



Improving the Performance of Deep Learning Techniques using Nature Inspired Algorithms and Applying them in Porosity Prediction

**A thesis submitted to Cardiff University in the candidature for the degree of
Doctor of Philosophy (Engineering)**

Name: Nawaf Mohammad H Alamri

ID #:1881324

Supervisors: Dr. Michael Packianather

Dr. Samuel Bigot

School of Engineering

Cardiff University

Wales, United Kingdom

March 2023

Abstract

Within the field of Artificial Intelligence (AI), Deep Learning (DL) based on Convolutional Neural Network (CNN) can be used for analysing images. However, the performance of the DL models depends on the design of the CNN topology to achieve their best performance. Hence, firstly, this work addresses this problem by proposing a novel nature inspired hybrid algorithm called BA-CNN where a swarm based Bees Algorithm (BA) is used to optimize the CNN parameters. In addition, another algorithm called BA-BO-CNN is proposed that combines the BA with Bayesian Optimization (BO) to increase the CNN performance and that of BA-CNN and BO-CNN. This study shows that applying the hybrid BA-CNN to the ‘Cifar10DataDir’ benchmark image did not improve the validation and testing accuracy compared to the existing CNN and BO-CNN. However, the hybrid BA-BO-CNN achieved better validation accuracy of 82.22% compared to 80.34% and 80.72% for the CNN and BO-CNN, and also with a better testing accuracy of 80.74% compared to 80.54% and 80.69% for the CNN and BO-CNN respectively. The BA-BO-CNN achieved lower computational time than the BO-CNN algorithm by 2 minutes and 11 seconds. Although applying both algorithms to the ‘digits’ dataset produced almost similar accuracies with a difference of 0.01% between BA-CNN and BO-CNN, the BA-CNN achieved a computational time reduction of 4 minutes and 14 seconds compared to the BO-CNN, making it the best algorithm in terms of cost-effectiveness. Applying BA-CNN and BA-BO-CNN to identify ‘concrete cracks’ images produced almost similar results to some of the other existing algorithms with a difference of 0.02% between BA-CNN and original CNN. Finally, applying them to the ‘ECG’ images improved the testing accuracy from 90% for the BO-CNN to 92.50% for the BA-CNN and 95% for the BA-BO-CNN with a similar trend for validation accuracy and computational time.

Secondly, the CNN that was adopted for the purpose of regression which is called RCNN was applied in the manufacturing context, particularly to predict the percent of porosity in the finished Selective Laser Melting (SLM) parts. Because testing the performance of the RCNN algorithm requires a large amount of experimental data which is generally difficult to obtain, in this study an artificial porosity image creation method is proposed where 3000 artificial porosity images were created mimicking real CT scan slices of the SLM part with a similarity index of 0.9976. Applying the RCNN to the 3000 artificial

porosity images slices showed the porosity prediction accuracy to improve from 68.60% for the image binarization method to 75.50% for the RCNN, while the proposed novel hybrid BA-BO-RCNN and BA-RCNN yielded better prediction accuracies of 83% and 85.33% respectively.

Thirdly, in order to improve the performance even further, this study proposes to add Long Short Term Memory (LSTM) to BA-CNN because of their ability to deal with sequential data to produce another novel hybrid algorithm called BA-CNN-LSTM and the results showed an increase in the prediction accuracy reaching 95.50%.

Acknowledgement

I would like to offer my thanks and genuine appreciation to my supervisors Dr. Michael Packianather and Dr. Samuel Bigot for their professional guidance, invaluable advice and unlimited support in completing this dissertation.

In addition, I would like to thank Cardiff University and the Saudi Arabian Cultural Bureau for their unlimited support. Also, I would like to thank the H2020 project partners of Additive Manufacturing using Metal Pilot Line (MANUELA) project (grant agreement No. 820774) for their collaboration.

Finally, I would like to express my deepest gratitude to my family and friends for giving me the composure and the strength to complete this work.

Table of Contents

Abstract.....	i
Acknowledgement.....	iii
List of Figures.....	ix
List of Tables	xiii
List of Abbreviations	xv
List of Symbols	xix
List of Publications	xx
Chapter 1: Introduction	1
1.1 Background.....	2
1.2 Aim and Objectives	5
1.3 Research Questions	6
1.4 Thesis Outline	6
1.5 Study Limitations and Assumptions.....	9
Chapter 2: Literature Review.....	10
2.1 Deep Learning	11
2.1.1 <i>History and Definition</i>	11
2.1.2 <i>Advantages, Limitations, and Applications</i>	12
2.2 Convolutional Neural Network	17
2.2.1 <i>Definition and Way of Working</i>	17
2.2.2 <i>Gaps and Open Issues</i>	19
2.3 Long Short-Term Memory Network.....	21
2.3.1 <i>Definition and Way of Working</i>	21
2.3.2 <i>Gaps and Open Issues</i>	22

2.4 Impact and Recent Applications of Optimizing Deep Learning Parameters using Nature Inspired Algorithms.....	23
<i>2.4.1 State of the Art Studies</i>	<i>23</i>
<i>2.4.2 Gap and Open Issue.....</i>	<i>27</i>
2.5 Machine Learning in Manufacturing.....	27
2.6 Additive Manufacturing Overview.....	29
2.7 Additive Manufacturing Processes	31
2.8 Powder Bed Fusion.....	32
<i>2.8.1 Definition and Way of Working</i>	<i>32</i>
<i>2.8.2 Thermodynamical Phenomena</i>	<i>34</i>
<i>2.8.3 Parameters</i>	<i>36</i>
<i>2.8.4 Open Issues</i>	<i>36</i>
<i>2.8.5 Gas Porosity</i>	<i>37</i>
<i>2.8.6 Keyhole Porosity</i>	<i>38</i>
<i>2.8.7 Lack of Fusion Porosity</i>	<i>39</i>
<i>2.8.8 State of the Art Studies</i>	<i>40</i>
2.9 Smart Manufacturing Technologies.....	42
2.10 Summary	43
Chapter 3: Convolutional Neural Network Parameters Optimization using Nature Inspired Algorithms.....	44
3.1 Proposed Novel Hybrid BA-CNN & BA-BO-CNN Algorithms.....	45
<i>3.1.1 Design of Experiments</i>	<i>45</i>
<i>3.1.2 Bayesian Optimization Method</i>	<i>47</i>
<i>3.1.3 The Bees Algorithm</i>	<i>48</i>
<i>3.1.4 Convolutional Neural Network Architecture</i>	<i>52</i>
<i>3.1.5 The Existing BO-CNN Algorithm.....</i>	<i>53</i>

3.1.6 <i>The Proposed Novel BA-CNN Algorithm</i>	55
3.1.7 <i>The Proposed Novel BA-BO-CNN Algorithm</i>	57
3.2 Results and Discussion of the Proposed BA-CNN & BA-BO-CNN Algorithms	60
3.3 Summary	70
Chapter 4: Artificial Porosity Images Creation for Selective Laser Melting Parts...	72
4.1 Artificial Porosity Images Creation Process	73
4.1.1 <i>Establishing Regression Equations</i>	75
4.1.2 <i>Generating Pores Number and Diameter</i>	78
4.1.3 <i>Creating 3D Cubes</i>	79
4.1.4 <i>Slicing 3D Cubes into 2D Images</i>	82
4.1.5 <i>Labelling 2D Slices</i>	83
4.1.6 <i>Adding Noisy Background</i>	83
4.2 Results and Discussion of the Similarity between Artificial and Real Porosity Images	87
4.3 Summary	90
Chapter 5: Hybrid Regression Convolutional Neural Network for Predicting the Percent of Porosity in Selective Laser Melting Parts	92
5.1 Predicting the Porosity using the Existing Image Binarization Method	93
5.1.1 <i>Porosity Assessment Gap using the Image Binarization Method</i>	93
5.1.2 <i>Porosity Assessment Gap Validation</i>	94
5.2 Predicting the Porosity using Hybrid Regression Convolutional Neural Networks	99
5.2.1 <i>Regression Convolutional Neural Network Architecture</i>	99
5.2.2 <i>The Proposed Hybrid BA-BO-RCNN & BA-RCNN Algorithms</i>	101

5.3 Results and Discussion for Hybrid Regression Convolutional Neural Networks	104
.....	
5.3.1 Results of Applying the Proposed Hybrid BA-BO-RCNN & BA-RCNN Algorithms to the Second Version of Artificial Porosity Images	104
5.3.2 Results of Applying the Proposed Hybrid BA-BO-RCNN & BA-RCNN Algorithms to the Final Version of Artificial Porosity Images	106
5.3.3 Sensitivity to Noise in Artificial Porosity Images	108
5.4 Summary	112
Chapter 6: Hybrid Long Short-Term Memory Network with Bees Algorithm for Enhancing the Porosity Prediction in Selective Laser Melting Parts	114
.....	
6.1 Proposed Novel Hybrid BA-CNN-LSTM Algorithm	115
6.1.1 Convolutional Neural Network Architecture	116
6.1.2 Long Short-Term Memory Architecture	117
6.1.3 The Novel Hybrid BA-CNN-LSTM Architecture Algorithm	121
6.2 Results and Discussion	124
6.2.1 Results of Applying BA-CNN-LSTM on Artificial Porosity Images	125
6.2.2 Results of Applying BA-CNN-LSTM on Electrocardiogram (ECG) Dataset	130
6.2.3 Results of Applying BA-CNN-LSTM on Turbofan Engine Degradation Simulation Dataset	135
6.3 Summary	137
Chapter 7: Conclusion	140
.....	
7.1 Conclusion	141
7.2 Contributions to Knowledge	144
7.3 Study Limitations	145
7.4 Future Work	146

References	147
Appendix 1: MATLAB Codes.....	165
1.1 Convolutional Neural Network (CNN).....	166
1.2 Bayesian Convolutional Neural Network (BO-CNN)	169
1.3 Bees Convolutional Neural Network (BA-CNN)	173
1.4 Bees Bayesian Convolutional Neural Network (BA-BO-CNN).....	181
1.5 Artificial Porosity Images Creation and Labelling	188
1.6 Actual Percent of Porosity Dataset	195
1.7 Regression Convolutional Neural Network (RCNN)	220
1.8 Bayesian Regression Convolutional Neural Network (BO-RCNN).....	231
1.9 Bees Bayesian Regression Convolutional Neural Network (BA-BO-RCNN) .	243
1.10 Bees Regression Convolutional Neural Network (BA-RCNN)	260
Appendix 2: Curriculum Vitae	277

List of Figures

Figure	Page
Figure 2.1: Inputs and Hidden Layers Connection in CNN (Le, 2015)	18
Figure 2.2: The Way of Working for CNN (MathWorks-1).....	19
Figure 2.3: The Basic Structure for LSTM (www.analyticsvidhya.com)	22
Figure 2.4: AM Technologies Classification According to ISO/ASTM 52900 (Lastra et al.,2022)	32
Figure 2.5: Elements of Powder Bed Fusion Process (Sun et al., 2017)	33
Figure 2.6: Ishikawa Diagram of Influential Factors in the SLM Process (Bauer, 2021)	34
Figure 2.7: Top Surface Morphology with Laser Power of 90 W (Gu et al., 2017) ...	35
Figure 2.8: High Magnitude Microstructure Morphology of Top Surface with Laser Power of 120 W (Gu et al., 2017)	35
Figure 2.9: Illustrative Example of Gas Porosity using SEM (Tan et al., 2020)	37
Figure 2.10: 3D View of Keyhole Pores (Shrestha et al., 2019)	38
Figure 2.11: 2D Images of Different Keyhole Pores Morphologies (Shrestha et al., 2019)	39
Figure 2.12: Lack of Fusion Pores Morphology (Tan et al., 2020).....	40
Figure 3.1: Regression Equation	47
Figure 3.2: The Basic Flowchart for the BA (Koc, 2010)	50
Figure 3.3: An Illustrative Diagram for BA-CNN Algorithm	54
Figure 3.4: An Illustrative Diagram for BA-BO-CNN Algorithm	55
Figure 3.5: The Steps of the MATLAB Code for BA-CNN	56
Figure 3.6: The Steps of the MATLAB Code for BA-BO-CNN	58
Figure 3.7: Workflow Diagram for the Hybrid BA-BO-CNN Algorithm	59
Figure 3.8: Training Progress for BA-CNN	61
Figure 3.9: Confusion Matrix for Training Data for BA-CNN	62
Figure 3.10: Confusion Matrix for Validation Data for BA-CNN	62
Figure 3.11: Confusion Matrix for Testing Data for BA-CNN.....	63
Figure 3.12: Number of Functions Evaluations	63

Figure 3.13: Training Progress for BA-BO-CNN	65
Figure 3.14: Confusion Matrix for Training Data for BA-BO-CNN	66
Figure 3.15: Confusion Matrix for Validation Data for BA-BO-CNN	66
Figure 3.16: Confusion Matrix for Testing Data for BA-BO-CNN.....	67
Figure 4.1: Flowchart for the Steps of Creating Artificial Porosity Images	74
Figure 4.2: Probability Plot for X-Position.....	80
Figure 4.3: Probability Plot for Y-Position.....	80
Figure 4.4: 3D Cube with 33 Pores and Average Pores Diameter of 36.94 μm.....	81
Figure 4.5: An Illustrative Example of the Cube Slicing	82
Figure 4.6: An Illustrative Example of the First Version of Artificial Porosity Images	84
Figure 4.7: An Illustrative Example of the Second Version of Artificial Porosity Images	85
Figure 4.8: An Illustrative Example of the Final Version of Artificial Porosity Images	86
Figure 4.9: An Illustrative Example of the Real Existing Porosity Images (Feng et al., 2022)	87
Figure 4.10: Real Image (Left) Vs First Version of Artificial Image (Right).....	88
Figure 4.11: Real Image (Left) Vs Improved First Version of Artificial Image (Right)	89
Figure 4.12: Real Image (Left) Vs Second Version of Artificial Image (Right)	89
Figure 4.13: Real Image (Left) Vs Final Version of Artificial Image (Right)	90
Figure 5.1: Original Slice and Binary Images (Gong et al., 2019).....	94
Figure 5.2: Second Version of Artificial Image (Left) Vs Binarized Artificial Image (Right)	95
Figure 5.3: Second Version of Artificial Image with No Pore (Left) Vs Binarized Artificial Image (Right)	95
Figure 5.4: Final Version of Artificial Image (Left) Vs Binarized Artificial Image (Right)	96
Figure 5.5: Final Version of Artificial Image with No Pore (Left) Vs Binarized Artificial Image (Right)	97

Figure 5.6: Binary Images with Different Sensitivity Factors and Percent of Porosity	98
Figure 5.7: An Illustrative Diagram for Hybrid BA-BO-RCNN Algorithm	102
Figure 5.8: An Illustrative Diagram for Hybrid BA-RCNN Algorithm	102
Figure 5.9: The Steps of MATLAB Code for Hybrid Regression CNN Algorithms	103
Figure 5.10: Artificial Image with Noise of 0.2 (Left) Vs Binarized Artificial Image (Right)	109
Figure 5.11: Artificial Image with Noise of 0.25 (Left) Vs Binarized Artificial Image (Right)	109
Figure 5.12: Prediction Accuracy for Percent of Porosity at Different Levels of Noise	111
Figure 6.1: The General Framework for the Proposed BA-CNN-LSTM Algorithm	116
Figure 6.2: Intuitive Diagram of LSTM (Thakur, 2018)	119
Figure 6.3: The Way of Working for the Bidirectional LSTM Layer (Newman, 2020)	120
Figure 6.4: The Steps of MATLAB Code for the Proposed Hybrid BA-CNN-LSTM Algorithm	123
Figure 6.5: Workflow Diagram for the Proposed Hybrid BA-CNN-LSTM Algorithm	124
Figure 6.6: Training Progress for the Proposed Hybrid BA-CNN-LSTM Algorithm (Artificial Porosity Images)	127
Figure 6.7: An Illustrative Example of the Three Classes of ECG Time Series Signals	130
Figure 6.8: Training Progress for the Proposed Hybrid BA-CNN-LSTM Algorithm (ECG Dataset)	132
Figure 6.9: Confusion Matrix for the Training Set of the Proposed Hybrid BA-CNN-LSTM Algorithm	133
Figure 6.10: Confusion Matrix for the Validation Set of the Proposed Hybrid BA-CNN-LSTM Algorithm	133

Figure 6.11: Confusion Matrix for the Testing Set of the Proposed Hybrid BA-CNN-LSTM Algorithm134
Figure 6.12: Training Progress for the Proposed Hybrid BA-LSTM Algorithm (Engine Dataset)136

