Solid Freeform Fabrication 2021: Proceedings of the 32nd Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference
Reviewed Paper

INFERENCE OF METAL ADDITIVE MANUFACTURING PROCESS STATES VIA DEEP LEARNING TECHNIQUES

Richard Anarfi[α*], Benjamin Kwapong[α], Kenneth Fletcher[α], Todd Sparks[β], Aaron Flood[β], and Mugdha Joshi[γ]

[α]University of Massachusetts Boston, Boston, MA 02125
[β]Product Innovation and Engineering LLC, St. James, MO 65559
[γ]Missouri University of Science and Technology, Rolla, MO 65409
[*]Corresponding author

August 20, 2021

Abstract

Numerical simulation of metal additive processes are computationally intensive tasks. Iterative solution techniques for physics-based methods can lead to lengthy solution times and convergence problems, particularly if fluid dynamics of the melt pool are considered. Deep learning (DL) techniques offer an opportunity to infer solution results quickly. In this paper we propose a DL method based on long short term memory (LSTM), network trained on rendered images from a metal AM process simulation and CAM data. We obtained vector representations of the images by training on an autoencoder. LSTM is a memory based recurrent neural networks (RNN) that is capable of processing long sequences of data while combating temporal stability problems encountered with conventional recurrent neural networks (RNN)s. This LSTM network is used to predict images of the process given scan path and process information. This could later be used to compare with process monitoring systems as part of a quality assurance or process control schema.
Key words: Additive manufacturing, autoencoder, deep learning

1 Introduction and Motivation

Additive manufacturing (AM) processes rely on a sets of technology parameters (e.g. slicing settings, infill settings, speeds, feeds, laser settings) to generate functional machine level code to produce printed parts. Sets of the technology parameters are commonly referred to as a 'profile'. The contents of a particular profile can be dependent upon process, AM system architecture, material, and desired part geometry. Complicating matters further, performance characteristics of the printed parts are a consideration in the profile selection for metal parts produced via Directed Energy Deposition (DED) and Laser Powder Bed Fusion (LPBF) processes.

Developing a new 3d print profile can be a costly endeavor. Users typically rely on some combination of experimentation and simulation to justify decisions and set values for critical parameters. This process can be both time consuming and costly. This paper presents an attempt to shortcut

the costly computational methods used to simulate metal 3d printing processes by inference using a LSTM [1, 2]. When successful, this inference schema has applications in process development, machine controls, and quality assurance.

2 Literature review

Several researches have proposed different neural network architectures based on build classification depending on the input laser process parameters. long short term memory (LSTM)-recurrent neural networks (RNN) are widely used in applications such as text completion [3, 4], image captioning [5], stock prices forecasting [6]. LSTM networks can handle long term dependencies for long data sequences therefore are advantageous over traditional recurrent neural networks (RNN) architectures [7]. Other than the mentioned applications, its applications in image series prediction is analogous to the current requirement. These networks can efficiently predict the trajectory of a pedestrian [8, 9, 10] as well as the traffic trajectory [11, 12, 13].

Different Neural network architectures have been used in the applications of in-situ monitoring and optimizing process parameters for DED processes[14, 15, 16]. However, recent studies show that this state of art technology is primarily used for prediction of temperature history in the DED process. According to the present literature, recurrent neural networks consisting of LSTM cells provide an accurate mapping of temporal information about thermal distribution in the DED process.

A study by Ren et al. [17] shows that in comparison of individual architectures, a combination of recurrent neural networks (RNN) and DNN shows consistency in contour maps and temperature distribution produced by finite element simulation. This type of architecture has an added advantage over individual networks that LSTM cell in recurrent neural networks (RNN) learns sequential data of laser scan path which is mapped to temperature field information via fully connected layer in DNN. However this model is confined to single layer process prediction with limited input information about laser process parameters and material properties.

In a similar study Mozaffar et al. [18] proposed that a stacked recurrent neural networks (RNN) with Gated Recurrent Unit (GRU) formulation is efficient in predicting thermal history at a single point for different geometries. GRU structure in the network can accurately find out the hidden correlations in the high dimensional data and predicts critical features in the thermal history such as sharp gradients, and melting and re-melting of the material.

Zhang, Liu, and Wu [19] used two Machine Learning (ML) algorithms XGBoost and LSTM-RNN to predict the melt pool temperatures of an eight layer thin-walled CarTech® 718 alloy sample made by DED process. Prediction accuracy of both algorithms was measured for different input process parameters. It is found out that LSTM outperforms XGBoost algorithm in all four special cases with highest and lowest values of Laser power and scan speed. Though LSTM is computationally ineffective than XGBoost, fluctuation in the melt pool temperature does not affect the accuracy of LSTM model.

Thus, LSTM model provides promising results in predicting thermal history.However, as far as we know, no previous research has been investigated in prediction of process images based on the laser scan path and process parameters. Although a lot of work has already been done in modeling and simulation of DED process, this technique is time consuming and cannot always guaranteed to be convergent. On the there hand, deep learning requires huge training data which is not easy

to generate in AM processes. Thus, this research proposes an innovative approach of integrating physics based iterative methods with deep learning techniques.

3 Research methodology

In this section, we describe the structure and various sections of our approach, starting with the general overview, followed by detailed description of the sections/parts.

