Using Machine Learning to Analyze Image Data from Advanced Manufacturing Processes

A Capstone Report
presented to the faculty of the
School of Engineering and Applied Science
University of Virginia

by

Ryan French

with

Shubham Patel
James Mekavibul
Jami Park
Anchit Kolla
Zachary Kersey
Gregory C. Lewin

May 2, 2019

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Signed: ____

Approved: ____ Date ________________________
Gregory C. Lewin, Department of Engineering Systems and Environment

***Abstract* - Additive manufacturing (AM) - also known as 3D printing - promises a new approach to creating parts in a manufacturing environment; the process allows more design freedom and the production of parts with more complex features, compared to traditional manufacturing processes. The laser powder bed fusion (L-PBF) printer operates by building a part layer by layer in an iterative process of spreading metal powder and melting the desired shape. One particular feature is an overhang (material being melted onto the part over loose un-melted parts). However, some of the un-melted powder from the process could become melted to the overhanging feature - which is known as dross. Overhangs tend to form dross, but the extent of dross created at these features is not fully understood. Due to this unpredictable nature of dross formation, the build process exhibits variability in build quality, deterring industry-wide adoption.**

The conducted research aims to develop a system that analyzes cross-sectional image data captured from each layer of the print in order to identify dross with a certain level of confidence. Using machine learning techniques, images are used in a model that identifies pixels as a region that contains dross. These images are first labeled with bounding boxes (a coordinate system that identifies features/objects as existing within its boundaries) to train a neural network. The result is an adaptive model that autonomously detects dross in image scans of the part, pointing out these impurities to the printers' users, especially in regions difficult to inspect like interior surfaces of parts. The model aims to further understand L-PBF processing by location regions of excessive dross to relate dross formation with specific design features.

Index terms - 3D printing, additive manufacturing, neural networks, quality assurance

Introduction

Additive manufacturing (AM) creates a physical object from a digital model by repeatedly adding layers of material. By using a computer-aided design (CAD) model and a variety of materials (e.g. plastic, metal, tissue, etc.), manufacturers have much more freedom in their ability to design parts, forgoing constraints found in traditional machining processes. AM has a competitive advantage in reducing time-to-market, increasing product quality, improving product performance, and reducing product manufacturing costs (Donmez et al., 2014). Integral to the continued success of AM is the ability to achieve predictable and repeatable builds.

The team is working with the Commonwealth Center for Advanced Manufacturing (CCAM) to improve the quality assurance of their builds. Specifically, we present a method for identifying dross, unwanted build-up of material that affects the quality of the finished part. CCAM's EOS M290 printer uses laser powder bed fusion (L-PBF), which operates using an iterative process of spreading metal powder and melting the desired shape into the powder. Through this process, the L-PBF printer can create complex 3D prints made of metal. However, an overhang in the part causes unmelted powder from previous layers to melt onto the current overhanging feature - this buildup is called dross. Though overhangs are often linked to having dross, the extent of dross buildup is unclear and unpredictable, which makes the quality of the part variable.

Ultimately, the project goal in automated quality assurance of AM builds is to implement a real time feedback loop in printers. Such a system would identify faulty prints immediately and adjust print settings or terminate the print, saving time and resources. Creating a computational system that performs dross detection from images is the first step towards this end goal.

Previous research into utilizing machine learning algorithms to improve manufacturing processes has included the application of these algorithms and models to AM processes. Previous research on anomaly detection in AM processes used computer vision algorithms to classify anomalies in prints by comparing real-time pictures and model data [1]. However, their model uses hand crafted filters chosen by researchers to determine differences between part and model; making it difficult to adapt it to different impurities. As an alternative, convolutional neural networks (CNN) are the proposed model to classify and detect dross, since CNN learning is surprisingly rapid and test error rates are low [2]. The image data collected from the build plate is used in the creation of the CNN model.

Two previous capstone teams from UVA worked with CCAM to develop processes and tools for quality assurance. The first team focused on building a non-destructive post-production verification process that render 3D models from build plate images. The second team built a suite of reference tools for parsing, visualizing, and analyzing build-time sensor data.

The conducted research furthers the work of previous capstone teams by identifying dross in cross-sectional image data through an image recognition model. The model allows for detection of dross within the manufactured part without physically destroying the build. The model searches through cross-sectional images for dross on inside edges and overhangs. This paper outlines the development of such an image recognition model.

