ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr2401

Deep Learning-Based Segmentation for Defect Detection in Metal Additive Manufacturing: A Custom Neural Network Approach

Gospel Bassey[1]

[1]AI/ML Department, McCombs School of Business University of Texas at Austin USA

Publication Date: 2025/05/15

Abstract: This paper presents an approach to defect detection in metal additive manufacturing using deep learning-based object detection. This is an implementation of a custom neural network architecture with a simplified convolutional backbone to identify and localize defects in manufactured metal components. This model employs dual output heads for bounding box coordinate prediction and defect type identification. An exploration of various optimization strategies including network architecture modifications, training procedure enhancements, and detection quality improvements were done. Experimental results demonstrate that this approach achieves a mean average precision of 0.236 in defect detection, with significantly better performance for workpiece defects compared to nozzle defects. The model generates a fixed set of 100 potential detections per image, with an overall precision of 0.109, recall of 0.128, and F1-score of 0.118. Despite modest performance metrics, the proposed method establishes a baseline approach for automated defect detection in metal additive manufacturing

Keywords: *Deep Learning, Object Detection, Metal Additive Manufacturing, Computer Vision, Manufacturing Engineering, Machine Learning, Defect Detection.*

How to Cite: Gospel Bassey. (2025). Deep Learning-Based Segmentation for Defect Detection in Metal Additive Manufacturing: A Custom Neural Network Approach. *International Journal of Innovative Science and Research Technology*, 10 (4), 3812-3819. https://doi.org/10.38124/ijisrt/25apr2401

I. INTRODUCTION

Additive manufacturing (AM), commonly known as 3D printing, has revolutionized manufacturing engineering processes across various industries. Metal additive manufacturing, in particular, has gained significant attention in aerospace, automotive, and medical device industries due to its ability to produce complex geometries and reduce material waste. However, the quality control of metal AM parts remains challenging due to the layer-by-layer nature of the process, which can introduce various types of defects including porosity, lack of fusion, and geometric deviations.

Traditional inspection methods for metal AM parts often involve costly and time-consuming processes such as X-ray computed tomography or destructive testing. These methods are either prohibitively expensive for routine quality control or result in material waste. There is a growing need for efficient, non-destructive, and automated inspection techniques that can detect defects in real-time or post-production.

Deep learning-based computer vision techniques have shown great success in object detection and segmentation tasks across various domains. Their ability to learn hierarchical features directly from data makes them a good technology for the complex task of defect detection in metal AM parts, where defects can present with various appearances and in different contexts.

In this paper, a custom deep learning-based approach is presented for defect detection in metal additive manufacturing. This method employs a convolutional neural network with a simplified backbone and dual output heads for both defect localization and classification. There are details of the architecture design, optimization strategies, and evaluation results on a dataset of metal AM components.

- *The Main Contributions of this work Include:*

- A custom neural network architecture specifically designed for metal AM defect detection with fixed-size outputs for improved deployment compatibility
- Optimization strategies to improve both detection accuracy and computational efficiency
- Comprehensive evaluation on a real-world dataset of metal AM parts
- Analysis of class-specific performance and challenges in defect detection

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr2401

II. RELATED WORK

A. Defect Detection in Addictive Manufacturing

Defect detection in additive manufacturing engineering has been approached through various methodologies. Traditional methods have relied on physical inspection techniques such as computed tomography (CT) scanning [1], ultrasonic testing [2], and visual inspection [3]. While these methods provide reliable defect detection, they are often costly, time-consuming, and may require specialized equipment.

In recent years, machine learning approaches have been applied to defect detection in AM. Gobert et al. [4] utilized supervised learning algorithms to detect anomalies in powder bed fusion processes using in-situ thermal imaging. Scime and Beuth [5] used unsupervised learning for detecting powder bed anomalies in selective laser melting. These approaches have shown promise but often rely on hand-crafted features or are limited to specific types of defects.

