Machine Learning for Quality Prediction in Additive Manufacturing

Presentor: Matt Behnke Mentor: Dr. Guo

Introduction to Laser Based Additive Manufacturing (LBAM)

What is LBAM?

- Products are made when metallic powder is melted layer-by-layer by a laser until the product is complete
- A much more cost effective option over subtractive engineering

Applications of LBAM

- Can produce many different types of complicated parts or models
- Benefiting industries like
  - Aerospace
  - Bioengineering / Medical
  - Consumer Products

Problems is LBAM

- Defects due to Porosity
  - Prevented adoption on a large scale
- No reliable / cost-efficient way of detecting these porosity defects
  - No in-situ detection
  - Post-production detection is expensive

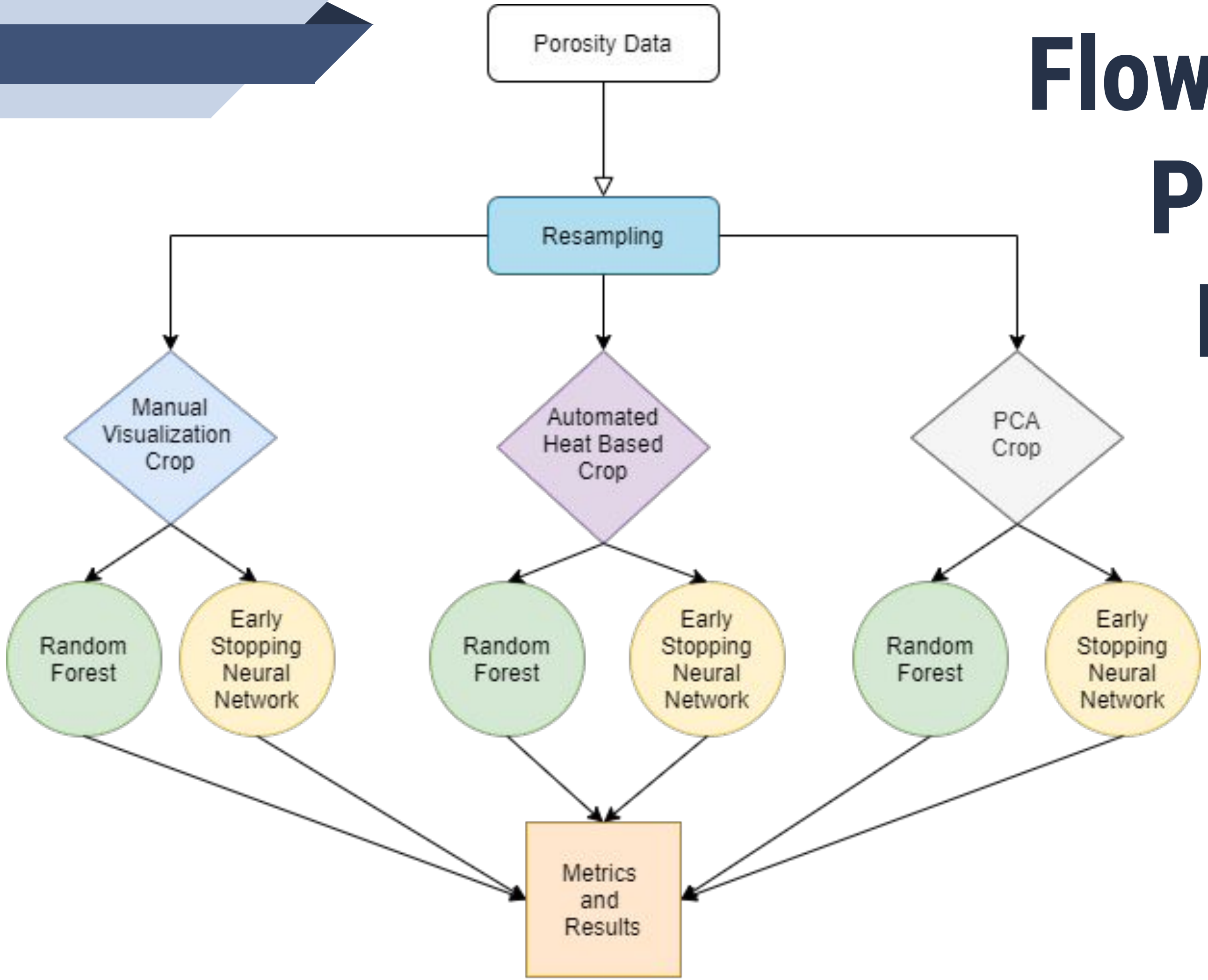
Purpose / Contribution of this Study

- Propose a consistent, accurate, and reliable way for *in-situ* for monitoring for porosity prediction in the LBAM Process

