Detecting Anomalies in Laser Powder Bed Fusion with Computer Vision

Sean Tronsen; Nathan DeBardeleben PhD | LANL HPC-DES
Colt Montgomery, PhD | LANL SIGMA-1

Overview

Introduction

Laser powder bed fusion (LPBF) is a technique in additive manufacturing (3D printing) which uses a laser and scanning mirror to build metal and ceramic components with complex geometries layer by layer.

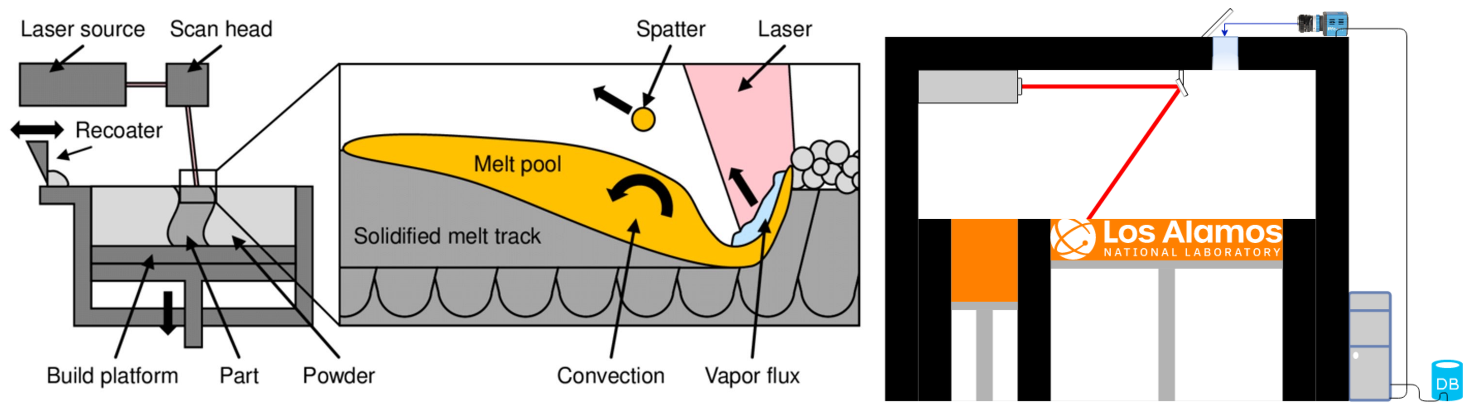


Figure 1. The diagrams above illustrate the process of spatter being ejected from the melt pool (left) and the physical positioning of the camera used to capture images for the project dataset relative to the powder bed.

Mechanical errors and fusion of redeposited debris remain at the source of issues in part quality and consistency. Over time, focus on these issues has increased alongside the adoption of this technology.

LA-UR-21-32202 Dataset

The project dataset consists of a brief sequence of 100 grayscale images which capture the state of the powder bed during a build. A scientific camera (sCMOS) was used to allow for a larger range of light intensities to be captured in comparison to common consumer devices. Despite this range, the laser remained bright enough to prevent dimmer features from being captured with fine detail. For this reason, spatter and anomalous debris tend to blend in with the image background despite being the target features of each image.