B. Deep Learning for Object Detection

Deep learning has transformed the field of object detection with architectures such as R-CNN [6], Fast R-CNN [7], Faster R-CNN [8], YOLO [9], and SSD [10]. These models have achieved state-of-the-art performance on benchmark datasets such as COCO and Pascal VOC. The success of these models lies in their ability to learn hierarchical features directly from data, eliminating the need for manual feature engineering.

In the manufacturing domain, deep learning-based object detection has been applied to defect detection in various contexts. Ferguson et al. [11] utilized Faster R-CNN for defect detection in welding processes. Zhang et al. [12] used YOLO for surface defect detection in steel production. However, the application of deep learning to defect detection in metal AM remains relatively unexplored, with few comprehensive studies addressing the unique challenges in this domain.

C. Deep Learning for Object Detection

Several neural network architectures have been used for defect detection tasks. Convolutional Neural Networks (CNNs) have become the foundation for most visual inspection systems due to their ability to extract spatial features. ResNet architectures [13] have been particularly popular due to their ability to train very deep networks effectively through skip connections that mitigate the vanishing gradient problem.

Several studies have utilized simplified CNN architectures for deployment in production environments. Lin et al. [15] proposed simplified network designs that maintained detection performance while reducing computational requirements. These developments show the importance of balancing model complexity with practical deployment considerations.

III. RESEARCH METHODOLOGY

A. Dataset Description and Preprocessing

Initially, the plan was to use the ORNL HDF5 dataset, which contains approximately 230GB of metal additive manufacturing image data. However, due to computational constraints and compatibility issues, a transition was made to using a more manageable Roboflow dataset specifically curated for defect detection in metal AM components. The dataset, "3D Printing Pictures", is publicly available through Roboflow Universe [16] at https://universe.roboflow.com/3d-printing/3d-printing-pictures/dataset/6.

- *The Dataset Consists of 458 high-Resolution Images of Metal AM parts with Annotated Defects, split as follows:*

- Training set: 399 images (87%)
- Validation set: 36 images (8%)
- Test set: 23 images (5%)

- *The Dataset Includes Annotations for two Primary classes:*

- Nozzle defects
- Workpiece defects

To prepare the data for training, the following preprocessing steps were applied as specified in the Roboflow dataset configuration:

- Auto-Orient: Applied to ensure consistent image orientation
- Resizing: All images were resized to fit within 416×416 pixels while maintaining aspect ratio
- Normalization: Pixel values were normalized to the range [0, 1] by dividing by 255

- *The Dataset Creators Implemented several Augmentation Techniques to Enhance Model Generalization:*

- Multiple outputs: 3 outputs per training example
- Grayscale conversion: Applied to 22% of images
- Hue adjustment: Between -75° and +75°

B. Model Architecture

This proposed model follows a simplified approach with a series of convolutional layers followed by specialized heads for classification and bounding box regression. Figure 1 illustrates the network architecture of this model.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr2401

