Data Validation Experiments with a Computer-Generated Imagery Dataset for International Nuclear Safeguards

Zoe N. Gastelum [a], **Timothy M. Shead** [b], **Matthew Marshall** [c]

[a] Sandia National Laboratories, International Safeguards & Engagements Department, 1515 Eubank SE, Albuquerque, NM, USA, 87123
[b] Sandia National Laboratories, Machine Intelligence and Visualization Department, 1515 Eubank SE, Albuquerque, NM, USA, 87123
[c] Sandia National Laboratories, Nuclear Verification Department, 1515 Eubank SE, Albuquerque, NM, USA, 87123

Abstract:

Computer vision models have great potential as tools for international nuclear safeguards verification activities, but off-the-shelf models require fine-tuning through transfer learning to detect relevant objects. Because open-source examples of safeguards-relevant objects are rare, and to evaluate the potential of synthetic training data for computer vision, we present the Limbo dataset. Limbo includes both real and computer-generated images of uranium hexafluoride containers for training computer vision models. We generated these images iteratively based on results from data validation experiments that are detailed here. The findings from these experiments are applicable both for the safeguards community and the broader community of computer vision research using synthetic data.

Keywords: Computer vision, synthetic data, international nuclear safeguards, uranium hexafluoride.

1. Introduction

The International Atomic Energy Agency (IAEA) operates under the United Nations and is responsible for verifying that nuclear materials and facilities across the globe are limited to peaceful use. They do so by implementing and monitoring international nuclear safeguards: measures to account for nuclear materials and verify the design and operation of nuclear facilities. Increasing interest in nuclear energy technologies, growing inventories of nuclear material, and limited IAEA safeguards resources are compelling the IAEA to be more efficient in safeguards monitoring.

Computer vision models could increase IAEA safeguards efficiency, by augmenting visual tasks conducted as part of the IAEA's safeguards mission. Examples of visual tasks for which computer vision research and development is currently underway throughout the safeguards community include:

- Object and change detection for nuclear-relevant sites via satellite imagery analysis (Rutkowski, Canty, & Nielsen, 2018).
- Collection, triage, and information recall for open-source images (Feldman, Arno, Carrano, Ng, & Chen, 2018) (Gastelum & Shead, 2018) (Arno, 2018).
- Reviewing surveillance camera data for specific objects or patterns of life (Smith, Hamel, Hannasch, Thomas, & Gaiten-Cardenas, 2021) (Thomas, et al., 2021) (Wolfart, Casado Coscolla, & Sequeira, 2022).
- Supporting inspector indoor localization at complex nuclear facilities (Wolfart, Sanchez-Belenguer, & Sequeira, Deep Learning for Nuclear Safeguards, 2021).
- Supporting inspectors with digital assistants for visual tasks (Smartt, Gastelum, Rutkowski, Peter-Stein, & Shoman, 2021).

Despite this surge in research, access to sufficiently large, relevant datasets remains a challenge. Relevant data for international safeguards research and development are rare for multiple reasons. First, real international safeguards data are sensitive and held in confidence by the IAEA and are therefore inaccessible for most research. Second, safeguards-relevant data may be either commercially sensitive or have national security sensitivities for states. Third, relevant data may be lost to history due to obsolete file formats, data corruption, or lack of digitization. Finally, relevant data might not exist; for example, images of technologies that are physically feasible but not widely adopted may be of interest to detect future proliferation activities, but images of these technologies are non-existent.

In response to the rarity of available safeguards data, we have created a large, open-source, safeguards-relevant imagery dataset called Limbo. Limbo contains one million synthetic (computer-generated) images intended for computer vision research and development. The images include detailed, automatically-generated segmentation mask, contour, and bounding box annotations, see Figures 8 – 10 for examples. We also provide a small collection of annotated real-world images for validation that include well-documented copyright information to simplify publication. Our goal for the Limbo data, and for synthetic data more broadly, is to develop computer vision models trained solely on synthetic data that can achieve state-of-the-art performance when evaluated on real-world data.

We applied several criteria in selecting a subject matter for our synthetic data. We wanted the subject to be:

- Unclassified, for easier development and dissemination of the data.
- Visually distinct, to facilitate labeling of real-world validation data.
- Relatively common, to ensure that we would have sufficient real-world data to support our validation activities.
- Prevalent within the nuclear fuel cycle, so the generated data could support the broadest possible research and development, without being tied to a single process or type of facility.[1]

Based on these criteria, we opted to generate images of containers used to store and transport uranium hexafluoride (UF6) throughout the commercial nuclear fuel cycle. We specifically focused on two general models of UF6 containers: 30B and 48-type containers.

30B containers are 30-inch cylinders used to transport uranium-235 enriched up to 5%. These containers are primarily found at uranium enrichment facilities (as the product output) and fuel fabrication plants (as the product input). See Figure 1 for a real-world example.

48-type containers refer to a class of 48-inch containers used to store and transport natural and depleted UF6. We included three common designs of 48s: 48X and 48Y containers are used for storage and transportation, while 48G containers are characterized by the lack of an apron and are used exclusively for storage. 48-type containers can be found at uranium conversion plants (as the product output), uranium enrichment plants (as the input, and to store depleted tails), and fuel fabrication facilities (as input for natural uranium fuel). See Figure 2 for a real-world example of 48Y containers.

In addition to relevant containers, the Limbo data includes examples (both real and synthetic) of distractor objects including propane tanks, gas canisters, beer kegs, 55-gallon drums, and more. Synthetic distractors have the full metadata suite, while real-world distractor metadata includes only the class “distractor”.

In the remainder of this paper, we describe the data generation process (Section 2), validation workflow (Section 3), data validation experiments and results (Section 4), and discussion and implications for future research (Section 5). We also provide information on how to access and use the Limbo data, and descriptions of the Limbo dataset contents (Section 6).

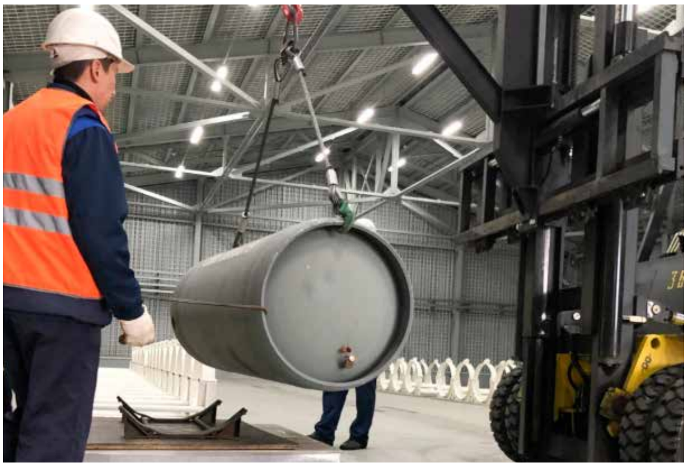


Figure 1: 30B uranium hexafluoride container at the IAEA Low Enriched Uranium Bank in Kazakhstan. Credit: IAEA, 2019.

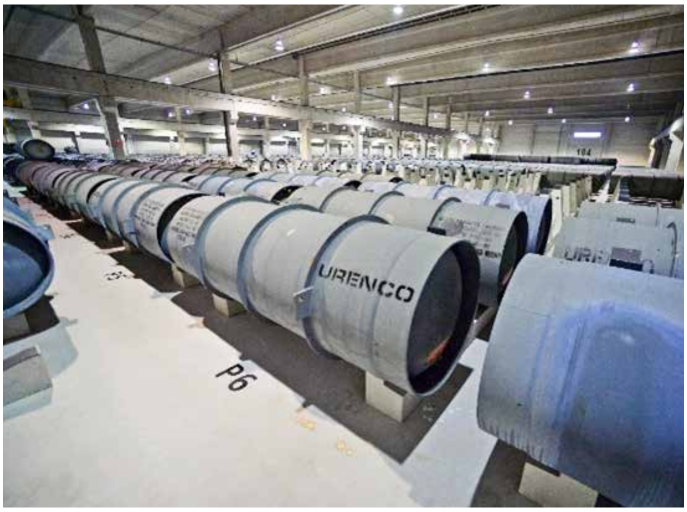


Figure 2: 48Y containers at Urenco, Netherlands. Credit: IAEA, 2015.

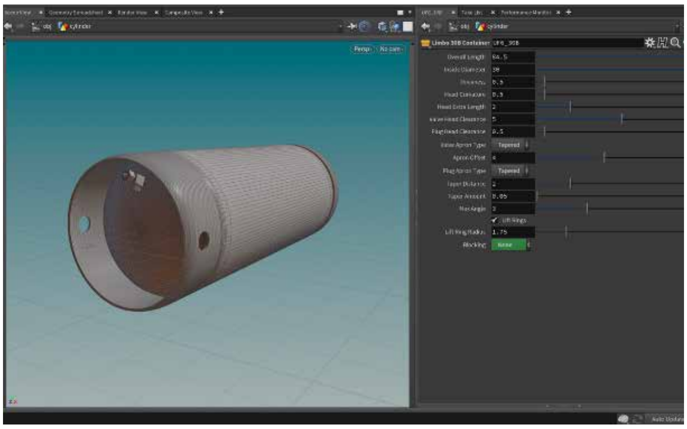
2. Data Generation

In this section, we describe our workflow to generate synthetic images. This process includes the creation of three-dimensional (3D) models of UF6 containers, random sampling of 3D model parameters, and placement in real or virtual environments, followed by rendering to produce 2D images and metadata.

2.1 Developing 3D Models

We developed 3D models of our UF6 containers using SideFX Houdini (https://www.sidefx.com/products/houdini/), a procedural 3D modeling and animation tool widely used in films, television, and game design. A screenshot of the Houdini workspace with a parameterized 30B container model is provided in Figure 3. The 3D models were informed by technical standards and specifications published in open sources by industry partners and professional societies, with some subjective adjustments to better match the containers in real-world images. Sources that were especially useful for our model development included:

[1] Through a collaboration with researchers at Lawrence Livermore National Laboratory, we had access to a set of images collected from open sources that provided an indication of overall prevalence in open sources and served as a seed for additional data collection.



Limbo 30B Container UF6_30B
Overall Length 64.5
Inside Diameter 30
Thickness 0.5
Head Curvature 0.5
Head Extra Length 2
Valve Head Clearance 5
Plug Head Clearance 0.5
Valve Apron Type Tapered
Apron Offset 4
Plug Apron Type Tapered
Taper Distance 2
Taper Amount 0.05
Max Angle 3
✓ Lift Rings
Lift Ring Radius 1.75
Blocking None

Figure 3: 3D CAD model of a 30B UF6 container in the Houdini software.

- Uranium Hexafluoride: A Manual of Good Handling Practices (United States Enrichment Corporation, 1995);
- American National Standard for Nuclear Materials - Uranium Hexafluoride – Packaging for Transport (American Nuclear Standards Institute, 2001); and
- Uranium Hexafluoride: Handling Procedures and Container Descriptions (Oak Ridge Operations, 1987).

Once the cylinder models were created in Houdini, we used the Allegorithmic (now Adobe) Substance 3D paint software to generate multiple sets of "paint job" textures for the cylinders in varying styles and levels of wear.

In addition to the cylinders, Limbo also includes a variety of distractor objects. Unlike the UF6 containers, the distractors are common objects not specific to the nuclear fuel cycle (such as propane tanks, welding gas cylinders, wine barrels, etc.) that are widely available commercially. Therefore, for the distractor objects we procured 3D models from an online 3D model marketplace (https://turbosquid.com) with appropriate permissions for use and distribution.

2.2 Model Placement and Environment

As backgrounds for our 3D container models, we provided two major classes of environment: real-world and synthetic.