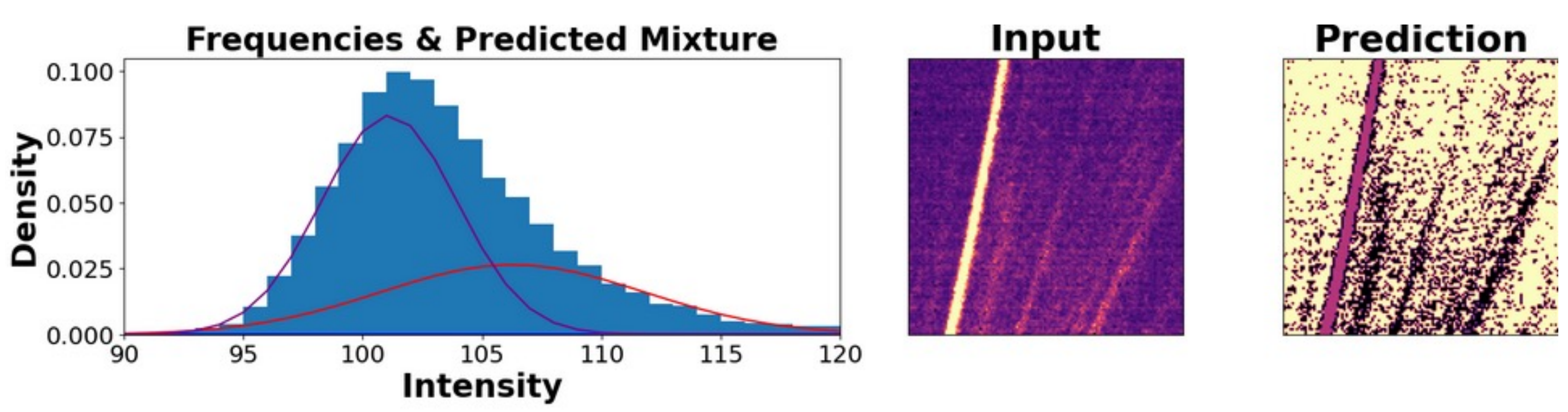


Figure 2. Similarities between spatter and background pixels cause both groups to blend, resulting in difficulties automating their separation. The image above represents an attempt by a simple machine learning model (Gaussian Mixture Model) to discern the difference between background pixels and spatter.

Modeling & Results

UNet for Semantic Segmentation with PyTorch

Traditional techniques fail to differentiate spatter from the noisy image background due to inability to consider both local and global contexts, a specialty of machine learning algorithms.

