

MSEC2025-155444

ENHANCING IN-SITU MONITORING OF COOPERATIVE 3D PRINTING VIA EDGE DETECTION AND IMAGE AUGMENTATION

Harshin Sanam, Zhenghui Sha¹

Walker Department of Mechanical Engineering
The University of Texas at Austin
Austin, TX, USA

ABSTRACT

Cooperative 3D printing (C3DP) is an emerging field in Swarm Manufacturing (SM) that allows larger format objects to be printed in parallel for improved efficiency without compromising print quality. Our previous work in C3DP has presented a real-time process monitoring framework for C3DP, capable of detecting defects in fused deposition modeling (FDM) 3D printing. However, the annotation accuracy and image-matching scores in side-view error detection are too low to achieve effective closed-loop control. In this study, we improve the in-situ monitoring framework by integrating the Canny edge detector for enhanced image-matching accuracy. Also, we retrained the computer vision model based on YOLOv8 with over 7000 images utilizing image argumentation, achieving an increase of 25% compared to the 276 images used initially. With this improvement, the system has gained robustness, adaptability, and reliability across diverse camera perspectives and error conditions in real-world applications. Therefore, this research will show the improvement from the original model with a side-by-side in-situ analysis, demonstrating the increased effectiveness of real-time detection and adjustment in cooperative 3D printing. The results will highlight the potential of this enhanced system to be adapted across various applications in the additive manufacturing industry, ultimately moving toward autonomous, high-precision 3D printing systems in complex manufacturing environments.

Keywords: In-Situ Monitoring, Cooperative 3D Printing, Machine Learning, Error Detection

1. INTRODUCTION

Additive Manufacturing (AM) technologies have not only advanced in their applications but have also developed in their methodology. Single printer fused filament fabrication has existed for many years, and a big limitation to this manufacturing process is that the workflow efficiency is limited

by deposition rate and the print area (e.g., limited by one nozzle and printer size). Cooperative 3D printing addresses these issues as an emerging technology in the realm of Swarm Manufacturing by utilizing multiple mobile robotic printers on a fully open factory floor.

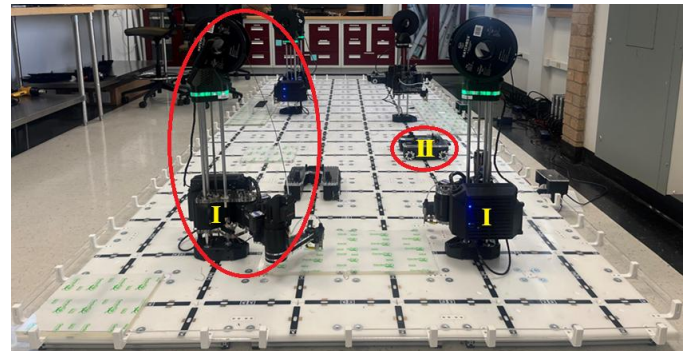


FIGURE 1: C3DP PLATFORM [1]. (I) SCARA PRINTING ROBOT AND (II) MOBILE TRANSPORTER ROBOT.

The C3DP system is a grid of tiles that have slots for printers and other robots. The printers utilize a Selective Compliance Articulated Robot Arm (SCARA) design that is secured to the slots on the floor and draws power from the nodes near each slot [1]. The current C3DP system has established methods for geometric partitioning, scheduling, placement, and path planning [2]. However, with the autonomy that AM and SM offer, and more specifically C3DP, there is a large need for in-situ monitoring of prints. To this end, preliminary progress was made in this field using an Intel RealSense Depth Camera D435. These cameras are screwed into a 3D-printed mount and can be placed in the slots located on the facility floor.

The system relies on dividing large objects into smaller, manageable sections that each printer can handle, a process

¹ Corresponding Author: zsha@austin.utexas.edu

called "chunking." There are two types of chunking: Z-chunking and XY-chunking [1, 3]. After the object is divided, each chunk is assigned a spot on the factory floor, and a schedule is created to ensure that the printers finish their parts as efficiently as possible [4]. Path planning is used to prevent collisions as the printers move around [5].

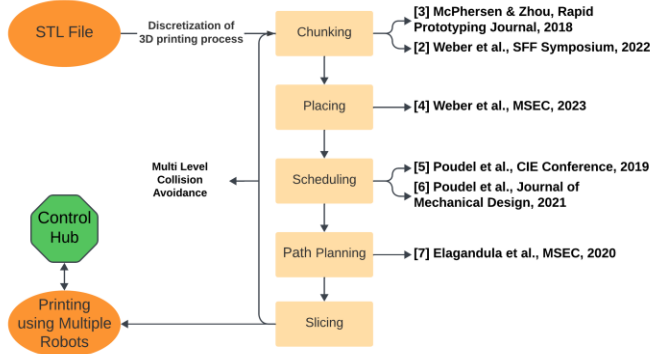


FIGURE 2: C3DP PROCESS FRAMEWORK [2]

The C3DP platform is managed through a control hub that can send commands to and receive data from the printers but currently operates in an open-loop system. Printers can report data such as nozzle temperature, expected position, and print status, while the control hub can send G-code commands to the printers. Although this setup provides operational control, there is no way to externally verify printer actions. This means that the actual nozzle position might differ from the expected position due to poor calibration or even collisions, potentially leading to print failures that the printer cannot detect on its own. To address this, we developed a monitoring framework for the C3DP system, which detects print failures and communicates with the control hub to halt faulty prints, preventing broader system failures.

