Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 126 (2024) 567–572

www.elsevier.com/locate/procedia

17th CIRP Conference on Intelligent Computation in Manufacturing Engineering (CIRP ICME '23)

17th CIRP Conference on Intelligent Computation in Manufacturing Engineering

Developing Rapid Metal AM deformations Prediction using CNN

Léopold Le Roux[a], Anthony Soroka[a], Ze Ji[a], Pierre Kerfiden[b] Samuel Bigot[a,*]

[a]*Cardiff University, The Parade, Cardiff CF24 3AA, Wales*
[b]*Centre des Matériaux, Mines ParisTech, Boulevard Saint-Michel ,Paris 75 272, France*

* Corresponding author. Tel.: +44 29208 75946. *E-mail address:* bigots@cardiff.ac.uk

Abstract

The production of complex 3D components using Metal Additive Manufacturing requires several steps combining different expertise, such as product design, process planning and manufacturing. This process chain would benefit significantly from faster deformation predictions enabling quicker iteration during parts design and manufacture, such as to evaluate or validate quickly a component expected final shape, internal stresses or deformation occurring during printing. This paper presents a new data-driven approach based on deep learning that can speed up the prediction of parts' geometrical deformations by creating a digital twin of an FEA simulation. A new CNN model founded on MeshCNN and designed for the processing of AM 3D models is described and methods for testing learning capabilities using training data generated by the FEA based software Simufact Additive are proposed.

Keywords: Deep learning; Metal Additive Manufacturing; Powder Bed Fusion; Simulation;CNN; MeshCNN

1. Main text

Powder Bed Fusion (PBF) is a highly promising metal powder based additive manufacturing (AM) technology, capable of fabricating metallic functional components with complex geometries whilst potentially using less raw material than subtractive methods. Metal BPF produced parts have the potential to be used in and to transform high-value manufacturing sectors including medical devices, aerospace, defense, energy and automotive industries.

Recent studies of metal AM have shown that defects need to be carefully controlled having identified that more research and theoretical modelling are necessary to overcome the challenges for metal AM such as the balling effect, micro-cracking/fractures, porosity, delamination, deformation, loss of alloying elements, oxide inclusions, intermetallic phases, and un-melted particles [1, 2].

Digital Twins (DT) as virtual replicas of physical devices have been extensively studied within manufacturing as they have shown great potential in enabling advanced process control, process optimization, and monitoring, incorporating (real-time) manufacturing data management. Furthermore the concept of DT is considered an appropriate approach to overcome current challenges associated with additive printing of metals such as: a lack of process robustness, stability and repeatability caused by the unsolved complex relationships between material properties, product design, process parameters, process signatures, post AM processes and product quality that have significantly impeded the broad acceptance in industry [3].

10.1016/j.procir.2024.08.241

MeshCNN is a general-purpose deep neural network for 3D triangular meshes, which can be used for tasks such as 3D shape classification or segmentation. This paper proposes a new data-driven approach based on deep learning that can speed up the prediction of parts' geometrical deformations by creating a digital twin of a finite element analysis (FEA) simulation. A new model founded on MeshCNN and designed for the processing of AM 3D models is described (Additive Manufacturing MeshCNN – AMMeshCNN) and methods of assessing its learning capabilities using training data generated by the FEA based software Simufact Additive are described.

2. Background

2.1. Powder Bed Fusion

Powder Bed Fusion (PBF) is an additive manufacturing technique that can create objects by melting and fusing layers of powder material on top of each other's, producing geometrical features with resolutions typically ranging from 80 to 250 µm [4]. The powder used can be made of metal, alloys, polymers or ceramics and different energy sources can also be used. This results in a range of different types of processing technologies, such as Selective laser melting (SLM) [5], Selective laser sintering (SLS) or Selective Electron beam melting (SEMB) [6].

An energy source sends an energy ray (for example a laser beam) to a group of mirrors and lenses that are used to redirect and focus its energy on the area of powder that is to be melted, is it termed the melt pool area [7]. By guiding the energy beam the melt pool is moved around the powder bed to create the layer shape, melting the powder layer and previous melted layers to ensure a strong link between the layers. After each layer, the printing bed is moved downward and new powder is taken from the powder reservoir and spread on the printing area by a recoater.