Real-world environments were created using panoramic High Dynamic Range (HDR) photographs, which capture a 360-degree image of a scene, and—unlike normal photographs—use special techniques to record the full range of light intensity for each pixel. In this way, an HDR image samples light intensity from all directions simultaneously. This makes it possible for an HDR image to provide the photographic backdrop for a scene while also supplying realistic, nuanced lighting. We used HDR images collected from open sources with appropriate permissions, including indoor and outdoor scenes. A majority of the images were industrial scenes similar to environments where real 30B and 48 containers would be found, but we also included several studio and other scenes for variety. Several examples of our HDR backgrounds are shown in Figure 4 (the images are warped by the panoramic perspective but display normally when projected into the final 2D rendered images).

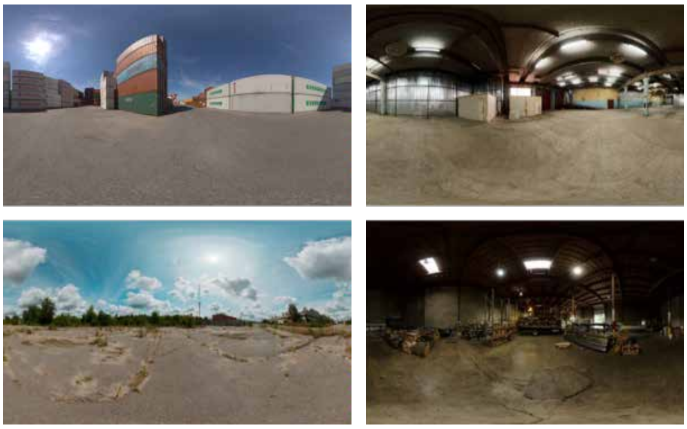


Figure 4: Sample indoor and outdoor HDR environments.

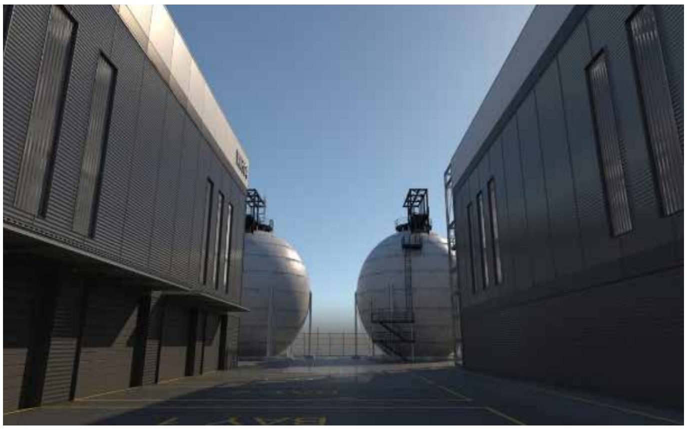


Figure 5: Synthetic 3D oil refinery environment.

A limitation of using real images as backgrounds is that the scale and perspective of the background may not match the 3D objects in the foreground. This can lead to cylinders that seem unusually large or small, relative to their surroundings, or appear to be floating in air instead of properly grounded. Although ultra-realistic synthetic images are not necessarily required for robust model training (Tremblay, et al., 2018), we addressed this by providing a fully synthetic 3D environment in later Limbo images, based on an outdoor scene of an oil refinery. The 3D oil refinery provided a large and diverse setting for our containers, had industrial features similar to a nuclear fuel cycle facility, and guaranteed that 3D foreground objects perfectly matched the background in proportions and perspective.

We inserted skies from the real-world HDR images into the synthetic oil refinery for additional control and manipulation of lighting. A scene from the refinery is shown in Figure 5.

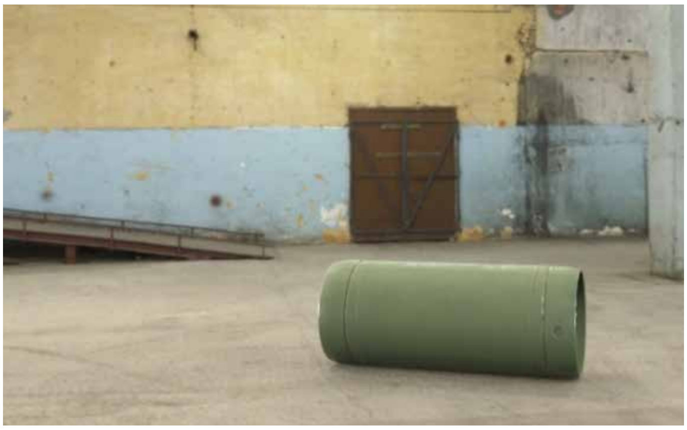


Figure 6: Examples of synthetic 30B containers in a variety of real-world HDR environments.

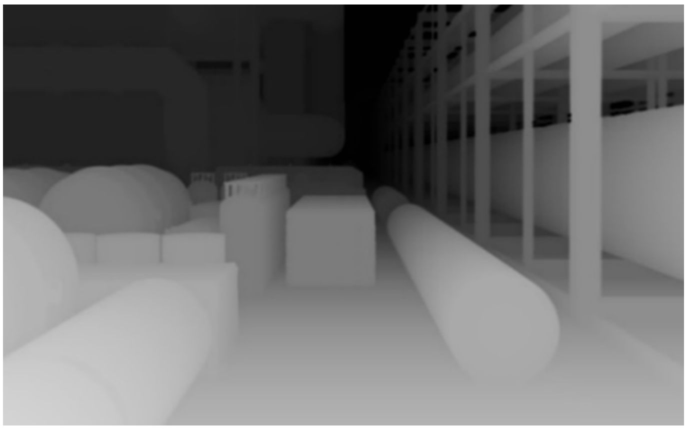


Figure 7: Depth map image suitable for use as LIDAR ground truth.

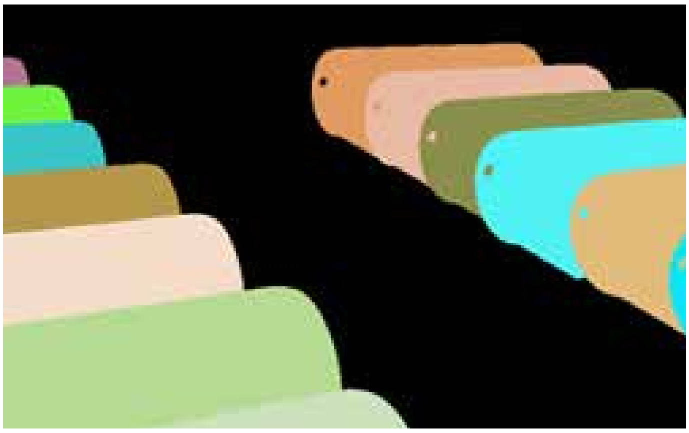


Figure 8: Segmentation masks derived from per-instance occupancy data.

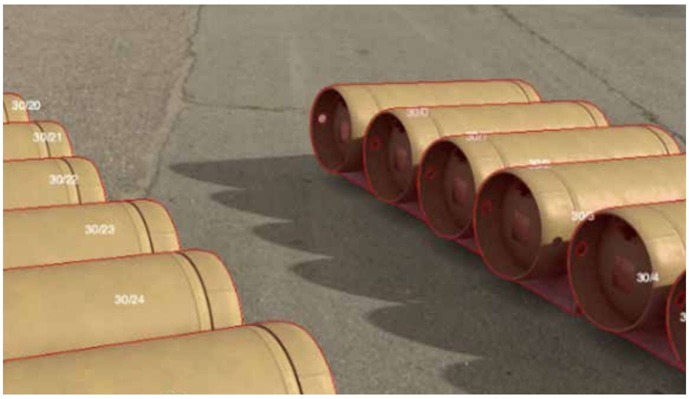


Figure 9: Object contours for scene objects.

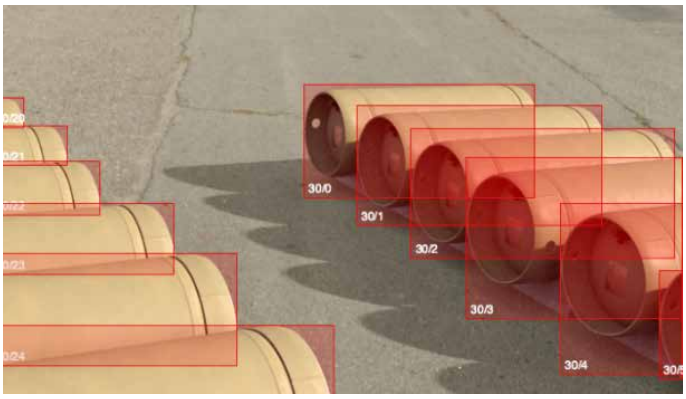


Figure 10: Bounding boxes derived from per-instance masks.

The refinery scene was procured from the same online 3D model marketplace as our distractor objects.

For both HDR and synthetic 3D environments, we used Houdini to assemble complete scenes via random sampling of parameters such as the number, type, organization (scattered versus rows) and placement of containers within the environment; camera location, orientation, and lens; environmental lighting conditions; container material appearance; container condition (new, scratched, rusty, etc.); and type of cradles (wood, concrete) used to support the containers.

2.3 Rendering and Metadata

Once the individual 3D scenes were assembled, we used Redshift 3D - a GPU-accelerated, biased render engine implementing a physically based rendering (PBR) lighting model – to render 2D images. Importantly, each of our 2D images comprises several layers and multiple files created explicitly with the needs of computer vison research in mind. Each of our images includes the following:

1. A 720 by 720-pixel HDR visible spectrum image (Figure 6).

2. A corresponding depth map image, where the value of each pixel is its distance from the camera. This data can be used by researchers interested in training models on light detection and ranging (LiDAR) information (Figure 7).

3. Sub-pixel occupancy data for every object in the image (Figure 8). Storing the per-pixel areas occupied by multiple objects allows us to generate a variety of perfect ground truth information, including per-instance and per-class segmentation masks, contours (Figure 9), bounding boxes (Figure 10), and tags for image classification. Sub-pixel occupancy data is stored in compressed form using the efficient and elegant Cryptomatte file format (Friedman & Jones, 2021).

4. Metadata including the contents of a scene, background, lighting, and camera parameters.

The images were rendered as a series of thematic "campaigns", which are used to describe the image sets in experiments in Section 4, and in describing the data in Section 6. Importantly, the synthetic images are perfectly labelled because the labels themselves are generated at the same time as the images, using the same 3D scene information.

In addition to the data proper, we also developed an application programming interface (API) to simplify accessing the full data and metadata for each image. Information on the API is available at: https://limbo-ml.readthedocs.io/.

3. Data validation procedure

We developed a data validation workflow to ensure that computer vision models could be trained using our synthetic images. In this section, we describe the data validation workflow, the findings from our validation activities, and how they informed later iterations of the Limbo data. This was a crucial step in the data generation process since the Limbo data is intended for computer vision research and development. Our workflow was iterative, including four steps: rendering synthetic images, training models on synthetic data, testing models on real data, and interpreting what the models learned. Then, we incorporated those lessons when rendering new synthetic data. The workflow is depicted in Figure 11, and each step will be described in additional detail below.

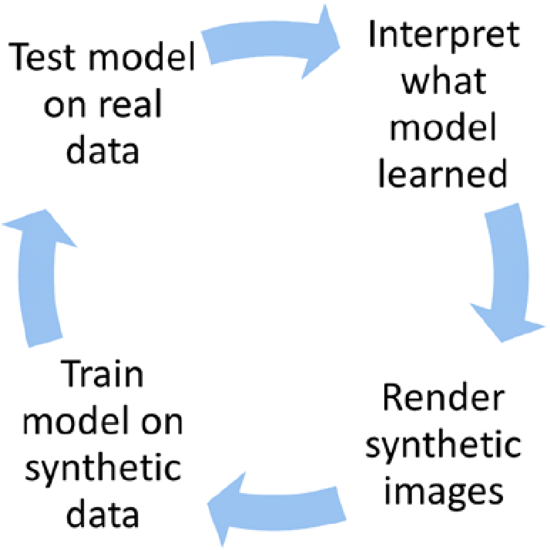


Figure 11: Synthetic data workflow.

