AN INTELLIGENT PROGRAM TO MONITOR 3D PRINTING AND DETECT FAILURES USING COMPUTER VISION AND MACHINE LEARNING

Christine Li[1], Yujia Zhang[2]

[1]Sage Hill School, 20402 Newport Coast Dr, Newport Beach, CA 92657
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This paper proposes a novel solution for tracking the 3D printing process using an application that provides users with real-time updates on its progress [1]. The approach involves taking pictures of the 3D printer during the printing process, which are then analyzed by an AI model trained on thousands of labeled images to detect print failures [2]. The system is implemented using a Raspberry Pi and a camera, which capture images of the 3D printer and upload them to an online database [3]. The proposed application accesses this database to keep the user informed of the printer's current state, ensuring a seamless printing experience.

KEYWORDS

3D Printing, Machine Learning, Computer Vision, AI

1. INTRODUCTION

Hideo Kodama from Japan is one of the first people, if not the first person, who initialized the idea of 3D printing by brainstorming a layer-by-layer procedure to prototype an item [4]. Although he failed to patent it, many others expanded upon the idea after Kodama died; three new types of printers emerged and CAD tools like SolidWorks were popularized [5]. 3D printers and prosthetics became more readily available to the public by the 2010s and were embraced by the masses. 3D printing holds great potential; currently, it can even produce 3D printed meat, which has shown signs of health benefits like lowered risk of cardiovascular disease, and it can be a great option over killing more and more animals for food. 3D printing has a promising future as it can be applied to more efficient manufacturing, healthcare, and aspects beyond that, but there are still a multitude of issues that people have encountered during the process. Sometimes it can go wrong when a design is not made properly or when there is simply a printer malfunction. This not only wastes energy and filament, which damages the environment, but it also causes frustration for the user: starting the printing process over is strenuous, especially because they are losing double the money in order to print the object. 3D printing failures must be combated because it is starting to affect the world socially and environmentally [6]. More and more people are using and relying on 3D printers and increasing risks for failures is detrimental.

My project's solution is similar in these 3 methodologies. It also uses computer vision and machine learning to monitor 3D printing and detect failures. However, my project proposes an app-based interface for users to interact with the system and receive real-time updates on the printing process. This may help to improve the user experience by providing users with more

David C. Wyld et al. (Eds): ARIN, CSITA, ISPR, NBIOT, DMAP, MLCL, CONEDU -2023
pp. 147-155, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130713

accessible and user-friendly updates. Additionally, my project's solution may have different limitations and challenges than the methodology described above, depending on the specific AI model and camera technology used in the system [7].

My solution is to create an application that can track the 3D printing process and keep users updated on whether or not it is going smoothly. This is first done by taking pictures of the 3D printer while it's in use and notifying the user when a print failure occurs [8]. The AI model is trained with thousands of images, labeled either as successful or failed prints. A Raspberry Pi using a camera and AI is set up to monitor the 3D printer. The pictures taken by the Raspberry Pi of the 3D printer are then uploaded to an online database, which our app accesses and notifies the user of the current state of the printer.

The experiment aimed to test the accuracy of an AI program in detecting failures during 3D printing. The program was trained using a dataset of 1000 images, and the experiment involved setting up a camera to capture images of the printer at regular intervals while printing a pencil case. The experiment was conducted 90 times, resulting in 77 successful prints and 13 failed prints. The results indicate that while the AI was able to detect most failed prints, there is still room for improvement with further training and larger sample sizes.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Training the Model

Training the model could prove to be a challenge. When creating epochs, which are datasets, it is imperative that the epochs are not too big or too small. Although big epochs are able to update neural networks better, they are overfitting and require more time to train [9]. On the other hand, small epochs spend less time but are often underfitting. This means that it is important to find the proper amount of batches so that the model can train with a moderate amount of time and accuracy. In addition, I would also need to decide how many pictures to train and how many to test since I only had 1000. I could process my pictures into 5 epochs with 180 images, which could give me the result I wanted.

2.2. Accurately Detect Print Failures

Another challenge is to accurately detect print failures using computer vision and machine learning, it needs to be trained on a large dataset of labeled images. This dataset should include examples of successful prints as well as various types of print failures, such as under-extrusion, over-extrusion, warping, and so on. Without this training, the program may not be able to accurately recognize print failures, leading to false positives or false negatives. However, our lab has limited access to 3D printers, which can make it challenging to collect a diverse and large enough dataset for training the program. This limitation can significantly affect the accuracy of the program, as it needs a substantial amount of data to be trained effectively.

