

# Continuous Data Collection of Under Extrusion in FDM 3D Printers for Deep-Learning Dataset

Muhammad Lut Liwauddin

Department of Electronic  
Faculty of Electrical and Electronic  
Engineering, Universiti Tun Hussein  
Onn Malaysia  
Johor, Malaysia  
muhdluth95@gmail.com

Mohammad Afif Ayob

Department of Electronic  
Faculty of Electrical and Electronic  
Engineering, Universiti Tun Hussein  
Onn Malaysia  
Johor, Malaysia  
afifayob@uthm.edu.my

Nurasyeera Rohaziat

Department of Electronic  
Faculty of Electrical and Electronic  
Engineering, Universiti Tun Hussein  
Onn Malaysia  
Johor, Malaysia  
nurasyeerarohaziat@gmail.com

**Abstract**— A shortcoming noted in fused deposition modelling (FDM) 3D printing technology refers to lack of intelligent monitoring and intervention during the printing process. Fail prints can still occur during the printing procedure even though the printer is of industrial grade and far more expensive than that of hobby grades. Under extrusion has been determined as one of the frequent failures in 3D printing. Such failure stems from insufficient extrusion rate and/or inadequate melting temperature of filament during the print. Under extrusion failure may result in undesired layer gaps, missing layers, unbalanced layers, and even holes in the printed models that would make the models completely unusable. Hence, an effective method that can reduce waste materials and overall costs is by integrating artificial intelligence (AI) into 3D printers. However, a large dataset is required prior to the training process of deep learning. Hence, this study proposes an automated and continuous data collection of under extrusion samples in FDM 3D printers using Raspberry Pi and webcam. As a result, adjustment of the G-code of the standard tessellation language (STL) models and repeated process of printing 3D models can effectively achieve the desired images.

**Keywords**— *deep-learning, under extrusion, 3D printer, fault detection*

## I. INTRODUCTION

The 3D printing technology of Fused Deposited Material (FDM) has progressed to such a point that it cannot only be made with several materials but demands the incorporation of a variety of technologies. During printing operations, nonetheless, defects in the printed model often go undetected. The printers actually continue their printing job while ignoring any defect on the printed model. Thus, quality control has remained a significant challenge in additive manufacturing (AM), which is also known as 3D printing. Detection of defect of any sort throughout the printing process can effectively minimise material and time wastage. Stages of printing may create an alarm to pause or stop the printing process so that corrective steps can be implemented to hinder reprinting of model parts.

A number of 3D printing equipment appears to lack a dedicated system for tracking and monitoring the printing process. Even if the filament has run out or there are potential faults in the print, 3D printers may continue to print the assigned part until all layers are completed [1].

Despite the apparent factor of FDM 3D printers, which can easily fabricate real parts within a shorter period at lower cost when compared to conventional manufacturing methods, they still fall short of any sort of intelligence required to detect and discern failures in the printed model during printing operations. These printers would execute the printing task

while completely ignoring any significant defect on the printed layout.

The material extrusion (ME) technique is one of the most extensively utilized 3D printing procedures, especially given the low-cost materials used. However, the ‘spaghetti-shape’ error that stems from filament tangling appears to be a prevalent issue with the ME process.

The occurrence of this problem demands a restart of the entire process, which consumes both time and materials [2]. Failure can occur due to misalignment of the print-bed and print-head, slippage of motors, warping of the printed material, lack of adhesion, and other many reasons [3].

A melt extrusion tractrix model was initiated based on the interaction of nozzle movement with an extruded filament on a support [4]. This model has displayed the best agreement with an actual extruded filament for a deposited straight line, circle, and arbitrary continuous curve.

In the worst-case scenario, print failures may affect and destroy the entire system; while under normal circumstances, the 3D printing process must be restarted, and failure materials are usually discarded. To prevent such issues, it is imminent for the human operator to oversee the printing process manually from time to time, signifying a burdensome role.

