

Convolutional Neural Network Approach for Predicting Melt Pool Dimensions in Laser Powder Bed Fusion Process

Oluwapelumi O. Adejumo¹, Nayan Pundhir¹, and K. Chandrashekhar¹

Department of Mechanical and Aerospace Engineering

¹Missouri University of Science and Technology, Rolla, MO 65409, USA

Heath Misak² and Cesar O. Rios²

²Spirit AeroSystems, Wichita, KS 67210, USA

Abstract

Laser powder bed fusion (LPBF) is an additive manufacturing process that uses a high-powered laser to selectively melt metal powder and build parts layer by layer. In this study, a computational fluid dynamics model that simulates the LPBF process was developed using Flow-3D software to generate melt pool images. These images were used to train a convolutional neural network, a deep learning model well-suited for image-based data, to predict the melt pool dimensions. The performance of the model was evaluated using mean absolute error and root mean square error, yielding values of 0.0275 and 0.0371, respectively. The model demonstrated predictive capabilities in estimating accurate melt pool dimensions directly from the melt pool images, highlighting its potential for data-driven monitoring and quality control in LPBF processes.

Keywords: CNN, LPBF, Flow-3D, Additive manufacturing, Inconel 718

1. Introduction

Laser powder bed fusion (LPBF), also referred to as selective laser melting, utilizes a concentrated laser beam to fuse metal powders within a predefined region corresponding to the cross-sectional layer of a 3D CAD model [1,2]. The molten material rapidly solidifies due to the high cooling rate. Once a layer is scanned, the build platform is lowered by a distance corresponding to the layer thickness. A recoater then distributes a fresh layer of powder, which is subsequently fused by the laser according to the next cross-sectional slice. This layer-by-layer process continues within a controlled atmosphere until all sections of the CAD model are fully processed [1]. LPBF is utilized in aerospace, automotive, biomedical, and other high-tech industries for manufacturing parts with intricate geometries [2].

Producing high-quality parts necessitates a deep understanding of defect formation mechanisms and strategies to prevent or mitigate them. In LPBF, the melt pool serves as the fundamental unit of the part-forming process and is directly linked to defect formation. Variations in melt pool geometry can lead to anisotropic microstructures, affecting mechanical properties and reliability [3]. Its characteristics are influenced by adjustable LPBF process parameters such as

laser power, scan speed, and layer thickness, as well as the material properties of the powder used [4]. Higher laser power and lower scan speeds generally increase melt pool depth and width, enhancing melting efficiency but also raising the risk of defects like keyhole porosity [5], so finding right melt pool size is important.

As mentioned earlier, the morphology of the melt pool alongside the temperature distribution has been identified to be considered more important amidst other factors that influence the final quality of the fabricated part. This has driven researchers to develop precise and reliable methods for their analysis and prediction [6]. Recent studies have employed machine learning to predict melt pool geometry and optimize process parameters, significantly reducing the time required for experimental validation [7]. Ogoke et al. [8] introduced a data-driven approach using convolutional neural network (CNN) to predict and visualize the 3D morphology of the melt pool and keyhole cavity from 2D surface temperature maps and processing parameters to optimize LPBF process parameters. Han et al. [9] developed a stacking ensemble model combining artificial neural network, gradient boosting, kernel ridge regression, and elastic net, with lasso regression as the meta-learner to predict the tensile strength of Ti6Al4V components fabricated using LPBF. The model achieved a high predictive accuracy, with a coefficient of determination (R^2) of 0.944.

Traditional monitoring methods based on simple pixel intensity summation emerged inadequate for capturing the complex variability in melt-pool shapes. To address this limitation, a multi-layer perceptron was used to classify melt-pool images captured during LPBF according to varying laser power settings. The trained model achieved a classification failure rate of less than 1.1% on 13,200 test images [10]. Data required for deep learning is usually vast, which can be challenging and costly to acquire experimentally, leading researchers to train their models using data from direct numerical simulations [6].

The rise of different architectures of deep neural networks, can be seen to have found application in AM processes. In this study, CNN, a deep learning model capable of extracting spatial features from images, is employed to predict the melt pool dimensions of the thermal images generated during LPBF. Flow-3D, a numerical modeling software, is used to simulate the LPBF process. By varying processing parameters during simulation, melt pools with different characteristics are produced. The resulting image data is preprocessed and used to train and evaluate the CNN model.

2. LPBF Modeling and Simulation

Flow-3D is a suitable modeling tool for solving complex fluid dynamics problems. It can simulate both 2D and 3D flow fields and is widely used in various additive manufacturing processes such as LPBF.