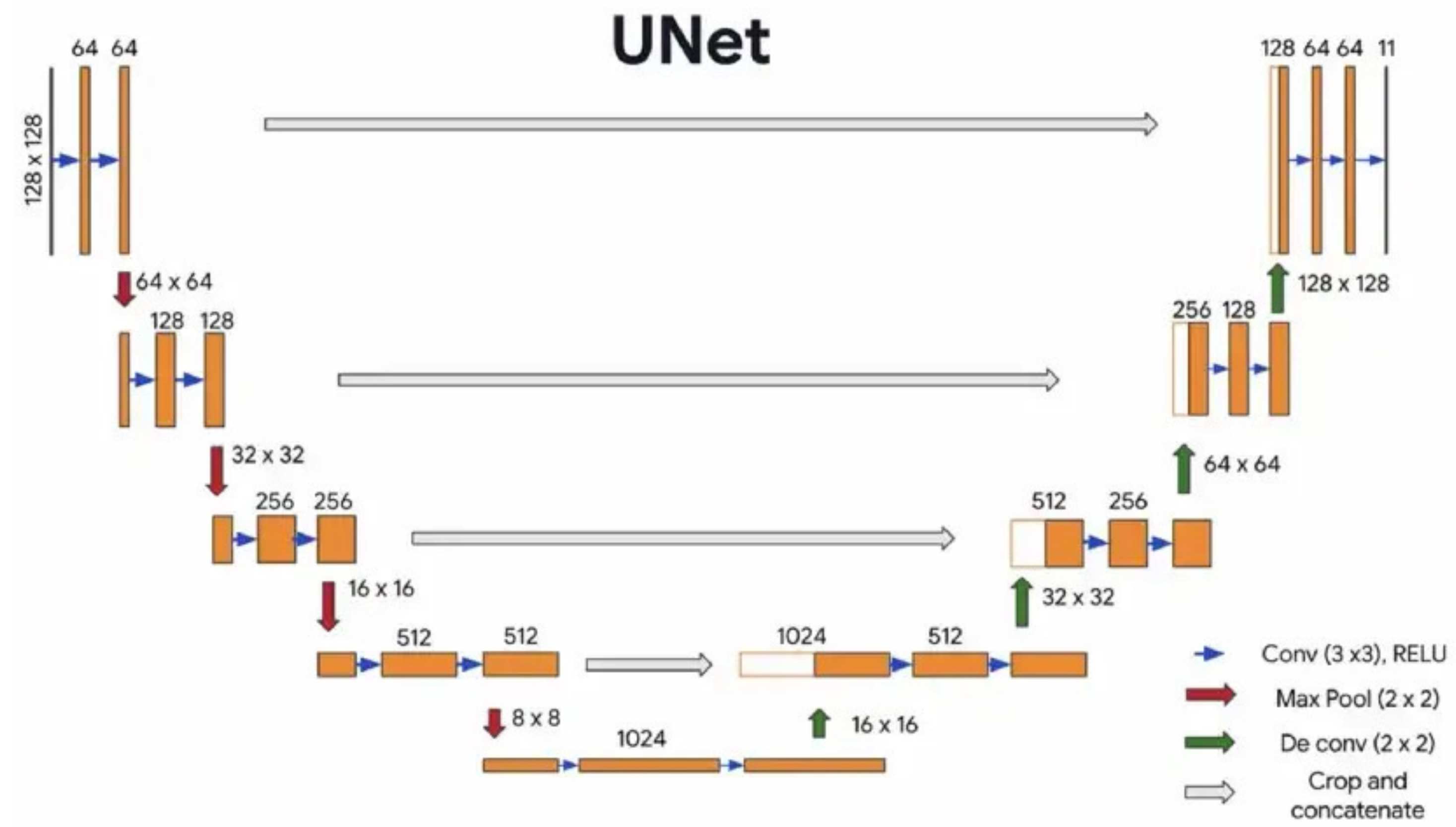


Figure 3. A diagram of the UNet architecture which illustrates the contracting, expanding, and residual paths used in its design.

Experimental Design

Three training designs were used to test 64 configurations of the architecture with the goal of determining best practices and to exploring performance tradeoffs between quality and time.

Results

Models trained on transformed images with weighted penalties for wrong predictions significantly outperformed their counterparts.

