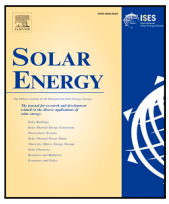
Solar Energy 300 (2025) 113728

Contents lists available at ScienceDirect

Solar Energy

journal homepage: www.elsevier.com/locate/solener



Bridging the sim2real gap: Training deep neural networks for heliostat detection with purely synthetic data

Rafal Broda [a,d,*], Alexander Schnerring [a,d], Dominik Schnaus [e], Michael Nieslony [a], Julian J. Krauth [a], Marc Röger [a], Sonja Kallio [a], Rudolph Triebel [c,f], Robert Pitz-Paal [b,d]

[a] German Aerospace Center (DLR), Institute of Solar Research, Almería 04005, Spain
[b] German Aerospace Center (DLR), Institute of Solar Research, Köln 51147, Germany
[c] German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Wessling 82234, Germany
[d] RWTH Aachen University, Chair of Solar Technology, Köln 51147, Germany
[e] Technical University of Munich (TUM), Chair of Computer Vision and Artificial Intelligence, Garching 85748, Germany
[f] Karlsruhe Institute of Technology (KIT), Institute for Anthropomatics and Robotics, Karlsruhe 76131, Germany

ARTICLE INFO

Keywords:
Heliostat
Deep learning
Object detection
Keypoint detection
Sim2real
Photorealism

ABSTRACT

Deep neural networks have demonstrated remarkable success in image processing across various domains. However, to achieve state-of-the-art performance, a substantial amount of high-quality training data is essential. In the context of optical heliostat monitoring, acquiring such data remains a challenge which is why deep neural networks are still scarcely used. We propose the use of synthetic training data to address this deficit and conduct a comprehensive investigation of scene parameters within our simulation environment to mitigate the sim2real gap. Our findings demonstrate that training models for object and keypoint detection in aerial images of heliostat fields with purely synthetic data is feasible and yields promising results with the appropriate scene configuration. Our best model achieves an average precision (AP) of 0.63 in heliostat detection and accurately detects 61% of outer mirror corners on our test dataset, comprising six manually annotated real-world drone images of a heliostat field. By evaluating the model on a simulated replication of this test dataset, we measure a remaining sim2real gap of 30% and 35% for the respective tasks. Furthermore, we showcase the model's transferability to other heliostat geometries. By generating an additional 200 synthetic images showing the new geometry and performing a brief fine-tuning of the model, we achieve promising qualitative results on real-world images of another plant. To the best of our knowledge, this work is the first application of deep learning achieving such results in mirror corner detection in airborne imagery of heliostat fields while offering a straightforward approach for power plant transfer.

1. Introduction

Solar tower power plants represent an important technology in the transition toward a decarbonized energy system. These plants typically comprise thousands to over hundred thousands of individual mirror modules, known as heliostats, which concentrate solar radiation onto a central tower receiver. This process enables the generation of high temperatures for electricity production in a steam power cycle, cost-effective thermal energy storage, or direct utilization as process heat. To ensure optimal efficiency, heliostats must be aligned with an accuracy of ≤ 1 mrad during the construction phase, with regular calibration recommended during operation to maintain this level of accuracy [1]. Drones equipped with high-resolution cameras have emerged as a promising solution for these tasks. During flight, they capture images of the heliostat field which are subsequently analyzed. Current measurement methods typically involve detecting heliostat mirror corners for photogrammetry-based camera pose estimation and, in some cases, coarse calibration. Additionally, reflex-based techniques analyze reflexes such as edges or point markers in the concentrator mirrors to determine heliostat orientation and surface slope. Different techniques are presented in [2–5].

To the best of our knowledge, existing approaches for mirror corner detection in the aforementioned works primarily rely on classical computer vision algorithms, which involves significant limitations. A missing or incorrect prior knowledge of the heliostat orientation can lead to the misplacement of search windows, which are required, for example, for the image processing pipeline for corner detection described by Prahl [6] and Jessen et al. [4]. Moreover, reflections

* Corresponding author at: German Aerospace Center (DLR), Institute of Solar Research, Almería 04005, Spain.
E-mail address: rafal.broda@dlr.de (R. Broda).

https://doi.org/10.1016/j.solener.2025.113728
Received 31 January 2025; Received in revised form 29 May 2025; Accepted 21 June 2025
Available online 25 July 2025
0038-092X/© 2025 The Authors. Published by Elsevier Ltd on behalf of International Solar Energy Society. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Acronyms

AP	Average precision
CNN	Convolutional neural network
DNN	Deep neural network
GAN	Generative adversarial network
GPU	Graphics processing unit
HDRI	High dynamic range image
HPC	High performance computing
IoU	Intersection over union
PCK	Percentage of correct keypoints
PD	Pixel deviation
PSA	Plataforma Solar de Almería
SDR	Structured domain randomization
STJ	Solar Tower Jülich
ViT	Vision transformer

other than the sky and overlapping objects can lead to false detections when using color thresholding and edge detection algorithms as proposed by Röger et al. [7]. These errors can lead to wrong conclusions and correcting them requires manual intervention, which delays the processing substantially and may introduce further inaccuracies. Not least, the use of classical computer vision algorithms, especially if performed iteratively for all objects or search windows, can result in relatively long processing times. These are all factors that undermine the application as a fast and ideally automated measurement method. For this reason, the topic was introduced in 2022 in the German AuSeSol-AI project [8]. Also, the HelioCon roadmap [9] underscores the need for further research in this domain to develop more efficient monitoring solutions for commercial power plants.

By means of deep neural network (DNN) algorithms the issues of classical computer vision methods can be circumvented opening a promising direction for further development. However, in the context of solar thermal collectors, the application of DNNs remains relatively unexplored, with only a handful of studies addressing their potential. Existing research primarily focuses on training models for object detection to determine heliostat tracking angles by means of cameras mounted on heliostats [10] or instance segmentation, which is subsequently combined with classical computer vision algorithms to calculate mirror corners and perform calibration [11–13]. These approaches, however, do not offer a fast, robust, and, above all, transferable image processing method. Adapting the presented models to different power plants, where heliostat geometries can vary, requires the acquisition and annotation of new image datasets, which poses significant challenges. On the one hand, access to image data from commercial or research power plants is limited and publicly available datasets are scarce. On the other hand, even when data is accessible, the manual annotation is both, error-prone and labor-intensive. Especially annotating mirror corners demands a high level of accuracy. We therefore propose the development of DNNs for object detection (heliostat) and keypoint detection (outer mirror corner) trained purely on synthetic datasets of aerial images of heliostat fields.

In this work, we investigate the sim2real gap associated with synthetic training data, i.e., the discrepancy in performance when a model trained in simulation is transferred to reality. Building on our previously introduced simulation environment for photorealistic image data generation [14], we systematically analyze how scene parameters, such as object placement, textures, and lighting, affect model performance for the aforementioned tasks. The primary objective of this work is to identify the optimal scene parameter configuration for training dataset generation with the aim to minimize the sim2real gap and improve real-world applicability.

2. Related work

An integral part of every deep learning problem is the availability of a sufficient amount of training image data. Over the past decade, the use of synthetic data to meet this demand and the investigation of the resulting sim2real gap has become a prominent area of research in deep learning. This approach is particularly prevalent in fields such as robotics [15], manufacturing [16,17], and autonomous driving [18–20], especially for tasks like object detection and 6D pose estimation, where acquiring accurate, diverse and, most importantly, annotated training data in sufficient quantities is particularly difficult and costly. In theory, generating synthetic data not only allows for virtually unlimited dataset sizes but also offers the advantage of being able to control diversity and automatically generate perfect annotations.

One of the most straightforward techniques for generating synthetic image data is known as cut-and-paste, or render-and-paste. In this approach, objects are either cut out from existing images or rendered, and then placed on an arbitrary background [21,22]. While this approach is highly scalable, it has the disadvantage of losing important effects, such as realistic lighting, shadows, interreflections, and a natural contextual relationship between objects.

