CrowdSim2: An Open Synthetic Benchmark for Object Detectors

Paweł Foszner[1,*][a], Agnieszka Szczęsna[1,*][b], Luca Ciampi[3,†][c], Nicola Messina[3,†][d], Adam Cygan[5,§], Bartosz Bizoń[5,§], Michał Cogiel[4,‡][e], Dominik Golba[4,‡][f], Elżbieta Macioszek[2,*][g] and Michał Staniszewski[1,*,**][h]

[1]*Department of Computer Graphics, Vision and Digital Systems, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Akademicka 2A, 44-100 Gliwice, Poland*
[2]*Department of Transport Systems, Traffic Engineering and Logistics, Faculty of Transport and Aviation Engineering, Silesian University of Technology, Krasińskiego 8, 40-019 Katowice, Poland*
[3]*Institute of Information Science and Technologies, National Research Council, Via G. Moruzzi 1, 56124 Pisa, Italy*
[4]*Blees sp. z o.o. Zygmunta Starego 24a/10, 44-100 Gliwice, Poland*
[5]*QSystems.pro sp. z o.o. Mochnackiego 34, 41-907 Bytom, Poland*
*

Keywords: Object Detection, Vehicle Detection, Pedestrian Detection, Synthetic Data, Deep Learning, Crowd Simulation.

Abstract: Data scarcity has become one of the main obstacles to developing supervised models based on Artificial Intelligence in Computer Vision. Indeed, Deep Learning-based models systematically struggle when applied in new scenarios never seen during training and may not be adequately tested in non-ordinary yet crucial real-world situations. This paper presents and publicly releases *CrowdSim2*, a new synthetic collection of images suitable for people and vehicle detection gathered from a simulator based on the *Unity* graphical engine. It consists of thousands of images gathered from various synthetic scenarios resembling the real world, where we varied some factors of interest, such as the weather conditions and the number of objects in the scenes. The labels are automatically collected and consist of bounding boxes that precisely localize objects belonging to the two object classes, leaving out humans from the annotation pipeline. We exploited this new benchmark as a testing ground for some state-of-the-art detectors, showing that our simulated scenarios can be a valuable tool for measuring their performances in a controlled environment.

1 INTRODUCTION

In recent years, Computer Vision swerved toward Deep Learning (DL)-based models that learn from vast amounts of annotated data during the supervised learning phase. These models achieved astonishing results in several tasks that nowadays are considered basic, such as image classification, causing interest in addressing more complex domains such as object detection (Cafarelli et al., 2022), image segmentation (Bolya et al., 2019), visual object counting (Ciampi et al., 2022c) (Avvenuti et al., 2022) (Ciampi et al., 2022a), people tracking (Staniszewski et al., 2016), or even facial reconstruction (Pęszor et al., 2016) and video violence detection (Ciampi et al., 2022b). However, these more cumbersome tasks often also require more structured datasets that come with challenges concerning bias, privacy, and cost in terms of human effort for the annotation procedure.

Indeed, more complex tasks correspond to more elaborated labels, and for each data sample, the effort shifts from annotating an image to annotating the objects present in it, even at the pixel level. Furthermore, more challenging tasks often go hand in hand with more complex scenarios that may rarely occur in the real world, yet correctly handling them can be crucial. Finally, privacy concerns surrounding Artificial Intelligence-based models have become increasingly important, further complicating data collection. Consequently, labeled datasets are often limited, and data scarcity has become the main stumbling block for the development and the in-the-wild application of Computer Vision algorithms. Deep Learning-based algo-

[a] https://orcid.org/0000-0001-5491-9096
[b] https://orcid.org/0000-0002-4354-8258
[c] https://orcid.org/0000-0002-6985-0439
[d] https://orcid.org/0000-0003-3011-2487
[e] https://orcid.org/0000-0002-9776-9654
[f] https://orcid.org/0000-0002-4542-3547
[g] https://orcid.org/0000-0002-1345-0022
[h] https://orcid.org/0000-0001-9659-7451
**Corresponding author