Our model trains an autoencoder (Section 3.1) to learn a compressed (encoded) version of the image we feed into it. This is achieved using several convolutional layers (Section 3.3). The compressed image (from a vector of length >49,000 to a vector of length 16) can then be decompressed (decoded) back into its original shape/dimension. Our model feeds the time-series embedded vectors of very small dimension (output of the autoencoder's encoder) into an LSTM (Section 3.2) which we train to predict vectors that are some time steps ahead. We then use our decoder from the autoencoder to decode these vectors into actual image vectors. The architecture of our model can be seen in Figure 1.

3.1 Autoencoder

The concept of autoencoders as seen in Figure 2 is that we want to force a network to try to reconstruct our data and hopefully it will learn a useful representation of the data. For traditional autoencoders this is used for feature learning. Normally the encoder will be a 4-layer downsampled

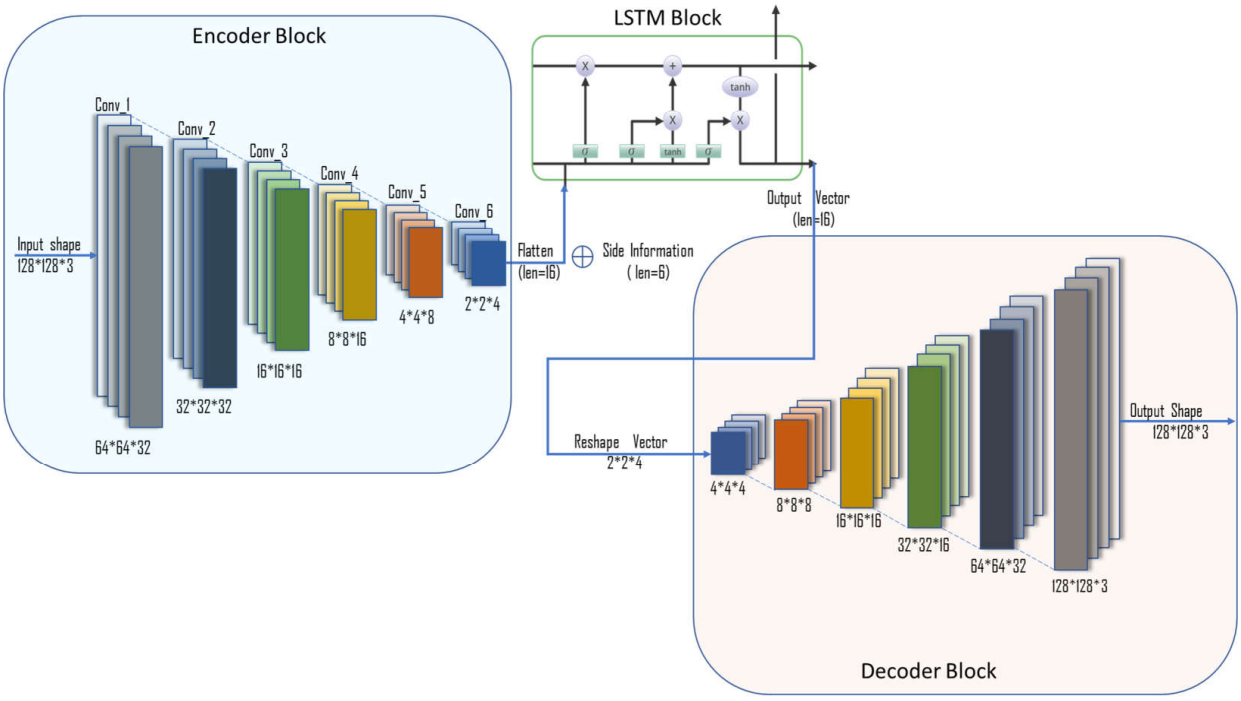


Figure 1: The architecture of our proposed model. There are 3 main parts: The encoder, which consists of convolutional networks, the LSTM, and the decoder which also has convolutional networks.

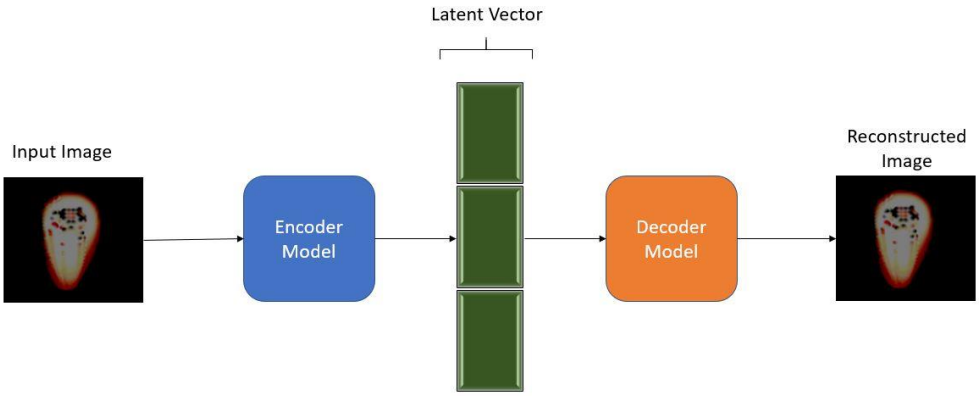


Figure 2: Autoencoder

convolution, and the decoder a 4-layer upsampled convolution. Sometimes they share weights: $\dim(x) = D, \dim(z) = H, w_e : H \times D, w_d : D \times H = w_e^T$.