2.3. Focusing the Camera

Another problem I could face would be focusing the camera. The accuracy of my program heavily relies on how well it can view the 3D printer; if it perceives whatever is in front of it as too blurry, it might detect it incorrectly. Our camera has autofocus features but does not appear on the Raspberry Pi, so I would have to manually learn how to autofocus the camera by myself.

3. SOLUTION

The main structure of the program consists of three primary components: the AI model trained in Google Colab, the app designed in Visual Studio Code through Flutter, and the database, Firebase, to store information and pictures [10]. After collecting 1000 pictures of both successful and failed 3D prints, 900 was fed into an AI model to train it into differentiating which one was which. The remaining 100 pictures were shown to the model for it to detect whether the 3D printing was going well or not. Separately, an app was created in VSC, where a login page and status page were coded. Finally, a real time database in Firebase was made to link the first and second component together. All the information inputted, like the login username, and shown in the app was saved into the database. The pictures taken periodically were also kept there, named with different time stamps, and included the status in which the AI model determined it as and the link to the picture.

The entire program starts with the AI model, which was transferred to a Raspberry Pi connected to an Arduino camera to test it in real life. We ran the Raspberry Pi and the model on trials in which the 3D printer was printing accurately and also trials in which it malfunctioned and produced spaghetti messes. The photo taken by the camera at intervals was uploaded to the database, where the timestamp and status was also shown. Then from the database, the app also displayed the same information so that the user could easily open it up on their phone and see.

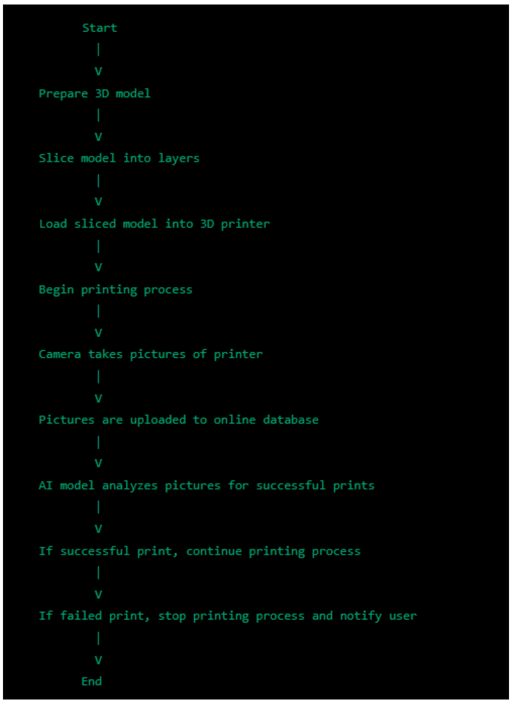

Figure 1. Overview of the model

The purpose of the AI model is to more accurately detect whether the print it is being tested is good or bad. I mainly used Google Colab to run my AI model and test it. The main concepts surrounding the model are machine learning, in which an application is able to make better predictions based on what it has seen previously, and computer vision, which is the ability to do those predictions through visuals, like pictures.

