





# Put Your PPE on: A Tool for Synthetic Data Generation and Related Benchmark in Construction Site Scenarios

Camillo Quattrocchi<sup>1</sup> <sup>a</sup>, Daniele Di Mauro<sup>1,2</sup> <sup>b</sup>, Antonino Furnari<sup>1,2</sup> <sup>c</sup>, Antonino Lopes<sup>3</sup>,  
Marco Moltisanti<sup>3</sup> <sup>d</sup> and Giovanni Maria Farinella<sup>1,2,4</sup> <sup>e</sup>

<sup>2</sup>*Next Vision s.r.l. - Spinoff of the University of Catania, Italy*

<sup>3</sup>*Xenia Network Solutions s.r.l, Italy*

<sup>4</sup>*ICAR-CNR, Palermo, Italy*

**Keywords:** Synthetic Data, Safety, Pose Estimation, Object Detection.

**Abstract:** Using Machine Learning algorithms to enforce safety in construction sites has attracted a lot of interest in recent years. Being able to understand if a worker is wearing personal protective equipment, if he has fallen in the ground, or if he is too close to a moving vehicles or a dangerous tool, could be useful to prevent accidents and to take immediate rescue actions. While these problems can be tackled with machine learning algorithms, a large amount of labeled data, difficult and expensive to obtain are required. Motivated by these observations, we propose a pipeline to produce synthetic data in a construction site to mitigate real data scarcity. We present a benchmark to test the usefulness of the generated data, focusing on three different tasks: safety compliance through object detection, fall detection through pose estimation and distance regression from monocular view. Experiments show that the use of synthetic data helps to reduce the amount of needed real data and allow to achieve good performances.


## 1 INTRODUCTION


Construction sites are one of the most dangerous place where to work<sup>1</sup> and the reduction of fatal accidents is crucial in this context. In recent years, due to the availability of low cost cameras, high bandwidth wireless connections, as well as hardware and software platforms to exploit computer vision and machine learning, methods to accomplish this goal have gained attention. Monitoring the compliance to safety measures and automatically triggering alarms are two of the main areas where computer vision algorithms can help reduce fatal accidents.


The main downside of approaches based on machine learning is “data hungeriness”: to solve complex problems, algorithms need a large amount of labeled


data from which to learn. More importantly, the required data tends to be domain-specific and hence a new collection and labeling effort may be required whenever a new task is considered or a new system is installed. Acquiring and labeling a dataset is a costly and time-consuming process and in some environments, such as construction sites, it faces problems which are not always surmountable, such as privacy concerns and the inability to capture a good amount of rare events such as accidents.


A consolidated way to get around the lack of data is to exploit realistic but synthetic data. Such data can be generated using a 3D simulator which can automatically label different properties of the data, such as the presence of objects and people in the scene, thus leading to consistent savings in terms of time. In this paper, we investigate a method for generating synthetic data automatically labeled to address several safety monitoring tasks in a construction site. The proposed approach aims to generate synthetic data using the Grand Theft Auto V video game rendering engine. We build on the work of (Di Benedetto et al., 2019) who proposed to generate synthetic data to de-

<sup>a</sup>  <https://orcid.org/0000-0002-4999-8698>

<sup>b</sup>  <https://orcid.org/0000-0002-4286-2050>

<sup>c</sup>  <https://orcid.org/0000-0001-6911-0302>

<sup>d</sup>  <https://orcid.org/0000-0003-3984-9979>

<sup>e</sup>  <https://orcid.org/0000-0002-6034-0432>

<sup>1</sup> [https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Fatal\\_and\\_non-fatal\\_accidents\\_5.png](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Fatal_and_non-fatal_accidents_5.png)

tect PPEs (Personal Protective Equipment) used by workers in a construction site using the same video game rendering engine. More specifically we extend this approach from different viewpoints: we generate data both from *first-person* perspective, which corresponds to cameras placed on the workers' helmets, and from two *third-person* views, corresponding to cameras mounted on the vehicles working on the construction site and cameras placed at the top of the four ends of the construction site. We also provide a mechanism to randomize the generation of the scenarios subjected to some constraints: the construction sites are built at different positions on the game map and they vary both in the arrangement of the property (i.e. quantity and type) and of the workers (i.e. position, clothing, and physical attributes). Our tool allows to generate automatic annotations and can be used to train algorithms to tackle different tasks. The tool was developed as a plugin for the Grand Theft Auto V<sup>2</sup> game engine. This tool is able to generate the synthetic dataset automatically.

