71.2	41.5	16.5
Mask R-CNN			52.3	32.1	87.65

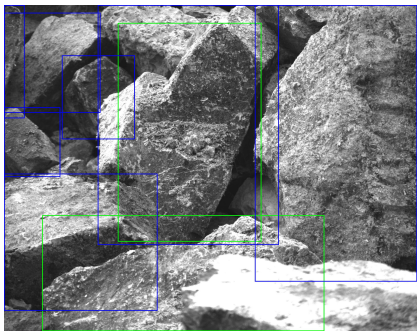
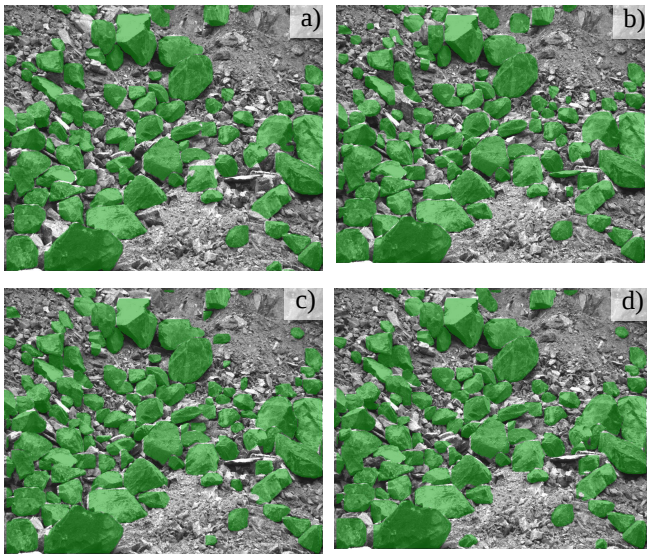


Fig. 3. Example of an image with a accuracy of 25% for model YOLOv8x. Green colored boxes are ground truth and blue boxes are obtained one.

Fig. 4 shows typical examples of the obtained segmentation for four investigated models. From the visual analysis of the Fig. 4, it can be seen that the YOLOv8x model detects more stones and segments large stones better.



Authorized licensed use limited to: b-on: UNIVERSIDADE DO PORTO. Downloaded on July 15,2024 at 12:17:05 UTC from IEEE Xplore. Restrictions apply.

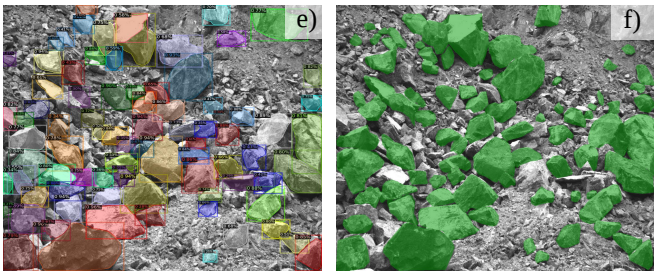


Fig 4. Typical examples of instance segmentation masks are predicted for various types of models: a)YOLOv5x, b)YOLOv7x, c)YOLOv8l, d)YOLOv8x, e) Mask R-CNN, f) Ground truth.

As a result, instance segmentation slightly reduces the performance of the Yolo models in both terms of accuracy and inference time. In consequence, if the problem can be solved as an object detection problem, then this mode of YOLO is preferable. One could think of it as the Yolo model designed specifically for object detection. In particular, as it was shown earlier in [9] the problem of blasting quality estimation can be reduced to the problem of object detection (estimation of the distribution of rock chunks is given according to the predicted bounding boxes). For high accuracy among the YOLO family models, the YOLOv8 might be recommended. Models YOLOv7 may be more suitable for low-performance devices due to their low inference time.

A contribution of this work lies in investigating and recommending real-time object detection and instance segmentation models of the YOLO family for fragmentation problem. The brand new models YOLO were compared and their performance was determined in the in the open-pit conditions.

Conclusion

This paper presents the results of detection and segmentation of rocks chunks in an open pit using convolutional YOLO neural networks. The study compares the accuracy and inference time of the YOLOv5, YOLOv7 and YOLOv8 models in object detection and instance segmentation for the task of estimating blasting quality using computer vision systems. In the segmentation task, the YOLOv8x model performs better than the YOLOv7x model by 5%, and 7% better than the YOLOv5x model in AP. In the object detection task, the YOLOv8x model performs better than the YOLOv7x model by 1% in AP and by 5% over the YOLOv5x model. In terms of output time we can distinguish YOLOv7x, which shows the faster speed, while at the same time showing average precision comparable with YOLOv8.