With the increasing realism of computer game graphics, game rendering engines such as *Unreal Engine* [23] have gained attention as tools capable of reproducing these effects. However, these engines are primarily designed to function efficiently in real-time, which comes at the cost of photorealism. At the same time, numerous works have demonstrated the advantages of photorealism in context with DNNs and the sim2real gap [24–26]. Photorealism can be achieved using ray tracing or path tracing-based rendering, utilizing physically based materials, as supported by software such as *Blender* [27]. Driven by these insights, a growing number of works have introduced pipelines for generating photorealistic synthetic images for deep learning using such software [28–30].

Photorealistic rendering can be seen as a form of domain adaptation, where one domain (in this case, simulation) is adapted to another (reality). A promising area of research in domain adaptation involves the use of DNNs, such as generative adversarial networks (GANs), to further enhance the photorealism of simulated data [31,32]. However, this approach is still in its early stages and designing and training such models remains challenging as stated by Hinterstoisser et al. [33].

A more practical approach is domain randomization, first introduced by Tobin et al. [34]. The concept behind domain randomization is to vary the scene parameters to such an extent that a trained model regards reality as just another variation of this randomization. The authors reported a substantial reduction in the sim2real gap for an object detection model, albeit only for the detection of simple geometric shapes. Building on these initial findings, subsequent studies have explored the combination of photorealism and domain randomization to tackle more complex tasks. In the work of Prakash et al. [35], the sim2real gap was categorized into two components: the appearance gap and the content gap. The appearance gap is defined as the pixel-level discrepancy between two images, encompassing differences in object detail, materials, and the capabilities of the rendering system. In contrast, the content gap refers to differences in the number of objects in a scene, their diversity and placement, and other contextual factors. The study demonstrated that context-aware object placement outperforms full randomization in vehicle detection, introducing the concept of structured domain randomization (SDR).

Building on these ideas, more recent studies have further explored scene parameters and their impact on the sim2real gap. Eversberg et al. [36] trained an object detection model on texture-less turbine blades. Synthetic training data was generated by means of Blender and scene parameters such as lighting, background, object materials, and foreground objects were investigated. Their findings suggest that realistic image backgrounds and realistic distractor objects are less critical, while realistic image-based lighting and realistic object materials play a

significant role in improving transferability from simulation to reality. Overall, their work emphasizes the significance of photorealism but highlights that variability in scene design is also important. In the work of Mayershofer et al. [26], an object detection model was trained based on synthetically generated data from Blender to identify various industrial objects. The study presented a scalable image generation approach that incorporates varying backgrounds, distractor objects, and lighting, all subject to domain randomization. Through an ablation study, the authors concluded that the inclusion of distractors and predefined relations when positioning the target objects is crucial for the model's transferability to reality. Another study aimed to develop a scalable pipeline using Blender and *BlenderProc* [30], incorporating procedures based on domain randomization for detecting parts and assemblies in a production environment [17]. The authors evaluated five different procedures involving scene parameters such as predefined and random object poses, object materials, background objects and their textures, as well as scene lighting. They found that combinations of different procedures outperformed individual ones, making it challenging to isolate the effect of each scene parameter on the sim2real gap. Nevertheless, their study demonstrates that combining domain randomization with domain knowledge can enhance the transferability from simulation to reality.

To the best of our knowledge, only two works directly measure the sim2real gap. In the study of Reway et al. [20], an object detection and tracking model was applied to both real-world video data captured during a vehicle drive and its simulation counterpart, and the difference in model performance was measured. To condense the various resulting metrics into a single value, they were plotted on a radar chart and the sim2real gap was measured as the intersection over union (IoU) between the radar charts of the real-world and simulated datasets. The second study on measuring the sim2real gap focused on object detection for industrial objects [37]. In that case, the sim2real gap was determined by calculating the difference in AP between a synthetic validation dataset and a real-world test dataset. However, as the two datasets differed in various ways, the resulting difference likely included factors beyond the sim2real gap.

In summary, many studies focus on automating dataset creation using domain randomization. However, it is evident that, along with domain randomization, domain knowledge and context play a crucial role, and manual creation or control of a scene can be beneficial in bridging the sim2real gap. Within this context only a few studies provide specific recommendations for scene parameters. These works generally focus on relatively simple objects in controlled production environments and, more importantly, do not directly measure the sim2real gap to draw their conclusions. Our goal is to address these research gaps by deriving scene design recommendations based on a directly measured sim2real gap for a more complex scenario involving reflective objects.

3. Methodology

Our methodology comprises three main components. We create scenes of simulated heliostat fields and define parameters of interest which modify their properties. For the object and keypoint detection tasks, we select a suitable model with appropriate training settings. Each model trained on synthetic images from a specific scene configuration constitutes an experiment. Finally, we establish an evaluation procedure with meaningful metrics to measure the sim2real gap and assess the model's real-world applicability.

3.1. Scene design and rendering

For scene creation and rendering, we utilize our previously presented simulation environment for photorealistic image data generation [14]. The environment is built on Blender 3.5 and BlenderProc, with custom extensions that include functionalities such as the generation of keypoint annotations. All scenes presented in the following are rendered in a highly parallelized manner on an HPC cluster equipped with *NVIDIA A100 SXM4 80 GB* GPUs.

Fig. 1. Visual representation of the heliostat geometries used in our study. From left to right: 2 × 6 collector, 4 × 6 collector and the backside of the 4 × 6 collector. Exemplarily, the right side of the 2 × 6 collector is shown with soiling on the mirror facets.

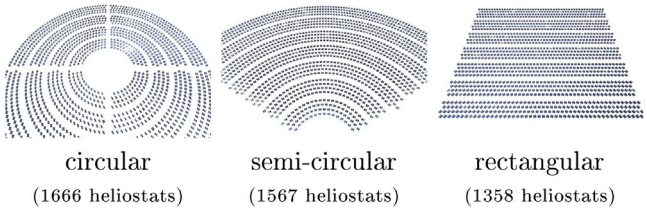


Fig. 2. A bird's eye view of the three different field arrangements with the respective number of heliostats indicated.

Objects. The target object of our investigation is the heliostat. We use our in-house parametric 3D heliostat model which allows us to freely modify the collector geometry and imitate soiling on the mirror facets. Heliostat tracking is achieved by adjusting both the azimuth and elevation angles. For this study, we create heliostats with two types of collectors: one featuring 2 × 6 mirror facets and another featuring 4 × 6 mirror facets, both divided vertically by a middle gap. These geometries correspond to those present in our real-world evaluation data (see Section 3.3). The resulting heliostat models, illustrated in Fig. 1, appear in equal proportions in our scenes.

In addition, we incorporate distractor objects that naturally occur in solar tower power plants, such as solar towers, storage units, buildings, vehicles, vegetation and other smaller objects. The 3D models and all other assets used in this work are either self-created or obtained from *Poly Haven* [38] under a CC0 license. A visual overview of all distractor objects is provided in Appendix A.

Base Scenes. In line with literature recommendations to design scenes both realistically and with variability, we introduce ten distinct "base scenes". This measure helps prevent our models from overfitting to specific scene attributes, such as a fixed arrangement of objects or the ground color. Every experiment conducted in this study builds upon these ten base scenes.