Foszner, P., Szczęsna, A., Ciampi, L., Messina, N., Cygan, A., Bizoń, B., Cogiel, M., Golba, D., Macioszek, E. and Staniszewski, M.
CrowdSim2: An Open Synthetic Benchmark for Object Detectors.
DOI: 10.5220/0011692500003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 676-683
ISBN: 978-989-758-634-7; ISSN: 2184-4321

Figure 1: Some samples of our synthetic dataset we rendered with our simulator, together with the bounding boxes localizing the objects of interest.

rithms systematically struggle in new scenarios never seen during the training phase and may not be adequately tested in non-ordinary yet crucial real-world situations.

One appealing solution that is recently arising relies on collecting *synthetic data* gathered from *virtual environments* resembling the real world. Here, by interacting with the graphical engine, it is possible to *automatically* collect the labels associated with the objects of interest, cutting off the human effort from the annotation procedure, thus reducing the costs. Furthermore, these reality simulators provide frameworks where it is possible to create specific scenarios by controlling and explicitly varying the factors that characterize them. Hence, they represent the perfect environments where automatically acquiring labeled data for the training phase but also be used as controlled testing grounds for evaluating the performance capabilities of the employed models.

In this paper, we consider the object detection task, focusing our attention on *people* and *vehicle* detection. We deem that people localization is crucial for security as well as for crowd analysis; on the other hand, vehicle detection constitutes the building block for urban and road planning, traffic light modeling, and traffic management, to name a few. In particular, we introduce and make publicly available *CrowdSim2*, a new vast collection of synthetic images suitable for object detection and counting, collected by exploiting a simulator based on the *Unity* graphical engine. Specifically, it consists of thousands of small video clips gathered from various synthetic scenarios where we varied some factors of interest, such as the weather conditions and the number of objects in the scenes. The labels are automatically collected and consist of bounding boxes that precisely localize objects belonging to two different classes — *person* and *vehicle*. We report in Figure 1 some samples of images together with the bounding boxes localizing the objects of interest in different scenarios we rendered with our simulator. Then, we present a detailed experimental analysis of the performance of several state-of-the-art DL-based object detectors pre-trained over general object detection databases present in the literature by exploiting our *CrowdSim2* dataset as a testing ground. More in-depth, we extracted, from the collected videos, batches of frames belonging to specific and controlled scenarios, and we measured the obtained performances by varying the factors that characterized them.

Summarizing, the contributions of this paper are listed below:

- we propose *CrowdSim2*, a new synthetic dataset suitable for *people* and *vehicle* detection, collected by exploiting a simulator based on the *Unity* graphical engine and made freely available in the Zenodo Repository at (Szczęsna et al., 2023);
- we test some state-of-the-art object detectors over this new benchmark, exploiting it as a testing ground where we varied some factors of interest such as the weather conditions and the object density;
- we show that our simulated scenarios can be a valuable tool for measuring detectors' performances in a controlled environment.

2 RELATED WORKS

2.1 Synthetic Datasets

Synthetically-generated datasets have recently gained considerable interest due to the need for huge amounts of annotated data. Some notable examples are *GTA5* (Richter et al., 2016) and *SYNTHIA* (Ros et al., 2016) for semantic segmentation, *Joint Track Auto (JTA)* (Fabbri et al., 2018) for pedestrian pose estimation and tracking, *Virtual Pedestrian Dataset (ViPeD)* (Ciampi et al., 2020) (Amato et al., 2019) for pedestrian detection, *Grand Traffic Auto (GTA)* (Ciampi et al., 2021) for vehicle segmentation and

counting, *CrowdVisorPPE* (Benedetto et al., 2022) for Personal Protective Equipment detection and *Virtual World Fallen People (VWFP)* (Carrara et al., 2022) for fallen people detection. These datasets are mainly exploited for training deep learning models, which benefit from the fact that these collections of images are vast since the labels are automatically collected. On the other hand, using synthetic data as ground test collections is a relatively unexplored field. Furthermore, the datasets mentioned above are collected from the GTA V (Grand Theft Auto V) video game by Rockstar North. Although it is a very realistic generator of annotated images, some limitations arise when new scenarios or behaviors are needed. By contrast, using a simulator based on an open-source graphical engine allows one to create more customized environments and easily modify some factors of interest — density of the objects, weather conditions, and object interactions.

