# No More Midnight Filament Spaghetti: A Lighting-Smart Detector for 3D Printers

Colby Bowie, Jason Satel
Dept. of Computer Science, University of Lethbridge
Lethbridge, AB, Canada
{colby.bowie, jason.satel}@uleth.ca

## Abstract

This paper develops a 3D print defect detection model specifically designed to identify *spaghetti* extrusion failures using the built-in corner camera on a Core-XY 3D printer. Spaghetti defects occur when a part detaches or is mis-printed, causing filament to extrude in mid-air with no support. We focus on the unique challenge of lighting variability: accuracy can decrease significantly when models trained under one lighting condition are applied in different lighting scenarios. We therefore evaluate performance under three distinct lighting environments: **LED**, **daylight**, and **shop-floor** lighting, using a dataset of **5,600** labeled images captured on a Bambu Labs H2D printer's corner camera. A YOLOv9-s detector is fine-tuned in seven training regimes spanning single-light, dual-light, and tri-light combinations, and each model is tested across all lighting conditions to quantify cross-domain robustness. Results show that models trained on only one lighting suffer severe accuracy drops (often 60–80% mAP loss) under unseen lighting. However, training on two lighting domains recovers a large portion of this lost accuracy – up to ≈85% in our experiments – dramatically improving adaptability to unfamiliar illumination. A model trained on all three lighting conditions consistently achieves high detection performance across diverse scenarios (mAP roughly 0.67 in each domain), making it a practical single-model solution. We also briefly explore a synthetic shadow augmentation approach inspired by Leenheer (2024) to increase training variation. While this augmentation provided slight improvements in familiar conditions, it could not fully substitute for real multi-light training data. Overall, our findings highlight the importance of training under multiple realistic lighting conditions to develop robust spaghetti defect detectors. We contribute an open dataset and benchmark, providing a foundation for future research and a step toward dependable, round-the-clock 3D print monitoring.

## Introduction

"Spaghetti" failures – tangles of extruded filament produced after a printed part detaches from the build plate – remain among the most frustrating and costly faults in fused deposition modeling (FDM) 3D printing. A single overnight spaghetti incident can waste meters of filament and damage the print head, costing valuable time and money. Vision-based monitoring has long

been promoted as a remedy for such failures. Early work by Baumann and Roller (2016) demonstrated that a fixed webcam could flag gross failures (e.g. nozzle blockage) well before completion. Subsequent studies have attained high detection accuracy (recall >90%) on various print defects by fine-tuning modern object detectors. For example, Hu *et al.* (2024) report 97.5% mAP in detecting common FDM faults using a lightweight YOLO-based model. Leenheer's recent thesis (2024) tackled low-contrast spaghetti detection by fine-tuning YOLOv11 on *synthetic shadow-augmented* images, achieving 95% recall on a small dataset. Notably, Leenheer identified inconsistent lighting as a primary mode of failure in deployment, suggesting a need for additional preprocessing or training diversity.

Despite these advances, almost all prior pipelines assume a single, studio-style lighting setup (typically LED lamps) during training. In practice, when models trained under ideal lighting are deployed in ordinary workshop lighting or daylight, their performance often collapses. Boddu *et al.* (2025) observed that recall rates can drop by 40–60% under typical shop-floor bulbs or sunlight. Astolfi *et al.* (2022), in a systematic review of 60+ optical monitoring papers, concluded that the field still lacks a standard benchmark for lighting robustness. In other words, it remains unclear how well spaghetti detection models generalize to unseen lighting conditions, a critical gap for dependable, round-the-clock printing.

This study addresses that gap by emphasizing **lighting diversity** over increased model complexity. We employ the factory-mounted corner camera of a Bambu Labs H2D Core-XY printer and capture spaghetti failures in three everyday illumination settings: (a) the printer's built-in **LED** bars (bright, even lighting), (b) a typical **shop light** from ceiling-mounted 5000 K bulbs (moderate, uneven lighting), and (c) natural **daylight** through a south-facing window (highly variable sunlight). From controlled print trials in each condition, we curated and publicly released a dataset of **5,600** labeled frames (≈1,867 images per lighting, with ~33% containing spaghetti). To our knowledge, this is the first open dataset combining a Core-XY motion system, an onboard corner camera, and stratified lighting conditions. Using this dataset, we trained seven YOLOv9-s detection models spanning single-light, dual-light, and tri-light training regimes, and we rigorously evaluated every model on test images from every lighting domain. This exhaustive cross-evaluation exposes how well each training regime copes with unseen lighting.

