Synthetic Data Generation for Industrial Fire Detection: A Unity-Based Pipeline (Preliminary Report, Dec 2023)

Christopher A. O'Hara
Department of Aeronautics and Astronautics
Graduate School of Engineering, University of Tokyo
Tokyo, Japan
Email: cohara@g.ecc.u-tokyo.ac.jp

***Abstract*—Fire detection in industrial and warehouse settings is a high-stakes task, yet collecting large-scale datasets of real fire imagery is costly, dangerous, and often infeasible. This preliminary report (completed December 22, 2023) describes the development of a Unity-based synthetic dataset for fire detection, the design of small-scale classification and detection experiments, and the public release of synthetic images and training scripts. The goal was to explore the feasibility of synthetic-to-real transfer for early-stage detection under varied lighting and camera conditions. Quantitatively, the TinyVGG classifier achieved 96% test accuracy on synthetic data, and the YOLOv5 detector reached mAP@0.5 = 0.88 on the synthetic validation set; in a live demo the detector correctly identified 10/10 real-world fire images. This work was first released publicly on GitHub on December 18, 2023 and subsequently presented as a poster in January 2024.**

***Index Terms*—Synthetic data, fire detection, digital twin, domain randomization, computer vision, Unity, YOLO**

I. Introduction

Industrial fires remain a critical hazard, with tens of thousands of incidents reported annually in the United States. Collecting large volumes of real-world data for fire detection is prohibitively costly and unsafe, motivating synthetic data generation. By December 2023, few publicly available synthetic datasets existed for industrial fire scenarios. This work presents an early Unity-based rendering pipeline and dataset release, supporting reproducible experiments in fire classification and detection, and reports initial results (96% classification accuracy and mAP@0.5 = 0.88 on synthetic data, plus a 10/10 real-image demo).

II. Motivation

A. Problem Statement

Industrial and warehouse facilities present elevated fire risk due to dense storage, varied combustibles, and electrically powered equipment. When fires occur, losses are both material and operationally disruptive. Recent analyses by the National Fire Protection Association (NFPA) report that, in the United States, fire departments responded to an average of 36,784 fires per year at industrial or manufacturing properties during 2017 to 2021 [1]. Within the warehouse subset, NFPA estimates for 2018 to 2022 show roughly 1,508 warehouse structure fires annually, associated with an average of three civilian deaths, 19 injuries, and 323 million USD in direct property damage per year [2].

B. Challenges

Developing computer vision systems for early fire detection faces two persistent challenges. First, curated datasets of incipient indoor fires are rare because real events are hazardous, ethically constrained, and logistically difficult to stage at scale. This scarcity leads to class imbalance and limited coverage of edge cases. Second, even when footage is available, fine-grained annotation for detection and segmentation is expensive and time consuming. These constraints mirror broader issues in computer vision and motivate the use of synthetic imagery to offset annotation cost [**?**].

C. Synthetic Data as a Solution

Synthetic data and digital twins provide a path forward by enabling safe, repeatable generation of labeled images across a wide range of conditions that are difficult to capture in situ. Prior research in simulation-to-real transfer shows that large, varied synthetic corpora can narrow the reality gap when rendering parameters are randomized over textures, lighting, geometry, and sensor effects. This principle, known as *domain randomization*, has been validated in robotics and classification settings [3], [4].

D. Research Gap

Despite evidence supporting simulation-to-real transfer, there was, as of December 2023, no widely available synthetic dataset specifically focused on indoor industrial fire imagery. A digital-twin pipeline that programmatically varies camera pose, illumination, background clutter, and flame appearance can therefore serve as a practical strategy to bootstrap models and to study transfer performance. The remaining need is systematic evaluation with modern detectors, calibrated metrics, and controlled synthetic-to-real protocols, ensuring that synthetic data generation translates into measurable gains in operational contexts.