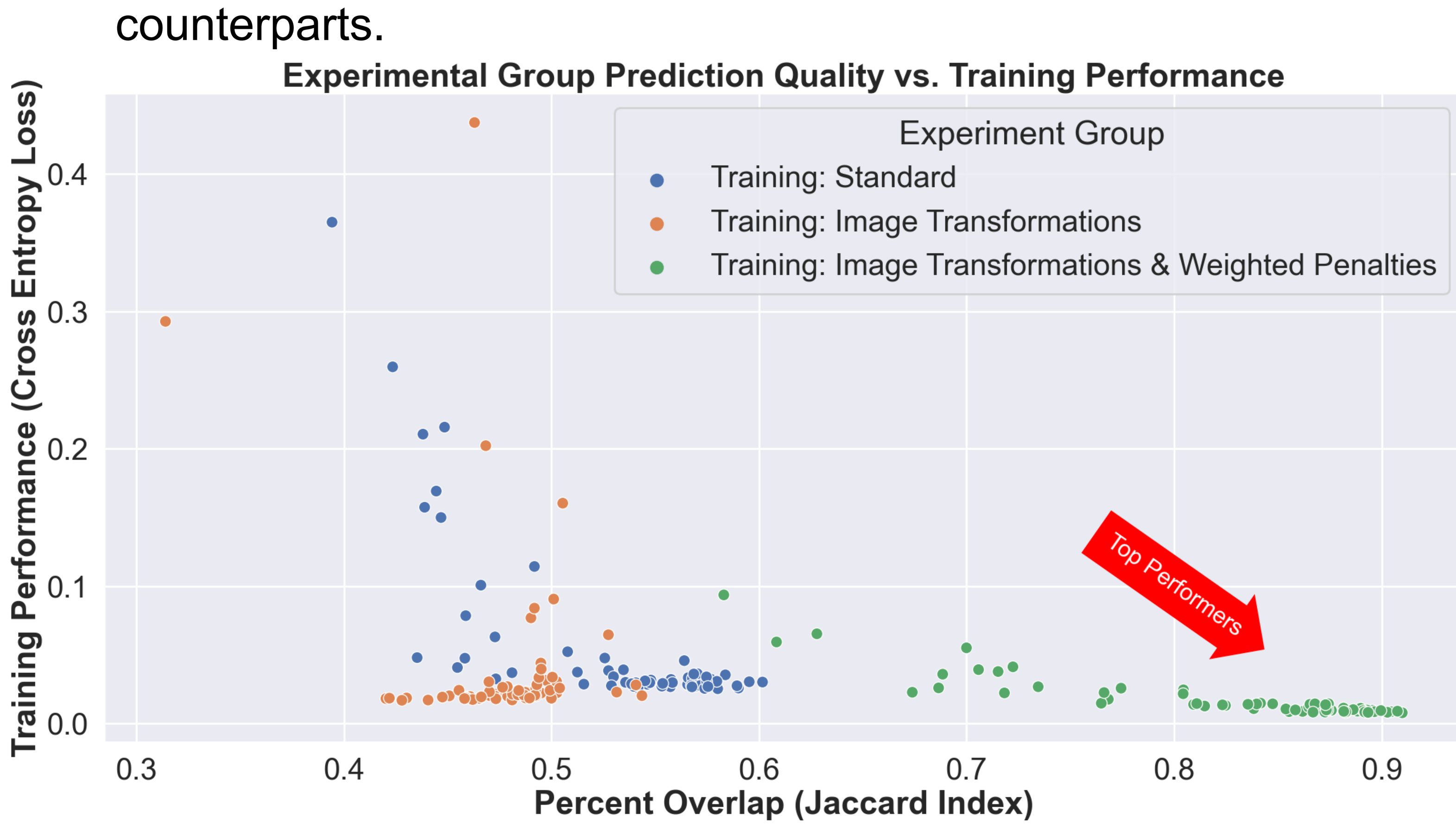


Figure 3. A scatterplot detailing performance for members of each training group. The clustering of the best performing group (green) and spacing from the others illustrates the importance of training strategies.