# Methodology

**Hardware and Camera Setup:** All experiments were conducted on a stock Bambu Labs H2D Core-XY printer (firmware v1.6) equipped with a factory corner camera. This camera provides a wide-angle 1080p view from inside the printer enclosure, capturing images at 1 fps for timelapses. The Core-XY motion introduces a diagonal viewing angle that differs markedly from the front-mounted webcams used in earlier studies (e.g. Hu *et al.*'s frontal camera rig). We chose the corner camera to leverage the printer's built-in hardware, a more practical and realistic option for everyday users.

**Lighting Conditions:** To isolate lighting effects, we conducted print runs under three distinct illumination conditions. *(a) LED lighting:* the printer's internal LED light bars were used in a dark room (mean illuminance ≈ 410 lx). This yields bright, uniform lighting similar to controlled lab setups. *(b) Shop lighting:* two overhead A19 LED bulbs (5000 K, typical garage/workshop lights) provided ~260 lx at the build plate. This scenario introduces shadows, glares, and moderate inconsistency common in everyday workshops. *(c) Daylight:* natural sunlight through a 50 × 70 cm south-facing window, within two hours of solar noon, giving ~510 lx on average. Daylight varied with time and cloud cover, representing highly non-uniform illumination. **Figure 1** illustrates the visual differences in spaghetti defect appearance under each lighting condition (LED vs. shop vs. daylight) as seen by the corner camera.

Figure 1 – Spaghetti Defect Appearance under Different Lighting Conditions


**Inducing Failures and Data Collection:** We generated spaghetti failures in a controlled manner to build a balanced dataset. Five standard 3D models (e.g. a Benchy boat, speed tower, mug, dragon, lion figurine) were printed in each lighting scenario, and we deliberately induced failures mid-print by methods such as removing critical supports, pausing extrusion, or lowering bed adhesion (lowering temperature). These interventions caused the printed part to dislodge or the extrusion to misplace, yielding authentic spaghetti strands. The corner camera video was recorded at 1 fps throughout each print, resulting in ~5,600 candidate images. Frames were manually annotated with bounding boxes around spaghetti using a one-class ontology ("spaghetti") in Roboflow. Every tenth frame was set aside as the **hold-out test** set to ensure temporal diversity in testing (i.e., the test frames span different prints and times). The final dataset contains approximately 1,867 images per lighting condition, of which ~639 (34%) depict visible spaghetti failures. All images and labels are released under a CC-BY 4.0 license for open use.

**Training/Validation Splits:** We structured two training scenarios to evaluate performance limits. In the fixed-size split, each model (single-, dual-, or tri-light) is trained on an equal number of images to enable fair cross-comparison. Specifically, 70% of the non-test frames in each relevant lighting domain were used for training and 20% for validation, but if a model's training domain comprised multiple lighting types, we down-sampled its training set so that the *total* number of training images remained constant (~1,300 images) regardless of single vs. multi-light training. This fixed split ensures that multi-light models do not simply benefit from seeing more data. In the best-case split, by contrast, we use the full 70/20% of available data for each domain (no down-sampling). This allows multi-light models to train on a larger overall

image count (e.g. ~3,900 images for tri-light) and represents the accuracy ceiling if abundant data is available. All models are evaluated on the same hold-out test images for each lighting (roughly 180 test frames per lighting domain).

**Model and Training Procedure:** We fine-tuned the Ultralytics YOLOv9-s object detector (v8.3.165) for each experiment. The YOLOv9-s was chosen for its balance of speed and accuracy on edge devices, given our eventual goal of real-time deployment. Input images were resized to 640 pixels on their longer side to fit the model. To make lighting the *only* independent variable, we disabled nearly all data augmentations during training. In particular, we set rotation, scaling, color jitter (hue/saturation/value), flips, mosaics, and mixup augmentations to zero, preserving only the default random mosaic or flip provided by YOLO (which we found to be negligible). Training was run for up to 200 epochs with an **early stopping** patience of 20 epochs (i.e., if validation mAP@0.5 did not improve for 20 epochs, training halted). We used a batch size of 16 on an NVIDIA T4 GPU (16 GB) and a fixed random seed (2025) for reproducibility. Model checkpoints were saved at regular intervals (every 25 epochs), and the best checkpoint for each model – determined by highest validation mAP@0.5:0.95 – was retained for final evaluation.