To compare the performance of the models trained with the synthetic images with respect to the ones trained on real data, a set of real data has also been acquired and manually labeled. We studied the robustness of synthetic data for construction site domain in a range of applications related to safety monitoring. In particular we focused on the following tasks: safety compliance through object detection, fall detection through pose estimation and distance regression from monocular view. The experiments show that the proposed paradigm is effective in filling the lack of real labeled data to tackle the considered tasks. Specifically, results show a strong contribution of synthetic data to improve the performances of the algorithms. To summarize, the contributions of the paper are the following:

1. A tool capable of generating large amounts of synthetic labelled data related to construction sites in a short time through randomly generated scenarios;
2. A benchmark that describes and shows that the use of the synthetic data can be useful to improve the performance of different algorithms;
3. A tool able to detect the use of PPE by workers, to evaluate distances within a construction site (e.g., distance between a worker and a working vehicle), and to recognize a worker on the ground (e.g., due to an accident).

## 2 RELATED WORKS

Our work focuses on machine learning algorithms to monitor safety compliance in a construction site through the use of synthetic data for training purpose. Both safety monitoring and synthetic data generation have been investigated in recent years, and several works have tackled these tasks. In the following paragraphs, we present some of the works most relevant to ours.

**Safety Monitoring.** The use of machine learning algorithms for safety monitoring is becoming increasingly popular. Many existing computer vision tasks can be exploited to reduce accidents and increase safety in workplaces (Sandru et al., 2021; Wu et al., 2019). (Kim et al., 2021) uses a YOLOv4 ((Bochkovskiy et al., 2020)) object detector to recognize workers and equipment from aerial images, in order to understand dangerous situations within a work site. (Taufeeque et al., 2021; Juraev et al., 2022) use the OpenPifPaf (Kreiss et al., 2021; Kreiss et al., 2019) algorithm to capture situations of domestic falls, managing to calculate the pose of the subjects and to assign a "Fall" or "No-Fall" label from the pose. (Jayaswal and Dixit, 2022) monitor distance between people in order to maintain social distancing in real time during the period of the Covid-19 pandemic.

**Synthetic Data Generation.** Thanks to the evolution of rendering engines and the greater availability of GPUs, the use of synthetic data in computer vision is a de-facto standard to obtain data for tasks which are hard to label. The synthetic data can be generated using 3D graphics tools (i.e. Blender, Maya, etc), or can be generated through the use of customizable video game engines (i.e. GTA-V, Unreal, etc). (Quattrocchi et al., 2022) used Blender to generate synthetic data to automatically and simultaneously obtain synthetic frames paired with ground truth segmentation masks to use for the Panoptic Segmentation task in an industrial domain. (Leonardi et al., 2022) also used synthetic data in an industrial domain, but focused on the Human Object Interaction task, where the goal was to simulate hand-object interactions. (Di Benedetto et al., 2019) used the rendering engine of Grand Theft Auto V to generate data in the scenario of a construction site in order to train an object detector capable of detecting the presence or absence of PPE. (Savva et al., 2019; Szot et al., 2021) simulate agents which navigate within 3D environments and perform many different tasks. The work ofgi (Sankaranarayanan et al., 2018) tackles the prob-