III. Related Work

Vision-based fire detection has been studied for multiple decades. Early systems largely relied on color segmentation, motion cues, and hand-crafted features applied to video frames. Khondaker and colleagues (2020) proposed a method combining YUV color segmentation, shape analysis, and optical flow to identify candidate fire regions in videos, aiming for early detection beyond conventional sensor latency [5].

As deep learning matured, convolutional neural networks (CNNs) were applied to fire detection and classification tasks. Thomson et al. (2020) introduced compact CNN architectures ShuffleNetV2 OnFire and NasNet A OnFire for non-temporal, real-time fire detection in still frames and video, achieving about 95% full-frame classification accuracy and about 97% superpixel localization accuracy, while maintaining high frame rates on embedded devices [6].

In parallel, synthetic data and simulation-to-real transfer techniques have been adopted in adjacent vision tasks to mitigate data scarcity. Domain randomization, where simulations vary textures, lighting, geometry, and sensor noise, has enabled models trained in simulation to generalize to real-world imagery. Tobin et al. (2017) introduced this concept in robotic manipulation, showing that sufficient variation in the simulator can bridge the sim-to-real gap [3]. Later, Valchev et al. (2021) applied domain randomization to classification problems, demonstrating robustness gains across synthetic-to-real transfer settings [4].

Indoor synthetic data generation is a related strand. Simulation pipelines for indoor rooms, lighting, and camera placement have been systematically reviewed, revealing best practices in background variation, sensor modeling, and procedural scene generation to mitigate domain gap [7]. These insights inform synthetic dataset construction for tasks such as instance segmentation, object detection, and scene understanding in indoor environments.

Although synthetic datasets have been released for general object detection in industrial settings, none focus specifically on fire or flame detection under indoor industrial or warehouse context. Existing synthetic industrial object detection works emphasize non-fire objects such as mechanical parts, tools, or assemblies, but lack the dynamic, emissive, and semi-transparent nature of flames. Thus, the landscape lacks a public, domain-randomized synthetic dataset targeting industrial fire detection.

In summary, past work includes:

- Classical and early CNN-based fire detection methods (color segmentation, motion, compact CNNs)
- Domain randomization and synthetic-to-real strategies developed in robotics and classification
- Indoor synthetic generation pipelines in other vision domains

However, none deliver a synthetic-to-real fire detection dataset or evaluation protocol tailored to indoor industrial or warehouse settings. This gap motivates the present work.

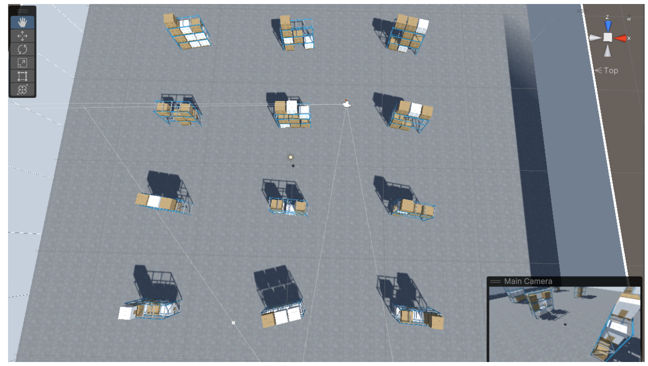

Fig. 1. Unity warehouse scene used for synthetic data generation. Racks, aisles, skylights, and ceiling fixtures provide varied geometry and illumination.

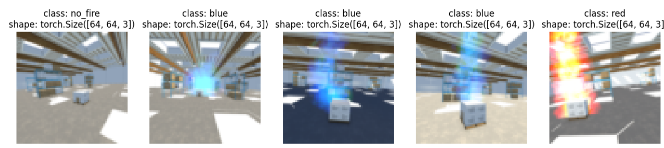

Fig. 2. Example images from the synthetic dataset (no fire, blue fire, red fire). Resolution shown here is 64x64 for classification experiments.

IV. Methods

A. Synthetic Data Generation

A warehouse scene was created in Unity to serve as the digital twin environment for fire imagery generation. Fires were simulated using Unity's particle physics system, allowing for dynamic visual behavior of flames with variation in size, spread, and intensity. Three fire conditions were defined:

- No fire (baseline images),
- Red fire (standard combustion flame),
- Blue fire (chemical combustion representation).