For each layer, the targeted 2D shapes are produced by slicing 3D models in multiple layers that are then converted into 2D machining paths for the focused energy. The quality of the produced parts is influenced by many parameters that will vary depending on the powder material, including focused energy path (e.g. hatch distance, scanning speed, power, recoating speed, recoating thickness).

Nomenclature

AM Additive Manufacturing
CNN Convolutional Neural Network
DT Digital Twin
FEA Finite Elements Analysis
FEM Finite Elements Methods
GNN Graph Neural Network
MeshCNN Mesh Convolutional Neural Network
MLP Multi-layer Perceptron
PBF Powder Bed Fusion
SLM Selective Laser Melting
SLS Selective Laser Sintering
SEMB Selective Electron Beam Melting

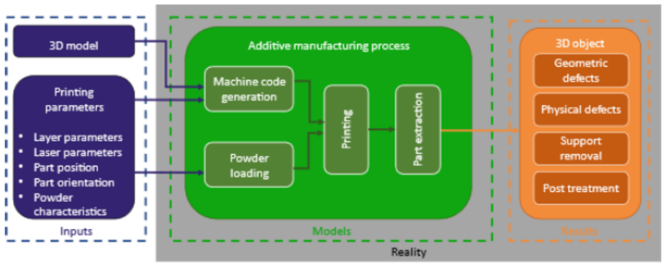

Fig. 1. AM Process

Furthermore, after printing, parts generally need to be post-processed, for example, to cut out supports, to remove un-melted powder or improve surface and metallurgic characteristics (Figure 1). These processes will also influence the quality of the produced parts.

Thus, the difficulties related to the proper tuning of parameters involved in these manufacturing steps become a major drawback in many PBF processes. In particular when processing metals, with a high chance of defect formation [1].

2.2. Modelling of deformations in additive manufacturing

Additive Manufacturing processes have enabled the fabrication of complex shapes that are not achievable with traditional manufacturing processes. However, this new technology is subject to numerous drawbacks, particularly in terms of difficulties in controlling defects that can affect the part's integrity, such as porosity [1], cracks, delamination or deformation. The cracks or delamination can be caused by the presence of residual stresses created by temperature variations during the process, which can alter the geometries or damage the part [8].

Dimensional and geometric accuracies can be affected by shrinkage or warping [8] effects. If these deformations are predicted correctly, some of them can be compensated for in the design stages to still meet the geometrical tolerances required by a product. However, in extreme cases, the increase in elevation caused by these deformations, if not contained during the printing stage, can cause the recoating system to collide with the part [8], potentially leading to damage to the machine or the part. Consequently, accurately predicting deformations is a critical aspect of the design stage involving PBF.

The deformations can be relatively accurately predicted using simulation methods such as finite element methods (FEM) with a physics-based model. However, the accuracy of these predictions is dependent on the settings chosen by the user (e.g. type of physics based models, time steps used, physical resolution), and this has a significant impact on the computational costs and accuracy. Generally, increasing the accuracy of a physics-based simulation will increase its computational cost, resulting in a trade-off between high accuracy and an acceptable computational time that reduces accuracy.

Surrogate modelling can be an effective solution to this computation time drawback, as data driven models learn the statistical behavior of the problem from data and not using physical equations. For general simulation, some examples of applications already exist, such as by Sanchez-Gonzalez et al. [9] who used deep learning with a graph neural network (GNN) to simulate water, goop, and sand physics. Where material particles at a time t are modelled using points clouds, with material features encoded in each simulation point, and using a GNN to predict the future acceleration of the particles.

This method provided good results for particle-based simulation. AM simulation-related research activities have been focusing on more traditional time-consuming simulation approaches. Thus, the use of deep learning could be a new promising approach to significantly speed up AM simulations.