<sup>2</sup><https://www.rockstargames.com/gta-v>

lem of the shift between real domain and synthetic domains, proposing an approach based on Generative Adversarial Networks (GANs). (Pasqualino et al., 2021) considers the problem of unsupervised domain adaptation for object detection in cultural sites between real images of the cultural site and synthetic images. (Dosovitskiy et al., 2017) introduced an open-source driving simulator for autonomous driving. The simulator runs on Unreal Engine 4 (UE4) and allows to have full control of different parameters, such as the positioning of vehicles and pedestrians, as well as changes in weather conditions. (Fabbri et al., 2021; Hu et al., 2019; Hu et al., 2021; Krähenbühl, 2018; Richter et al., 2017) also use a video game to generate synthetic data, but adopt a slightly different approach as compared to the proposed method and those presented previously. These works extract g-buffers from the GPU in order to extract intermediate representations from the rendering pipeline. In this way, they are able to automatically extract information such as depth maps, segmentation masks, optical flows. This approach was not used in our work due to the need to modify and generate custom entities, as well as the cameras. Also, the g-buffers extraction approach would not have allowed the extraction of worker keypoints.

### 3 DATA GENERATION

The video game Grand Theft Auto V (GTAV) is a popular video game based on Rockstar Advanced Game Engine (RAGE). It is set in a real world and thus it contains thousands of assets which are suitable in different domains. A third party developer distributed the RAGE Plugin Hook (RPH)<sup>3</sup> component that allows to hook pieces of custom source code, called plugins. Such plugins allow to manipulate the running game instance and perform actions such as the spawn of polygonal models (characters, vehicles, buildings, objects), as well as the ability to assign a behavior to each model, in the form of action sequences defined through the script. We relied on this component to create a plugin to extract and automatically annotate frames.

The plugin is composed of three main modules:

**Location Collector.** We generate data in different locations of the game map. The location collector takes care of collecting, within the game map, the positions in which the scenarios to be acquired will be generated.

<sup>3</sup><https://ragepluginhook.net/>

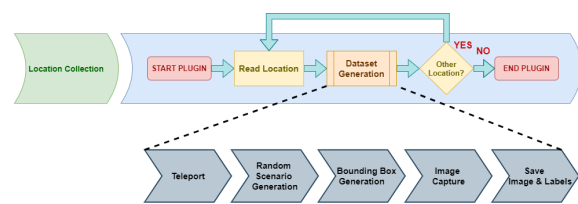


Figure 1: Plugin workflow. The plugin executes, in order, the processes of reading the locations, generating the scenario and acquiring the scenario. The scenario generation process includes several stages, such as the teleportation to the current location, the generation of the scenario perimeter, and the generation of workers and work items.



Figure 2: Synthetic construction site.

**Scenario Creator.** This module deals with the generation of random scenarios. The construction site, workers, vehicles, and objects are then generated for each location collected by the Location Collector.

**Auto Labeling.** This module takes care of the data acquisition process: for each construction site, images are collected from all points of view defined in the script, both from the first person and third person points of view. The module is also responsible for automatic annotation of the generated images.

The execution of the plugin follows the flow depicted in Figure 1. In Figure 2 is reported an example of a synthetic image generated by the plugin.

The first step is to read the first available position of the map available from the location collector module. Once the location is read, the playable character is teleported to the corresponding location. Once the playable character has been teleported, the construction site is generated randomly, positioning the cones that delimit the construction site, the vehicle, the objects (e.g., pneumatic hammer, concrete mixer, etc), the workers (with random body attributes and random clothing), and the cameras. Synthetic data are generated from different kinds of cameras: four third-person cameras, one for each corner of the construction site; four vehicle-centric cameras for each vehicle, one for each corner of the generated vehicle;

first-person cameras positioned at eye level for each of the generated workers. Once the construction site and the actors have been generated, the active camera is iterated over all deployed cameras and all the annotations corresponding to the entities present inside the frames are saved. The stored data are related to the 2D and 3D bounding boxes, the distance between the entity and the camera, the body joints of the workers. Once the iteration of all the generated cameras is finished, the entities are deleted and the playable character is moved to the next position to be visited on the map until termination of all positions in the list.