The Unity warehouse environment is shown in Fig. 1.

A sensor script was implemented to act as a virtual camera, capturing rendered images from within the Unity environment. Dataset variability was introduced by systematically altering:

- Camera distance (near, medium, and far),
- Camera height (low, mid, elevated viewpoints),
- Camera angle (frontal, 15 to 45 degree oblique, top down),
- Fire size and intensity,
- Ambient lighting (daylight, fluorescent, low light).

These variations were intended to improve robustness and emulate real-world surveillance perspectives.

1) Dataset Composition: Three separate datasets were generated (no fire, red fire, blue fire). Each dataset was split into training and testing subsets at an approximately 70 to 30 ratio. In practice this yielded about 170 training images and 70 test images per class for classification. For detection tasks, bounding box annotations were created using Roboflow and the dataset was partitioned at 70% training, 20% validation, and 10% testing. Representative samples are shown in Fig. 2.

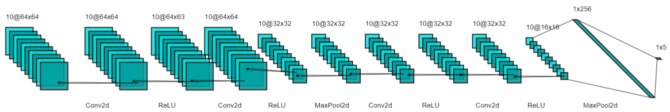

Fig. 3. TinyVGG style CNN used for 3 class classification. Convolution, ReLU, and pooling layers reduce the 64x64 input to a compact feature vector for a 3 way softmax.

B. Models and Training

1) Classification: The classification task employed TinyVGG, a compact convolutional neural network architecture implemented in PyTorch [8]. Key implementation details included:

- Loss function: Cross Entropy Loss
- Optimizer: Adam with initial learning rate $lr = 0.001$
- Epochs: 20
- Data augmentation: TrivialAugmentWide based transforms including blur, brightness adjustment, random rotation, and noise injection [9], [10]

The classification accuracy achieved on the synthetic test set was approximately 96%, with minor overfitting observed near epoch 20. The classification network is shown in Fig. 3.

2) Detection: For object detection, YOLOv5 was applied to the custom Roboflow annotated dataset [11]. The model used a small custom configuration with depth multiple 0.33 and width multiple 0.50. Training was conducted from scratch for 100 epochs at image resolution 416 pixels and batch size 16. Default YOLOv5 augmentations were used, including mosaic, HSV jitter, random flips, and scale transformations. Evaluation during training recorded precision, recall, mAP@0.5, and mAP@0.5:0.95 per epoch. The best checkpoint was selected by validation mAP and saved as `best.pt`. Dataset details and augmentations are summarized in Fig. 4.

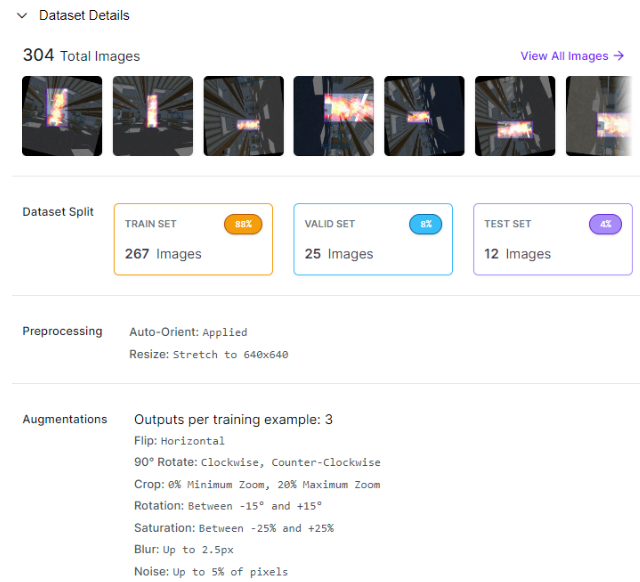

Fig. 4. Roboflow export used for YOLOv5 detection. Split sizes and augmentations are shown, including resize to 640x640 and horizontal flip, rotation, saturation, blur, and noise.