As the production of an object using AM methods requires several steps combining different expertise such as product design, process planning and manufacturing. They would benefit significantly from faster deformation predictions enabling quicker iterations during part design, such as evaluating or validating a part's expected shape or its internal stresses, to assess whether the amount of deformation generated during printing is compatible with the industrial needs or to avoid collision with the printer recoating system. But furthermore, a fast simulation could be used with in-process monitoring systems, to follow the real state of parts being printed and update simulation predictions in real-time based on monitored information. Thus, allowing more informed decision-making in case of defects or deviations during printing. This paper proposes a new data-driven approach based on deep learning that can speed up the prediction of parts' geometrical deformations by creating a digital twin of an FEA simulation. This approach can be used to assess the expected shape of a part, or to avoid collision with the printer's recoating system.

3. MeshCNN

MeshCNN (Mesh Convolutional Neural Network), the network adopted as the basis for the system being proposed in this paper, was introduced by Hanocka et al [10] and specifically developed to leverage the inherent features of mesh representation through the implementation of self-created convolution features and pooling layers. This approach has been demonstrated to provide favorable outcomes in mesh classification and segmentation tasks. The operators implemented by MeshCNN operate on the manifold mesh structure, exploiting its topology to gain insight into the object being represented. The input features for MeshCNN are stored on the edges of the mesh and its pooling operation is designed to reduce the number of mesh edges.

Figure 2 below illustrates the architecture of MeshCNN for the purpose of segmentation. The segmentation architecture is divided into two components, an encoding component that extracts information and features from the mesh, and a decoding component that uses the extracted features to classify the edges. The encoding component, referred to as the "up part", employs a MeshPool layer to collapse edges and simplify the mesh.

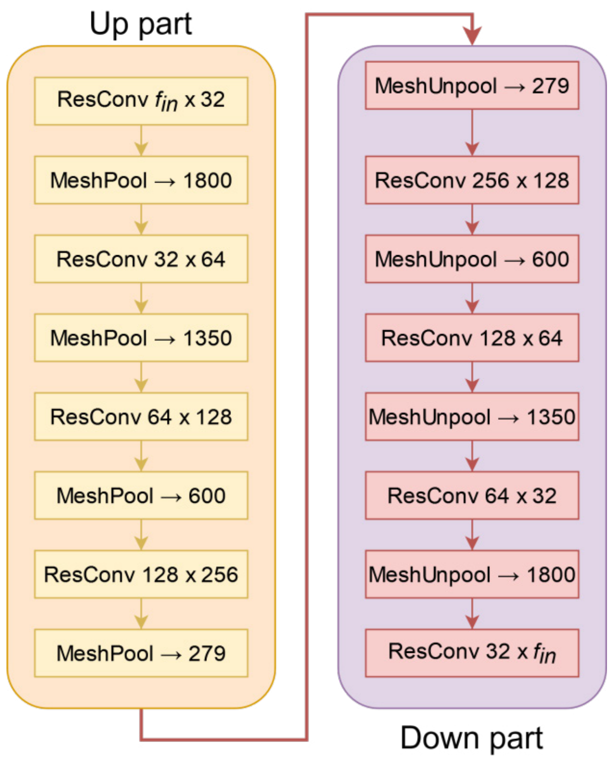

Fig. 2. MeshCNN architecture for segmentation

The MeshConv, a convolution operator, performs a convolution using the updated mesh to identify neighbors' edges. The decoding component referred to as the "down part", features a MeshUnpool operation that reverses the collapse executed by the previous MeshPool, restoring the mesh to its original shape. The output of the MeshCNN segmentation model is the reconstructed mesh with classification scores assigned to all its edges.

An advantage of MeshCNN is that its pooling layers respect the geometry of the part by merging linked vertices where a classical pooling layer will not know of the link between the vertices and will simply merge edge features that are close in the matrix features.

The features used as inputs of the convolution layers are computed for each edge, and 5 features are extracted per edge:

- The dihedral angle (Figure 3 α2)
- The 2 inner angles (Figure 3 α0 and α1)
- 2 edge-length (Figure 3 r0 and r1) ratios for each edge.

Those extracted features make MeshCNN not affected by all rotation, translation and scale variations as those features will not change if a part is moved and scaled. Extracted features are sent to the convolution layer for feature extraction. Classical convolution layers are expected to receive fixed feature orders. For example, on an image, changing the order, the position of the pixels will drastically change the images. But for a mesh, what matter is the connection between the vertices, not their orders, as changing the order of the vertex or the edges will not modify the part. Making MeshCNN using a custom logic and working on the edges of the mesh.