In LPBF simulation, the powder bed generation is modeled using the discrete element method (DEM) and the free surface of the melt pool is captured using the volume of fluid method. DEM simulates the random packing of powder particles, accounting for inter-particle forces and their

interactions with the environment [11]. Once the powder bed is formed, the conservation equations of mass, momentum, and energy are solved to describe the thermo-fluid behavior of the melt pool. These equations are solved by Flow-3D using a finite volume method [12]. The continuity equation given in Equation (1) governs mass conservation, ensuring that the mass of the powder bed is maintained throughout the process. Momentum equations given in Equation (2) predict the circulation and ejection of molten metal, influenced by laser movement and heating [13]. Equation (3) is the energy conservation equation, which models heat conduction, melting, and solidification processes.

$$\nabla \cdot \vec{v} = 0 \quad (1)$$

$$\frac{\partial h}{\partial t} + (\nabla \cdot \vec{v})h = \frac{1}{\rho} (\nabla \cdot k \nabla T) \quad (2)$$

$$\rho \left[\frac{\partial}{\partial t} (\vec{v}) + \vec{v} \cdot \nabla \vec{v} \right] = -\vec{v}p + \nabla \cdot \left(\mu (\nabla \vec{v} + \nabla \vec{v}^T) - \frac{2}{3} \delta_{ij} \nabla \cdot \vec{v} \right) - \frac{C_1(1-f_l)^2}{C_2 + f_l^2} \vec{v} - \rho g \beta (T - T_l) \quad (3)$$

Where, ρ represent fluid density, \vec{v} show velocity vector of flow field, ∇ represent gradient operator, and p represent hydrodynamic pressure. μ represent the kinematic viscosity of the liquid, and δ_{ij} represent Kronecker delta. C_1 and C_2 are empirical constants of order of 10^6 and 10^{-4} respectively used for modeling solidification drag force [14]. f_l represent the drag force coefficient, g is gravitational acceleration, and β represent coefficient of thermal expansion. T represent the fluid temperature, while T_l denote the melting temperature. k denotes thermal conductivity, and h represent specific enthalpy.

3. Convolutional Neural Network Model

CNN is a type of deep learning algorithm that can learn spatial hierarchies of features from images [15,16]. It typically consists of multiple layers, including convolutional, pooling, and fully connected layers [17]. The convolutional layer extracts image features by leveraging local correlations. A kernel scans the image, multiplying pixel values with corresponding weights, summing the results, and adding a bias. The process of convolution is shown in Figure 1. The pooling layer then downsamples the feature map extracted from the convolutional layers [18], enhancing invariance to translation, rotation, and scaling. Common pooling methods include max pooling, which selects the highest value in a region, and average pooling, which computes the mean as seen in Figure 2. Convolutional and pooling layers are used alternately in applications to progressively extract and condense features from the input image [19] before it is passed down to fully connected layers for prediction. The workflow of the CNN model implemented in this study is illustrated in Figure 3.

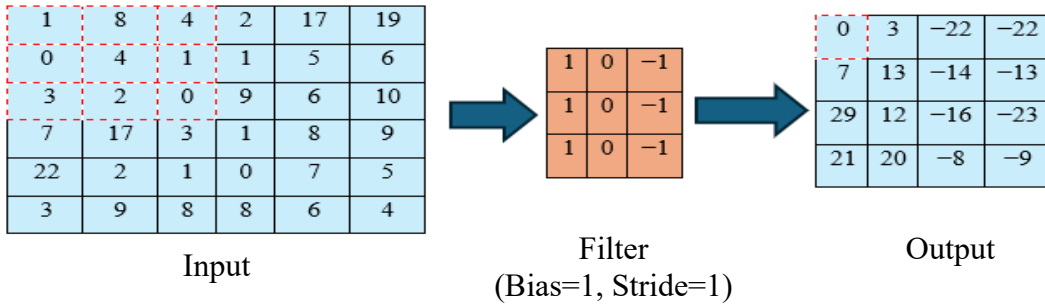


Figure 1: The process of convolution operation

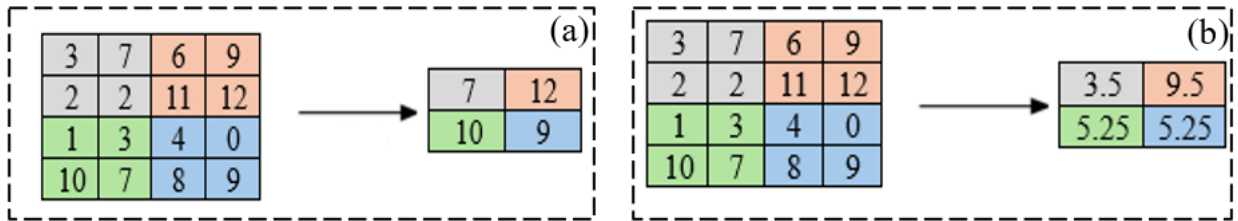


Figure 2: The process of pooling operation (a) maximum pooling (b) average pooling