- Provide insight on which layers are more likely to produce porosity defects

- Provide methods to automatically detect melt pool in LBAM
    - Melt Pool is a signature trait of LBAM

Flowchart of the Prediction Process



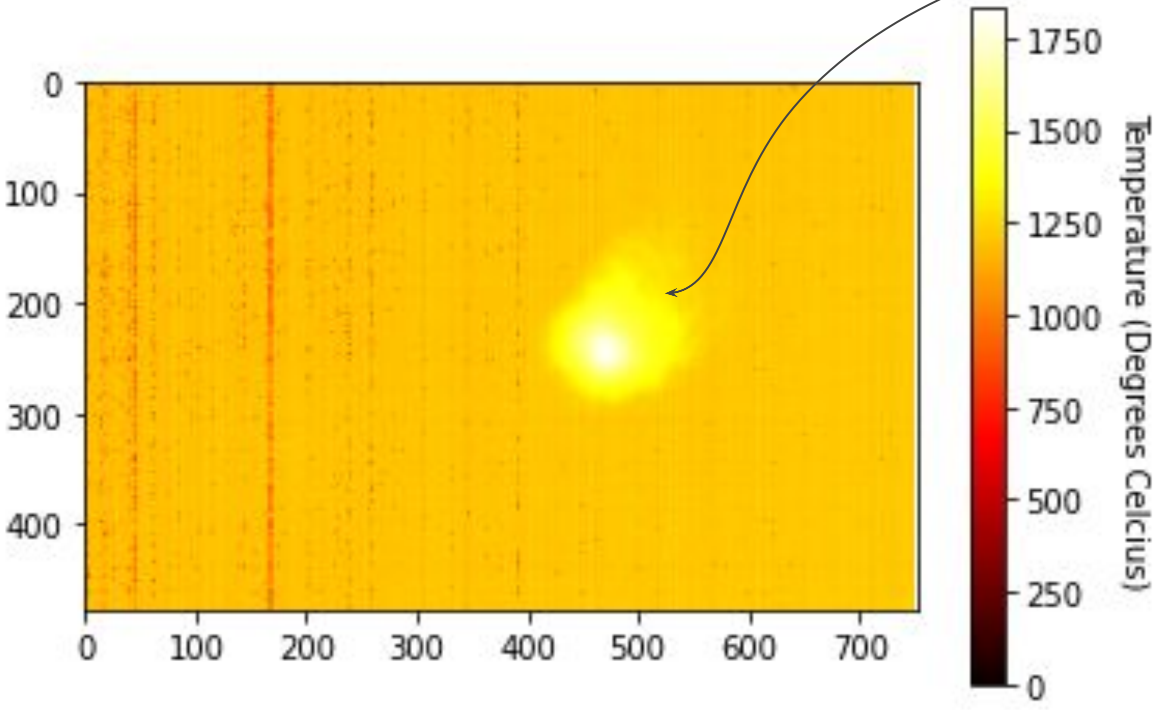
Porosity Data Set

- Data was collected from OPTOMEC LENS™750 LBAM system

- Data is described in *Data indicating temperature response of Ti−6Al−4V thin-walled structure during its additive manufacture via Laser Engineered Net Shaping* by Garrett J. Marshall, Scott M. Thompson, Nima Shamsaeia

- In total the data set consisted of 1556 csv files
  - Each data file is represented by a data matrix (479 Rows x 753 Columns)
  - Each data point in the matrix is a temperature value (°C) ranging from 0-1800
  - Each file is mapped to either a ‘0’ or a ‘1’ which represents a quality of good and bad respectively