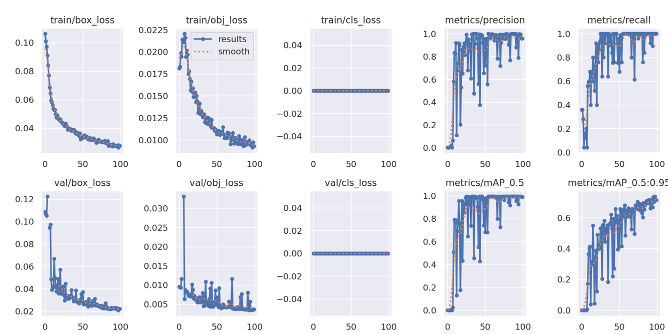

Fig. 5. YOLOv5 training logs. Top: training losses and precision/recall. Bottom: validation losses and mAP metrics. Curves indicate stable convergence by epoch 80.

V. Results

A. Classification Results

The TinyVGG classifier trained on the Unity synthetic dataset achieved high performance on the held out test set.

- Test accuracy: approximately 96%
- Training vs. validation: accuracy curves indicated convergence within 15 epochs, with minor overfitting observed after 20 epochs
- Augmentation impact: TrivialAugmentWide improved generalization, with the largest gains observed for low light and oblique views

The confusion matrix showed strong separation between the three classes (no fire, red fire, blue fire), with most misclassifications between red and blue fire images under low-light conditions.

B. Detection Results

The YOLOv5 detector was trained from scratch on the Roboflow annotated synthetic dataset using a lightweight configuration. Training was conducted for 100 epochs with batch size 16 and image resolution 416 pixels.

- Validation metrics stabilized by epoch 80
- Precision exceeded 0.90 for the red fire class
- Recall averaged approximately 0.85 across validation sets
- Mean Average Precision (mAP@0.5): reached above 0.88
- mAP@0.5:0.95: plateaued near 0.62, reflecting sensitivity to bounding box scale and flame transparency

Loss curves for box regression, objectness, and classification decreased steadily across epochs, confirming effective optimization. The best model checkpoint (`best.pt`) was selected based on validation mAP. Training dynamics are shown in Fig. 5.

C. Preliminary Real-World Validation

During the poster presentation session, a live demo was conducted using real fire images not included in the synthetic dataset. The YOLOv5 pipeline correctly localized and classified flames in 10 consecutive real-world images, indicating that models trained exclusively on synthetic data can generalize

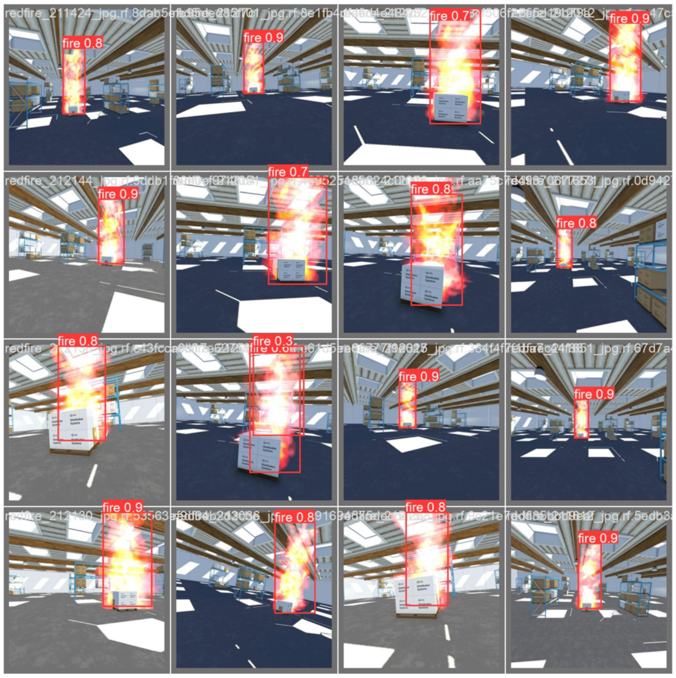

Fig. 6. Output detection results for a matrix of input test images.

in limited cases. This evaluation was qualitative and small scale but provides encouraging evidence for synthetic-to-real transfer.