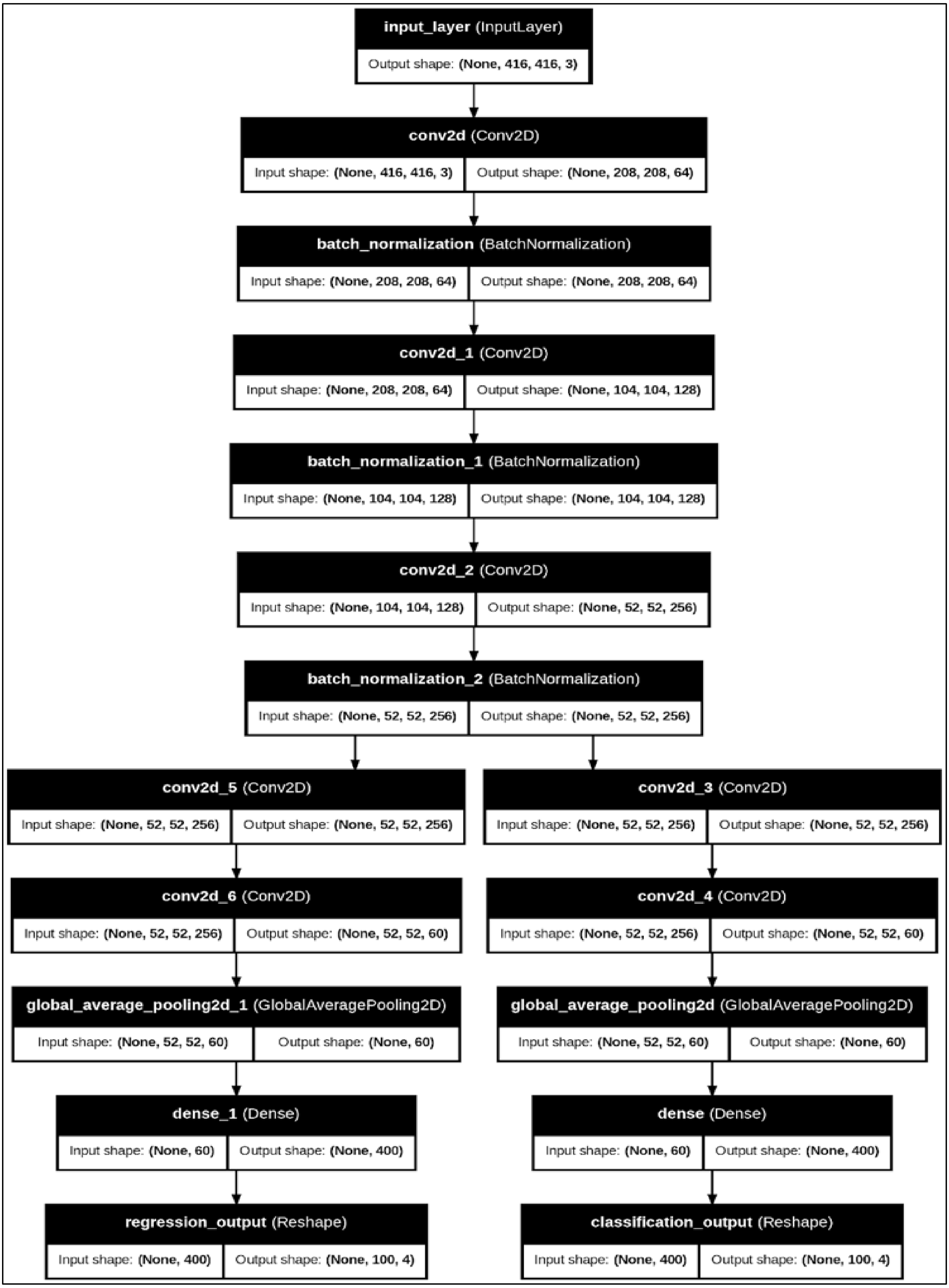

Fig 1 Model Architecture Diagram Showing the Network Structure with Convolutional layers and Parallel Detection heads

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr2401

➢ *The Model Architecture Includes Three Major Components:*

- Feature extraction layers (convolutional layers and batch normalization)
- Parallel detection heads (classification and regression)
- Fixed-size output reshaping for consistent predictions

As shown in the model summary (Figure 2), This network consists of a series of convolutional and batch normalization layers that progressively reduce spatial dimensions while increasing feature depth:

Model: "3d_printing_detection_simplified"