List of Tables

Table	Page
Table 2.1: Deep Learning Networks.....	12
Table 2.2: Hybrid CNN Accuracy	24
Table 2.3: Additive Manufacturing Processes (Farinia Group, 2018).....	31
Table 2.4: Smart Manufacturing Technologies (Autodesk).....	42
Table 3.1: CNN Optimization Variables Information.....	45
Table 3.2: Factors and Levels Definition	45
Table 3.3: Taguchi Orthogonal Array for CNN Parameters (L9) (Fraley et al., 2020)	46
Table 3.4: ANOVA Results	46
Table 3.5: Optimal CNN Parameters Values for BA-CNN.....	61
Table 3.6: Optimal CNN Parameters Values for BO-CNN	64
Table 3.7: Optimal Weight Learning Rate Factors for BA-BO-CNN	64
Table 3.8: Classification Accuracy and Computational Time for Algorithms (Cifar10DataDir Dataset).....	67
Table 3.9: Classification Accuracy and Computational Time for Algorithms (Digits Dataset)	68
Table 3.10: Classification Accuracy and Computational Time for Algorithms (Concrete Cracks Dataset)	68
Table 3.11: Classification Accuracy and Computational Time for Algorithms (ECG Dataset)	69
Table 4.1: Number of Pores Data (Shrestha, et al., 2019)	76
Table 4.2: Pores Diameter Data (Shrestha, et al., 2019).....	77
Table 4.3: Number of Pores and Pores Diameter for Combined Laser Power and Scanning Speed.....	78
Table 5.1: Prediction Accuracy and Computational Time for Algorithms (Second Version).....	104
Table 5.2: Optimal Weight Learning Rate Factors for BA-BO-RCNN (Second Version).....	105

Table 5.3: Optimal CNN Parameters Values for BA-RCNN (Second Version)	106
Table 5.4: Prediction Accuracy and Computational Time for Algorithms (Final Version).....	106
Table 5.5: Optimal Weight Learning Rate Factors for BA-BO-RCNN (Final Version).....	107
Table 5.6: Optimal CNN Parameters Values for BA-RCNN (Final Version).....	108
Table 5.7: Prediction Accuracy using Image Binarization Method with Different Levels of Noise	110
Table 5.8: Prediction Accuracy using BA-RCNN Algorithm with Different Levels of Noise	110
Table 5.9: Porosity Prediction Accuracy (Second Version).....	113
Table 5.10: Porosity Prediction Accuracy (Final Version)	113
Table 6.1: The Values of LSTM Parameters in the Four Evaluations of BA (Artificial Porosity Images).....	126
Table 6.2: The New Learning Rate Values of LSTM Parameters (Artificial Porosity Images).....	127
Table 6.3: The Average Error for Percent of Porosity (Artificial Porosity Images)	128
Table 6.4: The Prediction Accuracy and Time for Percent of Porosity (Artificial Porosity Images).....	129
Table 6.5: The Values of LSTM Parameters in the Evaluations of BA (ECG Dataset)	131
Table 6.6: The New Learning Rate Values of LSTM Parameters (ECG Dataset) ..	132
Table 6.7: The Classification Accuracy and Time (ECG Dataset).....	134
Table 6.8: The Values of LSTM Parameters in the Four Evaluations of BA (Engine Dataset)	135
Table 6.9: The New Learning Rate Values of LSTM Parameters (Engine Dataset)	136
Table 6.10: The Prediction Accuracy and Time (Engine Dataset).....	137
Table 6.11: Comparison between All Algorithms Used for Porosity Prediction Accuracy (Final Version)	138
Table Appendix 1.1: Actual Percent of Porosity Dataset.....	195

List of Abbreviations

Additive Manufacturing (AM)

Analysis of Variance (ANOVA)

Artificial Bee Colony (ABC)

Artificial Intelligence (AI)

Artificial Neural Network (ANN)

Auto Encoder Network (AEN)

Bayesian Convolutional Neural Network (BO-CNN)

Bayesian Optimization (BO)

Bayesian Regression Convolutional Neural Network (BO-RCNN)

Bees Algorithm (BA)

Bees Bayesian Convolutional Neural Network (BA-BO-CNN)

Bees Bayesian Regression Convolutional Neural Network (BA-BO-RCNN)

Bees Convolutional Neural Network (BA-CNN)

Bees Convolutional Neural Network Long Short-Term Memory (BA-CNN-LSTM)

Bees Regression Convolutional Neural Network (BA-RCNN)

Binder Jetting (BJT)

Charged-Coupled Device (CCD)

Computer Aided Design (CAD)

Computer-Aided Manufacturing (CAM)

Computed Tomography (CT)

Convolutional Neural Network (CNN)

Convolutional Neural Network Long Short-Term Memory (CNN-LSTM)

Convolutional Neural Network with Evolutional Algorithm (EA-CNN)

Convolutional Neural Network with Fuzzy Logic (FL-CNN)

Convolutional Neural Network with Genetic Algorithm (GA-CNN)

Cyber-Physical System (CPS)

Deconvolutional Neural Network (DNN)

Deep Belief Network (DBN)

Deep Convolutional Inverse Graphics Neural Network (DCIGNN)

Deep Convolutional Neural Network (DCNN)

Deep Learning (DL)

Deep Recurrent Network (DRN)

Denosing Auto Encoder Network (DAEN)

Design of Experiments (DOE)

Differential Evolution (DE)

Digital Light Processing (DLP)

Directed Energy Deposition (DED)

Direct Metal Deposition (DMD)

Direct Metal Laser Sintering (DMLS)

Electrocardiogram (ECG)

Expected Improvement (EI)

Evolutionary Algorithm (EA)

Fused Deposition Modelling (FDM)

Fuzzy Logic (FL)

Gated Recurrent Network (GRN)

Generative Adversarial Network (GAN)

Genetic Algorithm (GA)

Infrared (IR)

Initial Learning Rate (ILR)

Internet of Things (IoT)

Laminated Object Manufacturing (LOM)

Laser Engineered Lens Shaping (LENS)

Laser Metal Deposition (LMD)

Long Short-Term Memory (LSTM)

Machine Learning (ML)

Magnetic Resonance Imaging (MRI)

Material Jetting (MJT)

Momentum (M)

Paper Lamination Technology (PLT)

Particle Swarm Optimization (PSO)

PolyJet (PJ)

Powder Bed Fusion (PBF)

Recurrent Neural Network (RNN)

Recursive Network (RN)

Regression Convolutional Neural Network (RCNN)

Regularization (R)

Restricted Boltzmann Machine (RBM)

Root Mean Square Error (RMSE)

Scanning Electron Microscope (SEM)

Section Depth (SD)

Selective Laser Melting (SLM)

Selective Laser Sintering (SLS)

Sequence to Sequence Model (S2SM)

Sparse Auto Encoder Network (SAEN)

Stereolithography (SLA)

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent Momentum (SGDM)

Structural Similarity Index (SSI)

Ternary Bees Algorithm (BA-3+)

Ultrasonic Additive Manufacturing (UAM)

Validation Accuracy (VA)

Variational Auto Encoder Network (VAEN)

Vat Photopolymerization (VPP)

List of Symbols

X : The input vector to the SoftMax function

z_j : The input of all elements in the vector

e^{z_j} : The exponential function applied to each element

$\sum e^{z_k}$: The normalization term

K : The number of classes

$\mu(f(x))$: The prediction from the gaussian process for new data point x

$\Phi(z)$: The standard normal cumulative probability density

$\sigma(f(x))$: The standard deviation of the prediction for new data point x

$\phi(z)$: The standard normal probability density

μ_x : The local mean for image x

μ_y : The local mean for image y

σ_x : The standard deviation for images x

σ_y : The standard deviations for images y

σ_{xy} : The cross-covariance for images x and y .

α : The exponent for luminance

β : The exponent for contrast

γ : The exponent for structural

X_t : The current time cycle input

U_f : The input weight matrix

H_{t-1} : The previous time cycle hidden state

W_f : The hidden state weight matrix

List of Publications

Journal Articles:

1. Alamri NMH, Packianather M, Bigot S. Optimizing the Parameters of Long Short-Term Memory Networks Using the Bees Algorithm. *Applied Sciences*. 2023; 13(4):2536. <https://doi.org/10.3390/app13042536>.
2. Alamri NMH, Packianather M, Bigot S. Predicting the Porosity in Selective Laser Melting Parts Using Hybrid Regression Convolutional Neural Network. *Applied Sciences*. 2022; 12(24):12571. <https://doi.org/10.3390/app122412571>.
3. Alamri, N. M. H., Packianather, M., & Bigot, S. (2022). Deep learning: parameter optimization using proposed novel hybrid bees Bayesian convolutional neural network. *Applied Artificial Intelligence*, 1-25.

Conference Papers:

1. Alamri, N. M. H., Packianather, M., & Bigot, S. (2023). A Novel Hybrid Bees Regression Convolutional Neural Network (BA-RCNN) Applied to Porosity Prediction in Selective Laser Melting Parts. Cardiff University School of Engineering Research Conference 2023. **(Submitted on 28 Feb 2023 and it will be presented if accepted on 12-14 July 2023 – it will be published after the presentation)**.
2. N. M. H. Alamri, M. Packianather and S. Bigot, "Optimization of Convolutional Neural Network Topology and Training Parameters using Bees Algorithm," 2022 IEEE 2nd International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), Gunupur, Odisha, India, 2022, pp. 1-6, doi: 10.1109/iSSSC56467.2022.10051487.

Chapter Book:

1. Alamri, N. M. H., Packianather, M. (2022). Chapter Book with the title "Optimisation of Convolutional Neural Network Parameters using Bees Algorithm" **(Accepted on 14 Nov 2022 – expected to be published at the end of 2023)**.

Chapter 1: Introduction

1.1 Background

Artificial Intelligence (AI) is an essential concept used to facilitate the development of intelligent systems (Li et al., 2017) in order to increase the productivity and maximize the efficiency of the processes such as manufacturing machines. The most popular AI techniques are based on Artificial Neural Network (ANN), a type of Machine Learning (ML) inspired by the biological nervous system. These computing systems can be used to model big data and find complex relationships (De Filippis et al., 2017). ANNs are capable of handling high-dimensional real-time data and extracting implicit meaningful patterns that can be used to predict the future state of complex systems (Wuest et al., 2016). In addition, ANNs are capable of handling complex dynamic problems due to their ability to deal with nonlinearity. It is worth noting that ANNs are trained on historic data with relative ease by adjusting their control parameters such as learning rate and momentum.

Big Data analytics (Geissbauer et al., 2016) is an integral part of the industry 4.0 paradigm, known as the fourth industrial revolution which aims at creating smart systems where technologies are transformed by the Internet of Things (IoTs), Cyber-Physical Systems (CPSs) and cloud computing. Modelling an IoT system is based on modelling a stochastic system addressing the relationship between process and system performance and providing a quantitative analysis of system performance (Ciortea, 2018). On the other hand, many challenges are remaining when applying ANNs (Wuest et al., 2016), the biggest one is the data acquisition issue as the availability of relevant data is not guaranteed. In addition, applying appropriate data mining after collecting a dataset can also be a challenging task, particularly for cases where a high amount of irrelevant data may have been collected, thus affecting the performance of the produced models.

As an extension to ANN capabilities, Deep Learning (DL) techniques are now well-established, producing better learning capability as stated in (Singh et al., 2018). They have the advantage of using automatic feature extraction by learning a large number of nonlinear filters before the decision making stage. One of the most popular DL networks is Convolutional Neural Network (CNN) (Singh et al., 2018) which is used mainly with image data. It can be integrated with other intelligent swarm optimization algorithms to optimize its parameters in order to improve the performance of the CNN model in analysing a large number of images for classification and prediction problems.

Firstly, this thesis addresses the need for designing better CNN topology by proposing a novel nature inspired hybrid algorithm that takes into account the advantages of global, local, and intense searches in the Bees Algorithm (BA) which is known to mimic the behaviour of honeybees, to optimize the CNN parameters which is called the Bees Convolutional Neural Network (BA-CNN) algorithm. Furthermore, another nature inspired hybrid algorithm is proposed which combines the BA with Bayesian Optimization (BO) in order to increase the performance of the CNN which is referred to as BA-BO-CNN algorithm.

In addition, Long Short-Term Memory (LSTM) is one of the DL networks that deals with time series or sequential data, it is an advanced Recurrent Neural Network (RNN) that persists the information for a long period so that it can remember long-term dependencies (the-learning-machine). The RNN can persist the information as well, but for a shorter period, so it cannot remember the dependencies in the long term, the LSTM is established to address this problem in RNN (Thakur, 2018). It consists of three gates, the first one is the forget gate that decides if it is needed to remember the information coming from the previous time scale or not, the second one is the input gate that learns information from the cell input, and the last gate is the output gate where the cell transfers the updated information to the next time cycle. In addition, it has a cell state that carries the information along with all cycles and a hidden state for short-term memory (www.analyticsvidhya.com). The performance of the LSTM in dealing with the sequential data can be improved by optimizing the parameters related to the three gates and cell state.

Secondly, this thesis proposes a novel hybrid nature inspired algorithm that uses the BA to improve the performance of three gates and cell state, specifically by optimizing the learning rate factor so that each part has its learning rate determined based on the global learning rate. Having a more optimum learning rate means more optimum updates for the network weight (Brownlee, 2020). The hybrid Bees Convolutional Neural Network Long Short-Term Memory referred to as BA-CNN-LSTM combines the best DL networks in dealing with images and sequential data as CNN is the most powerful DL network in analysing images (Elngar et al., 2021) and (Chaganti et al., 2020). In addition, the LSTM network is the best algorithm for dealing with sequential data, as it persists the information

for a long period remembering long-term dependencies which address the problem of normal RNN (the-learning-machine).

CNN and LSTM algorithms are effective DL networks that can be applied in the manufacturing context, particularly to analyse images related to Additive manufacturing (AM). It is a process that builds objects by joining material in successive layers, under computer control and using data from a 3D model (Farinia Group, 2018). AM is also referred by other terms such as 3D printing, rapid prototyping, digital manufacturing, and layered manufacturing. Initially, AM or 3D printing technologies were used mainly for rapid prototyping, but with improvements in the reliability and efficiency of the AM processes as well as in the material properties of the components produced, they are increasingly used in more advanced applications such as to create highly customized products, producing a small volume of serial components and visualizing tool in design. Future applications may concern human organ creation, clothes manufacturing, and food confection (Farinia Group, 2018).

There are seven AM processes: powder bed fusion, material extrusion, material deposition, binder jetting, sheet lamination, vat photopolymerization, and directed energy deposition. The focus of this thesis will be on powder bed fusion. (Sun et al., 2017) stated that it is a laser-based additive manufacturing in which the laser beam scans the selected locations of the powder bed at a controlled speed and then it fuses the powder to the solid material by either partial melting in Selective Laser Sintering (SLS) or full melting in Selective Laser Melting (SLM) which is a metal-based process that uses the laser selectively to melt the powder after layer by layer fabrication as mentioned in (Shrestha, et al., 2019). They mentioned that metal-based AM has many issues such as porosity, part deformation and cracks. The porosity is the most challenging issue as stated in (Snell et al., 2020). The paper mentioned that the porosity has a significant effect on the mechanical properties as it causes structural failures. Assessing the porosity in the SLM parts is a challenging issue, the drawback of using the existing gray value analysis method to assess the porosity is the difficulty and subjectivity in selecting a uniform grayscale threshold to convert a single slice into binary images to highlight the porosity.

Thirdly, this thesis proposes a new approach based on the use of a Regression Convolutional Neural Network (RCNN) algorithm to predict the percent of porosity in CT scans of the finished SLM parts, without the need for subjective difficult thresholding determination to convert a single slice into binary images. In order to test the algorithm, as the training of the RCNN would require a large amount of experimental data, artificial porosity images mimicking real CT scan slices of the finished SLM parts are created using a new efficient method. In addition, the LSTM network is added to CNN (CNN-LSTM) to enhance the porosity prediction as it has better capability in dealing with the sequential layers of the SLM parts. BA is used as well to optimize the parameters of DL networks in order to improve the model's performance in predicting the percent of porosity.

1.2 Aim and Objectives

The aim of this thesis is to improve the performance of convolutional neural networks using nature inspired algorithms and apply them to predict the porosity in the finished parts manufactured by the selective laser melting process.

The objectives to achieve the aim in terms of CNN are:

1. Optimize CNN parameters using BA and BO techniques.
2. Apply the developed hybrid CNN algorithms to predict the percent of porosity in the finished parts of the SLM process based on porosity images.
3. Analyse the results and make further improvements to the developed algorithms as necessary.

The objectives to achieve the aim in terms of the SLM process are:

1. Investigate the limitations of the existing method used to predict the percent of porosity in the finished parts of the SLM process.
2. Apply the existing approach to predict the percent of porosity.
3. Compare the prediction results of the existing method with the hybrid CNN algorithms in order to validate the issue.

1.3 Research Questions

The research questions that will be addressed are:

- **Q1:** How to design the optimum topology for the Convolutional Neural Network using the Bees Algorithm that can be used for several applications including porosity prediction?
- **Q2:** How to improve the way of analysing the porosity in the parts produced by the Selective Laser Melting process using the novel optimised Bees Convolutional Neural Network?
- **Q3:** How to improve the performance of the Long Short-Term Memory Network in dealing with sequential data using the Bees Algorithm for enhancing the porosity prediction in the sequential layers of the Selective Laser Melting parts?