## II. RELATED WORK

The FDM AM technology has been vastly applied in recent years despite its numerous flaws that can affect the surface quality, accuracy, and even cause the parts to collapse [5]. As such, this present study is one of the first to present a multi-view and all-around vision detection method to identify defects on the outer surface of FDM additive processes. It also proposes a method to determine defects based on its laminate structure, while simultaneously introducing a mathematical matrix to represent the defects that can be used in quality assessment.

A low-cost, dependable real-time optical monitoring platform was proposed for open-source 3D printing based on fused filament fabrication [6]. The algorithms were tested for different 3D object geometries and filament colours. The outcomes revealed that for a wide range of 3D object geometries and filament colours, both algorithms with single and double camera systems were effective at detecting a clogged nozzle, incomplete project, and filament loss. The combined approach emerged as the most effective method with 100% failure detection rate.

Notably, AM enables the production of custom parts with previously impractical internal features, but it also introduces

the risk of internal defects due to print error or residual stress build up. Hence, a quality assurance system was proposed in [7] to monitor a part during the print process, to capture the geometry with 3D digital image correlation, and to compare the printed geometry with a computer model to detect print errors in real-time.

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

### III. METHODOLOGY

There are some ways to determine under extrusion failure manually for experimental purpose. Apparently, extrusion can yield missing layers, unbalanced layers, and layers with random dots and holes in the printed item. Hence, there are many variables to consider, such as G-code modification, to result in under extrusion printed samples:

- Temperature (hot bed & nozzle)
- Manual disturbance towards hot bed
- Flow percentage of filament
- Speed
- Grinded filament
- Clogged nozzle
- Filament diameter

The practical implementation for this design method provided here enables one to progressively describe every section of the print-path by defining the characteristics line by line including specifics about how the printer should work as it crosses each particular section. This particular design interprets the list of characteristics in the order in which they are defined by the user.

The G-code described for the first characteristic is generated before the next characteristic is assessed, which may produce a new G-code or adjust or replicate the G-code by the earlier characteristic.

### A. System Architecture

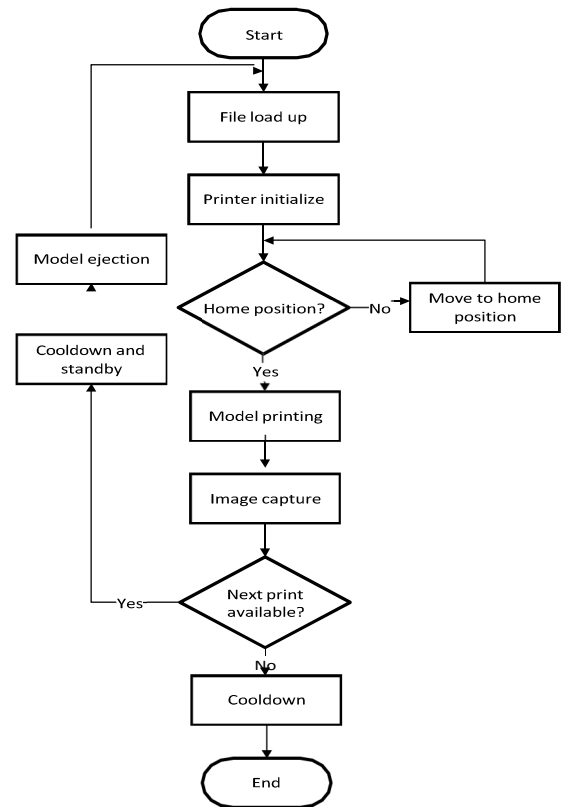


Fig. 1. New printing algorithm

Referring to Fig. 1, extruder heating can initiate a certain amount of heat based on user preferences before the printing process begins. Octolapse captures images during printing runtime until the end of printing instruction.

In continuous printing, ejection instruction is initiated before printing a new item, while the extruder returns to standby position and is set to home position.

### B. G-code introduction

G-code is a language that one uses to instruct machines on how to perform tasks. G-code is applied in 3D printing to move pieces within the printer. G-code is made up of g and m commands that each has a specific movement or action. Both G and M codes are included commands in the file to instruct the printer on how and where to extrude the material.