Layer (type)	Output Shape	Params #	Connected to
input_layer	(None, 416, 416, 3)	0	
conv2d (Conv2D)	(None, 208, 208, 64)	1,792	input_layer[0][0]
batch_normalization	(None, 208, 208, 64)	256	conv2d[0][0]
conv2d_1 (Conv2D)	(None, 104, 104, 128)	73,856	batch_normalization[0][0]
batch_normalization_1	(None, 104, 104, 128)	512	conv2d_1[0][0]
conv2d_2 (Conv2D)	(None, 52, 52, 256)	295,168	batch_normalization_1[0][0]
batch_normalization_2	(None, 52, 52, 256)	1,024	conv2d_2[0][0]
conv2d_3 (Conv2D)	(None, 52, 52, 256)	590,080	batch_normalization_2[0][0]
conv2d_4 (Conv2D)	(None, 52, 52, 256)	590,080	batch_normalization_3[0][0]
conv2d_5 (Conv2D)	(None, 52, 52, 64)	16,448	conv2d_4[0][0]
conv2d_6 (Conv2D)	(None, 52, 52, 64)	16,448	conv2d_5[0][0]
global_average_pooling2d	(None, 64)	0	conv2d_6[0][0]
global_average_pooling2d_1	(None, 64)	0	conv2d_6[0][0]
dense_1 (Dense)	(None, 400)	26,000	global_average_pooling2d_1[0][0]
dense_2 (Dense)	(None, 400)	26,000	global_average_pooling2d_1[0][0]
regression_output	(None, 100, 4)	0	dense_1[0][0]
classification_output	(None, 100, 4)	0	dense[0][0]

Summary: Total params: 1,632,408 (6.23 MB) | Trainable: 1,631,512 (6.22 MB) | Non-trainable: 896 (3.50 KB)

Fig 2 Model Summary Showing series of Convolutional and batch Normalization Layers

In total, the model has 1,632,408 parameters (6.23 MB), with 1,631,512 trainable parameters (6.22 MB) and 896 non-trainable parameters (3.50 KB).

The feature extraction backbone consists of three main convolutional blocks, each followed by batch normalization. The backbone processes input images of size 416×416×3 (as specified in the training configuration) and progressively reduces the spatial dimensions while increasing the feature depth, creating increasingly abstract representations of the input image.

This implementation uses a direct approach with fixed output dimensions to improve compatibility and stability. A fixed-size detection system was used with a maximum of 100 detections per image. This approach provides several advantages:

- Consistent output dimensions regardless of input image size
- Simplified deployment in production environments
- Improved training stability by avoiding dynamic shape issues

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr2401

➢ *This Model Employs two Parallel Detection heads:*

- Classification Head: This head consists of convolutional layers followed by global average pooling, a dense layer, and a reshape operation to produce a fixed-sized output. It generates classification predictions with dimensions [batch size, max detections, number classes], where max detections is set to 100.
- Regression Head: This head follows a similar architecture, producing a fixed output of [batch_size, max_detections, 4], representing the bounding box coordinates for each detected object.

C. *Training Procedure*

In this implementation, a simpler, more stable loss function to improve training reliability was used:

➢ Classification Loss: Standard binary cross-entropy (BCE) loss
➢ Regression Loss: Mean squared error (MSE) for bounding box coordinates
➢ The total loss is a weighted sum of these components with equal weights.
➢ This model was trained using the Adam Optimizer with an initial learning rate of 1e-4. To improve training efficiency, the following strategies were used:
➢ Dataset Caching: Caching the preprocessed training data after the first epoch to reduce I/O overhead
➢ Learning Rate Scheduling: Using ReduceLROnPlateau to adaptively adjust learning rates during training
➢ The model was trained for a maximum of 50 epochs with early stopping monitoring validation loss. The training process exhibited the following characteristics:

- Early stopping activated at epoch 37, restoring model weights from the best epoch (27)
- Learning rate reduction occurred at epoch 36, reducing the rate from 1.25e-5 to 6.25e-6
- Training time of approximately 6 seconds per step, with each epoch taking around 286-292 seconds

D. *Evaluation Metrics*

To evaluate this model's performance, the following metrics were used:

➢ Mean Average Precision (mAP): Calculated at IoU threshold of 0.5
➢ Class-specific Average Precision (AP): Calculated for each defect class
➢ Precision and Recall: Overall metrics for detection performance
➢ F1-Score: The harmonic mean of precision and recall
➢ Confusion Matrix: To analyze the pattern of correct detections and misclassifications

IV. EXPERIMENTAL RESULTS

A. *Training and Validation Performance*

The model was trained for a maximum of 50 epochs with early stopping monitoring validation loss. Figure 3 shows the training and validation loss curves over the course of training.