2.2 Object Detectors

In the last decade, object detection has become one of the most critical and challenging branches of Computer Vision. It deals with detecting instances of semantic objects of a specific class (such as humans, buildings, or cars) in digital images and videos (Dasiopoulou et al., 2005). This task has attracted increasing attention due to its wide range of applications and recent technological breakthroughs. Currently, most state-of-the-art object detectors employ Deep Learning models as their backbones and detection networks to extract features from images, classification, and localization, respectively. Existing object detectors can be divided into two categories: *anchor-based* detectors and *anchor-less* detectors. The models in the first category compute bounding box locations and class labels of object instances exploiting Deep Learning-based architectures that rely on anchors, i.e., prior bounding boxes with various scales and aspect ratios. They can be further divided into two groups: i) the two-stage paradigm, where a first module is responsible for generating a sparse set of object proposals and a second module is in charge of refining these predictions and classifying the objects; and ii) the one-stage approach that directly regresses to bounding boxes by sampling over regular and dense locations, skipping the region proposal stage. Some notable examples belonging to the first group are *Faster R-CNN* (Ren et al., 2017) and *Mask R-CNN* (He et al., 2017). At the same time, popular networks of the latter set are the *YOLO* family and *RetinaNet* (Lin et al., 2020) algorithm. On the other hand, anchor-free methods rely on predicting key points, such as corner or center points, instead of using anchor boxes and their inherent limitations. Some popular works existing in the literature are *CenterNet* (Zhou et al., 2019), and *YOLOX* (Ge et al., 2021). Very recently, another object detector category is emerging, relying on the newly introduced Transformer attention modules in processing image feature maps, removing the need for hand-designed components like a non-maximum suppression procedure or anchor generation. Some examples are *DEtection TRansformer (DETR)* (Carion et al., 2020) and one of its evolution, *Deformable DETR* (Zhu et al., 2021).

In this paper, we consider some networks belonging to the *"You Only Look Once" (YOLO)* family detectors, which turned out to be one of the most promising detector architectures in terms of efficiency and accuracy. The algorithm was introduced by (Redmon et al., 2016) as a part of a custom framework called *Darknet* (Redmon, 2013). Acronym *YOLO (You Only Look Once)* derived from the idea of single shot regression approach. The author introduced the single-stage paradigm that made the model very fast and small, even possible to implement on edge devices. The next version was *YOLOv2* (Redmon and Farhadi, 2017), which introduced some iterative improvements (higher resolution, BatchNorm, and anchor boxes). *YOLOv3* (Redmon and Farhadi, 2018) added backbone network layers to the model and some other minor improvements. *YOLOv4* (Bochkovskiy et al., 2020) introduced improved feature aggregation and mish activation. *YOLOv5* (Qu et al., 2022) proposed some improvements in feature detection, split into two stages - shallow feature detection and deep feature detection. The latest ones *YOLOv6* (Li et al., 2022) and *YOLOv7* (Wang et al., 2022) added some new modules like the re-parameterized module and a dynamic label assignment strategy, further increasing the accuracy.

3 THE Crowdsim2 DATASET

In this section, we introduce our *CrowdSim2* dataset, a novel synthetic collection of images for *people* and *vehicle* detection [1]. First, we describe the Unity-based simulator we exploited for gathering the data, and then we depict the salient characteristics of this new database.

[1]The dataset is freely available in the Zenodo Repository at https://doi.org/10.5281/zenodo.7262220

Figure 2: Samples of our synthetic data where we show the four different weather conditions we varied with our simulator.