The idea is to pass the input data through an encoder network that will produce features z. This step can be thought as PCA. We are transforming the input data and transforming it into another feature representation. The encoder network is usually a ReLU CNN. Normally z is smaller than x (dimensionality reduction). The problem is that we do not have any explicit labels to use to know which information is relevant and which one its not to decide how to make the dimensionality reduction. To decide which is the most relevant data we reconstruct the input data only with the information in the z features and compare the reconstruction with the original image. Specially if the z features are small, hopefully, it will force the network to summarize the useful statistics and to discover useful features of the input image.

3.2 Long-short term memory units

In this section we describe the deep LSTM. We let subscripts denote timesteps and superscripts denote layers. All our states are n-dimensional. Let $h_t^l \in \mathbb{R}^n$ be a hidden state in layer l in timestep t. Moreover, let $T_{n,m} : \mathbb{R}^n \to \mathbb{R}^m$ be an affine transform ($Wx + b$ for some W and b). Let $\odot$ be element-wise multiplication and let h_t^0 be an input word vector at timestep k. We use the activations h_t^L to predict y_t, since L is the number of layers in our deep LSTM. The LSTM is an advanced form of a recurrent neural network. The recurrent neural networks (RNN) dynamics can be described using deterministic transitions from previous to current hidden states. The deterministic state transition is a function

$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \to h_t^l$$

For classical recurrent neural networks (RNN)s, this function is given by

$$h_t^l = f(T_{n,n} h_t^{l-1} + T_{n,n} h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \tanh\}$$

The LSTM has complicated dynamics that allow it to easily memorize information for an extended number of timesteps. The long term memory is stored in a vector of *memory cells* $c_t^l \in \mathbb{R}^n$. Although many LSTM architectures that differ in their connectivity structure and activation functions, all LSTM architectures have explicit memory cells for storing information for long periods

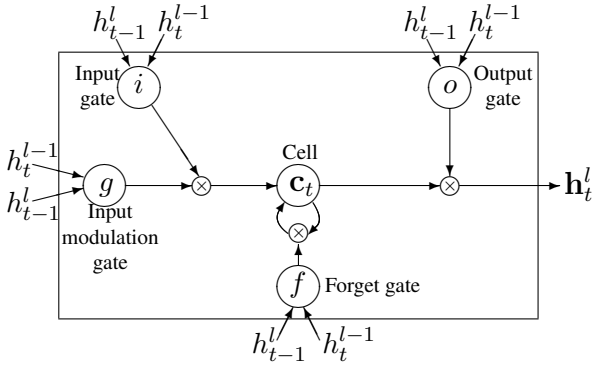


Figure 3: A graphical representation of LSTM memory cells used in this paper [21].

of time. The LSTM can decide to overwrite the memory cell, retrieve it, or keep it for the next timestep. The LSTM architecture used in our experiments is given by the following equations [20]:

$$\text{LSTM} : h_t^{l-1}, h_{t-1}^{l}, c_{t-1}^{l} \rightarrow h_t^{l}, c_t^{l}$$

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} T_{2n,4n} \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^{l} \end{pmatrix}$$

$$c_t^{l} = f \odot c_{t-1}^{l} + i \odot g$$

$$h_t^{l} = o \odot \tanh(c_t^{l})$$

In these equations, sigm and $\tanh$ are applied element-wise. Figure 3 illustrates the LSTM equations.

3.3 Convolutional Neural Networks

CNNs (Figure 4) generally consist of three elements: convolutional layers (Figure 5), pooling layers (Figure 6), and fully connected layers. In the convolutional layer, filters are convolved with the receptive field of the input image in a sliding window style to learn data-specific features. Basic features, such as lines, edges, and corners, are learned in the initial layers, while more abstract features are learned as layers go deeper [22].

Generally, a pooling layer follows each convolutional layer. Max-pooling is basically a nonlinear down-sampling procedure, which takes the maximum of $2x2$ neighborhoods of the image, and helps to reduce the computational complexity for the forward layers, as well as adding translation invariancy to the network. Fully connected layers are used to learn the nonlinear combinations of extracted features from previous layers. Dropout is recommended as a way of preventing overfitting by disabling randomly chosen neurons and their connections [23]. The dropped neurons stay inactive during the feedforward and backpropagation phases, thus forcing the network to learn different nonlinear combinations of features on each epoch.