As described above, the generated cameras can be grouped into three categories: third-person cameras (TPV), first-person cameras (FPV), cameras positioned on vehicles.

The third-person cameras mimic the cameras that are usually placed on the perimeter of the construction site to have a top view of the area (usually used for surveillance purpose). These cameras are positioned at an height of 6 meters. Third-person cameras simulate wide cameras with a Field of View (FoV) of 120 degrees. The images are acquired at a resolution of 1280x720 pixels.

The first-person cameras represent the cameras that in the real settings can be worn by the workers (e.g., on the helmets). A first-person camera is simulated for each worker generated within the construction site. The workers “on the ground” were not equipped with a camera in the first person, since their being lying on the ground led to acquire frames with artifacts due to interpenetration with the ground. The first-person cameras, simulating the cameras mounted on the helmets, have a FoV of 64.67 degrees, in order to simulate a HoloLens2 camera. The images captured by the cameras from the first person point of view are acquired at a resolution of 1280x720 pixels.

The cameras positioned on the vehicles are four per vehicle and they are positioned at an elevated position and rotated in order to have a view of everything that surrounds the vehicle. The cameras were positioned as if they were physically present at the 4 corners of the vehicle to understand if a worker is too close to a moving vehicle. The cameras positioned on the vehicles have a FoV of 120 degrees. The images are captured by the cameras positioned on the vehicles at a resolution of 1280x720 pixels.

While the plugin is running, different views of the scene are displayed, one for each acquisition point. These views are obtained by activating, deactivating and moving the created virtual cameras. For each generated view two files are created: a screen capture saved in JPG format and a text file containing the annotations for each entity present within the view.

The 2D and 3D bounding boxes are labeled with the following 12 classes: *head with work helmet*, *worker*, *torso with high visibility vest*, *pneumatic hammer*, *vehicle*, *head without work helmet*, *torso without high visibility vest*, *cone*, *worker on the ground*, *shovel*, *wheelbarrow*, *concrete mixer*.

For each labeled entity, distance from the camera was measured as the length of the segment connecting the camera position to the center of the 3D bounding box of the entity.

The worker joints that have been labeled are the following: *nose*, *neck*, *left\_clavicle*, *right\_clavicle*, *left\_thigh*, *right\_thigh*, *left\_knee*, *right\_knee*, *left\_ankle*, *right\_ankle*, *left\_wrist*, *right\_wrist*, *left\_elbow*, *right\_elbow*.

## 4 BENCHMARK

We tested the quality of the synthetic dataset generated by the proposed plugin running benchmarks on three tasks: safety compliance through object detection, fall detection through pose estimation and distance regression from monocular view. Experiments have been performed on both synthetic dataset and annotated real data. These last have been used for fine-tuning and for the evaluation of the algorithms.

### 4.1 Dataset

The dataset was collected by generating 200 building sites within the game map. In total, 76,580 frames were generated, with 44,580 in FPV (workers), 16,000 frames in FPV (vehicles) and 16,000 frames in TPV (construction site corners). The dataset contains 2,438,566 labels distributed as follows: 333,856 workers, 168,686 heads with helmet, 165,867 busts without high visibility vest, 20,454 pneumatic hammers, 35,850 vehicles, 165,170 heads without helmet, 167,989 busts without high visibility vest, 1,085,729 cones, 135,084 ground workers, 78,584 shovels, 39,999 wheelbarrows and 41,298 concrete mixers. Some of these images were discarded for occlusions or other glitches, bringing the final count to 51,081 synthetic images splitted in training (30,019), and validation (21,062).

The final dataset contains also a grand total of 9,698 real images splitted in training set (9,212) and validation (486).



Table 1: Object Detection mAP.