3.1 The Simulator

In this work, we exploited an extended version of the *CrowdSim* simulator, introduced in (Staniszewski et al., 2020), that was designed and developed by using the *Unity* graphical engine. The main goal of this simulator is to produce *annotated* data to be used for training and testing Deep Learning-based models suitable for object and action detection. For this purpose, it allows users to generate realistic image sequences depicting scenes of urban life, where objects of interest are localized with precise bounding boxes. More in-depth, the simulator is designed using the *agent-based* paradigm. In this approach, an agent – in our work either a human or a vehicle – is controlled individually, and decisions are made in the context of the environment in which the agent was placed. For instance, people can perform different types of movement thanks to the skeletal animation (Wereszczyński et al., 2021) and actions depending on the situation in which they find themselves, including running, walking, jumping, waving or shaking hands, etc. The related animations vary depending on the age, height, and posture of the agent. Also, interactions between agents are possible in the so-called *interaction zones*. Within this zone, the simulator continuously checks several conditions, such as the number of agents in the zone or random variables. If the conditions are met, the agents interact (fight, dance, etc.).

The environment in which agents are placed is important as the movement and behavior of the agents themselves. The considered simulator allows the user to generate a situation in four locations. They are:

- traffic with intersections, pedestrian crossings, sidewalks, etc., in a typical urban environment, captured from three different cameras;
- a green park for pedestrians without traffic, filmed from three cameras;
- the main square of an old town, captured with two cameras;
- a tunnel for cars captured at both the endpoints, perfect for issues related to re-identification.

General rules of road traffic were applied to car movements. The starting positions of the cars are randomized among pre-defined starting points, and then the vehicles move to the point where they need to change direction. In such a place, cars make random decisions regarding further movements. Cars can only move in designated zones (streets and parking bays).

3.2 Simulated Data

Using the simulator described in the previous section, we gathered a synthetic dataset suitable for *people* and *vehicle* detection. Specifically, for people detection, we used three different scenes, while for car detection, two different scenarios. We recorded thousands of small video clips of 30 seconds at a resolution of 800×600 pixels and a frame rate of 25 Frames Per Second (FPS), from which we extracted hundreds of thousands of still images. We varied several factors of interest, such as people's clothes, vehicle models, weather conditions (sun, fog, rain, and snow), and

Table 1: Summary of our generated synthetic data. Each row corresponds to different weather conditions we set using our simulator. We report the total number of the collected video clips and the number of frames we extracted from them.

	# video-clips	# frames
Sun	2,899	2,174,250
Rain	1,633	1,224,750
Fog	1,653	1,239,750
Snow	1,646	1,234,500

the objects' density in the scene. The ground truth is generated following the golden standard of the *MOT-Det Challenge* [2], consisting of the coordinates of the bounding boxes localizing the objects of interest — *people* and *vehicles* in our case. The summary of the generated data is presented in Table 1. We report in Figure 2 the four different weather conditions we considered as one of the factors we varied during the data recording.

4 RESULTS AND DISCUSSION

In this section, we evaluate several deep learning-based object detectors belonging to the *YOLO* family, described in Section 2, on our *CrowdSim2* dataset. Following the primary use case for this dataset explained in Section 1, we employed it as a test benchmark to measure the performance of the considered methods in a simulated scenario where some factors of interest are controlled and changed. Specifically, we compared the obtained results considering four different weather conditions – *sun*, *rain*, *fog*, *snow* – and different densities of objects present in the scene – from 1 object to hundreds of objects.

