A closed-loop in-process warping detection system for fused filament fabrication using convolutional neural networks

Aditya Saluja, Jiarui Xie, Kazem Fayazbakhsh *

Aerospace Engineering Department, Ryerson University, 350 Victoria Street, Toronto, ON, Canada M5B 2K3
**Corresponding author: kazem@ryerson.ca; Tel: (+1) 416-979-5000 ext. 6414; fax: (+1) 416-979-5056*

Abstract

Fused Filament Fabrication (FFF) is an additive manufacturing technology that can produce complicated structures in a simple-to-use and cost-effective manner. Although promising, the technology is prone to defects, e.g. warping, compromising the quality of the manufactured component. To avoid the adverse effects caused by warping, this paper utilizes deep-learning algorithms to develop a warping detection system using Convolutional Neural Networks (CNN). To create such a system, a real-time data acquisition and analysis pipeline is laid out. The system is responsible for capturing a snapshot of the print layer-by-layer and simultaneously extracting the corners of the component. The extracted region-of-interest is then passed through a CNN outputting the probability of a corner being warped. If a warp is identified, a signal is sent to pause the print, thereby creating a closed-loop in-process detection system. The underlying model is tested in an experimental set-up yielding a mean accuracy of 99.3%.

Keywords: Convolutional Neural Networks (CNN); Fused Filament Fabrication (FFF); warping detection; image analysis.

Journal of Manufacturing Processes 58 (2020) 407-415
https://doi.org/10.1016/j.jmapro.2020.08.036

1. Introduction

The emergence and development of Additive Manufacturing (AM) technologies reveal more potentials of cost-effective and sustainable manufacturing [1]. Additive Manufacturing, also known as 3D Printing, constructs a structure in a layer-by-layer methodology, directly guided by computer-aided design (CAD) data models [2]. Distinct from conventional subtractive manufacturing techniques, AM technologies can produce components with high structural complexity, as well as conserving a considerable amount of material [3]. The materials available for Additive Manufacturing methods include plastics, metals, ceramics, etc., adopted by various 3D Printing techniques such as Fused Filament Fabrication (FFF), Selective Laser Sintering (SLS), and Stereolithography (SLA) [4]. FFF technique is the most widely utilized technology for both industrial and small-scale prototyping, as it can be highly user-friendly, automatic, and safe [5, 6]. In this process, an extruder continuously feeds filament into a heated nozzle and follows the printing paths defined by slicing software [7]. The thermoplastic filament is melted, deposited along the designated path, and cooled down to the solid-state. Once one layer is completed, the build platform or extruder will adjust its height and the next layer will be printed upon the previous ones [8, 9].

Although FFF 3D printers offer an automated manufacturing technique to reduce time, cost, and material usage, they induce defects such as gaps, overlaps, and over/under extrusion into final parts. In addition, warping can happen during 3D printing that can result in low dimensional accuracy, print failure, and even a damage to the 3D printer [10]. After material deposition, since the cooling rate is not uniform throughout the print, the temperature gradient in the part creates residual thermal stress [11, 12]. As the number of deposited layers increase, the residual thermal stress builds up in the part and might distort it when the adhesion between the part and the build platform is not enough to overcome the bending force. Warping usually starts from the corners and results in the part being peeled away from the build platform [13].

Non-Destructive Testing (NDT) technologies have been utilized and incorporated into AM processes to avoid costly failures caused by warping and other defects. Lu et al. [14] reviewed the application of two in-process defect detection methods in AM techniques, i.e. thermography and acoustic emission (AE) testing. They discussed the suitability of these two methods and presented their advantages and disadvantages. Liu et al. [15] designed an in-situ inspection system for FFF 3D printers using AE sensors. The system can reliably detect machine state errors such as extruder jam and material run-out, and part defect such as warping. However, AE sensors must be attached to the build platform and extruder block; therefore, a method that can offer contactless defect detection would suffer from fewer restrictions than AE sensors. Other researchers used Digital Image Correlation (DIC) technique to create CAD models of 3D printed parts and identify their defects by comparing as-designed and as-manufactured models. Holzmond et al. [16] utilized a three-dimensional DIC system that incorporated two cameras to take images and constructed a data model of the print. The data model was then compared with a point cloud containing the information of the design model extracted from its G-code to check for anomalies in the printed part. This work did not provide sufficient test data to demonstrate the reliability of the 3D-DIC system or its accuracy. Furthermore, this technique involved intensive data transformation and simulation, which required high computational power and lowered the speed of defect detection.

In recent years, with the advancements of CPU and GPU performance, advanced image processing and computer vision have been deployed for in-situ defect detection of 3D printed parts. Baumann et al. [18] utilized image processing to detect missing material and movement caused by the deformation of 3D-printed components. The parts were recognized using a coloring threshold, followed by a binary difference image to identify defects. The model accuracy was severely affected by variations in lighting conditions and the color of the filament limiting the practicability of the monitoring system [18]. A similar approach was used by Yi et al. [19], wherein a camera was utilized to record a print, layer by layer. The contours of each image were then extracted through image processing and analyzed using statistical process control to find defects. Currently, potential defects in 3D printed parts are not well defined in AM standards and in-process quality inspection systems are not established [17]. Thus, a method to perform in-process automatic defect detection that can improve itself with data accumulation over time is desired.

Machine learning evolves with data accumulation and adds flexibility by adapting to different industries and standards based on the manufacturing environment. Three categories of machine learning algorithms have been applied to in-situ defect detection and quality enhancement of AM: classification, regression, and clustering [20]. In the classification category, the input data is divided and labelled based on normal and failure modes. Li et al. [21] designed an in-situ monitoring and diagnosing system for FFF 3D printers with vibration sensors attached to the build platform and extruder block. Some features such as root mean square and standard deviation were extracted from the acceleration data to train the support vector machine (SVM) and back-propagation neural network (BPNN) for the detection of machine state errors and part defects, respectively. This system has more than 95% accuracy in filament jam detection and 97% accuracy in warpage and filament stack detection. Regression models are used in process parameters optimization and in-process adjustment to improve print quality. Banadaki et al. [22] developed a regression model that predicts the print quality and adjusts the process parameters in real-time. This Convolutional Neural Network (CNN) model was trained by five thousand images of parts being printed with different extruder temperatures between 185 ℃ and 260 ℃ and printing speed between 50 mm/s and 1000 mm/s. The pictures were manually assessed and labelled from A to D as the overall print quality decreased. This system achieved an accuracy of 94% when predicting the print quality in terms of printing speed and extruder temperature. Contrary to supervised machine learning, which requires the user to divide the dataset, clustering autonomously groups the data inputs regarding their similarities for classification problems [23]. It helps to avoid biases and reveals more potentials of a monitoring system by avoiding the assumptions of detectable defects made by the designers. After investigating AE aided monitoring systems for FFF 3D printers utilizing machine learning algorithms including SVM [24], k-means clustering [25] and CFSFDP clustering [26], Wu et al. [27] used self-organizing map (SOM) to diagnose the failure modes. This monitoring system had a differential wide-band AE sensor attached to the build platform. The acoustic waves were transformed into AE hits, which are segments of waves recorded when the voltage of the AE sensor exceeds a threshold. The features such as amplitude and frequency were extracted from the AE hits to train the SOM clustering model. It was discovered that these features extracted successfully divided the dataset into multiple failure modes, including extruder scratching, warping and material sliding. Considering the above research works, there has not been a defect monitoring system that combines contactless warping monitoring (e.g. vision-based systems) with neural networks (NN).