Background & Problem Definition

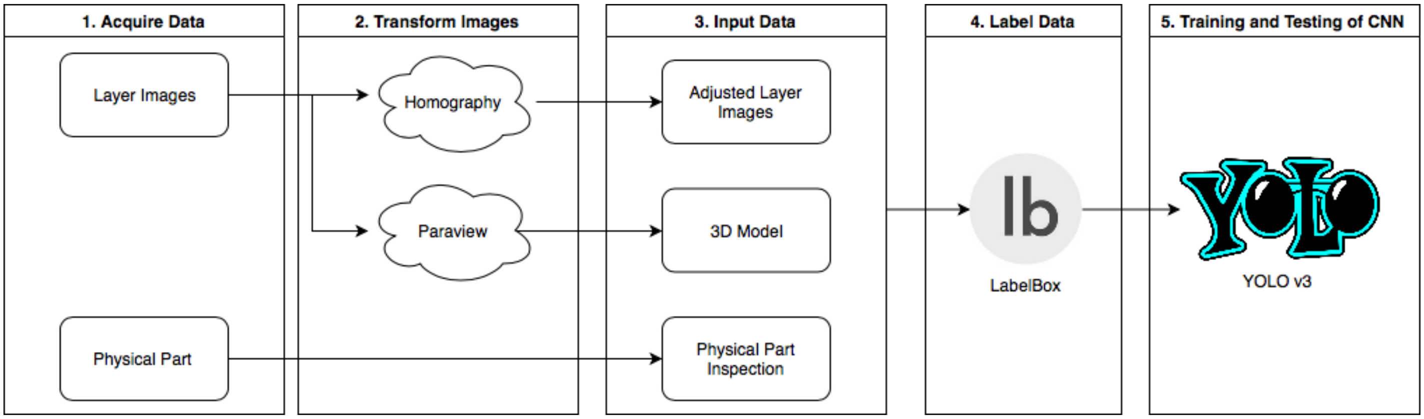
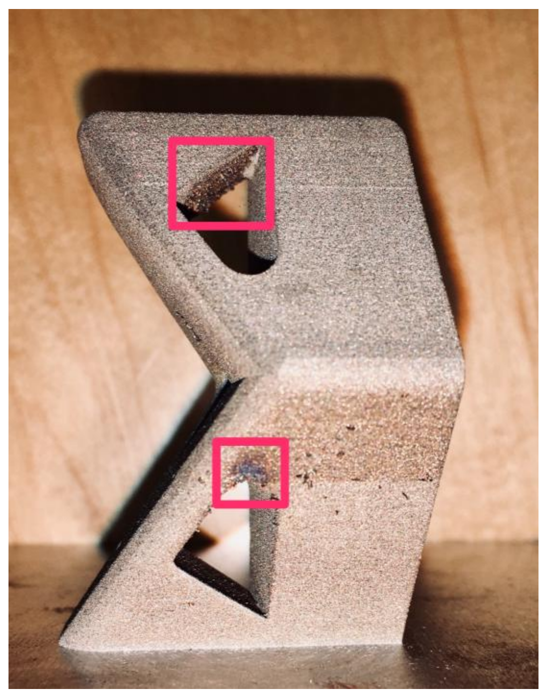


Figure 1. Overview of entire methodology for training our model.

Dross, the imperfection studied here, is unwanted, excess metal powder that remains on the build plate and gets melted into later cross-sectional layers, particularly at overhangs. These impurities can be seen as rough edges or excess material on the outsides of the print, as seen in Figure 2. The formation of dross increases the variability in prints and such inconsistency is a deterrent for industry-wide adoption [3]The identification of dross can improve the part design process; since dross is often linked with design flaws, build planners can utilize the identified dross to identify the features that lead to faulty builds. Currently, the only method of identifying dross at CCAM is through human detection, by way of cutting parts open or examining the exterior, but algorithms and models can circumvent human error and increase printing efficiency.