D. Qualitative Evaluation

Qualitative inference on test images confirmed that YOLOv5 produced bounding boxes aligned with flame regions under diverse conditions:

- Robust detection across varying camera distances and oblique angles
- Reliable performance under daytime and fluorescent lighting
- Reduced confidence in cases of small or partially occluded flames

Representative detection outputs from the test set are shown in Fig. 6, where bounding boxes localize fire sources across synthetic warehouse views.

E. Summary of Quantitative Results

Table I summarizes the key quantitative findings from classification and detection experiments.

TABLE I
SUMMARY OF CLASSIFICATION AND DETECTION RESULTS ON SYNTHETIC TEST DATA

Task	Metric	Value	Notes
Class. (TinyVGG)	Acc.	96%	Min. overfit. > 20 epochs
Detect. (YOLOv5)	Prec.	0.90	Red fire class
	Recall	0.85	Validation average
	mAP@0.5	0.88	Synthetic validation
	mAP@0.5:0.95	0.62	Sensitive to scale and occl.

VI. Discussion

The results demonstrate that synthetic data generated in Unity can provide a viable foundation for training early-stage fire detection models. Classification with TinyVGG reached about 96% test accuracy, showing that lightweight CNNs are sufficient for distinguishing between no fire, red fire, and blue fire classes under varied synthetic conditions. Detection experiments using YOLOv5 achieved mAP@0.5 scores near 0.88, indicating that synthetic imagery can support robust bounding-box localization of flames in diverse environments.

These findings align with prior work on domain randomization in robotics and computer vision, where procedural variation in texture, lighting, and geometry narrows the simulation-to-reality gap. The synthetic pipeline used here follows similar principles by altering camera pose, illumination, and flame appearance, ensuring that models encounter varied edge cases that would be difficult to reproduce safely with real fires. The strong synthetic-to-synthetic performance suggests that synthetic imagery is not only a practical alternative to real fire datasets but also an effective means to prototype detection models before real-world deployment.

In addition to synthetic experiments, a live demo during the poster session confirmed that the YOLOv5 detector generalized to real fire images. The model successfully identified and classified flames in 10 consecutive real-world test images. Although preliminary and limited in scope, this result supports the hypothesis that domain-randomized synthetic data can bootstrap models with transferable features for real-world fire detection tasks.

At the same time, qualitative inspection of detection results highlights sensitivity to small or occluded flames and reduced performance under low-light conditions. These issues point to limitations in current synthetic data rendering, particularly the modeling of emissive transparency and smoke dynamics, which may reduce realism in critical edge cases.

VII. Limitations and Future Work

Although this study demonstrates feasibility, several limitations remain:

- **Limited real-world validation:** Real fire images were only tested qualitatively in a live demo (10 images correctly classified). While promising, this validation is not sufficient for statistical assessment, and larger-scale benchmarking is required.
- **Scale of dataset:** The dataset was small (about 240 images per class), limiting the statistical robustness of reported accuracies.
- **Flame modeling constraints:** Unity's particle systems captured dynamic flame behavior but lacked detailed smoke and emissivity models, which may limit realism.
- **Limited spectrum of conditions:** While camera and lighting variation were included, more complex environments such as cluttered storage and multi-source fires were not modeled.

Future work will focus on:

- Expanding real fire validation with larger datasets and systematic benchmarking of synthetic-to-real transfer
- Increasing dataset size through programmatic generation of thousands of varied fire instances
- Introducing domain adaptation methods, such as fine-tuning synthetic-trained models on limited real data
- Enhancing realism by simulating smoke dynamics, emissivity, and thermal effects in the rendering pipeline
- Evaluating additional architectures, for example transformer-based detectors, for improved robustness under occlusion and noise

VIII. Conclusion

This preliminary study demonstrates the feasibility of generating synthetic fire imagery for industrial and warehouse environments using a Unity-based digital twin pipeline. Particle-system fire effects, combined with systematic variation of camera pose, lighting, and scene conditions, enabled the construction of a diverse dataset without the safety, cost, and ethical constraints of real fire collection.