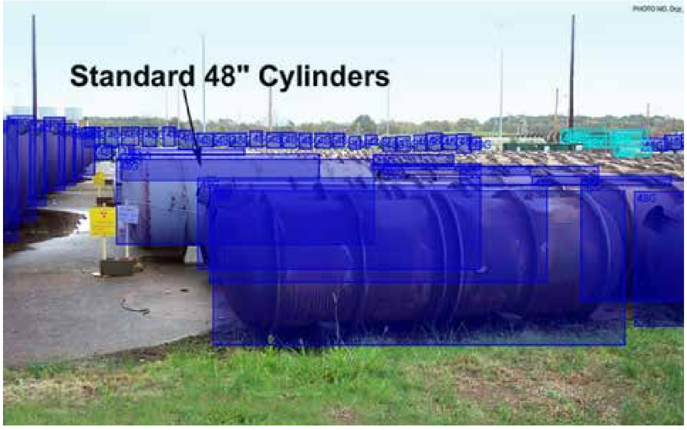


Figure 12: Sample Limbo reference image labelled by our research team (right).

3.1 Training Models on Synthetic Data

Our primary goal was to validate that popular computer vision models could learn from our synthetic images. We selected two main types of computer vision models for validation: image classifiers and object detectors. For each model type, we fine-tuned multiple pre-trained architectures. For image classification we used ResNet-50 (He, Xiangyu, Shaoqing, & Sun, 2016) and Inception (Szegedy, et al., 2015). For object detection, we used YOLO-v5s (Jocher, et al., 2022) (which is built upon Yolov3 (Redmon & Farhadi, 2018)), SSS (Liu, et al., 2016) and Faster R-CNN (Ren, He, Girshick, & Sun, 2015).

We conducted a series of experiments that trained the models using subsets of the synthetic Limbo data, to validate that the models could learn from the data and to identify any issues with the data. The results of those experiments are discussed in Sections 4.2 and 4.3.

3.2 Testing Models on Real Data

After training models using synthetic data, each model was tested on the curated collection of real-world data—which we refer to as reference data—that is included with Limbo. The reference data contains images of both types of relevant UF6 containers and numerous distractors. Each real-world image in the reference dataset is accompanied by metadata that includes copyright information and ground truth bounding boxes manually labelled by members of our project team (Figure 12). As one can see in Figure 12, the manually drawn bounding boxes are not as perfect as those generated automatically for our synthetic data. However, we followed a consistent protocol for bounding box labeling, which was subject to inter-rater quality checks within our team. We think this protocol resulted in higher quality labels than many of the open-sourced labels used in the large benchmark datasets, which have documented errors and quality issues (Northcutt, Athalye, & Mueller, 2021).

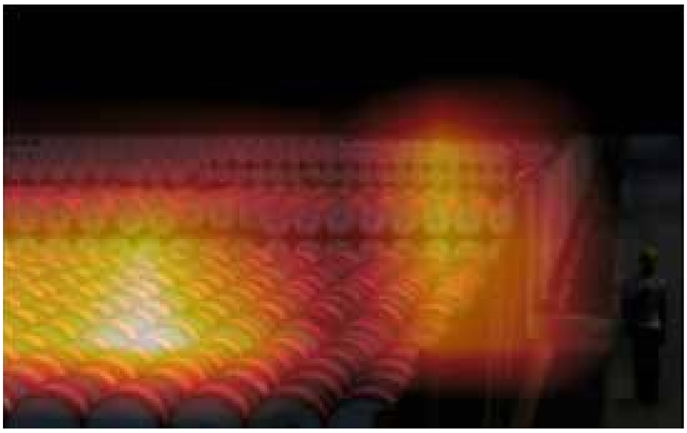


Figure 13: Example explanation from a false positive image classification, using GradCAM to visualize salient pixels.

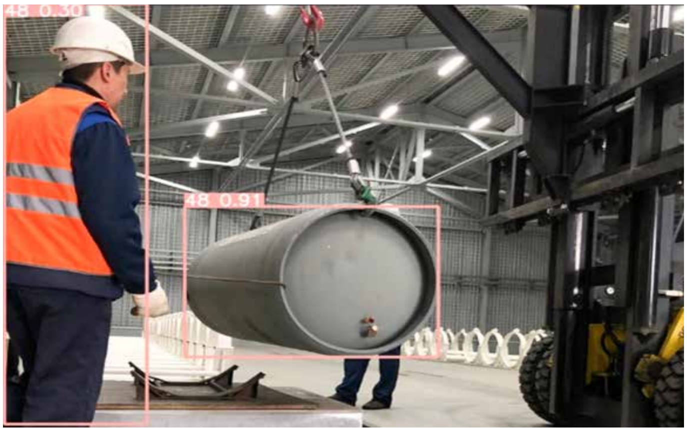


Figure 14: Example false positive detection results, in which the YOLO-v5s model identified a human and a 30B container as 48-type containers.

3.3 Interpreting What the Models Learned

We interpreted the results of our models on reference data to identify potential issues with how the computer vision models were learning from the Limbo data. The mechanisms we used to interpret model learning differed by model type.

For the image classification models, we used machine learning explainability methods to visualize the pixels of an image that were most influential in each prediction. We reviewed the false positive and true positive predictions to interpret the features that were informing positive classification results.

Due to the variation in responses from machine learning explainability techniques, we simultaneously viewed the explanations from three explainability models: GradCAM (Selvaraju, et al., 2017), Guided GradCAM, and Gradient SHAP. An example from an early classification model's false positive explanation is in Figure 13. From these explanations, we interpreted what features of the synthetic Limbo data were more relevant during model training and inference on the real-world data. Figure 13 shows an early example of indications that we needed to add distractor cylindrical objects into the dataset. Additional details of the classification model validation results are below in Section 4.2.

For the object detection models, we opted to use the placement of the bounding boxes to interpret the most relevant areas of an image used to make an inference. For example, in Figure 14 the model incorrectly detected 48-type containers around a human and a 30B container. Similar to how we interpreted the image classification results, we reviewed the object detection true positives and false positive detections and anecdotally devised potential implications of our Limbo data based on what we observed the object detection models were learning. The example in Figure 14 is one of dozens of false positive detections that prompted us to integrate synthetic people into our Limbo data.

We also evaluated images of false negative classifications and detections. We attempted to observe common features of the images that may have impacted the failure to correctly classify or detect the object of interest in the images. The process we used for reviewing misclassifications and what we learned from the process is described in (Gastelum, Shead, & Marshall, 2022).

3.4 Rendering New Synthetic Data

From our analysis of image classification and object detection results, we made multiple additions to our Limbo dataset, including the addition of cylindrical distractor objects and arranging containers into rows. After making updates to the Limbo data, we re-trained and re-tested our models. Selected results and findings from those activities, including experimentation with subsets of the Limbo data, are described in the following sections.

4. Data validation experiments

The purpose of the data validation experiments was to confirm that computer vision models could be successfully trained on our synthetic data and tested on real data. Though we imagined that researchers or model developers could have access to a small amount of real data, we intended to prepare this data under the assumption that it would not necessarily be augmented by real data. Prior research on the use of synthetic data for training models typically includes large quantities of real-world data such as (Ekbatani, Pujol, & Segui, 2017) (Gaidon, Wang, Cabon, & Vig, 2016) and (Movshovitz-Attias, Kanade, & Sheikh, 2016), and achieves good model performance. However, the sizes of the real datasets in these papers (tens of thousands of real images) are still beyond the reach of our intended application spaces. We have previously examined the impact of augmenting synthetic data with small numbers of real images, with resulting model performance being approximately the same (Gastelum & Shead, 2020). Therefore, our validation experiments focused on training models exclusively with synthetic data and testing them with real data.

Descriptions of our validation experiments in which we train models on synthetic data and test them on real data are detailed in Sections 4.2 and 4.3. These experiments utilize image classification models and object detection models. While there are other relevant computer vision model types available such as image segmentation models, we think that these two types provide sufficient evidence for our validation tests. Additional experiments with segmentation masks or other model types could prove to be interesting future research.

Model performance in these experiments was measured in two ways. First, for image classification models, accuracy measures are dependent on the class ratios present in the test data, so we evaluated model performance using two common computer vision performance metrics: precision and recall. Precision is the percentage of items predicted to be members of a class that actually are members of that class (true positives divided by the sum of true positives and false positives, while recall is the percentage of class members that are predicted to be members of that class (true positives divided by the sum of true positives and false negatives).

Second, for object detection models, we used a hybrid scoring approach. We first evaluated object detection models with the industry standard measure of performance mean Average Precision (mAP), which considers model performance on multiple object types, evaluation of positive and negative identifications, and evaluation of the predicted bounding box compared to the ground truth bounding box (for a useful tutorial, see (Tan, 2019)). For our evaluations, we set the intersection over union (IOU) threshold of 0.25. We used this lower-than-typical IOU standard based on our deployment assumption that the detection of a relevant object, even with an imperfectly aligned bounding-box, could still support analysts in finding indications of nuclear activity.

It is important to note that it was not the intent of these experiments to spend significant resources in fine-tuning hyper-parameters for best model performance. Rather, we used these validation experiments to suggest improvements for our synthetic data and to obtain a rough estimate of model performance when using it for training.

4.1 Confirmation of Model Implementation

Although our focus for eventual deployment is on the train-synthetic, test-real use-case described above, all of our experiments are tested on synthetic data during training too - this allows us to validate that the code is working properly and the models are successfully training. As one extant example, the following figures show train-synthetic, test-synthetic results for one set of experiments where we trained ten ResNet-50 models for 500 epochs using 5000 synthetic images of type 48 containers from campaign 17, and tested using 1000 additional synthetic images from the same campaign. As can be seen, we achieve excellent precision (>87%) (see Figure 15) and recall (>85%) (see Figure 16) on the type-48 identification task (metrics are the results averaged from evaluating all ten models).

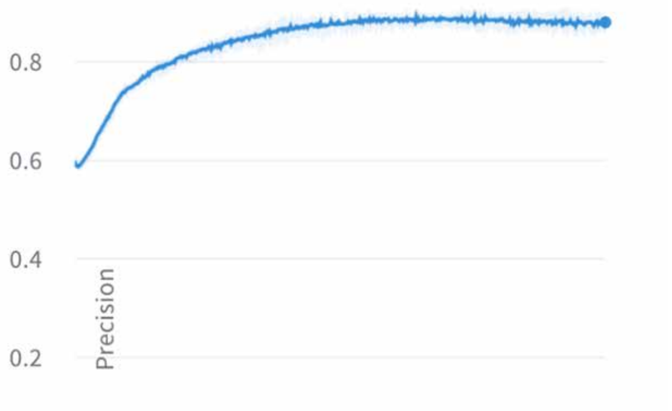


Figure 15: Precision results for image classification implementation test.

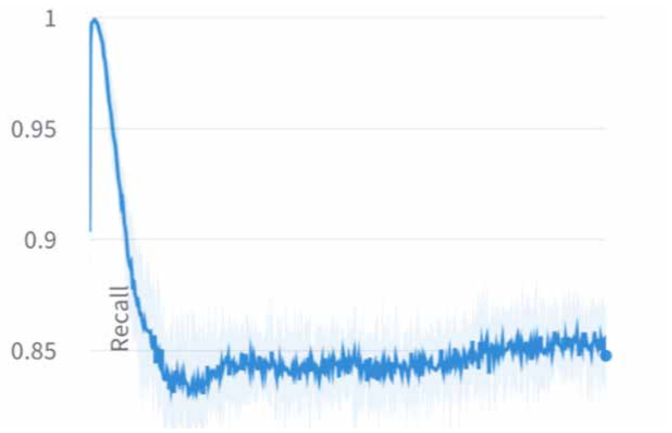


Figure 16: Figure 16. Recall results for image classification implementation test.

For our object detection models, we fine-tuned the pretrained YOLO-v5 model with synthetic images from the Limbo dataset. We used 8,000 synthetic images for training, and 2,000 synthetic images for testing. The dataset was comprised of images of single 30B or 48-type containers from Campaigns 2 and 3, respectively, and background (no containers) images from Campaign 6. We balanced the dataset with equal number of negative (background) and positive (either a 30B or 48-type container present) examples. For the positive examples, we had the same number of 30B and 48-type containers. The YOLO-v5 model was trained for 500 epochs.