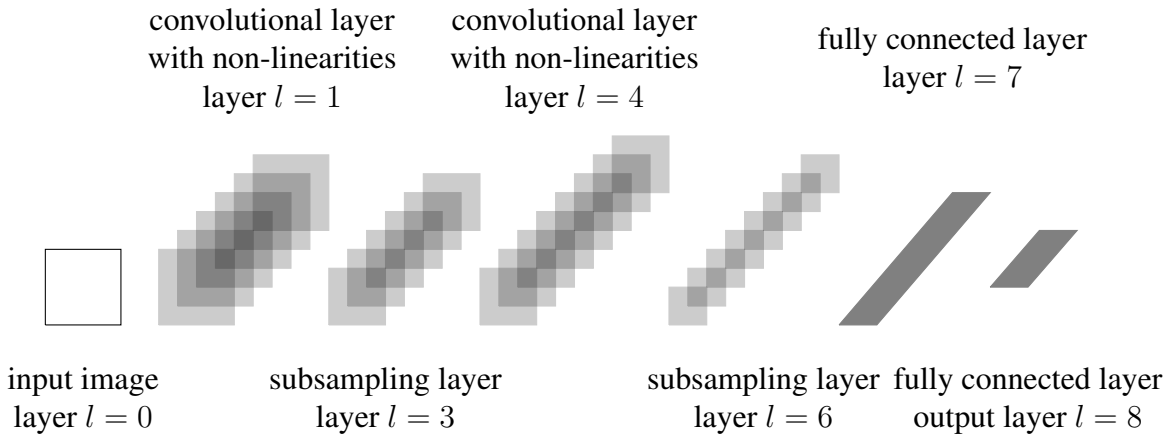


Figure 4: The architecture of the original convolutional neural network, as introduced by LeCun et al. [24], alternates between convolutional layers including hyperbolic tangent non-linearities and subsampling layers. In this illustration, the convolutional layers already include non-linearities and, thus, a convolutional layer actually represents two layers. The feature maps of the final subsampling layer are then fed into the actual classifier consisting of an arbitrary number of fully connected layers. The output layer usually uses softmax activation functions[25].

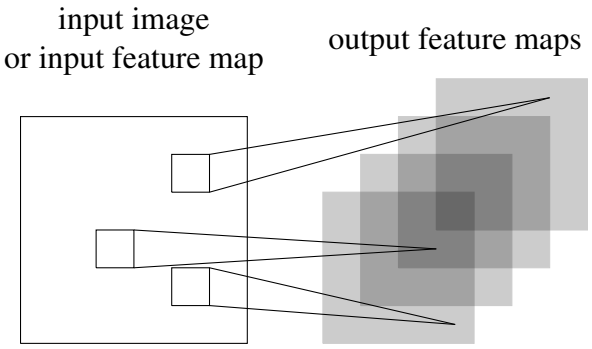


Figure 5: Illustration of a single convolutional layer. If layer l is a convolutional layer, the input image (if $l = 1$) or a feature map of the previous layer is convolved by different filters to yield the output feature maps of layer l.

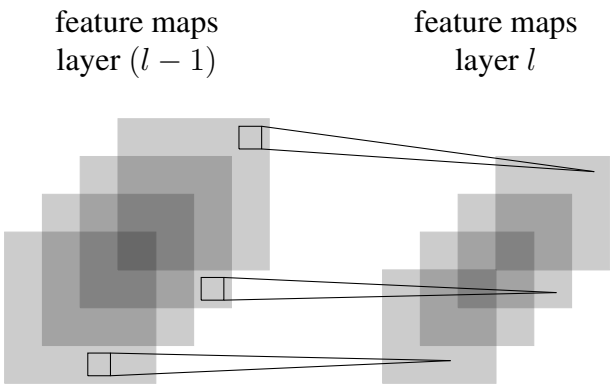


Figure 6: Illustration of a pooling and subsampling layer. If layer l is a pooling and subsampling layer and given $m_1^{(l-1)} = 4$ feature maps of the previous layer, all feature maps are pooled and subsampled individually. Each unit in one of the $m_1^{(l)} = 4$ output feature maps represents the average or the maximum within a fixed window of the corresponding feature map in layer $(l-1)$.

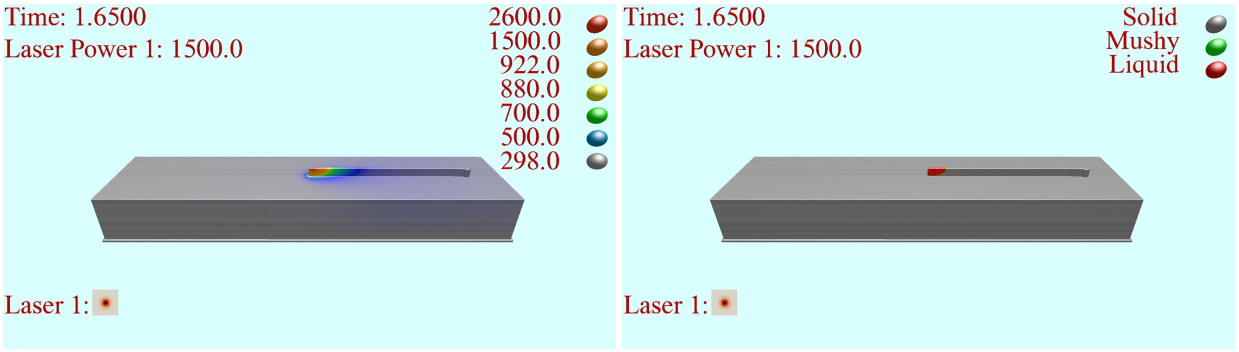
4 Simulation Description

With their work in AM [26, 27], Product Innovation and Engineering LLC (PINE) has developed several physics-based simulations in order to better understand the AM process. The simulation used in this study was developed to specifically target the thermal history of the material during a build. This simulation has the aims of being computationally efficient while still being based on physics. It has the expressed goals of being fast and giving general guidance to path planning by giving a thermal history for a given build. In order to accomplish this goal, a voxel based simulation was developed which heavily leverages a graphics processing unit (GPU) and is based on many image processing techniques. From past simulation development experience, it was understood that the calculation of the fluid flow is the most computationally expensive part of the simulation, therefore it was omitted in this simulation for the sake of efficiency.

The simulation is able to predict the thermal history of a part, Figure 7a, the phase map of the part at any given time, Figure 7b, and a cooling rate for any section of the part, Figure 7c. This helps to give a predictor of the micro-structures within the deposition which are the driving force behind the final mechanical properties.