Experiments with TinyVGG showed that synthetic data alone can yield high classification accuracy (about 96%) across multiple fire classes. Similarly, YOLOv5 achieved strong detection performance on the synthetic test set (mAP@0.5 about 0.88), highlighting the utility of procedurally generated fire datasets for training object detectors. The preliminary live demo with 10 real images provides initial evidence that synthetic-to-real transfer is achievable.

Future work will augment the dataset with real fire events, explore transfer learning and domain adaptation strategies, and systematically benchmark across mixed synthetic and real datasets. By documenting this Dec 2023 pipeline and experiment set, the present work establishes provenance and creates a reproducible reference point for subsequent synthetic-to-real research in fire detection.

Provenance Appendix

- **GitHub release:** Repository publicly available since Dec 18, 2023: https://github.com/Ohara124c41/Fire-Detection-with-Synthetic-Data-Generation
- **Poster presentation:** Poster publicly presented Jan 9, 2024, photograph with EXIF metadata confirms display date

Acknowledgment

The author thanks open source developers and tool providers who made Unity and YOLO pipelines accessible.

References

[1] National Fire Protection Association, "Industrial and manufacturing property fires research fact sheet," 2018, based on 2017–2021 NFPA research. [Online]. Available: https://www.nfpa.org/downloadable-resources/fact-sheets/industrial-and-manufacturing-property-fires-research-fact-sheet-2018

[2] ——, "Warehouse structure fires," 2023, estimates for warehouse structure fires (2018–2022). [Online]. Available: https://www.nfpa.org/News-and-Research/Data-research-and-tools/US-Fire-Problem/Warehouse-structure-fires

[3] J. Tobin *et al.*, "Domain randomization for transferring deep neural networks from simulation to the real world," *arXiv preprint arXiv:1703.06907*, 2017.

[4] S. Z. Valtchev, "Domain randomization for neural network classification," *Journal of Big Data*, vol. 8, no. 1, p. 75, 2021.

[5] A. Khondaker, A. Khandaker, and J. Uddin, "Computer vision-based early fire detection using enhanced chromatic segmentation and optical flow analysis technique," *International Arab Journal of Information Technology*, vol. 17, no. 6, pp. 973–982, 2020.

[6] W. Thomson, N. Bhowmik, and T. P. Breckon, "Efficient and compact convolutional neural network architectures for non-temporal real-time fire detection," *arXiv preprint arXiv:2010.08833*, 2020. [Online]. Available: https://arxiv.org/abs/2010.08833

[7] P. Chattopadhyay, B. Goyal, B. Ecsedi, V. Prabhu, and J. Hoffman, "Augcal: Improving sim2real adaptation by uncertainty calibration on augmented synthetic images," *arXiv preprint arXiv:2312.06106*, 2023. [Online]. Available: https://arxiv.org/abs/2312.06106

[8] Y. Patel, "Tinyvgg: Educational pytorch cnn implementation," https://github.com/ypatel789/TinyVGG, 2021, accessed 2025-10-02.

[9] S. G. Muller and F. Hutter, "Trivialaugment: Tuning-free yet state-of-the-art data augmentation," in *ICCV*, 2021, pp. 774–782. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2021/html/Muller_TrivialAugment_Tuning-Free_Yet_State-of-the-Art_Data_Augmentation_ICCV_2021_paper.html

[10] TorchVision Contributors, "torchvision.transforms.trivialaugmentwide," https://pytorch.org/vision/stable/generated/torchvision.transforms.TrivialAugmentWide.html, 2023, accessed 2025-10-02.

[11] G. Jocher, "Ultralytics yolov5," 2020. [Online]. Available: https://github.com/ultralytics/yolov5