Example Visualization of a Data File



- Example of the melt pool

Resampling the Data

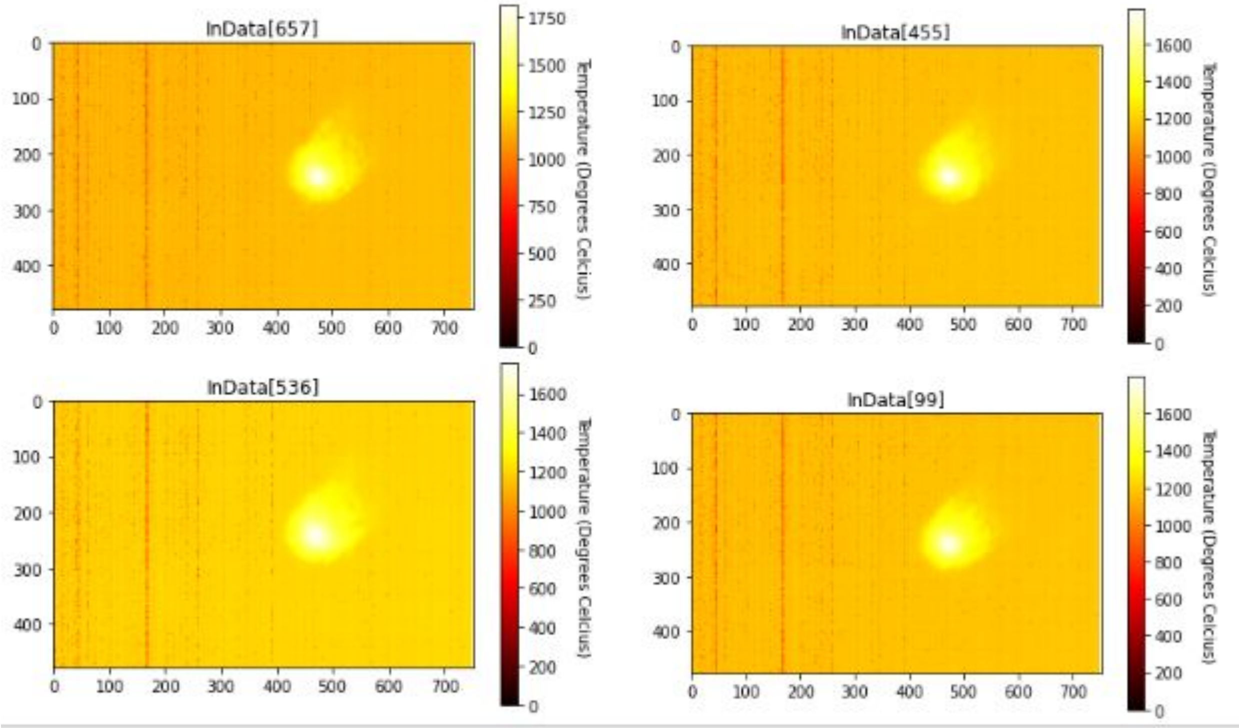
- The data set was heavily unbalanced
  - 1486 ‘Good’ or ‘0’ data files
  - 70 ‘Bad’ or ‘1’ data files
- This would be hard to train a model off of due ot the limited amount of ‘bad’ instances
- Bootstrap Resampling was used to increase the number of bad instances to 1486, which equaled the amount of good instances
- The final dataset that was used now contained 2972 data files, with an even split of ‘good’ and ‘bad’ instances

Why Crop the Datafiles?

- Reduce redundant information
  - Many data images had the same areas that had the exact same pattern, which ultimately won't contribute to a successful model

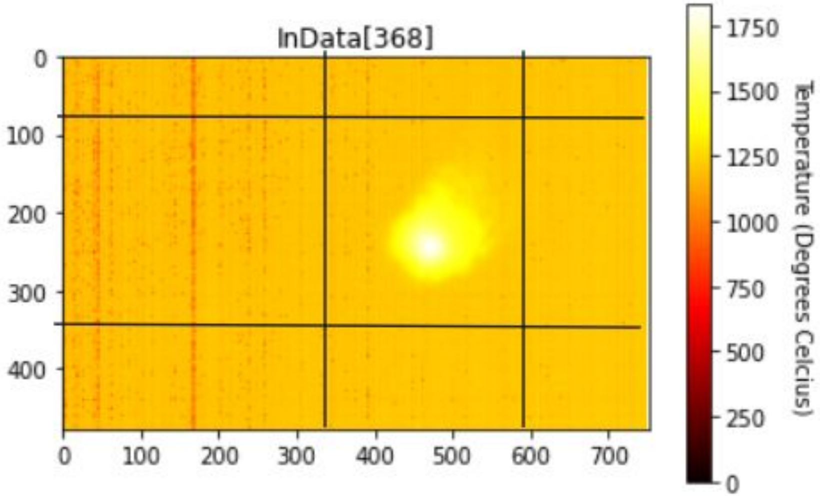
- Speed up Model Performance
  - Less data points means the model has less training, which results in faster speeds