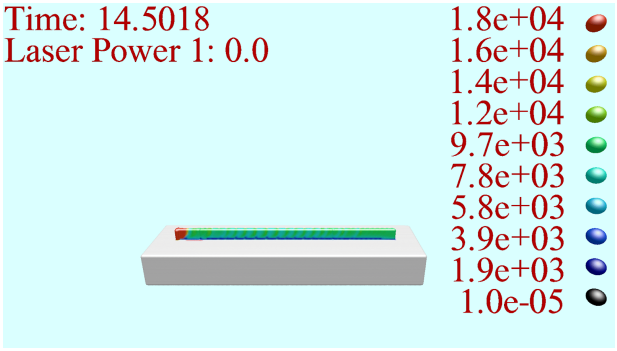
The laser in the simulation is modeled in 3-D to be able to take into account the beam quality. In many lasers the beam quality is defined using the beam parameter product (BPP) this is defined as $0.5\theta w_0$ where θ and w_0 are the divergence angle and the beam waist respectively. These are shown graphically in Figure 8a. When this is done in 3 dimensions the results can be seen in Figure 8b. In this figure, the laser profile for each slice is a Gaussian profile. In addition to taking into account the quality of the laser, it is ray traced onto the object in order to account for shadowing of the build by the incoming material and non-planar effects where the center axis of the laser is not perpendicular to the work surface.

Other key features which have been developed in the system are the inclusion of temperature dependent material properties and the inclusion of work holding, shown in Figure 9. The larger section of material on the bottom of the substrate is the work holding. This work holding has modified material properties so that it can emulate the mass and material properties of whatever



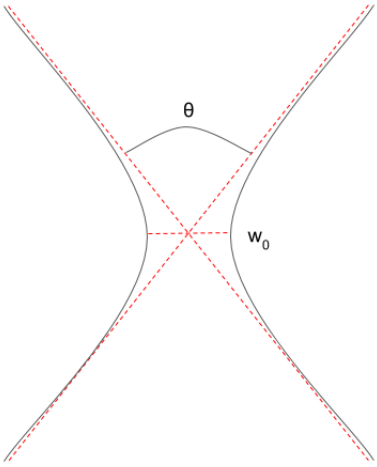
(a) Temperature profile

(b) Phase map

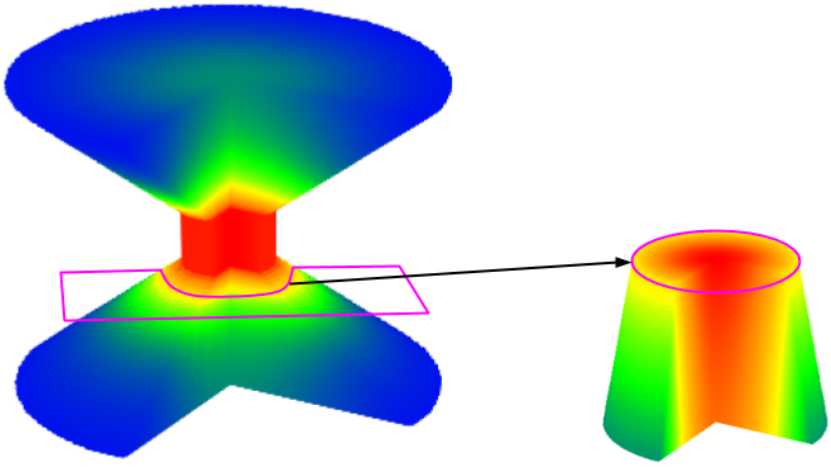


(c) Cooling rate map

Figure 7: Examples of data maps which can be expected from the simulation.



(a) 2-D laser representation, where the black lines are the edges of the laser, θ is the divergence angle and w_0 is the beam waist.



(b) 3-D representation of the laser profile.

Figure 8: 2-D and 3-D representation of the laser profile.

work holding is used in the process.

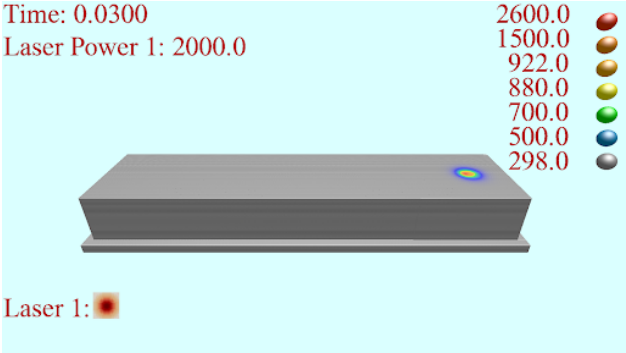


Figure 9: Example of the the inclusion of work holding in the simulation.

5 Case Study

The simulation that was described in Section 4 was used to generate the training data for the ML model. The parameters used in the simulation can be seen in Table 1.

Table 1: Parameters used in simulation

Parameter	Value
Material	Ti-64
Resolution	250 μm
Laser diameter	4.0 mm
Laser power	1000 W
Laser profile	TEM00
Laser scan speed	1273 mm/min
Powder stream diameter	15 mm
Powder stream rate	12 g/min

The training images were created in an attempt to emulate images which could be collected from a physical camera on deposition systems. These images were generated using the colors of steel which can be seen in Figure 10. These colors were used because of the extensive characterization which has been performed on steels and not been performed on Ti-64. These colors will serve as a starting point and proof of concept. In order to generate the images, the virtual camera was placed 50 mm above the melt pool and an image was created which was 1080 pixels by 1080 pixels based on what a physical camera would image. During the image generation, cold voxels which were recently inserted because of powder insertion were eliminated to clean up the image. Figure 11 shows a selection of the original and reconstructed (predicted) images from the test set after training of the autoencoder model.