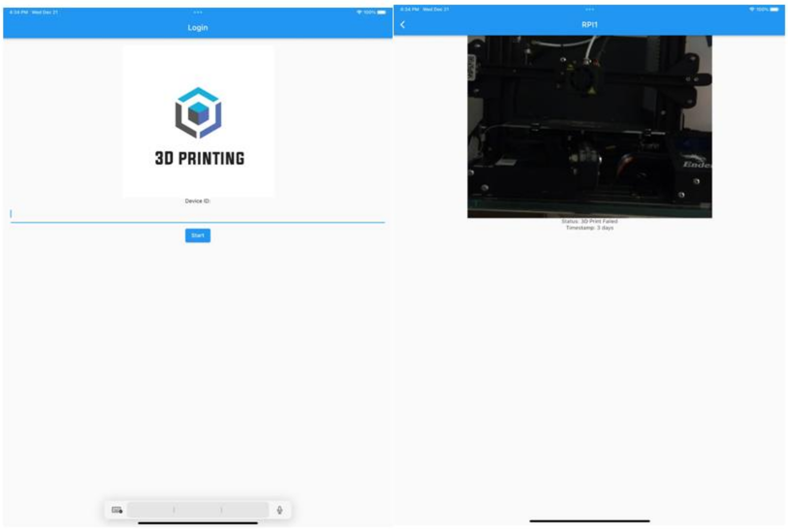

Figure 2. Screenshot of the app

```python
# load the model and give it the label and image, and then you predict

def predict(model_path,label_path,img_path):
  tflite_interpreter, input_details, output_details = load_tflite_model(model_path)
  labels = load_labels(label_path)
  img = load_image(img_path)

  numLabels = len(labels)

  # sets up our prediction
  tflite_interpreter.set_tensor(input_details[0]['index'], img)

  # runs it
  tflite_interpreter.invoke()

  # get the prediction results (not human readable results yet)
  predictions = tflite_interpreter.get_tensor(output_details[0]['index'])[0]

  # output details = predictions, get the first element of the predictions

  # make results readable

  # prediction isn't just a number, giving you guesses
  top_k_indices = np.argsort(predictions)[::-1][:numLabels]

  # getting the confidence in percentage of each label
  for i in range(numLabels):
    pred = predictions[top_k_indices[i]]/255.0
    pred = round(pred, 2)
    lbl = labels[top_k_indices[i]]
    print(lbl, "=", pred)
```

Figure 3. Screenshot of code 1

In this screenshot, the code programs the model to determine the status of a given 3D print. The code first loads up the model, gives it the label and image, and then predicts the status based on its confidence.

I designed my app using Flutter, a framework that produces mobile apps. Everything was coded in Visual Studio Code so that it could provide better access to content, like the pictures, statuses, and timestamps; instead of loading up the AI model or the database, everything could be reached with a few taps on the iPhone. The app is a place for the user to be able to see what is going on with the printer at the moment without having to search through hundreds of other images in a database.

```
    body: Padding(
      padding: const EdgeInsets.all(20.0),
      child: SingleChildScrollView(
        child: Column(
       children: <Widget>[
          //mainAxisAlignment: MainAxisAlignment.center,
          Image.asset(
            'assets/3d-printing-logo.png',
            height: 400,
            width: 400,
          ), // Image.asset
          const Text(
            'Device ID:',
          ), // Text
          TextField(
            controller: myController,
            autofocus: true,
          ), // TextField
          Padding(
            padding: const EdgeInsets.only(top: 10.0),
            child: ElevatedButton(
              onPressed: () {
                verify(context);
              },
              child: Text('Start')), // ElevatedButton
```

Figure 4. Screenshot of code 2

The real time database served as a place of storage for time stamps, the status it detected as, and of course, the photo taken at that time. It consisted of names of user IDs, which then had information about the Raspberry Pi and pictures when clicked. The pictures were labeled as the timestamps, which would be easy to locate and measure when the 3D print might have gone wrong.

4. EXPERIMENT

4.1. Experiment 1

A possible blind spot in the program that could be tested would be the AI's accuracy. Even though it read 1000 pictures, it could definitely be improved upon with more training.

We first made sure that the AI model and code was exported into the Raspberry Pi so that the camera could detect the 3D printer printing. The camera was set up right in front of the printer where it could clearly see what was going on and was programmed to take pictures at certain intervals (sometimes it was 5 seconds, sometimes it was one minute). Then, we ran experiments where the 3D printer was working properly and printing an actual object, a pencil case shaped like a castle. After that, we also ran tests where the 3D printer was just randomly printing without having a sense of what it was doing, which counted as the failed prints.

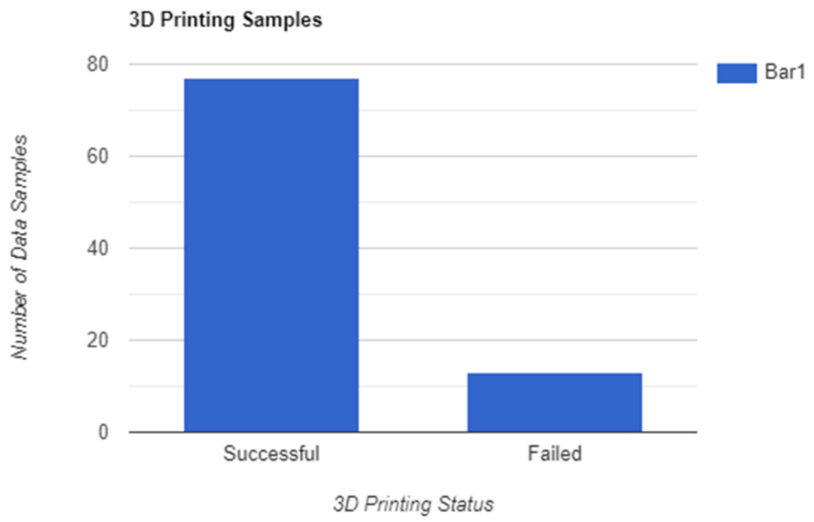

Figure 5. Graph of experiment 1

Based on the experiment conducted, it was found that the AI's accuracy in detecting print failures could be improved with more training. The experiment involved setting up a camera in front of the 3D printer and programming it to take pictures at regular intervals. The experiment was carried out in two phases - one where the 3D printer was printing an actual object and the other where the printer was randomly printing without any sense of direction. The latter phase was considered as the failed prints.

