Received 13 June 2023, accepted 27 June 2023, date of publication 12 July 2023, date of current version 18 July 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3294486

RESEARCH ARTICLE

Integrating Machine Learning Model and Digital Twin System for Additive Manufacturing

NURSULTAN JYENISKHAN[1], AIGERIM KEUTAYEVA[2], GANI KAZBEK[3], MD HAZRAT ALI[1,4], (Senior Member, IEEE), AND ESSAM SHEHAB[1]
[1]Mechanical and Aerospace Engineering Department, SEDS, Nazarbayev University, 010000 Astana, Kazakhstan
[2]Robotics Engineering Department, SEDS, Nazarbayev University, 010000 Astana, Kazakhstan
[3]Civil and Environmental Engineering Department, SEDS, Nazarbayev University, 010000 Astana, Kazakhstan
[4]School of Engineering and Digital Sciences, Nazarbayev University, 010000 Astana, Kazakhstan

Corresponding authors: Essam Shehab (essam.shehab@nu.edu.kz) and Md Hazrat Ali (md.ali@nu.edu.kz)

This work was supported by Nazarbayev University under the Faculty Development Competitive Research Grant Program (FDCRGP) under Grant 11022021FD2904.

ABSTRACT Additive manufacturing is a promising manufacturing process with diverse applications, but ensuring the quality and reliability of the manufactured products are key challenges. The digital twin has emerged as a technology solution to address these challenge, allowing real-time monitoring and control of the manufacturing process. This paper proposes a digital twin system framework for additive manufacturing that integrates machine learning models, employing Unity, OctoPrint, and Raspberry Pi for real-time control and monitoring. Particularly, the system utilizes machine learning models for defect detection, achieving an Average Precision (AP) score of 92%, with specific performance metrics of 91% for defected objects and 94% for non-defected objects, demonstrating high efficiency. The Unity client user interface is also developed for control and visualization, facilitating easy additive manufacturing process monitoring. This research article presents a detailed description of the proposed digital twin framework and its workflow for implementation, the machine learning models, and the Unity client user interface. It also demonstrates the effectiveness of the integrated system through case studies and experimental results. The main findings show that the proposed digital twin system met its functional requirements and effectively detects defects and provides real-time control and monitoring of the additive manufacturing process. This paper contributes to the growing field of digital twin technology and additive manufacturing, providing a promising solution for enhancing the quality and reliability in the field of additive manufacturing.

INDEX TERMS Additive manufacturing, digital twin, machine learning, unity, defect detection, real-time control, smart manufacturing.

I. INTRODUCTION

Digital twins (DT) and additive manufacturing (AM) are both key technologies in the fourth industrial revolution [1]. Additive manufacturing has the potential to produce complex components or products that are difficult to manufacture using conventional methods [2], [3], [4]. Unlike traditional subtractive manufacturing methods, additive manufacturing involves printing raw materials on a layer-by-layer basis to produce final products with minimal waste of materials [3], [5], [6], [7]. However, the quality of printed products is still a significant challenge [7], [8]. Digital twin technology has emerged as a solution to this problem, enabling real-time monitoring and controlling of the additive manufacturing processes [3], [7], [8].

A digital twin is a virtual representation of a physical system that allows for real-time monitoring and control through two-way data exchange between the physical system and its digital twin [4], [9], [10], [11], [12]. While the technology is still in its early stages of development, it has the potential to offer many benefits in additive manufacturing [10]. However, the lack of a standardized architecture or structure for creating generic digital twin models for 3D printers is a major challenge [3], [8], [13], [14].

The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Merlino.

Recently developed digital twin systems for additive manufacturing suffer from a number of problems. First, it is challenging to integrate these solutions easily due to a lack of a standard and effective architectural design and framework. Current solutions generally rely on an excessive quantity and variety of sensors, even though the machine has embedded sensors. This redundancy adds complications and expenses that are not essential. Furthermore, there is a lack of sufficient intelligence integration within these solutions, which is one of the most important functions of the digital twins, hindering their effectiveness. The lack of a standard digital twin implementation guide for additive manufacturing resulted in different approaches being adopted by various researchers, which were briefly addressed in this research paper.

The primary motivation of this research work is to solve the above-mentioned problems, which are the creation of a simple-to-follow universal digital twin implementation architecture and framework, using in-built sensor data to create a digital twin, and integrating a certain level of intelligence. Thus, this paper proposes a digital twin system for Fused Deposition Modeling (FDM) printers that incorporates machine learning models. The system employs machine learning models for defect detection and the Unity client user interface to provide control and visualization of the 3D printing status and processes. The main objective is to investigate recent research efforts in this area and to present a detailed description of the proposed framework of digital twin models for FDM printing, the machine learning models, and the Unity client user interface. Initially, this research reviewed the literature on FDM printing and digital twin technology and proposed a simplified yet effective digital twin implementation architectural design for FDM printers. The research study also demonstrates the effectiveness of the integrated system through a case study and experimental results. The digital twin system framework proposed in this paper is designed explicitly for fused deposition modelling (FDM) 3D printers, utilizing a Raspberry Pi as the core controller, OctoPrint as the communication software, and Unity for user interaction. The proposed digital twin system has the potential to provide a solution for enhancing the quality and reliability of additively manufactured products, enabling the realization of smart manufacturing and digitalization.

The key contributions of this research work are in the field of digital twin implementation for fused deposition modelling (FDM) 3D printers. The development of a framework and design for the digital twin that is especially suited for FDM printers is one of its significant accomplishments. Moreover, this research study emphasizes the importance of an approach that minimizes the use of external sensors, thus enhancing convenience and efficiency. Another notable accomplishment that increases the digital twin's overall efficiency is the integration of a certain level of intelligence. Finally, this study focuses on three essential functional criteria for the digital twin, which are also the critical contributions: bidirectional communication between the physical and the digital models, real-time monitoring and control capabilities, and integration of intelligence.

II. RELATED WORK

Due to the lack of a universal digital twin implementation method for FDM printers, several studies have explored different frameworks. Delli and Chang [15] utilized a camera to acquire top-view images of parts during the printing process, comparing them to a supervised machine learning model to detect defects. Henson et al. [16] employed an optical images method to detect defects of parts on a layer-by-layer basis. Mourtzis et al. [17] used an augmented reality approach to decrease part defects and provide a sophisticated user interface. Similarly, Yi et al. [18] used augmented reality to monitor greenhouse gas emissions, production costs, and energy consumption. They also used the Volume Approximation by Cumulated Cylinder (VACCY) approach to estimate the volume of printed parts, geometrical shape, current, and target positions of the nozzle or extruder. Paripooranan et al. [19] created an augmented reality-enabled digital twin for an FDM printer, developing a user-friendly virtual FDM printer using various microcontrollers and software. Odada et al. [20] and Pantelidakis et al. [21] used external sensors to measure and detect nozzle movements and mimic them in a virtual environment. Stavropoulos et al. [22] used a similar framework to the above researchers but with different equipment, using data visualization for visualization instead of augmented reality.

However, there is a lack of studies that integrated machine learning models along with the digital twin implementation for FDM printers. Hence, some level of intelligence is one of the essential parts of the digital twin. Before integrating this, it is important to address the recent developments of machine learning in FDM printing. Various supervised machine learning algorithms are used in FDM printing to detect or predict defects. According to Sandhu et al., 2019, geometrical anomalies of a printed part can be identified accurately through pre-trained models [23]. These models can even suggest the best printing settings based on the parameter value and geometry [24]. The research also demonstrated that problem detection in real-time processes of FDM printers might be implemented using pre-trained offline ML at low or reasonable computing and experimental expenses [5], [25].

Several studies have focused on real-time defect detection and monitoring in the context of additive manufacturing. The table below provides a summary of relevant papers that have employed machine learning techniques for defect detection in 3D printing processes (see Table 1).

III. METHODOLOGY

The adopted methodology is illustrated in Figure 1. The novel architectural framework design of a digital twin implementation are developed after carefully observing and analyzing

TABLE 1. Relevant studies that have employed machine learning techniques for defect detection in 3D printing processes.