1.4 Thesis Outline

This thesis consists of the following chapters:

Chapter 1- Introduction: gives the background to the topic of research, states the aim, objectives, research questions, thesis outline, and the limitations and assumptions made in this study.

Chapter 2- Literature Review: presents a review of DL in general and of CNN and LSTM in particular in order to find gaps and open issues that need to be fulfilled in the future to improve the performance of the DL models. Also, it shows the impact and recent applications of optimizing DL algorithms using nature inspired algorithms showing the gap that will be addressed in the next chapter. In addition, it states AM processes, applications, advantages, and gaps and open issues that need to be addressed. It focuses on powder bed fusion processes showing the way of working, thermodynamical phenomena, parameters, open issues, three porosity types, and state of the art studies about adopting DL techniques to improve the performance of SLM process.

Chapter 3- Convolutional Neural Network Parameters Optimization using Nature Inspired Algorithms: investigates the significant CNN parameters that affect the classification accuracy on the validation set. The chapter proposes a new hybrid BA-CNN

algorithm that uses the BA to optimize CNN parameters in order to increase the classification accuracy of the network. In addition, it proposes a novel nature inspired hybrid algorithm that combines the BA with the BO in order to increase the overall performance of CNN which is referred to as BA-BO-CNN algorithm. The results of applying the hybrid BA-CNN and BA-BO-CNN algorithms to four benchmark image data are shown in this chapter. In addition to these datasets, the following chapter creates artificial porosity images mimicking the real CT scans of the finished SLM parts that can be analysed using the hybrid algorithm developed in this chapter.

The contributions of the chapter are:

- Developing a novel hybrid Bees Convolutional Neural Network (BA-CNN) algorithm in order to improve the performance of CNN.
- Developing a novel hybrid Bees Bayesian Convolutional Neural Network (BA-BO-CNN) algorithm in order to improve the performance of CNN.

Chapter 4- Artificial Porosity Images Creation for Selective Laser Melting Parts: proposes a new efficient approach of creating artificial porosity images mimicking the real CT scan slices of the finished SLM parts with a similarity index of 0.9967. This chapter's contribution which will be continued further in the following chapter is:

- Developing a new approach to create a large amount of experimental artificial porosity images similar to real images by a similarity index of 0.9967 which can be used in the research environment efficiently and effectively in order to predict the percent of porosity in the finished SLM part using the developed hybrid CNN algorithms as will be described in the following chapter.

Chapter 5- Hybrid Regression Convolutional Neural Network for Predicting the Percent of Porosity in Selective Laser Melting Parts: highlights the limitations of predicting the percent of porosity in the finished SLM parts using the existing image binarization method and it proposes a new approach based on the use of the RCNN algorithm in order to predict the percent of porosity in CT scans of the finished SLM parts, without the need for subjective difficult thresholding determination to convert a single slice into binary images. The algorithm is then further developed by optimizing its parameters

using the BA to produce a better prediction accuracy using Bees Regression Convolutional Neural Network (BA-RCNN). Chapters 4 and 5 produce the following contribution:

- Proposing and validating a new approach for predicting the percent of porosity in the finished SLM parts, using hybrid Bees Regression Convolutional Neural Network (BA-RCNN). It was demonstrated that a better accuracy than the existing image binarization method could be achieved (approximately 17% improvement with the data set used). In order to test the algorithm, as the training of the RCNN would require a large amount of experimental data, artificial porosity images mimicking real CT scan slices of the finished SLM part were created with a similarity index of 0.9976 with real images.

Chapter 6- Hybrid Long Short-Term Memory Network with Bees Algorithm for Enhancing the Porosity Prediction in Selective Laser Melting Parts: proposes a nature inspired algorithm that improves the performance of LSTM in dealing with the sequential data by optimizing the parameters related to the three gates and cell state. The novel hybrid nature inspired algorithm uses the BA to improve the performance of three gates and cell state, specifically by optimizing the learning rate factor so that each part has its learning rate determined based on the global learning rate. Having a more optimum learning rate means more optimum updates for the network weight. Artificial porosity images are used for testing the algorithms; as the input data are images, CNN is added in order to extract the features in the images for feeding it into the LSTM in order to predict the percent of porosity in the sequential layers of artificial porosity images that mimic real CT scan images of products manufactured by SLM process. In addition, it is applied in other contexts such as in signal processing to classify Electrocardiogram (ECG) benchmark image data. Also, it can deal with time series numerical data, CNN is not needed in this case as there is no feature extraction task for the images. The contribution of this chapter is:

- Improving the performance of the LSTM network in predicting sequential data using BA. As the input data are images, CNN is added to extract the image features yielding a hybrid algorithm (BA-CNN-LSTM) that provides a 10% more accurate prediction of porosity percentage appearing in the sequential layers of artificial porosity images that mimic CT scan images of parts manufactured by SLM process.

Chapter 7- Conclusion: summarises the thesis contributions to knowledge, shows study limitations and suggests recommendations for future work.

1.5 Study Limitations and Assumptions

One of the limitations of this study is the number of evaluations used in conducting BA and BO to optimize DL parameters, they are limited to the computer capability in advanced research computing at Cardiff University. As DL networks require high computations, the number of evaluations is limited.

This research is not aimed to study the porosities in depth, so no experiments have been conducted, but the work proposes DL methods that will enhance such studies, particularly in predicting the percent of porosity in the finished SLM part.

The creation of artificial porosity images is only based on the laser power and scanning speed parameters as the dataset used to create the images presents only data about laser power and scanning speed. Finally, the created artificial porosity images illustrate only one type of pore which is keyhole porosity considering one shape of this type which is the nearly spherical shape.

Chapter 2: Literature Review

2.1 Deep Learning

This section presents a brief review of DL showing its history, definition, advantages, limitations, and applications.

2.1.1 History and Definition

The history of ANN started in 1943 with using threshold logic to create a computer model based on the neural network of the human brain by Walter Pitts and Warren McCulloch (Foote, 2022). Then, the perceptron was created in 1957 by Frank Rosenblatt (Machine Learning Knowledge, 2019). In 1960, the back propagation model was developed by Henry Kelley, it was simplified by Stuart Dreyfus to be based on chain rule only. In the early 1980s, John Hopfield created a recurrent neural network (Machine Learning Knowledge, 2019) and Kunihiko Fukushima designed the first CNN with multiple convolutional and pooling layers in the 1970s. The first practical lab for backpropagation demonstration was provided by Yann LeCun in 1989, he combined it with CNN to detect handwritten digits. The development of the LSTM network was in the 2000s to solve the vanishing gradient problem (Kamalika, 2018). In 2009, ImageNet was launched by a professor at Stanford Fei-Fei Li, it assembled 14 million of labelled images to be used for free. In 2011, GPU speed had increased significantly making DL an efficient tool which result in the development of a pre-trained CNN called AlexNet. Generative Adversarial Network was developed by Ian Goodfellow in 2014 to imitate images using a generator that generates the image and a discriminator that discriminates the real and generated images, both networks compete against each other until the near perfect image is produced (Foote, 2022).

DL techniques are well-established as an extension to ANN capabilities, producing better learning capability as stated in (Singh et al., 2018), they said that deep learning is a class of ML techniques which utilizes multiple processing layers where the output from a previous layer is used as an input for the following layer, to learn representations of data with multiple levels of abstraction. The difference between traditional learning and DL can be seen in terms of the feature extraction process, which depends on the data and the model types. Using the traditional learning leads to significant time being consumed while applying a trial-and-error approach to the feature extraction process and the success depends on the user experience. On the other hand, the DL approach will benefit from an

automatic feature extraction process through the learning of a large number of nonlinear filters before making decisions. Thus, DL combines feature extraction and decision making within one model and avoids the often-suboptimal manual handcrafting.

2.1.2 Advantages, Limitations, and Applications

DL have many network structures that can be used in different real-life applications. Each network will have its advantages and limitations depending on the applications as shown in the following table 2.1:

Table 2.1: Deep Learning Networks

Network	Advantages	Limitations	Applications	References
Convolutional Neural Network (CNN)	<p>Eliminating the need for extracting the features manually, it is achieved by alternating and stacking pooling and convolutional layers that convolve with the input data to extract the most significant features</p> <p>Can be used with image data, text data, time series data and sequence input data</p> <p>Can be built from scratch or retrained enabling to build on the previous existing networks</p> <p>Has many pre-trained architectures like AlexNet, SqueeZNet, GoogLeNet, ZFNet, RESNet and VGGNet</p> <p>Sparse interaction with local connectivity</p> <p>Reducing the number of parameters</p>	<p>High computational complexity in model training</p>	<p>Image, pattern, and speech recognition</p> <p>Image and sentence classification</p> <p>Prediction problems</p> <p>Regression problems</p>	<p>(Singh et al., 2018)</p> <p>(Wang et al., 2018)</p> <p>(MathWorks-1)</p> <p>(Tch, 2017)</p>

Network	Advantages	Limitations	Applications	References
Deconvolutional Neural Network (DNN)	Producing a vector related to an image and using that vector to draw that image. It is reversed convolutional neural network	High computational complexity in model training	Image drawing	(Tch, 2017)
Deep Convolutional Inverse Graphics Neural Network (DCIGNN)	Training images that have not previously been trained before Removing an object from the image and replacing it with another object or repainting it	High computational complexity in model training	Image processing	(Tch, 2017)
Deep Recurrent Network (DRN)	Making a directed cycle so that the outputs do not only depend on the immediately previous input Can be used with text data, speech data, time Series data, tabular data, and generative models Allowing the information to persist in a hidden layer and capturing the previous states and temporal correlation	Cannot be used with image data Difficult to memorize the long-term interaction in training	Generate novel sentences Document summaries Speech and connected handwriting recognition Classification problems Regression prediction problems	(Wang et al., 2018) (Tch, 2017) (Davydova, 2017)
Sequence to Sequence Model (S2SM)	It is a model with two recurrent networks, including an encoder to process the input and a decoder to produce the output	It is limited to sequence problems	Chatbots System for answering questions and machine translation	(Tch, 2017) (Davydova, 2017)

Network	Advantages	Limitations	Applications	References
Long Short-Term Memory (LSTM)	Modelling temporal sequences along with their long-term dependencies more accurately than traditional recursive neural network	There is no activation function and no modification for the stored values	Speech recognition Classification problems Regression prediction problems	(Tch, 2017) (Davydova, 2017)
Gated Recurrent Network (GRN)	Modelling temporal sequences along with their long-term dependencies more accurately than traditional recursive neural network It is similar to the LSTM network but without an output gate	There is no activation function and no modification for the stored values	Speech recognition Classification problems Regression prediction problems	(Tch, 2017)
Deep Belief Network (DBN)	Inferring the states of the unobserved variables Adjusting the interaction between variables to make the network generate observed data Can be trained without supervision Training using a very fast algorithm to initialize the network and parameters	It is limited to binary inputs The parameters are fine-tuned using a slow algorithm	Image recognition Face recognition Video sequence recognition Motion-capture data	(Wang et al., 2018) (Chandrayan, 2017)

Network	Advantages	Limitations	Applications	References
Restricted Boltzmann Machine (RBM)	<p>Can quickly discover unbiased samples from the posterior distribution</p> <p>Can be trained without supervision</p> <p>Automatic training for the datasets avoiding the local minimum values</p> <p>Robust to ambiguous inputs</p>	<p>It is limited to binary inputs</p> <p>There is no output layer</p> <p>Restricted connection between hidden units</p> <p>Time consuming</p>	<p>Filtering</p> <p>Prediction</p>	<p>(Wang et al., 2018)</p> <p>(Chandrayan, 2017)</p>
Recursive Network (RN)	<p>It has the ability to make structured predictions</p>	<p>Used only when applying the same weights recursively</p>	<p>Speech recognition</p>	<p>(Davydova, 2017)</p>
Auto Encoder Network (AEN)	<p>Can be trained without supervision</p> <p>Eliminating the irrelevance in the input and preserving the meaningful information</p>	<p>Used only when output cells are equal to input cells and hidden cells are less than input cells</p> <p>Error propagation</p>	<p>Classification</p> <p>Clustering</p> <p>Feature compression</p>	<p>(Wang et al., 2018)</p> <p>(Tch, 2017)</p>
Variational Auto Encoder Network (VAEN)	<p>Using probability compression instead of feature compression</p> <p>Can be trained without supervision</p> <p>Eliminating the irrelevance in the input and preserve the meaningful information</p>	<p>Used only when output cells are equal to input cells and hidden cells are less than input cells</p> <p>Error propagation</p>	<p>Classification</p> <p>Clustering</p> <p>Probability compression</p>	<p>(Wang et al., 2018)</p> <p>(Tch, 2017)</p>

Network	Advantages	Limitations	Applications	References
Denoising Auto Encoder Network (DAEN)	Varying the data by random bit and reconstructing output from a bit input making it more general which helps to select more common features Can be trained without supervision Eliminate the irrelevance in the input and preserve the meaningful information	Used only when output cells are equal to input cells and hidden cells is less than input cells Error propagation	Classification Clustering Feature compression	(Wang et al., 2018) (Tch, 2017)
Sparse Auto Encoder Network (SAEN)	Can reveal hidden grouping pattern Can be trained without supervision Eliminating the irrelevance in the input and preserve the meaningful information	Used only when the number of hidden cells is more than the input and output cells Error propagation	Classification Clustering Feature compression	(Wang et al., 2018) (Tch, 2017)
Generative Adversarial Network (GAN)	Very large family of double networks Learning a map from input to output image	Generator and discriminator fool each other	Image generation	(Tch, 2017)

These applications are coming from many fields, such as manufacturing. (Wang et al., 2018) mentioned that DL enables advanced analytics for smart manufacturing systems in terms of objects, equipment, process, people, and environment using aggregated big data. DL can help describe what happened in a process by capturing products' conditions and then by supporting the diagnoses of why particular issues occurred, examining the causes, and detecting specific failures. After that, DL can be used to predict what will happen, for example predicting products' quality deviations, and finally to assist in the prescription of corrective actions by identifying measures to be taken in order to improve the quality of a

product. The deep insights brought by DL can support a company's decision making process throughout a product lifecycle by improving the analyses of data emerging from the design, manufacturing, and supply chain, enhancing the process control, and reducing the downtime. DL has been applied in a wide range of manufacturing systems particularly in the area of fault diagnosis and product quality inspection.

In addition to the manufacturing context, there are other fields in which DL has great applications. It has been applied in the pharmaceutical context (Ekins, 2016), such as to predict aqueous solubility, the epoxidation site in molecules, and liver injuries induced by drugs as well as to diagnose cancer, extract pattern in gene expression, predict protein disorder, analyse the content of breast cancer, repurpose drugs, and classify microscope images. Thus, while DL appears to be a promising tool in the biological context, it is generally expected to have even greater applications in the future.

2.2 Convolutional Neural Network

This section presents a brief review of CNN principles in order to find the gaps and open issues that may need to be solved in the future in order to improve the performance of CNN models.

2.2.1 Definition and Way of Working

CNN is one of the most popular DL networks and is used mainly to perform image analysis tasks. (MathWorks-1) stated that it is useful for detecting patterns in images that help in the automatic recognition of real physical objects. These patterns are extracted by CNN directly from image datasets without the need for extracting features manually, which is the most important factor that makes CNN very popular. In addition, it produces highly accurate recognition results and has the flexibility to be retrained to perform new recognition and to be built on previously created models. It provides a better model architecture, enabling advances in detecting and recognizing objects, thus it is a key technology in automated facial recognition.

CNN might have hundreds of layers helping to detect patterns in images. Filters are used to extract information and can start from simple features, such as brightness, to more complex ones that uniquely identify an object. This filtering is applied to each training image and the output of each image after convolution can be used as an input to the following layer. CNNs are composed of an input layer, an output layer and many hidden

layers. Every neuron in the hidden layer connects to all inputs neurons as shown in figure 2.1, as mentioned in (Le, 2015):

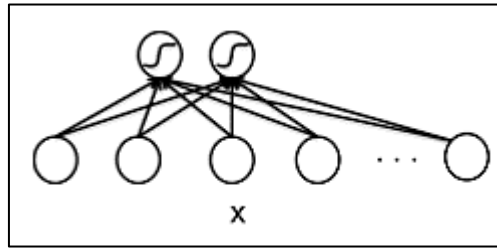


Figure 2.1: Inputs and Hidden Layers Connection in CNN (Le, 2015)

The hidden layers perform learning feature tasks using the most common feature learning layers which are:

- Convolution: it activates some features in the images using convolutional filters which are represented by a matrix of weights that slide along the pixel brightness input matrix to create a feature map matrix using special dot product as mentioned in (Hui, 2017).
- Rectified linear unit (ReLU): it is used after each convolutional layer to increase the speed and effectiveness of training by mapping negative values to zeros and maintaining positive values which is helpful as an activation.
- Batch normalization: it is used as supplement layers after each convolutional layer to mitigate the risk of overfitting by normalizing the input values of the following layers (Yamashita et al., 2018).
- Pooling: it is used between the convolutional layers to reduce the dimensionality of the output volume (McDermott, 2021) without losing the important features which contribute to minimize the computational cost. It reduces the number of parameters needed to learn by making nonlinear down-sampling that simplifies the output. There are two types of this layer, max pooling which takes the most activated feature and average pooling takes the average presence of the feature, so max pooling is better with a dark background and average pooling is better with a white background as mentioned in (Ouf, 2017).