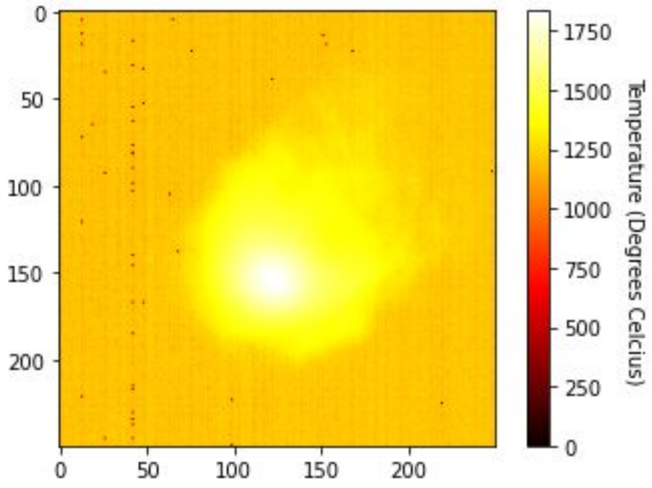
Example of Redundant Information



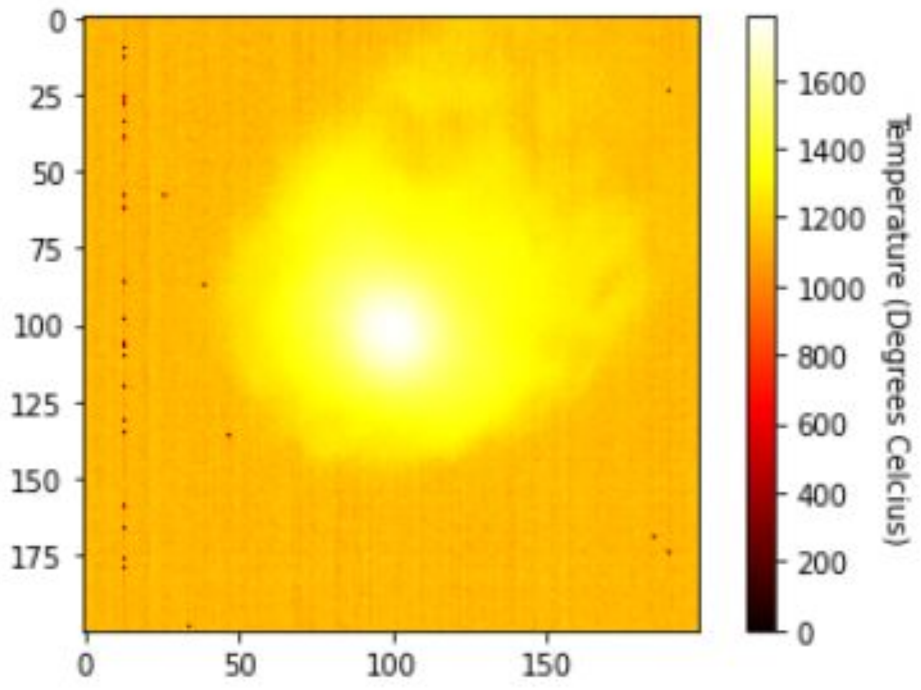
Visualization Based Crop

- By visualizing many random instances of data, a good manual based crop was determined to be $x \in [350,600]$, $y \in [90, 340]$, creating a new 250x250 matrix





Automated Heat Based Crop

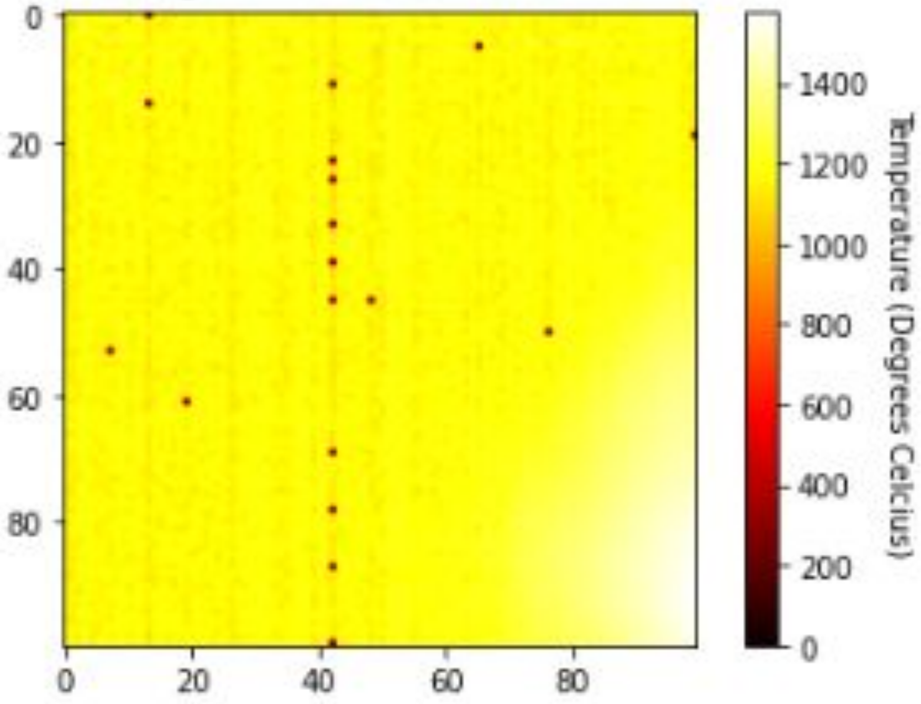


- Find hottest pixel from a linear search
  - If there is a tie, the average of their x & y coordinates are used
- Create a 200x200 box, around the hottest pixel found
- Benefit is it automatically finds the melt pool, regardless of the positioning