Figure 12 shows the results of the next image prediction task for a test set. Its corresponding graph of mean squared error and structural similarity score values are shown in Figure 13. This is the combined output of both the autoencoder and LSTM models. As can be seen from the graph, all

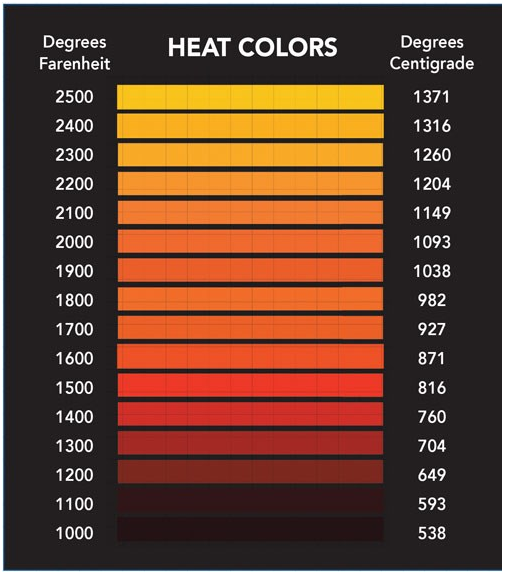


Figure 10: Temperature to color map used in image[28]

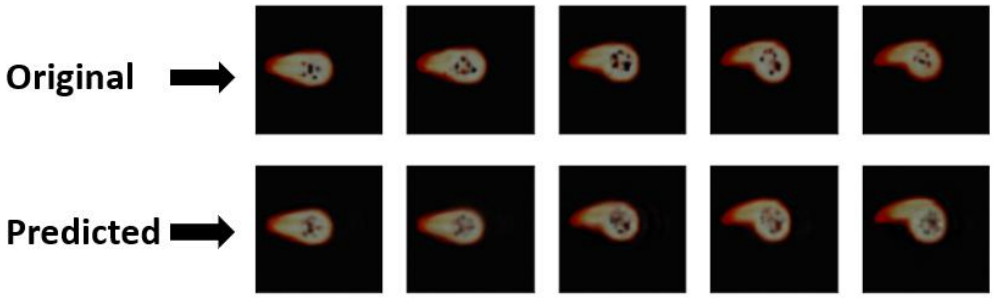


Figure 11: Image Reconstruction results from the trained autoencoder model

mean squared error values are below 0.001 across the test distribution and all structural similarity score values are above 0.985.

It was interesting to observe how our model will behave in the event of a change in direction. Figure 14 shows the results of the next image prediction task for a test set which covers a change in direction range. Its corresponding graph of mean squared error and structural similarity score values are shown in Figure 15. This is the combined output of both the autoencoder and LSTM models. As can be seen from the graph, all mean squared error values are below 0.001 across the test distribution except for a tiny spike to 0.004 around timestep 5 before it stabilizes and all structural similarity score values are above 0.97 except for a tiny spike to 0.94 around timestep 5. This is consistent with what one can observe from the images in Figure 14 as a start to a change in direction at timestep 5.

For an anomaly detection case study, we obtained simulated images with holes in the metal for the AM process at random timesteps. Figure 16 shows the results of the next image prediction task for a test set which covers examples of images with holes. Its corresponding graph of mean squared error and structural similarity score values are shown in Figure 17. As can be seen from the graph, all mean squared error values are below 0.001 across the test distribution except for huge