The sole distinction is that G-codes are generally understood by printers that use G-code, whereas M-codes are particular to each printer line.

```

Fan speed setting      ;Layer count: 25
Nozzle travel speed   ;LAYER:0
(without extrusion)   M107
                      G0 F9000 X52.235 Y55.800 Z0.300
Nozzle printing speed ;TYPE:SKIRT
(with extrusion)      G1 F2340 X56.093 Y55.800 E0.18815
                      G1 X56.346 Y55.605 E0.20373
                      G1 X57.299 Y55.078 E0.25684
                      G1 X58.540 Y54.758 E0.31934
                      G1 X59.404 Y54.719 E0.36152
                      G1 X60.320 Y53.688 E0.42878
  
```

Annotations in the image point to specific parts of the G-code: 'Fan speed setting' points to the first line, 'Nozzle travel speed (without extrusion)' points to 'M107', 'Nozzle printing speed (with extrusion)' points to 'G1 F2340', 'X, Y Coordinates' points to 'X56.093 Y55.800', 'Layer height' points to 'Z0.300', and 'Extrusion length' points to 'E0.18815'.

Fig. 2. G-code example

As illustrated in Fig. 2, the language is made up of numerous parameters. The most crucial line codes for this paper are M92, which can be used to set the steps per unit for one or more axes. This setting affects the number of steps that should be taken for each unit of movement. It enables the programming of steps per mm for motor drives.

```
M92 [E<steps>] [T<index>] [X<steps>] [Y<steps>] [Z<steps>]
```

Fig. 3. G-code M92 example

As presented in Fig. 3, this line code works closely with Esteps ('E' denotes the extruder and 'step' signifies the steps of the extruder stepper motor). In order to get the initial Esteps value, the 3D printer must be connected with the Pronterface software. Enter M503 command can print out all the current print settings saved in memory. If a setting has been changed since last saved, it may differ from the EEPROM content.

```
>>> m503
SENDING:M503
echo: G21 ;Units in mm
echo:Filament settings: Disabled
echo: M2000 D1.75
echo: M2000 D0
echo: Steps per unit:
echo: M92 X80.00 Y80.00 Z400.00 E93.00
echo:Maximum feedrates (units/s2):
echo M203 X500.00 Y500.00 Z5.00 E25.00
```

Fig. 4. G-code from EEPROM

Search for the line that begins with M92 and E values after steps per unit had been executed. As shown in Fig. 4, the Esteps value is 93. This value is crucial for under extrusion setting that can be adjusted based on the filament percentages that a user requires.

### C. Esteps Modification

Esteps should only be modified to assure dimensional precision, while flow rates should be adjusted to accommodate for material and model requirements. Physical movements, such as distance moved and filament extruded through the extruder, were applied to calibrate Esteps.

The volume of filament going through the extruder was affected by Esteps adjustment. In this paper, the volume of filament going through had reduced by 40% from its initial value to purposely cause Under Extrusion. The parameter for New *Esteps/mm* was calculated as follows:

$$\text{New Esteps/mm} = \text{Initial Esteps/mm} \left( \frac{40}{100} \right) \quad (1)$$

The initial Esteps calculated from the previous subchapter is 93 *steps/mm*. This indicates that the new Esteps value is 37 *steps/mm* based on (1).

### D. G-code Modification

```
G28 ;Home
M92 E93
G92 E0 ;Reset Extruder
G1 Z2.0 F3000 ;Move Z Axis up
G1 X10.1 Y20 Z0.28 F5000.0 ;Move to position
```

Fig. 5. G-code modification of Esteps value

With the latest value of Esteps, G-code was altered to a condition where the printed samples indicated Under Extrusion type of failure. Referring to Fig. 5, a user has to state the extrusion command and the initial value of Esteps under G28 G-code command, which will move the axes to their true zero position or home. This line can be used repeatedly to bring the initial volume of filament that passes through the extruder at certain layers.

```
;LAYER:40
M92 E37
:TYPE:SUPPORT
G1 F1500 E1314.69973
G1 X112.958 Y88.152 E1314.75999
```

Fig. 6. G-code of new Esteps value

At certain layers, the value of initial Esteps changed to the latest value of Esteps obtained recently. Based on Fig. 6, M92 E37 is noted at LAYER 40 of the G-code subsets. This indicates that the filament volume passing through the extruder was 40% from the initial value. This resulted in under extrusion condition at LAYER 40.