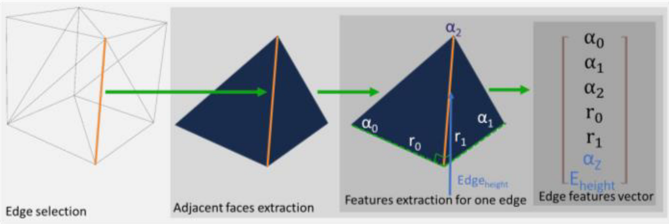

Fig. 3. MeshCNN features

The convolution will take the features of the edges (Figure 3) and those of its four neighbors on the mesh. The convolution is defined by equation 1.

$$e \cdot k_0 + \sum_{j=1}^{4} k_j \cdot e^j \qquad (1)$$

Where: e is the processed edge features, e^j the adjacent edges features, k the kernel matrix.

To guarantee that the convolution is invariant to the ordering of the input data, the inputs e^j are preprocessed with some basic operation, as described in equation 2 to ensure that the edge neighbors' order does not influence the results. With the neighbor edges a, b, c and d, the adjacent edges features are computed with:

$$(e^1, e^2, e^3, e^4) = (|a - c|, a + c, |b - d|, b + d) \qquad (2)$$

For edges with the lowest feature magnitudes, the MeshCNN pooling layer collapse them by merging voxels until a target number of remaining edges is achieved. When an edge is collapsed as in Figure 4, the collapse of one edge will create a new vertice with the average position of the two former vertices. This operation transforms 5 edges into 2.

4. Proposed AMMeshCNN

MeshCNN [10] was chosen as the foundation for a new CNN based on its ability to operate on 3D meshes and the feasibility of incorporating geometrical features into its feature extractors. This neural network was adapted to facilitate the prediction of deformation resulting from physics-based simulation. The training was conducted using parts processed under consistent printing conditions, thus, as demonstrated in Figure 5, only necessitating 3D models as inputs for the CNN to predict the deviations generated by the AM process. In AM manufacturing, post-printing effects such as shrinkage and thermal deformation modify the shape of parts (Paul et al [11]) due to the thermal stresses resulting from metal cooling and the geometry of previous layers, impacting the cooling rate and induced stress. Thus, the CNN model must consider the shape, scale, and orientation of the part, as these factors will impact the behavior of the process.

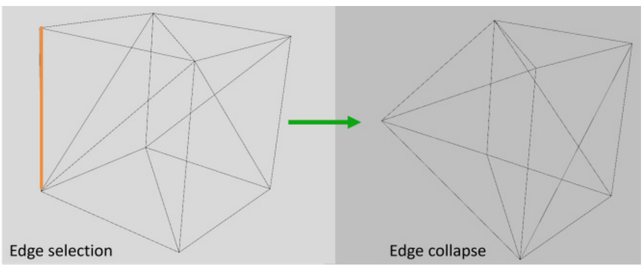

Fig. 4. Edge collapse

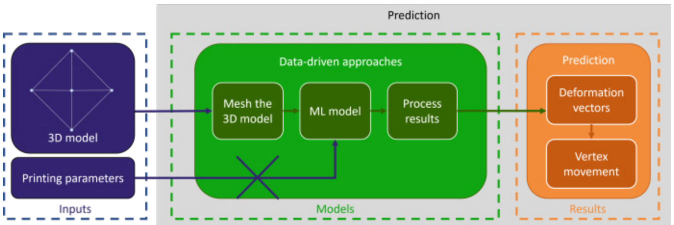

Fig. 5. Deep learning approaches

The original version of MeshCNN is inadequate for this purpose, as the computed features do not encompass information regarding part orientation. As in the case of predicting deviations in additive manufacturing, the orientation of the part can affect the laser path and cooling dynamics, thereby modifying the deviations. Additionally, the size of the part influences the thermal inertia, which can alter the deviations as well. In light of these considerations, this paper proposes the integration of specific features to address the impact of orientation and size on deviation prediction in additive manufacturing processes. To make the network suitable for AM, the following modification are proposed:

• Make the network sensitive to the part orientation by adding angle invariance features.
• Make the network sensitive to the part scale by adding relevant objects scale features
• Add layers of neurons to the network's up and down parts, to make it more flexible and capable of understanding more complex physical rules.
• Store MeshCNN results on to the meshes' vertices to loss computation and for display purposes.