Multilayer neural networks, consisting of input layer, hidden layer, and output layer, belong to supervised machine learning [28]. Each layer has neurons that each represents a nonlinear and parameterized function. It has been proven that a NN with at least one hidden layer can approximate any measurable function [29].

Thus, a NN with many hidden layers can approximate the manufacturing process of a 3D printer, despite the high complexity brought by process parameters, geometry parameters, and in-situ monitoring data [20, 30]. CNN earns its popularity in the domains of image analysis and AM defect detection as it accounts for the spatial relationship of pixels [31]. B. Zhang et al. and Z. Zhang et al. [32-33] inspected defects in welded components through Convolutional Neural Networks (CNN), and explored the versatility and potential of machine learning in manufacturing process inspection. They suggested that CNN can be used for in-process quality control in various AM techniques. Caggiano et al. [34] developed a CNN classification model that recognizes the in-process images of defective Selective Laser Melting (SLM) specimens due to non-ideal energy level with 99.4% accuracy, including discontinuities, delamination, and non-densification. 600 pairs of powder layer and part cross-section images were captured as training and testing sets through the printing processes of four specimens. Skip connections were adopted to avoid vanishing gradients and retain the information richness of input images. Scime and Beuth [35] generated an SLM quality monitoring system comprised of a CNN classification model and a defect visualization application. This system analyzed more than 70,000 images and showed an overall accuracy of 97% when diagnosing many defects such as debris and recoater streaking. Thus, a vision-based monitoring system with a CNN model exhibits promising potentials in warping detection for FFF 3D printing.

This investigation focuses on developing a robust closed-loop in-process method to detect warped corners in FFF 3D printed components using CNN. In the following sections, first a generalized CNN classification algorithm is proposed, which is capable of detecting warped corners in FFF 3D printed components in real time. Then, the impact of hyperparameters on the performance, robustness, and generalization of the classification model is studied. Next, a closed-loop architecture is laid out, which is capable of pausing a print once a warp deformation is detected. Finally, a discussion of the closed-loop system accuracy is presented before conclusions and recommendations for future work.

1.1. Theory

Convolutional Neural Network (CNN) is a subset of artificial neural networks that are commonly used for complex machine learning applications such as image classification. Traditionally, Multilayer Perceptron (MLP) networks were used for image classification, where the presence of fully connected nodes added computational complexities and disregarded spatial information leading to scalability issues. In contrast, a CNN divides an image into smaller portions extracting features through a sparsely connected arrangement of convolutional, pooling, and fully connected layers.

A convolution layer comprises a series of learnable filters known as kernels. A kernel is a matrix convolved over a subset of input pixels to produce a feature map. The creation of multiple feature maps through distinct filters in each convolutional layer help extract different features from the input image. A convolutional layer can be mathematically represented as:

$$a_{ij}^{k} = f\left(\left(W^{k} \times x\right)_{ij} + b_{k}\right) \tag{1}$$

where a_{ij}^{k} represents the k^{th} feature map in a given layer. The feature map is calculated by convolving the weight matrix, W^{k}, with a set of input pixels represented by x_{ij}. The Bias, b_{k}, is added to the convolved matrix and is passed through an activation function $f(\cdot)$ adding nonlinearity to the system. As the classification function modeled through the neural network is unlikely to have a linear relation with the input, nonlinearity generalizes the system enhancing the performance of the neural network.

Depending on the complexity of classification, various activation functions can be utilized. This paper utilizes the Leaky Rectified Linear Units (Leaky-ReLU) activation function for its simplicity, scalability, and resistance towards vanishing gradient problem. Mathematically, Leaky-ReLU is defined as:

$$f(x) = \begin{cases} 0.01x, x < 0 \\ x \quad\quad , x \geq 0 \end{cases} \tag{2}$$

Typically, a convolution layer is followed by a pooling layer. Feature maps are down-sampled using max pooling to reduce dimensionality and to make features invariant to small translations. This way, overall performance of the model is improved and at the same time there is a reduction in computational complexity.

The extracted features are then flattened and passed through fully connected layers completing the architecture of a typical CNN. The layers are simultaneously trained through a backpropagation algorithm, which optimizes the weight matrix and the biases by minimizing the loss function. For the purpose of this paper a cross entropy loss function was used and is mathematically represented as:

$$H(p,q) = -\sum_{x} p(x) log\ q(x) \tag{3}$$

where $H(p,q)$ represents the cross entropy, $p(x)$ is the predicted outcome, and $q(x)$ is the actual output of sample x.

2. Methodology

This section provides details regarding the experimental set-up, dataset preparation, and an in-depth analysis of the proposed CNN classification model.

2.1. Experimental set-up

Figure 1 shows the experimental setup used to create and validate the warping detection system for a commercial desktop 3D printer. All tests were carried out using a Prusa i3 MK2S 3D printer and the images were acquired using a camera (Sony A5100). The camera was positioned at the same level as the build platform to have a clear view of the specimen.