To maintain a realistic framework, we use real solar tower power plants as reference. For heliostat placement, we use three different field arrangements — circular, semi-circular and rectangular (see Fig. 2). For the ground, we use a flat plane and incorporate four different image-based textures: grass, sand, earth and concrete (see Fig. 3). In addition, we simulate realistic heliostat orientations in our base scenes by aligning the heliostats with a specific sun position. For these calculations, we assume three different timestamps: a March morning with the rising sun, a June midday, and a December midday. The assumed solar tower is located at the heliostat field coordinate system's origin, at latitude 36.838 and longitude −2.460. Note that the timestamps are only used to compute the heliostat orientations and do not affect the sky or lighting. Following the recommendation of Eversberg et al. [36], we employ a HDRI of a clear midday sky for lighting (see Fig. 4). In our experience, this represents the standard condition for a measurement flight, which is why we do not yet vary the lighting in our base scenes. All other scene attributes are combined with each other in different ways. The resulting base scene configurations are presented in Table 1.

Fig. 3. Comparison of the four different ground textures. Top row: simple textures based on repeating image patches. Bottom row: procedural, randomized ground textures with 3D effects.

Fig. 4. Comparison of the HDRIs used in the base scene and in the scenario with variations in lighting. Note that the shown images are thumbnails and that the HDRIs cover the entire sphere in the simulation.

For each scene, we simulate 1000 random camera poses representative of a measurement flight, resulting in a total of 10 000 images per experiment. We model an ideal camera with a resolution of 6000 × 4000 px, corresponding to our test data, and the following *OpenCV* camera calibration parameters: $f_x = f_y = 4000$ px, $c_x = 3000$ px, and $c_y = 2000$ px. The camera's (x, y) position is uniformly sampled within the heliostat field. For flight altitudes z, we define four equidistant intervals of 20 m width between 20 m and 100 m ([20, 40] m, [40, 60] m, [60, 80] m, [80, 100] m) and distribute 250 poses evenly across each altitude interval. For the camera orientation, we sample a random point of interest in the field at a height of 0–5 m above ground, from which we compute the camera's pitch angle (θ) and yaw angle (ψ). This ensures that the camera is always oriented downwards. As is typical for such measurement flights, the camera's roll angle (ϕ) is set to zero. Once sampled, these camera poses remain fixed throughout all experiments.

Scene Parameters. Following the work of Prakash et al. [35], we subdivide the scene parameters for our experiments into two categories: appearance and content. According to the definition, appearance encompasses parameters affecting the visual details of the scene or objects. In our case, these include the use of procedural ground textures, varying lighting and soiling on the heliostat mirror surfaces. Content parameters comprise distractor objects, random orientations, and random positions. In the context of the sim2real gap, appearance and content are considered orthogonal [19], allowing us to evaluate them separately. Each individual parameter modifies or enhances the previously defined base scenes, thereby increasing rendering time. A side-by-side comparison of the base configuration (×) and the enhancements (✓), along with a detailed description of their effects, is provided in Table 2. In total, we investigate the influence of six scene parameters on the sim2real gap, divided into the two aforementioned categories.

3.2. Model and training setup

A variety of models for object and keypoint detection can be found in the literature. For our purposes, we select the encoder–decoder model architecture with heatmap output proposed in the work of Zhou et al. [39]. Compared to other widely adopted models such as Mask R-CNN [40] or YOLO [41], it offers several advantages for our particular use case. One major advantage of this architecture is its flexibility. The encoder network can easily be replaced by lightweight, fast convolutional neural networks (CNNs) like MobileNet [42] – useful for real-time applications on a drone – or by more recent transformer-based architectures, such as vision transformer (ViT) [43]. Additionally,

Table 1
Overview of the configuration of the created base scenes. Ground texture, field arrangement and time (UTC+0) are combined in such a way that a high degree of variability is achieved.

Scene	1	2	3	4	5	6	7	8	9	10
Ground	Grass	Grass	Grass	Sand	Sand	Sand	Soil	Soil	Soil	Concrete
Field	○	⊓	□	○	⊓	□	○	⊓	□	○
Time	M	J	D	J	D	M	D	M	J	M

M ≙ March 1st, 8:00; J ≙ June 1st, 12:00; D ≙ December 1st, 12:00.

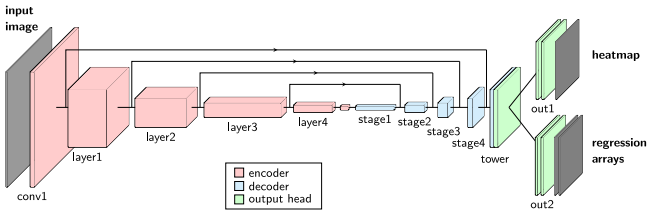


Fig. 5. Simplified representation of the employed encoder–decoder model architecture with a ResNet50 encoder, an output stride of 2 and extended by skip connections.

the output stride of the model (i.e. the ratio between the input image resolution and the output heatmap resolution) is easily adjustable by modifying the number of decoder stages, enabling the production of higher-resolution heatmaps for more precise keypoint detections. Lastly, the model generates its output in a single pass, which is faster than two-stage methods like Mask R-CNN. Detailed information on the model architecture can be found in the original publication.

For our experiments, we use a ResNet50 [44] encoder, pretrained on ImageNet [45] and adopt an output stride of 2. Inspired by U-Net [46], we also incorporate skip connections. Our preliminary tests suggest that these settings yield the best performance. Specifically, skip connections add virtually no extra computational cost, yet significantly improve the detection of smaller objects, making them a valuable addition to our model. Furthermore, we equip the model with four output heads: two for object detection (one peak heatmap for integer pixel positions (u, v) and one 2-channel regression array for sub-pixel offsets $(\Delta u, \Delta v)$) and two corresponding heads for keypoint detection. The resulting model architecture, with only two of the four output heads depicted, is shown in Fig. 5.

In terms of training settings, we largely follow the original model paper [39]. We adopt the same training objective and transform our object and keypoint annotations accordingly, with one exception: to generate scale-aware heatmaps, we scale the Gaussian Kernel standard deviations based on the distance between the camera and the object. For data augmentation, we apply a random scale-and-crop transform that first resizes the input image by a factor between 0.5 and 2.0 to account for scale invariance, then crops a 2048 × 2048 px patch at a random position to optimize GPU utilization. We use the Adam optimizer [47] with a one-cycle policy [48] and a maximum learning rate of 1e−3. Each model is trained for 200 epochs on eight *NVIDIA A100 SXM4 80 GB* GPUs with an effective batch size of 64. To mitigate effects of random initialization and optimization, each evaluated model is trained with five different random seeds. Training one model takes approximately 19 h.

3.3. Evaluation procedure

The primary objective of our evaluation is to quantify and minimize the sim2real gap. Achieving this requires both representative real-world drone images of a heliostat field with corresponding accurate annotations and a precise replication of those images within our simulation environment. By doing so, we can use the simulated case as a reference and directly measure the sim2real gap, free from distorting factors such as scale variance in the model or potential dataset bias.

Table 2
Summary of the scene parameters relevant for this study, along with a detailed description of their main effects on the scene.

		Base (×)	Enhancement (✓)
Appearance	Procedural ground texture	• simple image-based texture with a repetitive pattern	• Randomized texture with 3D details such as bumps and displacement
	Varying lighting	• Uniform illumination without clouds and with a midday sun, resulting in less pronounced shadows	• Partly cloudy skies and three different sun positions (see Fig. 4) resulting in color shifts and more pronounced shadows
	Soiling	• Clean heliostat surface with perfect mirror reflectivity	• Realistically reduced mirror reflectivity especially in the bottom part of mirror facets, gradually increasing toward the top of the facet (see Fig. 1) • Randomized levels of soiling throughout the whole field
Content	Distractor objects	• Only heliostats in the scene, i.e. only reflections of the sky, neighboring heliostats and the ground are present	• A variety of realistic distractor objects added to the scene resulting in a more diverse scene representation as well as more diverse reflections in the mirrors
	Random orientations	• Except for a 5% share of outliers, the heliostats are perfectly aligned to reflect the sun onto the tower • Neighboring heliostats appear very similar and the occurrence of interreflections is negligible	• Heliostat azimuth and elevation angles are set randomly • Reflections in the mirrors are highly diverse and interreflections occur frequently
	Random positions	• Heliostats are placed according to the previously presented field arrangements • Other objects are placed in a realistic manner in and around the heliostat field	• All objects are placed randomly on the ground plane of the scene resulting in a change of context

Fig. 6. Top row: masked images from Jessen et al.'s [4] measurement flight at the PSA (owned and operated by CIEMAT), selected for the evaluation performed in this work. Bottom row: replication of the images in our simulation environment.