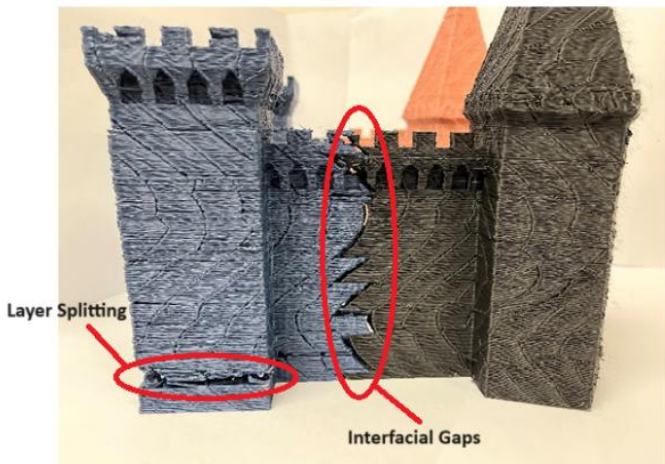


FIGURE 3: COMMON ISSUES IN C3DP

Despite using optimized parameters on the C3DP platform, there are still risks of failure, such as extrusion issues, bed miscalibration, warping, and stringing—issues we have encountered in our own tests. Figure 3 shows examples of common fused deposition modeling (FDM) defects, like layer splitting and interfacial gaps, which occurred during a cooperative test print of a castle due to poor extrusion and calibration. While FDM defects are generally manageable given the low material costs and print times, integrating FDM or advanced methods like C3DP into an automated factory setting will require a system to ensure accurate, error-free prints to maintain cost-efficiency and timely production.

The remaining sections are organized as follows: First, we will survey existing literature to validate current gaps seen in in-situ monitoring. Next, we will briefly discuss the existing 2D vision methods regarding the C3DP platform, which includes nozzle tracking and stringing detection. Then, we will detail our approach to implement image augmentation to enhance the stringing detection. Finally, we will finish the discussion with the image-matching improvement seen with edge detection.

2. RELEVANT LITERATURE

Process monitoring in additive manufacturing is an active research area due to the variability in part quality, which is a notable drawback of the process. This issue is especially critical for metal AM, where industries like aerospace require parts to meet strict quality standards that are challenging to verify with AM methods [7]. Consequently, extensive monitoring and control techniques are being explored to ensure that additively manufactured parts are defect-free and meet these high standards. For example, industrial solutions like EOS Smart Monitoring adjust parameters like laser power during builds to meet quality specifications [8].

For FDM printing, however, this level of quality assurance is generally unnecessary, as parts are typically not used in such demanding applications. Still, FDM has a higher failure rate, around 20% for unskilled users [9]. As a result, much research has been done in process monitoring for FDM printing, with 2D vision being the most explored method [10, 11]. For instance, deep learning methods with live camera capture have been used to track, detect, and correct extrusion errors from the nozzle [12, 13]. Like the setup used in this research, externally mounted cameras can watch the printed object and compare the image to the STL file [14]. Moreover, 2D vision techniques can be applied layer by layer and compared to the relevant step in the G-code [15]. These external cameras can detect warping, stringing, and other printing defects using machine learning methods [16, 17].

The biggest gap seen in process monitoring is the lack of accurate defect detection and in-situ capabilities. As stated, there are existing 2D vision techniques, and through trained models, these techniques can detect a variety of FDM defects. However, determining the fatality of an error is limited by the quality of the trained model. Previous machine learning methods for stringing and warping have proved ineffective on

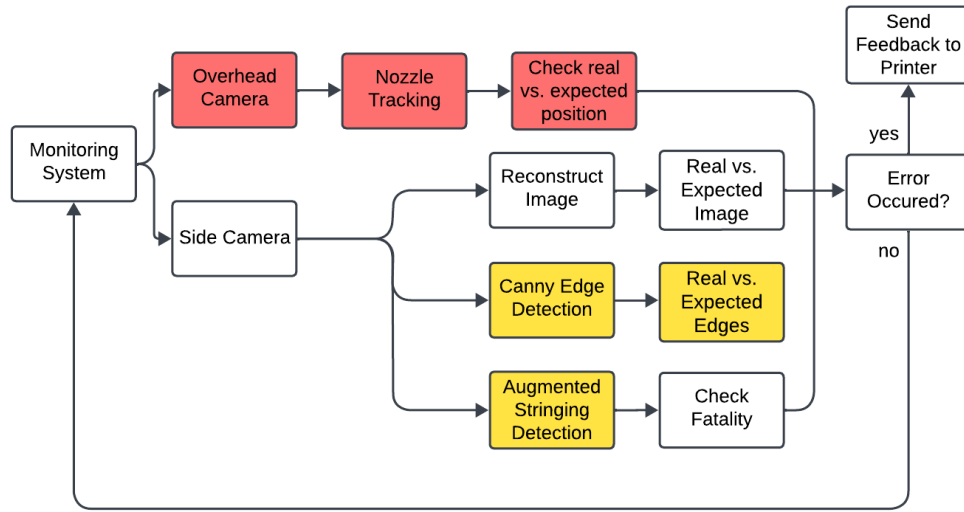


FIGURE 5: AN ENHANCED C3DP MONITORING FRAMEWORK

new datasets and in-situ monitoring, showing much lower levels of precision from the test data [17]. In our prior work, we attempted to fill the gap using a computer vision model, yet the performance was inadequate. In this study, we aim to address this using image augmentation and a more extensive dataset [16].