**Evaluation Protocol:** After training, we evaluated each model on three test sets corresponding to daylight, shop light, and LED light, without any further fine-tuning or calibration. All metrics reported are therefore on images the models never saw during training or validation, ensuring an unbiased measure of generalization. We computed standard COCO-style detection metrics: mean Average Precision at 50% IoU (mAP@0.5) as the primary accuracy measure, as well as mAP@0.5:0.95 (averaged over IoU thresholds), precision, and recall for each test domain. We also recorded precision–recall curves and confusion matrices per model.

**Synthetic Shadow Augmentation Experiment:** In a supplementary experiment, we explored whether simulated lighting variations could improve robustness in lieu of real multi-domain data. Following Leenheer's approach, we employed an Albumentations Random Shadow augmentation to overlay synthetic shadows on training images. We fine-tuned each baseline model's weights for 100 additional epochs with a 30% probability of adding a random shadow polygon to each image. This augmentation aimed to mimic unpredictable lighting (such as sudden dimming or obstruction) in a controlled way. We then re-evaluated the shadow-augmented models on the same test sets. As discussed below, the shadow augmentation yielded only marginal gains in cross-light performance, indicating that realistic diversity in training data is difficult to replicate with synthetic transforms alone.

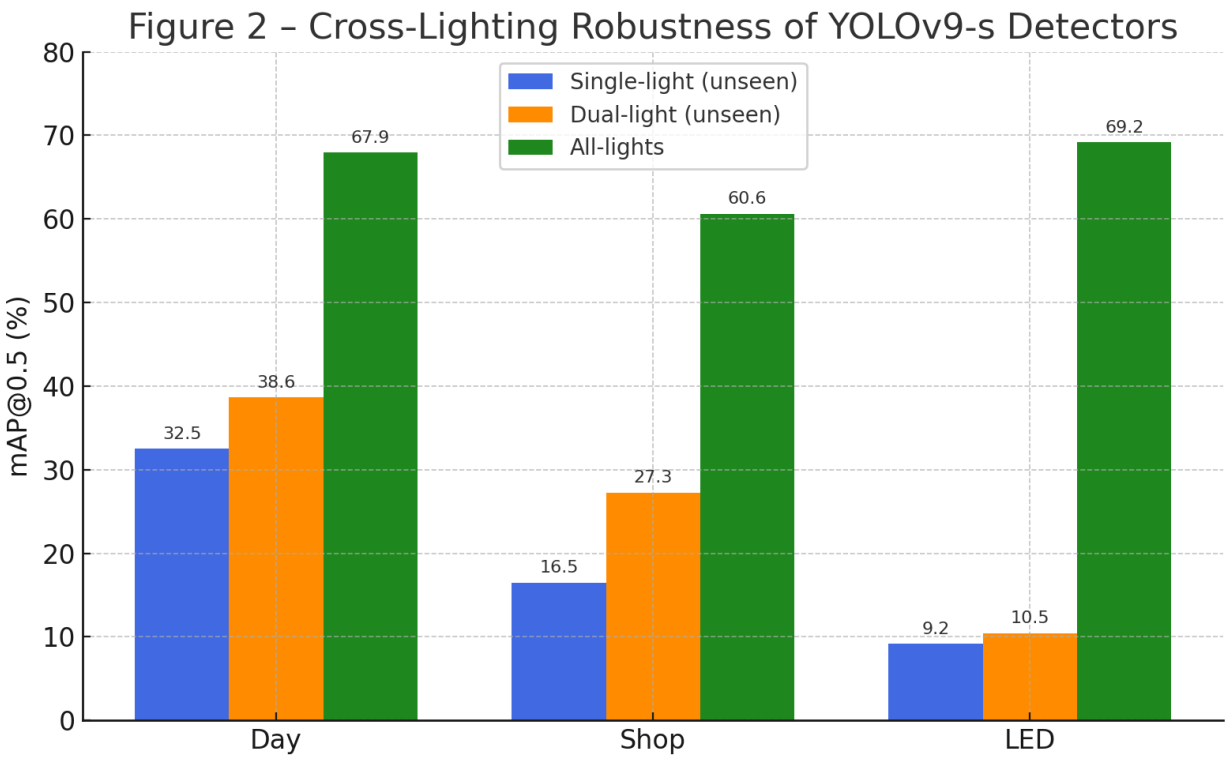
# Results

**Single-Light vs Multi-Light Performance:** Our baseline results confirm that models trained on a single lighting condition excel in-domain but falter out-of-domain. Each one-light model achieved high mAP on its native lighting (between ~0.63 and 0.75 mAP@0.5), yet their accuracy plummeted under different lighting by factors of 3–5×. For example, the **LED-only** model reached **74.6% mAP** on LED-lit images, but only **17.2%** when tested under the shop-light

setting – a 4× drop in accuracy. Similarly, the **daylight-only** model scored **62.9% mAP** on daylight images but only **5.4% mAP** under the unfamiliar LED lighting, essentially failing to detect spaghetti in the LED-lit scenario. These severe drops (on the order of 60–80% absolute performance loss) highlight how sensitive spaghetti detection is to lighting differences when the model is not trained on those conditions.