We use an existing dataset recorded on 2020-07-09 at 14:00–16:00 local time as part of a measurement flight conducted by Jessen et al. [4] over the *CESA-1* heliostat field at the Plataforma Solar de Almería (PSA; owned and operated by CIEMAT). This enables us to leverage the results from that work as a starting point. From that dataset, we reduce the number of images to six, selecting them to achieve a balance between capturing the essential camera perspectives of the flight and minimizing redundancy. In the selected images, we mask the rear heliostat rows which contain heavily corroded mirror facets. This level of corrosion is uncommon in operating plants and could otherwise distort our results. We then thoroughly create and refine bounding box and keypoint annotations for all images using *labelme* [49], concluding with a double-check to ensure high accuracy. The resulting test dataset contains 380 heliostat bounding boxes and 1391 individual outer mirror corner keypoints in total.

For replication in the simulation environment, we use the optimized camera poses and heliostat orientations from the existing measurement, along with the corresponding sun position, a procedural ground texture, and carefully reconstructed scene objects. Fig. 6 shows the selected real-world images along with their simulated counterparts. It is important to note that, due to the inherent difficulties of real-world data collection, our evaluation dataset is relatively small and does not encompass every possible condition (e.g., atmospheric variations, time of day, ground composition, or heliostat geometry) that could occur during a measurement flight. However, we demonstrate below how our method can be easily adapted to other conditions with little effort.

To evaluate object detection performance, we use the state-of-the-art metric AP computed over an IoU range of [0.50: 0.05: 0.95]. For the keypoint detection of the four outer mirror corners, we employ the PCK metric with a distance threshold of 3 px, as in [11], and a model confidence threshold of 50%. Under these criteria, a detected keypoint is counted as true positive if it lies within 3 px of the ground-truth point and has a confidence (peak heatmap value) above 0.5. For all true positives, we then compute the mean pixel deviation (PD), which is the Euclidean distance between the correctly detected keypoint and its ground-truth location.

Each model is trained exclusively with synthetic data from the respective experiment, and evaluated on the six real-world images and their simulated counterparts respectively. The sim2real gap for a given metric m is then calculated as follows:

$$m_{\text{gap}} = \frac{m_{\text{sim}} - m_{\text{real}}}{m_{\text{sim}}} \tag{1}$$

4. Results and discussion

4.1. Experiments

We investigate the sim2real gap in two parts. First, we focus on appearance-related scene parameters. Then, using findings from this initial analysis, we generate the data for the second part – the examination of content-related scene parameters – and continue the evaluation. Each part is studied via a full-factorial approach, resulting in a total of 15 individual experiments: eight for appearance and seven

Table 3
Evaluation results of the appearance-related scene parameters. The models were trained with exclusively synthetic training data and evaluated with real-world images (real) and their replication in the simulation (sim). The computed metrics correspond to the mean and the standard deviation of 5 models trained with different random seeds. The sim2real gap (gap) is calculated according to Eq. (1). Note that for AP and PCK, a value close to one is desirable.

Experiment	proc. ground	var. lighting	Soiling	AP_{sim}	AP_{real}	AP_{gap}	$PCK^{50,3\,px}_{sim}$	$PCK^{50,3\,px}_{real}$	$PCK^{50,3\,px}_{gap}$	$PD^{50,3\,px}_{sim}$	$PD^{50,3\,px}_{real}$
1	×	×	×	0.19 ± 0.027	0.12 ± 0.025	0.36 ± 0.117	0.07 ± 0.024	0.02 ± 0.007	0.63 ± 0.174	0.33 ± 0.04	**0.93** ± 0.08
2	×	×	✓	0.17 ± 0.028	0.12 ± 0.013	0.32 ± 0.131	0.04 ± 0.027	0.02 ± 0.006	**0.22** ± 0.488	0.33 ± 0.03	1.01 ± 0.08
3	×	✓	×	0.26 ± 0.047	0.13 ± 0.042	0.49 ± 0.094	0.05 ± 0.011	0.01 ± 0.003	0.76 ± 0.068	0.30 ± 0.02	1.13 ± 0.12
4	×	✓	✓	0.24 ± 0.040	0.11 ± 0.011	0.54 ± 0.099	0.05 ± 0.018	0.02 ± 0.006	0.63 ± 0.216	0.29 ± 0.05	1.02 ± 0.05
5	✓	×	×	0.86 ± 0.023	0.55 ± 0.033	0.35 ± 0.031	0.76 ± 0.032	0.43 ± 0.025	0.43 ± 0.031	0.29 ± 0.01	1.02 ± 0.02
6	✓	×	✓	**0.88** ± 0.014	**0.61** ± 0.033	**0.30** ± 0.037	0.80 ± 0.033	**0.49** ± 0.054	0.39 ± 0.045	0.26 ± 0.01	0.99 ± 0.04
7	✓	✓	×	0.81 ± 0.017	0.50 ± 0.029	0.39 ± 0.034	0.86 ± 0.019	0.29 ± 0.033	0.66 ± 0.039	0.27 ± 0.01	0.96 ± 0.03
8	✓	✓	✓	0.84 ± 0.015	0.57 ± 0.074	0.32 ± 0.080	**0.90** ± 0.015	0.36 ± 0.055	0.60 ± 0.055	**0.25** ± 0.01	0.96 ± 0.02

for content, with one shared experiment spanning both. A visual side-by-side comparison of one sample image per experiment is provided in Appendix B.

Appearance. Table 3 summarizes the results for the appearance-related parameters. Our first observation is that experiments 1–4, which use a simple rather than a procedural ground texture, show significantly reduced performance in both object and keypoint detection compared to experiments 5–8 — across simulated and real-world images. We hypothesize that the repeated, image-based ground texture pattern in these scenes leads the models to overfit to this pattern during training. When confronted with test images lacking this pattern, the models fail to generalize, resulting in a drop in detection performance. Consequently, for scene design we recommend the use of procedural, highly randomized textures for all objects to avoid similar overfitting.

Next, we focus on experiments 5–8, starting by evaluating the metrics for real-world images. Comparing the base lighting (experiments 5 and 6) with varying lighting (experiments 7 and 8) shows that introducing realistic but varied lighting - a form of SDR - negatively affects both object detection and, more notably, keypoint detection. The AP value drops with an associated sim2real gap increase from [30%, 35%] to [32%, 39%], while the PCK value drops with an associated sim2real gap increase from [39%, 43%] to [60%, 66%]. Since the base lighting more closely resembles the test conditions, we conclude that aligning scene design with the later application scenario can be beneficial. Randomizing the lighting, in our case, results in an adverse effect on the sim2real gap.

Regarding soiling on the mirror surfaces, experiments with soiling (6 and 8) outperform those without (5 and 7). We attribute this to the improved matching between simulated and real-world heliostats, as the soiling captures an important aspect of the actual mirror appearance. Thus, we conclude that increasing the target object's realism positively influences transferability from simulation to reality. We do not observe any significant second-order effects among these three scene parameters.

When examining the metrics derived from the simulated data, we notice that no experiment produces a "perfect" model (i.e., AP = PCK ≈1). This implies that factors beyond the sim2real gap are influencing the calculated metrics, thus confirming our previous assumptions. At the same time it also suggests potential for achieving higher detection rates. For instance, the test datasets include heliostats with damaged or missing mirror facets, a scenario not covered in the training data. In addition, one test image features a camera placed very close to certain heliostats (see Fig. 6, second image from the left) which is also not represented in the training dataset. Nevertheless, the trends in AP and PCK observed in the simulated setting follow those in the real-world evaluation.