Source	Real-time	Dataset	Type of defects	Type of sensors	3D Printer type	ML Model	Accuracy	Comments
[28]	Yes	1.2 million images from 192 different parts labelled with printing parameters	Various	Camera	Extrusion 3D printing system	Multi-head neural network	-	Used Raspberry Pi
[29]	Yes	Images where the stringing issue is clearly displayed	Stringing	Camera	Fused Filament Fabrication (FFF)	Deep CNN	-	used microprocessors
[30]	-	-	Surface defects	Camera	FDM	Self-feature extraction method of shape defect detection	-	-
[31]	-	3D data from the printing base, 3D data from the head accelerometer, and a tension, measured every 0.1s.	Arm failure, Bowden tube fallout, Failure in plastic finish, Wrong retraction, Unsticking models, Normal or no-fault	Printing base, Head accelerometer, Tension measured every 0.1s.	FDM	CNN	99.67%	-
[32]	-	4140 image blocks of LMD build parts quality optical images.	Crack, Gas Porosity, Lack of Fusion, Good Quality.	-	Laser Metal Deposition (LMD)	CNN	92.10%	-
[33]	-	Images crawled from Google and YouTube	Layer shift, strings, under extrusion, warping	Camera	FDM	CNN	Binary: 96.72%, Multi: 93.38%	Used Raspberry Pi
[15]	-	Images taken at various checkpoints during the printing process	failure defects, structural defects, geometrical defects	Camera	Lulzbot Mini (FDM)	Support Vector Machine (SVM)	-	-
[34]	-	Process data from printer itself and additional vibration sensor	Uncompleted product, Poor-quality product	Temperature and humidity sensor, Vibration sensor, Acceleration sensor	-	Logistic Regression (LR), SVM, Random Forest (RF).	Poor-quality product: LR - 91.76%, SVM - 91.56%, RF - 90.54%. Uncompleted product: LR - 92.5%, SVM - 94.93%, RF - 95.43%	Used Arduino
This paper	Yes	Images from a 3D FDM printer comprising both standard and defective prints.	Defected product, Non-Defected product	Camera and built-in sensors.	FDM	EfficientDet-Lite0 - Lite4	High accuracy rates were reported for the models.	Used Raspberry Pi 3B

the current literature review. Additionally, the development process of a digital twin for FDM printers is discussed in detail in further sections of this paper. The primary purpose of the digital twin framework design is to employ open-source platforms such as Octoprint due to its accessibility by the public and free of charge for 3D FDM printer control. Regarding real-time data and printer status monitoring and controlling, the Unity client platform was used due to its ability to design and implement high-quality custom design and visualization tools. Three main functionalities of the digital twin were intended to be solved by this research work, such as real-time monitoring and controlling, bidirectional communication between digital and physical models, and integration of some level of intelligence. Finally, testing the developed digital twin's efficacy and meeting its functionality requirements.

The most critical outcome for widely used architecture for FDM digital twins should be bidirectional communication and machine learning models to predict possible errors in printing parts. The bidirectional communication function of the digital twin is the automatic data flow between physical and digital models [9], [12]. However, achieving this without proper middleware is not always possible due to a lack of in-built or embedded communication technology compatible with digital models and other technologies. Thus, middleware plays an essential role in receiving data from physical models and sending received data to digital models promptly and with minimal errors. Similarly, digital models send data to physical models through middleware in real time without any errors so that the system can have proper bidirectional communication. The lack of proper middleware architecture in recent machines and technologies results in challenges in acquiring a viable digital twin model.

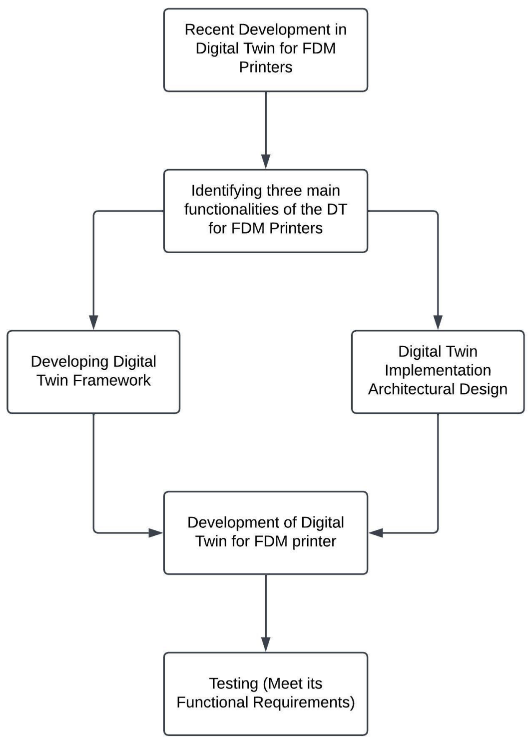


FIGURE 1. Adopted research methodology.

A. THE DEVELOPMENT OF A DIGITAL TWIN FRAMEWORK FOR FDM PRINTERS

The following framework for the digital twin structure, as shown in Figure 2, is created based on the needs and capabilities of a digital twin. The physical device, data acquisition, simulation, and digital model layers are the four main core layers of the DT framework. There are FDM printers, built-in sensors in the printer, a camera, and extra sensors in the physical layer. The camera is used for remote visual monitoring and video processing of pre-trained machine-learning models during printing.

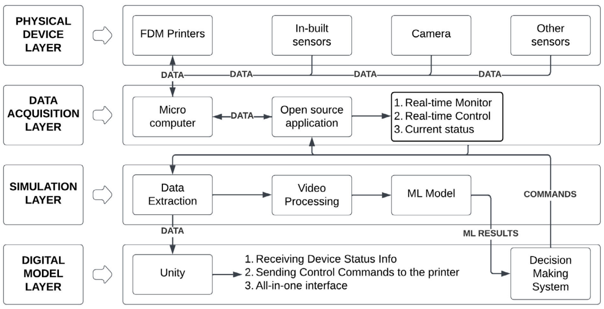


FIGURE 2. The digital twin framework.

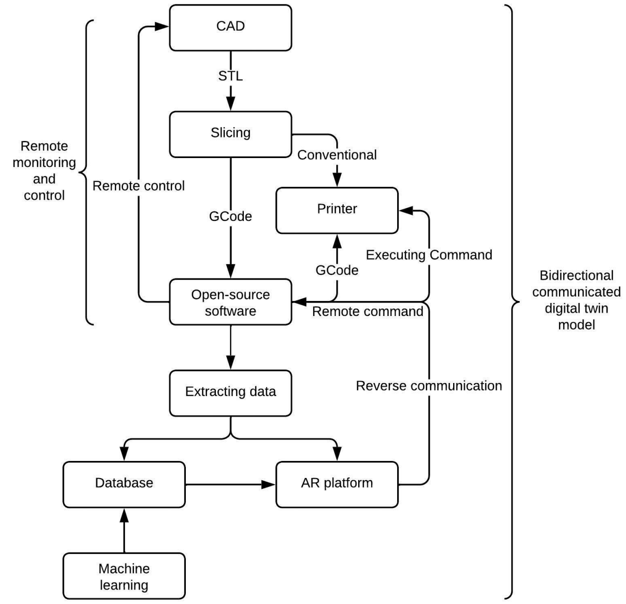


FIGURE 3. Digital twin workflow of FDM printers.

The data collection layer, which is the second layer, is important for gathering information from the attached camera and the built-in sensors of the FDM printer. A free, open-source program that works with the FDM printer can be used for this process. By employing a microcomputer to act as a bridge between the physical printer and the open-source application, it is possible to monitor and control the FDM printers using open-source software. Processing data is very important through gathered data. The digital model and simulation layers should receive all extracted data from the physical FDM printer for processing.

The machine learning model plays a significant role in analyzing real-time video data and identifying whether there are defects or not via video processing in the simulation layer. The machine learning model does not interfere with the system or alert it if there is no defect detected. To prevent wasting more time, resources, and money, if a defect is found, a signal is sent to the digital model layer informing it that there is a defect so the operator can intervene in the system. Humans should, however, get involved in this research scenario. With the help of more coding and development can be done to disregard human intervention throughout the system completely.

The digital model layer cannot exist without the Unity client because the Unity client user interface gathers all of the important functions of the digital twin, such as real-time monitoring, controlling functions, and some degree of intelligence integration. Intelligence is one of the key characteristics and functions of the digital twins. As a result, combining machine learning with remote control and real-time monitoring of the printing process yields a minimally viable digital twin model for FDM printers. Finally, reverse communication is required to achieve two-way communication, which is a very important function of digital twins.