	Real	Synthetic + Real
<b>10%</b>	0.819	0.853
<b>25%</b>	0.852	0.875
<b>50%</b>	0.875	0.889
<b>75%</b>	0.885	0.900
<b>100%</b>	<b>0.888</b>	<b>0.905</b>

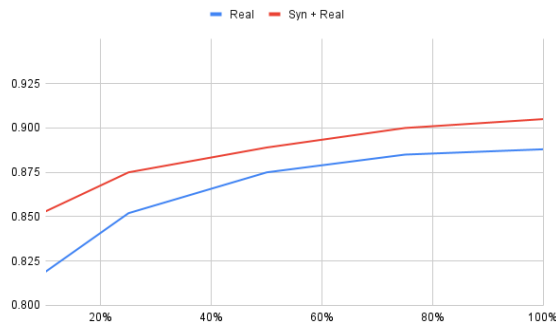


Figure 3: Object Detection mAP.

## 4.2 Safety Compliance Through Object Detection

In order to understand if all the workers in a construction site wear Personal Protective Equipment (PPE), we performed experiments using the YOLO object detector (Bochkovskiy et al., 2020) and post-processing the inferred bounding boxes to distinguish whether or not a worker is wearing a helmet and a high visibility jacket. As a first step, we analyze if and how the synthetic data can improve the quality of object detection, then we study the results obtained on the safety task.

### 4.2.1 Object Detection

In order to analyze the use of synthetic data in the object detection task, we trained and tested the object detection algorithm in two different settings. In the first setting, the model is trained using only the real data. In the second one, we train the model using synthetic data and then the real data is used to fine-tune it. In both settings, we vary the amount of real data used for training (10%, 25%, 50%, 75%, 100%) in order to assess the amount of real data needed to have a working model.

Table 1 reports the results in terms of mAP metric. Figure 3 depicts the graph showing the trend of the mAP for varying quantities of real data used to train the model.

As can be seen, the use of synthetic data helps to increase the performance of the model. We can observe that when we use all the synthetic data and fine-

Table 2: No Helmet mAP.

	Real	Syn + Real
<b>10%</b>	0.723	0.759
<b>25%</b>	0.757	0.799
<b>50%</b>	<b>0.800</b>	0.798
<b>75%</b>	0.796	0.807
<b>100%</b>	0.799	<b>0.819</b>

Table 3: No Vest mAP.

	Real	Syn + Real
<b>10%</b>	0.880	0.889
<b>25%</b>	0.892	0.899
<b>50%</b>	0.913	0.904
<b>75%</b>	0.906	0.904
<b>100%</b>	<b>0.918</b>	<b>0.915</b>

tune using 50% of the real data, the detector performs at the same manner than when using 100% of only real data (88.9% vs 88.8%).

### 4.2.2 No Vest / no Helmet

The settings of the experiments are the same of object detection. Tables 2 and 3 show the detection results of the absence of the helmet and absence of the high visibility vest, whereas Figure 4 shows a qualitative example using a real image.

Results show that helmets detection takes advantage from the synthetic data. With 25% of real data for fine-tuning, the detector reaches an accuracy value close to 100% of only real data.

Vest detection, on the other hand, benefits less from synthetic data, with best results obtained with only real data.

## 4.3 Fall Detection Through Pose Estimation

The estimated positions of the human joints in conjunction with the bounding box around the human can be used to classify workers in two classes: “no Fall” and “Fall”. The choice to use both the bounding boxes and the human body joints was driven by the fact that human pose estimation algorithms could find only a subset of joints at inference time, thus the exploitation of bounding box can improve the final quality.

We used OpenPifPaf (Kreiss et al., 2021) to infer human pose and a simple Multilayer Perceptron to classify worker status. We performed four tests, the results of which are reported in Table 4:

1. Training joint ground truth labels (GT) and testing on validation joint ground truth labels (GT). This is the baseline case.



Figure 4: No Vest/No Helmet detection on a real image.



Figure 5: Fall detection on a synthetic image.