The only discrepancy arises in the PD metric, which shows a relatively large gap between simulation and reality. While the PD of correctly detected keypoints in simulation is around 0.3 px, it increases to roughly 1 px on real-world data. A visual inspection of keypoint detections on the real-world images shows no clear systematic errors; on the contrary, in many cases, the model predictions even appear to be more accurate than the ground-truth (see Fig. 7). We therefore believe that the limited accuracy of manual annotation in the real-world dataset accounts for a significant portion of this discrepancy. This finding underscores the high quality of simulation-based annotations, which can allow models to achieve sub-pixel accuracy that outperforms human annotation. Overall, we find that experiment 6, including a procedural ground texture, base lighting, and simulated soiling, yields the best performance among the appearance-related parameters. With this configuration, we achieve an AP of 0.61 on the real-world test data for object detection and successfully detect 49% of the outer mirror corners with a PD of 0.99 px. Compared to the other experiments (excluding 1–4), the sim2real gap is minimal and amounts to 30% for AP and 39% for PCK. This scene configuration is fixed and used for the generation of the datasets for the second part of our investigation.

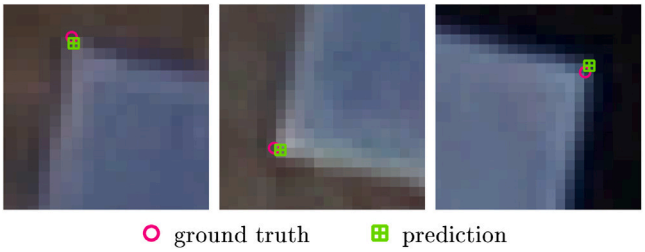


Fig. 7. Close-up images of keypoint predictions and the corresponding ground truths on real-world images.

Content. Table 4 presents the results for the content-related scene parameters. Experiment 6 serves as the starting point for this step and is also included in the table. Analyzing the metrics for both object and keypoint detection on real-world test data reveals that experiments featuring random orientations (10, 11, 14, 15) yield the poorest performance, with AP values in the range 0.27–0.43, while the remaining experiments (6, 9, 12, 13) achieve AP values of 0.55–0.63. The sim2real gap is also substantially higher for the former experiments, reaching up to 68%. Intuitively, one might expect that these models would handle heliostats with reflections other than the sky more effectively, but a comparison of predictions from experiments 6 and 10 shows otherwise (see Fig. 8). The strong randomization of object orientations introduces frequent interreflections that rarely occur in the test data. This appears to impair real-world applicability.

Table 4
Evaluation results of the content-related scene parameters analysis. The models were trained with exclusively synthetic training data and evaluated with real-world images (real) and their replication in the simulation (sim). The shown metrics correspond to the mean and the standard deviation of 5 models trained with different random seeds. The sim2real gap (gap) is calculated according to Eq. (1). Note that for AP and PCK, a value close to one is desirable.

Experiment	Distractors	rand. orient.	rand. pos.	AP_{sim}	AP_{real}	AP_{gap}	$PCK^{50,3\,px}_{sim}$	$PCK^{50,3\,px}_{real}$	$PCK^{50,3\,px}_{gap}$	$PD^{50,3\,px}_{sim}$	$PD^{50,3\,px}_{real}$
6	×	×	×	0.88 ± 0.014	0.61 ± 0.033	0.30 ± 0.037	0.80 ± 0.033	0.49 ± 0.054	0.39 ± 0.045	0.26 ± 0.01	0.99 ± 0.04
9	×	×	✓	0.86 ± 0.018	0.56 ± 0.040	0.34 ± 0.037	0.89 ± 0.026	0.56 ± 0.037	0.37 ± 0.029	0.25 ± 0.00	0.98 ± 0.02
10	×	✓	×	0.76 ± 0.034	0.28 ± 0.071	0.63 ± 0.091	0.75 ± 0.043	0.32 ± 0.086	0.57 ± 0.136	0.26 ± 0.01	0.98 ± 0.04
11	×	✓	✓	0.68 ± 0.048	0.27 ± 0.080	0.60 ± 0.100	0.76 ± 0.055	0.33 ± 0.079	0.56 ± 0.103	0.26 ± 0.01	1.00 ± 0.01
12	✓	×	×	0.88 ± 0.006	0.55 ± 0.048	0.38 ± 0.051	0.84 ± 0.029	0.42 ± 0.047	0.50 ± 0.062	0.28 ± 0.01	1.01 ± 0.02
13	✓	×	✓	**0.90** ± 0.009	**0.63** ± 0.034	**0.30** ± 0.038	**0.93** ± 0.012	**0.61** ± 0.046	**0.35** ± 0.053	**0.24** ± 0.01	0.98 ± 0.02
14	✓	✓	×	0.84 ± 0.010	0.27 ± 0.061	0.68 ± 0.073	0.85 ± 0.023	0.33 ± 0.054	0.62 ± 0.059	0.24 ± 0.01	0.98 ± 0.05
15	✓	✓	✓	0.82 ± 0.023	0.43 ± 0.030	0.48 ± 0.046	0.86 ± 0.027	0.41 ± 0.058	0.52 ± 0.073	0.24 ± 0.01	**0.97** ± 0.02

Fig. 8. Visual comparison of model predictions from experiment 6 and experiment 10. Object detections are indicated by red bounding boxes and keypoint detections by light green square crosses. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Regarding distractor objects, a comparison of experiments 6 and 12 indicates that simply adding them reduces detection rates for both objects (AP drops from 0.61 to 0.55) and keypoints (PCK drops from 49% to 42%) while at the same time increasing the sim2real gap. When only object positions are randomized (comparing experiments 6 and 9), the impact varies by task: object detection suffers (AP drops from 0.61 to 0.56), whereas keypoints detection improves (PCK increases from 49% to 56%). However, combining distractor objects with random object positions (experiment 13) boosts both tasks, making it the only experiment that outperforms experiment 6 from the previous step. These results contradict the concept of SDR, which proposes that context-based object placement is beneficial for bridging the sim2real gap. Instead, our findings suggest that distractor objects should be used together with randomized placement, provided that it remains within a realistic framework; highly unrealistic scenarios should be avoided.

With this configuration, we achieve an AP of 0.63 on real-world data and correctly detect 61% of mirror corners at a PD of 1 px. The remaining sim2real gap amounts to 30% for object detection and 35% for keypoint detection. Fig. 9 illustrates final predictions of a model from experiment 13 on the real-world test images. Obtaining the predictions for such an image with a resolution of 6000 × 4000 px on a GPU requires 0.44 s.

By examining the predictions in detail, we identify two remaining shortcomings of the model: it fails to detect heliostat very close to the camera and does not recognize heliostats that either fully reflect surfaces other than the sky or have their backside oriented toward the camera. The first issue can likely be addressed by sampling the relevant camera poses. Building on our previous findings, we infer for the second issue that enhancing the realism of the heliostat model may be beneficial. Specifically, modeling distorted mirrors instead of perfectly flat ones is expected to improve detection performance in these scenarios. Overall, we observe that the model rarely produces false positive detections, which is highly advantageous for a subsequent optimization such as photogrammetry. We expect that the achieved detection rate, when considering several camera perspectives, will allow such a measurement to be carried out successfully.