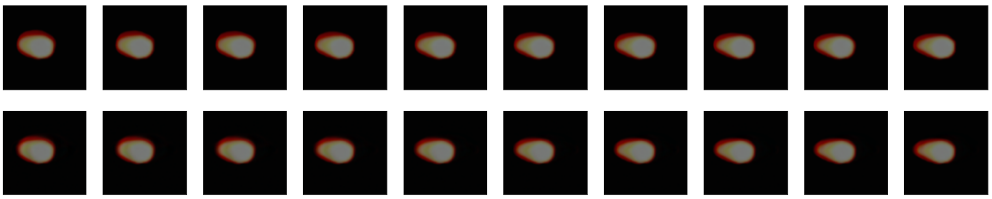


Figure 12: Autoencoder normal and predicted

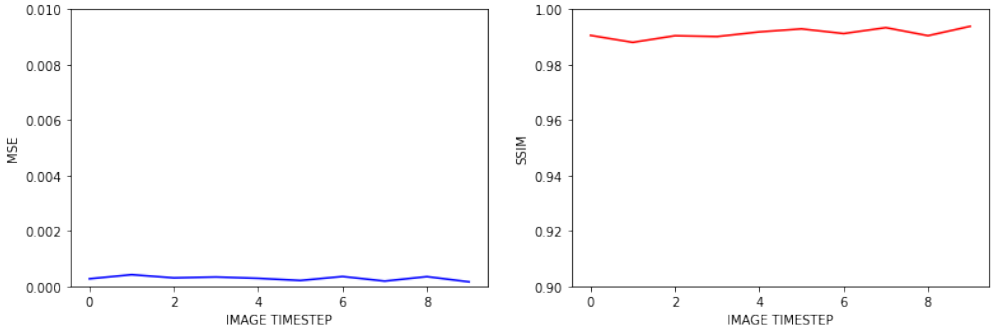


Figure 13: Autoencoder normal and predicted graph

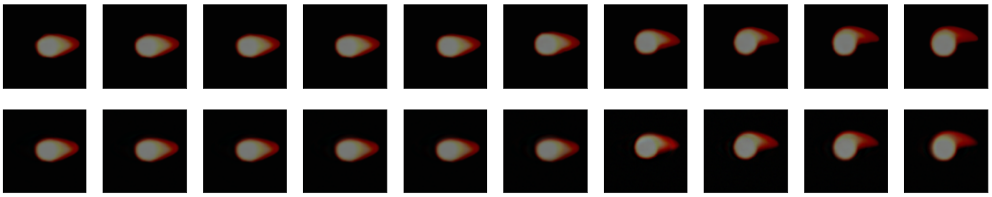


Figure 14: Autoencoder normal with change in direction

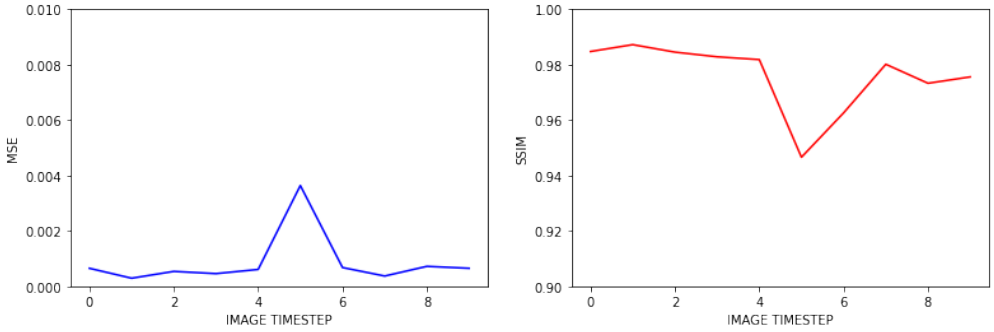


Figure 15: Autoencoder graph of normal with change in direction

spikes at timesteps 4 and 8 and all structural similarity score values are above 0.97 except for huge spikes at timesteps 4 and 8. This is also consistent with what one can clearly observe as holes in the images in Figure 16 at timesteps 4 and 8.

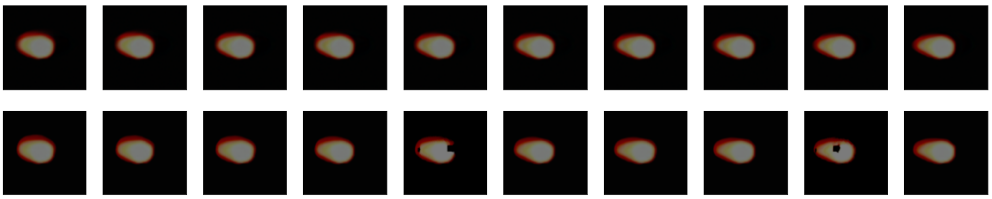


Figure 16: Autoencoder normal vs anomaly

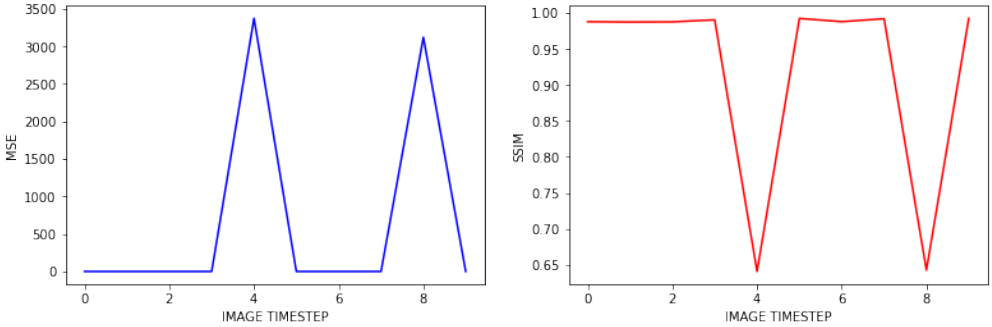


Figure 17: Autoencoder normal vs anomaly graph

6 Conclusion and Future Work

In this work, we framed the inference of Metal AM process states as a next image prediction problem. We first trained a CNN based autoencoder to carry out the identity function of reconstruction of a given image through a series of convolutions and deconvolutions. Then, we augmented the obtained latent vectors with process parameters to form our input at each timestep to be fed into an LSTM model for time series prediction. We used these two models to carry out the next image prediction in two stages: latent vector prediction with the LSTM model and image reconstruction from the decoder of the autoencoder model. Lastly, we simulated anomalies resulting from holes and our model was able to detect these anomalies through spikes in the response obtained from the selected metrics of mean squared error and structural similarity.

The next step in this work is to apply the Encoder/Decoder and LSTM models to physical AM systems. The intent of using the simulation images rather than raw simulation data for the original model training is to facilitate the transition to real in-process camera data. Two promising applications of the present work are:

Anomaly detection: Using methods analogous to the one presented in Section 3 on real in-process imagery, such as a welding camera, will enable real-time anomaly detection.

Internal state inference: While the present work focuses on surface-facing data, as captured by a simulated camera, the simulation contains data throughout the volume. Completing the link between the observed surface states and the behavior of what is happening below the surface will allow for exploration of the root causes of the anomaly detection. Understanding the causes of a problem create the opportunity to develop in-situ corrective strategies.