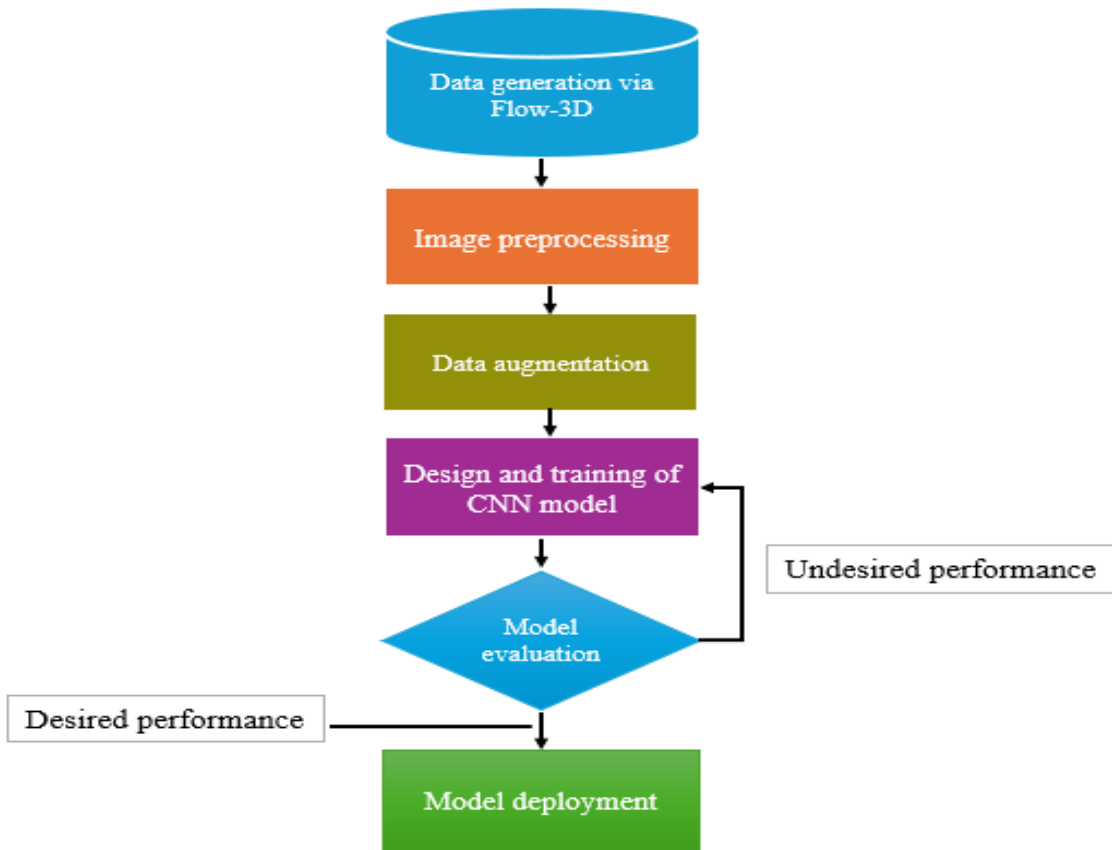


Figure 3: CNN model flowchart

3. 1 Data source

The dataset used to train the CNN model was generated by simulating the LPBF process in Flow-3D, with key parameters such as laser power and scan speed carefully configured. A single-layer, single-track melting process with Inconel 718 as deposited powder is modeled. During each simulation run, images were captured at multiple time steps to monitor the evolution of the melt pool from initiation to completion. Each image was annotated with its corresponding melt pool width (W) and height (H). To expand the dataset and ensure variability [20], the process parameters were varied across five different simulation runs. The specific parameter combinations are summarized in Table 1. On average, each simulation produced approximately 10 images, resulting in a total dataset of 50 annotated samples. These images were then used as input for training and validating the CNN model, with an 80:20 data split.

Melt pool images were captured at the cross-section of the deposited Inconel 718 layer, following the process parameters defined in Table 1. Table 2 shows the material properties of Inconel 718. Figure 4 shows the cross-section of the melt pool with their corresponding dimensions as obtained from Flow-3D and used as input for the CNN model.