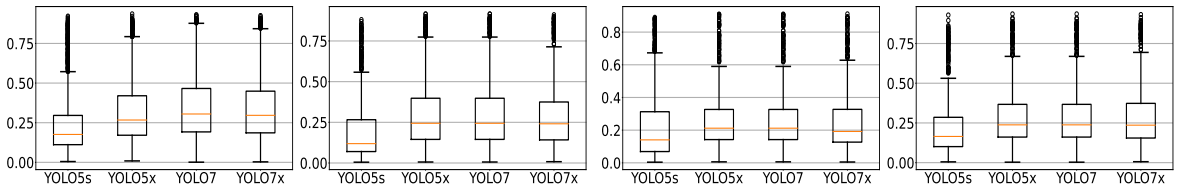
We considered two different *YOLO*-based models: *YOLOv5* and *YOLOv7*. Concerning *YOLOv5*, we selected two different architectures having a different number of trainable parameters – a light version we called *YOLO5s* and a more deep architecture we referred to as *YOLO5x*. Concerning *YOLOv7*, we exploited the standard architecture (we referred to as *YOLO7*) and a deeper version which we called *YOLO7x*. Our decision to consider models having different architectures has been dictated by the fact that we wanted to prove that their behavior in the simulated data reflects the one observable over the real-world datasets – shallow models are expected to exhibit moderate performances compared to deeper architectures. We refer the reader to Section 2 and the related papers for further details about the architectures of the employed detectors. All the models were fed with images of 640×640 pixels, and the models were pre-trained using the *COCO* dataset (Lin et al., 2014), a popular collection of images for general object detection.

We performed two different sets of experiments — the first related to people detection and the second to vehicle detection. We evaluated and compared the above-described detectors following the golden standard Average Precision (AP), i.e., the average precision value for recall values over 0 to 1. Specifically, we considered the MS COCO AP@[0.50], i.e., the AP computed at the single IoU threshold value of 0.50 (Lin et al., 2014). We report the results concerning people detection varying the weather conditions and the people density in Figure 3 and Figure 4, respectively. On the other hand, results regarding vehicle detection varying the same two factors are depicted in Figure 5 and Figure 6, respectively.

Concerning people detection, the considered models perform slightly better when the *sun* weather condition is set. On the other hand, concerning the *rain*, *snow*, and *fog* weather conditions, the detectors obtain lower APs. This is an expected outcome since, also in the real world, the detectors have to face more challenges when they are required to work in that specific conditions since the objects are more difficult to find. This trend is even more pronounced considering the car detection experiments, where some detectors particularly struggle in the *rain* and *fog* settings. On the other hand, the trend of both people detection and car detection exhibits performance degradation with the increasing of the objects present on the scene. Again, in this case, this behavior is expected and reflects that detecting instances is way more challenging in overcrowded scenarios.

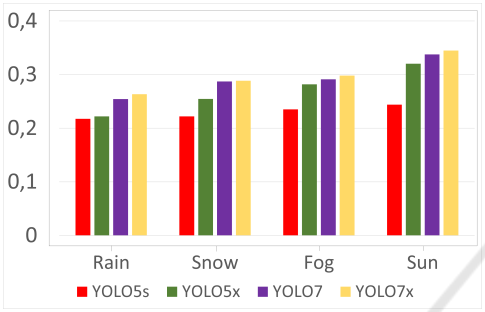
Looking at Figure 3, note how in the people detection scenarios, the performance difference among the different detectors is negligible, although the *YOLO7x* seems to achieve the best mean AP and the *YOLO5s* exhibits the worse results. Also, considering Figure 4a, we can observe how *YOLO7*, *YOLO7x* and *YOLO5m* maintain certain robustness even in the most challenging conditions, while *YOLO5s* – besides starting with a worse detection performance even in the *sun* setting – has a decreasing trend for the other weather conditions, reaching the worst AP of around 0.19 in the *fog* setting. Contrarily, the performance of the different models shows steeper differences in the car detection scenarios. In that case, the *YOLO5s* completely struggles in the *fog*, *snow* and *rain* scenarios, as shown in Figure 5 and in Figure 6a. On the other hand, *YOLO7x* seems more robust to all weather conditions, except in the *fog* setting, for which it exhibits moderate performances. This higher sensitivity

[2] https://motchallenge.net/

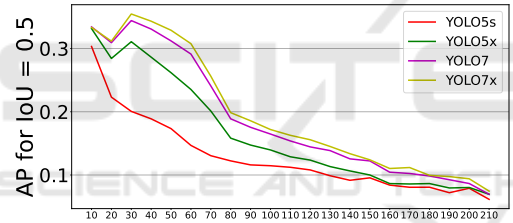


(a) *sun* weather condition. (b) *fog* weather condition. (c) *rain* weather condition. (d) *snow* weather condition.