Like the image classification models, we expected the performance of our train synthetic-test synthetic object

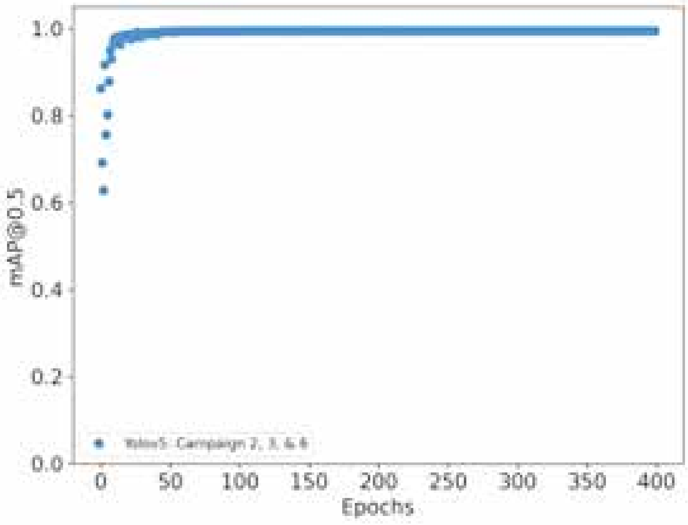


Figure 17: mAP scores using 0.5 detection threshold implementation test.

detection models to be high. Using a threshold of 0.5 for mAP, performance of the object detection models was near-ceiling as shown in Figure 17.

4.2 Image Classification Validation

Our first set of computer vision validation experiments were focused on image classification. For these experiments, we fine-tuned pre-trained ResNet-50 models using our synthetic Limbo data. The models were trained as one-class classifiers, with a sigmoid output between zero and one, where larger numbers indicated stronger predictions of the container class, and lower numbers indicated lower prediction of the container class. We elected 0.5 as the threshold for container classification, so that images with scores higher than 0.5 were considered a container class and images with scores lower than 0.5 considered a non-container class for the purposes of our evaluation.

We trained 10 models for each experimental run, using randomized initiation points for each model to ensure that training results were not serendipitous. We used this approach instead of cross-validation in order to train the models exclusively on synthetic data in each run and test them exclusively on real data (where cross-validation techniques would shuffle these training and test data sets). And we tested their performance on our full set of real images and recorded the average of the models' performance.

In Figure 18 and Figure 19, we show the results for all ten of the models but describe overall performance in relation to the mean of the ten models. Our classification experiment focused on single 30B container classifications and the experimental manipulation of the content of the negative training examples—either plain backgrounds, or synthetic distractors. For each trained model, we used an equal split of positive and negative examples.

In the first set of models (yellow/green tones along the bottom of Figure 18 and the top of Figure 19), the ResNet-50 model was trained on synthetic images of single 30B containers, with negative examples from background images without any containers. For these initial models, precision scores centered around 0.28 (lower cluster in Figure 18), and recall scores around 0.5 (higher cluster in Figure 19). The second set of models (in red/orange tones along the top of Figure 18 and the bottom of Figure 19), the ResNet-50models were trained on the same relevant containers, but with distractor containers as negative examples instead of backgrounds only. The precision scores for these models increased significantly, to around 0.58 (higher cluster in Figure 18), while the recall scores were around 0.35 (lower cluster in Figure 19).

The large increase in precision between the first and second set of models indicates that the models trained with synthetic distractors were better at selecting images with relevant containers and not selecting images without relevant containers. The decrease in recall scores between the first and second set of models indicates that the models became less likely to classify relevant containers than before.

We observed that as we made changes to the content type of the synthetic data, the models reacted in predictable ways—specifically, learning to be more discriminating with cylindrical objects before classifying them as relevant containers.

4.3 Object Detection

Our second set of validation experiments focused on object detection models. The object detection experiments evaluated models trained using subsets of Limbo to see how those subsets impacted model performance. In these experiments, we used an equal number of positive and negative examples to train the model. We considered a positive example to include one or more relevant containers of interest, and a negative example to contain no objects of interest (only background images or distractor containers). In these experiments, we used the YOLO-v5s object detection model, with an intersection over union (IOU) threshold of 0.25. Additionally, we calculated mean Average Precision (mAP) scores only for the 30B and 48-type containers.

As a baseline for performance, we trained models with 10,300 images containing individual containers (30B and 48-type containers). We compared performance of the baseline models to two alternatives: first, we trained models with images containing single containers (30B and 48) and single rows of containers (30B and 48). Second, we trained models with the same images, plus images containing distractors and individual containers.

Like we did for image classification, for each experiment, we trained 10 models with randomized initiation

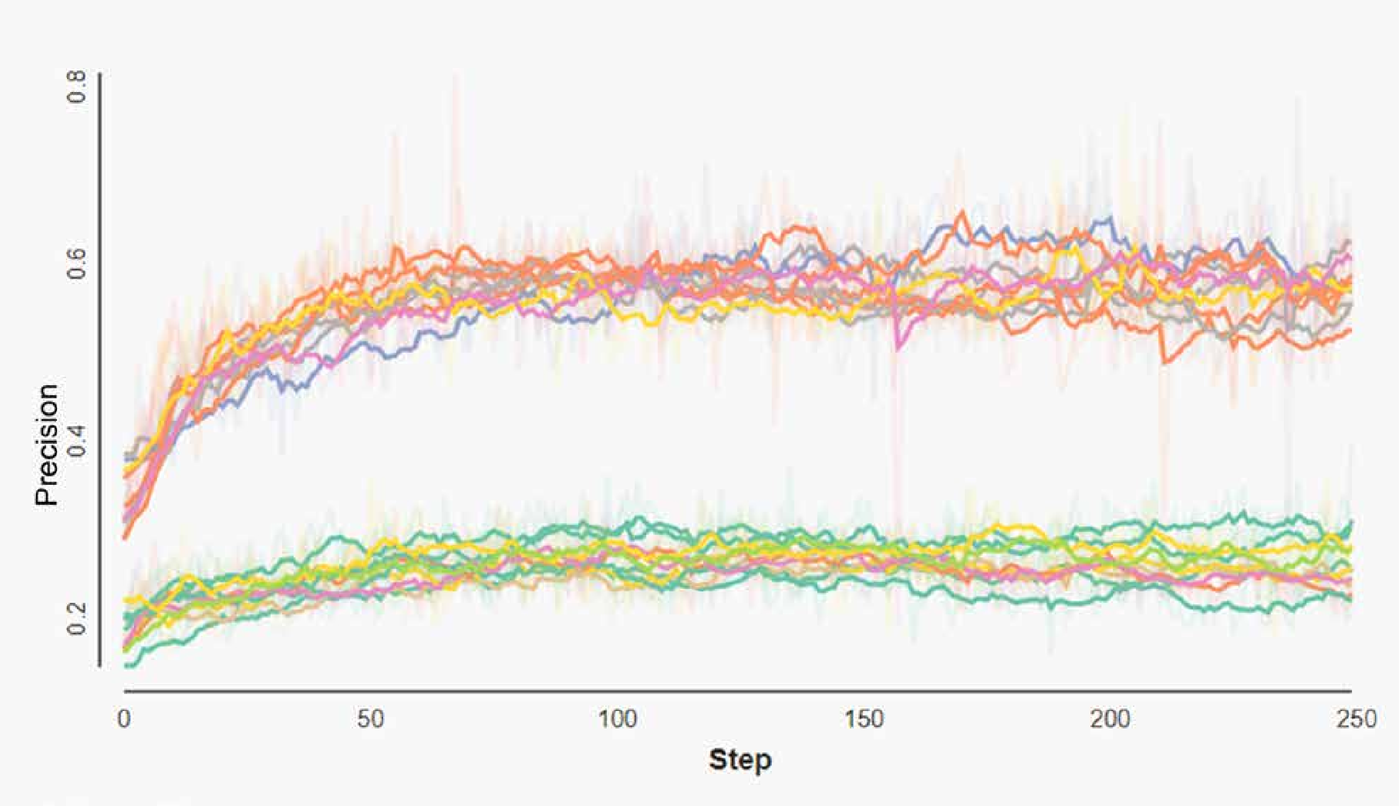


Figure 18: Precision scores from our image classification experiment show that image classification models trained with distractor objects (top cluster of red/orange lines) had higher precision than models trained without distractors using only background scenes as negative examples (bottom cluster of yellow/green lines).

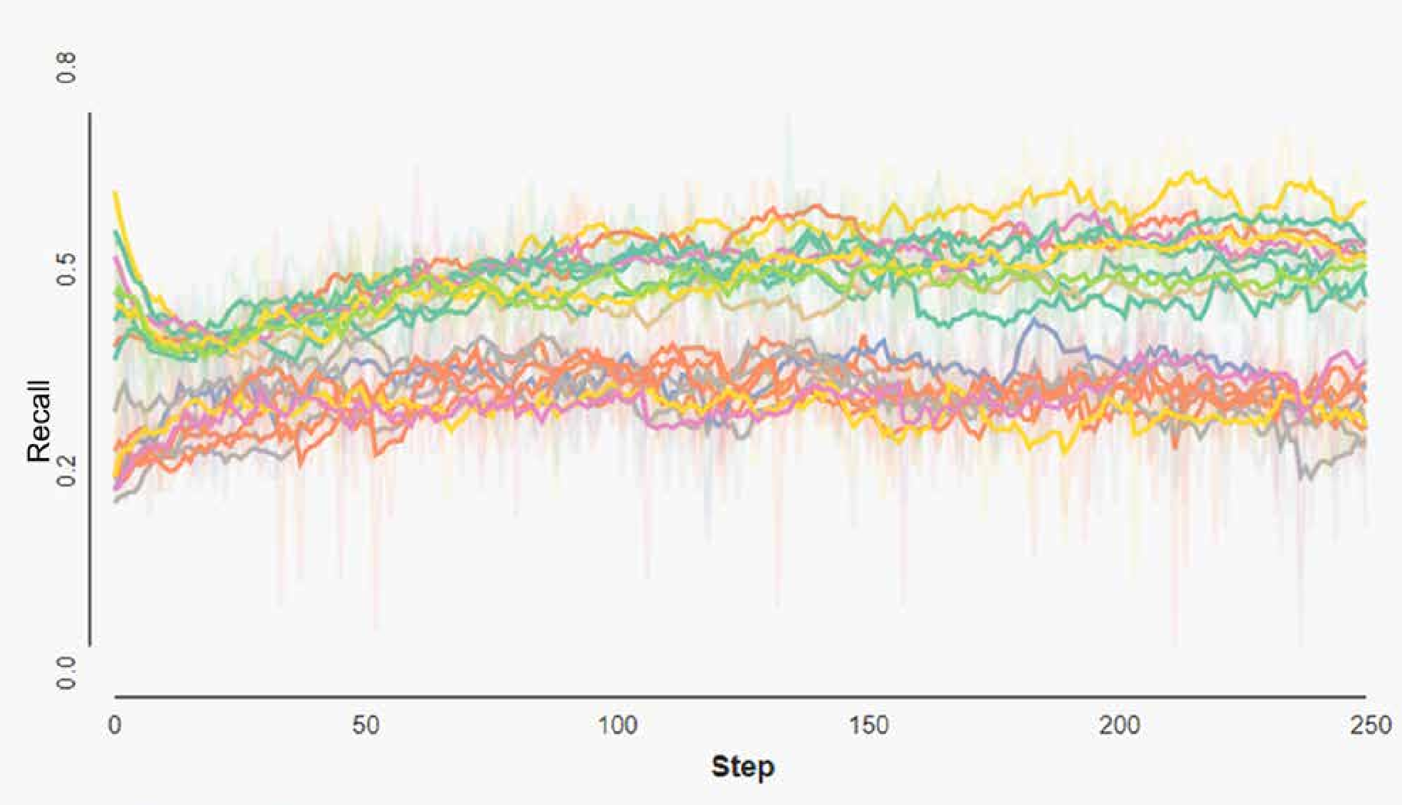


Figure 19: Recall scores for an image classification experiment show that image classification models trained with distractor objects as negative examples (bottom cluster of red/orange lines) had lower recall than models trained with background scenes as negative examples (top cluster of yellow/green lines).

parameters and took the mean of their results to ensure that test results were not the product of an especially high- or low-performing model. We found that by including images with rows of containers along with others showing individual containers during training, the mAP score improved compared to the baseline model where only individual containers were present, as shown in Figure 20. The real-world images contain scenarios where the relevant containers are in rows, and through inspection of the object detection results, we noticed models trained without examples of containers in rows, i.e., only using individual containers, struggled to identify examples when presented with a row of containers. By providing the model with examples of containers in rows in the training data, the model was able to learn that more containers were present and detect them.