Furthermore, it uses two classification layers:

- Fully connected: it shows the probability of each image being classified for each class.

- SoftMax: it is an activation function that works better with multi-class classification problems rather than a binary classification problem that requires a sigmoid logistic function which is a special case of SoftMax function as shown in (McDermott, 2021), it provides the classification output and may not have any parameters as mentioned in (Wu, 2017). It turns the real values inputs into values between 0 and 1 (Wood) through the following equation (www.redcrab-software.com):

$$\sigma(x)_j = e^{z_j} / \sum e^{z_k} \quad (\text{for } j = 1, 2, \dots, K) \quad \text{(Equation 2.1)}$$

Where:

X is the input vector to the SoftMax function

z_j is the input of all elements in the vector

e^{z_j} is the exponential function applied to each element

$\sum e^{z_k}$ is the normalization term

K is the number of classes

Figure 2.2 demonstrates the way of working for CNN showing feature learning and classification layers.

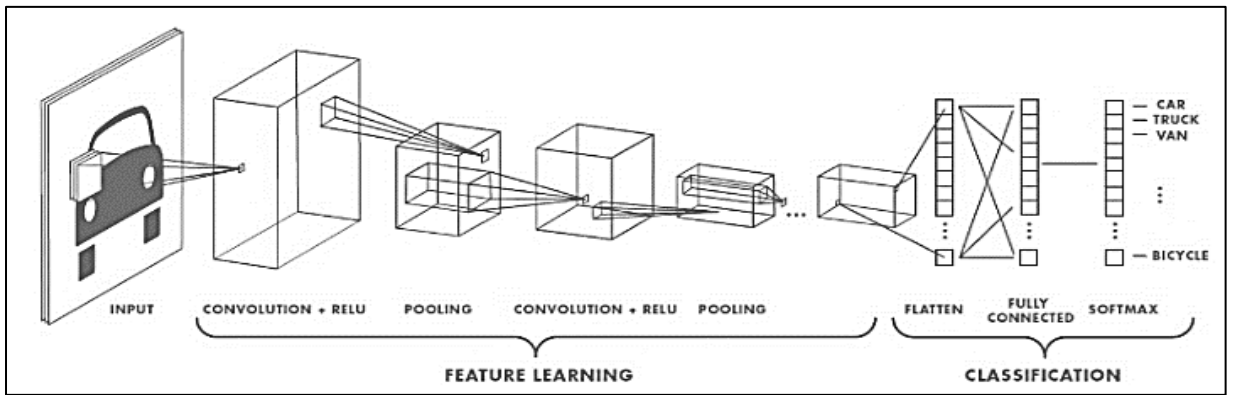


Figure 2.2: The Way of Working for CNN (MathWorks-1)

2.2.2 Gaps and Open Issues

According to (Joshi et al., 2019), the most important challenge in training CNN is the generalization to unseen datasets so that the model does not overfit datasets and can give judgment on unknown data. Overfitting is a common issue in training CNN, where the

model fits well enough to the training data but does not have the capability to generalize to other datasets. It can be controlled by increasing the sample data using data augmentation techniques, reducing the complexity of the architecture and stopping the training earlier. In addition, (Liang & Liu, 2015) and (Cogswell et al., 2015) discussed the same overfitting issue and suggested another way to prevent it using dropout, (Pan, 2017) highlighted that it is an efficient method to randomly remove a unit from the network with related edges independently for each hidden unit and sample. (Wu et al., 2017) used a novel regularization technique that helps in reducing kernel redundancy and thus preventing overfitting. However, improving the learning capability is still an open challenge and can be enhanced continuously. In addition, (Joshi et al., 2019) reported other issues, such as exploding gradient problem where the model stops learning after a certain number of epochs which causes instability in the learning process resulting in NAN values, this issue can be overcome by redesigning the network architecture and selecting appropriate activation function. Also, (Shah et al., 2016) suggested using residual and highway networks that learn in earlier layers allowing for earlier representation. In addition to (Joshi et al., 2019), (Fu et al., 2016), and (Masko & Hensman, 2015) addressed the third issue which is training the data using imbalanced classes where the sample is not uniformly distributed, it is a significant and long-standing challenge in training CNN models. Improving the convergence speed is another future challenge because it sometimes increases the time of convergence in order to get better accuracy as stated in (Chiroma et al., 2019).

Furthermore, five research papers (Zhang et al., 2018), (Baldominos, et al., 2018), (Sinha et al., 2017), (Ma et al., 2018), and (Panwar et al., 2017) discussed the most challenging aspect in training CNN, which is designing better topology, the traditional heuristic approach of using trial and error might result in a less accurate model depending on user experience. Applying optimization techniques such as nature inspired algorithms to optimize the parameters of CNN can improve the performance of the model. However, designing a better CNN topology is still an open issue and there is no approach found yet that can give the best CNN topology. Section 2.4 will discuss the impact of developing hybrid CNN with nature inspired algorithms in improving the model performance.

2.3 Long Short-Term Memory Network

This section shows a review of LSTM network presenting the definition and way of working along with the gaps and open issues that need to be addressed.

2.3.1 Definition and Way of Working

LSTM is one of the DL networks that deal with time series or sequence problems. It is an extension of RNN that can remember long-term dependencies as it persists in the information for a long period, the normal RNN is not able to do so resulting in a vanishing gradient problem (www.analyticsvidhya.com). It is a situation when the RNN is not able to propagate useful information from the end of the network back to the beginning of the network as information is stored only for a short period (www.engati.com).

The structure of the LSTM consists of three gates, the first one is the forget gate that decides if the information coming from the previous time scale is relevant to be remembered or irrelevant to be forgotten, the second one is the input gate that tries learning new information from the cell input, and the last gate is the output gate where the cell transfers the updated information from the previous time cycle to the next time cycle. In addition, it has a cell state that carries the information along with all cycles as it stores the information for a long period and a hidden state for short-term memory, so it is present in RNN as well (www.analyticsvidhya.com). The following figure 2.3 illustrates the basic structure of the LSTM network where the first part is the forget gate, the second part is the input gate, and the last one is the output gate. In addition, C_{t-1} is the cell state for the previous time cycle, C_t is the cell state for the current time cycle, H_{t-1} is the hidden state for the previous time cycle, and H_t is the hidden state for the current time cycle.

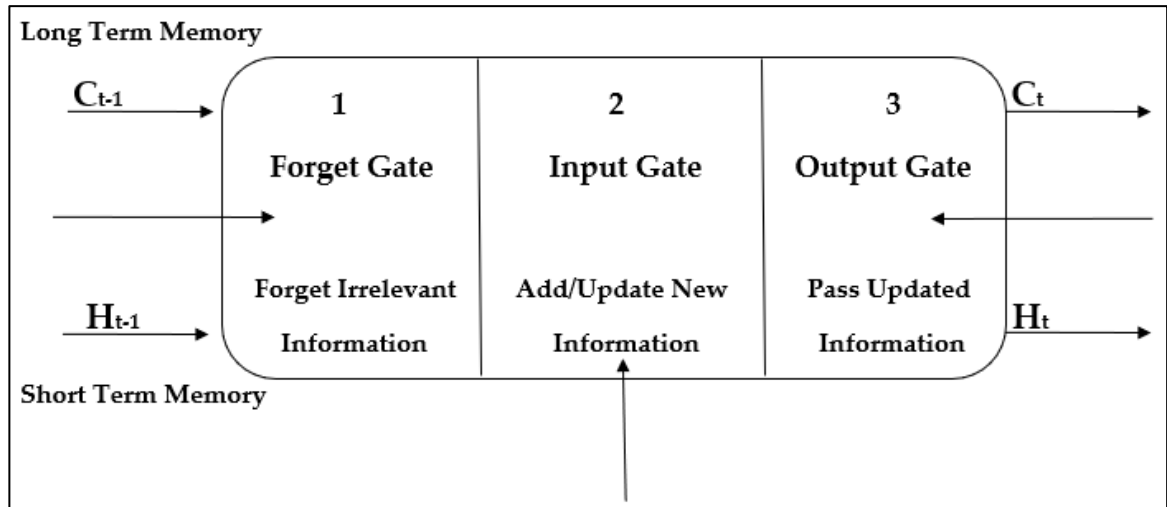


Figure 2.3: The Basic Structure for LSTM (www.analyticsvidhya.com)

The way of working for the gates starts with the forget gate that decides if the information coming from the previous time scale is relevant to be remembered or irrelevant to be forgotten. Then, the input gate is used to quantify the importance of the input of new information. In the output, the cell transfers the updated information from the previous time cycle to the next time cycle. The output with the maximum score is the predicted value.

2.3.2 Gaps and Open Issues

One of the most important challenges in training LSTM is reducing overfitting (www.machinelearningmastery.com), it is the case when the model fits well enough for training data and performs poorly in validation and testing data so that it is not able to generalize to unseen data (Joshi et al., 2019). This issue can be overcome by adding a dropout layer (Liang & Liu, 2015) and (Cogswell et al., 2015), trying more optimum regularization value (Wu et al., 2017), reducing the number of epochs or augmenting the datasets (www.machinelearningmastery.com).

Exploding and vanishing gradient is a problem in RNN when the model stops learning after a certain number of epochs, LSTM addressed this issue (www.analyticsvidhya.com). However, the optimum weight change can be improved further by having a more optimum learning rate as the gradient is multiplied by the learning rate resulting in the optimum set of weights (Varikuti, 2021). So, having a more optimum learning rate means more optimum updates for the network weight (Brownlee, 2020).

Furthermore, improving the LSTM performance is an ongoing challenge (Mattioli et al., 2019), and there is no approach found yet that develops the best architecture. However,

adopting nature inspired algorithms in optimizing the parameters of the network may improve the performance of the model as it reduces the need for human input in assigning the parameters. The following section will present state of the art studies about using nature inspired algorithms to optimize LSTM parameters (Zhang et al., 2018).

2.4 Impact and Recent Applications of Optimizing Deep Learning Parameters using Nature Inspired Algorithms

This section presents the state of the art studies about integrating nature inspired algorithms with DL algorithms along with the gaps and open issues that need to be addressed.

2.4.1 State of the Art Studies

(Chiroma et al., 2019) discussed the synergy between nature inspired algorithms with DL. They mentioned that the inspiration for such algorithms can be from animals' behaviour, human activities and biological systems. The paper presented nature inspired algorithms such as harmony search, firefly, cuckoo search, evolutionary, ant colony optimization, practical swarm optimization, genetic, simulated annealing and gravitational search algorithm. The authors stated that combining DL with nature inspired algorithms has the advantage of solving local minimum problems and improving the performance of the network by increasing the accuracy of its models. In addition, the need for trial and error techniques in determining the parameters of DL architecture is eliminated as nature inspired algorithms can realize the best parameters values automatically. Though, the optimum parameters setting is still an open problem in the research area. The authors suggested to eliminate the need for human interventions in determining the parameters by obtaining parameter-less nature inspired algorithms in the future. Finally, the paper suggested applying meta-optimization which is excessive in the DL area, and it helps to tune optimization methods by using another optimization method.

Furthermore, other research papers discussed the hybrid CNN with nature inspired sward-based optimization techniques such as CNN with Evolutionary Algorithm (EA-CNN) and CNN with Genetic Algorithm (GA-CNN) in addition to other techniques like CNN with Long Short-Term Memory (CNN-LSTM), Artificial Bee Colony with CNN-

LSTM (ABC-CNN-LSTM) and CNN with Fuzzy Logic (FL-CNN). Table 2.2 summarizes the accuracy of the original CNN and improved accuracy after hybridization.

Table 2.2: Hybrid CNN Accuracy

Hybrid Algorithm	Accuracy		Dataset	Reference
	Original CNN	Hybrid CNN		
(EA-CNN)	-	98.88%	MNIST database for handwritten digits recognition	(Badan, 2019)
	-	62.37%	CIFAR10 dataset for animals' image classification	
(GA-CNN)	71.69%	75.95%	Stock market fluctuation prediction	(Chung & Shin, 2019)
(CNN-LSTM)	82.1%	97%	Motion data and site video to recognize workers' unsafe actions	(Ding et al., 2018)
ABC-CNN-LSTM	95%	97%	Products review dataset to detect fake reviews	(Jacob et al., 2022)
(FL-CNN)	97.35%	99.10%	Handwritten digits recognition	(Popko & Weinstein, 2016)

Applying the evolutionary algorithm yielded a highly accurate CNN with 98.88% accuracy for handwritten digit recognition and a lower percentage of 62.37% for animal image classification. The author used the weight inheritance technique which considers the training process as a kind of mutation that reduces the evolution cycle time. (Bernard & Leprevost, 2018) explained the process of evolution by reproducing the population through generation by crossing members and inducing random mutation, evolving input image would maximize feature activation.

GA is one of the evolutionary algorithms that has been applied to optimize the parameters of the CNN model to predict the stock market, (Mallawaarachchi, 2017) described the natural selection of selecting the fittest individual in the population which improved the accuracy from 71.69% to 75.95%. The algorithm produces offspring that

inherit the parents' characteristics, so that they have a chance to survive if their parents have better fitness. So, the algorithm consists of five phases: initial population, fitness function, selection, crossover and mutation.

In addition to nature inspired algorithms, CNN can be integrated with another DL algorithm which is the LSTM to automatically recognize worker unsafe actions in motion data, the use of LSTM would enable the sequence of learning features. Dealing with sequential data is an important advantage of using this algorithm as stated in (Motepe et al., 2019), it is an effective technique when capturing dependencies in the long term avoiding RNN challenges such as vanishing gradient problem using nonlinear gating that regulates the flow of information. The hybrid CNN-LSTM analyses motion data in a video to recognize unsafe actions done by the workers. Applying only CNN achieved an accuracy of 82%, but adding LSTM improved the model accuracy to 97% as it stores the information for a long period which allows considering the long-term dependencies (Ding et al., 2018). The hybrid CNN-LSTM was integrated as well with ABC that optimized the type of the network, the number of epochs, LSTM hidden units, global learning rate, activation function, step size, fully connected layer, and pooling size (Jacob et al., 2022). The authors used the hybrid ABC-CNN-LSTM algorithm to detect the fake reviews of the product with an accuracy of 97% compared to 95% for the CNN-LSTM algorithm (Jacob et al., 2022).

Furthermore, the hybridization of CNN with FL would add one more layer, a fuzzy self-organization layer. (Korshunova, 2018) explained its function that distributes input data into clusters not equivalent to the number of output classes where the output of this layer is the membership function values for the fuzzy clusters. This new hybrid model improved the handwritten digit recognition accuracy from 97.35% to 99.10%.

However, there are still opportunities to test the integration of ANN with other popular swarm-based algorithms, such as ABC. Such integration was done in other applications, for example, to optimize the hyperparameters of ANN as presented in (Rashid, & Abdullah, 2018). They integrated ABC, genetic algorithm, and back propagation neural network that is used to classify and diagnose diabetes. Adding a genetic operator helps to avoid sucking in local optima, an issue mentioned in (Packianather et al., 2014).

In addition, (Bullinaria & AlYahya, 2014) made a comparison between training ANN with the back propagation and ABC, they found that back propagation is significantly better than ABC. (Xu et al., 2019) applied a modified ABC that has better performance in utilizing the neighbour information in order to accelerate the convergence, this new algorithm was used to train ANN. (Qolomany et al., 2017) optimized two variables of the DL model which are the number of hidden layers and the number of neurons in each layer using Particle Swarm Optimization (PSO). Furthermore, (Badem et al., 2017) applied BA along with limited memory Broyden–Fletcher–Goldfarb–Shannon to train autoencoder network while (Lee et al., 2018) optimized the hyperparameters of CNN using free harmony search technique.

In addition, (Zeybek et al., 2021) presented a novel metaheuristic algorithm that trains deep RNN using an enhanced Ternary Bees Algorithm (BA-3+) for the sentiment classification task. BA-3+ algorithm finds the optimal set of parameters for deep RNN architecture by collaborative search of three bees, the authors found that it outperformed other optimization algorithms such as Stochastic Gradient Descent (SGD), Differential Evolution (DE) and PSO. Training deep RNN using the BA-3+ algorithm achieved an accuracy rate between 80%-90%; while training it using SGD produced an accuracy between 50%-60% for most datasets.

Furthermore, the GA is one of the nature inspired algorithms used to find the optimal parameters in the LSTM network for predictive maintenance (Kim & Choi, 2021), the authors optimized the time steps, the number of LSTM layers, and the number of hidden neurons in each layer using GA. They suggested a procedure that starts with population initialization followed by fitness computation for chromosomes, then genetic operators application for new population creation if needed. The design of GA is based on chromosome structure, fitness function, crossover operator, mutation operator, and population updating and termination (Kim & Choi, 2021). The proposed model achieved an accuracy of 98.14%. Another study used GA to optimize five parameters related to LSTM hidden layer size, number of hidden layers, batch size, number of times steps and number of epochs. The hybrid GA-LSTM is used to predict the next word in the sentence which achieved an accuracy of 56% (Gorgolis et al., 2019).