Figure 3: Average Precision with IOU = 0.5 calculated for different weather conditions (*sun, fog, rain and snow*), obtained for the *people* detection task by exploiting the four considered *YOLO* methods.



(a) Results varying weather conditions.



(b) Results varying object densities.

Figure 4: Summary of Average Precision with IOU = 0.5 obtained with the four *YOLO*-based considered methods by varying the two main simulated factors of interest: *weather condition* and *density* of the objects.

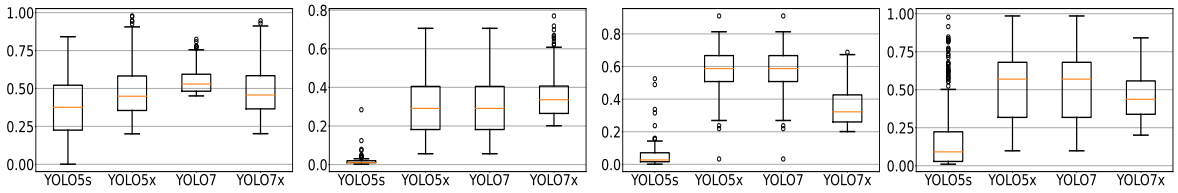
of the detectors in the vehicle detection compared to the people scenario may be due to how the different *YOLO* versions have been trained, demonstrating their major robustness to people detection – even in very challenging weather scenarios – than cars. This result contributes to validating our main claim that synthetic scenarios are crucial during the testing phase for finding biases or robustness breaches of largely-used detector models. Finally, by analyzing the results depicted in Figure 4b and in Figure 6b, we can again confirm that the performances of the considered detectors are more similar in the people detection task, while they show significant differences in detecting vehicles, especially in crowded scenarios.

5 CONCLUSION

In this work, we introduced a new synthetic dataset for *people* and *vehicle* detection. This collection of images is automatically annotated by interacting with a realistic simulator based on the *Unity* graphical engine. This allowed us to create a vast number of different simulated scenarios leaving out humans from the annotation pipeline, in turn reducing costs and tackling the data scarcity problem affecting supervised Deep Learning models. At the same time, we kept control over some factors of interest, such as weather conditions and object densities, and we measured the performances of some state-of-the-art object detectors by varying that factors. Results showed that our simulated scenarios can be a valuable tool for measuring their performances in a controlled environment. The presented idea has an extensive number of possible applications. People and car detection can lead to different usages, such as object counting and traffic analysis or object tracking. Furthermore, crowd simulation development is also desirable in the direction of action recognition. We also plan to enrich our simulator by introducing the possibility of viewing from multiple cameras in urban environments to create a new benchmark for multi-object tracking.

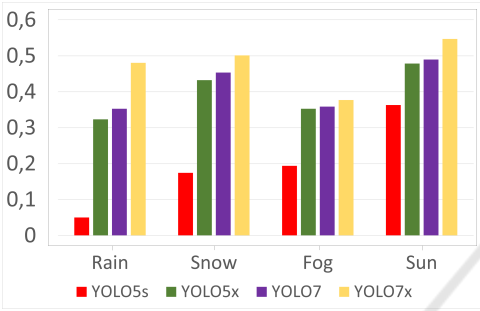
ACKNOWLEDGMENTS

This work was supported by: European Union funds awarded to Blees Sp. z o.o. under grant POIR.01.01.01-00-0952/20-00 “Development of a system for analysing vision data captured by public transport vehicles interior monitoring, aimed at detecting undesirable situations/behaviours and passenger counting (including their classification by age group) and the objects they carry”); EC H2020 project “AI4media: a Centre of Excellence delivering next generation AI Research and Training at the service of Media, Society and Democracy” under GA 951911; research project (RAU-6, 2020) and