Table 1: Laser power and scan speed settings

Case	Laser power (W)	Scan speed (mm/s)
1	200	1200
2	250	1000
3	300	800
4	350	600
5	400	400

Table 2: Inconel 718 powder material properties

Property	Value
Solidus temperature T_s (K)	1523.15
Liquidus temperature T_l (K)	1608.15
Density ρ (kg/m ³)	5.96
Thermal conductivity k_l (erg/cm/s/K)	2.9×10^6
Specific heat capacity c (erg/g/K)	7.2×10^6
Latent heat of fusion L (erg/g)	2.1×10^9
Surface tension coefficient σ (g/s ²)	1882
Temperature coefficient of surface tension $\frac{\partial \sigma}{\partial t}$ (g/(s ² K))	0

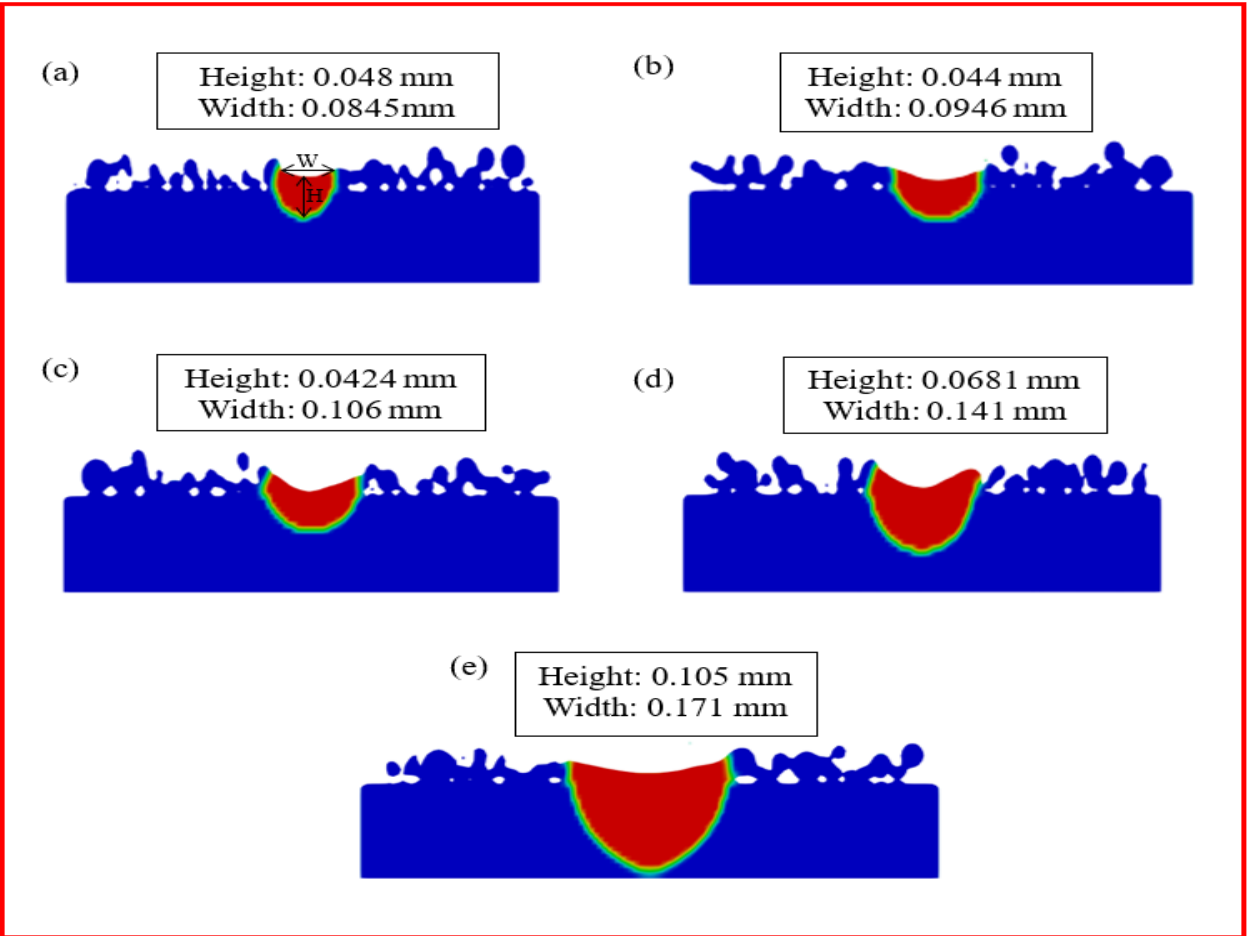


Figure 4: Cross-section of simulated melt pool for (a) case 1, (b) case 2, (c) case 3, (d) case 4, and (e) case 5

3.2 Image preprocessing

Image preprocessing is a critical initial step in computer vision and image analysis workflows. It helps fix common problems in images, like shadows, noise, distortion, and busy backgrounds. By improving the image quality, it enables efficient handling of the images in the steps that follow [21]. Standardization or normalization is a preprocessing method that adjusts pixel intensity values using a specific formula to minimize differences caused during data collection [22].