Another big gap in process monitoring is ineffective comparison algorithms for corners and edges [14]. This research aims to address this gap by implementing Canny edge detection because other edge detection methods use only one threshold value. In contrast, the Canny method uses hysteresis thresholding, as seen in Figure 4, where two values are used to filter strong and weak edges [18]. If the pixel gradient is above an upper threshold (e.g., maxVal), the edge is detected. If the pixel gradient is under a lower threshold (e.g., minVal), the edge is not detected. Finally, if the pixel gradient is between the two thresholds and if the adjacent pixel is an edge, the edge is detected [19]. Another effective edge detection method is using the Sobel operator, but literature has shown that Canny edge detection has been used for successful in-situ monitoring for Selective Laser Sintering AM to detect contours in a printed part [19].

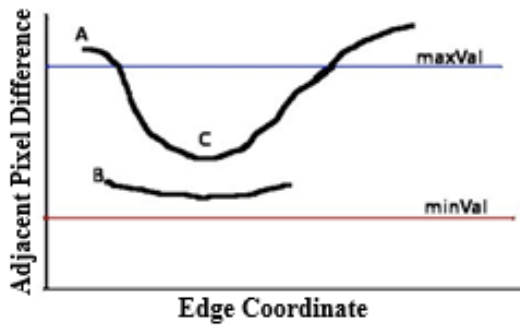


FIGURE 4: HYSTERESIS THRESHOLDING FOR CANNY EDGE DETECTION [19]

3. C3DP MONITORING FRAMEWORK

Before enhancing the monitoring accuracy, we first need to reevaluate the gaps in the original C3DP process monitoring framework. The original C3DP monitoring framework utilizes two cameras: one for nozzle tracking and the other one for image matching and stringing detection. The logic for the program is seen in Figure 5. By the end of the first iteration of this research, we realized that nozzle tracking proved ineffective, so we concluded only with the machine learning and comparator models used for stringing and image matching, respectively [20]. The nozzle tracking was ineffective because it utilized ArUco markers, and the computer vision detection has a delay causing inaccurate tracking of the nozzle. With closed-loop communication, the nozzle could be tracked purely of intrinsic data collection allowing for the monitoring framework to be simplified to only a side camera.

With that, we have outlined in Figure 5 the enhancements made to the framework by applying Canny edge detection and augmented stringing detection while discontinuing our nozzle tracking workflow. Regarding the machine learning model for string detection, we use the F1 score to quantitatively evaluate the performance of the training dataset (i.e., internal validity) and then use both unseen image data and newly generated data (i.e., external validity) based on our own C3DP system to qualitatively evaluate the model performance. The edge-matching algorithm follows the same methodology used for image matching by overlapping a pair of edge-enhanced images and calculating the similarity.

3.1 Image Matching

The image matching algorithm in the original framework was done by producing a 2D mask of the part from a stationary, standardized camera position and compared to the rendered mask from the STL using the Similarity Score Index Measure (SSIM) [21]. This methodology proved decently well, and we

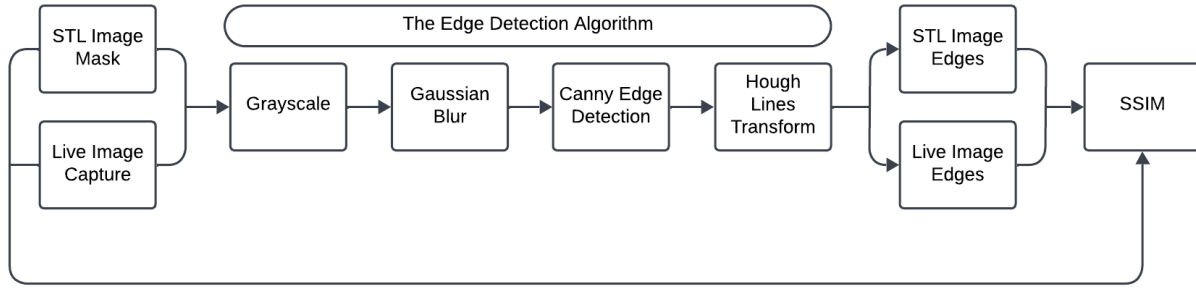


FIGURE 6: EDGE DETECTION IMAGE MATCHING ALGORITHM

chose to keep this algorithm in place. However, using just this algorithm caused near-perfect parts to consistently output around 65% image matching. This extremely large discrepancy is what impeded the first iteration of this research from real-world applications. As a result, we developed another layer of image processing and comparison via Canny edge detection to produce a more efficient image matching algorithm.

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (1)$$

μ_x the pixel sample mean of x ;
 μ_y the pixel sample mean of y ;
 σ_x the variance of x ;
 σ_y the variance of y ;
 σ_{xy} the covariance of x and y ;
 $c_1 = (0.01L)^2, c_2 = (0.03L)^2$;
 $L = 2^{Bits/Pixel} - 1$.

The methodology for this iteration of the monitoring framework also utilizes SSIM. It involves placing the edge-detected images over each other and determining the similarity measure. As shown in Figure 5, this will not replace the original SSIM scheme but rather enhance it, allowing for both overall image comparison and edge matching.

To enhance the image matching algorithm, Canny edge detection is implemented to bound any edges and compare them to the edges of the STL file. The algorithm is laid out in Figure 6, where both the image capture and images taken from the STL file are processed through Canny edge detection and then compared using the same SSIM model as the 2D image matching, as shown in Equation (1). The original model produces an SSIM of 0.6-0.8 for a nearly perfect printed part to the STL file 2D mask. As the first iteration of this research proved to work during in-situ analysis, this iteration is more focused on enhancing the pre-existing model. Therefore, the tests done for this research were done with already printed parts. The setup used for the analysis is shown below in Figure 7.