Principal Component Analysis (PCA) Crop

- PCA is a method to reduce the number of features in a dataset by extracting the most important features
  - These features being pixels
- Important to pick a number of components that yields a high level of variance, 40 components
  - 95% variance was chosen to be sufficient
- Each component has a list of the magnitude of each feature's (each pixel's) eigenvalue
  - The higher the magnitude the more 'importance'
- The mode for the x, y coordinates of these pixels were taken to be the center of the crop
  - If there was no mode

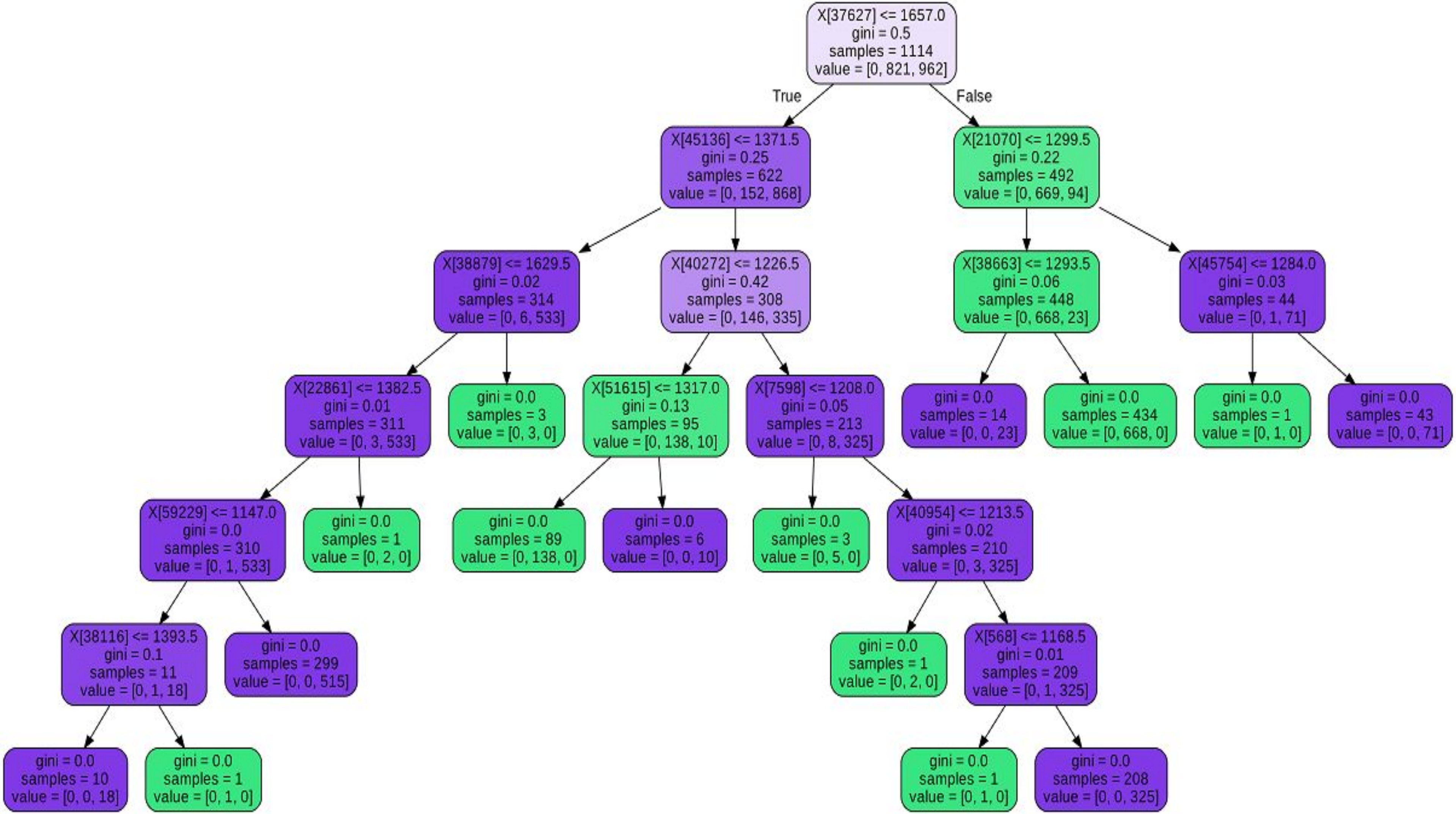
PCA Crop (cont.)