Embedded in the EOS M290 printer is a camera that takes pictures of every printed layer, in order to reconstruct a virtual model of the part for post-production analysis, including the identification of dross. A computational model can be used to identify features in these images. A feature (also known as an object) is an item of interest in the image. When a computer receives an image, it receives an array of pixel values. By identifying the image pixel values as features, a machine learning model can begin to create a pattern and filter other images for similar values. This is the essence of the computer vision discipline of machine learning.

Though there are multiple techniques that one can use to implement computer vision, we chose to use a Convolutional Neural Network (CNN). CNNs utilize classification data in order to identify different portions of an image. By training a CNN with pre-labeled images (labeling being identifying portions in the image as a specific entity, such as in Figure 3), the CNN can utilize complex mathematics to create a filter. When unlabeled images are input into the CNN, the CNN will use the filter created during training to determine if any of the pixel values seen in the new, unlabeled images, are similar to the patterns that the CNN has established. If so, the CNN will mark that section of the image as the appropriate feature.



Examples of Dross on the Physical 3D Printed Part

Methods

The first step of our process was acquiring the data to train the network on. This can be seen in the first step of Figure 1. In order to generate data to train the network on, CCAM developed and printed a test part with an increased likelihood of dross formation, and took images throughout the entire print process. The part, seen in Figure 2, has multiple overhangs which are known to be prone to dross formation. By labelling what dross looks like in these images, the CNN can be trained to accurately and precisely identify dross in other images.

In order to properly label the data, we first had to do some preprocessing. The steps we took can be summarized in Step 2 of Figure 1. The image data provided by CCAM was taken from a non-normal point of view above the print. In order to adjust the images to a top-down view, transformations through homography were implemented. Homography is a mathematical technique used to take a coordinate system and project those coordinates to a different perspective. To make the labeling process easier, the image data was first visualized as a 3D surface in Paraview, a data visualization software, in order to determine manually where the dross was located on the test piece. Additionally, the physical print was analyzed to determine major locations of dross on the exterior surfaces.

In order to create a model to predict dross using the image data, it needs to first be labeled. The steps we took here can be summarized in Step 3 & 4 in Figure 1. The ideal method of labeling is through bounding box segmentation around the areas that include dross. Bounding box segmentation makes use of the array that the pixels of image data are organized in. The bounding box is effectively a coordinate mapping that claims everything inside of the corners of the box falls under the associated classification. By taking the adjusted layer images, and comparing them to the physical part and 3D model generated in Paraview, we were able to label dross within the data.

Labeling the dross on the print was managed in LabelBox, an online tool used to quickly label each image and ensure internal agreement for each label [4]. This labeling was done manually by the members of the team based on comparisons with dross on the physical part and where key features of dross were located in the layer images. In Figure 3, an example of dross on an edge can be seen: within the bounding box, the lighter section is the part, and the bumps along the edge of that part are dross. LabelBox has the capacity to allow for multiple team members to label the same image for quality assurance and consistency. The multi-labeled images had their labels consolidated by using a script to determine the intersection of the label boxes across team members. Only the space with overlapping bounding boxes were determined to accurately identify dross, eliminating inconsistency in labeling. By looking at overlapping spaces, labels that might accurately identify dross could be lost if multiple team members did not identify the same regions. However, the decision to take only the intersection is based on an understanding that identifying dross is difficult to even the human eye. As a result, false positives from the human labeler would translate to the CNN, propagating further false positives.

The last step of this process is feeding the data through a CNN to have it learn how to properly label dross, as seen in step 5 of Figure 1. Though many implementations of image detection through CNNs exist, the YOLO v3 (You Only Look Once) object detection system was chosen due to its unique approach to object detection [5]. Rather than repurposing classifiers/localizers, YOLO divides the given image into regions and generates bounding boxes for each individual region, doing classification and localization in a single step (hence the name, You Only Look Once). A generic example of YOLO can be seen in Figure 4: the image of the bike and dog is split into gradually smaller regions, examined for the desired features (in this case, the dog and bike), and then compiled into boxes classifying the detected objects. Additionally, YOLO's fast classification times mean that it is ideal for real time object detection, making it potentially useful for real-time print analysis. For this project, YOLO takes in a 2000x2000 px image, resizes it to 416x416, and splits the image into 16 equivalent sections. Each section outputs its own potential bounding boxes for a total of 16 potential boxes. The bounding boxes with low confidence are removed, leaving the final labelled bounding boxes.