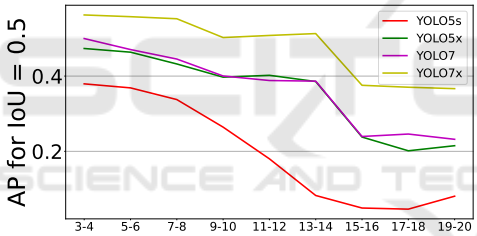


(a) *sun* weather condition. (b) *fog* weather condition. (c) *rain* weather condition. (d) *snow* weather condition.

Figure 5: Average Precision with IOU = 0.5 calculated for different weather conditions (*sun, fog, rain and snow*), obtained for the *vehicle* detection task by exploiting the four considered *YOLO* methods.



(a) Results varying weather conditions.



(b) Results varying object densities

Figure 6: Summary of Average Precision with IOU = 0.5 obtained with the four *YOLO*-based considered methods by varying the two main simulated factors of interest: *weather condition* and *density* of the objects.

projects for young scientists of the Silesian University of Technology (Gliwice, Poland); research project INAROS (INtelligenza ARtificiale per il mOnitoraggio e Supporto agli anziani), Tuscany POR FSE CUP B53D21008060008. Publication supported under the Excellence Initiative - Research University program implemented at the Silesian University of Technology, year 2022. This research was supported by the European Union from the European Social Fund in the framework of the project "Silesian University of Technology as a Center of Modern Education based on research and innovation" POWR.03.05.00- 00-Z098/17. We are thankful for students participating in design of Crowd Simulator: P. Bartosz, S. Wróbel, M. Wola, A. Gluch and M. Matuszczyk.

REFERENCES

Amato, G., Ciampi, L., Falchi, F., Gennaro, C., and Messina, N. (2019). Learning pedestrian detection from virtual worlds. In *Lecture Notes in Computer Science*, pages 302–312. Springer International Publishing.

Avvenuti, M., Bongiovanni, M., Ciampi, L., Falchi, F., Gennaro, C., and Messina, N. (2022). A spatio- temporal attentive network for video-based crowd counting. In *IEEE Symposium on Computers and Communications, ISCC 2022, Rhodes, Greece, June 30 - July 3, 2022*, pages 1–6. IEEE.

Benedetto, M. D., Carrara, F., Ciampi, L., Falchi, F., Gennaro, C., and Amato, G. (2022). An embedded toolset for human activity monitoring in critical environments. *Expert Systems with Applications*, 199:117125.

Bochkovskiy, A., Wang, C., and Liao, H. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934.

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. (2019). YOLACT: Real-time instance segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE.

Cafarelli, D., Ciampi, L., Vadicamo, L., Gennaro, C., Berton, A., Paterni, M., Benvenuti, C., Passera, M., and Falchi, F. (2022). MOBDrone: A drone video dataset for man OverBoard rescue. In *Image Analysis and Processing – ICIAP 2022*, pages 633–644. Springer International Publishing.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *Computer Vision – ECCV 2020*, pages 213–229. Springer International Publishing.

Carrara, F., Pasco, L., Gennaro, C., and Falchi, F. (2022). Learning to detect fallen people in virtual worlds. In *International Conference on Content-based Multimedia Indexing*. ACM.

Ciampi, L., Carrara, F., Totaro, V., Mazziotti, R., Lupori, L., Santiago, C., Amato, G., Pizzorusso, T., and Gennaro, C. (2022a). Learning to count biological structures with raters' uncertainty. *Medical Image Analysis*, 80:102500.

Ciampi, L., Foszner, P., Messina, N., Staniszewski, M., Gennaro, C., Falchi, F., Serao, G., Cogiel, M., Golba,

D., Szczęsna, A., and Amato, G. (2022b). Bus violence: An open benchmark for video violence detection on public transport. *Sensors*, 22(21).