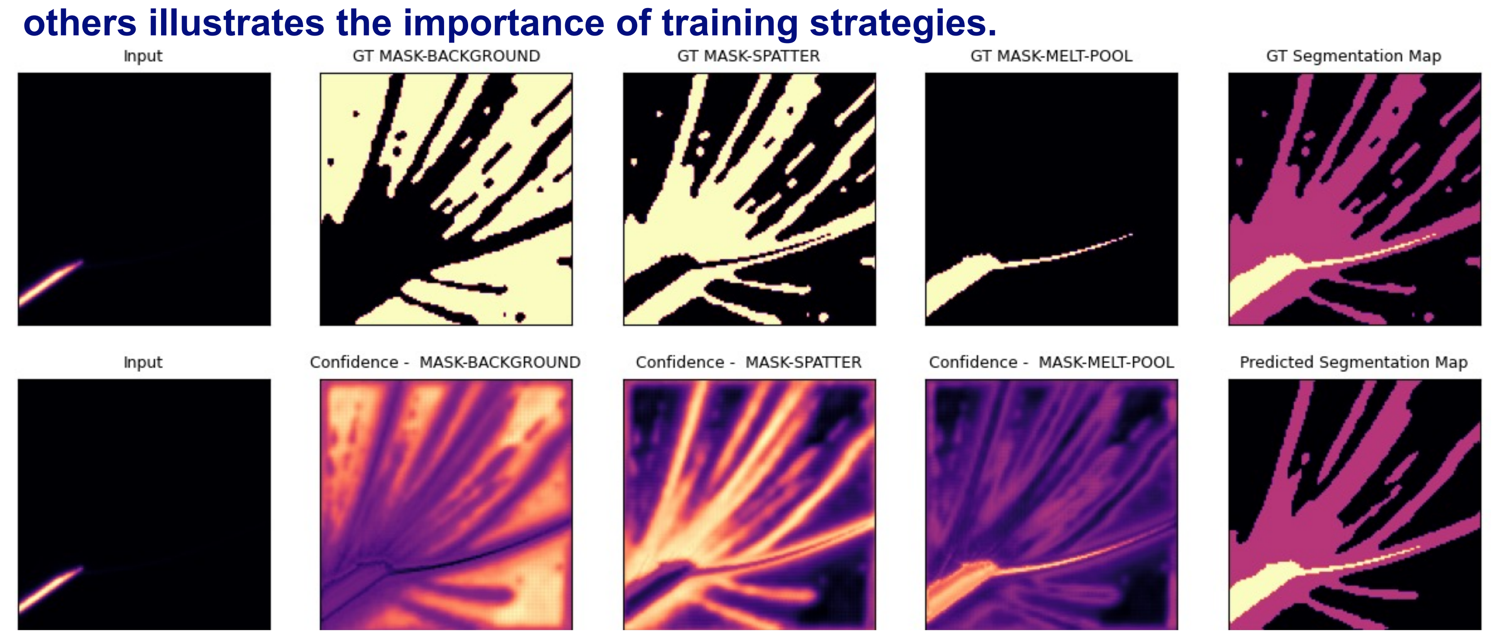
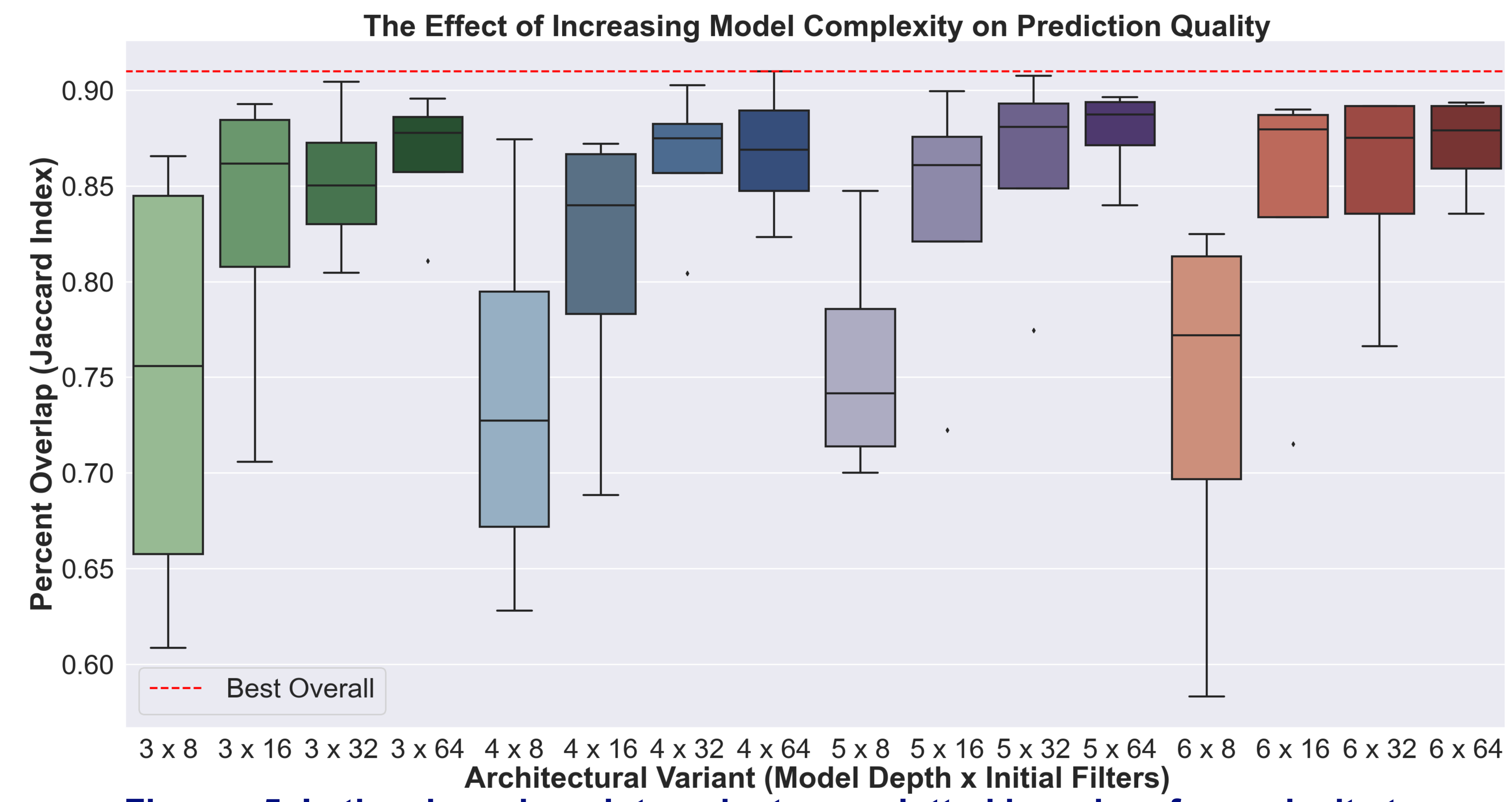
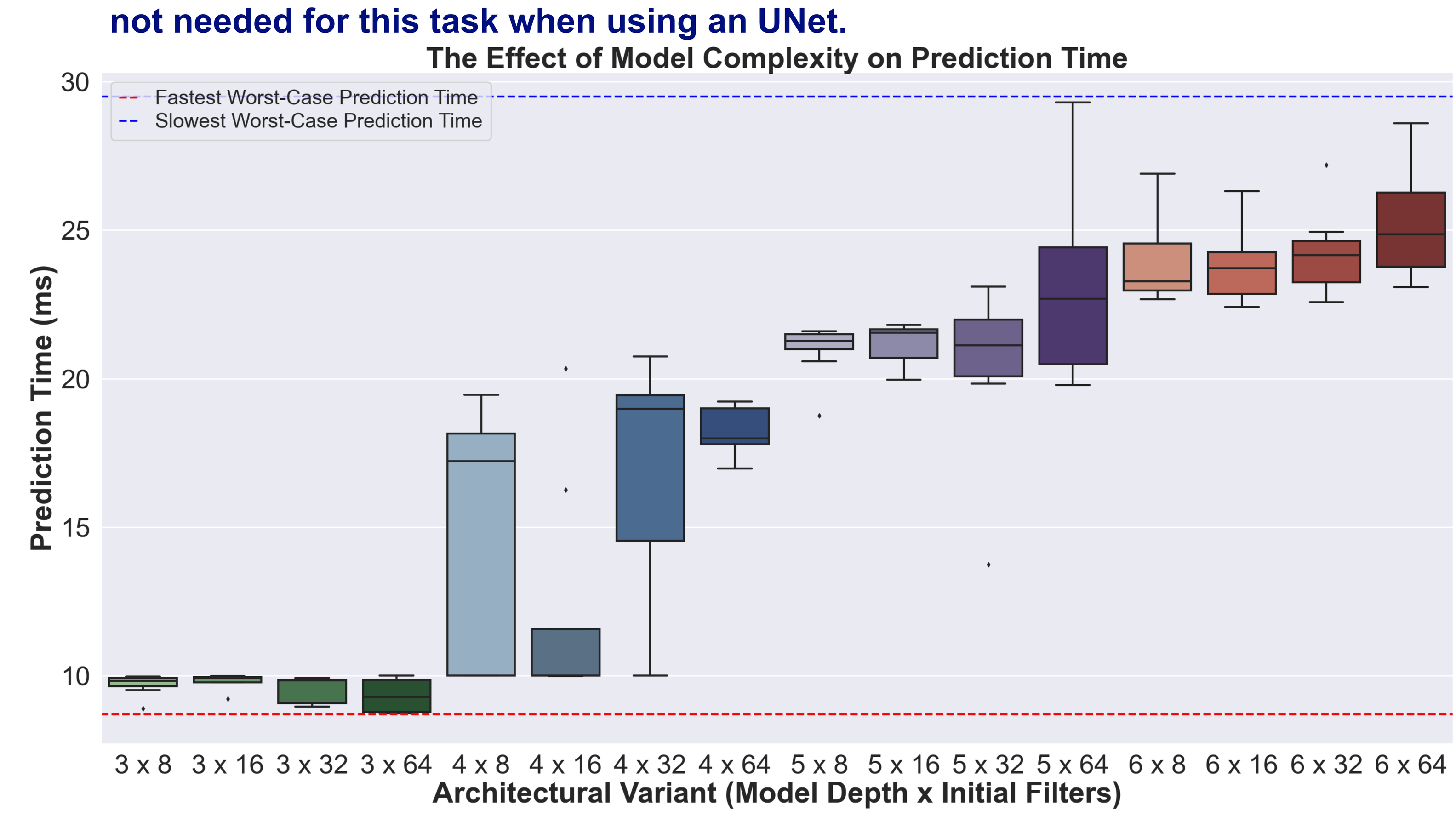


Figure 4. Model confidence (bottom) compared to the ground truth (top).



Figures 5. In the above boxplot, variants are plotted in order of complexity to illustrate similarities in prediction quality. From this, we can gather complexity is not needed for this task when using an UNet.



Figures 6. The boxplot above illustrates worst-case prediction times for each variant as a spread. As complexity and computation cost increase, so too does the amount of time required to perform inference.

Conclusion

Results from the experiment illustrate neither complexity nor expensive computation is required for accurate segmentation on this task, allowing future focus to be placed on reducing prediction time.

Current / Future work

Recent effort has gone toward modifying the image capture program to collect information on material properties and machine configuration for each build. We hope to use this to produce analytics unique to different materials and enhance future machine learning on this problem. Our current focus is on applying tools provided by Nvidia's NGC TAO Toolkit and SambaNova's SambaFlow API (our motivation for using PyTorch) to draw performance comparisons.