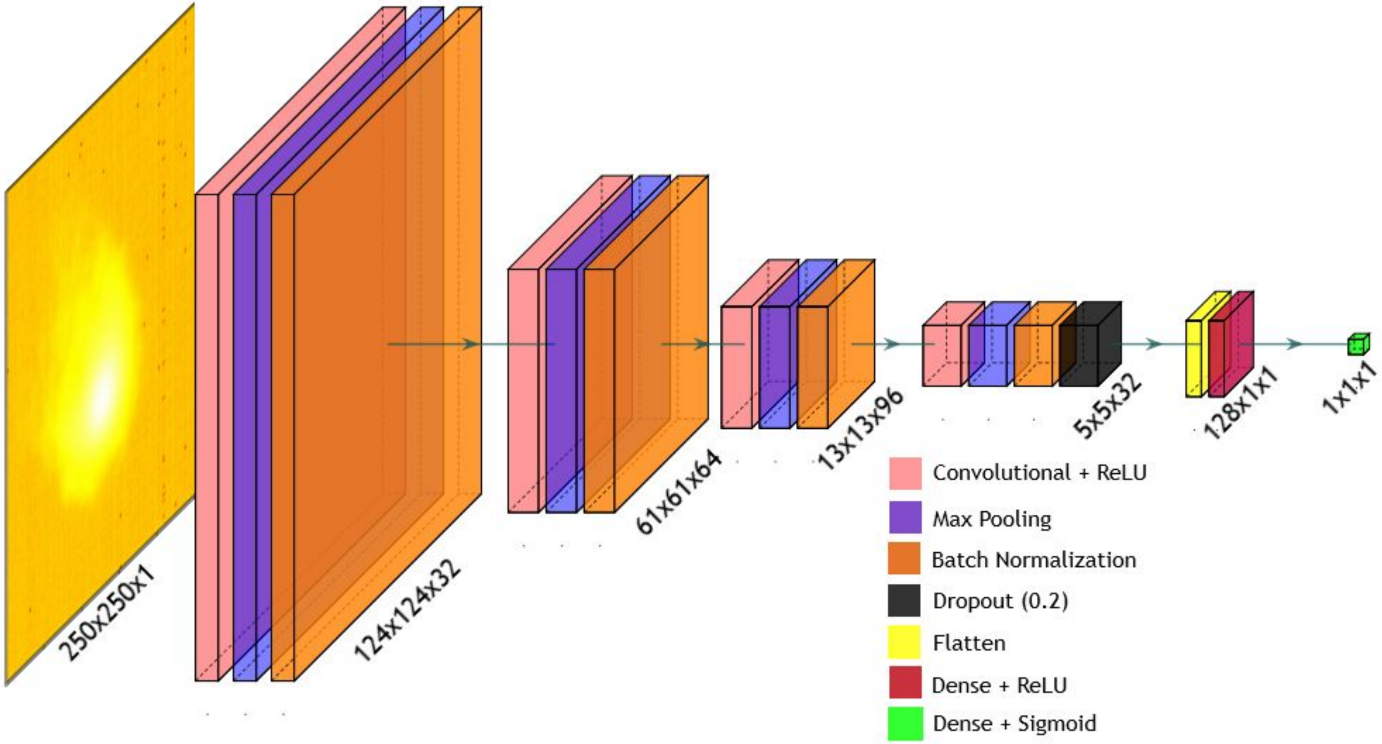
- 100x100 Grid
- From the dataset PCA determined that the x ∈ [300,400], y ∈ [148, 248] would be the new bounds with (350, 198) as the center

Random Forest Model

- Random Forest tends to be a good model in computer vision classifying tasks
  - Ex: Random Forest was used to classify human body part poses on the kinect for the xbox 360
- Data is vectorized before using the Random Forest Model
- Random Forest is built from an ensemble of decision trees
  - In our model, 10
- Using sklearn package