3.2 Stringing Detection

The original framework used a machine learning model to detect stringing. First, a public dataset from Roboflow with 276 images is annotated using a bounding box [22]. This dataset is then outputted in a YOLOv8 training model [23]. The dataset

was trained for 50 epochs. For the test dataset, the catastrophic fatality of the print due to stringing was well annotated and detected. However, this was not the case with the real data. If stringing is very light, it will still detect it and output a range of values with no specific trend in its behavior. This makes the program volatile in its data unless proper calibration is done with camera placement to the print.

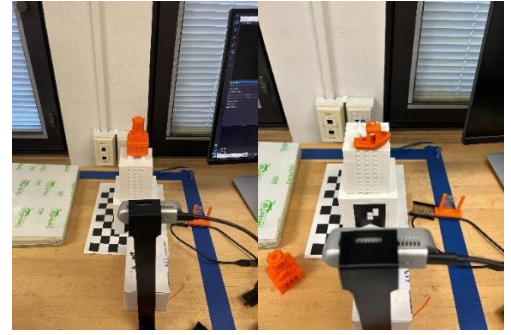


FIGURE 7: TEST SETUP WITH UT TOWER AND 3DBENCHY

To stabilize the data, allowing various camera orientations and distances to the part, we implemented image augmentation techniques in this iteration of the research [24]. Moreover, instead of the 276 images, a dataset of 1085 images, augmented into 7030 images, is used to enhance the monitoring framework. To test the two models, the training data is first examined, then tested on a new dataset, and some parts in our lab are done. The training data gives us insights into the strength of the model. The new dataset allows us to examine the quality of our model's annotations. Finally, the tests on varied levels of stringing help us understand the quality of our model's detections.

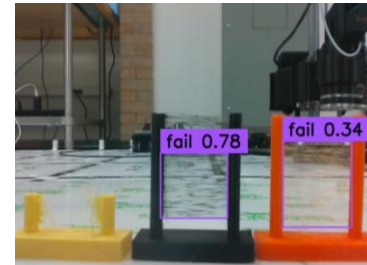


FIGURE 8: YOLOV8 STRINGING DETECTION

4. ENHANCEMENT VIA IMAGE AUGMENTATION

Stringing detection and nozzle tracking was improved by using image augmentation to enhance the accuracy of the machine-learning model. YOLOv8 was used as the machine learning model to produce the annotation model file. YOLOv8 is a computer vision model from Ultralytics that commonly uses bounding boxes to train a model given a dataset. There are better, more improved versions of YOLO annotation models, but YOLOv8 is a widely used and well-supported version [25]. The following sections show improvement from the strength of the existing model using the smaller dataset and the new model with image augmentation. After comparing the training data, both models are tested against a new model that does not cross over either training set. Finally, the models are tested in our lab setup on parts with low and fatal stringing.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

The F1 score is a predictor of the effectiveness of any machine learning or classification model [26]. It is determined, as seen in Equation (2), by multiplying the ratio of the product and the sum of precision and recall by two. Precision answers the following question: Of all the images the model predicted as stringing, how many were actually stringing? On the other hand, recall answers the following question: Of all the actual stringing images, how many did the model correctly identify as stringing? These both are determined using Equations (3) and (4), where TP is true positive, FP is false positive, and FN is false negative.

4.1 F1 Score Comparison

The F1 Confidence Curve shown in Figure 9 is for the original machine learning model used in the first iteration of this research—the F1 score peaks at 0.34 with a confidence threshold of 0.191. This process is done through a train-validation-test split, where there is training, testing, and valid data to perform the analysis. During training, YOLOv8 evaluates its performance on the validation set at each epoch. Through trial and error during the first iteration of this research, a confidence threshold of 0.5 was determined. However, it is seen that 0.191 is the optimal threshold given in the graph above. When this was selected as the threshold, we experienced too many false positives. By increasing the threshold to 0.5, we observed fewer false positives (higher precision) but also fewer true positives (lower recall).

The F1 Confidence Curve above in Figure 10 is for the new current model that utilizes image augmentation and a much larger dataset. The curve shows a large improvement in the model, maintaining a steady F1 score across a confidence level

of 0.8. By maintaining the F1 score across confidence levels, we are now able to minimize the number of false positives while still achieving the same number of true positives. The old model drastically reduced its F1 score after its peak at 0.19, whereas our current model can determine true positives at a higher level of confidence. We can now implement a higher confidence threshold than the original 0.5; in this case, a threshold of 0.75 is used.

It is important to note the F1 score in Figure 10 peaks at 0.47 for a confidence level of 0.328. Although this is better than the old model, we could still achieve a much better F1 score as it ranges from 0 to 1. This limited F1 score, even though a large dataset with image augmentation is used, could be due to improper annotations or augmentations done on the original dataset. There are many ways to improve this, and they will be discussed in Section 6.