Furthermore, PSO which is a swarm-based metaheuristic optimization algorithm is applied to improve the performance of the LSTM network by optimizing hidden layers, number of neurons, activation function, loss function, optimizer, batch size, and number of epochs. The hybrid PSO-LSTM is applied to predict the pollution level based on the weather dataset; it achieved a lower RMSE than the original LSTM by a value of 0.0007 (Pranolo et al., 2022).

A study used ABC to optimize the weights of ANN (Qureshi et al., 2019). The author used the hybrid algorithm to propose a new intrusion detection system achieving an accuracy of 95.02% (Qureshi et al., 2019). Another study used BA to train RNN for sentiment classification which improved the accuracy from 60% for traditional RNN to 90% for the BA-RNN algorithm (Zeybek et al., 2021). Also, ABC was used to optimize LSTM parameters (window size, LSTM units, dropout probability, number of epochs, batch size and global learning rate) (Kumar et al., 2022). They used the hybrid ABC-LSTM algorithm for stock market prediction which achieved a lower RMSE by 5.6836 (Kumar et al., 2022).

2.4.2 Gap and Open Issue

Looking at the literature, there is a lack of hybridization between BA and CNN which is an important gap as BA is one of the most popular swarm-based optimization techniques that use a global search, followed by a local search and an intense search in order to find the optimal parameters that yield the minimum error.

In addition, there is a lack of optimizing the learning rate adjustment factor for convolutional layers in CNN and each gate in the LSTM network which is an important gap as the learning rate control the weight update. Having a more optimum learning rate means more optimum updates for the network weight (Brownlee, 2020). The CNN gap will be addressed in the following chapter by proposing two novel hybrid algorithms that take the advantage of global, local and intense searches in the BA in order to optimize the parameters and train CNN.

2.5 Machine Learning in Manufacturing

(Li et al., 2017) stated that AI is an essential concept used to facilitate the development of intelligent systems to increase the productivity and maximize the efficiency of various processes such as in the manufacturing context. The life cycle of new intelligent

manufacturing systems uses autonomous sensing, learning, interconnection and decision making to integrate and optimize different aspects of manufacturing enterprise leading to increase the productivity, maximize the efficiency, improve the quality, and reduce the cost. (Wuest et al., 2016) stated that the applications of machine learning in manufacturing include machine condition monitoring, fault diagnosis leading to applying predictive maintenance, image recognition that helps to classify damaged products and building a simulator for the advanced manufacturing process to predict its parameters.

In addition, (De Filippis et al., 2017) showed that manufacturing applications include modelling and scheduling the processes which solve issues related to operational decision making. The procedure for using ANN starts by collecting experimental observations and pre-processing them to be ready for network training. Then, establishing a numerical relationship between the parameters and mechanical features of the part. Finally, the time and cost parameters of the process will be evaluated to identify the benefits and costs incurred from the prediction model. Multilayer perceptron can be used for process modelling and product quality prediction for manufacturing processes such as injection moulding and arc welding processes (De Filippis et al., 2017).

(Saric et al., 2013) simulated and predicted steel surface roughness using three different neural network algorithms which are back-propagation, modular and radial basis function neural networks. The input variables were the parameters that control the process (feed rate, depth of cut, cutting speed, and the number of revolutions) while the output variable is machined surface roughness. The results show that Root Mean Square Error (RMSE) in the radial basis function is 5.24 % in the learning phase and 8.53 % in the validation phase, the modular neural network produced an error of 6.02 % in the learning phase and 8.87 % in the validation phase. Finally, the results of the back-propagation neural network in terms of RMSE is 6.46 % in the learning phase and 7.75 % in the validation phase. Furthermore, (Moyne & Iskandar, 2017) applied big data analytics in the semiconductor manufacturing industry which helps to improve the current capabilities such as detecting faults and supporting the recent capabilities like predictive maintenance. The most important factor was the quality of the dataset in order to deliver high quality solutions. In the future, they expected that digital twin will be used to improve the ability of the analytics.

2.6 Additive Manufacturing Overview

Additive manufacturing (AM) is defined in ISO/ASTM 52900:2021 standard (ISO/ASTM 52900:2021) as a process that builds parts by joining a material layer by layer, using 3D model data. The standard defined seven process categories for AM namely binder jetting, directed energy deposition, material extrusion, material jetting, powder bed fusion, sheet lamination, and vat photopolymerization. AM is also referred to using other terms such as 3D printing, rapid prototyping, digital manufacturing, and layered manufacturing. AM started in the 1980s when Dr. Hideo Kodama used 3D scanning knowledge and 3D topographical maps layering patterns to create a prototyping machine (www.TriMech.com). Then, in 1987 stereolithography process was invented by Chuck Hull, the long processing time of production led to this patent that creates 3D objects using Ultraviolet laser (Office of Energy Efficiency and Renewable Energy, 2017). Fused Deposition Modelling (FDM) was invented in 1991 by Scott Crump. In 1995 German scientists invented Selective Laser Melting (SLM) (Ratna, 2022), and in the 2000s more companies were interested in 3D printing benefits and capabilities (Markforged).

There are two levels for AM processes, the first one is the digital level where the CAD model is prepared, converted to a stl.file, and G-code is generated. The second level is the physical level which contains part manufacturing using one of the seven processes mentioned previously (Lastra et al.,2022). Initially, AM or 3D printing technologies were used mainly for rapid prototyping, but with improvements in the reliability and efficiency of AM processes as well as in the material properties of the components produced, they are increasingly used in more advanced applications such as to create highly customized products, producing a small volume of serial components and visualizing tool in design. Future applications may concern human organ creation, clothes manufacturing, and food confection (Farinia Group, 2018).

(Abdulhameed et al., 2019) stated that there are certainly many advantages associated with the use of AM processes such as the flexibility in the producible designs, the facilitated customization of products and the capability to print highly complex structures. However, many challenges and drawbacks remain. More specifically, some major disadvantages are void formation between the sequential layer due to reduced binding, the appearance of stair

stepping effect in the fabricated part, the variation in microstructure and mechanical properties, the small build volume that leads to scaling down large parts or cutting them to subparts consuming effort and long time, complying with safety standards for fabricated food and medical devices, and finally fabricating and producing parts that can be used for criminal purposes such as drugs and weapons as it can deal with complex structures as mentioned previously.

The typical applications for AM processes are prototyping during the development phase of a product, producing parts in pilot series production or short series where costs related to casting or injection moulding are high and producing parts with complex geometry which cannot be done by other manufacturing means (www.MetalAM.com). This is the main advantage of using AM when compared to other manufacturing processes, for example, it allows the integration of additional functionality in components such as the production of repeating internal patterns. Thus, AM is capable of combining complex internal structures with more regular outer geometries, reducing weight while keeping structural and aesthetic integrity. (Bauer, 2021) mentioned that there are many applications of the SLM process in real life, such as in the medical field with dental implants (highly complex, small in dimension, and one-off patient products) or with specialized surgery equipment. Another large field is in the aerospace industry where hard to machine materials, such as nickel-based super alloy, require high quality, complex shapes, and low to medium production lots like a hydraulic manifold. Also, it can be used in the manufacturing of a bike frame. Furthermore, applications are emerging in the context of the highly regulated aerospace and automotive sector. For instance, metal parts suitable for aircraft were directly manufactured using titanium material, reducing the lead time by 30% - 70%, non-recurring fabrication costs by 45% and abatement in manufacturing cost for parts with low volume by 30% - 35% (Abdulhameed et al., 2019).

2.7 Additive Manufacturing Processes

As shown in table 2.3, seven AM processes can be used for the production of complex parts:

Table 2.3: Additive Manufacturing Processes (Farinia Group, 2018)

Process	Techniques	Materials
Powder bed fusion	Direct Metal Laser Sintering (DMLS), Selective Laser Sintering (SLS), Selective Laser Melting (SLM), and Electron Beam Melting (EBM)	Polymers, metals: miraging steel, stainless steel 316L, 15-5PH, nickel-based superalloys: Inconel 718, Inconel 625, Hastelloy X, titanium TA6V, chrome-cobalt, and aluminium AISi10mg
Material extrusion	Fused Deposition Modelling (FDM)	Thermoplastic filament
Material Jetting	Material jetting	Photopolymers and wax-like materials
Binder jetting	PolyJet (PJ)	Photopolymers and wax-like materials
Sheet lamination	Laminated Object Manufacturing (LOM), Paper Lamination Technology (PLT), and Ultrasonic Additive Manufacturing (UAM)	Adhesive-coated papers, metal tapes and foils, and plastic sheet material
Vat photopolymerization	Stereolithography (SLA) and Digital Light Processing (DLP)	Light curable resin
Directed energy deposition	Laser Engineered Lens Shaping (LENS), Direct Metal Deposition (DMD), Laser Metal Deposition (LMD), blown powder, and laser cladding	Metals such as nickel-based alloys and aluminium

(Lastra et al.,2022) presented AM technologies classification based on material according to ISO/ASTM 52900 as shown in figure 2.4 below:

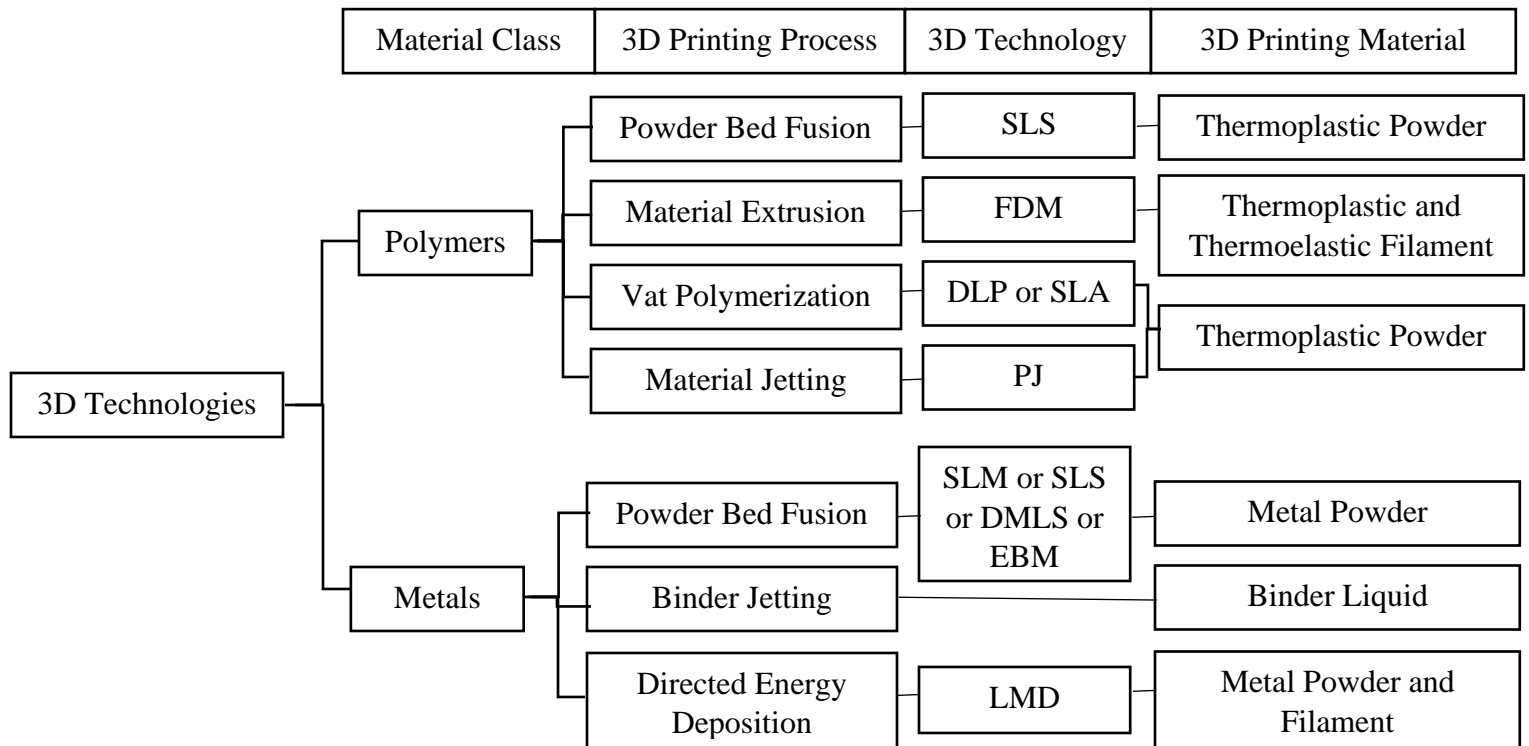


Figure 2.4: AM Technologies Classification According to ISO/ASTM 52900 (Lastra et al.,2022)

2.8 Powder Bed Fusion

Powder bed fusion (PBF) is one of the most promising AM processes due to its ability to process a wide range of hard metals. This section presents a review of the PBF process, showing the way of working, thermodynamical phenomena, parameters, open issues, three porosity types, and state of the art studies about adopting DL techniques to improve the performance of SLM process.

2.8.1 Definition and Way of Working

(Sun et al., 2017) stated that PBF processes are laser-based additive manufacturing in which the laser beam scans selected locations of a powder bed at a controlled speed and then fuses the powder to obtain solid material layer by layer by either partial melting, such as in SLS or full melting, such as in the metal-based SLM process as mentioned in (Shrestha, et al., 2019). Figure 2.5 presented in (Sun et al., 2017) illustrates the laser-based PBF process.

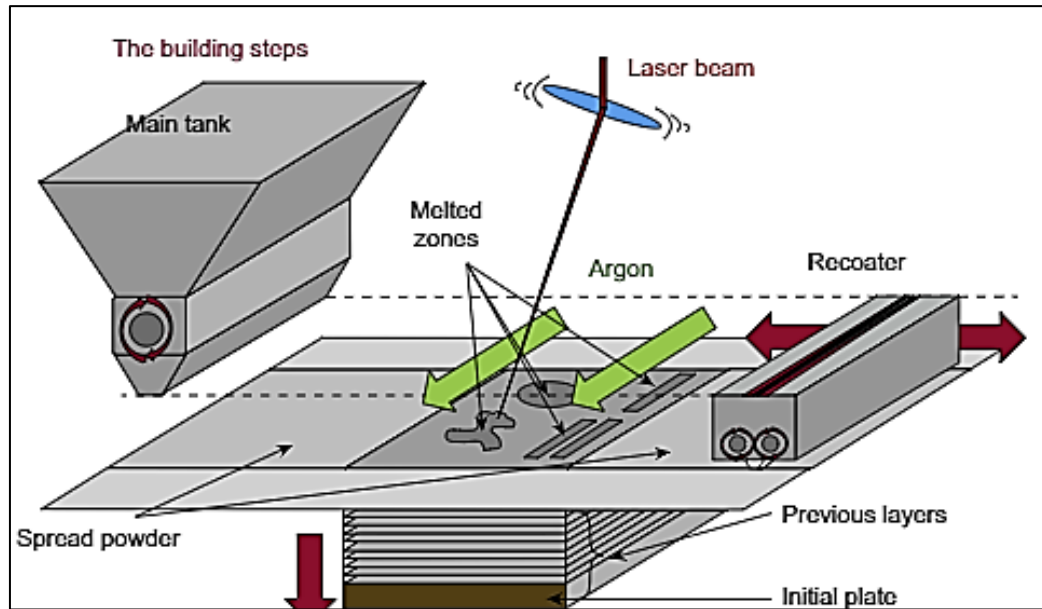


Figure 2.5: Elements of Powder Bed Fusion Process (Sun et al., 2017)

(Bauer, 2021) stated that the first crucial part of the SLM process is the build job preparation for any given 3D model. It consists of four main steps: geometry importation, alignment/orientation within a build envelope (critical as surfaces with an angle less than 45° need to be supported), support creation to allow stable processing conditions and finally slicing based on machine specific requirements. After preparing and loading a build job onto a machine, the manufacturing process can be initiated starting with the alignment of the recoating device or levelling of the substrate to enable the deposition of powder. Then, the laser scans a specific area based on a predefined scanning strategy and controllable parameters to selectively solidify the material. After finishing a layer scanning, the build plate is lowered, and new powder is deposited on top of the build area to produce a new layer. The following figure 2.6 illustrates the influential factors of a typical SLM process:

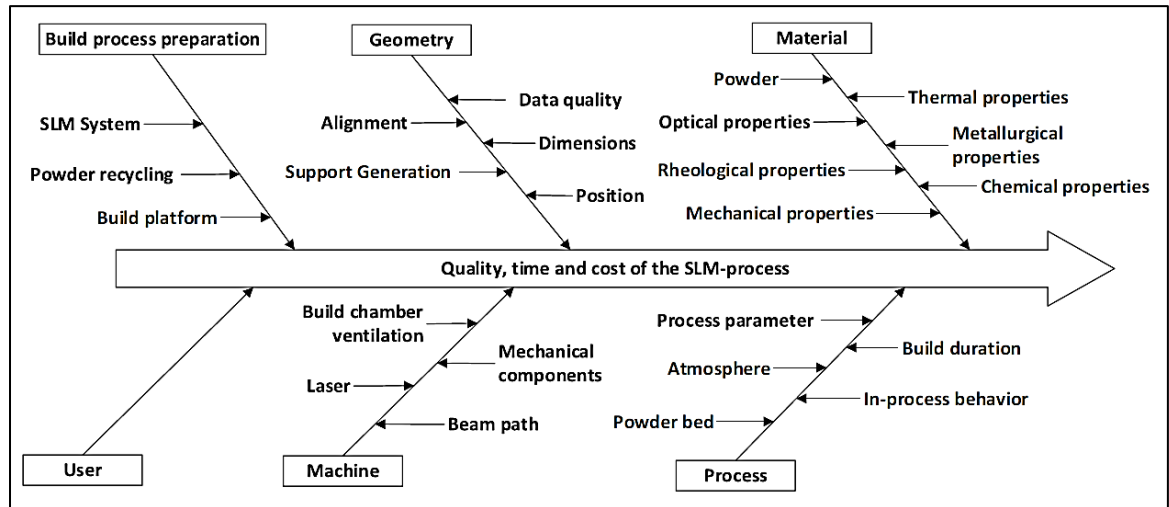


Figure 2.6: Ishikawa Diagram of Influential Factors in the SLM Process (Bauer, 2021)

2.8.2 Thermodynamical Phenomena

The PBF process involves complicated thermodynamical phenomena and physiochemical behaviour as powder particles are melted using a laser beam with high energy. Thermodynamic monitoring is important to control the performance of the process, but it is difficult as the molten pool has a small size and moves quickly. Recently, numerical modelling has been developed to study the physical mechanism deeply. Metal SLM process involves principles for multiscale coordinate control, they include deformation and stress (macroscale), melting behaviour and laser absorption (mesoscale), and the development of microstructure (microscale) (Gu et al., 2017).