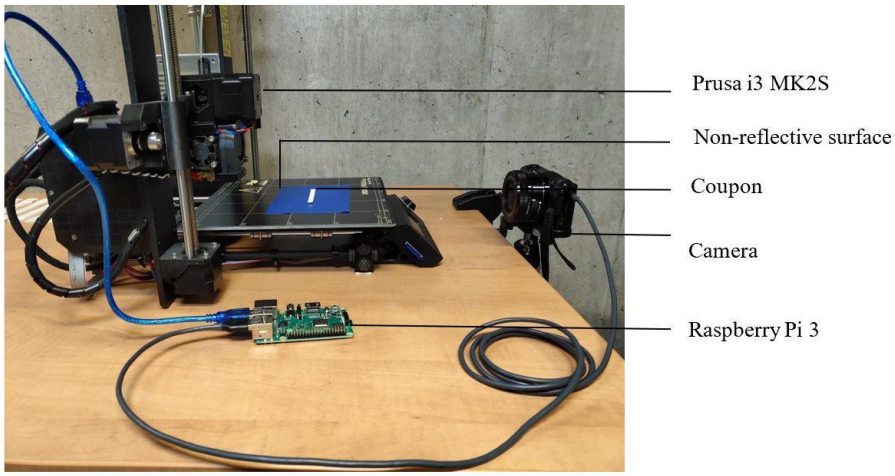


Figure 1. The experimental ste-up for creating and validating the detection system.

Figure 2 illustrates the proposed closed loop architecture for warping detection system. A Raspberry Pxsi is connected to the printer through the serial port and wirelessly interfaced with a computer using OctoPrint. Due to the computational limits of the Raspberry Pi, all scripts were executed on a local computer. These scripts were written in Python 3.6 and were executed on an environment of Intel® Core™ i5-6300U CPU @ 2.40GHz and 8GB memory running on Windows 10 64-bits. The neural network was programmed in a Python environment utilizing the Keras framework.

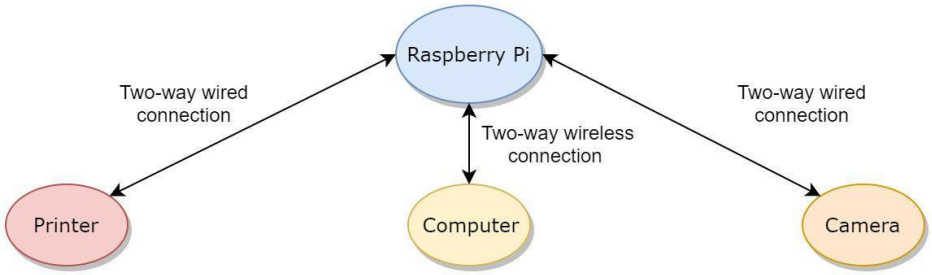


Figure 2. The proposed closed-loop architecture.

To have a clear view of the specimen corners, the G-Code was modified to pause the printer at the end of each layer and resume automatically after two-seconds. This allowed the extruder to move out of the frame, when a script triggered the camera. An additional script was then autonomously executed automatically to extract, resize, and grayscale the corners of the specimen from the raw image, and classify the extracted region using the CNN model (Figure 3). A predicted output value, P(X), below 0.5 from the CNN model indicates warping occurrence. In this case, a signal will be sent from the computer to the Raspberry Pi to pause the print, thereby creating a closed-loop in-process warping detection system. A P(X) value above 0.5 indicates an unwarped specimen as it can be seen for the printed part in Figure 3 (P(X)=0.98).

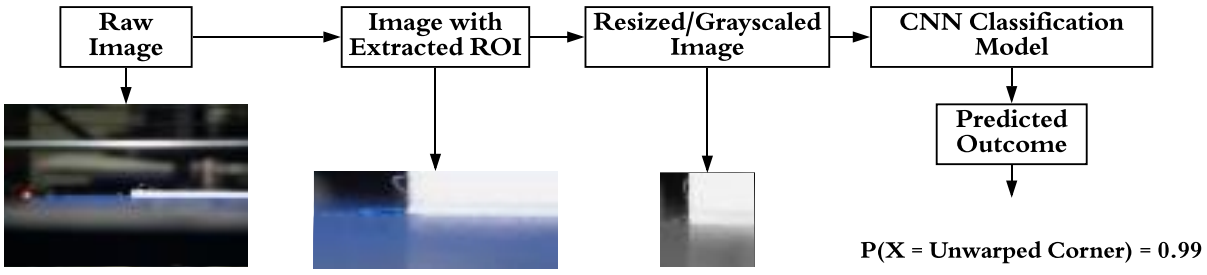


Figure 3. Image acquisition and analysis pipeline used for the classification model.

2.2. Dataset Preparation

The raw dataset consists of 520 colored, 6000 × 4000-pixel images divided equally into two classes. The images for unwarped corners were collected by printing 70 mm × 15 mm × 5 mm rectangular cuboids using the Prusa i3 MK2S. The images for warped corners were obtained by peeling a corner of specimens from the build platform during the print process and the average value of distortion was 4°.

Figure 4 shows three resized samples from the training set including unwarped and warped classes in the dataset. It should be noted that in the current set-up warping is considered as the part being peeled away from the build platform.

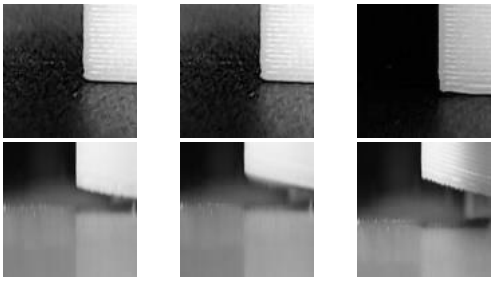


Figure 4. Resized samples from the training set: (a) unwarped; and (b) warped.

The raw data was then processed to diversify the dataset, which increased the generalization ability of the underlying CNN classification model and reducing computational complexity. The new dataset was then shuffled and randomly split into training and validation sets.

2.3. CNN classification model

This section describes the underlying CNN classification model including its architecture, hyperparameter selection, and optimization.

2.3.1. CNN Architecture

Table 1 presents the architecture of the CNN model used to detect warped corners in FFF 3D printed parts in real-time.

Table 1[1]: Summarized architecture of the proposed CNN classification model.

Layer	Operator	Kernel size	Stride	Number of filters/nodes/%	Method
LY1 – C1	Convolution	2 × 2	2	8	-
LY2 – P1	Pooling	2 × 2	2	-	Max Pooling
LY3 – C2	Convolution	2 × 2	2	8	-
LY4 – P2	Pooling	2 × 2	2	-	Max Pooling
LY5 – C3	Convolution	2 × 2	2	8	-
LY6 – P3	Pooling	2 × 2	2	-	Max Pooling
LY7 – C4	Convolution	2 × 2	2	16	-
LY8 – P4	Pooling	2 × 2	2	-	Max Pooling
LY9 – C5	Convolution	2 × 2	2	16	-
LY10 – P5	Pooling	2 × 2	2	-	Max Pooling
LY11 – C6	Convolution	2 × 2	2	16	-
LY12 – P6	Pooling	2 × 2	2	-	Max Pooling
LY13 – C7	Convolution	2 × 2	2	24	-
LY14 – P7	Pooling	2 × 2	2	-	Max Pooling
LY15 – C8	Convolution	2 × 2	2	24	-
LY16 – P8	Pooling	2 × 2	2	-	Max Pooling
LY17 – C9	Convolution	2 × 2	2	24	-

LY18 – P9	Pooling	2 × 2	2	-	Max Pooling
LY19 – FC1	Fully Connected	-	-	1024 (Nodes)	-
LY20 – DP1	Dropout	-	-	Retain 60% Nodes	-
LY21 – FC2	Fully Connected	-	-	2	SoftMax (Activation)

[1] LY is a shortened form for Layer. C, P, FC, and DP represent the different types of layers in a neural network, namely convolution, pooling, fully connected and dropout, respectively.