The model trained using both containers and distractors increased the mAP score relative to the baseline model but did not improve performance relative to individual containers and rows. The two scenarios are within statistical deviation of each other, but the mean mAP score for models trained on individual containers and rows is higher. In this case, we hypothesize that by training the model with examples of distractors, especially distractors occluding UF6 containers, the model learned features of the occluding object and incorrectly associated it with the 30B or 48 containers, lowering the mAP score. Furthermore, by including distractors in the categories for the model to learn from in the training set, the object detection problem becomes harder because the model has more options to choose from, and we observed that the model confused 30B or 48-type containers for distractors in some instances, which also lowered the mAP score.

To better compare the impact of different synthetic images on our image classification and object detection models, we conducted an analysis in which we judged both models using common metrics. We adapted the signal detection

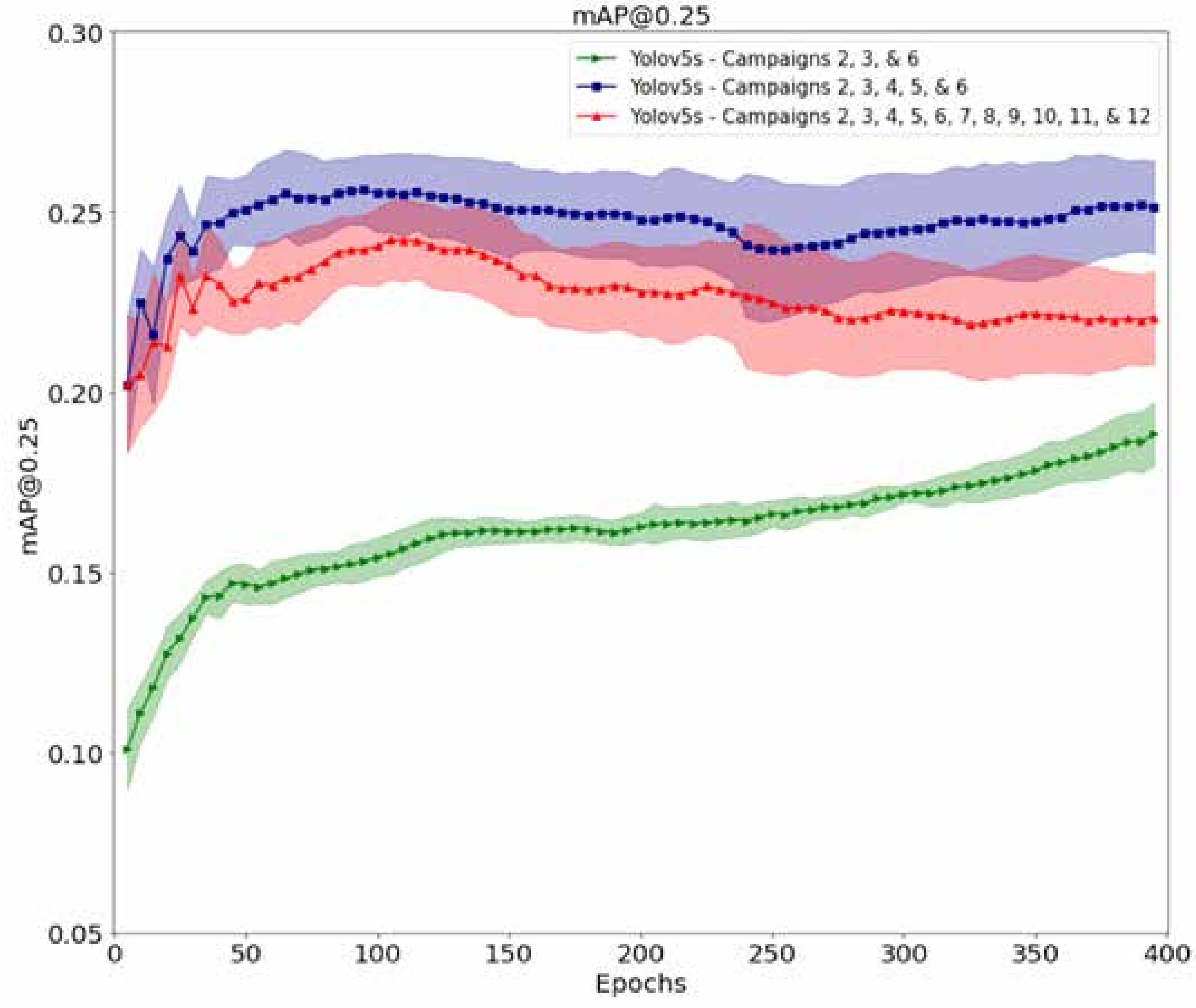


Figure 20: Mean Average Precision (mAP) Scores for Object Detection Experiments. As variety and complexity of training data increased, so did model performance. There was a minor difference in performance between models that were trained with individual and rows of relevant containers and models that also included distractor objects. The green, blue, and red lines indicate training runs with single containers (30B and 48) and background; single containers, background, and single rows of containers; and single containers, background, distractors plus single containers, and single rows of containers, respectively.

performance metrics used in image classification (true positive, true negative, false positive, and false negative) for object detection as follows:

- For any image that had an object of interest (as defined by our team's labeling), an object detection-generated bounding box for that type of object in the image was considered a true positive regardless of its location within the image.
- For any image that did not have an object of interest, the absence of an object detection-generated bounding box of that type in the image was considered a true negative.
- For any image that had an object of interest, but the object detection model did not place a bounding box of that type anywhere in the image, it was considered a false negative.
- For any image that did not have an object of interest, but the object detection model placed a bounding box of that type anywhere in the image, it was considered a false positive.

We present the performance of our object detection models when they were evaluated as classifiers in Figure 21. We provide a summary of observations from assessing our object detection models as classifiers, and model-to-model performance comparisons, below.

First, we observed that increases in performance from including more diverse images in training (as shown in Figure 20) was not as great for the object detection models when they were used as classifiers. This is likely due to an increase in the baseline model performance coming from the lower bar for true positives than for correct object detection.

Second, we had not previously tested differences in computer vision model performance between 30B and 48-type containers. Our early image classification testing focused mostly on 30B containers, and we did not differentiate container types in performance reporting in earlier object detection models. With this new testing, which included both types of containers and easily differentiated results based on how classification results are reported, we found that the object detector—when measured like a classifier—has a higher precision and recall with 48-type containers compared to the 30B containers. This may be due to the more visually distinct features of the 48-type containers compared to the 30B containers.

Third, we found stable patterns in the trade-off between precision and recall as we increased the variety in the training data. In Figure 18 and

Figure 19, we show that as we increased the variety, precision scores increased (i.e., a higher proportion of data retrieved was relevant) and recall scores decreased (i.e., fewer of the total relevant items were recalled), we see the same pattern in the object detection models when they are evaluated as classifiers.

We had anticipated this result, that as the models learn more about other types of containers that exist, they become more discriminating in their classifications and therefore may also miss more relevant items.

Finally, when we compared the performance of our image classifiers and object detectors on the same metrics of precision and recall, we found that both model types had similar precision scores, but recall scores were significantly higher for the object detection models. The underlying models are different, so it is difficult to make broad generalizations about what this could mean for computer vision generally, but it could indicate that models trained as object detectors are better able to identify—based on the more specific nature of the training data—relevant features that could increase recall, thereby decreasing the potential number of relevant items missed by these models.

4.4 Interpreting Model Results

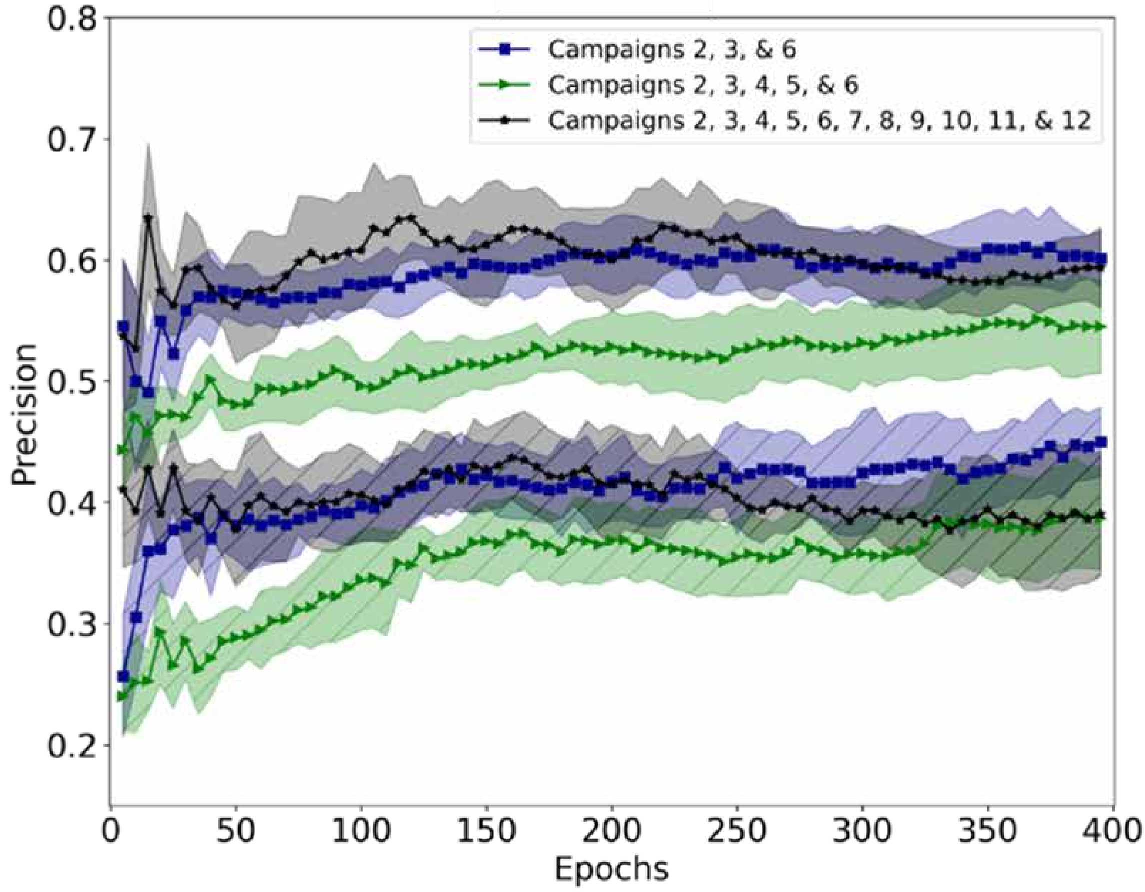
As described in Section 4, we interpreted the computer vision results using explainability techniques for the image classification models and visualization of the bounding boxes for the object detection models. Our most notable observations, and their subsequent impacts on the Limbo data, are described here.

Relevant containers in rows. One of our first observations from the image classification explainability activities was that when relevant containers were pictured in rows, a model that was trained on single containers only appeared to be focusing primarily on the first one or two containers. In response, we began generating rows of relevant containers such as might be seen in a shipping or storage area. These changes can be observed in campaigns 4 and 5.