The limited energy on the powder bed and low operating temperature resulting from low laser power value in SLM lead to generating residual pores between the neighbouring small molten pools. The low temperature decreased the liquid surface tension leading to the melt flow (Gu et al., 2017). These conditions cause a remarkable reduction of the melt pool convection, and also simultaneously they lead to a weakening of the melt migration between the current and solidified neighbouring tracks. As a result, the porosity is shown obviously on the top surface and cross-section of the SLM part. Conversely, increasing the laser power results in a larger molten pool size with a longer lifetime of the liquid. The considerable laser energy input leads to an intensified convection within the melt pool with sufficient melt migrations between the high laser power tracks which produce high-quality SLM parts without distinct defects on both the cross-section and top surface. In addition,

cellular morphology is presented with the microstructure surrounding the tracks with no obvious defects. The following two figures 2.7 and 2.8 for the top surface morphology of the produced Inconel 718 part with two different laser power 90 W and 120 W are presented in (Gu et al., 2017):

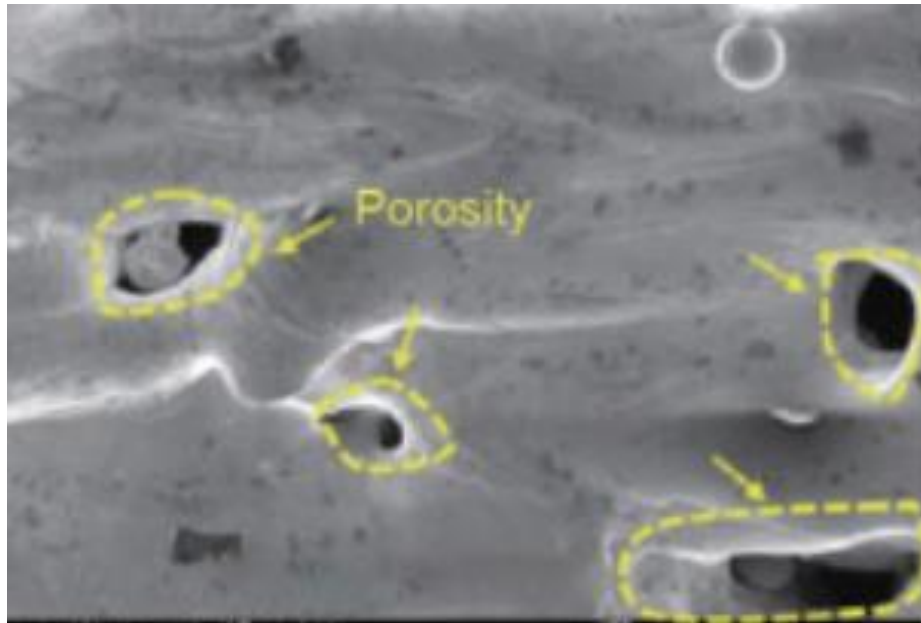


Figure 2.7: Top Surface Morphology with Laser Power of 90 W (Gu et al., 2017)

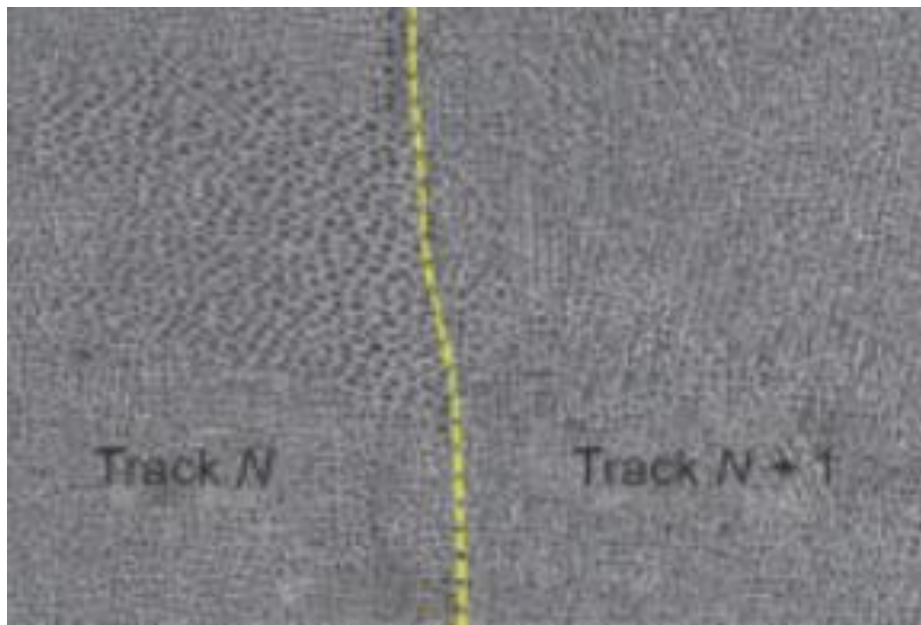


Figure 2.8: High Magnitude Microstructure Morphology of Top Surface with Laser Power of 120 W (Gu et al., 2017)

So, with a low laser power value of 90 W, irregularly shaped porosity was shown distinctively in the tracks where heat transfer and limited mass occurred. Increasing the laser power value to 120 W leads to decreasing porosity. So, producing high-quality Inconel 718 parts are achievable with optimal laser power that can be specified using mesoscale simulation and analysis (Gu et al., 2017).

2.8.3 Parameters

(Sun et al., 2017) mentioned that the parameters for PBF processes can be divided into four categories, the first one is the laser-based set of parameters that includes laser power, wavelength, spot size, pulse duration, and pulse frequency. The second category includes parameters related to the scanning strategy (scanning speed, scanning spacing, scanning patterns, and layer thickness). The third category is the set of powder-related parameters which include particle size and distribution, particle shape, powder bed density, layer thickness and material properties. The last category is the set of temperature-related parameters, including powder bed temperature, powder feeder temperature and temperature uniformity.

2.8.4 Open Issues

(Shrestha et al., 2019) mentioned that metal-based AM has many issues such as porosity, part deformation and cracks, but as mentioned previously porosity is one of the most challenging issue due to its effect on the mechanical properties, structural integrity, strength and Young's modulus of the produced material. In addition, producing a large amount of experimental data is not cost-effective for SLM parts (www.facfox.com, 2022) since there are many types of production costs for pre-processing, processing and post-processing cost including preparing geometry data, CAD model, machine setup, material cost, building up the part, and postprocessing cost (Rickenbacher, 2013).

Furthermore, when using CT scans of sample parts, there is a problem related to assessing accurately the porosity in SLM parts. One main drawback is when using gray value analysis to assess the porosity of SLM parts visible in CT scan slices. The difficulty is the subjectivity in selecting an appropriate grayscale threshold that would convert a single slice into binary images highlighting defective regions, as well as determining the true level of porosity. When an inappropriately low grayscale threshold is applied to the

original slice image for binary image conversion, a certain amount of tiny undesired white spots are not filtered. However, if a higher grayscale threshold is adopted, the morphological features of the defective area, specifically near the boundary, are altered dramatically. These thresholds would result in significantly different predictions of porosity levels (Gong et al., 2019). Hence an intelligent method is needed for assessing and predicting the porosity level in SLM parts.

There are three main types of pores, namely gas, keyholes and lack of fusion porosity (Snell et al., 2020). The following subsections will explain each type showing its characteristics and mechanism of formation.

2.8.5 Gas Porosity

Gas pores belong to the most common type of pore. They are the most spherical and the smallest type. It is characterized by smooth edges and a wide range of sizes from submicron to several microns (Tan et al., 2020). The following figure 2.9 shows an illustrative example of the gas porosity using the scanning electron microscope (SEM):

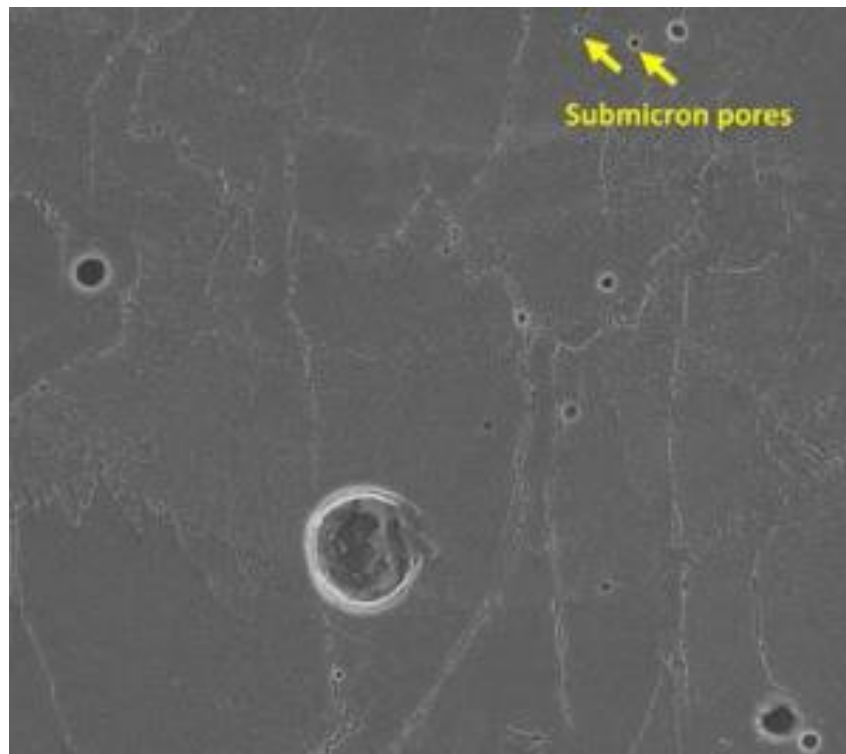


Figure 2.9: Illustrative Example of Gas Porosity using SEM (Tan et al., 2020)

Gas pores are connected to trapped gas that might have different sources of origin whether during or before the process starts such as the entrapped gas in the powder of the alloy during the process of gas atomization. However, increasing the laser power or decreasing the scanning speed might enlarge this type of pore. (Tan et al., 2020) fabricated SLMed 2024 Al alloy samples at a specific scanning speed of 1200 mm/s and with laser power values between 225 W to 375 W to investigate the gas porosity evolution. They found that increasing the laser power from 225 W to 300 W leads to raising the fraction of gas pore from 1.3% to 1.6%, but increasing the laser power further from 300 W to 375 W reduced the fraction to 0.7% as the thermal gradient between the centre and boundary of melt pool is increased which leads to facilitating the process of outgassing. So, it was concluded that it is difficult to fully understand the effect of SLM process parameters on gas porosity formation since it is a dynamic process that involves pore nucleation, growth, and outgassing during rapid solidification (Tan et al., 2020).

2.8.6 Keyhole Porosity

Keyhole pores arise from energy input excess with high energy density (Tan et al., 2020), which is a result of increasing the laser power or decreasing scanning speed. It is characterized by large-size cavities reaching hundreds of microns and tended to be near-spherical in shape, so sometimes it is difficult to distinguish between large gas pores and keyhole pores. The following two figures 2.10 and 2.11 show the 3D view of the keyhole pore at a laser power value of 195 W and scanning speed value of 400 mm/s and different shapes of 2D images:

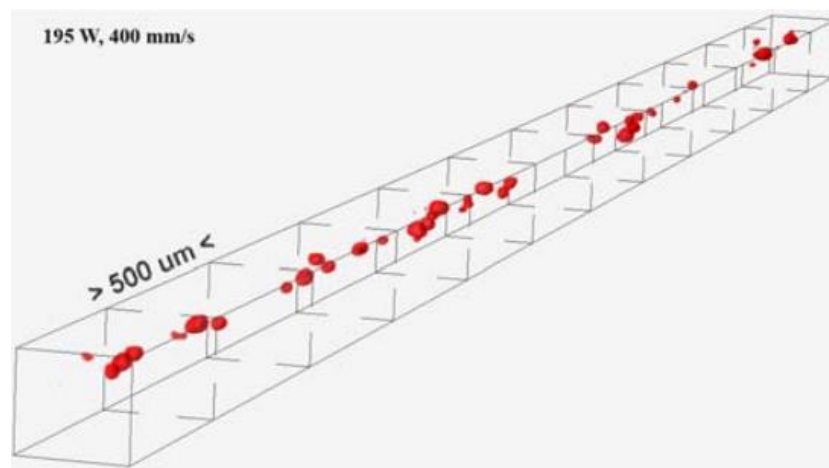


Figure 2.10: 3D View of Keyhole Pores (Shrestha et al., 2019)

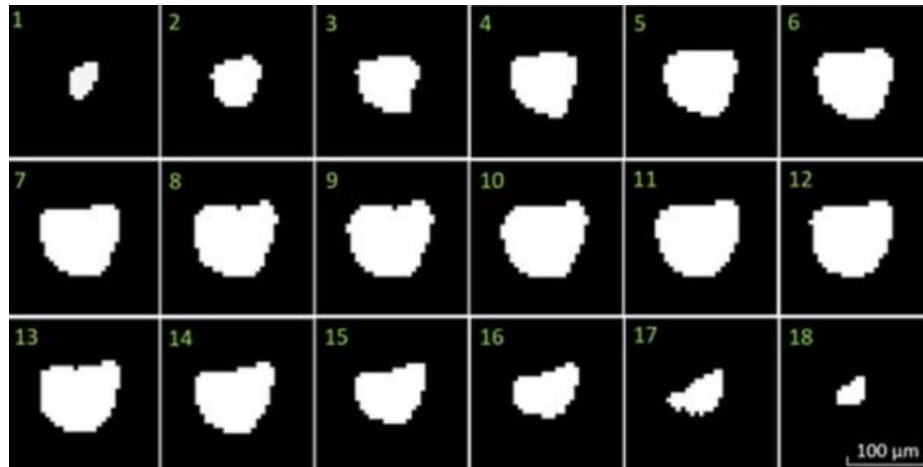


Figure 2.11: 2D Images of Different Keyhole Pores Morphologies (Shrestha et al., 2019)

Unlike gas pores, the surface and contour are bumpy, these cavities are the most common type of keyhole porosity that appear in the SLM part (Tan et al., 2020). It is caused when trapping bubbles of vapour within the melt pool (Shrestha et al., 2019), changing the melting mode from conduction to keyhole mode at high laser energy intensity leads to the formation of this type of pore. In the conduction mode, the laser energy input is low to moderate which mediates the heat transfer through conduction. The material melting happens in this mode without vaporization which results in less porosity, but in keyhole mode, the high laser intensity causes metal evaporation which generates recoil pressure at the bottom of the melt pool. The instability of the dynamic behaviour in the melt pool leads to the formation of keyhole porosity with near-spherical morphology at the bottom of the melt pool, which acts as a stress concentrator causing material property deterioration (Tan et al., 2020).