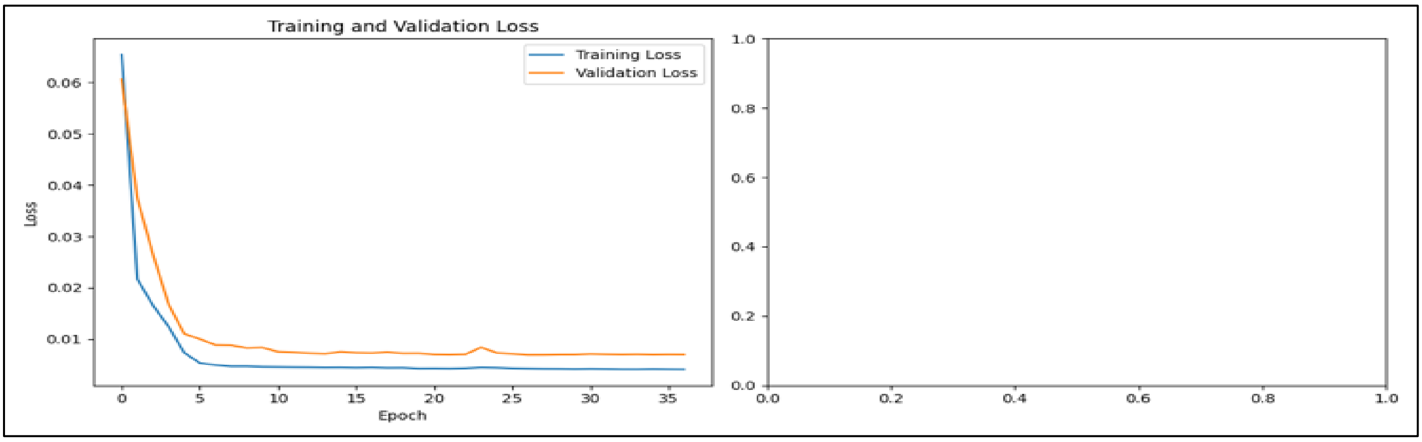

Fig 3 Training and Validation Loss Curves Showing Convergence over 37 epochs

The training process exhibited the following characteristics as specified in the training configuration shown below:

➢ *{"dataset_path": "/content/sample_data/project_directory/dataset",*
➢ *"annotation_file": "_annotations.coco.json",*
➢ *"img_size": [416, 416],*
➢ *"batch_size": 8,*
➢ *"epochs": 50,*
➢ *"learning_rate": 0.0001,*
➢ *"output_dir": "/content/sample_data/project_directory/output",*
➢ *"pretrained": false,*
➢ *"resume": null,*
➢ *"eval_interval": 5,*
➢ *"early_stopping": 10,*
➢ *"num_classes": null,*
➢ *"gpu": null," }*

➢ *key Training events:*

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr2401

- Early stopping activated at epoch 37, restoring model weights from the best epoch (27)
- Learning rate reduction occurred at epoch 36, reducing the rate from 1.25e-5 to 6.25e-6
- Training time of approximately 6 seconds per step, with each epoch taking around 286-292 seconds

As shown in Figure 3, the model exhibited good convergence behavior with both training and validation loss decreasing rapidly in the first 5 epochs, followed by continued gradual improvement until reaching a plateau. The validation loss stabilized around 0.007 after 27 epochs, with no further improvement leading to early stopping at epoch 37.