X[37627] <= 1657.0
gini = 0.5
samples = 1114
value = [0, 821, 962]
True
False
X[45136] <= 1371.5
gini = 0.25
samples = 622
value = [0, 152, 868]
X[21070] <= 1299.5
gini = 0.22
samples = 492
value = [0, 669, 94]
X[38879] <= 1629.5
gini = 0.02
samples = 314
value = [0, 6, 533]
X[40272] <= 1226.5
gini = 0.42
samples = 308
value = [0, 146, 335]
X[38663] <= 1293.5
gini = 0.06
samples = 448
value = [0, 668, 23]
X[45754] <= 1284.0
gini = 0.03
samples = 44
value = [0, 1, 71]
X[22861] <= 1382.5
gini = 0.01
samples = 311
value = [0, 3, 533]
gini = 0.0
samples = 3
value = [0, 3, 0]
X[51615] <= 1317.0
gini = 0.13
samples = 95
value = [0, 138, 10]
X[7598] <= 1208.0
gini = 0.05
samples = 213
value = [0, 8, 325]
gini = 0.0
samples = 14
value = [0, 0, 23]
gini = 0.0
samples = 434
value = [0, 668, 0]
gini = 0.0
samples = 1
value = [0, 1, 0]
gini = 0.0
samples = 43
value = [0, 0, 71]
X[59229] <= 1147.0
gini = 0.0
samples = 310
value = [0, 1, 533]
gini = 0.0
samples = 1
value = [0, 2, 0]
gini = 0.0
samples = 89
value = [0, 138, 0]
gini = 0.0
samples = 6
value = [0, 0, 10]
gini = 0.0
samples = 3
value = [0, 5, 0]
X[40954] <= 1213.5
gini = 0.02
samples = 210
value = [0, 3, 325]
X[38116] <= 1393.5
gini = 0.1
samples = 11
value = [0, 1, 18]
gini = 0.0
samples = 299
value = [0, 0, 515]
gini = 0.0
samples = 1
value = [0, 2, 0]
X[568] <= 1168.5
gini = 0.01
samples = 209
value = [0, 1, 325]
gini = 0.0
samples = 10
value = [0, 0, 18]
gini = 0.0
samples = 1
value = [0, 1, 0]
gini = 0.0
samples = 1
value = [0, 1, 0]
gini = 0.0
samples = 208
value = [0, 0, 325]

Early Stopping Neural Network

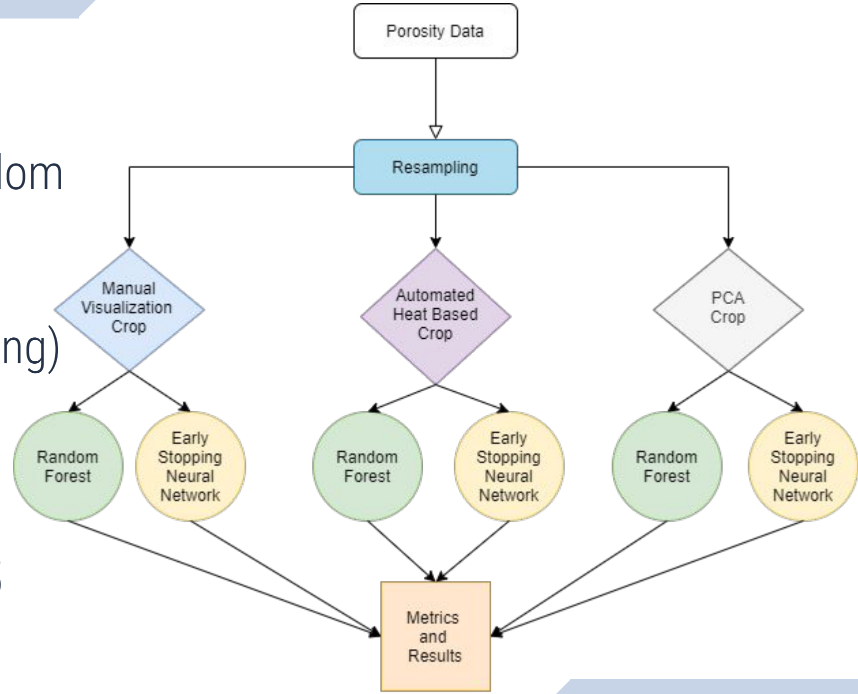


Early Stopping Neural Network (cont.)

- Throughout Training Validation Accuracy was highly related to the test accuracy on a random split
- If the validation accuracy was low on the last epoch of training the test accuracy scored at a similar level and vice-versa
- Wanted a way to ensure training stops on an epoch with a high, which was accomplish with an Early Stopping Callback Function
- The Number of Epochs were increased, to 50, and a Patience value was set
  - Patience refers to the number of epochs the model will train without an improved score in a specified metric, Validation Accuracy
  - If the patience value is reached, the model will revert back to the epoch with the highest score in the specified metric

Procedure for Obtaining Metrics

- For each cropping style (Manual, Heat-Based, PCA)
  - The cropped data was tested for both the Random Forest and Early Stopping Neural Network
  - Each set of 25 trials were don over a certain training / testing ratios (ex: 10/90 training/testing)
  - 25 trials per testing
    - Each trial had a random split
    - Testing accuracy and Time was recorded
  - Metrics were recorded as the average of the 25 trials