Ciampi, L., Gennaro, C., Carrara, F., Falchi, F., Vairo, C., and Amato, G. (2022c). Multi-camera vehicle counting using edge-AI. *Expert Systems with Applications*, 207:117929.

Ciampi, L., Messina, N., Falchi, F., Gennaro, C., and Amato, G. (2020). Virtual to real adaptation of pedestrian detectors. *Sensors*, 20(18):5250.

Ciampi, L., Santiago, C., Costeira, J., Gennaro, C., and Amato, G. (2021). Domain adaptation for traffic density estimation. In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications.

Dasiopoulou, S., Mezaris, V., Kompatsiaris, I., Papastathis, V., and Strintzis, M. G. (2005). Knowledge-assisted semantic video object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(10):1210–1224.

Fabbri, M., Lanzi, F., Calderara, S., Palazzi, A., Vezzani, R., and Cucchiara, R. (2018). Learning to detect and track visible and occluded body joints in a virtual world. In *Computer Vision – ECCV 2018*, pages 450–466. Springer International Publishing.

Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021). YOLOX: exceeding YOLO series in 2021. *arXiv preprint arXiv:2107.08430*.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. (2017). Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017*, pages 2980–2988. IEEE Computer Society.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., and Wei, X. (2022). Yolov6: A single-stage object detection framework for industrial applications. *CoRR*, abs/2209.02976.

Lin, T., Goyal, P., Girshick, R. B., He, K., and Dollár, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755. Springer.

Pęszor, D., Staniszewski, M., and Wojciechowska, M. (2016). Facial reconstruction on the basis of video surveillance system for the purpose of suspect identification. In Nguyen, N. T., Trawiński, B., Fujita, H., and Hong, T.-P., editors, *Intelligent Information and Database Systems*, pages 467–476, Berlin, Heidelberg. Springer Berlin Heidelberg.

Qu, Z., yuan Gao, L., ye Wang, S., nan Yin, H., and ming Yi, T. (2022). An improved yolov5 method for large objects detection with multi-scale feature cross-layer fusion network.

Redmon, J. (2013). Darknet: Open source neural networks in c.

Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788. IEEE Computer Society.

Redmon, J. and Farhadi, A. (2017). Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster r-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149.

Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016). Playing for data: Ground truth from computer games. In *Computer Vision – ECCV 2016*, pages 102–118. Springer International Publishing.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M. (2016). The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Staniszewski, M., Foszner, P., Kostorz, K., Michalczuk, A., Wereszczyński, K., Cogiel, M., Golba, D., Wojciechowski, K., and Polański, A. (2020). Application of crowd simulations in the evaluation of tracking algorithms. *Sensors*, 20(17):4960.

Staniszewski, M., Kloszczyk, M., Segen, J., Wereszczyński, K., Drabik, A., and Kulbacki, M. (2016). Recent developments in tracking objects in a video sequence. In *Intelligent Information and Database Systems*, pages 427–436. Springer Berlin Heidelberg.

Szczęsna, A., Foszner, P., Cygan, A., Bizoń, B., Cogiel, M., Golba, D., Ciampi, L., Messina, N., Macioszek, E., and Staniszewski, M. (2023). Crowd simulation (CrowdSim2) for tracking and object detection.

Wang, C., Bochkovskiy, A., and Liao, H. M. (2022). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *CoRR*, abs/2207.02696.

Wereszczyński, K., Michalczuk, A., Foszner, P., Golba, D., Cogiel, M., and Staniszewski, M. (2021). ELSA: Euler-lagrange skeletal animations - novel and fast motion model applicable to VR/AR devices. In *Computational Science – ICCS 2021*, pages 120–133. Springer International Publishing.

Zhou, X., Wang, D., and Krähenbühl, P. (2019). Objects as points. *arXiv preprint arXiv:1904.07850*.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2021). Deformable DETR: deformable transformers for end-to-end object detection. In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net.