2.8.7 Lack of Fusion Porosity

Lack of fusion pores have irregular voids morphology that commonly appeared in SLM alloy, they are featured with large sizes reaching hundreds of micrometres as shown in the following figure 2.12 (Tan et al., 2020):

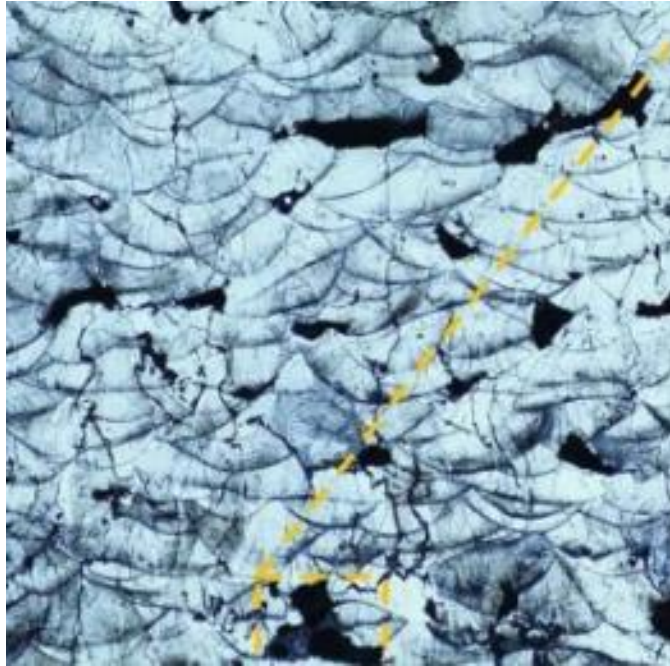


Figure 2.12: Lack of Fusion Pores Morphology (Tan et al., 2020)

Irregular voids are formed because of un-melting in some regions caused by the reduction in energy density, which is generally a result of the low value of laser power or fast scanning speed (Tan et al., 2020). The energy density is a function consisting of laser power, scanning speed, layer thickness, and hatch distance, these parameters have a significant effect on pores formation (Shrestha et al., 2019). Having insufficient laser energy to induce the overlapping between the adjacent layers leads to the formation of this type of pore. Also, the high reflectivity and dense oxide presence on the powder particles' surface might form such defects as the laser penetrability is lowered (Tan et al., 2020).

Thus, when optimizing the SLM process it is essential to accurately study the pore formation occurring when using various combinations of parameters.

2.8.8 State of the Art Studies

AM products need advanced quality control processes to achieve the desired reliability, so it is valuable to review quality-related research in AM context by focusing on issues and considerations that need development and optimization, in addition to future directions toward improving the quality of AM products. This section shows the new trend of using cameras and sensors to acquire real-time data for quality monitoring purposes in addition

to showing advanced data analytics and predictive algorithms to optimize the process parameters.

(Kim et al., 2018) mentioned that most of the powder bed fusion processes melt powder-based material using a high source of thermal energy, it might be partial melting as in SLS, full melting as in SLM or using EBM. Detecting porosities has a significant contribution toward improving the quality of products.

(Kim et al., 2018) showed a study that developed a method for continuous data capturing using an infrared (IR) camera to detect porosities inside materials. In addition, they showed another study that developed an automatic feedback control system that stops the printing process at a certain porosity level, they used an IR camera as well to capture images and applied image processing techniques to optimize the process parameters.

(Zhang et al., 2019) monitored the porosity during the process of laser AM, they used a deep learning-based method namely CNN to predict the porosity based on melt pool cross-section images acquired by the coaxial camera. The model achieved an accuracy of 91.2% in detecting porosity occurrence. Similarly, (Coeck et al., 2019) predicted a lack of fusion porosity based on melt pool data collected during the SLM process, they used a DMP monitoring system to monitor the melt pool during the processing with titanium alloy material, and then they created an algorithm that correlates the porosity with melt pool monitoring data which achieved prediction sensitivity of 90% for lack of fusion events with pores volume grater than 160 μm .

Recently, the Deep Belief Network (DBN) has been used to analyse plasma acoustic signals generated at the surface of the powder bed specifically in the SLM process. The signals are collected using a microphone, and then DBN is used to recognize the conditions of the melt track (slight balling, balling, normal, slight overheating, and overheating) (Wang et al., 2020). In addition, they used acoustic signals in porosity classification, particularly in achieving the balance between classification accuracy and spatial resolution in porosity detection. High sensitivity fibre Bragg grating sensor was adopted to collect acoustic signals for airborne, the time span for each running window was 160 ms and 300 patterns represented each of the three porosity levels without overlapping between the

training and testing set in terms of the running window. The parts with different porosity levels were classified using spectral CNN with a classification accuracy between 83-89%.

2.9 Smart Manufacturing Technologies

There are six smart manufacturing technologies (Autodesk) that can be used to improve the product performance and maximize the operational efficiency as shown in the following table 2.4:

Table 2.4: Smart Manufacturing Technologies (Autodesk)

Technology	Techniques	Benefits
Manufacturing-Led Design	Computer-Aided Design & Computer-Aided Manufacturing (CAD/CAM)	Design and manufacture prototypes
3D Printing (AM)	Powder Bed Fusion, Material Extrusion, Material Jetting, Binder Jetting, Sheet Lamination, Vat Photopolymerization, and Directed Energy Deposition	Create highly complex geometry products
CNC Machining and Probing	Roughing, 2 Axis Vs 3 Axis Milling and Turing	Control the movement cutting tools through generated codes from computer software
Hybrid Manufacturing	Netfabb (Additive) and Power Mill (Subtractive)	Create high quality products with less material waste
Simulation	Finite Element Analysis, Computational Fluid Dynamics, Plastic Injection Molding, and Generative Design	Predict the product performance and optimize the product design
Robot Automation	Autonomous 3D Printing Robots	Automate large-scale additive and subtractive processes

2.10 Summary

This chapter presented a review of DL in general and of CNN and LSTM in particular in order to find gaps and open issues that need to be fulfilled in the future to improve the performance of DL models. It was found that the most challenging aspect in training DL is designing a better topology where the traditional heuristic approach of using trial and error will result in a less accurate model depending on user experience. In this case, applying optimization techniques such as nature inspired algorithms to optimize the parameters of DL algorithms can improve the performance of the model. However, designing a better DL topology is still an open issue and no approach has been found yet that can give the best DL topology.

In addition, this chapter presented the AM processes, applications, advantages, and gaps and open issues that need to be addressed. It focused on the powder bed fusion process showing the way of working, thermodynamical phenomena, parameters, open issues, three porosity types, and state of the art studies about adopting DL techniques to improve the performance of SLM process. The review found an important gap in using CT scans of sample parts, there is a problem related to assessing accurately the porosity in SLM parts. One main drawback is when using gray value analysis to assess the porosity of SLM parts visible in CT scan slices. The difficulty is the subjectivity in selecting an appropriate grayscale threshold that would convert a single slice into a binary image highlighting defective regions, as well as determining the true level of porosity. When an inappropriately low grayscale threshold is applied to the original slice image for binary image conversion, a certain amount of tiny undesired white spots are not filtered. However, if a higher grayscale threshold is adopted, the morphological features of the defective area, specifically near the boundary, are altered dramatically. These thresholds would result in significantly different predictions of porosity levels. Hence an intelligent method is needed for assessing and predicting the porosity level in SLM parts.

**Chapter 3: Convolutional Neural Network
Parameters Optimization using Nature Inspired
Algorithms**

3.1 Proposed Novel Hybrid BA-CNN & BA-BO-CNN Algorithms

Designing a better CNN topology is an ongoing issue. This chapter addresses this issue by proposing a novel approach for optimizing the parameters of CNN through the BA and BO.

3.1.1 Design of Experiments

First, the design of experiments (DOE) technique is conducted to investigate the significant factors affecting CNN validation accuracy (VA). Four factors and three levels are used in designing Taguchi orthogonal array (L9). The four variables which are selected based on the topology of the CNN and its learning capability as mentioned in (MathWorks-3). It stated the benefit of CNN parameters that will be optimized using BO in the hybrid BO-CNN and BA-BO-CNN algorithms and using BA in the hybrid BA-CNN algorithm which are shown in the following table 3.1:

Table 3.1: CNN Optimization Variables Information

Variable	Benefit	Range	Type
Section Depth	Control the depth of the network	1 - 3	Integer
Initial Learning Rate	Allow for learning the features	1e-2 - 1	Logarithm
Momentum	Update the hyperparameters	0.8 - 0.98	Logarithm
Regularization	Prevent overfitting	1e-10 – 1e-2	Logarithm

The four factors with three levels are used in designing Taguchi orthogonal array (L9) (Fralely et al., 2020) as shown in table 3.2:

Table 3.2: Factors and Levels Definition

Factors	Level 1	Level 2	Level 3
Section Depth	1	2	3
Initial Learning Rate	0.01	0.055	0.1
Momentum	0.88	0.9	0.92
Regularization	1e-10	2.5e-4	0.0005

The nine experiments are applying the original CNN without hybridization to nine different combinations of parameters on the ‘Cifar10DataDir’ benchmark image data (Machine Learning Repository) that consists of 10 classes airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each class has 6,000 images so, the total sample size is 60,0000 images, the results of VA are shown in the following table 3.3:

Table 3.3: Taguchi Orthogonal Array for CNN Parameters (L9) (Fralely et al., 2020)

Experiment #	Section Depth	Initial Learning Rate	Momentum	Regularization	VA (%)
1	1	0.01	0.88	1e-10	67.96
2	1	0.055	0.9	2.5e-4	72.20
3	1	0.1	0.92	0.0005	70.90
4	2	0.01	0.9	0.0005	76.40
5	2	0.055	0.92	1e-10	78.44
6	2	0.1	0.88	2.5e-4	77.40
7	3	0.01	0.92	2.5e-4	77.74
8	3	0.055	0.88	0.0005	79.56
9	3	0.1	0.9	1e-10	80.34

After Running the experiments and recording VA for each run, analysis of variance (ANOVA) is applied using Minitab software in order to investigate the most influential factors affecting VA, the ANOVA results are shown in table 3.4:

Table 3.4: ANOVA Results

Source	Degree of Freedom	Sum of Squares	Mean Square	F-value	P-Value
Section Depth	2	131.58	65.79	*	*
Initial Learning Rate	2	12.31	6.15	*	*
Momentum	2	2.69	1.34	*	*
Regularization	2	0.06	0.03	*	*
Error	0	*	*		
Total	8	146.66			

The regression equation is shown in figure 3.1:

Coefficients					
Term	Coef	SE	T-Value	P-Value	VIF
Constant	75.66	*	*	*	*
Section Depth					
1	-5.307	*	*	*	1.33
2	1.753	*	*	*	1.33
Initial Learning Rate					
0.010	-1.627	*	*	*	1.33
0.055	1.073	*	*	*	1.33
Momentum					
0.88	-0.6867	*	*	*	1.33
0.90	0.6533	*	*	*	1.33
Regularization					
0.0000000	-0.08000	*	*	*	1.33
0.0002500	0.1200	*	*	*	1.33
Regression Equation					
VA (%) = 75.66 - 5.307 Section Depth_1 + 1.753 Section Depth_2 + 3.553 Section Depth_3 - 1.627 Initial Learning Rate_0.010 + 1.073 Initial Learning Rate_0.055 + 0.5533 Initial Learning Rate_0.100 - 0.6867 Momentum_0.88 + 0.6533 Momentum_0.90 + 0.03333 Momentum_0.92 - 0.08000 Regularization_0.0000000 + 0.1200 Regularization_0.0002500 - 0.04000 Regularization_0.0005000					

Figure 3.1: Regression Equation

Looking at the results, it is revealed that all four factors have a significant effect on VA, so they will be optimized using the BA and BO in order to minimize the classification error on the validation set, which is the objective function. It is a complex function of section depth, initial learning rate, momentum, and regularization, it is complicated to be formulated.

3.1.2 Bayesian Optimization Method

BO algorithm builds the probability model of the objective function to be used to select the hyperparameters of the network and then evaluate them in the true objective function. It maintains a Gaussian process model as a surrogate model in the objective function that uses network variables as inputs to specify the network architecture, (Nag, 2021) mentioned that initial runs of the function are used as starting points “prior” and then they are enriched with “posterior” data points in each iteration, the acquisition function is used as a metric function to decide the optimal parameters, the formula of expected improvement (EI) function is as follow (Nag, 2021):

$$EI(x) = [\mu (f(x)) - \max \{ f(x) \}] \times \phi(z) - \sigma (f(x)) \times \phi(z) \quad \text{if } \sigma(f(x)) > 0 \quad \text{(Equation 3.1)}$$

$$EI(x) = 0 \quad \text{if } \sigma(f(x)) \leq 0 \quad \text{(Equation 3.2)}$$

Where:

$$Z = \mu(f(x)) - \max\{f(x)\} / \sigma(f(x)) \quad \text{(Equation 3.3)}$$

$\mu(f(x))$ is the prediction from the gaussian process for new data point x

$\max\{f(x)\}$ is the maximum prediction from the entire list of prior at the current stage

$\Phi(z)$ is the standard normal cumulative probability density

$\sigma(f(x))$ is the standard deviation of the prediction for new data point x

$\phi(z)$ is the standard normal probability density

The improvement is checked by the difference between the prediction at the current stage and the maximum prediction of the entire list of prior. If the difference is high, it means that the new prediction is significantly higher than the maximum obtained so far. Multiplying the difference by the standard normal cumulative probability density gives overall or expected mean improvement which is the exploitation part. The standard deviation gives the uncertainty and it is the secret of the exploration part. Applying BO in conjunction with Stochastic Gradient Descent Momentum (SGDM) as one of the training options in CNN architecture, momentum adds inertia that helps the current update to make a proportional contribution to the previous iteration update.

3.1.3 The Bees Algorithm

According to (Al-Musawi, 2019), BA is one of the most important swarm-based optimization techniques that performs an intense search after a local search to optimize the variables, so combining it with the BO technique will contribute to optimize CNN parameters that minimize the classification error on the validation set. Its way of working is inspired by honeybees' foraging behaviour. The algorithm requires setting the following parameters:

- Number of scout bees: n
- Number of selected bees: m
- Number of elite bees: e
- Number of recruited bees for elite (e) sites: nep

- Number of recruited bees for other best (m-e) sites: nsp
- Neighbourhood size for each selected patch (local search): ngh

The algorithm starts with the global search when the scout bees (n) arrive at random positions and evaluate them based on the fitness value. Then, a local search stage starts by selecting the best sites (m) and abandoning the remaining sites. After that, an intense search is initiated by selecting elite sites (e), which are the best among the best sites. The next step is selecting the size of the neighbourhood search space in order to recruit more bees for the elite sites (e) and fewer bees for non-elite sites (m-e) to conduct a local search. The global and local searches will be performed simultaneously, while the recruited bees are exploring the best solutions around the neighbourhood, the global search on the remaining sites is carried out randomly. This iterative process is stopped by one of the following conditions:

- The optimal solution found
- The iteration number exceeded
- No improvement over a specific sequential number of iterations

(Koc, 2010) described the basic flowchart for BA as shown in the following figure 3.2:

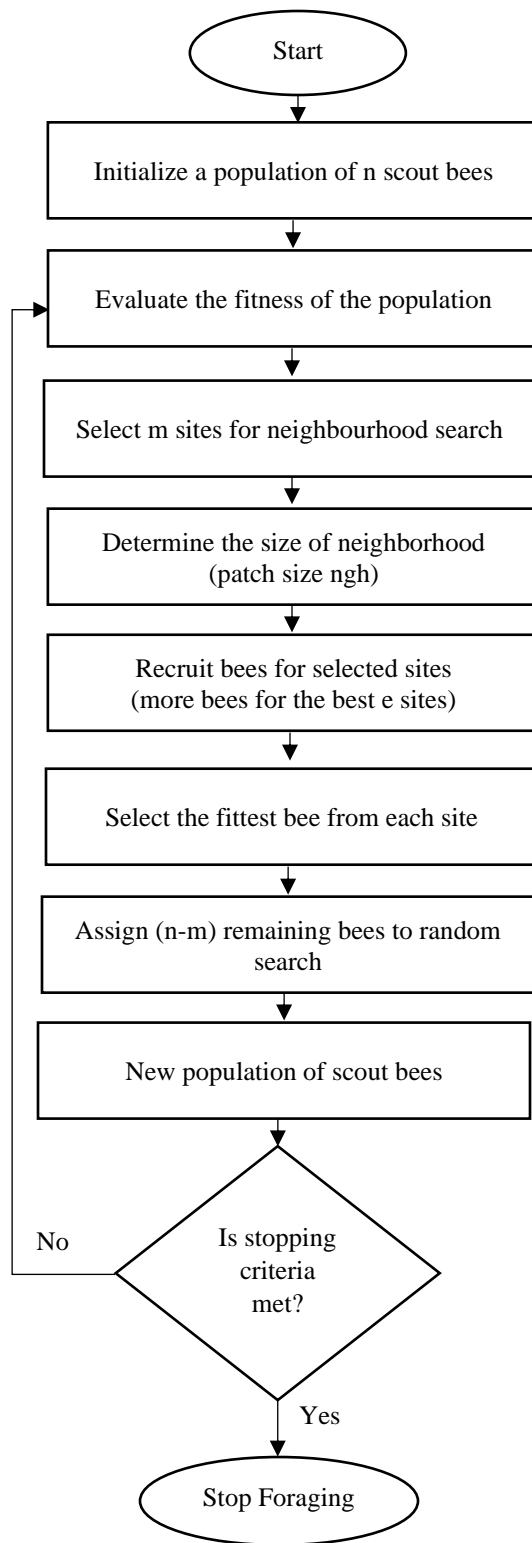


Figure 3.2: The Basic Flowchart for the BA (Koc, 2010)

In addition to the basic BA, (Pham et al., 2005) showed other two types of this technique shrinking and the standard algorithm. The idea behind the shrinking approach is taking the samples from increasingly small regions in solution space during the local search while the standard algorithm includes abandoning a site when stagnating local search in addition to the shrinking procedure. (Lindfield & Penny, 2017) introduced further modification by counting the number of times a recruited bee failed to explore an improved site, it is used as a guide in the exploration process to improve the efficiency and effectiveness of the search. Other enhancements which were introduced in (Imanguliyev, 2013) include an early neighbourhood search strategy that starts the search from promising patches, an efficiency based recruitment strategy that changes the number of recruited bees dynamically based on the efficiency of the related sites, a hybrid Tabu bee's algorithm, and autonomous bees algorithm. (Packianather et al., 2019) proposed a new version of BA discovering the rule automatically by adding two parameters namely quality weight and coverage weight to avoid ambiguous situations in the prediction stage. They formulated the new two parameters to carry out meta-pruning and make the algorithm suitable for classification tasks, it achieved better classification accuracy and reduced the number of rules making it a more efficient algorithm than other classification methods such as Jrip and other evolutionary algorithms.

