BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2026
http://excel.fit.vutbr.cz

Monitoring of 3D Printing Using Machine Vision

Jakub Smička, Supervisor: Ing. Jakub Liška

Abstract
While Fused Deposition Modeling (FDM) 3D printing has revolutionized prototyping, it still suffers from unpredictable mechanical or extrusion failures, such as the spaghetti effect or visual inconsistencies. In shared laboratory environments, these issues are magnified because constant manual supervision is impossible, leading to wasted material and risks of printer damage. To address this, we developed a low-cost, edge-cloud monitoring system using a Raspberry Pi, a custom-trained YOLO classification model, and PyVista-based G-code geometric verification with custom-trained YOLO segmentation background removal. The YOLO model achieves a precision of 0.89 and effectively detects anomalies in real-time, automatically pausing the printer upon critical failure. The proposed solution directly benefits the VUT FIT Open Space shared laboratory by increasing print reliability and optimizing material usage. This is integrated with a Discord notification system, upgrading the current community server used by Open Space. The overall solution minimizes wasted filament, prevents hardware damage, and provides users with secure, role-based remote control directly via Discord without requiring any changes in the current workflow of Open Space or requiring expensive hardware.

xsmick00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

[Motivation] 3D printing is an essential prototyping tool, but it is highly prone to unpredictable mechanical or extrusion failures. In VUT FIT Open Space[1], users currently rely on passive remote monitoring via a simple live camera feed, which requires constant human attention to detect errors. This means errors are often noticed only after hours of wasted print time and filament. We need a system that actively watches the print for us.

[Problem definition] To address the lack of automated supervision, the required solution must ensure print failure detection, severity evaluation, and secure remote management.

The implementation must integrate seamlessly with future Prusa MK4(S) printers (planned upgrade), run reliably without disrupting the local network, and ensure that only authorized users can interact with the active print jobs to prevent accidental cancellations or unauthorized/forbidden actions.

[Existing solutions] Current solutions either involve passive monitoring or proprietary cloud services like Obico[2] or OctoEverywhere[3]. These commercial solutions are often closed-source, require paid subscriptions for high framerates, and do not natively support the custom Role-Based Access Control needed for a university laboratory environment. These solutions also rely exclusively on OctoPrint[4] plugins, which is not ideal for native printer usage. In academia, recent studies have demonstrated the high potential of YOLO models [1] and G-code visual analysis [2] for defect detection.

[Our solution] We built an edge-cloud architecture. A Raspberry Pi 5 acts as an edge node pulling data from the printer and camera, while a more powerful external PC/server handles the machine vision inference. Discord[5] was selected as the primary interface for student and user alerts because membership in the lab's Discord server is mandatory for accessing print services. Staff and students have permission to print only when they have joined this server. Building upon the theoretical groundwork laid by the aforementioned academic papers, we focused on implementing a prac-

[1] https://www.fit.vut.cz/study/open-space/.en

[2] https://www.obico.io/
[3] https://octoeverywhere.com/
[4] https://octoprint.org/
[5] https://discord.com/

tical, deployable system tailored for the production environment of the laboratory.

[Contributions] We created a custom-trained YOLO model highly optimized for the Open Space environment, an experimental G-code reverse-rendering verification module, and a secure Discord integration that maps print jobs directly to the users' logins.

2. System Architecture

As you can see in the Schema section on the poster, the workload is distributed to ensure scalability. The Edge Device (Raspberry Pi 5) acts as a lightweight client attached to the printer. It periodically pulls telemetry and G-code data via the PrusaLink API[6], captures images from the Raspberry Pi Camera Module 3, and sends this data to the server. The External PC/Server runs a FastAPI application that performs the actual inference asynchronously.

3. Dual-Path Anomaly Detection

The core of the analysis, illustrated in the Pipeline section, relies on two parallel detection methods to ensure high reliability.

1. Classification via YOLO Model

The primary detection layer uses the Ultralytics YOLO26 [3] (Small) architecture. As shown in the Results section, we trained the model on a custom dataset [4] of over 7,000 annotated images, combining public datasets with data gathered directly from the Open Space to ensure lighting consistency. More details can be seen on the poster. We intentionally tuned the training to favor precision over recall to minimize false-positive alerts, which would otherwise spam users.

2. Geometric Verification

While YOLO is excellent for identifying known visual defects, it might struggle with subtle geometric shifts or fallen structures. To solve this, we implemented a custom G-code reverse-rendering pipeline. Using the PyVista library, the server renders a 3D mask of the currently printed layers from the exact perspective of the physical camera. The physical image is segmented with a custom-trained segmentation model to remove the background, and both masks are compared. This provides a fallback mechanism that detects if the physical object is missing or significantly distorted. This approach is partly experimental because of hardware limitations, and the main focus was given to the YOLO classification. For this reason, users can toggle off this function if it provides bad results with dark filaments or frequent camera position changes. A manual script for render calibration is implemented to ensure that the PyVista rendering script has the best alignment with reality. Users can adjust the coordinates based on the output from the calibration script to improve the process.

[6] `https://help.prusa3d.com/article/prusa-connect-and-prusalink-explained_302608`

4. Decision Logic and Security

The raw outputs from both computer vision modules are processed by a decision algorithm. Cosmetic issues (like minor stringing) trigger only a warning on Discord. However, critical failures trigger an automatic printer pause to prevent hardware damage, followed by an immediate alert with a captured image - an example of this notification is on the poster Figure 3.

A key feature of the system is its security. The Discord bot extracts the `xlogin`[7] from the active G-code file name (it is mandatory to provide this xlogin in the G-code). Using Role-Based Access Control, the system ensures that:

- Notifications are pinged only to the actual owner of the print job.
- Commands (such as `/pause`, `/stop`) can only be executed by the print job owner or server administrators (with the given role).

5. Conclusions

The resulting system provides a robust, low-cost upgrade to the Open Space laboratory. The automated error detection combined with targeted Discord notifications successfully minimizes wasted material and saves users' time. The architecture is modular and ready to be scaled to monitor multiple printers simultaneously.

Acknowledgements

I would like to thank my supervisor Ing. Jakub Liška, for his guidance, valuable feedback, and support throughout the development of this project. I would also like to thank Mr. Zdeněk Juříček for his valuable knowledge about 3D printing and the opportunity to gather images for the dataset and testing environment.

References

[1] Abdul Rehman Sani, Ali Zolfagharian, and Abbas Z. Kouzani. Automated defects detection in extrusion 3D printing using YOLO models. *Journal of Intelligent Manufacturing*, 2026.

[7] xlogin is an unique ID given to every VUT FIT student

[2] Ala Eddine Ben Hammouda et al. Defect detection in additive manufacturing using image processing and G-code analysis. *Procedia Computer Science*, 2024.

[3] Ultralytics. YOLO26: State-of-the-Art Vision Models. Official Documentation, 2026. https://docs.ultralytics.com/.

[4] Jakub Smička. 3D Printing Failure Detection Dataset for OpenSpace. Roboflow, 2026. https://app.roboflow.com/bp3d/bp3d-online-openspace/7.