B. THE WORKFLOW OF A DEVELOPED DIGITAL TWIN

It is crucial to understand and define the proposed digital twin implementation framework mentioned in the previous subsection, and its workflow is shown in Figure 3. In a conventional method, the sliced Gcode file of any CAD file is uploaded to an SD card and inserted into the FDM printer for printing. However, in this research case, created CAD file is sliced to generate G-code. Then, the printer receives the G-code remotely without using any SD card or physical device and starts to print layer-by-layer when the command is given to initiate the printing. All the printing data can be extracted from the printer to open-source software with the help of a microcontroller connected to the printer. All manual work can be done remotely now via open-source software. Then, all the printing details and parameter data can be extracted from the software to the database to implement a pre-trained machine learning model to predict possible errors of printing parts and the AR platform to acquire a sophisticated, responsive, user-friendly interface with the help of data fed in real-time. At the same time, the Unity client receives a signal from the database as to whether there is a defect. If there is a defect, a special script in Unity triggers the stop command to the printer. As can be seen, this methodology and proposed research fulfil the most important parts of the digital twin model: two-way communication between physical and digital machines and intelligence with the help of a machine learning algorithm.

C. DIGITAL TWIN IMPLEMENTATION DEVELOPMENT FOR FDM PRINTERS

A physical object is the FDM printer, in this case, the digital twin to be created. All information from the FDM printer can be extracted from in-built sensors and transferred to a digital twin through the data channel. The digital twin, in this case, is the digital replica of a physical FDM printer, which can be in the form of data visualization or CAD file viewed and controlled through an augmented reality platform. More importantly, digital twins should be fed with real-time data from physical FDM printers through a data channel.

The middleware is the most important part that plays a significant role in communication among physical and digital models. The data channel should be efficient and reliable. Hence there are different types of data from various sensors and protocols. Moreover, middleware receives data from the physical model, sends it to the digital model, and processes it in parallel. Then, after analyzing the data, the digital twin sends a command to the physical model according to the decisions made based on the result of a data process.

The InterPrint FDM printer is selected as a physical object in this research. This printer requires an extra microcontroller to make this printer remotely accessible and controllable. Thus, as a microcontroller, Raspberry Pi 3B is selected. In addition to these, the Logitech camera and chair lamp are chosen. The camera takes pictures of parts during printing at certain stages and compares them with a trained machine-learning model to determine defects. Due to the lack of enough light and the constantly moving printer bed, it is challenging to shoot good-quality pictures. Thus, with the help of extra light, it can be done more precisely.

Octoprint is selected to control and monitor the 3D printer remotely. It is an open-source platform, so that can be done in terms of user customization. The Raspberry Pi was used to run Octoprint software and make the 3D printer accessible for requests. Moreover, Octoprint has a REST.API is a set of rules defining how applications or devices can connect to and communicate. Octoprint is an open-source application with many ready-to-use plugins, making it easy to use. MS Azure was chosen to create a cloud server to hold a database and run ML algorithms. The starting point for integrating all components of the project in Unity was to retrieve data from the Octoprint web interface. Conveniently Octoprint already has an API that, in turn, uses the REST API; for this purpose was found a public GitHub repository called Printerface which was used as a basis since it already contains methods for both data output and various commands.

Raspberry Pi 3B was used as a bridge between the 3D printer and the digital environment. Then, Octoprint software was chosen as a part of cyberspace, where all information comes from the 3D printer. Through Octoprint, 3D printers can also be monitored (printer status, bed and nozzle temperatures, real-time streaming, and G-code viewer). The core of cyberspace is the MS Azure server, where the database collects datasets from the 3D printer and ML algorithms analyze and make predictions. Eventually, as a part of the cognition level, the Unity client was developed for remote visualization for users. All printing statuses and parameters can be seen through a user interface and control. It is not possible to directly control the 3D printer. Thus, the commands are first sent to Octoprint so that the software can intervene or prevent the physical 3D printer. It is interesting to note that all the data comes from in-built sensors in the 3D printer.

D. MACHINE LEARNING FOR DEFECT DETECTION

Machine learning plays a significant role in digital twin applications for identifying whether a printed object has defects. Here are some key reasons why machine learning is important in this context:

- **Defect Detection:** Machine learning algorithms can be trained to analyze 3D models and identify potential defects in printed objects. By learning patterns and characteristics of defective prints from labelled data, machine learning models can detect anomalies, such as structural imperfections, surface irregularities, or other printing errors. This helps in ensuring the quality and integrity of the printed objects.
- **Automation and Efficiency:** Digital twin systems generate vast amounts of data, including sensor readings, images, and other measurements. Machine learning algorithms can process this data at scale and automate the defect detection process. This significantly reduces manual inspection efforts, speeds up the analysis, and enables real-time print quality monitoring.

1) DATASET COLLECTION AND PREPROCESSING

The data collection process for the OctoPrint interface involved capturing both video and images to monitor the 3D printing process. The OctoPrint interface provided these visual data sources for analysis and evaluation.

Initially, images were utilized to gather information about the printing progress. However, if the images were deemed insufficient in terms of quality or if there were issues with repeated frames, a Python script was employed to divide the video into individual images. This allowed for a more detailed examination of the printing process.

In addition to visual data, OctoPrint also provided temperature and other relevant information. This data was directly fed into Unity, which could be analyzed and processed for further use.

To account for potential defects or anomalies in the printing process, some deliberately flawed images were captured. This was achieved by manually shaking the printer, increasing the temperature beyond recommended levels, or modifying certain parameters of the 3D printer. These intentionally induced flaws served as valuable test cases for evaluating the system's robustness and identifying potential areas for improvement. The images were then manually labelled (defect and non-defect) with bounding box annotations for the objects of interest using a labelling tool.

Due to their accessibility, simplicity of use, and wide range of materials they can work with, fused deposition modelling (FDM) printers are not without their own unique set of difficulties and potential flaws, just like any other technology. Stringing, Layer Separation and Splitting, and Clogging, which cause excess and under extrusion, were found to be the three main fault types in a dataset collected from FDM printers.

1) Stringing (Hairy Prints/Spaghetti):
   Stringing, often known as hairy prints, spaghetti, or oozing, is an issue that frequently affects FDM printers. Thin plastic strings or strands linking various print components are the defining features of this fault. When the printer's extruder moves from one area to another without ceasing to extrude, this problem frequently arises. Numerous things, such as improper retraction settings, an excessively high printing temperature, or insufficient cooling, might result in stringing.
2) Layer Separation and Splitting:
   When a print's component layers do not adhere to one another effectively, layer separation and splitting take place. This flaw frequently leads to a final product that is fragile and easily separated along the layer boundaries. Low extrusion temperature, a high cooling fan speed, and an incorrect printing speed are all factors that might cause layer separation and splitting. The choice of material is particularly important since some materials, like ABS, are more prone to this flaw than others, like PLA, because of their greater thermal contraction.
3) Clogging (Over and Under Extrusion):
   A clogged printer has a nozzle that is plugged, preventing the flow of melted filament. Both over- and under-extrusion are effects of this problem. Over-extrusion is the process of extruding too much filament, which results in excess plastic and can produce blobby, uneven prints. It may be caused by wrong filament diameter or flow rate settings, inaccurate step-per-millimetre calibration of the extruder, or even a combination of these. On the other hand, insufficient filament extrusion leads to under-extrusion. It results in missing layers, print gaps, or prints that are not solid. A partially clogged nozzle, improper extrusion multiplier or filament diameter settings, or a problem with filament feed, such as a tangle in the spool, can all contribute to this.

In general, the data collection process for the OctoPrint interface involved capturing video and images, utilizing a Python script to extract images from the video, incorporating temperature and other data from OctoPrint directly into Unity, and intentionally generating defective images to assess system performance. This comprehensive approach facilitated a thorough analysis of the 3D printing process and contributed to enhancing the overall system's efficiency and reliability.

The data was then augmented with horizontal flipping and scale jittering [0.1, 2.0] to make the data-rich and the model perform accurately. Overall, 120 images were collected, and 25 images were used for testing. Feature extraction was done using pre-trained models and transfer learning techniques, and the models were then trained on the extracted features to detect image defects.

2) TRANSFER LEARNING

EfficientDet-Lite, an object detection model, was chosen as the source model for transfer learning. It is specifically designed for performance on mobile CPUs, GPUs, and EdgeTPUs, making it suitable for real-time inference in the additive manufacturing system. EfficientDet-Lite is a variant of the EfficientDet architecture introduced by Mingxing Tan et al. in [27]. The EfficientDet-Lite model was originally trained on the COCO 2017 dataset, which consists of a large number of images across multiple object classes. It utilizes a bi-directional feature pyramid network (BiFPN) to combine feature maps of different resolutions and a classification and regression subnetwork to predict the presence and location of objects (see Figure 4).

The loss function used to train the models was a weighted sum of the focal and smooth L1 loss.