Synthetic distractors. We also observed in our image classification explainability tests that the models were recognizing many real-world cylindrical objects as 30B or 48-type containers. We think this was caused by negative examples in early trials, which consisted of backgrounds without any additional synthetic content, such as synthetic cylindrical distractors. In response, we introduced synthetic distractors—primarily cylindrical, round, industrial objects. These changes can be observed starting in Campaign 7.

Synthetic distractors in groups. We thought it would be informative to render our distractor objects in groups or clusters, instead of the well-aligned rows of campaigns 4 and 5. This change can be seen starting in campaigns 8 and 9.

Partially occluded containers. As a follow-up to the changes made in point 1, we also wanted to occlude containers with distractor objects rather than just relevant containers.



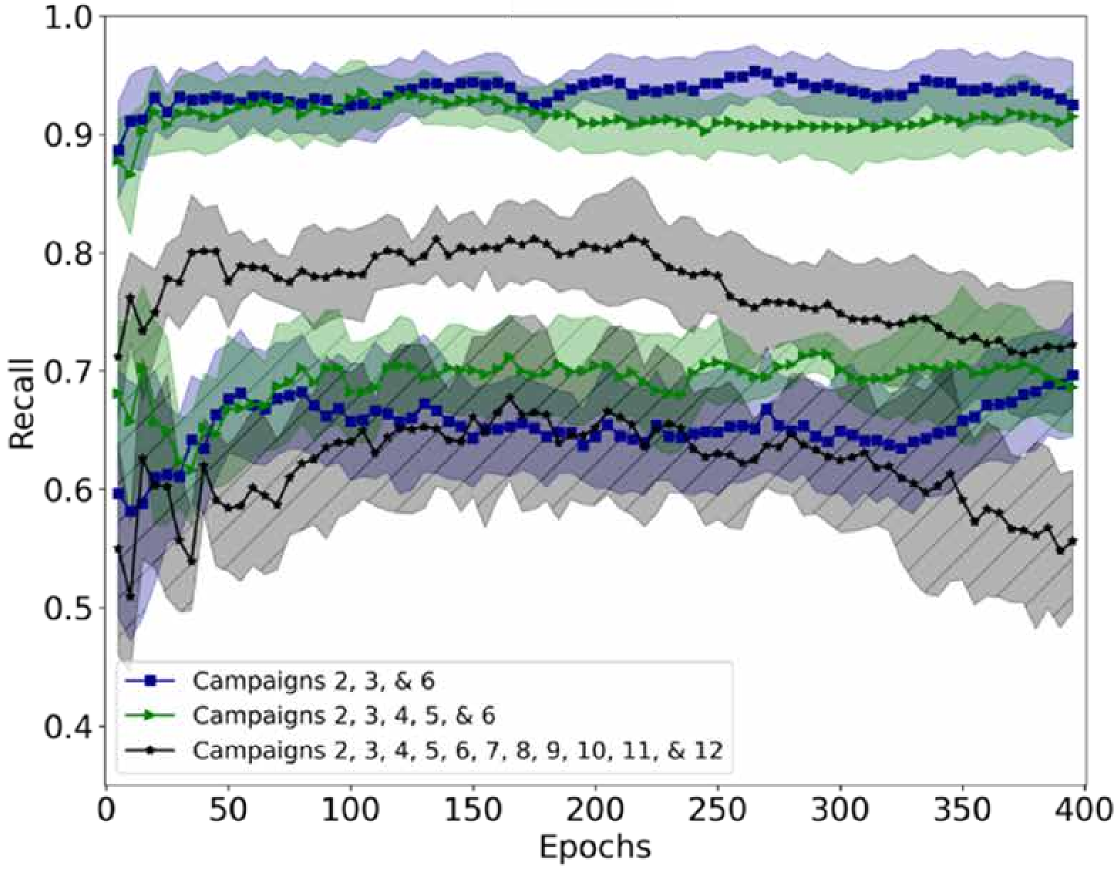


Figure 21: Applying signal detection performance assessments for determining precision (top) and recall (bottom) to the object detection results. The hatched lines represent performance on 30B containers, and the solid (no hatch marks) represent performance on the 48-type containers. See Table 1 for a description of the campaign details.

Combinations of distractors with relevant containers appear starting with campaigns 8 and 9.

Synthetic people. As seen in Figure 14, our object detection models frequently mis-labelled people as containers. In response, we introduced synthetic 3D people in campaigns 18 and 19.

Animated walk-through. During its development, we collaborated with partners who wanted to use the Limbo data for their own R&D. One project—the 3D Computer Vision for Safeguards project—is developing container counting capabilities intended for use by a safeguards inspector walking through a facility. In anticipation of their needs, Limbo campaign 20 provides an extensive animated walk-through of the synthetic environment that could be used for frame-by-frame tracking and counting of objects.

5. Discussion and future work

During our iterative image validation process, we made several general observations about training computer vision models with synthetic data, which we briefly summarize here along with thoughts on additional research.

First, negative examples are more effective when they include distractors. This observation came directly from our validation activities and is described in Section 4.2 and 4.3, as well as our discussion of updates to the data as an outcome of the validation process in Section 4.4

Second, object configuration and positioning had a larger influence on detection rates than expected. This was also addressed in Section 4.2 and 4.3, and included an update in our synthetic data described in Section 4.4.

Third, training computer vision models to be more discriminating through the inclusion of distractor objects in training data can lead to a classic performance trade-off of improved precision, but lower recall.

Fourth, computer vision models are generally learning the wrong lessons from training data. Anecdotally, there are many synthetic images in the Limbo dataset that our human colleagues found difficult to distinguish from real-world data. The problem of domain shift between datasets has been well-documented in computer vision research, and (Movshovitz-Attias, Kanade, & Sheikh, 2016) describes its relevance to synthetic as well as real datasets. However, we note that even when human observers can tell which images are real and which are synthetic, they still have no difficulty correctly recognizing the (real or synthetic) cylinders. Yet computer vision models display significant differences in performance when evaluating real and synthetic images. This implies not only that there are differences between the synthetic and real feature distributions, but that the models are making decisions based on image features that humans somehow ignore as irrelevant.

We acknowledge that the premise of computer vision models learning the wrong features may be controversial. However, it is our aspiration that computer vision models respond more like human observers and recognize the intended item across many varied environments. We think the ability of computer vision models to learn the defining visual characteristics of relevant objects is especially crucial for high consequence domains such as nuclear nonproliferation, where learning irrelevant features could have serious security consequences.

At this point, we think attention must be placed back on feature engineering and the models themselves: what are the features they are learning, and can we force them to learn only the features we deem important? Can we identify and prune features that are irrelevant? We believe the next step in computer vision research and development—especially for high-consequence domains where real-world data is limited and synthetic data will likely play a significant role—will require new ideas and new architectures that allow model trainers to explicitly specify the relevance of data.

6. Accessing and using the data

The images, metadata, reference data, and documentation for the Limbo dataset are available to the public as unclassified, unlimited release data. While Sandia does not own the reference data, we have checked copyright information to the best of our ability and have included only data that we believe is shareable. The full Limbo dataset, including one million synthetic images, hundreds of real-world reference images, and all associated metadata is hosted in the Lawrence Berkely National Laboratory's Berkeley Data Cloud (BDC). The data is open source and available to anyone with a free BDC account. Before accessing BDC, however, we recommend reading the documentation, terms of use, and API information detailed at:

https://limbo-ml.readthedocs.io/

The Limbo data is organized into a series of topical campaigns that provide a manageable file structure of roughly 50000 images each and reflect the lessons and observations from our data validation experiments (see Section 4). The rendering campaigns are described in Table 1, and in the documentation provided at our website.

Campaign No.	Campaign Description and Associated Figure.
2	30B containers viewed individually, in the relative center of the frame of real-world 3D HDR backgrounds. 50,000 of the images depict 30B containers, and 5,000 images show only the backgrounds without containers for use as negative examples. See Figure 1.
3	48-type containers (X, Y, and G designs intermixed) viewed individually, in the relative center of the frame of real-world 3D HDR backgrounds. 50,000 of the images depict 48 containers, and 5,000 images show only the backgrounds without containers for use as negative examples. See Figure 22.
4	48-type containers (X, Y, and G designs intermixed) arranged in rows in real-world 3D HDR backgrounds. 50,000 of the images depict 48 containers, and 5,000 images show only the backgrounds without containers for use as negative examples. See Figure 23.
5	30B containers arranged in rows framed in real-world 3D HDR backgrounds. 50,000 of the images depict 48 containers, and 5,000 images show only the backgrounds without containers for use as negative examples. See Figure 24.
6	No containers. This campaign contains images from our 3D HDR backgrounds as negative examples. See Figure 25.
7	Single synthetic distractor objects arranged in our real-world 3D HDR backgrounds. See Figure 26.
8	Single 30B containers pictured with a single distractor, in the real-world 3D HDR background. Depending on camera placement and container size, one of the containers might not be visible in some images. See Figure 27.
9	Single 48 containers pictured with a single distractor, in the real-world 3D HDR background. Depending on camera placement and container size, one of the objects might not be visible in some images. See Figure 28.
10	Clusters of distractor objects, including up to three distractor types, in real-world 3D HDR backgrounds. See Figure 29.
11	Single 30B container with up to three types of distractor objects clustered around the container, in real-world 3D HDR backgrounds. This campaign offers more views of occluded containers than previously demonstrated. See Figure 30.
12	Single 48 container with up to three distractor objects clustered around the container, in real-world 3D HDR backgrounds. This campaign offers more views of occluded containers than previously demonstrated. See Figure 31.
13	Highly complex environment with a single 48 container and many distractors of up to 10 different types filling the frame, in real-world 3D HDR backgrounds. These images are intended to test the limits of computer vision applications. See Figure 32.
14	Highly complex environment with a single 48 container and many distractors of up to 10 different types filling the frame, in real-world 3D HDR backgrounds. These images are intended to test the limits of computer vision applications. See Figure 33.
15	Each individual UF6 container type developed for this project, with every possible surface type, viewed from many angles. Backgrounds are real-world 3D-HDR backgrounds. See Figure 34.
16	Between 0 – 50 30B containers with multiple distractors placed in synthetic 3D oil refinery background. See Figure 35
17	Between 0 – 50 48 containers with multiple distractors placed in synthetic 3D oil refinery background. See Figure 36.
18	Single 30B containers with multiple distractors and with the addition of people placed in synthetic 3D oil refinery background. See Figure 37.
19	Single 48 containers with multiple distractors and with the addition of people placed in synthetic 3D oil refinery background. See Figure 38.
20	30B and 48 containers pictured together, with distractor objects, in an animated walkthrough of the synthetic oil refinery background. This campaign is intended for use in computer vision research involving video data. See Figure 39.

Table 1: Limbo campaign descriptions.

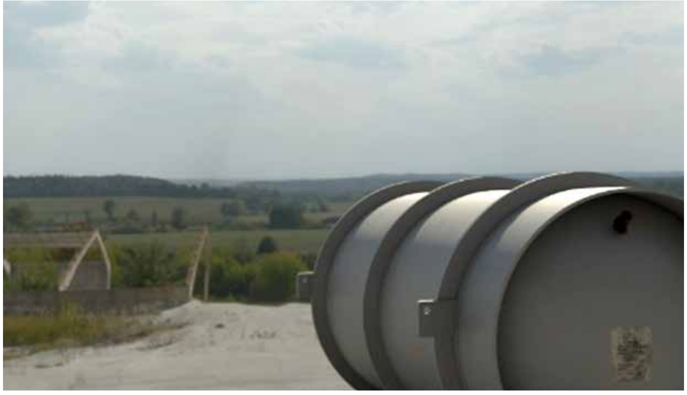


Figure 22: Example from campaign 3, with a single 48-type container.