Results for every Cropping Style

Manual Crop

Training /Testing Split Ratio	Test Accuracy (%) Random Forest	Test Accuracy (%) Early Stopping Neural Network	Time (sec) Random Forest	Time (sec) Early Stopping Neural Network
10/90	98.45	59.83	2.49	12.02
20/80	99.26	94.66	4.30	31.58
30/70	99.67	98.19	6.18	37.02
40/60	99.68	99.18	7.82	43.39
50/50	99.57	99.00	9.56	50.98
60/40	99.82	99.69	11.37	60.78
70/30	99.79	99.50	13.07	72.20
80/20	99.75	99.90	14.66	75.84

Heat-Based

Training /Testing Split Ratio	Test Accuracy (%) Random Forest	Test Accuracy (%) Early Stopping Neural Network	Time (sec) Random Forest	Time (sec) Early Stopping Neural Network
10/90	96.48	66.23	2.20	10.12
20/80	99.14	98.56	3.95	21.89
30/70	99.58	98.08	5.71	29.14
40/60	99.84	99.01	12.17	31.69
50/50	99.46	96.29	10.09	15.21
60/40	99.64	99.49	12.00	22.18
70/30	99.78	99.72	13.80	24.05
80/20	99.90	99.53	15.55	25.63

PCA

Training /Testing Split Ratio	Test Accuracy (%) Random Forest	Test Accuracy (%) Early Stopping Neural Network	Time (sec) Random Forest	Time (sec) Early Stopping Neural Network
10/90	95.46	49.63	0.80	4.13
20/80	97.88	62.00	1.57	6.86
30/70	98.80	93.64	2.25	11.98
40/60	99.04	99.43	2.95	23.99
50/50	99.46	98.25	3.53	34.46
60/40	99.64	99.70	4.19	43.71
70/30	99.78	99.81	4.75	48.02
80/20	99.91	99.13	5.46	47.79

Misclassified Data

Over 25 trials with the Early Stopping Neural Network at the random 50/50 split

- 75 different files out of the original 1556 were misidentified
- ‘0’ status refers to a positive, while 1 refers to ‘negative’, so when they are incorrectly identified as so it refers to ‘false positive’ and ‘false negative’
    - 42 False Positives
    - 33 False Negatives
- By counting files more than once (predicted incorrectly over multiple trials)
    - 79 False Positives
    - 539 False Negatives
- 38 different layers were misidentified
    - Out of the 75 different files, Layer 1,2 & 3 constituted of 29/75
        - 12 Layer-2 Files were misidentified
        - 10 Layer 1 files were misidentified
        - 7 Layer-3 files were misidentified
        - All other files had 2 or less instances

Conclusion

- Models provided very consistent and accurate results that could be used for in-situ monitoring during the LBAM process

- Models performed well over different types of cropping methods, showing their adaptability
    - Preferring Heat-Based crop

- Future work could include trying to find out where in the image the model predicts is causing porosity defects

Acknowledgements

- Huge Thanks to Dr. (Grace) Guo for her mentorship throughout the summer

- Thank you to NSF grant CCF-1852215 and the Rutgers DIMACS Team for the opportunity to work on the project

References

1. Marshall, Garrett & Thompson, Scott & Shamsaei, Nima. (2016). Data indicating temperature response of Ti-6Al-4V thin-walled structure during its additive manufacture via Laser Engineered Net Shaping. Data in Brief. 7. 10.1016/j.dib.2016.02.084.
2. Khanzadeh, Mojtaba & Chowdhury, Sudipta & Marufuzzaman, Mohammad & Tschopp, Mark & Bian, Linkan. (2018). Porosity prediction: Supervised-learning of thermal history for direct laser deposition. Journal of Manufacturing Systems. 47. 69-82. 10.1016/j.jmsy.2018.04.001.
3. Guo, Weihong & Tian, Qi & Guo, Shenghan & Guo, Yuebin. (2020). A physics-driven deep learning model for process-porosity causal relationship and porosity prediction with interpretability in laser metal deposition. CIRP Annals. 10.1016/j.cirp.2020.04.049.
4. Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. Return of the devil in the details: Delving deep into convolutional nets. In Proc. BMVC., 2014.
5. He, K., Zhang, X., Ren, S., and Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. CoRR, abs/1406.4729v2, 2014.
6. Wu, Ren, Yan, Shengen, Shan, Yi, Dang, Qingqing, and Sun, Gang. Deep image: Scaling up image recognition, 2015.
7. Seifi, S.H., W. Tian, H. Doude, M.A. Tschopp, and L. Bian, Layer-wise modeling and anomaly detection for laser-based additive manufacturing.Journal of Manufacturing Science and Engineering, 2019.
8. Oquab, M., Bottou, L., Laptev, I., and Sivic, J. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In Proc. CVPR, 2014.