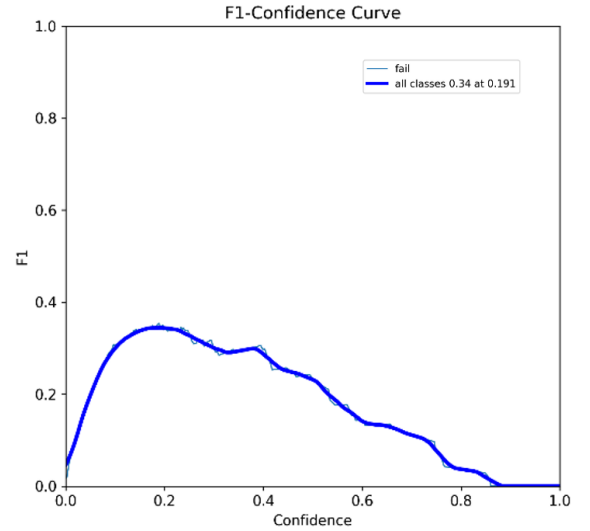


FIGURE 9: F1 CONFIDENCE CURVE FOR THE ORIGINAL MONITORING

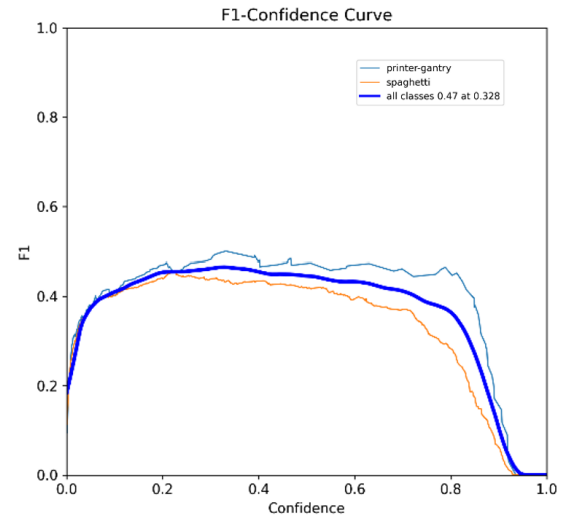


FIGURE 10: F1 CONFIDENCE CURVE FOR THE NEW IMPROVED MONITORING

4.2 Testing Data Results

Along with the training data analysis using the F1 score, we ran a test with both models on a completely new dataset. With this test, both the original model and the new model are applied to completely different publicly available testing data found on Roboflow. The dataset contains 98 images of fatal stringing scenarios. We selected this dataset because it will give us insights into the effectiveness of the models on data that has not crossed over with the training data.

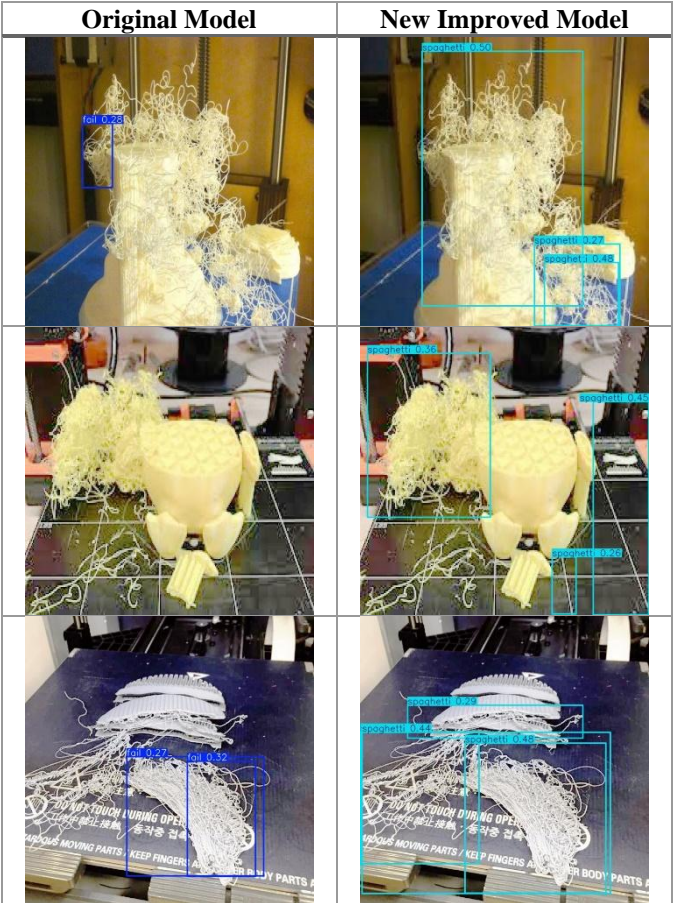


FIGURE 11: ORIGINAL VS. NEW MODEL TESTING DATA

In the dataset of 98 images, we selected three images to show in Figure 10. These three cases are very interesting and represent the major differences between the test data. The left side represents the images from the original model, and the right side represents the images from the new, improved model. The first pair of images represents the improvement of the bounding box annotation. While the original model predicted stringing, its location is only a small part of the entire defect area. On the second pair of images, the original model failed to detect stringing, while the improved model did a great job detecting the stringing in the back. However, the improved model annotated two cases of stringing where there is evidently no stringing. This could possibly be due to incorrect annotations

done on the training data or the model misinterpreting the lines on the print bed as stringing. Lastly, the third pair of images shows how, even in cases of multiple annotations, the original model fails to achieve the entire area, whereas the new model bounds all visible stringing areas.

4.3 Fatality Test Results

By applying image augmentation and a larger dataset to train a machine-learning model, we can now understand different levels of stringing. This is important because if different levels of stringing can be determined, then the fatality of the stringing can be determined. To test whether the new model can detect fatality, a 3DBenchy model with low, medium, and high stringing is tested [27]. These 3DBenchys are from our SCARA printers, which provide us with insights into possible issues in annotating parts from our printers. While the test data were all cases of extreme stringing, which gave us insights into the effectiveness of the annotating, this test gave us an understanding of its strength.