4.2. Ablation study

Training Dataset Size. As discussed earlier, the success of DNNs heavily depends on the availability of a sufficient amount of training data. Generally, increasing the size of a dataset that is well-suited to the specific problem should yield better model performance. We therefore also expect a scaling effect in our investigations. However, this effect may differ across datasets and distort our results. It might be particularly pronounced in highly randomized scenes, which could benefit stronger from an increase in data than less or non-randomized scenes. To rule out this potential bias, we conduct a training dataset size independence study. For this study, we select seven experiments based on Taguchi L4 orthogonality arrays [50] and, for each of them, generate additional training data by adding random camera poses up to a dataset size of 20 000 images. We then sample four separate datasets of different sizes: 2000, 5000, 10 000 (the size used in the main evaluation), and 20 000. Following the procedure in our previous experiments, each model is trained with five different random seeds. The number of epochs is adjusted in proportion to the dataset size to maintain a constant training duration. For evaluation, we focus on the metric AP computed on real-world images as our previous findings indicate that it mostly correlates with both the PCK and the sim2real gap. The results of this study are shown in Fig. 10. We find that increasing the training dataset size has only a minimal impact on performance, remaining within the uncertainties, and that most of the selected experiments exhibit a similar scaling behavior. In experiment 11, we observe a unique decrease in model performance as the dataset size increases. Since both positions and orientations are randomized in this experiment, we attribute this outcome to the growing presence of unrealistic scenarios within the larger dataset, which we previously identified as having a negative impact on model performance. The overall results imply that training dataset size has a negligible influence on our outcomes, indicating that our findings are likely to hold for larger datasets as well. At the same time, the study suggests that a smaller dataset of 2000 images would have been sufficient for our experiments.

Heliostat Geometry. We propose using transfer learning to adapt the developed model for power plants with different heliostat geometries. To demonstrate its applicability, we consider the heliostat field of the Solar Tower Jülich (STJ; owned and operated by the German Aerospace Center), which includes heliostats comprising a 2 × 2 collector without middle gap and therefore a significantly different geometry compared to those considered before. We render an additional 200 images employing this geometry and the optimized scene configuration, and

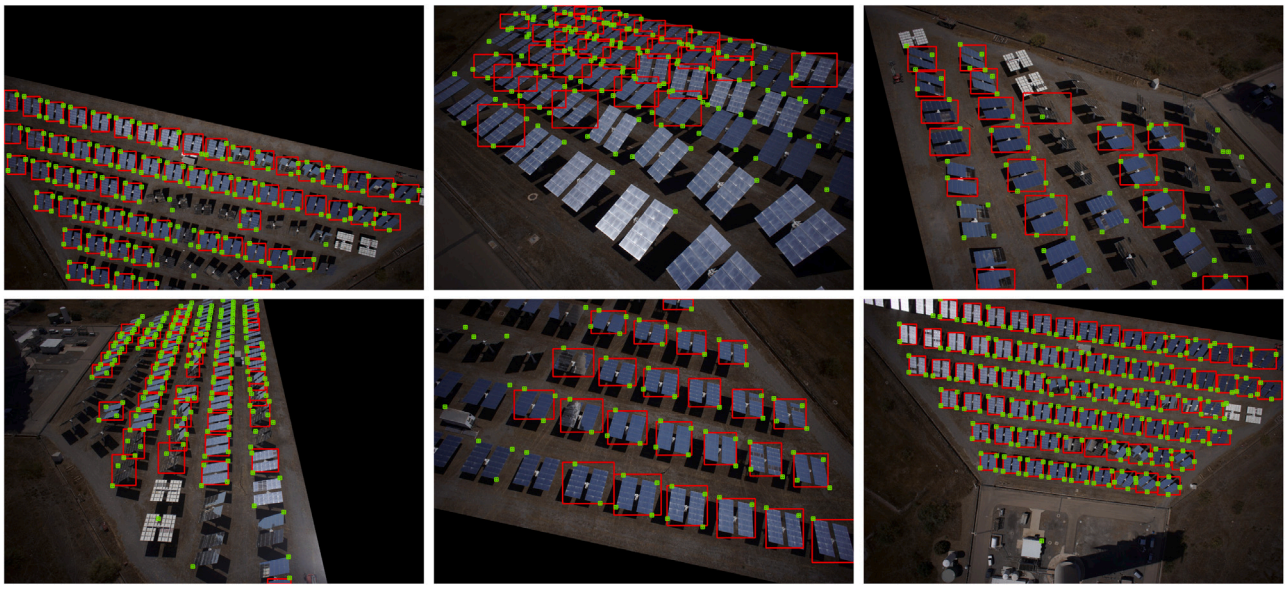


Fig. 9. Predictions of a model from experiment 13 applied on the real-world test images from Jessen et al.'s [4] measurement flight at the PSA (owned and operated by CIEMAT). The model confidence threshold is set to 50%. Object detections are indicated by red bounding boxes and keypoint detections by light green square crosses. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

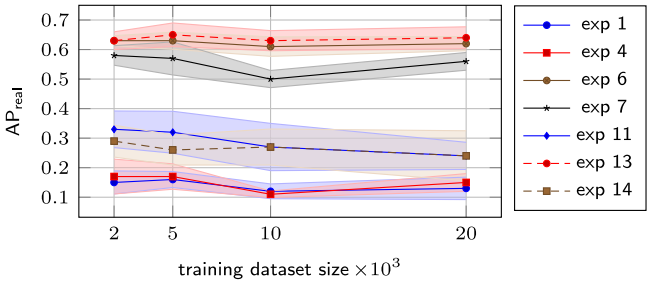


Fig. 10. Results of the training dataset size independence study indicating minor dependence. The diagram shows the mean AP values on the real images of the training runs with different random seeds. The area around the graphs represents the standard deviation.

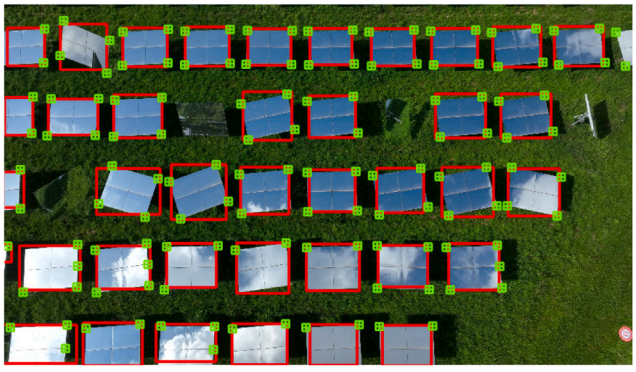


Fig. 11. Predictions of our fine-tuned model on a real-world image from the STJ test plant. Object detections are indicated by red bounding boxes and keypoint detections by light green square crosses. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

fine-tune the existing model for another 200 epochs. The rendering process takes roughly 12 h, while fine-tuning requires about 1 h. We then apply the adapted model on a recently recorded real-world image of the heliostat field at the STJ.

Fig. 11 shows the model predictions for a model confidence threshold of 50%. These qualitative results demonstrate that the model is capable of accurately detecting the heliostats and their outer mirror corners, largely unaffected by external conditions such as a different ground composition or the presence of clouds. This indicates that our approach can be adapted to different heliostat geometries and external factors with little effort. A more detailed, quantitative investigation of this transferability is planned for future work.

5. Conclusion and outlook

In this work, we address the challenge of bridging the sim2real gap to enable the development of DNNs for object and keypoint detection in drone images of heliostat fields using only synthetic data. We introduce an approach for directly measuring the sim2real gap and carry out a detailed analysis of various scene parameters, grouped into the categories *appearance* and *content*.

Our results highlight the crucial role of procedural textures in scene design for reducing the sim2real gap. Additionally, incorporating fine details of the target object – represented here by simulating mirror soiling – proved beneficial. When adding distractor objects to the scene, simultaneous randomization of all object positions emerges as important. On the other hand, randomization that leads to unrealistic scenarios, such as excessive interreflections or lighting conditions beyond the scope of the test case, negatively affects model transferability from simulation to reality in our study. Using the optimized scene configuration, we achieve an AP of 0.63 and detect 61% of all mirror corners on real-world images of a heliostat field, while the remaining sim2real gap amounts to 30% and 35% respectively.