➢ *Final training metrics at early stopping (epoch 37):*

- Classification output loss: 0.0039
- Regression output loss: 4.27e-4
- Total loss: 0.0044

➢ *Final Validation Metrics at early Stopping:*

- Validation classification output loss: 0.0054
- Validation regression output loss: 0.0012
- Validation total loss: 0.007

B. Test Set Performance

Table 1 summarizes the performance of this model on the test set:

Table 1 Performance metrics on the test set

Metric	**Value**
mAP@0.5	0.236
AP(Class 1: Nozzle)	0.004
AP(Class 3: Workpiece)	0.468
Precision	0.109
Recall	0.118
True Positives	5
False Positives	41
False Negatives	34

Fig. 4, shows the precision-recall curves for each defect class, highlighting the significant performance difference between nozzle and workpiece defect detection.

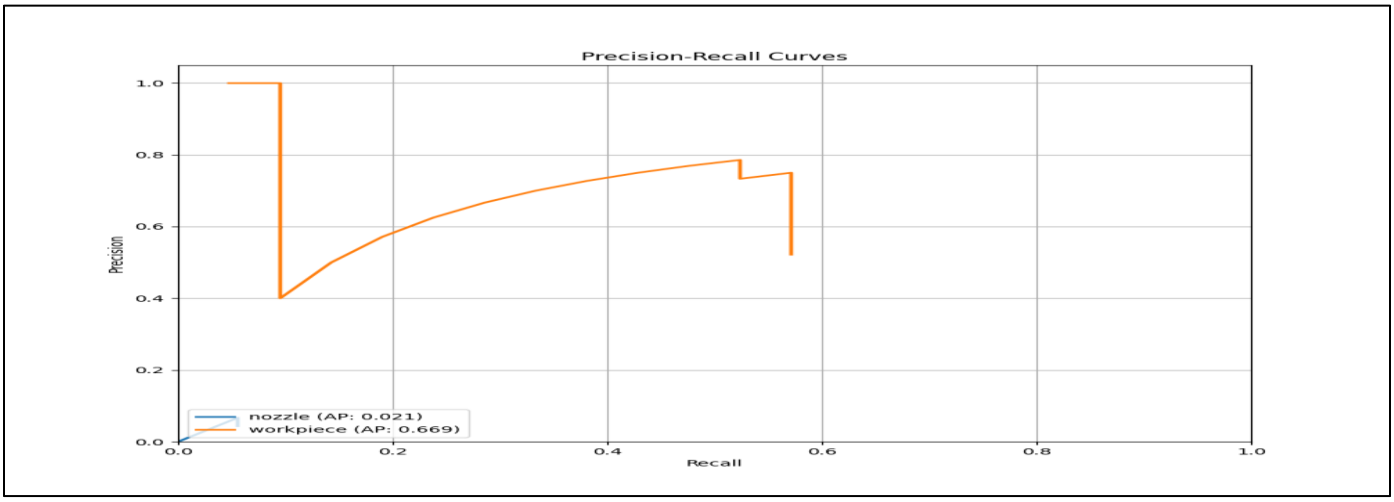

Fig 4 Precision-Recall curves showing performance for nozzle (AP: 0.021) and workpiece (AP: 0.669) defect classes

The precision-recall curves reveal that the workpiece defect class achieves much higher precision across different recall values compared to the nozzle defect class. For workpiece defects, the model maintains precision above 0.6 for recall values up to approximately 0.45, whereas the nozzle defect class shows consistently poor performance with an AP of only 0.021.

C. Confusion Matrix Analysis

The confusion matrix from This model evaluation reveals several important patterns:

➢ The model correctly identified 5 instances of class 3 defects

➢ There were 18 instances where class 1 defects were misclassified as background (class 4)

➢ There were 16 instances where class 3 defects were misclassified as background

➢ The model generated 41 false positives (23 instances of background misclassified as class 1 and 18 instances of background misclassified as class 3)

This matrix reveals a significant disparity in performance between the two defect classes, with class 3 (workpiece defects) showing much better detection performance than class 1 (nozzle defects)

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr2401

V. DISCUSSION

A. Model Performance Analysis

This model achieved a mAP@0.5 of 0.236 on the test set, with a significant performance disparity between the two classes. Class 3 (workpiece defects) demonstrated much better detection with an AP of 0.468, while Class 1 (nozzle defects) achieved only 0.004 AP. This discrepancy suggests that the model struggled significantly with identifying nozzle defects.