2.3.2. Input, Convolution Pooling, Fully Connected and Output Layers

As discussed in Section 2.2, the raw dataset consists of 520 colored images of high resolution (6000 × 4000 pixels for each color channel (RGB)). Although such resolutions could provide high classification accuracy, training a CNN classification model with the raw dataset would require optimizing 3.744×10^{10} nodes (520 × 6000× 4000× 3) reducing computational efficiency. To reduce the computational cost, a region of interest was extracted from the raw image (Figure 3). This region was then resized to an image of 100 × 100 pixels by performing the nearest-neighbor interpolation, thereby reducing the number of trainable nodes. The processed image was converted to grayscale to further reduce the number of input nodes by a factor of 10. The two classes, warped and unwarped corners, were one-hot encoded, shuffled, and split into training and validation sets of 416 and 104 samples, respectively.

The processed dataset was passed through the convolution layer LY1 – C1, wherein a kernel of size 2 × 2 was applied to the input image with a stride of 2. The stride represents the number of pixels to be skipped in both the horizontal and vertical directions while performing convolution. Applying multiple filters in each layer helps extract distinct features generalizing the input image. Eight unique kernels were applied to the input image in the first layer to produce eight smaller images. These smaller images were then passed through subsequent convolutional layers with additional filters as outlined in Table 1.

Although convolutional layers are useful in extracting essential features, they record their location in the input image. In the case of slight movements in the input image, the model fails to identify key features, thereby affecting the overall efficiency. To better generalize the model, each convolutional layer was followed by a pooling layer. As listed in Table 1, even layers from 2 to 18 perform max pooling of stride 2. Max pooling extracts the local maxima in a patch of pixels creating a down-sampled feature map invariant to feature shift and distortion. As a result, it improves the overall accuracy of the underlying model.

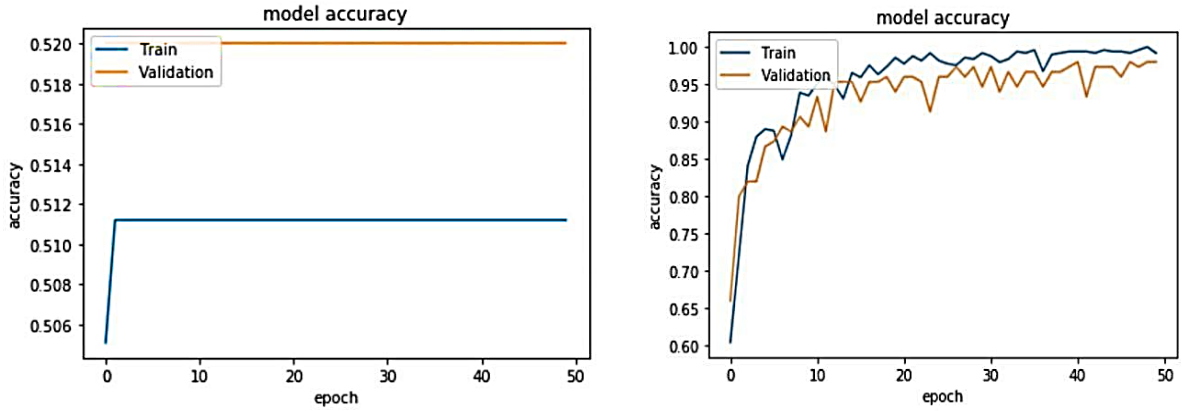
The last pooling layer, LY18 – P9, was followed by a fully connected layer, LY19 – FC1. Based on the last convolution and pooling layer, 24 images were produced from a single input image of size 100 × 100 pixels. The images were then flattened, creating a one-dimensional tensor, fully connected to 1024 nodes in the subsequent layer. As the underlying model performs a binary classification, the last hidden layer has two nodes, one for each class. Every node in layer LY19 – FC1 is fully connected to both nodes in layer LY21 – FC2 completing the overall architecture of the CNN classification model.

2.3.3. Activation, Optimization, Epoch, Backpropagation and Regularization

Once the architecture was laid out, all the layers were simultaneously optimized using a backpropagation algorithm. Before optimizing the underlying model, it is essential to select a suitable activation function. In an artificial neural network, a neuron is responsible for calculating the weighted sum of inputs, adding bias, and executing a function if a certain threshold is met. Based on the number of neurons activated, the model

classifies the input to one of the classes. For this paper, two activation functions were considered: Rectified Linear Units (ReLU), and Leaky Rectified Linear Units (Leaky ReLU).

To select an appropriate activation function, the model was trained on 520 training images for 50 epochs, which determines the number of times the optimization algorithm would iterate over the entire dataset. The number of epochs plays an essential role in determining the performance of the model. Thus, Early-Stopping was implemented, wherein the Callback class in Keras was utilized to prevent the model from training once a threshold for validation accuracy was achieved. The training and validation accuracy for both activation functions were then graphed and compared (Figure 5).



(a) **(b)**

Figure 5. Training and validation accuracy of the model with different activation functions: (a) ReLU; and (b) Leaky Re-LU.

As shown in Figure 5, ReLU provides a training and validation accuracy of 0.52 and 0.51, respectively. The poor performance could be attributed to high bias and the vanishing gradient problem, where the gradient of the error function with respect to the weights of the training data is infinitesimal. This prevents the weights from updating, thereby halting the model to train further. When Leaky-ReLU was used as the activation function for the classification model, training and validation accuracy of 0.98 and 0.96 was achieved, respectively.

The model is trained by minimizing the loss obtained by simultaneously optimizing and backpropagating the weights and biases of each layer. Figure 6 compares the evolution of loss function with the number of epochs minimized through three common optimization algorithms: Stochastic Gradient Descent (SGD), Root Mean Square (RMS), and Adaptive Moment Estimation (Adam) optimizers.