For this project, utilizing YOLO's methodology allows the network to save time and processing power by narrowing its scope to search specific areas for dross instead of running a general classification of the entire image. The goal is to create a system that detects and predicts the formation of dross. A successful system would consist of a model that can predict dross formation with at least 70% accuracy, a threshold that allows for generalization in other systems by avoiding overfitting. Overfitting occurs when a model is fit too closely to the training data that the model fails to generalize to the new testing data and can only make decisions on the same type of data used to train it. The threshold, therefore, allows for the system to be used for new parts. Overfitted models that perform well on dross detection on a specific part will be ineffective detecting dross on other parts, since dross might look different in images of a different par

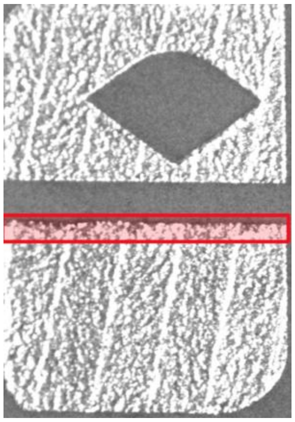


Figure 3. Dross Manually Labeled in Red Bounding Box

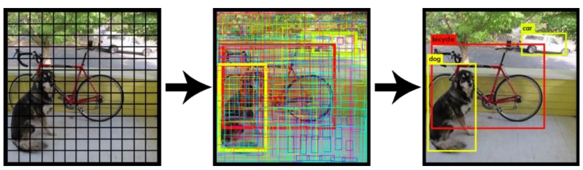


Figure 4. Illustration of the YOLO Model

	Dross	No dross/Poor picture	Total2
Training	80	116	196
Test	41	64	105
Total	121	181	301

Table 1. Dataset of Labeled Dross Images

In total, the labeled dataset consisted of 301 images: 121 images with dross, and 181 images without any dross. 70% images were randomly chosen to be training data, and 30% to be testing data. The data was split into 2 groups: roughly two-thirds of our labeled data was used to train the model and help it learn how to properly identify dross, and the remainder was used to test the model to verify that it could correctly identify dross. A detailed breakdown of images is detailed in Table 1.

Results

The YOLO network was trained for 2000 iterations on the dross and nondross images. Each iteration, the network changes its weights to improve the output to better approximate the input data. The network was trained with a batch size of 64, momentum of 0.9, and decay of 0.0005. Batch size refers to the number of samples that would be propagated through the network through each iteration and momentum prevents the network from getting stuck in a local minima. Decay is the value the determines how much outlier weights in the network are penalized to 0, allowing for better generalization. During the training of the network, the model will use stochastic gradient descent until the error converges to a local or global minimum. The process took 11 hours using 1 NVIDIA Tesla K80 Graphical Processing Unit (GPU).

The results were measured with the mean average precision (mAP) and recall. Recall measures the proportion of true positives for dross found out of the total images seen. mAP measures the area under the precision recall curve, where precision is the proportion of positives found out of all true positives in the dataset. With a confidence threshold of 0.50, the results from our training achieved a mean average precision (mAP) of 67.69% and a recall of 59%. The average time for our network to detect dross in an image is .088 seconds, meaning that it is also possible to utilize this network for real-time monitoring of prints for defects, since many printers take a longer time to build each layer.

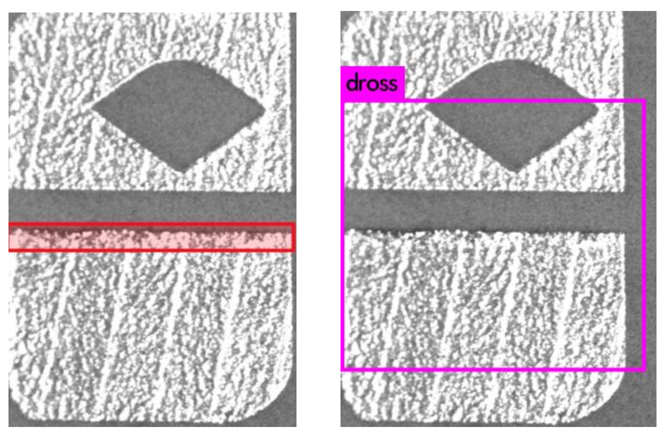


Figure 5. Manually Labelled Dross Image (Right) Compared To Model Labelled Image (Left)

Conclusion and Impact

For the part CCAM specifically manufactured for this project, the model can identify dross with some certainty. As seen in Figure 5, the output of the YOLO network is a bounding box that identifies a region that does indeed have dross in it (the dark, irregularly shaped line in the middle of the box). Based on the the numerical results from the model, the degree of certainty is below that which was originally desired (59% recall versus the desired 70%). However, the results suggest that, with better tuning, the method has the potential to reliably identify dross.