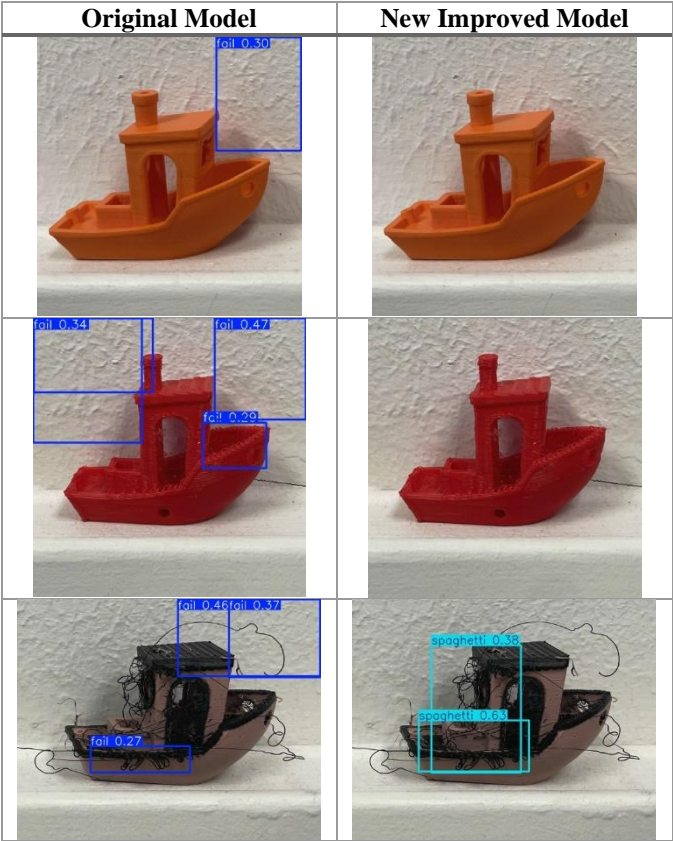


FIGURE 12: IMAGE MATCHING ALGORITHM

The orange 3DBenchy is a case of a near-perfect print with almost no stringing. We noticed that the original model incorrectly annotated the wall texture as stringing. This is like the red 3DBenchy, which represents a case of medium stringing. Both parts had no detections using the improved

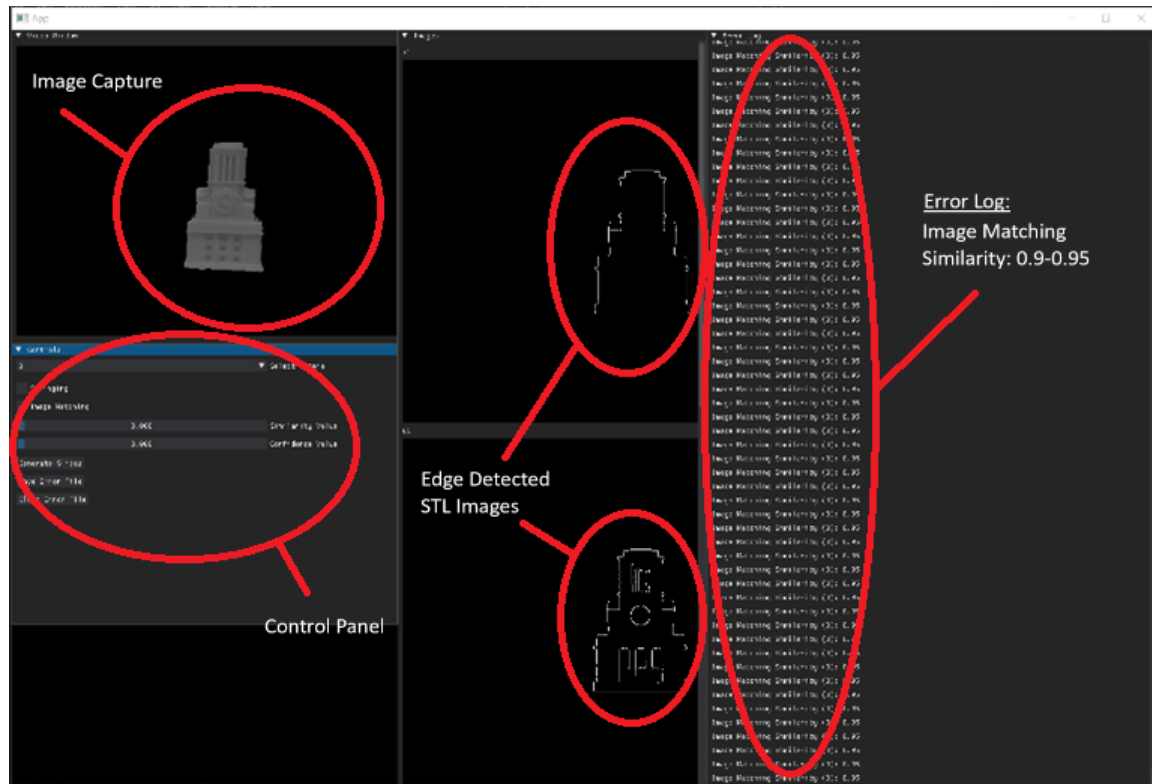


FIGURE 14: GUI WITH IMAGE CAPTURE OF UT TOWER AND EDGE DETECTED STL IMAGES ON THE RIGHT

model. This is acceptable for the orange low stringing part but is an interesting result in the red medium stringing part. Visually, the stringing in this part is very light and could have been the reason for the model not detecting it properly. Lastly, the highly stringed part is somewhat annotated in the original model but missing major areas of stringing. The improved model annotates the major areas of fatality of the print with a confidence of 0.63, which is much higher than the 0.27 annotation on the original model.