### E. Layer Alteration

Relapsed to previous section, the initial value of Esteps was introduced at the start of G-code under the homing command, G28. At LAYER 40, M92 E37 was introduced. This generated an under-extrusion condition beginning from LAYER 40 until a new command of initial Esteps value was initiated. In the absence of a new Estep command, the under-extrusion condition seemed to start from LAYER 40 until the last layer of the printed samples.

```
;LAYER:60
M92 E93
:TYPE:SUPPORT
G1 F1500 E2005.13315
G1 X112.983 Y88.126 E2005.1948
```

Fig. 7. G-code of new Esteps value

As illustrated in Fig. 7, the initial value of Esteps was initiated at LAYER 60. This generated layers at under extrusion and normal conditions. From this set of G-code modification, the printed samples were as follows:

- LAYERs 1 to 39 were normal layers
- LAYERs 40 to 59 were under extrusion layers
- LAYERs 60 until the end were normal layers

#### F. OctoPrint for Remote Control System

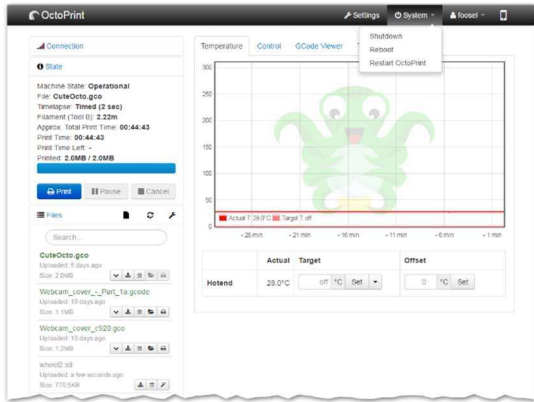


Fig. 8. OctoPrint dashboard

OctoPrint, as shown in Fig. 8, is a popular technique to transmit G-code commands to 3D printer and to observe the print status through a live video broadcast [8]. OctoPrint offers printer server services with a web interface that allows network access 24 hours a day and seven days a week [9].

To operate, all 3D printers require host software. The host software is in charge of sending commands to the 3D printer on how to construct an object [10]. OctoPrint uses a serial RS232 connection to communicate with the printer. One can upload already-sliced models (.gcode files) and select them for printing using online interface [8].

The following are some vital features of this software:

- Complete control over the printer, including axis movement, tool temperature, and extruder behaviour, among other things
- Observe the printing area in real-time
- 3D printing can be started, paused, resumed or stopped
- G-code commands are used to communicate with the 3D printer over serial communication
- Time-lapse generation that enables for the detection and correction of printing problems.

By implementing this instruction at the very last G-code of each print item, the algorithm lets the 3D printer in loop for printing until the user introduces the end printer instruction (see Fig. 9). This G-code initiates the ejection of the model by instructing the movement of nozzle at X, Y, and Z axes after the cooling down instruction of print bed to 30 °C.

```
G0 X0 Y235 Z15 ; STANDBY POS CORNER
M104 S65 ; HOTEND STANDBY TEMP
M190 R30 ; WAIT FOR BED TO COOL DOWN TO 30°C
G0 X94 F8000 ; MOVE X
G0 Z4 ; MOVE Z DOWN
G0 Y0 ; MOVE Y (EJECT)
G0 Z30 ; MOVE Z UP
G0 X0 Y234 Z15 ; COME BACK TO CORNER
M84 X Y E ; DISABLE ALL STEPPER BUT Z
M82 ; ABSOLUTE EXTRUSION MODE
```