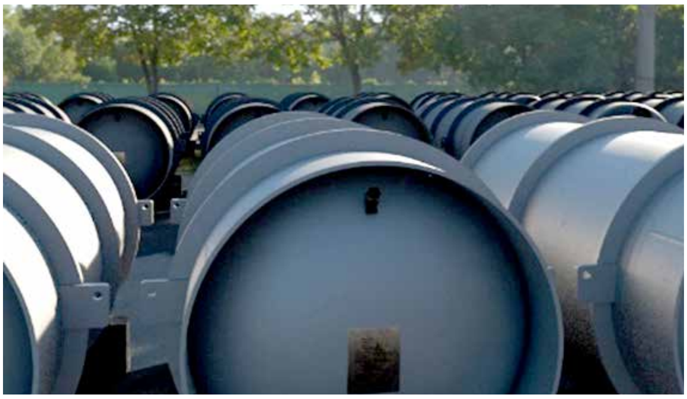


Figure 23: Example from campaign 4, showing rows of 48-type containers.

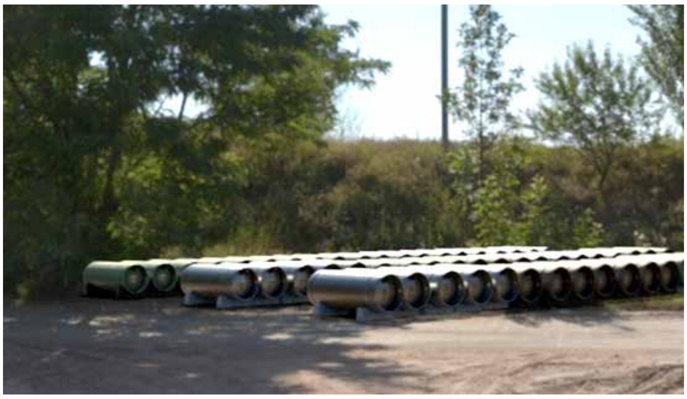


Figure 24: Example from campaign 5, with rows of 30B containers.

Figure 25: Example from campaign 6, showing a background image with no containers.

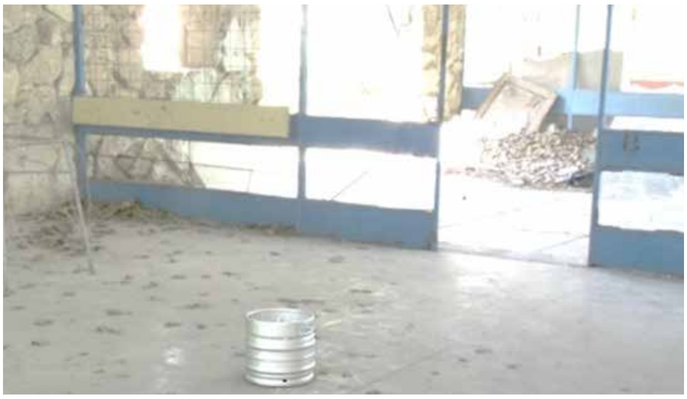


Figure 26: Example from campaign 7, with a single distractor object.

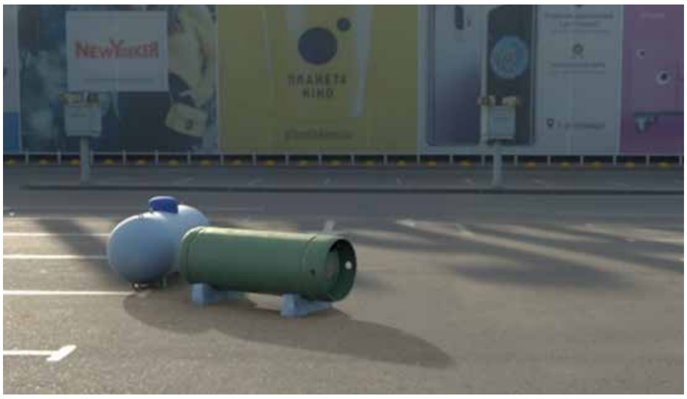


Figure 27: Example from campaign 8, showing one distractor and one 30B container.

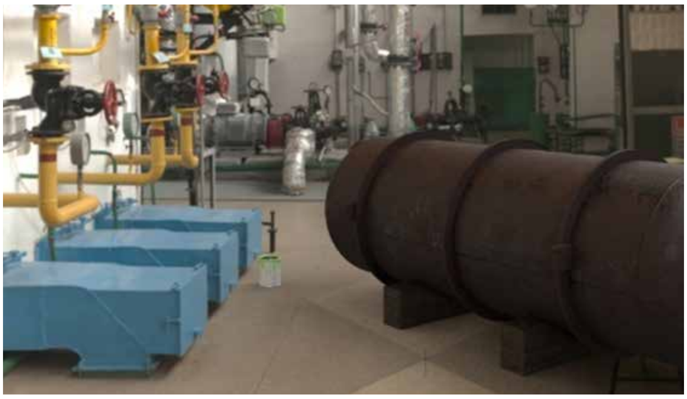


Figure 28: Example from campaign 9, with a single 48-type container and one distractor.

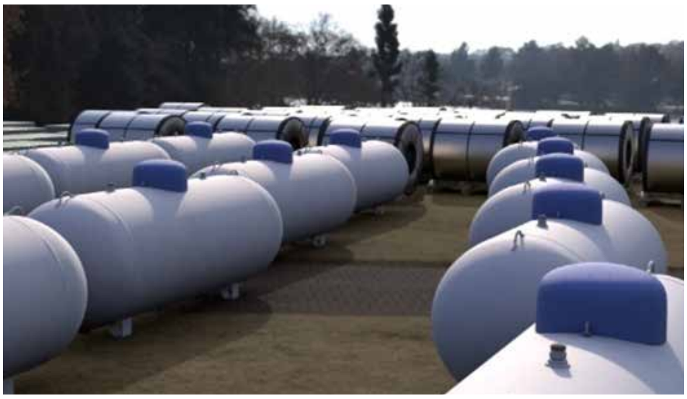


Figure 29: Example from campaign 10, with groups of distractors.

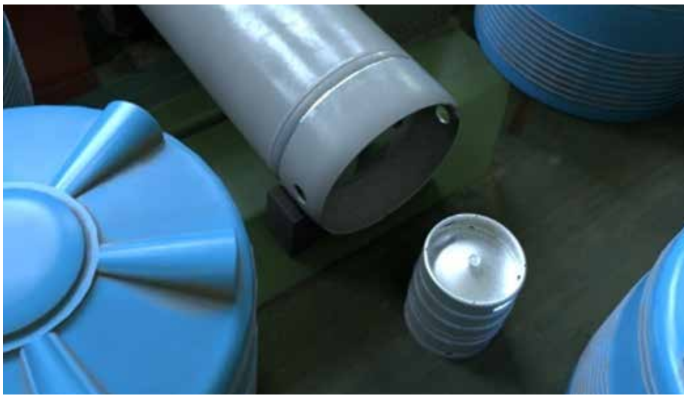


Figure 30: Example from campaign 11, showing groups of distractors with one 30B container.

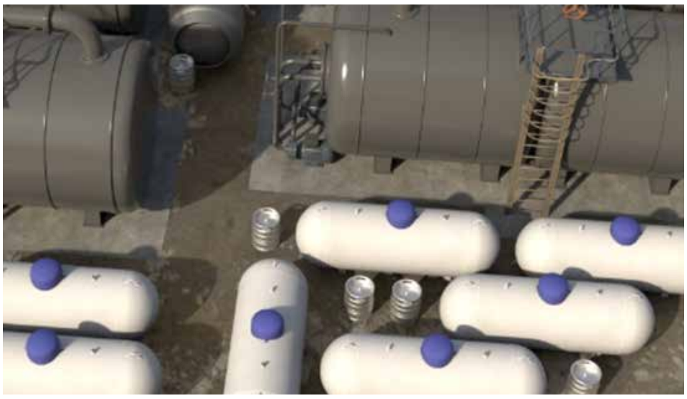


Figure 31: Example from campaign 12, with groups of distractors with one 48-type container.

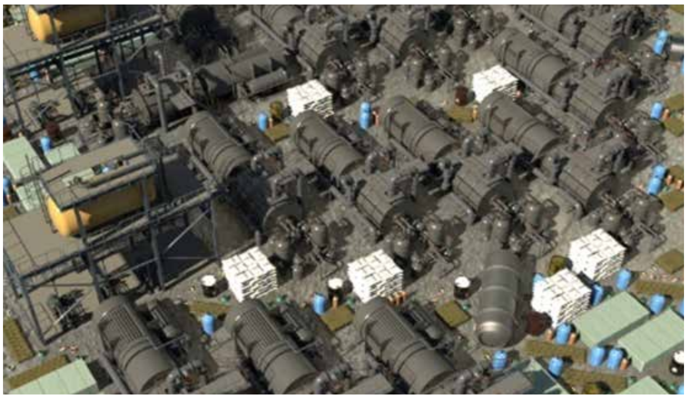


Figure 32: Example from campaign 13, with many distractors and one 48-type container.

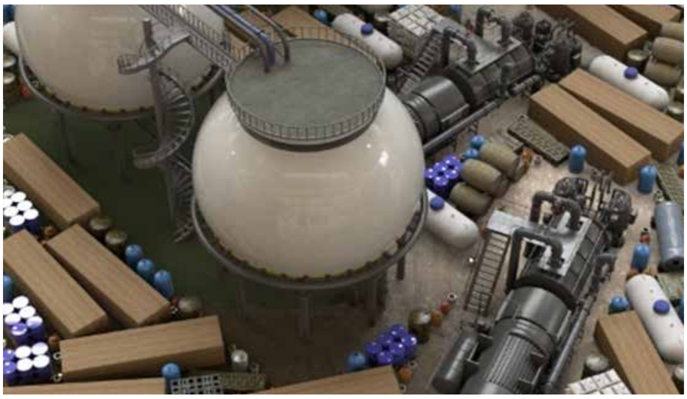


Figure 33: Example from campaign 14, with many distractors with one 30B container.

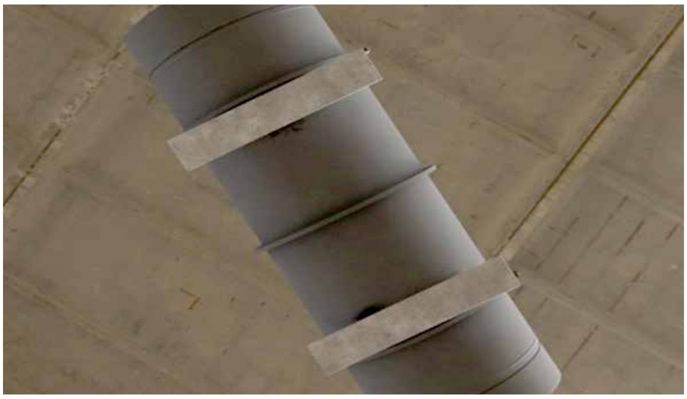


Figure 34: Example from campaign 15, in which a 30B container is pictured from below.

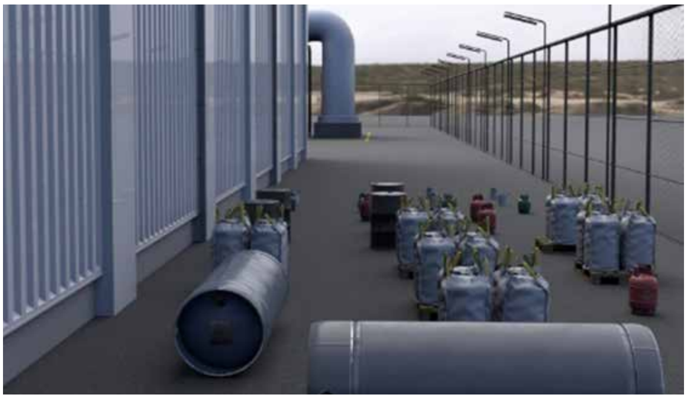


Figure 35: Example from campaign 16, with several 30B containers and distractors in a synthetic background.