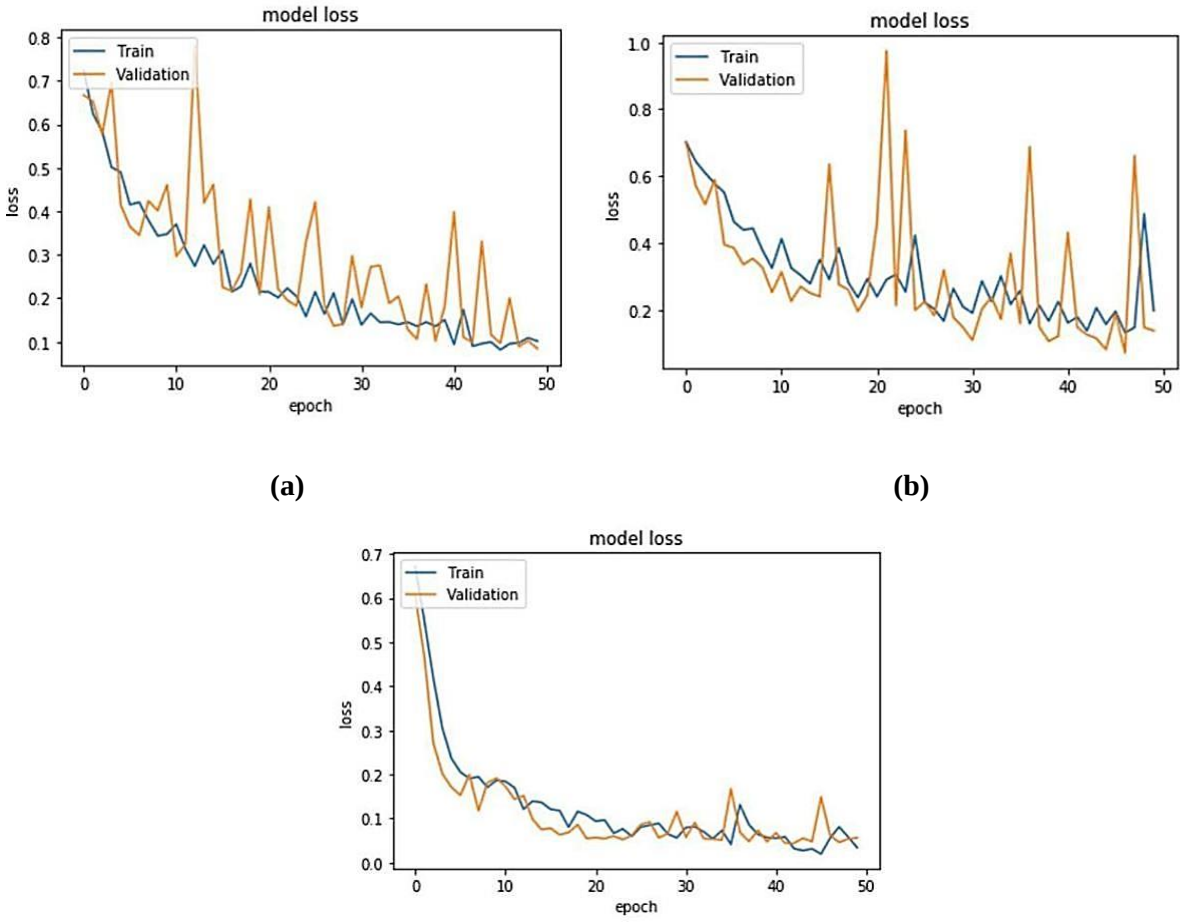
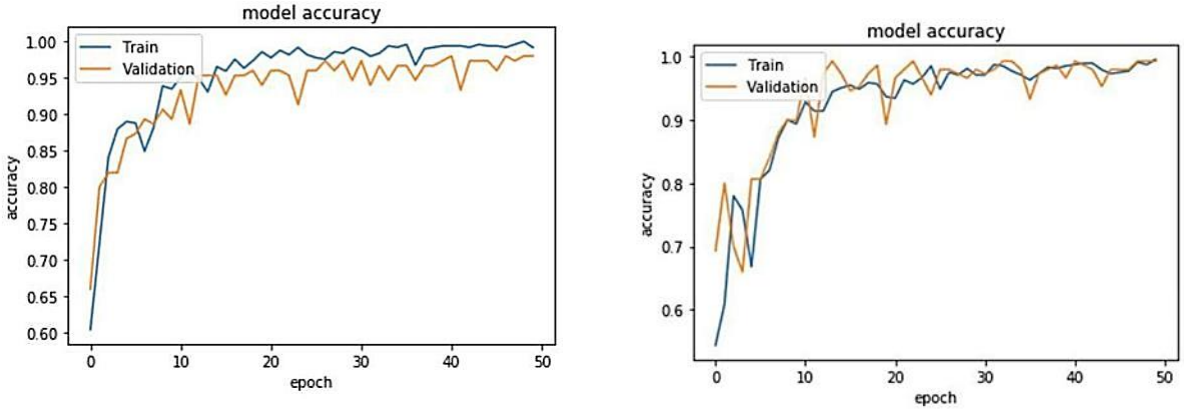


Figure 6. Comparison of loss function minimized by common optimizing algorithms with number of epochs: (a) SGD; (b) RMS; and (c) Adam Optimizers

Although all three algorithms minimize the loss function, SGD and RMS are prone to heavy validation fluctuations for the training dataset. In contrast, Adam optimizer successfully minimizes the error function to less than 10% with the least amount of validation fluctuations. The training set used for this study is small and the fluctuations are partly caused by a large network being trained on this small dataset. As SGD trains on the entire training set, it is often slow for larger datasets and is used in conjuncture with momentum to yield accurate results efficiently. The exact reason for choosing Adam over RMS is beyond the scope of this paper and is covered in [36]. Since Adam optimizer is frequently used in literature and is easier to implement, it was selected for this research work.

While certain models perform well during training, a discrepancy between the validation and training accuracy is often observed suggesting overfitting. As shown in Figure 7a, there is a difference in accuracy between training and validation that can be reduced by adding a dropout layer (Figure 7b). The dropout layer discards some nodes at random allowing the model to learn from all the inputs and not to over-rely on few of its inputs, thereby regularizing the neural network.