To prepare the image data for model training, a preprocessing function was used to standardize all input images. The process involved reading each image from its file path, resizing it to a fixed size of 224×224 pixels, and then normalizing the pixel values to a range between 0 and 1 by dividing each pixel by 255. This was done to ensure that all images were the same size and intensity scale, which helps improve the performance and stability of deep learning models. The file paths for the images were extracted, and the corresponding labels (width and height) were also retrieved to be used as target values during training.

3.3 Data augmentation

Various preprocessing techniques are commonly used to augment datasets. These include image flipping, noise addition, gamma adjustment, rotation, scaling, cropping, resizing, and zooming. Data augmentation involves generating additional data by modifying existing samples through these methods. In deep neural network applications, a larger dataset is generally preferred; therefore, augmentation is often employed when the original dataset is limited [21].

Different image augmentation was applied to the dataset, random rotations of 30 degrees, horizontal and vertical shifts by up to 20% of the image dimensions, and shear transformations with a range of 0.2. Further augmentation applied includes zooming operations within a 20% range to simulate scale variations, and horizontal flipping to introduce left-right invariance. Brightness adjustments were also incorporated, allowing the pixel intensity to vary randomly within a specified range (from 80% to 120% of the original brightness). The fill_mode parameter was set to 'nearest' to ensure that any newly created pixels during transformations were filled appropriately using the nearest-neighbor approach. Data augmentation helps mitigate overfitting and improves the model's ability to generalize across varied image inputs, especially in cases like this where the available dataset is limited in size. Figure 5 shows how image data gets to transform when augmentation is applied.

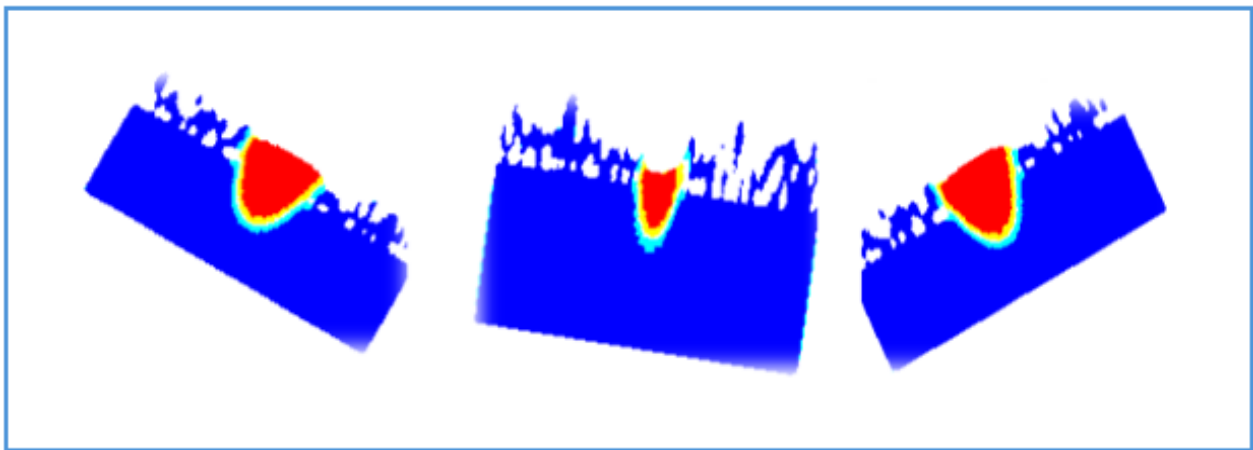


Figure 5: Augmented image samples

3.4 Model architecture

The model is a CNN, which is typically designed for image classification tasks but in this case, it was adapted for an image regression task. It accepted input images with a shape of 224×224 pixels and three-color channels (RGB). It began with a convolutional layer consisting of 32 filters of size 3×3 , followed by a ReLU activation function to capture low-level features. A max-pooling layer with a pool size of 3×3 was applied to downsample the feature map. The second convolutional layer employed 64 filters of size 3×3 and was followed by another max-pooling layer with a 2×2 pool size. The third convolutional block used 128 filters of size 3×3 , followed

by a max-pooling layer with a 2×2 pool size. After the convolutional and pooling layers, the model flattened the output and passed it through a fully connected (dense) layer with 256 neurons and a ReLU activation function. A dropout layer with a rate of 0.5 was included to reduce overfitting. The output layer consists of two neurons with a linear activation function to suit the regression task. The model used the Adam optimizer with a learning rate of 0.001.