The experiment was run 90 times, out of which 77 were successful prints, while 13 were failed prints. The data suggests that the AI was able to detect a majority of the failed prints, but there is still room for improvement. It is worth noting that the small sample size may limit the generalizability of the findings.

4.2. Experiment 2

Another blind spot is the performance in detecting 3D printing failures may be different with different AI Models. The aim of this experiment is to compare the performance of two different AI models in detecting 3D printing failures.

The first model will be trained using a dataset of 100 images, while the second model will be trained using the same dataset of 100 images. The performance of the models will be evaluated based on their ability to accurately detect failed prints. Two AI models will be trained using the two different datasets. Both models will use computer vision and machine learning techniques to detect 3D printing failures. The models will be trained to classify images as either a successful print or a failed print. The performance of the two AI models will be evaluated based on their accuracy in detecting failed prints.

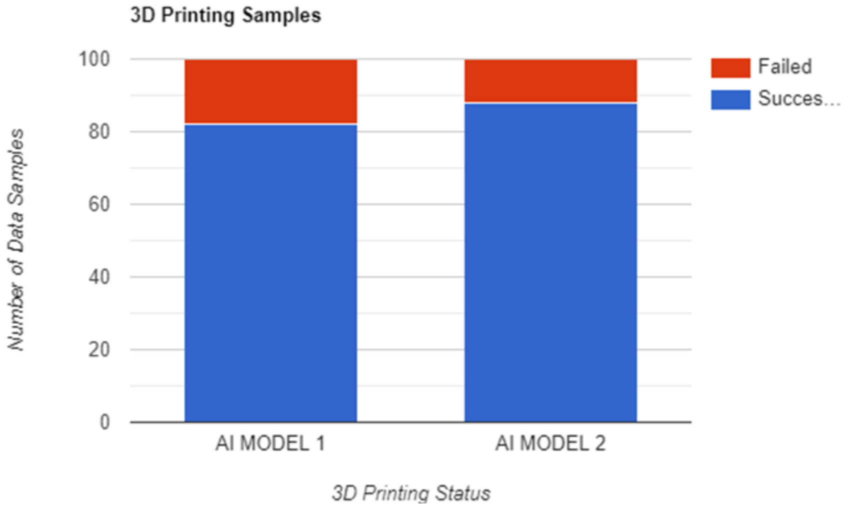

Figure 6. Graph of experiment 2

Based on the results of the experiment, AI Model 2 performed better than AI Model 1, with a higher success rate of 88% compared to 82%. However, it is unclear whether this difference is statistically significant without further testing. The two results seem to be very similar. Therefore, the experiment may not account for other factors that could contribute to 3D printing failures, such as humidity and temperature.

5. RELATED WORK

"Automated Process Monitoring in 3D Printing Using Supervised Machine Learning" by UgandharDelli et al [11].This methodology proposes a method to automatically assess the quality of 3D printed parts with the integration of a camera, image processing, and supervised machine learning. Images of semi-finished parts are taken at several critical stages of the printing process according to the part geometry. A machine learning method, support vector machine (SVM), is proposed to classify the parts into either ‘good’ or ‘defective’ category. Parts using ABS and PLA materials were printed to demonstrate the proposed framework.The main drawback of the proposed method is that the printing process needs to be paused while the images of a semi-finished part are taken. Another drawback is that since only top view images are taken, the proposed method might not be able to detect the defects on the vertical plane which cannot be seen in the top view image.

"Real-Time Monitoring of Fused Deposition Modeling 3D Printing Using Convolutional Neural Networks" by X. Zhang et al [12]. This methodology proposes a real-time monitoring system for FDM 3D printing that uses convolutional neural networks (CNNs) for print failure detection. The system consists of a camera and Raspberry Pi to capture images of the printing process and a CNN to analyze the images and detect print failures. The CNN is trained on a dataset of labeled images to recognize different types of print failures, such as under-extrusion and over-extrusion. The effectiveness of the proposed system is evaluated on a set of benchmark parts, showing that it can detect print failures with high accuracy. However, the methodology notes that the system may not be able to detect certain types of print failures, such as warping, and that the system's performance may be affected by changes in lighting or printing conditions.