2. Training joint ground truth labels (GT) and test on inferred joint validation labels (INF). In this case ground truth labels and inferred labels may vary a lot, e.g. many keypoints are not found.
3. Training on inferred joint labels (INF) of the training-set and testing on the validation ground truth labels (GT). In this case, we measure what happen when there are no labels for training data.
4. Training on inferred joint labels (INF) of the training-set and testing on the validation set inferred labels (INF).

In Table 5 and Table 6 we present the performance on the experiments (2 - 4) varying the distance from the camera of the worker annotated boxes. Best results are for workers at a distance under 5 meters from the camera. Learning from the inferred labels increase robustness to missing values. Figure 5 shows a qualitative example using a synthetic image.

#### 4.4 Distance Regression from Monocular View

We evaluated distance regression using monocular view on the created synthetic dataset. We followed the work in (Haseeb et al., 2018) who used a multi-layer perceptron of 3 hidden layers with 100 neurons each. The network to regress the distance needs the average 3D bounding box size in the real world, for the Worker class. It has been set an average dimensions of 1.75 m, 0.55 m, 0.30 m. In Table 7 we show the results with 3 different training setups: using images with boxes at every distance, at no more than 10m and at no more than 5m. In the first case an average error of 1.73m is obtained.

Table 4: Results of the four test for fall detection.

Train	Test	no Fall Boxes	Fall Boxes	Accuracy no Fall	Accuracy Fall	Average Accuracy
GT Labels	GT Labels	100,562	41,105	<b>0.996</b>	<b>0.938</b>	<b>0.979</b>
GT Labels	INF Labels	41,317	2,666	0.708	0.551	0.698
INF Labels	GT Labels	100,562	41,105	0.968	0.236	0.756
INF Labels	INF Labels	41,317	2,666	0.979	0.877	0.973

Table 5: GT Labels vs INF Labels varying distance.

Distance	no Fall Boxes	Fall Boxes	Accuracy no Fall	Accuracy Fall	Average Accuracy
< 2m	625	62	0.242	<b>0.998</b>	<b>0.930</b>
< 5m	6,123	767	<b>0.939</b>	0.494	0.890
< 10m	19,999	2,329	0.800	0.564	0.775
All	41,317	2,666	0.708	0.551	0.698

Table 6: INF Labels vs INF Labels varying distance.

Distance	no Fall Boxes	Fall Boxes	Accuracy no Fall	Accuracy Fall	Average Accuracy
< 2m	625	62	0.978	0.903	0.971
< 5m	6,123	767	<b>0.986</b>	<b>0.952</b>	<b>0.982</b>
< 10m	19,999	2,329	0.981	0.905	0.973
All	41,317	2,666	0.979	0.877	0.973

Table 7: Results of the network using as training set images of workers at all possible distances (a), with workers at no more than 10m (b) and at no more than 5m (c).

Distance	All Images (a)	Under 10m (b)	Under 5m (c)
< 1m	0.76m (91%)	0.48m (57%)	0.37m (44%)
< 2m	0.83m (63%)	0.60m (44%)	0.52m (37.15%)
< 5m	1.21m (39%)	0.93m (30%)	0.53m (18%)
< 10m	2.28m (35%)	0.88m (16%)	-
all	1.73m (16%)	-	-

## 5 CONCLUSIONS

In this work, we presented a pipeline to generate synthetic data in the domain of construction sites using the Grand Theft Auto V videogame graphics engine. A benchmark of the generated dataset on three different tasks has been also performed. We focused the study on training machine learning algorithms using a large amount of synthetic data and a small set of real images with the aim of measuring the usefulness of such data to reduce real labeling effort without decreasing inference quality, evaluating algorithms behaviour varying the amounts of real data used. The results show that the use of synthetic data is a viable way to reduce the need to acquire and label new real data.

## ACKNOWLEDGEMENTS

This research is supported by project SAFER developed by Xenia Network Solutions s.r.l. (GRANT: CALL N3 ARTES 4.0 - 2020) and by Next Vision s.r.l.