4. Results and Discussion

4.1 Loss curve

Training of the model was done across 15 epochs with a batch size of 32 which allowed for frequent weight updates and computational efficiency. The model is compiled using mean squared error (MSE) as the loss function. As seen in figure 6, both the training and validation loss curves show a sharp decline within the initial few epochs, indicating that the model quickly learns the underlying patterns in the data. After the third epoch, the loss values plateau at very low levels and remain stable through to the fifteenth epoch. This stabilization of both training and validation loss suggests that the model is not overfitting, as the gap between the two remains small throughout training. The validation loss closely follows the training loss, indicating the ability to generalize to unseen data. Moreover, the early and sustained convergence of the curves highlights the effectiveness of the model's configuration, particularly the use of the Adam optimizer and the MSE loss function, which efficiently minimizes prediction error. The nearly flat loss trajectory in the later epochs implies that the model has reached an optimal solution, and further training would likely yield minimal improvements.

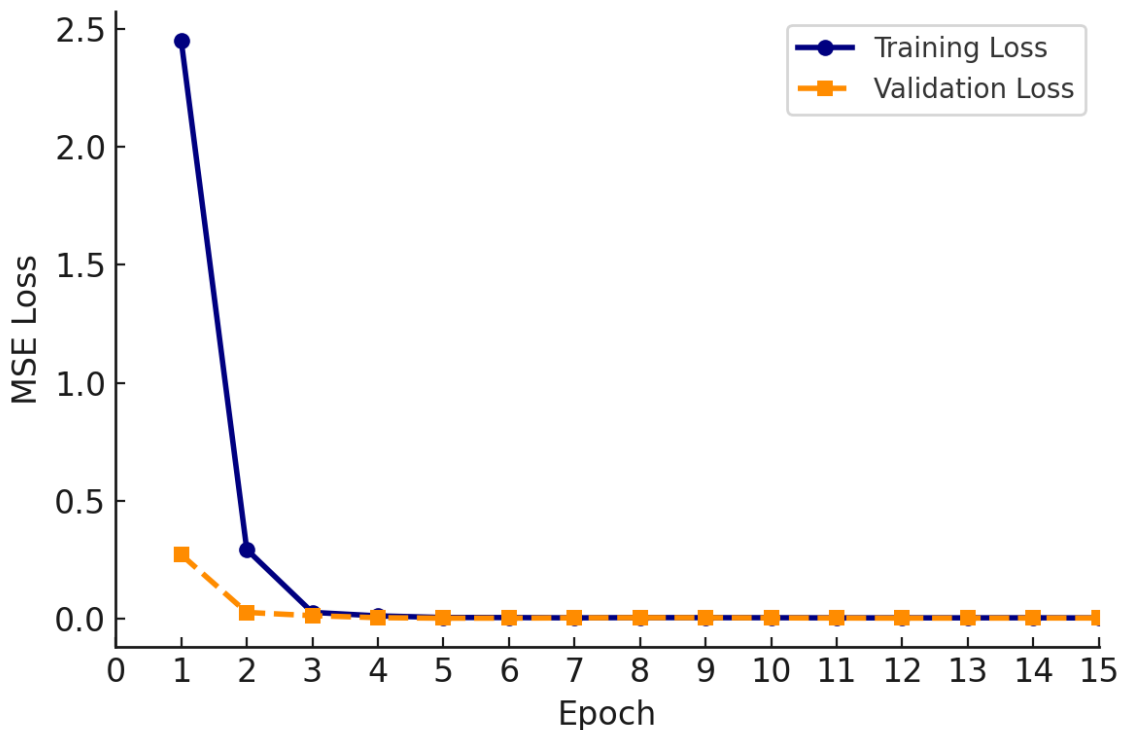


Figure 6: MSE loss curve

4.2 Model evaluation

Two evaluation metrics were tracked across the epochs: MAE and RMSE. Both metrics showed clear improvement during training. By the fifth epoch, both training and validation error curves had plateaued at low levels, indicating that the model had begun to converge. The training MAE and RMSE decreased from 0.5085 and 1.3144 in the first epoch to 0.0311 and 0.0394 by the final epoch. Similarly on the validation set, MAE dropped from 0.3802 to 0.0275, while RMSE decreased from 0.5199 to 0.0371, with lower values indicating better model performance. Figure 7 shows how MAE and RMSE progressed during training and validation across the epochs. This performance trend indicates that the model is learning effectively, generalizing well to unseen data.