5. ENHANCEMENT VIA EDGE DETECTION

The edge detection enhances the SSIM calculated for image matching by using the algorithm outlined in Figure 6. By utilizing traditional edge detection, we achieve a 25% increase in SSIM. This is a big achievement as our previous research showed a max achievable score of around 0.7-0.8 with an average score of less than 0.7. Two parts are examined using the new, improved image-matching algorithm and then compared to the original algorithm. The results are summarized below in Figure 13 and further discussed in the subsequent subsections.

5.1 UT Tower

The UT Tower has a unique geometry in that it is a simple rectangular prism with an intricate clock tower design, allowing for more complex image matching. By having straight edges, a common computer vision technique called Hough Lines Transformation provides a straighter edge during edge

detection for the STL images and the live capture. The image-matching algorithm consistently produces an SSIM of 0.9-0.95, which is much better than the original model that produced a metric of about 0.65 for the same UT tower without anything moving.



	Image Mask	Image-only SSIM	Edge-Enhanced SSIM
UT Tower		0.65	0.93
3DBenchy		0.78	0.89

FIGURE 13: IMAGE MATCHING COMPARISON

4.3 3DBenchy

Similar to the UT Tower, the 3DBenchy consistently achieved an SSIM of .89-.90. This part was selected for two reasons. First, the original model computed this to have an SSIM of .78, the highest score seen with that model, so we wanted to see if that was the case. However, it is seen that the UT tower came out to be higher than this test. This could have been due to the extra enhancement of Hough Lines Transformation, giving a better straight edge, whereas the 3DBenchy is rounded and only has a straight edge around the

interior structure. This is the second reason this part was selected for analysis, as we wanted to see how edge detection performs purely through the Canny algorithm without the aid of other computer vision techniques such as Hough Lines Transformation.

6. CONCLUSION AND FURTHER WORK

In this work, we demonstrated the potential of computer vision to advance in-situ monitoring of C3DP. With a focus on integrating edge detection and image augmentation into the monitoring and detection system, the system performance improved a lot, bridging the gaps in in-situ error detection seen in the first iteration of this research. By utilizing image augmentation methods, we were able to enhance our machine learning model's robustness, preparing it to detect a wider range of potential stringing errors. Our results showed that with these augmentations, stringing is better detected, annotated, and bounded using a YOLOv8 model. The Canny edge detection algorithm was able to isolate boundaries of printed material more effectively. Moreover, it proved valuable in preprocessing the STL image masks and the image capture. In conclusion, image augmentation and edge detection effectively enhance detection accuracy and ensure consistent performance across print jobs.

The biggest limitation of the current work is that perfect image matching is still not achieved, and proper trial and error with the SSIM model could enhance this. Furthermore, testing needs to be done on various colors and backgrounds to see the consistency of the model. On the other hand, the stringing detection model still has some false positives seen in Figure 11. This can be enhanced by applying better annotations in the training data and using some of our data to continuously train the model. Moreover, the optimal F1 score improved from 0.34 to 0.47 using image augmentation and a larger dataset, but this is still not a great machine learning model F1 score as better models have scores close to 1.0. This is also a reason for false positives seen using the model, and reannotating could help improve the strength.

One of the biggest accomplishments in this work lies in our development of an interactive GUI that has successfully done in-situ analysis and monitoring for C3DP. Looking ahead, we believe several key directions can elevate this framework to the next level. First, we envision developing a more sophisticated integration with printer firmware to allow our system to control and receive feedback from individual printers on layer completion or material flow rates. This system stands out as a central hub that not only displays real-time print progress but should also enable active control of both the printers and cameras. Having all of this in one interface is the goal of our software and this research. This is the next step in this research. A preliminary closed-loop system based on a similar algorithm used to communicate between multiple printers is currently being worked on. With this implementation, our GUI should offer operators and engineers unprecedented control, providing actionable information at each stage of the printing process.

This would allow users to fully control their project offsite remotely as the printing errors are monitored and logged, giving the user a series of corrective options and then automatically restarting the print.

Finally, future work will also explore integrating more advanced AI models for predictive analytics, such as utilizing a more improved version of YOLO for image annotations and large language models for printers to understand the error received and the impact on the scope of the project. Our system could forecast and preempt potential errors before they manifest physically in the printed part. Such developments would further reduce the necessity for human intervention, moving the framework closer to fully autonomous C3DP. This vision aligns with the broader goal of establishing a cohesive, automated, and highly adaptable SWARM manufacturing system capable of accommodating the dynamic demands of modern additive manufacturing environments.

ACKNOWLEDGEMENTS

We want to acknowledge our senior design team, Travis Bouchard, Rafiq Hamzeh, Kyle Hodowany, and Manuel Salazar, for producing the first iteration of the GUI.