(a) **(b)**

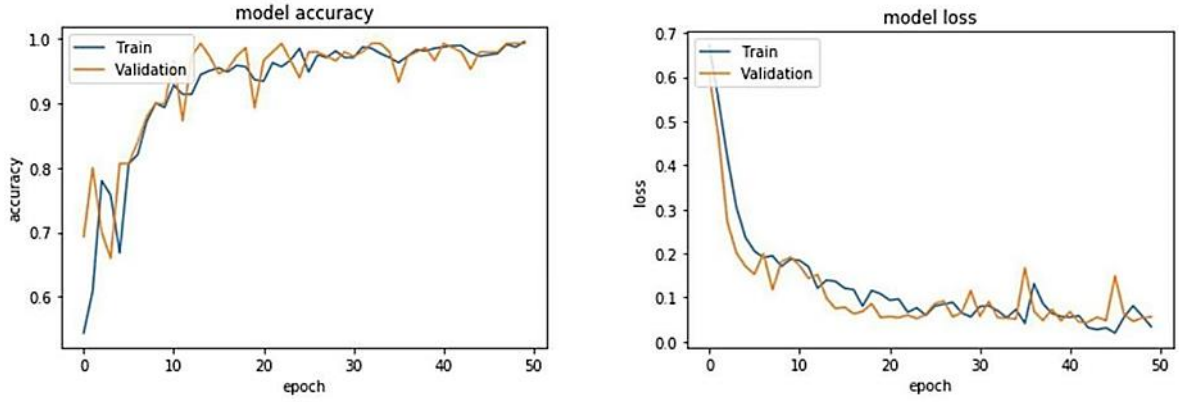
Figure 7. Model accuracy: (a) without dropout; and (b) with dropout.

3. Results and discussion

This section summarizes the results from the proposed CNN classification model and closed-loop in-process warping detection system.

3.1. The CNN classification model

After laying out the architecture of the CNN classification model, and selecting a suitable activation function, optimization algorithm, and regularization, the model was trained with 50 epochs. Figure 8 shows the model accuracy and decay of loss function with the number of epochs for the proposed CNN classification model.



(a) **(b)**

Figure 8: The proposed CNN classification model training: (a) accuracy; and (b) decay of loss function.

3.2. Warping detection system

To examine the underlying model, the experimental test set-up (Figure 1) was used and 70 mm × 15 mm × 5 mm rectangular cuboids were printed. The manufacturing process and design parameters for 3D printing are summarized in Table 2.

Table 2: 3D printing process parameters.

Manufacturing parameter	Value	Manufacturing parameter	Value
Print direction	XYZ	Nozzle diameter (mm)	0.4
Material	PLA	Nozzle temperature (°C)	215
Raster angle	0	Cooling	No fan cooling
Layer height (mm)	0.14	Infill (%)	100
Bed temperature (°C)	60	Filament diameter (mm)	1.75
Print speed (mm/min)	2400		

As defined in Table 1, the output layer utilizes a Softmax activation function, wherein logits from the last fully connected layer are converted to probabilities. A probability of greater than 0.5 corresponds to an unwarped corner. Using the architecture laid out in Figures 2, the results obtained for a sample print are summarized in Table 3 with the accompanying plots in Figure 9. It can be observed that layers 2 through 24 were categorized as unwarped (P(X)> 0.5), whereas layer 25 was classified as a warped corner (P(X)< 0.5).

Table 3: Predicted output in the warping detection system for a specimen printing.

Layer #	$P(X = Unwarped\ Corner)$	Layer #	$P(X = Unwarped\ Corner)$
2	0.79590	14	0.98971
3	0.91555	15	0.98044
4	0.98808	16	0.98572
5	0.95081	17	0.99264
6	0.96677	18	0.99149
7	0.94187	19	0.99372
8	0.98959	20	0.99214
9	0.99174	21	0.99657
10	0.99476	22	0.99613
11	0.99119	23	0.99620
12	0.99327	24	0.99370
13	0.98547	25	0.00004

The classification results (Table 3) match exactly the actual print depicted in Figure 9. Once the warpage was detected, the scripts were terminated, and a signal was sent to pause the printing.