The size of the bounding boxes outputted by the model are a potential source of error, however. In Figure 5, the outputted bounding box (the image on the right) takes up over half of the layer image. Though there is dross in that image, the dross is actually only located in the space seen in the bounding box in the image on the left. By identifying the majority of the image as dross, the model has done a poor job of selecting the desired object and excluding all else. At the current time, The results neither confirm or deny that the network is capable of narrowing down the bounding box to only the dross, but it is being actively pursued. Ultimately, it would be ideal to remove all of the false positives existing within the bounding box.

The results of this project suggest that object detection can be implemented for other 3D printed parts. This project was, in many ways, a proof of concept to determine whether a CNN be trained to identify dross and, if so, with what certainty? Though the recall value of 59% is below the desired 70%, these results are promising. The model was not trained with additional images from different parts. However, the results lead us to believe that, at the very least, a model can be trained to identify dross in a singular, specific part; future work could implement training a model to work with multiple parts.

Future Work

Future work for this project includes improving the model developed from the training. The model utilized in this research was only trained with 301 images. However, if the model were given more pre-labeled data, the CNN would be able to further train on the new data and improve its ability to detect dross in the images. Furthermore, due to the difficult nature of labeling dross, there is room for human error in the manually labelled data. Since multiple people labelled data and only the intersection of the labels were taken, there could exist other examples of dross in the images that the model did not train on. By improving the labeling process through standardization of how to label dross, the labeled images would be more accurate and precise, giving the model an opportunity to train on "better" data. Similarly, training the model on another set of images from a different part than the one used in this research would help generalize the model. In its current stage, the model is apt at identifying dross in a very specific style of image; including labeled images from another part would aid the model from identifying dross in layer images from any part.

Though this research used YOLO as its preferred object detection network, other networks, such as Single-Shot Detection (SSD), or Regions with CNN features (R-CNN). Both of these networks. Additionally, other label formats, such as image masks or classification labels, could be explored which may simplify the creation of a well-trained object detection model. By exploring other frameworks, one could determine what CNN platform/framework is the most effective at detecting dross.

Acknowledgements

This project was supported by the Commonwealth Center for Advanced Manufacturing (CCAM). A special thanks to Kyle Snyder and Arthur French for assisting with the design, creation, and experimentation of the image recognition model, without which the project would not have been possible. A special thanks to Joseph Fitzsimmons and Justin McAuliff for assisting with the SIEDS paper and presentation.

Authors

Shubham Patel, Student, Department of Engineering Systems and Environment, University of Virginia.
James Mekavibul, Student, Department of Computer Science, University of Virginia.
Jami Park, Student, Department of Computer Science, University of Virginia
Anchit Kolla, Student, Department of Engineering Systems and Environment, University of Virginia
Ryan French, Student, Department of Engineering Systems and Environment, University of Virginia.
Zachary Kersey, Student, Department of Engineering Systems and Environment, University of Virginia.
Gregory C. Lewin, Assistant Professor, Department of Engineering Systems and Environment, University of Virginia

References

[1] Scime, Luke, and Jack Beuth. "Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm." *Additive Manufacturing* 19 (2018): 114-126.

[2] Ciresan, Dan Claudiu, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. "Flexible, high performance convolutional neural networks for image classification." In *Twenty-Second International Joint Conference on Artificial Intelligence*. 2011.

[3] Huang, Yong, Ming C. Leu, Jyoti Mazumder, and Alkan Donmez. "Additive manufacturing: current state, future potential, gaps and needs, and recommendations." *Journal of Manufacturing Science and Engineering* 137, no. 1 (2015): 014001.
[4] Labelbox. “The Best Way to Create and Manage Training Data.” *Labelbox*, 2019, labelbox.com.
[5] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788. 2016.