REFERENCES

- [1] Weber, D., Zhou, W., & Sha, Z. (2022). Z-Chunking for Cooperative 3D Printing of Large and Tall Objects. *Proceedings of the 33rd Annual International Solid Freeform Fabrication Symposium (SFF)*. <http://dx.doi.org/10.26153/tsw/44190>
- [2] Weber, D., Zhou, W., & Sha, Z. (2023). Job Placement for Cooperative 3D Printing. *Manufacturing Science and Engineering Conference (MSEC)*. <https://doi.org/10.1115/MSEC2023-104613>
- [3] McPherson, J., & Zhou, W. (2018). A chunk-based Slicer for cooperative 3D printing. *Rapid Prototyping Journal*, 24(9), 1436–1446. <https://doi.org/10.1108/rpj-07-2017-0150>
- [4] Poudel, L., Zhou, W., & Sha, Z. (2021). Resource-constrained scheduling for multi-robot cooperative three-dimensional printing. *Journal of Mechanical Design*, 143(7). <https://doi.org/10.1115/1.4050380>
- [5] Elagandula, S., Poudel, L., Sha, Z., & Zhou, W. (2020). Multi-robot path planning for cooperative 3D printing. *International Manufacturing Science and Engineering Conference (MSEC)*. <https://doi.org/10.1115/msec2020-8390>
- [6] Poudel, L., Sha, Z., & Zhou, W. (2018). Mechanical strength of chunk-based printed parts for cooperative 3D printing. *Procedia Manufacturing*, 26, 962–972. <https://doi.org/10.1016/j.promfg.2018.07.123>
- [7] Usha, S. (2021). In situ monitoring of Metal Additive Manufacturing Process: A Review. *Additive Manufacturing*, 275–299. <https://doi.org/10.1016/b978-0-12-822056-6.00007-2>

- [8] *Eos Smart Monitoring for 3D printing*. EOS. <https://www.eos.info/en-us/enabement/software/eos-smart-monitoring>
- [9] Wittbrodt, B. T., Glover, A. G., Laureto, J., Anzalone, G. C., Oppliger, D., Irwin, J. L., & Pearce, J. M. (2013). Life-Cycle Economic Analysis of distributed manufacturing with open-source 3-D printers. *Mechatronics*, 23(6), 713–726. <https://doi.org/10.1016/j.mechatronics.2013.06.002>
- [10] Fu, Y., Downey, A., Yuan, L., Pratt, A., & Balogun, Y. (2021). In situ monitoring for fused Filament Fabrication Process: A Review. *Additive Manufacturing*, 38, 101749. <https://doi.org/10.1016/j.addma.2020.101749>
- [11] Oleff, A., Küster, B., Stonis, M., & Overmeyer, L. (2021). Process monitoring for material extrusion additive manufacturing: A state-of-the-art review. *Progress in Additive Manufacturing*, 6(4), 705–730. <https://doi.org/10.1007/s40964-021-00192-4>
- [12] Brion, D. A., & Pattinson, S. W. (2022). Generalisable 3D printing error detection and correction via multi-head neural networks. *Nature Communications*, 13(1). <https://doi.org/10.1038/s41467-022-31985-y>
- [13] Jin, Z., Zhang, Z., & Gu, G. X. (2019). Autonomous in-situ correction of fused deposition modeling printers using computer vision and Deep Learning. *Manufacturing Letters*, 22, 11–15. <https://doi.org/10.1016/j.mfglet.2019.09.005>
- [14] Straub, J. (2015). Initial work on the characterization of Additive Manufacturing (3D printing) using software image analysis. *Machines*, 3(2), 55–71. <https://doi.org/10.3390/machines3020055>
- [15] Petsiuk, A. L., & Pearce, J. M. (2020). Open source computer vision-based layer-wise 3D printing analysis. *Additive Manufacturing*, 36, 101473. <https://doi.org/10.1016/j.addma.2020.101473>
- [16] Brion, D. A. J., Shen, M., & Pattinson, S. W. (2022). Automated recognition and correction of warp deformation in extrusion additive manufacturing. *Additive Manufacturing*, 56, 102838. <https://doi.org/10.1016/j.addma.2022.102838>
- [17] Paraskevoudis, K., Karayannis, P., & Koumoulos, E. P. (2020). Real-time 3D printing remote defect detection(stringing) with Computer Vision and artificial intelligence. *Processes*, 8(11), 1464. <https://doi.org/10.3390/pr8111464>
- [18] Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679–698. <https://doi.org/10.1109/TPAMI.1986.4767851>
- [19] Heintl, M., Schmitt, F. K., & Hausotte, T. (2018). In-situ contour detection for additive manufactured workpieces. *Procedia CIRP*, 74, 664–668. <https://doi.org/10.1016/j.procir.2018.08.051>
- [20] Mensch, C., Swaminathan, A., & Sha, Z. (2024). A real-time monitoring framework for cooperative 3D printing Proceedings of the 35rd Annual International Solid Freeform Fabrication Symposium (SFF). <https://utw10945.utweb.utexas.edu/2024-table-contents>
- [21] Nilsson, J & Akenine-Möller, T (2020). Understanding SSIM <https://doi.org/10.48550/arXiv.2006.13846>.
- [22] 3Dprinting Computer Vision Project Dataset. *Roboflow Universe*. <https://universe.roboflow.com/edge-detection-1e4oy/3dprinting-22xzg>
- [23] Ultralytics YOLOv8 Documentation Overview (2023). <https://docs.ultralytics.com/models/yolov8/>
- [24] Nagaraju, M., Chawla, P. & Kumar, N. (2022). Performance improvement of Deep Learning Models using image augmentation techniques. *Multimed Tools Appl* 81, 9177–9200 <https://doi.org/10.1007/s11042-021-11869-x>
- [25] Terven J, Córdova-Esparza D-M, Romero-González J-A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*. <https://doi.org/10.3390/make504008>
- [26] Hand, D.J., Christen, P. & Kirielle, N. (2021). F*: an interpretable transformation of the F-measure. *Mach Learn* 110, 451–456 <https://doi.org/10.1007/s10994-021-05964-1>
- [27] Dwamena, M. (2024) How to Improve 3D Printing Quality – 3D Benchy – Troubleshoot & FAQ. *3D Printerly* <https://3dprinterly.com/how-to-improve-3d-printing-benchy-troubleshooting-faq/>