Therefore, object detection mode may be recommended to solve the problem of estimation of blasting quality in an open pit condition as a fragmentation task. The YOLOv8 (in particular Yolov8x) shows the best performance while slightly increasing the inference time in comparison with other models. If the lowest inference time is required, then the use of models of the YOLOv7 series is preferable.

The novelty of the presented work is the analysis of state-of-the-art real-time object detection and instance segmentation models of the YOLO family in the task of rock chunks fragments in open pits.

References

[1] T. Vu, T. Bao, Q.V. Hoang, C. Drebenstetd, P.V. Hoa, and H.H. Thang, “Measuring blast fragmentation at Nui Phao open-pit mine, Vietnam using the Mask R-CNN deep learning model”, Mining Technology, vol. 130 no 4, 2021. pp.232-243.

[2] T. Bamford, K. Esmaeili and AP. Schoellig, “A deep learning approach for rock fragmentation analysis”, International Journal of Rock Mechanics and Mining Sciences, vol 145, 2021, p.104839.

[3] V. Zyuzin, M. Ronkin, S. Porshnev and A. Kalmykov, “Automatic Asbestos Control Using Deep Learning Based Computer Vision System” Applied Sciences, vol. 11 no. 22, 2021. p.10532.

[4] V. Zyuzin, M. Ronkin, S. Porshnev and A. Kalmykov, “Computer vision system for the automatic asbestos content control in stones” J. Phys.: Conf. Ser., vol. 1727, 2021. p. 012014.

[5] S.G. Mkwelo, V. Nicolls, and G. de Jager, “Watershed-based segmentation of rock scenes and proximity-based classification of watershed regions under uncontrolled lighting”, SAIEE Africa Research Journal, vol. 96 no. 1, 2005. pp.28-34.

[6] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask r-cnn”, In Proceedings of the IEEE international conference on computer vision, 2017. pp. 2961-2969.

[7] S.S. Sumit, J. Watada, A. Roy, and D.R.A. Rambli, “In object detection deep learning methods, YOLO shows supremum to Mask R-CNN. In Journal of Physics: Conference Series”, vol. 1529, no. 4, 2020. p. 042086.

[8] *Gold Size 2.0*. [Online] Available: https://civilnode.com/download-software/10159053855788/gold-size-20. [Accessed: 20.01.2023].

[9] M. Ronkin, A. Kalmykov, K. Reshetnikov, V. Zyuzin. “Investigation of Object Detection Based Method for Open-Pit Blast Quality Estimation”, In 2022 Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT) 2022, pp. 248-251.

[10] Ultralytics. *Yolov5:v6.1* [Online]. Available: https://github.com/ultralytics/yolov5. [Accessed: 20.01.2023].

[11] C.Y. Wang, A. Bochkovskiy, H.Y.M. Liao. “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”. arXiv preprint arXiv:2207.02696, 2022.

[12] Ultralytics. *Ultralytics YOLOv8: The State-of-the-Art YOLO Model* [Online]. Available: https://ultralytics.com/yolov8. [Accessed: 20.01.2023].

[13] P. Jiang, D. Ergu, F. Liu, Y. Cai, B. Ma. “A Review of Yolo algorithm developments”, Procedia Computer Science, 199, pp.1066-1073.

[14] T. Diwan, G. Anirudh, J.V. Tembhurne, “Object detection using YOLO: Challenges, architectural successors, datasets and applications”. Multimedia Tools and Applications, 2022, pp.1-33

[15] Real-Time Object Detection on COCO. *Object Detection on COCO test-dev* [Online]. Available: https://paperswithcode.com/sota/object-detection-on-coco. [Accessed: 20.01.2023].

[16] J. Redmon, S. Divvala, R. Girshick, A. Farhadi “You only look once: Unified, real-time object detection”. InProceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779-788.

[17] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. pp. 7263-7271.

[18] J. Redmon, A. Farhadi, . Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[19] Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W. and Li, Y., 2022. YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976.

[20] CVAT, *Open Data Annotation Platform* [Online]. Available: https://www.cvat.ai/. [Accessed: 20.01.2023].

[21] T. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. Lawrence Zitnick, P. Dollar, “Microsoft COCO: Common Objects in Context”, arXiv:1405.0312, 2014.

[22] Y. Wu, A. Kirillov, F. Massa, W.Y.Lo, and Ross Girshick. Detectron2 [Online]. Available: https://github.com/facebookresearch/detectron2 [Accessed: 07.04.2023].

Authorized licensed use limited to: b-on: UNIVERSIDADE DO PORTO. Downloaded on July 15,2024 at 12:17:05 UTC from IEEE Xplore. Restrictions apply.