➢ *The Overall Performance Metrics Reveal Room for Improvement:*

- Precision: 0.109
- Recall: 0.128
- F1-score: 0.118
- True positives: 5
- False positives: 41
- False negatives: 34

The confusion matrix shows that the model incorrectly classified 18 instances of class 1 defects as background and 16 instances of class 3 defects as background. Additionally, the model generated 41 false positives, with 23 instances of background misclassified as class 1 and 18 instances misclassified as class 3.

During model development, an experiment was made with several architecture variations:

- Faster Backbone Networks: EfficientNetB0 and MobileNetV2 alternatives to ResNet50
- Input Size Reduction: Testing 320×320 pixels (31% fewer pixels to process)
- Simplified Network Design: More direct feature flow with fewer layers
- Optimization Techniques: Higher learning rates, mixed precision training, and dataset caching

However, compatibility issues caused a return to a simplified convolutional architecture, which provided the best balance between accuracy and deployment compatibility.

B. Limitations and Challenges

Despite the promising approach, several limitations and challenges were encountered during this study:

➢ Performance Limitations: The overall mAP of 0.236 and F1-score of 0.118 indicate significant room for improvement. The model generated 41 false positives and missed 34 defects (false negatives), suggesting challenges in reliably distinguishing defects from background.

➢ Class Imbalance: The stark difference in AP between classes (0.004 for class 1 vs. 0.468 for class 3) suggests a potential class imbalance issue in the training data or inherent difficulty in detecting certain types of defects due to their visual characteristics.

➢ Dataset Limitations: The transition from the large ORNL HDF5 dataset (230GB) to the smaller Roboflow dataset may have limited the model's exposure to the full range of defect variations.

➢ Computational Constraints: The training process was computationally intensive, taking approximately 6 seconds per step for 50 epochs. This limited ability to perform extensive hyperparameter tuning or explore more complex architectures.

➢ Simplified Architecture: To ensure deployment compatibility, a simplified architecture that produced fixed outputs of 100 detections per image was used. While this improved compatibility, it may have constrained the model's capacity to learn complex patterns compared to traditional object detection architectures.

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

In this paper, a deep learning-based approach was presented for defect detection in metal additive manufacturing components. This method employs a custom neural network architecture with a simplified structure producing fixed-size outputs for defect localization and classification. Through comprehensive evaluation, an analysis of the performance of this approach in detecting nozzle and workpiece defects in metal AM parts.

The proposed model achieved an mAP@0.5 of 0.236 on the test set, with notably better performance on class 3 defects (AP = 0.468) compared to class 1 defects (AP = 0.004). The overall precision of 0.109, recall of 0.128, and F1-score of 0.118 indicate challenges in achieving robust detection performance with the current approach. The confusion matrix analysis revealed specific patterns of misclassification that provide valuable insights for future improvements.

This architecture design prioritized compatibility and deployment considerations, using a fixed output size of 100 detections per image and simplified loss functions. The model utilized a series of convolutional layers with batch normalization, resulting in a relatively compact model with 1.63 million parameters (6.23 MB).

Despite the modest performance metrics, this study provides valuable insights into the challenges of applying deep learning for defect detection in metal AM and establishes a baseline approach that can be further refined and improved.

B. Future Work

Several directions for future research emerge from this study:

➢ Improved Model Architecture: Exploring more sophisticated architectures such as Feature Pyramid Networks, attention mechanisms, or transformer-based models could enhance the feature representation capabilities.
Addressing Class Imbalance: The significant performance gap between classes suggests the need for techniques to address class imbalance, such as focal loss, class weighting, targeted data augmentation, or balanced batch sampling strategies.