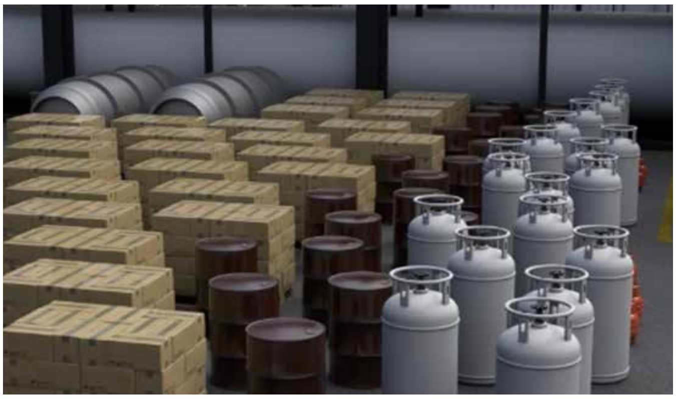


Figure 36: Example from campaign 17, with several 48-type containers and distractors in a synthetic background.

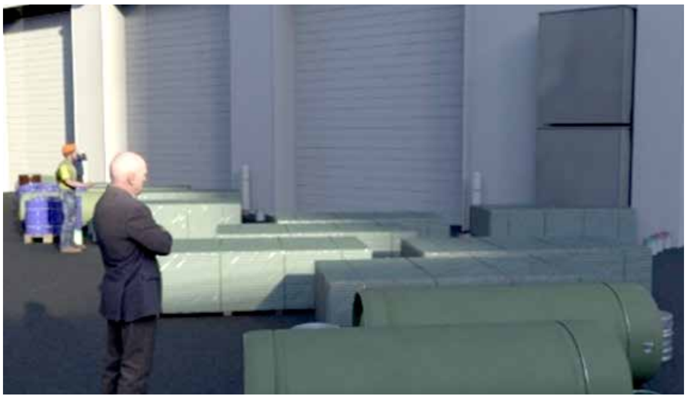


Figure 37: Example from campaign 18, with 30B containers, distractors, and people in a synthetic background.

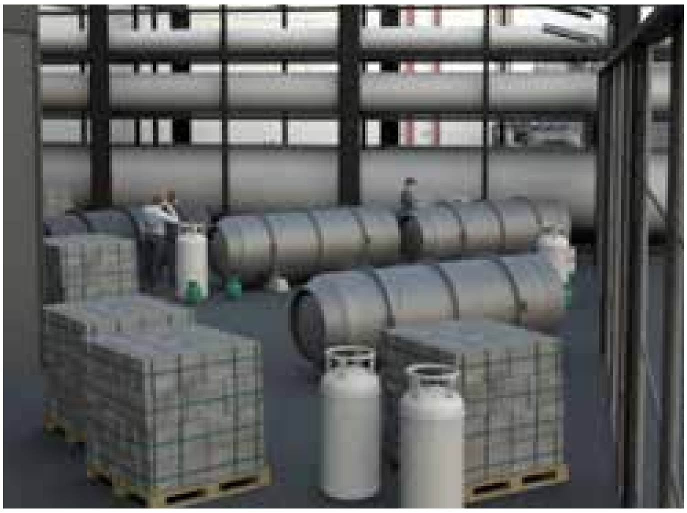


Figure 38: Example from campaign 19, with 48-type containers, distractors, and people in a synthetic background.distractors, and people in a synthetic background.

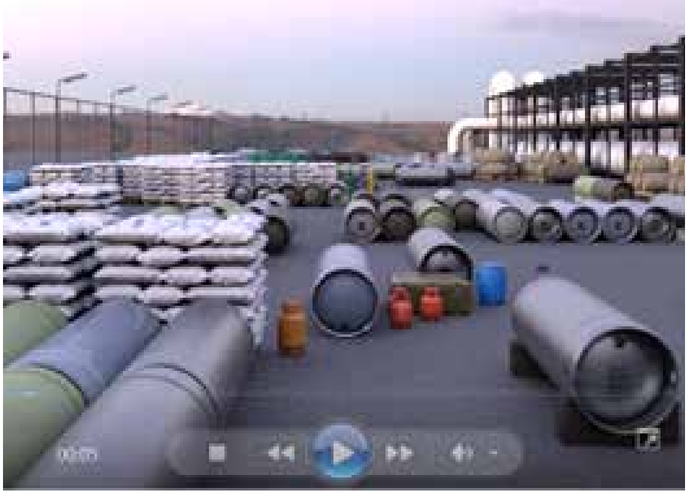


Figure 39: Example from campaign 20, an animated walk-through of 30B and 48-type containers with distractors in a synthetic background.

7. Acknowledgments

This work was funded by the U.S. Department of Energy's National Nuclear Security Administration, Defense Nuclear Nonproliferation Research & Development program. Additional thanks go to Lawrence Berkeley National Laboratory for hosting the data developed by this project on the Berkeley Data Cloud, and to Lawrence Livermore National Laboratory for providing seed data for our real-world image collection. Thank you to Sydney Dorawa, Arielle Mattes, and Jason Bolles for their support in collecting and labeling real-world images.

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC (NTESS), a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration (DOE/NNSA) under contract DE-NA0003525. This written work is authored by an employee of NTESS. The employee, not NTESS, owns the right, title and interest in and to the written work and is responsible for its contents. Any subjective views or opinions that might be expressed in the written work do not necessarily represent the views of the U.S. Government. The publisher acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this written work or allow others to do so, for U.S. Government purposes. The DOE will provide public access to results of federally sponsored research in accordance with the DOE Public Access Plan. Sandia Tracking Number: 1724688.

8. References

[1] American Nuclear Standards Institute. (2001). American Nuclear Standard for Nuclear Materials - Uranium Hexafluoride - Packaging and Transport. Retrieved from https://law.resource.org/pub/us/cfr/ibr/002/ansi.n14.1.2001.pdf

[2] Arno, M. (2018, October). Streamlining Open-Source Proliferation-Relevant Video Identification, Collection, and Processing. Presented at the Emerging Information Analysis Concepts for Nuclear Nonproliferation and Security Workshop. Ann Arbor, MI.

[3] Carlini, N., & Wagner, N. (2017). Toward Evaluating the Robustness of Neural Networks. IEEE Symposium on Security and Privacy.

[4] Ekbatani, H., Pujol, O., & Segui, S. (2017). Data Generation for Deep learning in Counting Pedestrians. ICPRAM, (pp. 318-323). doi: 10.5220/0006119203180323

[5] Feldman, Y., Arno, M., Carrano, C., Ng, B., & Chen, B. (2018). Toward a Multi-Modal Deep Learning Retrieval System for Monitoring Nuclear Proliferation Activities. Journal of Nuclear Materials Management, XLVI(3), 68-80.

[6] Friedman, J., & Jones, A. (2021, May 02). Cryptomatte. Retrieved December 05, 2023, from Github: https://github.com/Psyop/Cryptomatte

[7] Gaidon, A., Wang, Q., Cabon, Y., & Vig, E. (2016). Virtual Worlds as Proxy for Multi-Object Tracking Analysis. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (pp. 4340-4349).

[8] Gastelum, Z. N., & Shead, T. M. (2018). Inferring the Operational Status of Nuclear Facilities with Convolutional Neural Networks to Support International Safeguards Verification. Journal of Nuclear Materials Management, XVLI(3), 37-47.

[9] Gastelum, Z. N., & Shead, T. M. (2020). How Low Can You Go? Using Synthetic 3D Imagery to Drastically Reduce Real-World Training Data for Object Detection. Sandia National Laboratories. Retrieved from https://www.osti.gov/servlets/purl/1670874

[10] Gastelum, Z. N., Shead, T. M., & Marshall, M. R. (2022). But it Looks so Real! Challenges in Training Models with Synthetic Data for International Safeguards. Proceedings of the Institute of Nuclear Materials Management Annual Meeting.

[11] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. arXiv preprint. Retrieved from https://arxiv.org/pdf/1412.6572.pdf

[12] He, K., Xiangyu, Z., Shaoqing, R., & Sun, J. (2016). Deep Residual Learning for Image Recogntition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (pp. 770-778). Retrieved from https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf

[13] Jocher, G., Stoken, A., Chaurasia, A., Borovec, J., NanoCode012, Kwon, Y., . . . Jain, M. (2022). ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation (v7.0). Zenodo. doi:https://doi.org/10.5281/zenodo.7347926

[14] Joshi, T., Cooper, R., Okamura, A., von Sudderth, A., Roberts, B., & Valentine, J. (2021). Multi-Sensor Fusion for Nuclear Material Container Counting and Assay: 3D Computer Vision for Nuclear Material Accountancy in Large Complex Environments. (internal) NSARD program review. Virtual.

[15] Lin, T.-Y., Maire, M., Belongie, S., Bouredev, L., Girshick, R., Hays, J., . . . Dollar, P. (2015). Microsoft COCO: Common Objects in Context. arXiv. Retrieved from https://arxiv.org/pdf/1405.0312v3.pdf

[16] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single Shot Multibox Detector. European Conference on Computer Vision (pp. 21-37). Springer.

[17] Movshovitz-Attias, Y., Kanade, T., & Sheikh, Y. (2016). How Useful is Photo-Realistic Rendering for Visual Learning. Computer Vision-ECCV 2016 Workshops. Amsterdam: Springer International Publishing. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-49409-8_18

[18] Northcutt, C. G., Athalye, A., & Mueller, J. (2021). Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks. 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Retrieved from https://arxiv.org/abs/2103.14749

[19] Oak Ridge Operations. (1987). Uranium Hexafluoride: Handling Procedures and Container Descriptions. Retrieved from https://www.osti.gov/servlets/purl/6304596

[20] Papernot, N., McDaniel, P., Wu, X., Somesh, J., & Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. IEEE Symposium on security and privacy , (pp. 582-597).

[21] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv: 1804.02767.

[22] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Toward Real-Time Object Detection with Regional Proposal Networks. Advanced in Neural Information Processing Systems, 28.

[23] Rutkowski, J., Canty, M. J., & Nielsen, A. A. (2018). Site Monitoring with Sentinel-1 Dual Polarization SAR Imagery Using Google Earth Engine. Journal of Nuclear Materials Managament, XLVI(3), 48-59.

[24] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Gran-cam: Visual Explanations from Deep Networks via Gradient-Based Localization. Proceedings of the IEEE International Conference on Computer Vision. .

[25] Smartt, H. A., Gastelum, Z. N., Rutkowski, J. E., Peter-Stein, N., & Shoman, N. (2021). Hey Inspecta! Proceedings of the INMM & ESARDA Joint Virtual Annual Meeting.

[26] Smith, M. R., Hamel, M., Hannasch, D., Thomas, M., & Gaiten-Cardenas, C. (2021). A Deep Learning Workflow for Spatio-Temporal Anomaly Detection in NGSS Camera Data. Proceedings of the Institute of Nuclear Materials Management and European Safeguards Research & Development Association Joint Annual Meeting. Virtual.

[27] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovivh, A. (2015). Going Deeper with Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (pp. 1-9).

[28] Tan, R. J. (2019, March 24). Breaking Down Mean Average Precision (mAP). Retrieved November 27, 2023, from Medium: https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52

[29] Thomas, M., Passerini, S., Cui, Y., Rutkowski, J., Yoo, S., Lin, Y., . . . Moeslinger, M. (2021). Deep Learning Techniques to Increase Productivity of Safeguards

Surveillance Review. Proceedings of the Institute of Nuclear Materials Management and European Safeguards Research & Development Association Joint Annual Meeting. Virtual.

[30] Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., . . . Birchfield, S. (2018). Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (pp. 969-977).

[31] United States Enrichment Corporation. (1995). Uranium Hexafluoride: A Manual of Good Handling Practices. Retrieved from https://www.osti.gov/servlets/purl/205924

[32] Wolfart, E., Casado Coscolla, A., & Sequeira, V. (2022). Deep Learning for Video Surveillance Review. Proceedings of the Institute of Nuclear Materials Management Annual Meeting.

[33] Wolfart, E., Sanchez-Belenguer, C., & Sequeira, V. (2021). Deep Learning for Nuclear Safeguards. Proceedings of the INMM & ESARDA Joint Virtual Annual Meeting, (pp. 1-10).