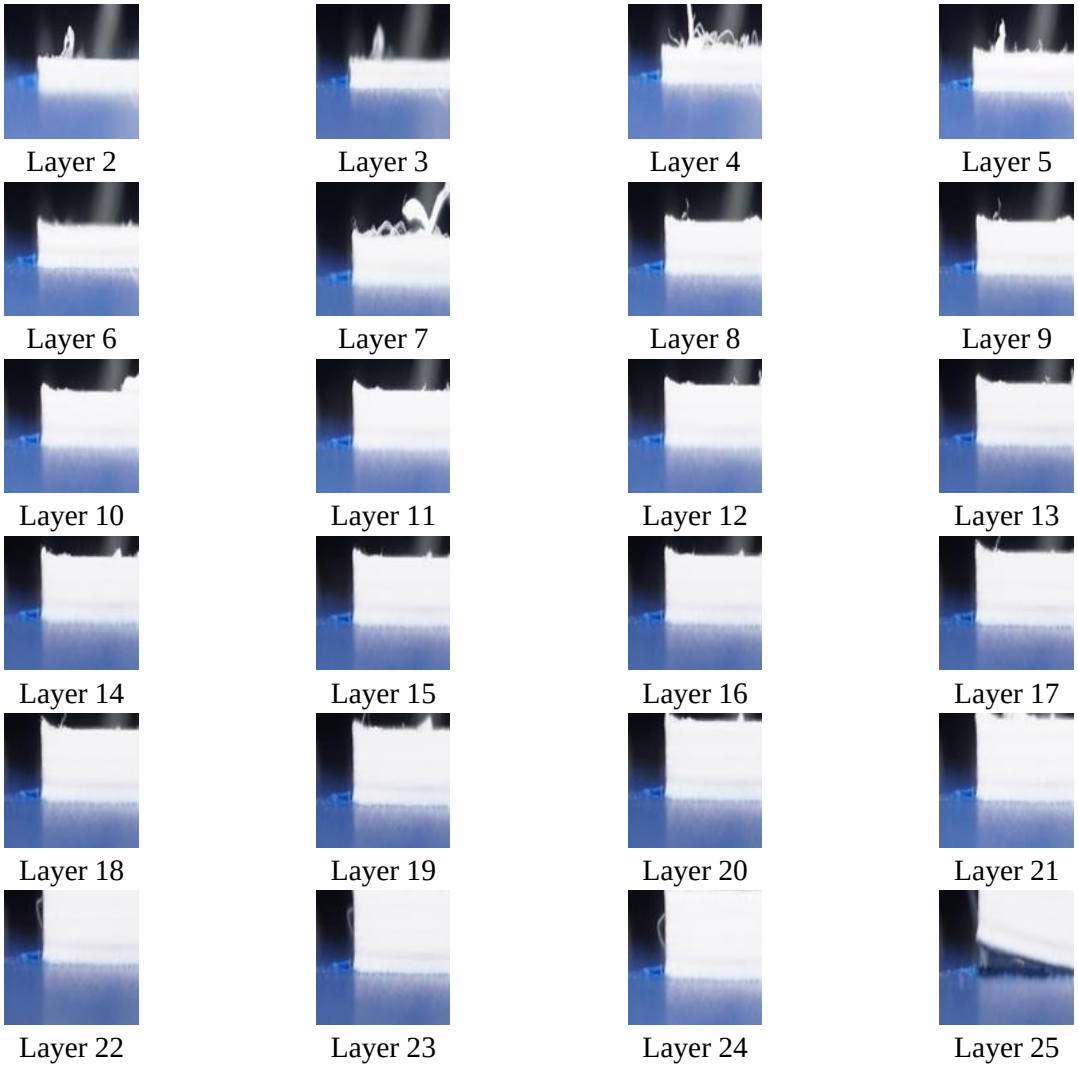


Figure 9: A subplot of different layers of a sample print showing warping in layer 25.

In addition to rectangular cuboids, triangular prism (base = 48 mm, height = 48 mm, and length = 5 mm) and cylinders (20 mm in radius and 50 mm in length) were 3D printed in the experimental test set-up (Figure 1). The experiments were repeated three times for each geometry and the results are summarized in Table 4. As soon as the onset of part warping was detected, the printing process was paused automatically. As a result, even for the same geometry, total number of printed layers in Table 4 differs among tests.

Table 4: The CNN classification model results in the experimental test set-up.

Test geometry	Test number	Total number of printed layers	Number of unwarped layers	Number of warped layers	Layers correctly classified
Triangular Prism	1	13	12	1	13
	2	13	12	1	13
	3	21	20	1	21
Cylinder	1	10	9	1	10
	2	16	15	1	15
	3	27	26	1	27
Cuboid	1	10	9	1	10
	2	20	19	1	20
	3	24	23	1	24

As seen in Table 4, the CNN classification model was tested on 154 images taken during 3D printing and 153 of them were identified correctly, which means an accuracy of 99.3% for the model. For test #2 in the cylindrical specimens (Table 4), layer #15 started to peel from the build platform (Figure 10a) and was not recognized as warped until layer #16 was printed (Figure 10b). Compared to rectangular profiles (the training dataset), a warped cylindrical corner has less shadow on the build platform induced by greater edge curvature, especially for slight warping (Figure 10a). Features such as shadow and curvature extracted by filters in the CNN layers are essential evidence of the classification model to recognize warped corners. The CNN model misclassified layer #15 image as unwarped since it is near the boundary between two classes.

(a) **(b)**

Figure 10: Test number 2 in cylindrical specimens: (a) layer #15; and (b) layer #16.

Although the training dataset contained images of rectangular profiles, the model managed to identify warping in the triangular prisms and cylinders. This validates that the classification model had correctly been trained without overfitting the dataset. It should be noted that 3D printing and experimental tests were conducted in an ideal environment. The specimens were printed on a non-reflective surface, the lighting conditions were closely monitored, and the code was executed after the first layer had been printed.

4. Conclusions

This paper presented the application of Convolutional Neural Networks (CNN) in a detection system to identify the onset of part warping during Fused Filament Fabrication (FFF) 3D printing. The impact of activation function, optimization, and regularization on the overall accuracy of the classification model has been investigated. By changing the activation function from ReLU to Leaky-ReLU, the training and validation mean accuracy increased from 52% to 98%, rspectively. Three commonly used optimization algorithms were compared in this paper: Stochastic Gradient Descent (SGD), Root Mean Square (RMS), and Adaptive Moment Estimation (Adam). Although SGD and RMS optimization algorithms were able to

minimize the cost function, the heavy fluctuations made them unsuitable for the application. In contrast, Adam optimizer had a smooth decay of losses and minimized the error function to less than 10%; therefore, it was selected for the classification model. In the model, a dropout layer between the fully connected layers was added, which caused a consistent validation and training accuracy of 97%. An experimental test set-up for closed-loop in-process warping detection during FFF 3D printing has been created. Three geometries, rectangular cuboids, triangular prisms, and cylinders were 3D printed in the test set-up and a classification accuracy of 99.3% was achieved.

The proposed system provides a non-contact method for warping detection that reduces structural restrictions and set-up difficulties. In addition, utilizing CNN classification models, this system offers accurate warping detection instead of assessing the overall part quality. Although the results are promising, the system has limitations that need to be addressed. The current underlying CNN classification model was developed through a manual approach, wherein the hyperparameters were optimized via trail and errors. Some machine learning models, such as CNNs, take several days to be trained, making the manual process time-consuming. Automating hyperparameter optimization via blackbox optimization techniques needs to be explored to reduce training times while adding flexibility to account for other defects. The existing image cropping subroutine is manual and can be automated to extract the corner image of different geometries during printing. Furthermore, the training images were taken in an environment where the lighting conditions were closely monitored to ensure input images were clear to assist the underlying classification model. The impact of ambient lighting conditions should be investigated to ensure the system could reliably be implemented in a real-life manufacturing environment.

Acknowledgment

Financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC), RGPIN-2018-04144, helped this investigation. We also would like to thank Larkin Lee and Levi Gregorash for their support in this project.

References

1. S. Ford and M. Despeisse, “Additive manufacturing and sustainability: An exploratory study of the advantages and challenges,” *Journal of Cleaner Production*, vol. 137, pp. 1573–1587, 2016.

2. M. S. Alsoufi, M. W. Alhazmi, D. K. Suker, W. K. Hafiz, S. S. Almalki, and R. O. Malibari, “Influence of multi-level printing process parameters on 3D printed parts in Fused Deposition Molding of poly(lactic) acid plus: A comprehensive investigation,” *American Journal of Mechanical Engineering*, vol. 7, no. 2, pp. 87–106, 2019.

3. D. Dimitrov, W. V. Wijck, K. Schreve, and N. D. Beer, “Investigating the achievable accuracy of three-dimensional printing,” *Rapid Prototyping Journal*, vol. 12, no. 1, pp. 42–52, 2006.

4. I. Gibson, B. Stucker, and D. W. Rosen, *Additive manufacturing technologies: rapid prototyping to direct digital manufacturing*. New York: Springer, 2010.

5. Z. Chen, Z. Shen, J. Guo, J. Gao, and X. Zeng, “Line drawing for 3D printing,” *Computers & Graphics*, 02-Jun-2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0097849317300687. [Accessed: 14-Sep-2019].