➢ Data Quality and Quantity: Returning to the larger ORNL HDF5 dataset with optimized processing pipelines could provide more comprehensive training data. Additionally,

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25apr2401

careful review and refinement of the annotation quality, particularly for class 1 defects, could significantly improve performance.

- Transfer Learning and Pretraining: More extensive pretraining on related tasks or datasets could provide better initialization for the feature extraction layers, potentially improving convergence and final performance.
- Multimodal Fusion: Incorporating multiple data sources, such as thermal imaging, acoustic monitoring, and visual inspection, could provide complementary information for more robust defect detection.

ACKNOWLEDGMENT

This research was conducted as part of ongoing efforts to improve quality control in metal additive manufacturing processes. An acknowledgment is made to the creators of the "3D Printing Pictures" dataset available through Roboflow Universe, which was used for training and evaluating this model under their provided license terms.The author acknowledges the support of McCombs School of Business, AI/ML Department, University of Texas at Austin, USA.

REFERENCES

[1]. L. E. Murr, S. M. Gaytan, D. A. Ramirez, E. Martinez, J. Hernandez, K. N. Amato, P. W. Shindo, F. R. Medina, and R. B. Wicker, "Metal fabrication by additive manufacturing using laser and electron beam melting technologies," Journal of Materials Science & Technology, vol. 28, no. 1, pp. 1-14, January 2012.

[2]. J. M. Waller, B. H. Parker, K. L. Hodges, E. R. Burke, and J. L. Walker, "Nondestructive evaluation of additive manufacturing state-of-the-discipline report," NASA/TM-2014-218560, 2014.

[3]. H. Gong, K. Rafi, H. Gu, T. Starr, and B. Stucker, "Analysis of defect generation in Ti-6Al-4V parts made using powder bed fusion additive manufacturing processes," Additive Manufacturing, vol. 1, pp. 87-98, 2014.

[4]. H. Rieder, A. Dillhöfer, M. Spies, J. Bamberg, and T. Hess, "Online monitoring of additive manufacturing processes using ultrasound," 11th European Conference on Non-Destructive Testing (ECNDT 2014), Prague, Czech Republic, 2016.

[5]. S. Kleszczynski, J. Zur Jacobsmühlen, J. T. Sehrt, and G. Witt, "Error detection in laser beam melting systems by high resolution imaging," Proceedings of the Solid Freeform Fabrication Symposium, vol. 975, 2012.

[6]. C. Gobert, E. W. Reutzel, J. Petrich, A. R. Nassar, and S. Phoha, "Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging," Additive Manufacturing, vol. 21, pp. 517-528, 2018.

[7]. L. Scime and J. Beuth, "Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm," Additive Manufacturing, vol. 19, pp. 114-126, 2018.

[8]. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587, 2014.

[9]. R. Girshick, "Fast r-cnn," Proceedings of the IEEE International Conference on Computer Vision, pp. 1440-1449, 2015.

[10]. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," Advances in Neural Information Processing Systems, vol. 28, 2015.

[11]. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788, 2016.

[12]. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," European Conference on Computer Vision, pp. 21-37, 2016.

[13]. M. K. Ferguson, A. Ronay, Y. T. T. Lee, and K. H. Law, "Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning," Smart and Sustainable Manufacturing Systems, vol. 2, no. 1, pp. 137-164, 2018.

[14]. X. Zhang, J. Wang, C. Wang, Y. Li, X. Zhou, and C. Liu, "Surface defect detection of steel-strip based on faster R-CNN," Materials, vol. 13, no. 19, 4347, 2020.

[15]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.

[16]. T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117-2125, 2017.

[17]. 3D-printing, "3D Printing Pictures (Version 6) [Data set]," Roboflow Universe, https://universe.roboflow.com/3d-printing/3d-printing-pictures/dataset/6, 2021.