"A study of failure detection and prediction for FDM 3D printers" by J. Cao et al. This article proposes a novel algorithm for failure detection and prediction in FDM 3D printers. The algorithm uses real-time monitoring of the 3D printing process, coupled with machine learning techniques, to detect and predict failures. The effectiveness of the proposed algorithm is evaluated on a benchmark dataset, showing promising results. However, the article notes that the algorithm has limitations, such as not being able to detect certain types of failures, such as warping.

6. CONCLUSIONS

There are several limitations to my project, but the biggest one is that I only used one 3D printer when I was testing out my model. Because I was only using the same printer with the same filament with the same wall background, my model was only used to one type of environment. To fix this, I would need to use various types of 3D printers, multicolored filaments, different colored walls, and separate timestamps located throughout the day [15]. By doing this, I would be able to simulate my environment better.

Along with different testing environments, something else that could be worked on is the aesthetic of the app. Right now, it's very simple: it has a login page, and another page where the user can see an image of the 3D printer status. However, the app could be improved by having designated folders labeled with the project name to help organize the user's 3D prints easier. In addition, having multiple images on one page would be much more convenient to navigate through.

Based on the first experiment, it was found that the AI-powered program was able to detect the status of a 3D printer with a high degree of accuracy [14]. In the second experiment, two different AI models were tested for their ability to monitor 3D printing processes. However, further analysis is required to determine if this difference is statistically significant. Overall, these experiments demonstrate the potential of using AI and machine learning for improving the accuracy and efficiency of 3D printing processes.

REFERENCES

[1] Gopinathan, Janarthanan, and Insup Noh. "Recent trends in bioinks for 3D printing." Biomaterials research 22 (2018): 1-15.
[2] McLaren, Bruce M. "Extensionally defining principles and cases in ethics: An AI model." Artificial Intelligence 150.1-2 (2003): 145-181.
[3] Zhao, Cheah Wai, JayanandJegatheesan, and Son Chee Loon. "Exploring iot application using raspberry pi." International Journal of Computer Networks and Applications 2.1 (2015): 27-34.
[4] Shahrubudin, Nurhalida, TeChuan Lee, and R. J. P. M. Ramlan. "An overview on 3D printing technology: Technological, materials, and applications." Procedia Manufacturing 35 (2019): 1286-1296.
[5] Robertson, B. F., and D. F. Radcliffe. "Impact of CAD tools on creative problem solving in engineering design." Computer-aided design 41.3 (2009): 136-146.
[6] Wolfs, R. J. M., and A. S. J. Suiker. "Structural failure during extrusion-based 3D printing processes." The International Journal of Advanced Manufacturing Technology 104 (2019): 565-584.
[7] Green, Siân E., et al. "Innovations in camera trapping technology and approaches: The integration of citizen science and artificial intelligence." Animals 10.1 (2020): 132.
[8] Pérez, Bianca, et al. "Impact of macronutrients printability and 3D-printer parameters on 3D-food printing: A review." Food chemistry 287 (2019): 249-257.
[9] Bishop, Chris M. "Neural networks and their applications." Review of scientific instruments 65.6 (1994): 1803-1832.
[10] Alves, Francisco Regis Vieira, and Renata Passos Machado Vieira. "The Newton fractal's Leonardo sequence study with the Google Colab." International Electronic Journal of Mathematics Education 15.2 (2019): em0575.

[11] Delli, Ugandhar, and Shing Chang. "Automated process monitoring in 3D printing using supervised machine learning." Procedia Manufacturing 26 (2018): 865-870.
[12] JinZeqing, Zhizhou Zhang, and Grace X. Gu. "Autonomous in-situ correction of fused deposition modeling printers using computer vision and deep learning." Manufacturing Letters 22 (2019): 11-15.
[13] Verana, Mark, et al. "Deep learning-based 3d printer fault detection." 2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN). IEEE, 2021.
[14] Fountaine, Tim, Brian McCarthy, and Tamim Saleh. "Building the AI-powered organization." Harvard Business Review 97.4 (2019): 62-73.
[15] Fuji, Takao, et al. "Experimental and theoretical investigation of a multicolor filament." Physical Review A 80.6 (2009): 063822.

© 2023 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.