6. B. N. Panda, K. Shankhwar, A. Garg, and Z. Jian, “Performance evaluation of warping characteristic of fused deposition modelling process,” *The International Journal of Advanced Manufacturing Technology*, vol. 88, no. 5-8, pp. 1799–1811, May 2016.
7. W. Gao, Y. Zhang, D. Ramanujan, K. Ramani, Y. Chen, C. B. William, C. Y. Wang, Y. C. Shin, S. Zhang, and P. D. Zavattieri, “The status, challenges, and future of additive manufacturing in engineering,” *Computer-Aided Design*, pp. 65–89, 2015.
8. Liu, X. and V. Shapiro, “Homogenization of material properties in additively manufactured structures,” *Computer-Aided Design*, 2016. **78**: p. 71 82.
9. R. T. L. Ferreira, I. C. Amatte, T. A. Dutra, and D. Bürger, “Experimental characterization and micrography of 3D printed PLA and PLA reinforced with short carbon fibers,” *Composites Part B: Engineering*, vol. 124, pp. 88–100, 2017.
10. T.-M. Wang, J.-T. Xi, and Y. Jin, “A model research for prototype warp deformation in the FDM process,” *The International Journal of Advanced Manufacturing Technology*, vol. 33, no. 11-12, pp. 1087–1096, 2006.
11. A. Armillotta, “Assessment of surface quality on textured FDM prototypes,” *Rapid Prototyping Journal*, vol. 12, no. 1, pp. 35–41, 2006.
12. A. Peng and X. Xiao, “Investigation on reasons inducing error and measures improving accuracy in Fused Deposition Modeling,” *INTERNATIONAL JOURNAL ON Advances in Information Sciences and Service Sciences*, vol. 4, no. 5, pp. 149–157, 2012.
13. A. Guerrero-De-Mier, M. Espinosa, and M. Domínguez, “Bricking: A new slicing Mmethod to reduce warping,” *Procedia Engineering*, vol. 132, pp. 126–131, 2015.
14. Q. Y. Lu and C. H. Wong, “Additive manufacturing process monitoring and control by non-destructive testing techniques: Challenges and in-process monitoring,” *Virtual and Physical Prototyping*, vol. 13, no. 2, pp. 39–48, 2017.
15. J. Liu, Y. Hu, B. Wu, and Y. Wang, “An improved fault diagnosis approach for FDM process with acoustic emission,” *Journal of Manufacturing Processes*, vol. 35, pp. 570–579, 2018.
16. O. Holzmond and X. Li, “In situ real time defect detection of 3D printed parts,” *Additive Manufacturing*, vol. 17, pp. 135–142, 2017.
17. T. S. Srivatsan and T. S. Sudarshan, *Additive manufacturing: innovations, advances, and applications*. Boca Raton: CRC Press, 2016.
18. F. Baumann and D. Roller, “Vision based error detection for 3D printing processes,” *MATEC Web of Conferences*, vol. 59, p. 06003, 2016.
19. W. Yi, H. Ketai, Z. Xiaomin, and D. Wenying, “Machine vision based statistical process control in fused deposition modeling,” *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2017.
20. L. Meng, B. McWilliams, W. Jarosinski, H.-Y. Park, Y.-G. Jung, J. Lee, and J. Zhang, “Machine learning in Additive Manufacturing: A review,” *JOM*, Apr. 2020.
21. Y. Li, W. Zhao, Q. Li, T. Wang, and G. Wang, “In-Situ monitoring and diagnosing for Fused Filament Fabrication process based on vibration sensors,” *Sensors*, vol. 19, no. 11, p. 2589, 2019.

22. Y. Banadaki, N. Razaviarab, H. Fekrmandi, and S. Sharifi, “Toward enabling a reliable quality monitoring system for Additive Manufacturing process using deep convolutional neural networks,” arXiv preprint arXiv, Mar. 2020.
23. F. Hahne, W. Huber, R. Gentleman, R. Gentleman, and V. J. Carey, “Unsupervised Machine Learning,” in *Bioconductor Case Studies*, New York, NY: Springer New York, 2008, pp. 137–138.
24. H. Wu, Y. Wang, and Z. Yu, “In situ monitoring of FDM machine condition via acoustic emission,” *The International Journal of Advanced Manufacturing Technology*, May 2016.
25. H. Wu, Z. Yu, and Y. Wang, “A new approach for online monitoring of additive manufacturing based on acoustic emission,” *Volume 3: Joint MSEC-NAMRC Symposia*, 2016.
26. J. Liu, Y. Hu, B. Wu, and Y. Wang, “An improved fault diagnosis approach for FDM process with acoustic emission,” *Journal of Manufacturing Processes*, vol. 35, pp. 570–579, 2018.
27. H. Wu, Z. Yu, and Y. Wang, “Experimental study of the process failure diagnosis in additive manufacturing based on acoustic emission,” *Measurement*, vol. 136, pp. 445–453, 2019.
28. Dreyfus Gérard, *Neural Networks: Methodology and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.
29. K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
30. X. Qi, G. Chen, Y. Li, X. Cheng, and C. Li, “Applying neural-network-based machine learning to Additive Manufacturing: Current applications, challenges, and future perspectives,” *Engineering*, vol. 5, no. 4, pp. 721–729, 2019.
31. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
32. Z. Zhang, G. Wen and S. Chen, "Weld image deep learning-based on-line defects detection using convolutional neural networks for Al alloy in robotic arc welding", *Journal of Manufacturing Processes*, vol. 45, pp. 208-216, 2019. Available: 10.1016/j.jmapro.2019.06.023.
33. B. Zhang, P. Jaiswal, R. Rai, P. Guerrier and G. Baggs, "Convolutional neural network-based inspection of metal additive manufacturing parts", *Rapid Prototyping Journal*, vol. 25, no. 3, pp. 530-540, 2019. Available: 10.1108/rpj-04-2018-0096.
34. A. Caggiano, J. Zhang, V. Alfieri, F. Caiazzo, R. Gao, and R. Teti, “Machine learning-based image processing for on-line defect recognition in additive manufacturing,” *CIRP Annals*, vol. 68, no. 1, pp. 451–454, 2019.
35. L. Scime and J. Beuth, “A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process,” *Additive Manufacturing*, vol. 24, pp. 273–286, 2018.
36. S. Reddi, S. Kale and S. Kumar, "On the Convergence of Adam and Beyond", arXiv.org, 2019. [Online]. Available: https://arxiv.org/abs/1904.09237. [Accessed: 23- May- 2020].