The Focal Loss:

$$F(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t), \tag{1}$$

where p_t is the predicted probability of the true class, $\alpha_t = 0.25$ is the weighting factor, and $\gamma = 1.5$ is the focusing parameter.

The Smooth L1 Loss:

$$loss(x, y) = \begin{cases} (x-y)^2\sigma^2/2, & if\ |x-y| < 1/\sigma^2 \\ |x-y| - 1/2\sigma^2, & otherwise \end{cases}, \tag{2}$$

where x and y are the predicted and target values, respectively, and $\sigma \in \{0.5, 1, 2\}$ is the point where the loss changes from L2 to L1.

3) MODEL ADAPTATION

Transfer learning was employed to adapt the EfficientDet-Lite model to the specific requirements of additive manufacturing. The pre-trained EfficientDet-Lite model was fine-tuned using the collected dataset of 120 images related to additive manufacturing. The last few layers of the EfficientDet-Lite model were replaced with new layers to accommodate the specific number of object classes relevant to additive manufacturing. During fine-tuning, the weights of the pre-trained EfficientDet-Lite model were frozen, and only the newly added layers were trained using the collected dataset. The transfer learning process enables the model to leverage the knowledge learned from the large-scale COCO 2017 dataset while adapting to the specific domain of additive manufacturing.

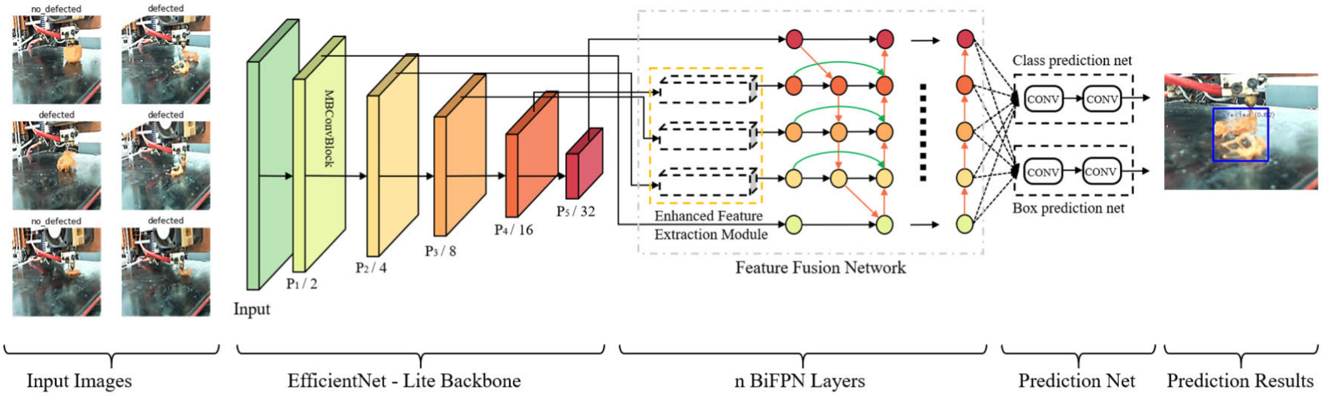


FIGURE 4. The EfficientDet architecture utilizes EfficientNet-Lite [27] as its backbone network, and features a BiFPN layer as its feature network, along with a shared network for class and box predictions.

TABLE 2. The algorithmic hyperparameters of the EfficientDet-Lite model.

Batch size	Epochs	Learning rate	Optimizer	Batch Normalization	Activation
4	20	Linearly increased from 0 to 0.16 in the first training epoch and then annealed down using cosine rule	SGD optimizer with momentum 0.9 and weight decay 4e-5	Batch norm decay 0.99 and epsilon 1e-3	SiLU (Swish-1) activation

TABLE 3. The performance of each EfficientDet-Lite models compared to each other.

Model architecture	Size(MB)	Latency(ms)	Average Precision
EfficientDet-Lite0	4.4	146	25.69%
EfficientDet-Lite1	5.8	259	30.55%
EfficientDet-Lite2	7.2	396	33.97%
EfficientDet-Lite3	11.4	716	37.70%
EfficientDet-Lite4	19.9	1886	41.96%

4) MODEL TRAINING AND EVALUATION

The adapted EfficientDet-Lite model was trained using the collected dataset of annotated images. The training process involved iterative optimization of model parameters to minimize the detection loss and improve the accuracy of object detection. The dataset was divided into training and validation sets to monitor the model's performance during training and prevent overfitting. Model performance was evaluated using various evaluation metrics such as precision and mean average precision (mAP) on the validation set. Hyperparameters, including learning rate, batch size, and the number of training epochs, were tuned to achieve the best possible performance (see Table 2).

The EfficientDet-Lite models used in this paper have varying sizes, latencies, and average precisions. This paper experiments with five different EfficientDet-Lite models, as it is important to choose the best one in lightness, accuracy, and effectiveness for the whole digital twin system. As shown in the table below, the models become more extensive and more accurate as their index increases, with EfficientDet-Lite4 being the largest and most accurate of the models.

The size of the models refers to the size of the integer quantized models, with larger models requiring more storage space. The latency of the models was measured on a Raspberry Pi 4 using four threads on the CPU, with larger models having the highest latencies. Finally, the average precision of the models is the mean average precision (mAP) on the COCO 2017 validation dataset, with larger models having the highest average precision.

The EfficientDet-Lite models were chosen due to their high accuracy and low computational requirements. By choosing a smaller model, such as EfficientDet-Lite0 or EfficientDet-Lite1, the system can achieve lower latencies and use less storage space while sacrificing some accuracy. On the other hand, using a larger model such as EfficientDet-Lite4 can achieve higher accuracy at the cost of higher latencies and larger storage requirements.

5) MODEL DEPLOYMENT

After training and evaluation, the final adapted EfficientDet-Lite0-Lite4 models were ready for deployment in the additive manufacturing system. The models were converted to TensorFlow Lite (TFLite) format, optimized for deployment on Raspberry Pi. The deployment of the model enables real-time object detection and monitoring within the digital twin system of the additive manufacturing setup. By leveraging transfer learning and fine-tuning the EfficientDet-Lite model, it was possible to develop an accurate and efficient object detection system tailored to the specific requirements of additive manufacturing. The trained model can now be integrated into the digital twin system to enhance monitoring, control, and quality assurance in the additive manufacturing process.

6) DECISION-MAKING

The decision-making is done in two ways: manually using notifications and automatically. In both cases, the decision is made using the confidence level probability, where the results below 30% mean the defect is minor. The $P \geq 0.30$ was chosen to make an automatic stop of the printing process, while $0 < P < 0.3$ makes defect notifications to the user to stop manually if needed. The defect detection results with confidence level are saved in real-time automatically and accessed by Unity to control the whole system.

Overall, integrating machine learning models in a digital twin system shows promise for enhancing the quality and reliability of additively manufactured products. The system can provide operators with immediate feedback by detecting defects in real time, allowing adjustments before defective products are produced. The use of machine learning techniques, including data cleaning, feature extraction, and transfer learning, helped improve the models' accuracy, demonstrating the potential of machine learning in defect detection for additive manufacturing.

E. UNITY CLIENT USER INTERFACE

Unity engine is a powerful and versatile tool for building digital twins of cyber-physical systems. Unity engine has the ability to create digital twins of a physical system that are highly realistic and immersive. These digital twins can also correctly imitate the behaviour and performance of the original system. With capabilities like powerful physics simulation, real-time rendering, and support for VR and AR technologies, Unity is a great choice for creating digital twins. In addition, Unity has a significant, involved developer community that produces and distributes a variety of tools, documentation, and tutorials meant to streamline and accelerate the production cycle. Overall, building digital twins for additive manufacturing processes can be made much more accurate, efficient, and useful by using the Unity engine.

Development of Unity client includes the following steps:

1) COLLECTING DATA

Gathering information about the 3D printing process, such as temperature, its size, the materials it uses, and any other useful information about how it works. This information was obtained through Rest API from Octoprint.

2) DESIGNING A 3D MODEL

Designing a dimensionally accurate representation of the 3D printer in CAD software such as 3DsMax. The model should be as accurate as possible and include all the printer's parts and pieces.

3) ADDING FUNCTIONALITY

Making the 3D model of printer interactive by giving it features like temperature and speed controls, as well as the ability to manipulate the printer's nozzle. All commands were sent to the physical 3D printer through Rest API to Octoprint.

4) IMPLEMENTING PHYSICS SIMULATION

Unity scripts were written to add physics simulation to the 3D model so that it can accurately simulate how the printer moves and acts. The movement of the 3D printer was coordinated due to the information from the G-code file.