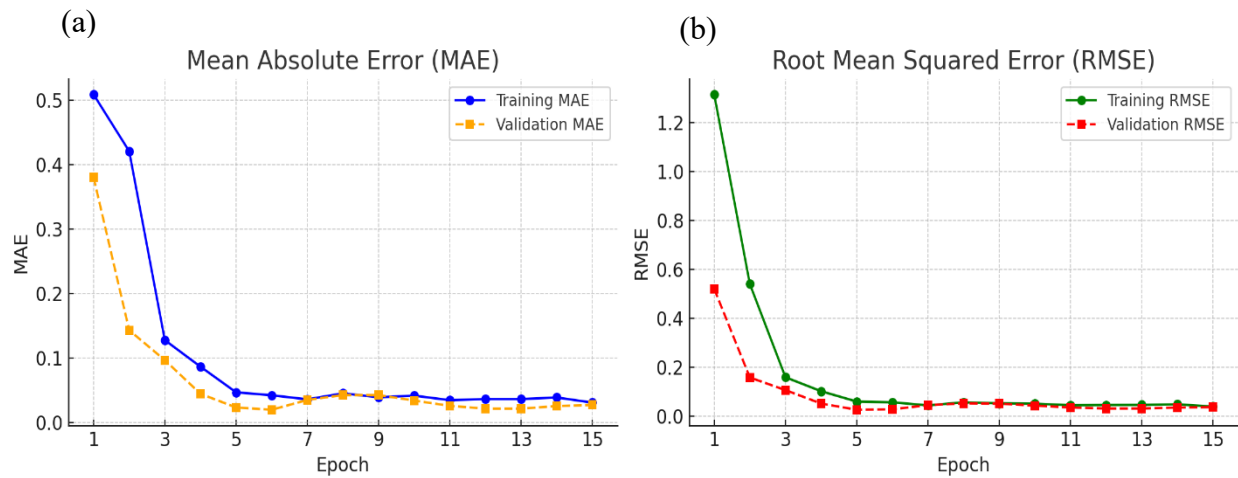


Figure 7: Training and validation performance metrics across epochs (a) mean absolute error (b) root mean squared error

5. Conclusion

Compared to other studies where CNNs are mainly used for image classification, this study demonstrated the use of CNN for a regression task. It was designed to simultaneously predict two continuous output variables, the melt pool height and width during LPBF. Training images were taken at different time steps to capture changes over time, and data augmentation was used to increase dataset diversity. Although these factors introduced additional variability, the model effectively captured the underlying patterns necessary for accurate predictions. Accurate melt pool prediction is critical for real-time process monitoring and quality assurance in additive manufacturing, as it facilitates early detection of potential defects, such as porosity or lack-of-fusion. Wide but shallow melt pools, for instance, typically indicate insufficient fusion and the presence of porosity. Real-time CNN-based monitoring during LPBF would therefore enable automatic adjustment of process parameters, such as laser power and scan speed. For melt pool

dimensions that likely fall outside acceptable limits, the system could halt the build and flag it as likely to contain defects.

6. Acknowledgments

Support from the Center for Aerospace Manufacturing Technologies (CAMT) at Missouri University of Science and Technology is gratefully acknowledged.

7. References

- [1] Cao, S., Zou, Y., Lim, C. V. S., and Wu, X., “Review of laser powder bed fusion (LPBF) fabricated Ti-6Al-4V: process, post-process treatment, microstructure, and property,” *Light: Advanced Manufacturing*, Vol. 2, pp. 313–332, 2021. doi: 10.37188/lam.2021.020.
- [2] Obeidi, M. A., Ahad, I. U., and Brabazon, D., “Investigating the melt-pool temperature evolution in laser-powder bed fusion by means of infra-red light: a review,” *Key Engineering Materials*, Vol. 926, pp. 235–241, 2022. doi: 10.4028/p-6fn67z.
- [3] Qadeer, A., Akhtar, S. S., Abubakar, A. A., Arif, A. F. M., Mekid, S., and Al-Athel, K. S., “Accelerating microstructure and melt pool prediction in laser powder bed fusion processes using data driven neural network,” *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, Vol. 2, pp. 1–8, 2025. doi: 10.1115/imece2024-145983.
- [4] Yang, A., Zhao, Z., and Zhang, X., “A bidirectional prediction framework for melt pool size and process parameters in LPBF,” *Engineering Research Express*, Vol. 7, pp. 1–18, 2025. doi: 10.1088/2631-8695/adb0a2.
- [5] Ninpetch, P., Kowitwarangkul, P., Mahathanabodee, S., Tongsri, R., and Ratanadecho, P., “Thermal and melting track simulations of laser powder bed fusion (L-PBF),” *IOP Conference Series: Materials Science and Engineering*, Vol. 526, pp. 1–5, 2019. doi: 10.1088/1757-899x/526/1/012030.
- [6] Hemmasian, A., Ogoke, F., Akbari, P., Malen, J., Beuth, J., and Farimani, A. B., “Surrogate modeling of melt pool temperature field using deep learning,” *Additive Manufacturing Letters*, Vol. 5, pp. 1–12, 2023. doi: 10.1016/j.addlet.2023.100123.
- [7] Rahman, M. S., Sattar, N. S., Ahmed, R. U., Ciaccio, J., and Chakravarty, U. K., “A machine learning framework for melt-pool geometry prediction and process parameter optimization in the laser powder-bed fusion Process,” *Journal of Engineering Materials and Technology*, Vol. 146, pp. 1–16, 2024. doi: 10.1115/1.4065687.
- [8] Ogoke, F., Lee, W., Kao, N. Y., Myers, A., Beuth, J., Malen, J., and Farimani, A. B., “Convolutional neural networks for melt depth prediction and visualization in laser powder bed fusion,” *International Journal of Advanced Manufacturing Technology*, Vol. 129, pp. 3047–3062, 2023. doi: 10.1007/s00170-023-12384-z.