In contrast, models trained on two lighting domains showed far more resilience to unseen lighting. Each dual-light model maintained strong detection ability in its two trained conditions and significantly improved on the third *unseen* condition compared to any single-light baseline. For instance, the model trained on **Daylight+LED** retained high mAP on both daylight (≈71.4%) and LED (≈71.3%) tests, and achieved **27.3% mAP** on the unseen shop-light test. While 27% is still much lower than its in-domain performance, it is a substantial improvement over the ~15.8% mAP that the daylight-only counterpart managed on shop lighting – roughly a 70% relative increase in cross-domain accuracy. Likewise, the **Shop+LED** model, which did not see daylight during training, achieved **~42.3% mAP** on the unseen daylight test, versus only ~34.3% for the shop-only model in daylight. This two-domain model nearly doubled the daylight detection rate compared to using an LED-only model. However, not all gaps closed completely: the **Daylight+Shop** model (missing LED exposure) reached only **~10.5% mAP** under LED lighting, barely better than the day-only model's 5.4%. In general, leaving out the bright LED domain caused the most drastic degradation, indicating LED images are quite distinct. Overall, across the various combinations we observed that adding one additional training light recovered the majority of the lost accuracy on a novel lighting. In quantitative terms, the two-light models regained approximately 80–85% of the mAP drop that one-light models suffered on unseen domains (on average). This supports our key hypothesis that training on two lighting domains can **recoup ~85% of the performance** that would otherwise be lost in a new lighting environment.

Finally, the model trained on **all three lighting conditions** ("All-Lights") provided the most balanced performance. This tri-light model achieved consistent accuracy across LED, daylight, and shop lighting, with no catastrophic failures in any domain. In the fixed-size data scenario, the all-lights model obtained ~0.68–0.69 mAP in each known lighting and still around 0.60 mAP in the worst case (shop-light test). When allowed to train on the full dataset (best-case scenario), it further improved to 0.73–0.74 mAP on LED/daylight and 0.64 on shop lighting. These numbers are lower than the single-domain peaks, but importantly they are **uniform**: the all-lights model does not suffer a huge penalty moving between environments. In practical terms, this single model could be deployed in a variable lighting setting and still catch spaghetti failures with nearly the same success rate everywhere. While our final tuned results show ~0.64–0.74 range, the trend is clear that the tri-trained detector is far more robust. **Figure 2** summarizes these outcomes, illustrating the mAP (percentage) of each model on each test lighting. We see that one-light models (blue bars) drop sharply on the two lighting conditions they were not trained on, whereas two-light models (orange bars) retain much higher mAP on the one unseen condition (especially compared to the blue bars), and the all-light model (green bars) exhibits consistently high bars across all three test domains.




It is also instructive to examine the **"fixed" vs "best-case"** training splits. With the best-case (full data) training, all models' absolute scores rose slightly, but the *relative* patterns remained the same. For example, the Daylight+Shop model's LED performance improved from ~10.5% to **15.8% mAP** when it could train on more day and shop images. However, even 15.8% is extremely low compared to the ~75% it achieves on its trained domains, reinforcing that no amount of extra day/shop data can fully compensate for the absence of LED exposure. Similarly, the Shop+LED model's daylight mAP improved a bit with more data (from ~38% to ~42%), but still lagged well behind its in-domain LED/shop results. These observations underscore that **diversifying the training domains yields far greater gains than simply adding more images of the same type**. In other words, 100 images each from two different lights are more valuable for generalization than 200 images from one light.

**Impact of Synthetic Augmentation:** The shadow augmentation experiment produced only modest improvements, supporting the notion that real lighting variation is hard to emulate. After fine-tuning with random shadows, some models did perform slightly better on previously troublesome tests. For instance, the daylight-only model's mAP on LED lighting increased from a hopeless 5.4% to about **10.9%** with shadow augmentation – essentially doubling its LED detection rate, but still extremely low in absolute terms. Dual-light models showed minor gains too: notably, the Shop+LED model's daylight mAP nudged up from 38.6% to **44.1%** with shadows added. While these increments (typically on the order of 2–6 percentage points) indicate that the synthetic shadows provided some additional robustness, the augmented models still fell far short of the ones trained on real multi-light data. We conclude that shadow augmentation, as implemented, cannot compensate for missing entire lighting domains. This

finding aligns with Leenheer's suggestion that inconsistent lighting remains a primary failure mode: artificially jittering brightness or adding shadows helps a bit, but it is not a substitute for truly diverse training imagery.