Fig. 9. New Gcode for automated printing

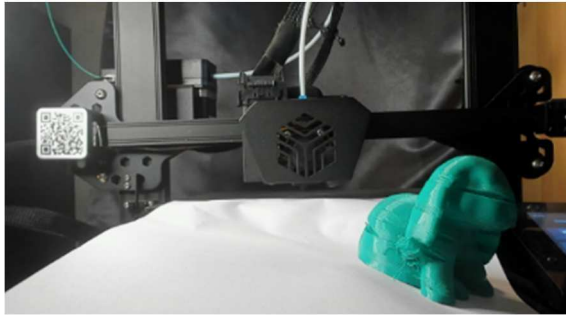
#### IV. RESULTS AND DISCUSSION



(a) Nozzle head centred



(b) Nozzle head at the back of the printed model



(c) Printed item pushed by the nozzle head

Fig. 10. Real-time view of automated printing with OctoPrint

In Figure 10, OctoPrint generated real-time view of printing algorithm and instruction by using web interface. The nozzle head was set to centre and lowered to the back of the model before start pushing in a positive manner for model ejection.

For the ejection task, the optimum temperature for the bed to cool down was room temperature. This is because; the material printed was closer to molten condition when the bed was warm. Both bed and model seemed to shrink as they cooled down at varied speed rates. In order to avoid error in ejection, print bed was cooled down to 30 °C.

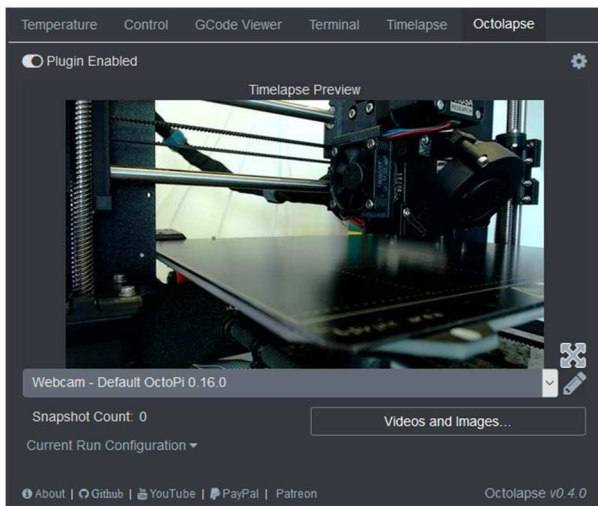


Fig. 11. Octolapse plugin

Referring to Figure 11, Octolapse, which serves as a plugin, had been included into the proposed system. This is a plugin that yields smooth time lapse video of the print, whereby the print head is moved to a different corner for each image gathered.

Additionally, this plugin allows the user to control what the printer does before beginning the next layer of printing. Such an ability enables one to capture images by having the printhead press a physical switch. In this system, Octolapse was deployed to create time lapse of printed item that monitors if every item is well printed.

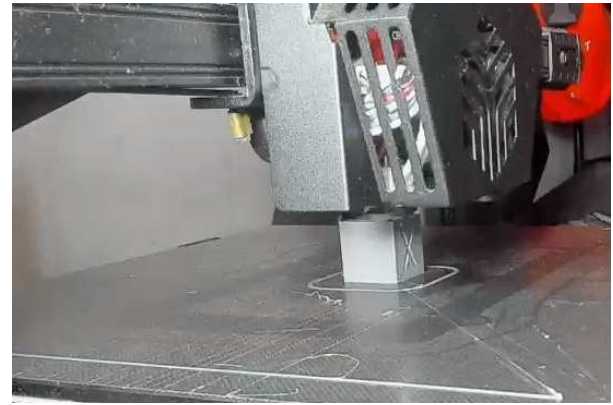


Fig. 12. Octolapse monitoring

Octolapse monitoring is illustrated in Figure 12. The print quality was affected by Octolapse plugin as it caused the printer to pause its printing job, move to a specific location to take a snapshot, and later resume printing. When set up incorrectly, the print quality could be adversely affected.