5) CREATING AN INTERFACE FOR THE SOFTWARE

User-friendly interface was developed to enable users interact with the digital twin, control the printer, and observe how its behavior. The design of the Unity client was developed in Figma software as shown in Figure 5.

IV. RESULTS

A. FUNCTIONAL REQUIREMENTS OF THE DIGITAL TWIN

Three main digital twin functional requirements are met with this design, such as bidirectional communication between models, real-time remote control and monitor, and a certain level of intelligence integration. The following illustrates how the developed digital twin model met its functional requirements.

Representational state transfer application programming interface (REST API) is a set of rules that provide how communication and connection are defined with each other. It offers greater flexibility to system developers. In this research work, the data retrieval from the FDM printer is through REST API protocol which is clearly explained in "REST API, OctoPrint documentation." The simplified design layout of all these connections from physical to digital and digital to physical is depicted in Figure 6.

Firstly, Octopi installed Pi is connected to the FDM printer. Then, through Octoprint, it can be remotely accessible to the physical printer and can all data retrieval from the physical printer to the Unity client through REST API directly from Octoprint. At the same time, the machine-learning model runs on Raspberry Pi. If it detects unusuality, then it can be seen from the monitor that the defect is detected. All the final data and messages are in the Unity client interface. The data flow from the physical to digital model is shown in green color arrows in Figure 6.

The reverse communication either starts with the Unity client or the machine learning model. There are virtual buttons such as cancel, pause, continue, and start. If one of the buttons is pressed, the command goes to Octoprint so that Octoprint initiates the command to the physical printer through the Raspberry Pi microcomputer. Moreover, if any defect is detected on the machine learning model, it sends a signal to the unity client so the printing process can be stopped. The reverse communication data flow is shown in red colour arrows in Figure 6. The following results are discussed in detail to illustrate how the developed machine learning is effective in this digital twin model in the following subsections.

B. UNITY CLIENT FEATURES

The goal of the project is to create a cutting-edge digital twin solution for additive manufacturing using the capabilities of Unity software. The innovative digital technology displays a number of capabilities intended to transform the field of additive manufacturing.

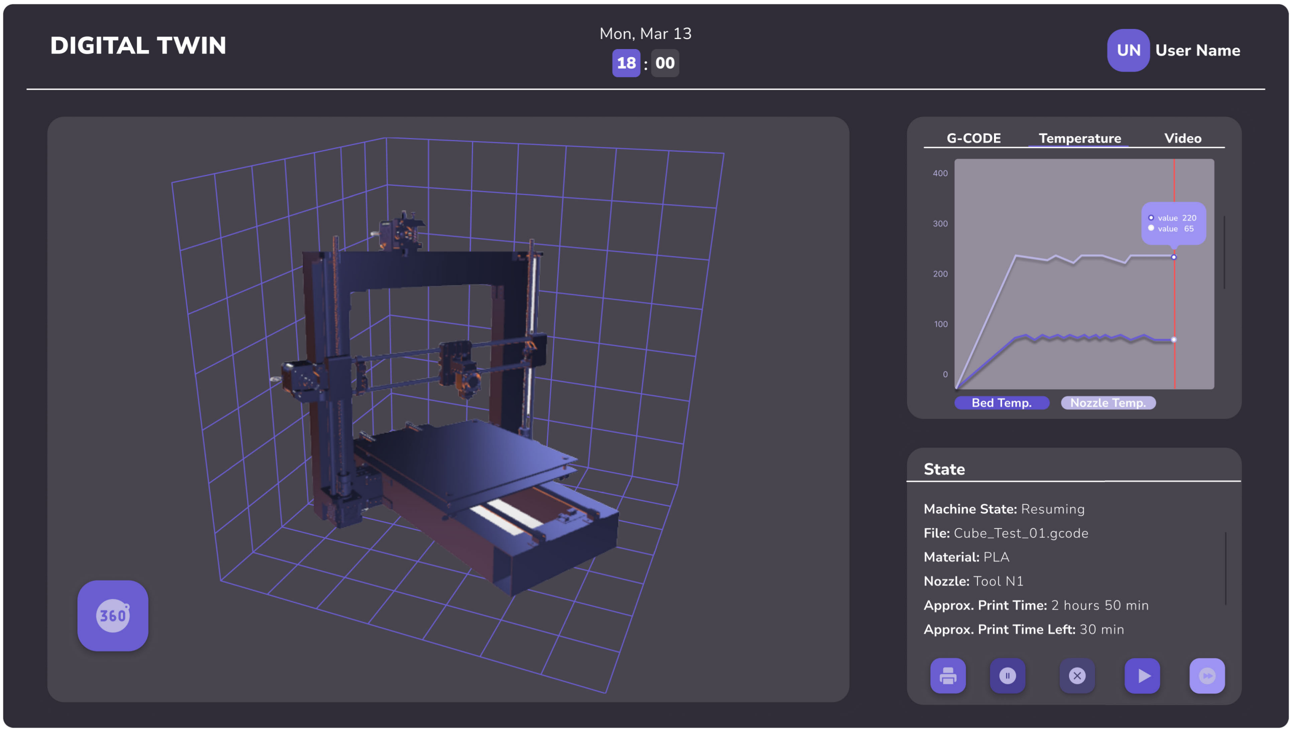


FIGURE 5. Unity client interface.

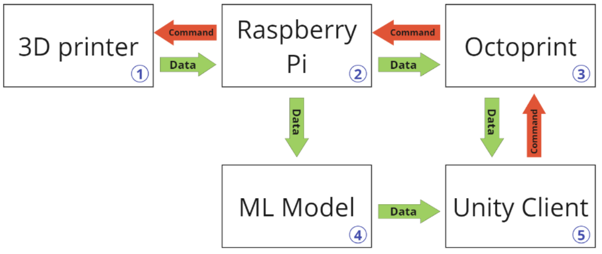


FIGURE 6. Physical to Digital (green arrows) and Digital to Physical (red arrows) model data flow.

Real-time Data Synchronization: Data is sent between the physical and digital models in real-time. It is critical to obtain precise and up-to-date information from the physical environment in order to depict the condition of the digital twin appropriately.

Command Execution: Unity client can send commands such as Start, Stop, and Resume to control the printing process. This feature can be enhanced by changing print speed, temperature, and layer thickness.

Sensor Integration: By incorporating sensor measurements and readings into the digital model, significant information can be used to monitor and improve the additive manufacturing process. The unity client accurately receives and displays sensor data (e.g., nozzle temperature, bed temperature).

Visualization and Simulation: One of the critical features is the digital mimicking of the movement of the real nozzle. The visual representation gives users a more realistic and intuitive experience. This feature can be further developed to model the printing procedure in the digital environment before the actual printing.

Camera Monitoring/Image Processing: Using cameras to keep an eye on the printing process is a fantastic tool. A user is able to see the printing process from various angles, and the camera feed is shown in the Unity client. Camera monitoring is enhanced by an ML algorithm, which examines camera feeds and delivers insights or warnings based on predefined criteria.

User interface: An intuitive user interface allows users to easily interact with the digital twin. Users can send control commands, view sensor readings, and have access to the camera feed.

Data Analytics and Optimization: The Machine Learning algorithm is used to analyze the collected data during the printing process, which helps to identify patterns and anomalies and improve the quality of the printing process. ML model constantly monitors the printing parameters and sends signals to the Unity client; then Unity client sends a request to stop printing if the 3D object defected. This feature optimizes the printing parameters and enhances the overall efficiency.

C. MACHINE LEARNING

Table 4 shows the performance of each EfficientDet-Lite model in detecting defects in the additive manufacturing process. The models were evaluated based on their average precision (AP) in detecting both defected and non-defected prints, as well as their AP in detecting only defected prints.

Moreover, Table 4 and Figure 7 show that as the model size increases, the average precision in detecting both defected and non-defected prints improves. This is evident in the higher AP_No_Defected scores for efficientdet_lite3 and efficientdet_lite4 compared to the smaller models. However, for detecting only defected prints, the efficientdet_lite3 model outperformed the larger efficientdet_lite4 model, suggesting that the former may be a better choice for defect detection specifically.

TABLE 4. The performance of each EfficientDet-Lite models for each defected, nondefected data, and general.