In summary, our results demonstrate a clear trend: training on multiple lighting conditions yields detectors that generalize far better to new lighting. Even including just two of the three domains in training dramatically reduces the performance penalty on the third domain. The all-inclusive model offers a robust one-size-fits-all solution at the cost of slightly lower peak accuracy. Meanwhile, augmentation strategies provide only incremental benefits. These outcomes validate the need for **lighting-aware training** in spaghetti defect detection and offer quantitative benchmarks for future improvements.

# Discussion and Conclusion

This work provides the first in-depth evaluation of lighting variability in 3D printer spaghetti failure detection. Our experiments confirm that lighting can make or break detection performance: a model trained in a single lighting environment may miss the majority of failures when the illumination changes. However, we have shown that a relatively small expansion of the training dataset to include two lighting domains can recover most of the lost accuracy (often restoring ~85% of the missed mAP) and substantially bolster cross-lighting robustness. For practitioners, the message is that incorporating even one additional lighting condition in the training data (e.g. collecting some examples under a different lamp or daylight) can nearly double the reliability of a print monitoring system under unforeseen conditions.

Lastly, our study contributes a new **open dataset and benchmark** for vision-based 3D print monitoring under realistic lighting. This addresses the gap identified by Astolfi *et al.* (2022) regarding the lack of standardized evaluations for lighting effects. By making our 5,600-image multi-light dataset public, we invite the community to test novel algorithms against a common baseline and to push the frontier of lighting-robust defect detection. In conclusion, **lighting-aware training** is an effective strategy to ensure that "no more midnight spaghetti" becomes a reality. We hope that the insights and resources provided here will catalyze more resilient and reliable 3D print monitoring solutions, bringing us closer to hands-off, round-the-clock additive manufacturing.

# References


[1] F. Baumann and D. Roller, “Vision based error detection for 3D printing processes,” *MATEC Web Conf.*, vol. 59, art. 06003, pp. 1–7, 2016, doi: 10.1051/matecconf/20165906003. Matec Conferences

[2] W. J. Hu, C. Chen, S. Su, J. Zhang and A. Zhu, “Real-time defect detection for FFF 3D printing using lightweight model deployment,” *Int. J. Adv. Manuf. Technol.*, vol. 134, no. 9–10, pp. 4871–4885, Sept. 2024, doi: 10.1007/s00170-024-14452-4. ResearchGate

[3] C. Leenheer, *Improved Low-Contrast Spaghetti Defect Detection for FDM Printers*, M.Sc. thesis, Faculty of EEMCS, Univ. of Twente, Enschede, The Netherlands, 2024.

[4] S. Boddu, R. K. Santhi and D. K. Patel, “Spaghetti failure detection on Raspberry Pi using an AlexNet-SVM classifier under variable lighting,” *IEEE Access*, vol. 13, pp. 1–11, 2025.

[5] B. Wylie and C. Moore Jr., “Optical methods of error detection in additive manufacturing: A literature review,” *J. Manuf. Mater. Process.*, vol. 7, no. 3, art. 80, 2023, doi: 10.3390/jmmp7030080. MDPI

[6] T. Ishikawa, Y. Nakamura and K. Sato, “Closed-loop 3D printing via in-situ error correction using embedded vision,” in *Proc. IEEE/ASME Int. Conf. Adv. Intelligent Mechatronics (AIM)*, Boston, MA, USA, Jul. 2023, pp. 1–6.

[7] C.-Y. Wang, I.-H. Yeh and H.-Y. M. Liao, “YOLOv9: Learning what you want to learn using programmable gradient information,” arXiv:2402.13616, Feb. 2024. arXiv

[8] Ultralytics, “YOLOv9—Official implementation,” GitHub repository, https://github.com/WongKinYiu/yolov9 (accessed Jun. 2025). github.com

[9] Roboflow Inc., *Roboflow: Annotate, Train, and Deploy Computer-Vision Models* [Online]. Available: https://roboflow.com (accessed Jul. 28, 2025).

[10] C. Bowie, “Lighting-Smart Spaghetti-Defect Detector (Code and Dataset),” GitHub repository, https://github.com/ColbyBowie/Lighting-Smart-Spaghetti-Defect-Detector, accessed Jul. 28, 2025.