The results of G-code modification are presented in this part. This modification appears to be obvious in terms of layer surfaces and printed rate.

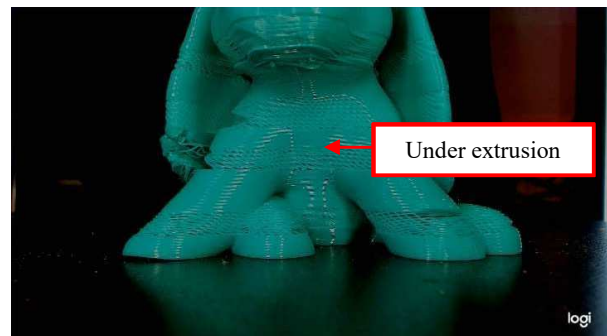


Fig. 13. Under extrusion result

As given in Figure 13, the variance between under extrusion layer and normal layer is vivid as the flow rate of the filament decreased to 30%. The visibility of under extrusion layer is crucial in this system to effectively and accurately identify fault. With a large margin of differences between the rates of filament, both normal and under extrusion layers were very fragile towards each other. The detachment between the layers seems possible if not handle with care.

## V. CONCLUSION

In this paper, an automated system 3D printer algorithm for under extrusion fault detection is proposed. Upon incorporating G-code modification, continuous printing, and monitoring plugin system; a foundation system is proposed for the automated printing fault detection system. Recent research work has revealed a wide range of methods for fault detection in FDM 3D printer. Upon considering both challenges and factors that could possibly affect the performance of the system, more experimental work is called for to generate a more efficient system. Overall, advances in FDM 3D printer fault detection systems have proven many positive outcomes.

Despite that, further investigation is in need to enhance the technology for broad applications in engineering of 3D printer.

#### ACKNOWLEDGMENT

This research was supported by Ministry of Higher Education (MOHE) through Fundamental Research Grant Scheme for Research Acculturation of Early Research (RACER/1/2019/TK04/UTHM//2).

#### REFERENCES

- [1] Delli, U., & Chang, S. (2018). Automated process monitoring in 3D printing using supervised machine learning. 26, 865–870. <https://doi.org/10.1016/j.promfg.2018.07.111>
- [2] Kim, C., Espalin, D., Cuaron, A., Perez, M. A., Macdonald, E., & Wicker, R. B. (2015). A study to detect a material deposition status in fused deposition modeling technology. 779–783.
- [3] Baumann, F., & Roller, D. (2016). Vision based error detection for 3D printing processes. MATEC Web of Conferences, 59, 3–9. <https://doi.org/10.1051/mateconf/20165906003>
- [4] Sun Q, Rizvi G, Bellehumeur C, Gu P. Effect of processing conditions on the bonding quality of FDM polymer filaments. Rapid Prototyping J 200;14:72-80
- [5] Shen, H., Sun, W., & Fu, J. (2019). Multi-view online vision detection based on robot fused deposit modeling 3D printing technology. Rapid Prototyping Journal.
- [6] Nuchitprasitchai, S., Roggemann, M.C., & Pearce, J.M. (2017). Factors effecting real-time optical monitoring of fused filament 3D printing. Progress in Additive Manufacturing, 2, 133-149.
- [7] Holzmond, O.B., & Li, X. (2017). In situ real time defect detection of 3D printed parts. Additive manufacturing, 17, 135-142.
- [8] Bas, J., Preston, H., & Moiseyev, L. (2020). OctoPlus: 3D Printing Error Detection System. In Design Review Report.
- [9] Luque, A. L., Rodríguez, J. M. J., Donoso, J. L. C., & Feito Higuera, F. R. (2018). Advances for 3D printing: Remote control system and multi-material solutions. Computer Science Research Notes, 2802(May), 160–163. <https://doi.org/10.24132/CSRN.2018.2802.20>
- [10] Shinde, Y., Madaki, R., & Nadaf, S. (2008). IoT based 3D Printer. International Research Journal of Engineering and Technology, 1015. [www.irjet.net](http://www.irjet.net)