Model	AP_Defected	AP_No_Defected	AP
EfficientDet-Lite0	82.52%	77.55%	80.04%
EfficientDet-Lite1	78.86%	70.51%	74.69%
EfficientDet-Lite2	83.37%	86.29%	84.83%
EfficientDet-Lite3	90.86%	93.81%	92.34%
EfficientDet-Lite4	88.81%	96.91%	92.86%

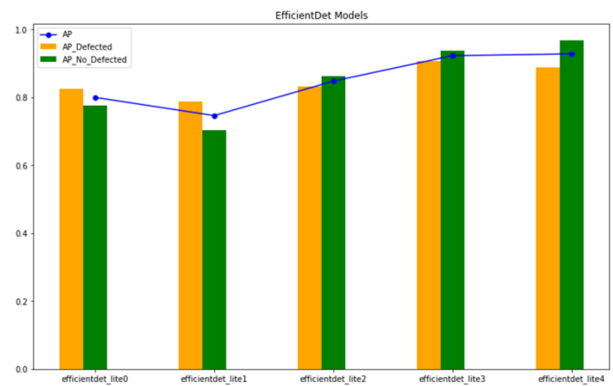
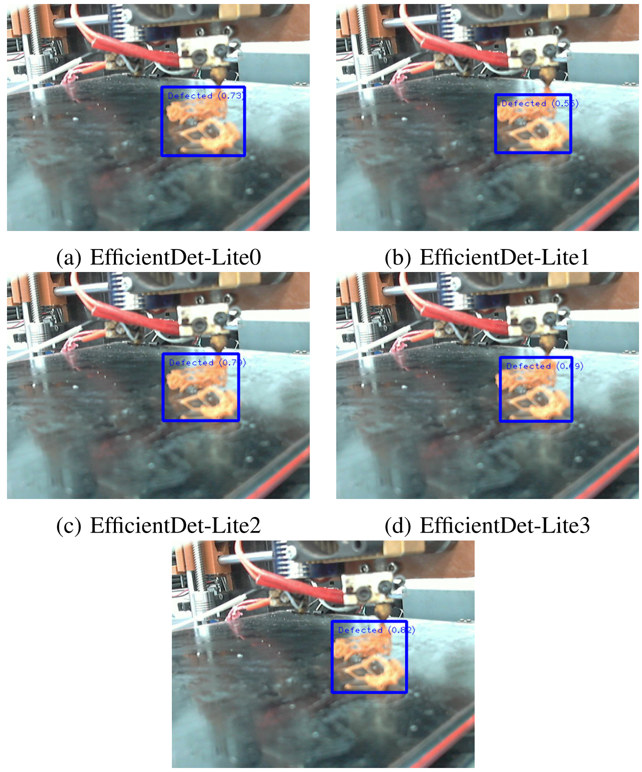


FIGURE 7. The EfficientDet-Lite 0 - 4 models and their performance using average precision metrics.

According to the figures provided, the use of EfficientDet-Lite models in the digital twin system allowed for the accurate detection of defects in real time during the additive manufacturing process. The choice of model will depend on the specific requirements of the system, such as accuracy, latency, and storage space. Moreover, there is another factor that is crucial when working with images, which is the image quality. Figure 13 shows the performance of the models on one test image with confidence level values. The confidence level is a probability score assigned by the classification network for the detected object belonging to a certain class (defect, no_defect). Though the quality of the images is not the best, the figure below shows that model works perfectly.

Overall, during the real-time tests, the EfficientDet-Lite2-4 models showed better performance than EfficientDet-Lite0-1. However, in terms of the model's size, only EfficientDet-Lite2 -3 showed the best results in speed efficiency. The test on how the image quality might affect the performance was also made. Figure 9 shows the loss for



(a) EfficientDet-Lite0

(b) EfficientDet-Lite1

(c) EfficientDet-Lite2

(d) EfficientDet-Lite3

(e) EfficientDet-Lite4

FIGURE 8. The EfficientDet-Lite 0 - 4 models and their performance on the same image.

EfficientDet-Lite2-4 models, which is near 0 for all models, showing how effective the models performed. Furthermore, table 6 shows the performance of all models on specific test images. The table displays the confidence level and whether the model detected a defect (D) or not (T=True, F=False) for each image. The results show that the EfficientDet-Lite models were able to accurately detect defects in the additive manufacturing process with varying levels of confidence.

V. DISCUSSION

In this research work, three functional requirements of the digital twins are addressed. Real-time monitoring and controlling, bidirectional communication (data flow from the physical to the digital model and digital to the physical model manually or automatically), and intelligence integration are among the essential functions of a digital twin. The developed digital twin for the FDM printer met its defined functional requirements. Most state-of-the-art research works also fully or partially meet these functions by adopting different methodologies. However, in terms of implementation, simplicity and integration of machine learning algorithms to leverage digital twins are the main benefits of this research work.

Regarding the comparison of the functionalities with the existing state-of-the-art studies, this research work performs

TABLE 5. The Performance of each EfficientDet-Lite 0-4 model on specific images (C=Confidence level, D=Defected, T=True, F=False).

Models	Img 1		Img 2		Img 3		Img 4		Img 5		Img 6		Img 7		Img 8		Img 9		Img 10	
	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C
EfficientDet-Lite0	T	0.77	F	0.73	F	0.73	T	0.79	T	0.73	T	0.77	T	0.75	F	0.69	T	0.79	F	0.75
EfficientDet-Lite1	T	0.71	F	0.62	F	0.60	T	0.72	T	0.56	T	0.71	T	0.72	F	0.65	T	0.72	F	0.63
EfficientDet-Lite2	T	0.91	F	0.77	F	0.76	T	0.91	T	0.79	T	0.91	T	0.84	F	0.79	T	0.91	F	0.75
EfficientDet-Lite3	T	0.91	F	0.73	F	0.75	T	0.91	T	0.69	T	0.91	T	0.70	F	0.78	T	0.91	F	0.70
EfficientDet-Lite4	T	0.86	F	0.83	F	0.82	T	0.88	T	0.82	T	0.86	T	0.82	F	0.81	T	0.88	F	0.84

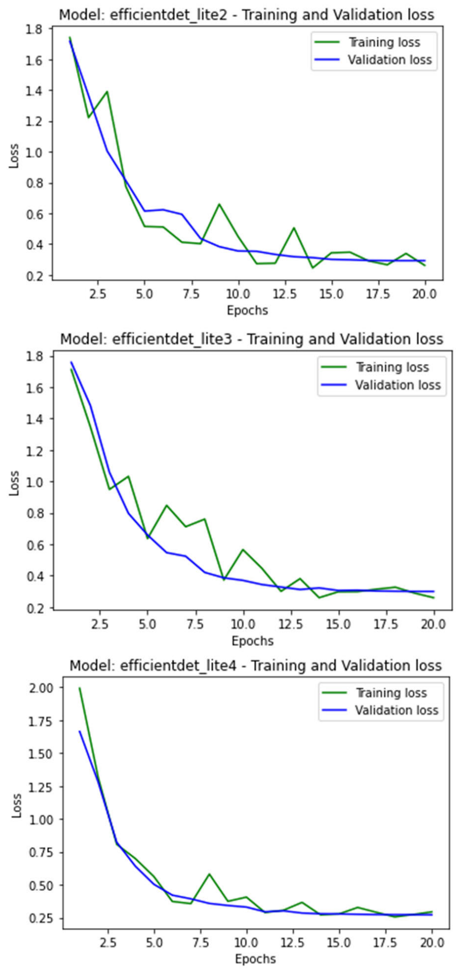


FIGURE 9. The EfficientDet-Lite 2 - 4 models training and validation losses.

better and employs the most important parts of the digital twin, as shown in Table 6.

The proposed digital twin system for additive manufacturing that integrates machine learning models for defect detection and a Unity client user interface shows promise for enhancing the quality and reliability of additive manufacturing. The system provides operators with immediate feedback by detecting defects in real time, allowing adjustments before producing defective products. The use of machine learning techniques, including data cleaning, feature extraction, and transfer learning, helped improve the models' accuracy, demonstrating the potential of machine learning in defect detection for additive manufacturing. Though the proposed architecture works well, it brings challenges and limitations. One of the frequently faced issues was the constant change in the IP address of Raspberry Pi, which created difficulty in remote access to the system. Moreover, it required an extra monitor to see and access the IP address. In addition to this, it can be accessed by the system only through the same wireless network. Thus, it creates functionality limitations. For this issue, the Octoeverywhere plugin was tested, which allowed access to the system regardless of the network difference. Though the plugin makes it easy to access the IP address, it has limitations in visually monitoring the printing system through the camera. It allows 20 seconds of monitoring every 5 minutes through the Logitech webcam camera. In addition, this plugin creates problems in the communication between models.