Furthermore, we present a straightforward yet effective method for adapting the developed model to different heliostat geometries, showing promising qualitative results. This highlights the potential of the method to be applied to a wide range of solar tower power plants with

little effort. We recommend to further explore the transferability of the models, particularly considering varying heliostat geometries. Another key direction is extending the approach to further tasks relevant to solar tower power plant operation, such as detecting defective mirror facets or directly estimating heliostat orientations.

Overall our study offers valuable insights for developing and using DNNs for drone-based heliostat field inspection, contributing to the further development of automated monitoring systems for solar tower power plants. Finally, it also provides valuable findings and recommendations for bridging the sim2real gap in synthetic data-driven deep learning.

CRediT authorship contribution statement

Rafal Broda: Formal analysis, Writing – original draft, Conceptualization, Methodology, Visualization, Validation, Writing – review & editing, Data curation, Investigation, Software. **Alexander Schnerring:** Writing – review & editing, Methodology, Conceptualization. **Dominik Schnaus:** Writing – review & editing, Software, Methodology. **Michael Nieslony:** Writing – review & editing, Methodology, Supervision, Conceptualization. **Julian J. Krauth:** Supervision, Conceptualization, Writing – review & editing, Methodology. **Marc Röger:** Project administration, Funding acquisition, Writing – review & editing, Supervision, Resources, Methodology, Conceptualization. **Sonja Kallio:** Supervision, Writing – review & editing. **Rudolph Triebel:** Supervision, Conceptualization, Funding acquisition. **Robert Pitz-Paal:** Funding acquisition, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded by the Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection through the AuSeSol-AI project (grant agreement no. 67KI21007A), based on a decision by the German Bundestag. Additionally, we gratefully acknowledge the computational and data resources provided through the joint high-performance data analytics (HPDA) project “terrabyte” of the German Aerospace Center (DLR) and the Leibniz Supercomputing Center (LRZ). Lastly, we extend our thanks for the use of image data from the Plataforma Solar de Almería (owned and operated by CIEMAT) and the Solar Tower Jülich (owned and operated by the German Aerospace Center).

Appendix A. Distractor objects

See Fig. A.1.

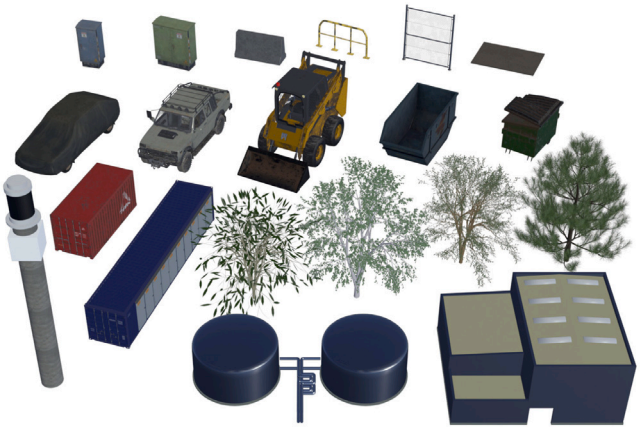


Fig. A.1. Overview of all the distractor objects used in this study. Note that the objects have been resized for easier visualization and are therefore not shown true-to-scale.

Appendix B. Experiments comparison

See Fig. B.1.

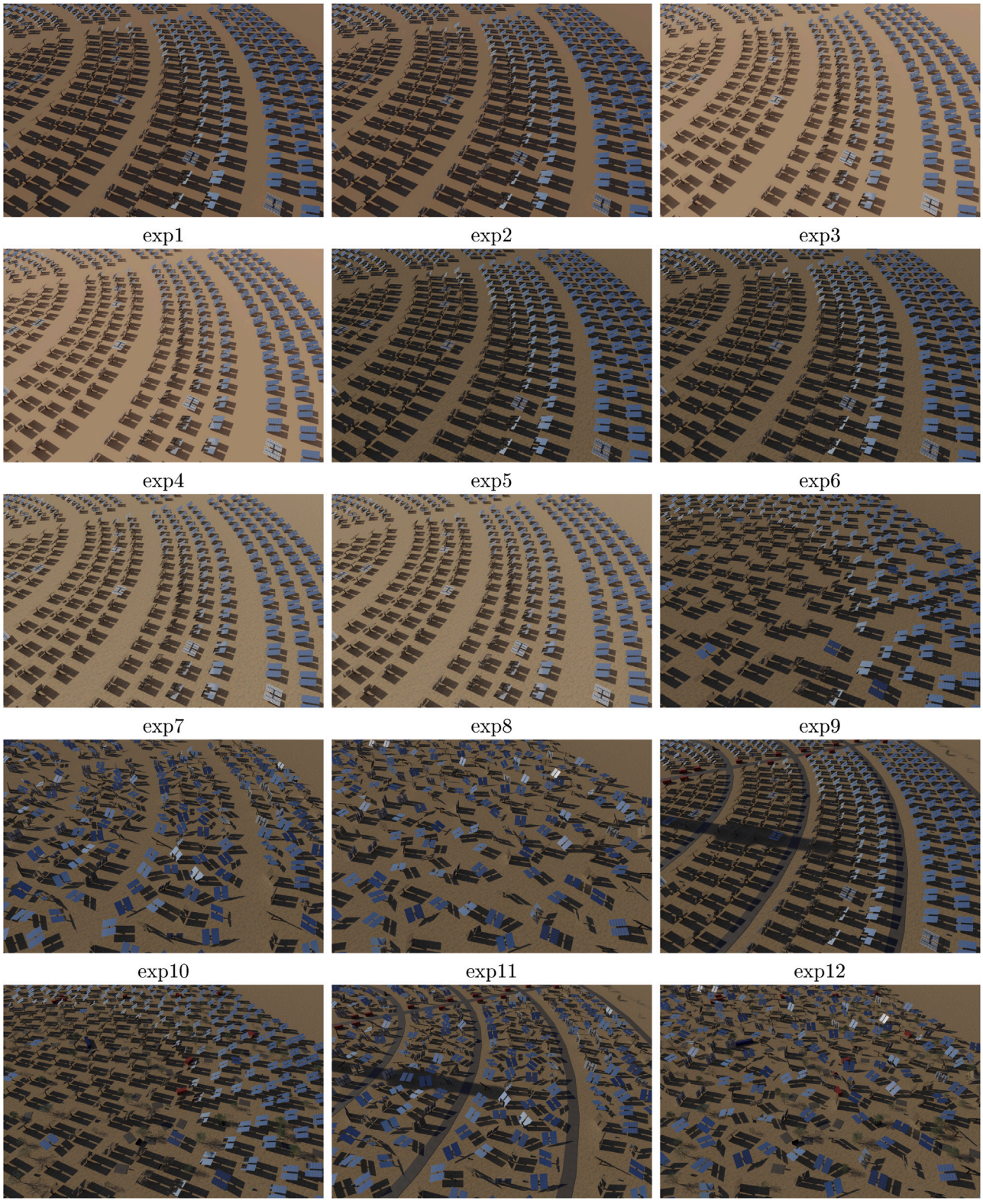


Fig. B.1. Side-by-side comparison of a randomly selected image generated with the same camera pose from each experiment.

Data availability

The data that support the findings of this study are available from the corresponding author upon request.

References

[1] J.C. Sattler, M. Röger, P. Schwarzbözl, R. Buck, A. Macke, C. Raeder, J. Göttsche, Review of heliostat calibration and tracking control methods, Sol. Energy 207 (2020) 110–132.

[2] R.A. Mitchell, G. Zhu, A non-intrusive optical (NIO) approach to characterize heliostats in utility-scale power tower plants: Methodology and in-situ validation, Sol. Energy 209 (2020) 431–445.