In this study, the modified version of MeshCNN referred to as AMMeshCNN – Additive Manufacturing MeshCNN has been proposed. The network has been altered in order to predict continuous values for edge displacement rather than discrete values for classification and segmentation. With changes to the loss function used for training and the final layers of the network in order to predict three values for each axis rather than a single class probability output.

5. AMMeshCNN

5.1. Angle invariance

The first type of invariance addressed was the rotation invariance in the X and Y axes, as it was known that the part orientation on those axes had an influence on their deformation results and will have a significant impact on the accuracy of MeshCNN in predicting edge displacement values. To achieve this, the angle between the edge and the Z axis was calculated and included as an additional feature in the edge feature vector.

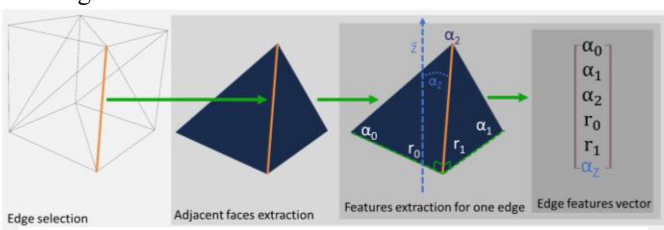

Fig. 6. Edges features with α_z feature

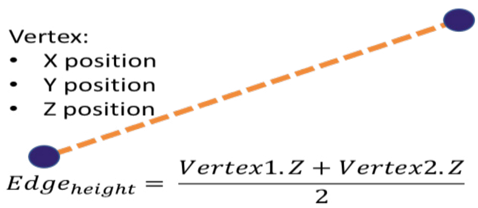

Fig. 7. Edge height feature

It was hypothesized that rotation around the Z-axis would not significantly affect the accuracy of the predictions as our simulation software was not calibrated with different printing axis, making this type of invariance not considered in the study. The calculation of the angle between the edge and the Z was performed using the following equations 3 and 4.

$$\alpha_z = \pi - \cos^{-1}(\overrightarrow{Z_{normal}} \cdot \overrightarrow{edge}) \quad (3)$$

$$\alpha_z = \pi - \cos^{-1}\left(\begin{bmatrix}0\\0\\1\end{bmatrix} . \begin{bmatrix}vertice_1.x - vertice_2.x\\vertice_1.y - vertice_2.y\\vertice_1.z - vertice_2.z\end{bmatrix}\right) \quad (4)$$

With this new feature, the feature extracted α_z, was added to the features extraction procedure described in Figure 6.

5.2. Scale invariance

The scale invariance was the second to be removed in this research. Given that the size of a printed object affects the deviation behavior (such as shrinkage, thermal deformation, and stress), it was crucial for the network to be mindful of this. To accomplish this, the height position of each edge was added to the feature vector. This enabled the network to be aware of the object's size and to perceive the width and depth of the object in relation to its height. The height of an edge was computed as the average height of the two vertices connected by that edge, as depicted in Figure 7. This new feature was then incorporated into the edge feature vectors, as demonstrated in the updated feature extraction illustration in Figure 8.