TABLE 6. The comparison of this research work with state-of-the-art.

Source	Monitor & Control	Dataflow	Visualisation	Defect Detect
[16]	-	two-way	-	+
[17]	+	manual	3D	-
[18]	+	two-way	3D	-
[19]	+	two-way	3D	-
[20]	+	one-way	3D	-
[21]	+	two-way	3D	-
[22]	Monitor	one-way	data	-
This paper	+	two-way	3D	+

Another issue is with the backend of the system. The machine learning model works efficiently and well, but the challenging part is integrating the machine learning into the digital twin model, which makes the system very complex. However, the problem might be solved with a cloud database that collects detection results and makes it accessible for the Unity part. The proposed ML approach used a transfer learning method with EfficientDet-Lite0 – Lite4 models. The choice of EfficientDet-Lite models for defect detection in this digital twin system is a good fit due to their high accuracy and low computational requirements. However, the choice of model will depend on the specific requirements of the system, such as accuracy, latency, and storage space. In addition, the performance of the models is affected by the quality of the images, which should be taken into consideration. Nevertheless, it was proposed to use EfficientDet-Lite2-3 as they are smaller in size than EfficientDet-Lite4 and show similar performance.

The integration of the Unity engine for creating the digital twins of a physical system that are highly realistic and immersive, correctly imitating the behaviour and performance of the original system, further enhancing the accuracy, efficiency, and usefulness of the proposed system. The Unity client user interface lets users talk to the digital twin, control the printer, and keep an eye on how it's doing in a user-friendly way.

In this research, the digital twin is fed in real-time using varieties of data obtained from 3D printers, including nozzle and bed temperatures, Gcode, printing time estimates, and others. A camera can also be used for a number of other things, including providing video and image data to the machine learning component for fault identification and enabling remote access for real-time monitoring outside of the laboratory. The integration of additional printing process data, such as temperature and vibrations, along with image and video data, will be investigated as part of future studies and the project's continuation. The machine learning algorithm's accuracy and detectability are intended to be improved by this integration. The initiative aims to increase overall performance and contribute to the creation of more effective and efficient systems by combining these developments.

The printed objects may also contain various defects due to variations in the materials used, printing techniques, and process conditions. The printing materials utilized in our study, including their features and characteristics, are crucial factors that affect the efficiency of the digital twin system. In our experiment with the ML model, we used simple cubic from PLA material to train the algorithm. It is important to mention that relying exclusively on image processing may not completely reflect the nature of faults. Therefore, it is necessary to implement a multifaceted strategy for defect detection, considering not just visual inspections but also incorporating data from sensors and other pertinent sources (e.g., bed temperature, nozzle temperature, and vibration). This strategy will be implemented in future research work.

This project is a step toward the development of digital twin solutions.

VI. CONCLUSION AND FUTURE WORK

The proposed digital twin system for additive manufacturing that integrates machine learning models for defect detection and a Unity client user interface provides operators with immediate feedback by detecting defects in real time, allowing adjustments before defective products are produced. The use of machine learning techniques, including data cleaning, feature extraction, and transfer learning, helped improve the models' accuracy, demonstrating the potential of machine learning in defect detection for additive manufacturing. Integrating the Unity engine further enhances the proposed system's accuracy, efficiency, and usefulness. Despite the research progress made as a step towards the development of digital twin solutions, challenges and limitations still need to be solved.

This work is part of an ongoing research programme which aims to improve the proposed system for defect detection in additive manufacturing processes using the digital twin with machine learning algorithms. One of the challenges is connecting to different wireless networks, which can cause connectivity issues and delay in bidirectional communications. To address this challenge, future work can focus on developing a more robust and reliable wireless connection system, possibly using mesh networks or other technologies that can ensure seamless connectivity across different networks.

The integration of more sophisticated machine learning algorithms in digital twin implementation systems requires further investigation. While the EfficientDet-Lite models used in this study showed promising results, more complex geometries and defect types may require more advanced machine-learning models. Future work can focus on exploring other state-of-the-art algorithms, such as deep learning techniques, that can improve the accuracy and efficiency of defect detection in additive manufacturing processes.

Data augmentation with random brightness and contrast can be used to increase the variability of the training data, which can improve the generalization capability of the machine learning models. Future research work can focus on implementing more advanced data augmentation techniques to enhance the performance of the models.

Incorporating Augmented Reality (AR) for interactive use can be another area of future work. AR can provide an immersive and interactive experience for operators to visualize and manipulate the digital twin model, allowing for better control and monitoring of the additive manufacturing process.

Finally, developing an automatic two-way communication system can help minimize human intervention and improve the efficiency of the defect detection process. Further research efforts should focus on developing a system that can automatically detect defects and stop the printing process or send notifications to the operator to take necessary actions. Moreover, such a system can also provide feedback to the machine learning models, allowing for continuous learning and improvement of the models over time.

REFERENCES

[1] M. Perno, L. Hvam, and A. Haug, "Implementation of digital twins in the process industry: A systematic literature review of enablers and barriers," *Comput. Ind.*, vol. 134, Jan. 2022, Art. no. 103558, doi: 10.1016/j.compind.2021.103558.

[2] M. Pant, P. Pidge, L. Nagdeve, and H. Kumar, "A review of additive manufacturing in aerospace application," *Revue des Composites et des Matériaux Avancés-J. Compos. Adv. Mater.*, vol. 31, no. 2, pp. 109–115, Apr. 2021, doi: 10.18280/rcma.310206.

[3] R. K. Phanden, S. V. Aditya, A. Sheokand, K. K. Goyal, P. Gahlot, and A. Jacso, "A state-of-the-art review on implementation of digital twin in additive manufacturing to monitor and control parts quality," *Mater. Today Proc.*, vol. 56, no. 2, pp. 88–93, 2022, doi: 10.1016/j.matpr.2021.12.217.

[4] R. Wagner, B. Schleich, B. Haefner, A. Kuhnle, S. Wartzack, and G. Lanza, "Challenges and potentials of digital twins and industry 4.0 in product design and production for high performance products," *Proc. CIRP*, vol. 84, pp. 88–93, 2019, doi: 10.1016/j.procir.2019.04.219.

[5] V. Kadam, S. Kumar, and A. Bongale, "Real-time monitoring of FDM 3D printer for fault detection using machine learning: A bibliometric study," *Library Philosophy Pract.*, vol. 5440, pp. 1–21, 2021. [Online]. Available: https://digitalcommons.unl.edu/libphilprac

[6] M. Kalender, S. E. Kiliç, S. Ersoy, Y. Bozkurt, and S. Salman, "Additive manufacturing and 3D printer technology in aerospace industry," in *Proc. 9th Int. Conf. Recent Adv. Space Technol. (RAST)*, Jun. 2019, pp. 689–694, doi: 10.1109/RAST.2019.8767881.

[7] L. Scime, A. Singh, and V. Paquit, "A scalable digital platform for the use of digital twins in additive manufacturing," *Manuf. Lett.*, vol. 31, pp. 28–32, Jan. 2022, doi: 10.1016/j.mfglet.2021.05.007.

[8] A. Kantaros, D. Piromalis, G. Tsaramirsis, P. Papageorgas, andH. Tamimi, "3D printing and implementation of digital twins: Current trends and limitations," *Appl. Syst. Innov.*, vol. 5, no. 1, pp. 1–13, 2022, doi: 10.3390/asi5010007.

[9] A. D. Khandare, V. S. Jadhav, S. P. Deshmukh, and K. K. Rane, "Digital twins for manufacturing process and system: A state of the art review," *ECS Trans.*, vol. 107, no. 1, pp. 15497–15508, 2022, doi: 10.1149/10701.15497ecst.

[10] Y. Lu, C. Liu, K. I.-K. Wang, H. Huang, and X. Xu, "Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues," *Robot. Comput.-Integr. Manuf.*, vol. 61, Feb. 2020, Art. no. 101837, doi: 10.1016/j.rcim.2019.101837.

[11] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21980–22012, 2020, doi: 10.1109/ACCESS.2020.2970143.

[12] T. O. R. F. Shimamura and D. Timper, "A feasibility study of digital twin for additive manufacturing from the perspective of resources efficiency," School Ind. Eng. Manag., KTH Roy. Inst. Technol., Sweden, 2021. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1584151/FULLTEXT01.pdf

[13] P. M. Bhattarai, P. Shrestha, and R. Chohan, "Digital twin and its implementation in 3D printing: A research review," *Int. J. Sci. Res. Manage.*, vol. 10, no. 7, pp. 49–68, Jul. 2022, doi: 10.18535/ijsrm/v10i07.ms01.