[3] J. Yellowhair, P.A. Apostolopoulos, D.E. Small, D. Novick, M. Mann, Development of an aerial imaging system for heliostat canting assessments, AIP Conf. Proc. 2445 (1) (2022) 120024.

[4] W. Jessen, M. Röger, C. Prahl, R. Pitz-Paal, A two-stage method for measuring the heliostat offset, AIP Conf. Proc. 2445 (1) (2022) 070005.

[5] J.J. Krauth, C. Happich, N. Algner, R. Broda, A. Kämpgen, A. Schnerring, S. Ulmer, M. Röger, HelioPoint – a fast airborne calibration method for heliostat fields, J. Sol. Energy Eng. 146 (6) (2024) 061005.

[6] C. Prahl, Photogrammetric measurement of the optical performance of parabolic trough solar fields (Ph.D. thesis), RWTH Aachen, 2019.

[7] M. Röger, C. Prahl, S. Ulmer, Heliostat shape and orientation by edge detection, J. Sol. Energy Eng. 132 (2) (2010) 021002.

[8] AuSeSol-AI, grant agreement no. 67KI21007A, start: 2022-07-04, end: 2025-07-03.

[9] G. Zhu, C. Augustine, R. Mitchell, et al., HelioCon: A roadmap for advanced heliostat technologies for concentrating solar power, Sol. Energy 264 (2023) 111917.

[10] J.A. Carballo, J. Bonilla, M. Berenguel, J. Fernández-Reche, G. García, New approach for solar tracking systems based on computer vision, low cost hardware and deep learning, Renew. Energy 133 (2019) 1158–1166.

[11] D. Kesseli, V. Chidurala, R. Gooch, G. Zhu, A combined computer vision and deep learning approach for rapid drone-based optical characterization of parabolic troughs, J. Sol. Energy Eng. 145 (2) (2022) 021008.

[12] B. Liu, A. Sonn, A. Roy, B. Brewington, Deep learning method for heliostat instance segmentation, SolarPACES Conf. Proc. 1 (2024).

[13] F. Xu, C. Li, F. Sun, On-line measurement of tracking poses of heliostats in concentrated solar power plants, Sensors 24 (19) (2024).

[14] R. Broda, A. Schnerring, J.J. Krauth, M. Röger, R. Pitz-Paal, Towards deep learning based airborne monitoring methods for heliostats in solar tower power plants, in: Advances in Solar Energy: Heliostat Systems Design, Implementation, and Operation, Vol. 12671, SPIE, 2023, 1267108.

[15] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, S. Birchfield, Deep object pose estimation for semantic robotic grasping of household objects, in: Conference on Robot Learning (CoRL), 2018.

[16] S. Moonen, B. Vanherle, J. de Hoog, T. Bourgana, A. Bey-Temsamani, N. Michiels, CAD2Render: A modular toolkit for GPU-accelerated photorealistic synthetic data generation for the manufacturing industry, in: IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, WACVW, 2023, pp. 583–592.

[17] P. Rawal, M. Sompura, W. Hintze, Synthetic data generation for bridging Sim2Real gap in a production environment, 2023, arXiv preprint arXiv:2311.11039.

[18] J. Tremblay, A. Prakash, D. Acuna, et al., Training deep networks with synthetic data: Bridging the reality gap by domain randomization, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2018, pp. 1082–10828.

[19] A. Kar, A. Prakash, M.-Y. Liu, et al., Meta-Sim: Learning to generate synthetic datasets, in: IEEE/CVF International Conference on Computer Vision, ICCV, 2019, pp. 4550–4559.

[20] F. Reway, A. Hoffmann, D. Wachtel, W. Huber, A. Knoll, E. Ribeiro, Test method for measuring the simulation-to-reality gap of camera-based object detection algorithms for autonomous driving, in: IEEE Intelligent Vehicles Symposium, IV, 2020, pp. 1249–1256.

[21] S. Hinterstoisser, V. Lepetit, P. Wohlhart, K. Konolige, On pre-trained image features and synthetic images for deep learning, in: Computer Vision – ECCV 2018 Workshops, Springer International Publishing, 2019, pp. 682–697.

[22] M. Rudorfer, L. Neumann, J. Krüger, Towards learning 3d object detection and 6d pose estimation from synthetic data, in: 24th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2019, pp. 1540–1543.

[23] Epic Games, Unreal Engine https://www.unrealengine.com.

[24] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, T. Funkhouser, Physically-based rendering for indoor scene understanding using convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 5057–5065.

[25] Z. Li, N. Snavely, CGIntrinsics: Better intrinsic image decomposition through physically-based rendering, in: Computer Vision - ECCV 2018: 15th European Conference, Proceedings, Part III, Springer-Verlag, 2018, pp. 381–399.

[26] C. Mayershofer, T. Ge, J. Fottner, Towards fully-synthetic training for industrial applications, in: 10th International Conference on Logistics, Informatics and Service Sciences, LISS, 2020.

[27] Blender Online Community, Blender - a 3D modelling and rendering package http://www.blender.org.

[28] N. Morrical, J. Tremblay, Y. Lin, S. Tyree, S. Birchfield, V. Pascucci, I. Wald, NViSII: A scriptable tool for photorealistic image generation, in: ICLR Workshop on Synthetic Data Generation, 2021, arXiv:2105.13962.

[29] K. Greff, F. Belletti, L. Beyer, et al., Kubric: A scalable dataset generator, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 3739–3751.

[30] M. Denninger, D. Winkelbauer, M. Sundermeyer, W. Boerdijk, M. Knauer, K.H. Strobl, M. Humt, R. Triebel, BlenderProc2: A procedural pipeline for photorealistic rendering, J. Open Source Softw. 8 (82) (2023) 4901.

[31] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, D. Krishnan, Unsupervised pixel-level domain adaptation with generative adversarial networks, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 95–104.

[32] S.R. Richter, H.A. Alhaija, V. Koltun, Enhancing photorealism enhancement, IEEE Trans. Pattern Anal. Mach. Intell. 45 (2) (2023) 1700–1715.

[33] S. Hinterstoisser, O. Pauly, H. Heibel, M. Martina, M. Bokeloh, An annotation saved is an annotation earned: Using fully synthetic training for object detection, in: IEEE/CVF International Conference on Computer Vision Workshop, ICCVW, 2019, pp. 2787–2796.

[34] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, Domain randomization for transferring deep neural networks from simulation to the real world, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE Press, 2017, pp. 23–30.

[35] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, S. Birchfield, Structured domain randomization: Bridging the reality gap by context-aware synthetic data, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE Press, 2019, pp. 7249–7255.

[36] L. Eversberg, J. Lambrecht, Generating images with physics-based rendering for an industrial object detection task: Realism versus domain randomization, Sensors 21 (23) (2021).

[37] D. Horváth, G. Erdős, Z. Istenes, T. Horváth, S. Földi, Object detection using Sim2Real domain randomization for robotic applications, Trans. Rob. 39 (2) (2023) 1225–1243.

[38] Poly Haven, 2025, (Last accessed 27 January 2025) https://polyhaven.com/.

[39] X. Zhou, D. Wang, P. Krähenbühl, Objects as points, 2019, arXiv preprint arXiv:1904.07850.

[40] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2980–2988.

[41] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 779–788.

[42] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017, arXiv preprint arXiv:1704.04861.

[43] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, ICLR, 2021.

[44] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) (2015) 770–778.

[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[46] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, 2015, pp. 234–241.

[47] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, ICLR, 2015.

[48] L.N. Smith, N. Topin, Super-convergence: very fast training of neural networks using large learning rates, in: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, 11006, SPIE, 2019, 1100612.

[49] K. Wada, labelme: Image polygonal annotation with python https://github.com/wkentaro/labelme.

[50] G. Taguchi, S. Konishi, A.S. Institute, Orthogonal Arrays and Linear Graphs: Tools for Quality Engineering, American Supplier Institute, 1987.