References

[1] Benjamin A Kwapong, Richard Anarfi, and Kenneth K Fletcher. "Collaborative Learning Using LSTM-RNN for Personalized Recommendation". In: *International Conference on Services Computing*. Springer. 2020, pp. 35–49.

[2] Benjamin A Kwapong, Richard Anarfi, and Kenneth K Fletcher. "Personalized service recommendation based on user dynamic preferences". In: *International Conference on Services Computing*. Springer. 2019, pp. 77–91.

[3] Heewoong Park, Sukhyun Cho, and Jonghun Park. "Word RNN as a baseline for sentence completion". In: *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*. IEEE. 2018, pp. 183–187.

[4] Kazuki Tsunematsu et al. "Neural Speech Completion". In: *Proc. Interspeech 2020* (2020), pp. 2742–2746.

[5] Huda A Al-Muzaini, Tasniem N Al-Yahya, and Hafida Benhidour. "Automatic Arabic Image Captioning using RNN-LST M-Based Language Model and CNN". In: *International Journal of Advanced Computer Science and Applications* 9.6 (2018).

[6] Sreelekshmy Selvin et al. "Stock price prediction using LSTM, RNN and CNN-sliding window model". In: *2017 international conference on advances in computing, communications and informatics (icacci)*. IEEE. 2017, pp. 1643–1647.

[7] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to Sequence Learning with Neural Networks". In: *CoRR* abs/1409.3215 (2014). arXiv: 1409.3215. URL: http://arxiv.org/abs/1409.3215.

[8] Javier Lorenzo et al. "RNN-based Pedestrian Crossing Prediction using Activity and Pose-related Features". In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020, pp. 1801–1806.

[9] Gen Karoji et al. "Pedestrian Dynamic Behaviour Modeling-An application to commercial environment using RNN framework". In: (2019).

[10] Niraj Bhujel, Eam Khwang Teoh, and Wei-Yun Yau. "Pedestrian trajectory prediction using rnn encoder-decoder with spatio-temporal attentions". In: *2019 IEEE 5th International Conference on Mechatronics System and Robots (ICMSR)*. IEEE. 2019, pp. 110–114.

[11] Yuexin Ma et al. "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 6120–6127.

[12] Yonghwan Jeong, Seonwook Kim, and Kyongsu Yi. "Surround vehicle motion prediction using LSTM-RNN for motion planning of autonomous vehicles at multi-lane turn intersections". In: *IEEE Open Journal of Intelligent Transportation Systems* 1 (2020), pp. 2–14.

[13] Khaled Saleh, Mohammed Hossny, and Saeid Nahavandi. "Intent prediction of vulnerable road users from motion trajectories using stacked LSTM network". In: *20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2017, pp. 327–332.

[14] Sarah K Everton et al. "Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing". In: *Materials & Design* 95 (2016), pp. 431–445.

[15] Sergey A Shevchik et al. “Acoustic emission for in situ quality monitoring in additive manufacturing using spectral convolutional neural networks”. In: *Additive Manufacturing* 21 (2018), pp. 598–604.

[16] K Wasmer et al. “In situ quality monitoring in AM using acoustic emission: A reinforcement learning approach”. In: *Journal of Materials Engineering and Performance* 28.2 (2019), pp. 666–672.

[17] K Ren et al. “Thermal field prediction for laser scanning paths in laser aided additive manufacturing by physics-based machine learning”. In: *Computer Methods in Applied Mechanics and Engineering* 362 (2020), p. 112734.

[18] Mojtaba Mozaffar et al. “Data-driven prediction of the high-dimensional thermal history in directed energy deposition processes via recurrent neural networks”. In: *Manufacturing letters* 18 (2018), pp. 35–39.

[19] Ziyang Zhang, Zhichao Liu, and Dazhong Wu. “Prediction of melt pool temperature in directed energy deposition using machine learning”. In: *Additive Manufacturing* 37 (2021), p. 101692. ISSN: 2214-8604. DOI: https://doi.org/10.1016/j.addma.2020.101692.

[20] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. “Speech recognition with deep recurrent neural networks”. In: *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee. 2013, pp. 6645–6649.

[21] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. “Recurrent neural network regularization”. In: *arXiv preprint arXiv:1409.2329* (2014).

[22] Mehmet Saygın Seyfioğlu, Ahmet Murat Özbayoğlu, and Sevgi Zubeyde Gürbüz. “Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities”. In: *IEEE Transactions on Aerospace and Electronic Systems* 54.4 (2018), pp. 1709–1723.

[23] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[24] Yann LeCun et al. “Backpropagation applied to handwritten zip code recognition”. In: *Neural computation* 1.4 (1989), pp. 541–551.

[25] Jayanth Koushik. *Understanding Convolutional Neural Networks*. 2016. arXiv: 1605.09081 [stat.OT].

[26] Frank Liou et al. “Multiscale and multiphysics modeling of additive manufacturing of advanced materials”. In: *NASA Langley Research Center, Hampton, VA, Technical, Report No. NASA/CR-2015-218691. https://ntrs. nasa. gov/search. jsp* (2015).

[27] Todd Eugene Sparks. *System and method for determining beam power level along an additive deposition path*. US Patent 9,573,224. Feb. 2017.

[28] Miles Free. *Heat Treat Colors for Steel*. Oct. 2015. URL: www.productionmachining.com/articles/heat-treat-colors-for-steel.