## REFERENCES

- Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolo4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- Di Benedetto, M., Meloni, E., Amato, G., Falchi, F., and Gennaro, C. (2019). Learning safety equipment detection using virtual worlds. In *International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE.
- Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR.
- Fabbri, M., Brasó, G., Maugeri, G., Cetintas, O., Gasparini, R., Ošep, A., Calderara, S., Leal-Taixe, L., and Cucchiara, R. (2021). Motsynth: How can synthetic data help pedestrian detection and tracking? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10849–10859.
- Haseeb, M. A., Guan, J., Ristic-Durrant, D., and Gräser, A. (2018). Disnet: a novel method for distance estimation from monocular camera. *10th Planning, Perception and Navigation for Intelligent Vehicles (PP-NIV18), IROS*.
- Hu, Y.-T., Chen, H.-S., Hui, K., Huang, J.-B., and Schwing, A. G. (2019). Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3105–3115.
- Hu, Y.-T., Wang, J., Yeh, R. A., and Schwing, A. G. (2021). Sail-vos 3d: A synthetic dataset and baselines for object detection and 3d mesh reconstruction from video data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1418–1428.
- Jayaswal, R. and Dixit, M. (2022). Monitoring social distancing based on regression object detector for reducing covid-19. In *2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT)*, pages 635–640. IEEE.
- Juraev, S., Ghimire, A., Alikhanov, J., Kakani, V., and Kim, H. (2022). Exploring human pose estimation and the usage of synthetic data for elderly fall detection in real-world surveillance. *IEEE Access*, 10:94249–94261.
- Kim, K., Kim, S., and Shchur, D. (2021). A uas-based work zone safety monitoring system by integrating internal traffic control plan (itcp) and automated object detection in game engine environment. *Automation in Construction*, 128:103736.
- Krähenbühl, P. (2018). Free supervision from video games. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2964.
- Kreiss, S., Bertoni, L., and Alahi, A. (2019). Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11977–11986.
- Kreiss, S., Bertoni, L., and Alahi, A. (2021). Openpifpaf: Composite fields for semantic keypoint detection and spatio-temporal association. *IEEE Transactions on Intelligent Transportation Systems*.
- Leonardi, R., Ragusa, F., Furnari, A., and Farinella, G. M. (2022). Egocentric human-object interaction detection exploiting synthetic data. In *International Conference*

- on *Image Analysis and Processing*, pages 237–248. Springer.
- Pasqualino, G., Furnari, A., Signorello, G., and Farinella, G. M. (2021). An unsupervised domain adaptation scheme for single-stage artwork recognition in cultural sites. *Image and Vision Computing*, page 104098.
- Quattrocchi, C., Di Mauro, D., Furnari, A., and Farinella, G. M. (2022). Panoptic segmentation in industrial environments using synthetic and real data. In *International Conference on Image Analysis and Processing*, pages 275–286. Springer.
- Richter, S. R., Hayder, Z., and Koltun, V. (2017). Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2213–2222.
- Sandru, A., Duta, G.-E., Georgescu, M.-I., and Ionescu, R. T. (2021). Super-sam: Using the supervision signal from a pose estimator to train a spatial attention module for personal protective equipment recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2817–2826.
- Sankaranarayanan, S., Balaji, Y., Jain, A., Lim, S. N., and Chellappa, R. (2018). Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al. (2019). Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347.
- Szot, A., Clegg, A., Undersander, E., Wijmans, E., Zhao, Y., Turner, J., Maestre, N., Mukadam, M., Chaplot, D. S., Maksymets, O., et al. (2021). Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in Neural Information Processing Systems*, 34:251–266.
- Taufeeque, M., Koita, S., Spicher, N., and Deserno, T. M. (2021). Multi-camera, multi-person, and real-time fall detection using long short term memory. In *Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications*, volume 11601, pages 35–42. SPIE.
- Wu, J., Cai, N., Chen, W., Wang, H., and Wang, G. (2019). Automatic detection of hardhats worn by construction personnel: A deep learning approach and benchmark dataset. *Automation in Construction*, 106:102894.