[14] A. Hyre, G. Harris, J. Osho, M. Pantelidakis, K. Mykoniatis, and J. Liu, "Digital twins: Representation, replication, reality, and relational (4Rs)," *Manuf. Lett.*, vol. 31, pp. 20–23, Jan. 2022, doi: 10.1016/j.mfglet.2021.12.004.

[15] U. Delli and S. Chang, "Automated process monitoring in 3D printing using supervised machine learning," *Proc. Manuf.*, vol. 26, pp. 865–870, 2018, doi: 10.1016/j.promfg.2018.07.111.

[16] C. M. Henson, N. I. Decker, and Q. Huang, "A digital twin strategy for major failure detection in fused deposition modeling processes," *Proc. Manuf.*, vol. 53, pp. 359–367, 2021, doi: 10.1016/j.promfg.2021.06.039.

[17] D. Mourtzis, T. Togias, J. Angelopoulos, and P. Stavropoulos, "A digital twin architecture for monitoring and optimization of fused deposition modeling processes," *Proc. CIRP*, vol. 103, pp. 97–102, Jan. 2021, doi: 10.1016/j.procir.2021.10.015.

[18] L. Yi, M. Glatt, S. Ehmsen, W. Duan, and J. C. Aurich, "Process monitoring of economic and environmental performance of a material extrusion printer using an augmented reality-based digital twin," *Additive Manuf.*, vol. 48, Dec. 2021, Art. no. 102388, doi: 10.1016/j.addma.2021.102388.

[19] C. S. Paripooranan, R. Abishek, D. C. Vivek, and S. Karthik, "An implementation of AR enabled digital twins for 3-D printing," in *Proc. IEEE Int. Symp. Smart Electron. Syst. (iSES)*, Dec. 2020, pp. 155–160, doi: 10.1109/iSES50453.2020.00043.

[20] C. A. Odada, J. B. Byiringiro, and F. M. Madaraka, "Development of data-driven digital twin for real-time monitoring of FDM 3D printer," *J. Mech. Eng. Automat.*, vol. 2021, no. 2, pp. 25–35, 2021, doi: 10.5923/j.jmea.20211002.01.

[21] M. Pantelidakis, K. Mykoniatis, J. Liu, and G. Harris, "A digital twin ecosystem for additive manufacturing using a real-time development platform," *Int. J. Adv. Manuf. Technol.*, vol. 120, nos. 9–10, pp. 6547–6563, Jun. 2022, doi: 10.1007/s00170-022-09164-6.

[22] P. Stavropoulos, A. Papacharalampoloulos, and K. Tzimanis, "Design and implementation of a digital twin platform for AM processes," *Proc. CIRP*, vol. 104, pp. 1722–1727, Nov. 2021, doi: 10.1016/j.procir.2021.11.290.

[23] K. Sandhu, S. Singh, and C. Prakash, "Analysis of angular shrinkage of fused filament fabricated poly-lactic-acid prints and its relationship with other process parameters," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 561, no. 1, Oct. 2019, Art. no. 012058, doi: 10.1088/1757-899X/561/1/012058.

[24] J. M. Gardner, K. A. Hunt, A. B. Ebel, E. S. Rose, S. C. Zylich, B. D. Jensen, K. E. Wise, E. J. Siochi, and G. Sauti, "Machines as craftsmen: Localized parameter setting optimization for fused filament fabrication 3D printing," *Adv. Mater. Technol.*, vol. 4, no. 3, pp. 1–10, 2019, doi: 10.1002/admt.201800653.

[25] V. Kadam, S. Kumar, A. Bongale, S. Wazarkar, P. Kamat, and S. Patil, "Enhancing surface fault detection using machine learning for 3D printed products," *Appl. Syst. Innov.*, vol. 4, no. 2, p. 34, May 2021, doi: 10.3390/asi4020034.

[26] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, Jun. 2019, pp. 10691–10700.

[27] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 10778–10787, doi: 10.1109/CVPR42600.2020.01079.

[28] D. A. J. Brion and S. W. Pattinson, "Generalisable 3D printing error detection and correction via multi-head neural networks," *Nature Commun.*, vol. 13, no. 1, p. 4654, Aug. 2022, doi: 10.1038/s41467-022-31985-y.

[29] K. Paraskevoudis, P. Karayannis, and E. P. Koumoulos, "Real-time 3D printing remote defect detection (stringing) with computer vision and artificial intelligence," *Processes*, vol. 8, no. 11, p. 1464, Nov. 2020, doi: 10.3390/pr8111464.

[30] H. Shen, W. Du, W. Sun, Y. Xu, and J. Fu, "Visual detection of surface defects based on self-feature comparison in robot 3-D printing," *Appl. Sci.*, vol. 10, no. 1, p. 235, Dec. 2019, doi: 10.3390/app10010235.

[31] M. Verana, C. I. Nwakanma, J. M. Lee, and D. S. Kim, "Deep learning-based 3D printer fault detection," in *Proc. 12th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jeju Island, Republic of Korea, Aug. 2021, pp. 99–102, doi: 10.1109/ICUFN49451.2021.9528692.

[32] W. Cui, Y. Zhang, X. Zhang, L. Li, and F. Liou, "Metal additive manufacturing parts inspection using convolutional neural network," *Appl. Sci.*, vol. 10, no. 2, p. 545, Jan. 2020, doi: 10.3390/app10020545.

[33] H. Jeong and J. Yoo, "3D-EDM: Early detection model for 3D-printer faults," 2022, *arXiv:2203.12147*.

[34] B. Li, L. Zhang, L. Ren, and X. Luo, "3D printing fault detection based on process data," in *Proc. Chin. Intell. Syst. Conf.*, Oct. 2018, pp. 385–396, doi: 10.1007/978-981-13-2291-4_38.

NURSULTAN JYENISKHAN received the B.S. degree in mechanical engineering from Marmara University, Istanbul, Turkey, in 2015. He is currently pursuing the M.S. degree in mechanical and aerospace engineering with Nazarbayev University, Astana, Kazakhstan.

From 2016 to 2020, he was a mechanical engineer in manufacturing industry. Since 2021, he has been a Research Assistant with the School of Engineering and Digital Sciences, Nazarbayev University. His research interests include digital twin, digital manufacturing, additive, and sustainable engineering.

AIGERIM KEUTAYEVA received the B.S. degree in robotics and mechatronics from Nazarbayev University, Astana, Kazakhstan, in 2021, where she is currently pursuing the M.S. degree.

Since 2019, she has been a Research Assistant with the School of Engineering and Digital Sciences, Nazarbayev University, and a member of Young Researchers Alliance, Astana. Her research interests include machine learning, brain–computer interfaces, and digital twin.

GANI KAZBEK received the B.S. degree in civil and environmental engineering from Nazarbayev University, Astana, Kazakhstan, in 2022, where he is currently pursuing the M.S. degree in engineering management.

He has experience and knowledge in the field of digital solutions and development of digital twins. He participates to many forums in Kazakhstan related to digital solutions in the industry.

MD HAZRAT ALI (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in mechatronics engineering from IIUM, Malaysia, and the Ph.D. degree in mechanical engineering from Kyushu University, Japan.

His Ph.D. thesis focused on developing integrated control systems in precision measurement for variable deep-holes applying laser beam, optical components, CCD camera, servo, and stepper motors. He has attracted over 1.0 million dollar in research grants funded by internal and external organizations. He has published over 100 scientific papers in high-impact journals, top-tier conferences, and symposiums. He has supervised over 40 students, including Ph.D. and M.Sc. thesis and capstone projects.

ESSAM SHEHAB is currently the Head of the Mechanical and Aerospace Engineering Department, Nazarbayev University, Kazakhstan. He is also a Visiting Professor with Cranfield University, U.K. He has sustained his strong research leadership with world-leading aerospace companies, including Airbus and Rolls-Royce. He secured 15.91 m research funding in cash from government grants (U.K., EU, and KZ) and industry. He was invited to provide plenary and keynote presentations in international conferences. He generated 310 research publications which have received over 11,500 Citations. He has successfully completed the supervision of 31 Ph.D. researchers and 110 M.Sc. theses. He also served as the external examiner for over 30 Ph.D. theses at U.K., United Arab Emirates, Singapore, and Ireland universities.

He is also a Chartered Engineer (C.Eng.) and a fellow of the U.K. Higher Education Academy (HEA), the Institute of Engineering and Technology (IET), and the Association of Cost Engineers (ACostE).