5.3. More neurons

The architecture of AMMeshCNN was further modified to incorporate a multi-layer perceptron (MLP) at the end of the decoder. The modification is depicted in Figure 9. To implement the MLP, the output dimension of the last ResConv of the down part was set to a constant value of 512 to provide features to the MLP. The perceptron layers are represented as "Layer Xin − > Xout ", where Xi n refers to the input dimension and Xout refers to the output dimension. The

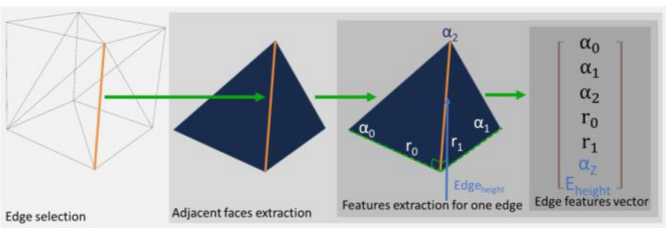

Fig. 8. Deep learning approaches

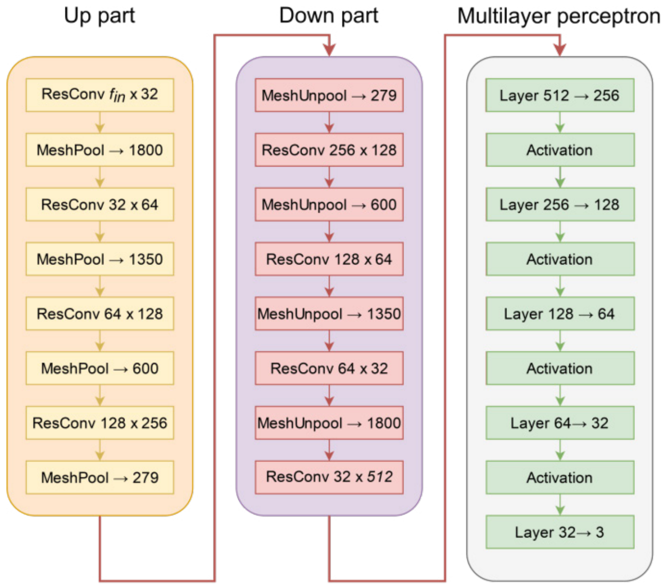

Fig. 9. AMMeshCNN architecture for mesh deformation

activation function, represented as "*Activation*", is a Rectified Linear Unit (ReLU) [12], defined as shown in equation 5:

$$f(x) = \max(0, x) \quad (5)$$

5.4. From vertex features to edge features

In this research, a challenge was encountered when comparing the results of AMMeshCNN, which predicts deformation on edges, with the simulation results stored on mesh vertices. To address this discrepancy, a method was developed to transfer the edge results of AMMeshCNN onto the vertices (as depicted in Figure 5.20), by calculating the average values of the edges connected to a vertex. Making it possible to compute the loss of the model through the comparison between the results and labels. This operation is described in equation 6 below.

$$V_i = \frac{1}{n}\sum_{j \in N_E(V_i)} E_j \quad (6)$$

Where Vi is the deformation features of the vertices *i, n* the number of edges connected to these vertices, $N_E(V_i)$ the edges neighbourings to the vertices i and Ej the deformation features of the edges j.

4. Methodologies for testing AMMeshCNN

In order to determine if the proposed technique is fit for this purpose, AMMeshCNN will have to prove its reliability on both simple and more complex datasets that will be produced using Simufact Additive. To assess the capacity of AMMeshCNN to learn simple deviations on simple basic mesh shapes, a simple shape dataset was created, existing primarily of simple cubes. This dataset is comprised of randomly generated cubes with deformations generated through an inherent strain method. AMMeshCNN was trained with the MSE loss with a learning rate of 0.001, for 50 epochs and then 40 epochs with a learning rate decay method. To assess the quality of the prediction,