- [9] Han, C., Yan, F., Yuan, D., Li, K., Yang, Y., Zhang, J., and Wang, D., “Machine learning enabling prediction in mechanical performance of Ti6Al4V fabricated by large-scale laser powder bed fusion via a stacking model,” *Frontiers of Mechanical Engineering*, Vol. 19, pp. 1–15, 2024. doi: 10.1007/s11465-024-0796-0.
- [10] Kwon, O., Kim, H. G., Ham, M. J., Kim, W., Kim, G. H., Cho, J. H., Kim, N. I., and Kim, K., “A deep neural network for classification of melt-pool images in metal additive manufacturing,” *Journal of Intelligent Manufacturing*, Vol. 31, pp. 375–386, 2020. doi: 10.1007/s10845-018-1451-6.
- [11] Chen, H., Sun, Y., Yuan, W., Pang, S., Yan, W., and Shi, Y., “A review on discrete element method simulation in laser powder Bed fusion additive manufacturing,” *Chinese Journal of Mechanical Engineering: Additive Manufacturing Frontiers*, Vol. 1, pp. 1–13, 2022. doi: 10.1016/j.cjmeam.2022.100017.
- [12] Ayoob, N. S. and Hamad, A. M., “Numerical simulation for flow over a broad-crested weir using Flow-3D program,” *Civil Engineering and Architecture*, Vol. 10, pp. 2157–2171, 2022. doi: 10.13189/cea.2022.100534.
- [13] Leclerc, W. and Chams, A., “Discrete element method framework to simulate metallic Laser powder-bed fusion additive manufacturing process,” *Proceedings of the Fourteenth International Conference on Computational Structures Technology*, Vol. 3, pp. 1–7, Aug. 2023. doi: 10.4203/ccc.3.17.2.
- [14] Alphonso, W. E., Baier, M., Carmignato, S., Hattel, J. H., and Bayat, M., “On the possibility of doing reduced order, thermo-fluid modelling of laser powder bed fusion (L-PBF) – Assessment of the importance of recoil pressure and surface tension,” *Journal of Manufacturing Processes*, Vol. 94, pp. 564–577, 2023. doi: 10.1016/j.jmapro.2023.03.040.
- [15] Parkavi, J., “A detailed study of deep learning using convolutional neural network approach,” *International Journal for Research in Applied Science and Engineering Technology*, Vol. 8, pp. 831–837, 2020. doi: 10.22214/ijraset.2020.32306.
- [16] Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K., “Convolutional neural networks: an overview and application in radiology,” *Insights into Imaging*, Vol. 9, pp. 611–629, 2018. doi: 10.1007/s13244-018-0639-9.
- [17] Sowmya, B. P. and Supriya, M. C., “Convolutional neural network (CNN) fundamental operational survey,” *Learning and Analytics in Intelligent Systems*, Vol. 21, pp. 245–258, 2021. doi: 10.1007/978-3-030-65407-8_21.
- [18] Purwono, P., Ma’arif, A., Rahmani, W., Fathurrahman, H. I. K., Frisky, A. Z. K., and Haq, Q. M. ul, “Understanding of convolutional neural network (CNN): A review,” *International Journal of Robotics and Control Systems*, Vol. 2, pp. 739–748, 2023. doi: <https://doi.org/10.31763/ijrcs.v2i4.888>.

- [19] Lu, J., Tan, L., and Jiang, H., “Review on convolutional neural network (CNN) applied to plant leaf disease classification,” *Agriculture*, Vol. 11, pp. 1–18, 2021. doi: 10.3390/agriculture11080707.
- [20] Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., Chen, Y., Zeng, W., and Yu, P. S., “Generalizing to unseen domains: A survey on domain generalization,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, pp. 8052–8072, 2023. doi: 10.1109/tkde.2022.3178128.
- [21] Vishnoi, V. K., Kumar, K., and Kumar, B., “Plant disease detection using computational intelligence and image processing,” *Journal of Plant Diseases and Protection*, Vol. 128, pp. 19–53, 2020. doi: 10.1007/s41348-020-00368-0.
- [22] Li, X. T. and Huang, R. Y., “Standardization of imaging methods for machine learning in neuro-oncology,” *Neuro-Oncology Advances*, Vol. 2, pp. 49–55, 2020. doi: 10.1093/noajnl/vdaa054.