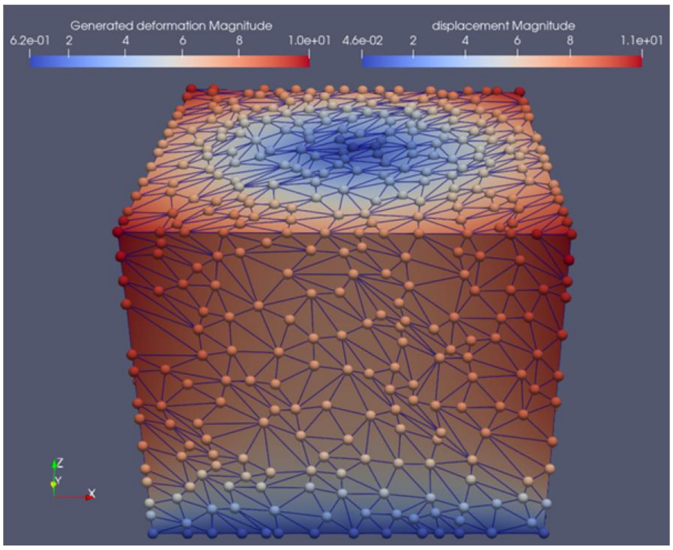

Fig. 10. Cube Prediction

Figure 10 shows a cube and its label displayed and on top of it, the predicted vertex is displayed. The cubes and the predicted vertex (spheres) are colored using the magnitude of the deformation (predicted for the vertex, label for the cubes). The differences between the labels and prediction is hard to see demonstrating the network's effective learning of the functions.

However, the performance of this method needs to be evaluated using more details and on more complex 3D shapes. This is currently being tested, using real-life examples from the Thingi10k dataset [13] and the results will be published in the near future to confirm the accuracy of the proposed method.

5. Conclusions

An enhanced version of a Mesh Convolutional Neural Network (called AMMeshCNN), modified to be suitable for Additive Manufacturing has been proposed in this paper. With the aim of proving a CNN based surrogate model is capable of performing quick predictions of printing deformations.

The proposed model enhances MeshCNN by adding:

• Sensitivity to part orientation through angle invariance.

• Sensitivity to the part scale

• New layers of neurons, to make the CNN more able to learn complex physical rules.

• The ability to store MeshCNN results on the meshes' vertices for loss computation and display purposes.

Further work will focuse on the full training and testing of AMMeshCNN on more complex and real-world 3D models.

Acknowledgements

This research was supported by the project Additive Manufacturing using Metal Pilot Line (MANUELA), which received funding from the European Union's Horizon2020 research and innovation programme under grant agreement No. 820774.

References

[1] Aminzadeh, M. and T.R. Kurfess, Online quality inspection using Bayesian classification in powder-bed additive manufacturing from high-resolution visual camera images. Journal of Intelligent Manufacturing, 2019. 30: p. 2505-2523.

[2] Sing, S.L., et al., Emerging metallic systems for additive manufacturing: In-situ alloying and multi-metal processing in laser powder bed fusion. Progress in Materials Science, 2021. 119: p. 100795.

[3] Liu, C., et al., Digital twin-enabled collaborative data management for metal additive manufacturing systems. Journal of Manufacturing Systems, 2022. 62: p. 857-874.

[4] Amini, M., S.I. Chang, and P. Rao, A cybermanufacturing and AI framework for laser powder bed fusion (LPBF) additive manufacturing process. Manufacturing Letters, 2019. 21: p. 41-44.

[5] Wong, K.V. and A. Hernandez, A review of additive manufacturing. International scholarly research notices, 2012. 2012.

[6] Körner, C., Additive manufacturing of metallic components by selective electron beam melting—a review. International Materials Reviews, 2016. 61(5): p. 361-377.

[7] Schoinochoritis, B., D. Chantzis, and K. Salonitis, Simulation of metallic powder bed additive manufacturing processes with the finite element method: A critical review. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2017. 231(1): p. 96-117.

[8] Grasso, M. and B.M. Colosimo, Process defects and in situ monitoring methods in metal powder bed fusion: a review. Measurement Science and Technology, 2017. 28(4): p. 044005.

[9] Sanchez-Gonzalez, A., et al. Learning to simulate complex physics with graph networks. in International conference on machine learning. 2020. PMLR.

[10] Hanocka, R., et al., Meshcnn: a network with an edge. ACM Transactions on Graphics (TOG), 2019. 38(4): p. 1-12.

[11] Paul, R., S. Anand, and F. Gerner, Effect of thermal deformation on part errors in metal powder based additive manufacturing processes. Journal of manufacturing science and Engineering, 2014. 136(3).

[12] Nair, V. and G.E. Hinton. Rectified linear units improve restricted boltzmann machines. in Proceedings of the 27th international conference on machine learning (ICML-10). 2010.

[13] Zhou, Q. and Jacobson, A. Thini10k: A dataset of 10,000 3d-printing models. 2016.