In this thesis, the values of hyperparameters for BA are assigned based on the computer capability since the maximum number of iterations that can be performed for complicated CNN function is one and the number of scout bees that can be assigned is six with one elite site to be selected, the remaining parameters are selected based on the equations in (MathWorks-4) which are shown below:

- Maximum number of iterations = 1
- Number of scout bees (n) = 6
- Number of selected bees (m) = $0.5 \times n = 0.5 \times 6 = 3$
- Number of elite bees (e) = 1
- Number of recruited bees for elite (e) sites (nep) = $2 \times m = 2 \times 3 = 6$
- Number of recruited bees for other best (m-e) sites (nsp) = $0.5 \times n = 0.5 \times 6 = 3$
- Neighbourhood size for each selected patch (local search) (ngh) = $0.1 \times (\text{Var max} - \text{Var min})$
 - Section depth: $0.1 \times (3 - 1) = 0.1 \times 2 = 0.2$
 - Initial learning rate: $0.1 \times (1 - 1e-2) = 0.1 \times 0.99 = 0.099$
 - Momentum: $0.1 \times (0.98 - 0.8) = 0.1 \times 0.18 = 0.018$
 - Regularization: $0.1 \times (1e-2 - 1e-10) = 0.1 \times 0.01 = 0.001$
 - Weight learning rate factor: $0.1 \times (1.1 - 0.9) = 0.1 \times 0.2 = 0.02$

3.1.4 Convolutional Neural Network Architecture

The CNN architecture for classifying the ‘Cifar10DataDir’ images data into 10 classes composes of 15 layers of which one input layer, three convolutional layers, three rectified linear unit layers, three batch normalization layers, two max pooling layers, one fully connected layer, one SoftMax layer, and one classification layer.

The input layers are represented by a matrix of size height by width (32x32) that consists of pixel brightness numbers between 0 to 255 (0 for black and 255 for white). The convolutional layers contain filters represented by matrix of weights that slide along the pixel brightness input matrix to create a feature map matrix using a special dot product as mentioned in (Hui, 2017). Rectified linear unit layers are used after each convolutional layer to increase the speed and effectiveness of the training by mapping negative values to zero and maintaining positive values as mentioned in (MathWorks-1). In addition, batch normalization layers are used as supplement layers after each convolutional layer to mitigate the risk of overfitting by normalizing the input values of the following layers (Yamashita et al., 2018). Pooling layers are used between the convolutional layers to reduce the dimensionality of the output volume (McDermott, 2021) without losing the important features which contribute to minimize the computational cost. The classification layer performs the classification with a fully connected layer that shows the probability of each

image being classified for each class and the SoftMax layer that provides the classification output as presented in (MathWorks-1).

The normal number of convolutional layers to start is between two to three layers with a filter size of 3x3 or 5x5 as advised in (Hui, 2017). CNN is designed with three convolutional layers and two max pooling layers in between, the number of filters ranges between 11 for the first layer and 44 for the last one, and each layer has twice the number of filters of the previous layer (Brownlee, 2019). The filter size is 3x3, the section depth is 3, and the padding that helps to detect the edges of the images is set as 'same' so that the software calculates the size of the padding at the training time automatically and produces output size equal to the input size if the stride (number of pixel shift) is one.

The pooling type is 'max' which takes the most activated feature while the average pooling layer takes the average presence of the feature, so the average pooling is better with white background and max pooling is better with dark background as mentioned in (Ouf, 2017), the default size of pooling layer is 2x2 (Hui, 2017), but it yielded high computational cost since the image input size is big (32x32), the size was changed to 3x3 with a stride value of 3 to minimize the training time.

The classification layer that performs the classification contains a fully connected layer with 10 classes parameters (airplane, automobile, bird, cat deer, dog, frog, horse, ship, and truck) and a SoftMax activation function that works better with multi-class classification problems rather than a binary classification problem that requires sigmoid logistic function as shown in (McDermott, 2021).

The CNN is trained using SGDM which is the most common training algorithm (Yamashita et al., 2018), the default values for this algorithm are an initial learning rate of 0.01, a momentum of 0.9, a regularization of 1e-04, and the maximum number of epochs is 20 as presented in (MathWorks-6). After applying some experiments and monitoring the validation accuracy, the initial learning rate was changed to 0.1, regularization became 1e-10 and the number of epochs was 30.

3.1.5 The Existing BO-CNN Algorithm

First, the existing hybrid Bayesian Convolutional Neural Network (BO-CNN) (MathWorks-3) is applied where BO is used to optimize the four CNN parameter values

for section depth, initial learning rate, momentum, and regularization in order to minimize the classification error on the validation set.

In the hybrid BA-CNN, the BA is used instead of BO to optimize the same four CNN parameter values in order to minimize the classification error on the validation set.

Finally, the hybrid BA-BO-CNN algorithm uses BO to optimize the same four parameters while BA is used to optimize the weight learning rate factor to adjust the global learning rate obtained by BO algorithm in each convolutional layer and fully connected layer. Optimal learning rate means the optimum amount of weight update (Brownlee, 2020) in the convolutional filters and fully connected layer that performs the classification, so the classification accuracy on the validation set is improved.

The following figures 3.3 and 3.4 illustrate where BO and BA are inserted in the newly developed algorithms in order to optimize CNN parameters. It is followed by two subsections explaining each algorithm in detail.

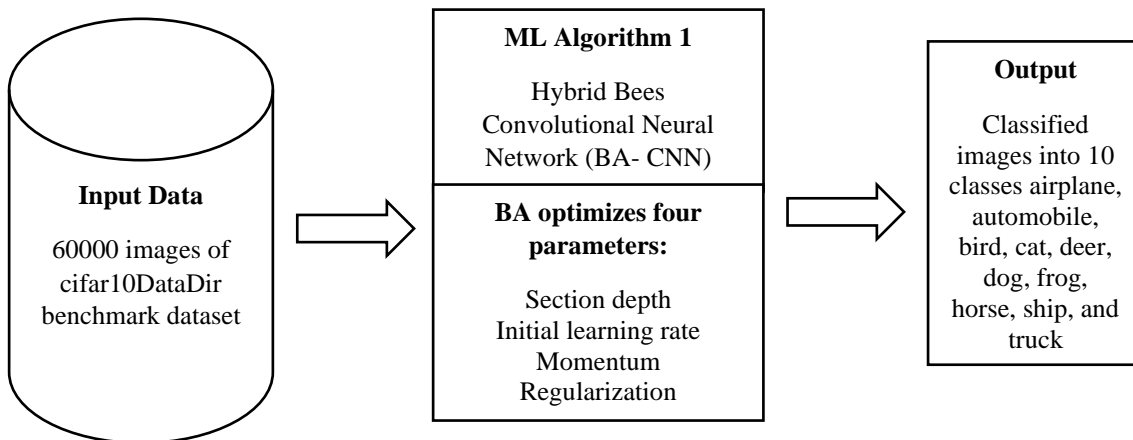


Figure 3.3: An Illustrative Diagram for BA-CNN Algorithm

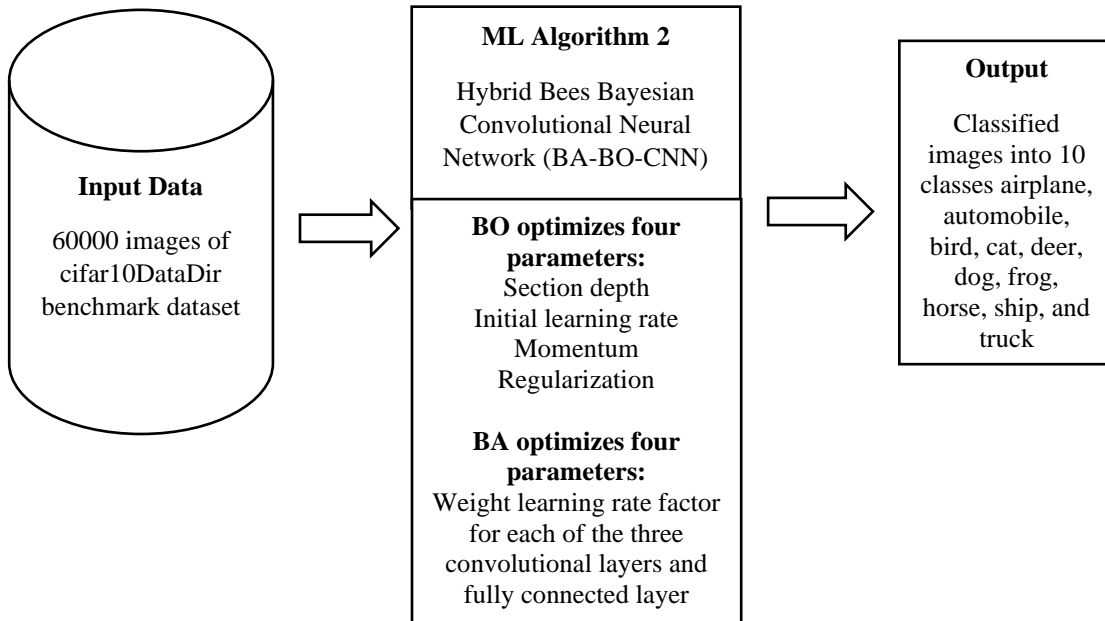


Figure 3.4: An Illustrative Diagram for BA-BO-CNN Algorithm

3.1.6 The Proposed Novel BA-CNN Algorithm

In the proposed hybrid BA-CNN algorithm, BA is used to optimize the four parameter values for section depth, initial learning rate, momentum, and regularization that yielded the minimum classification error on the validation set, they are the same four parameters that were optimized using BO technique in BO-CNN algorithm. The steps of the MATLAB code for the proposed hybrid BA-CNN algorithm are shown in figure 3.5. Some of the steps are taken from (MathWorks-4), but the architecture of the CNN, the selection of training, validation, and testing sets, defining the objective function and optimization variable, and adopting BA steps to optimize CNN parameters are developed in this thesis.

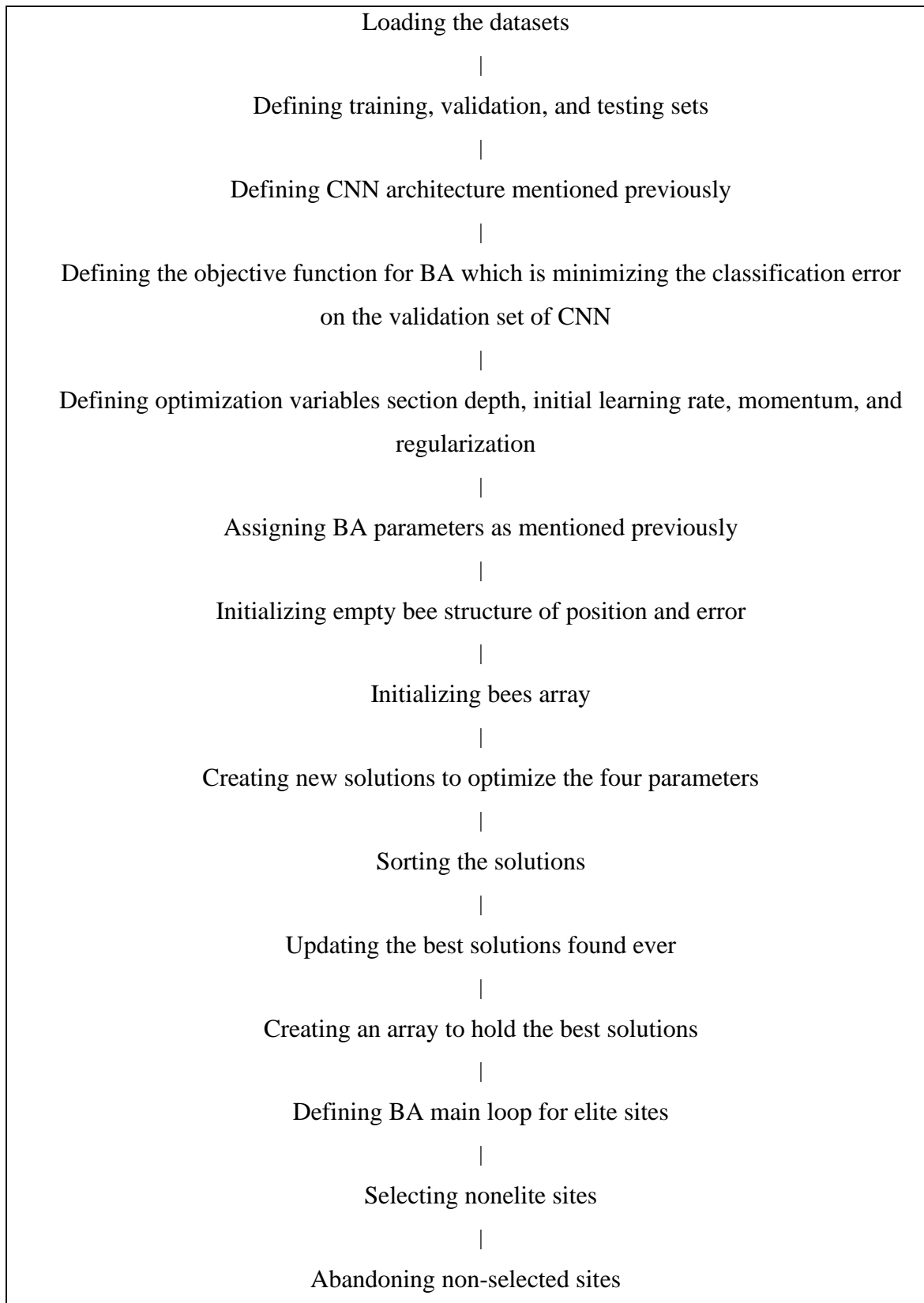


Figure 3.5: The Steps of the MATLAB Code for BA-CNN

In the BA-CNN algorithm, BA is used to optimize section depth, initial learning rate, momentum, and regularization which are the same parameters optimized using BO in the previous BO-CNN algorithm. BA starts with six scout bees arriving at random positions and evaluating them based on the validation error value. Then, a local search starts by selecting the three best sites and abandoning the remaining sites. The next step is selecting the size of the neighbourhood search space, which is 0.2 for section depth, 0.099 for initial learning rate, 0.018 for momentum, and 0.001 for regularization in order to recruit six bees for the elite sites and three bees for non-elite sites to conduct the local search within neighbourhood size and update the four parameters. The global and local searches will be performed simultaneously, while the recruited bees are exploring the best solutions around the neighbourhood, the global search on the remaining sites is carried out randomly.

3.1.7 The Proposed Novel BA-BO-CNN Algorithm

The hybrid BA-BO-CNN algorithm uses BO to optimize the same four parameters (section depth, initial learning rate, momentum, and regularization) while BA is used to optimize the adjustment factor for the global learning rate in each convolutional layer and fully connected layer in order to have a more optimum learning rate which results in a more optimum amount of weight update (Brownlee, 2020) in the convolutional filters and fully connected layer that perform the classification, so the classification accuracy on validation set is improved. The steps of the MATLAB code for the proposed hybrid BA-BO-CNN algorithm are shown in figure 3.6. Some of the steps are taken from (MathWorks-4), but the architecture of the CNN, the selection of training, validation, and testing sets, defining the objective function and optimization variable, and adopting BA steps to optimize CNN parameters are developed in this thesis.

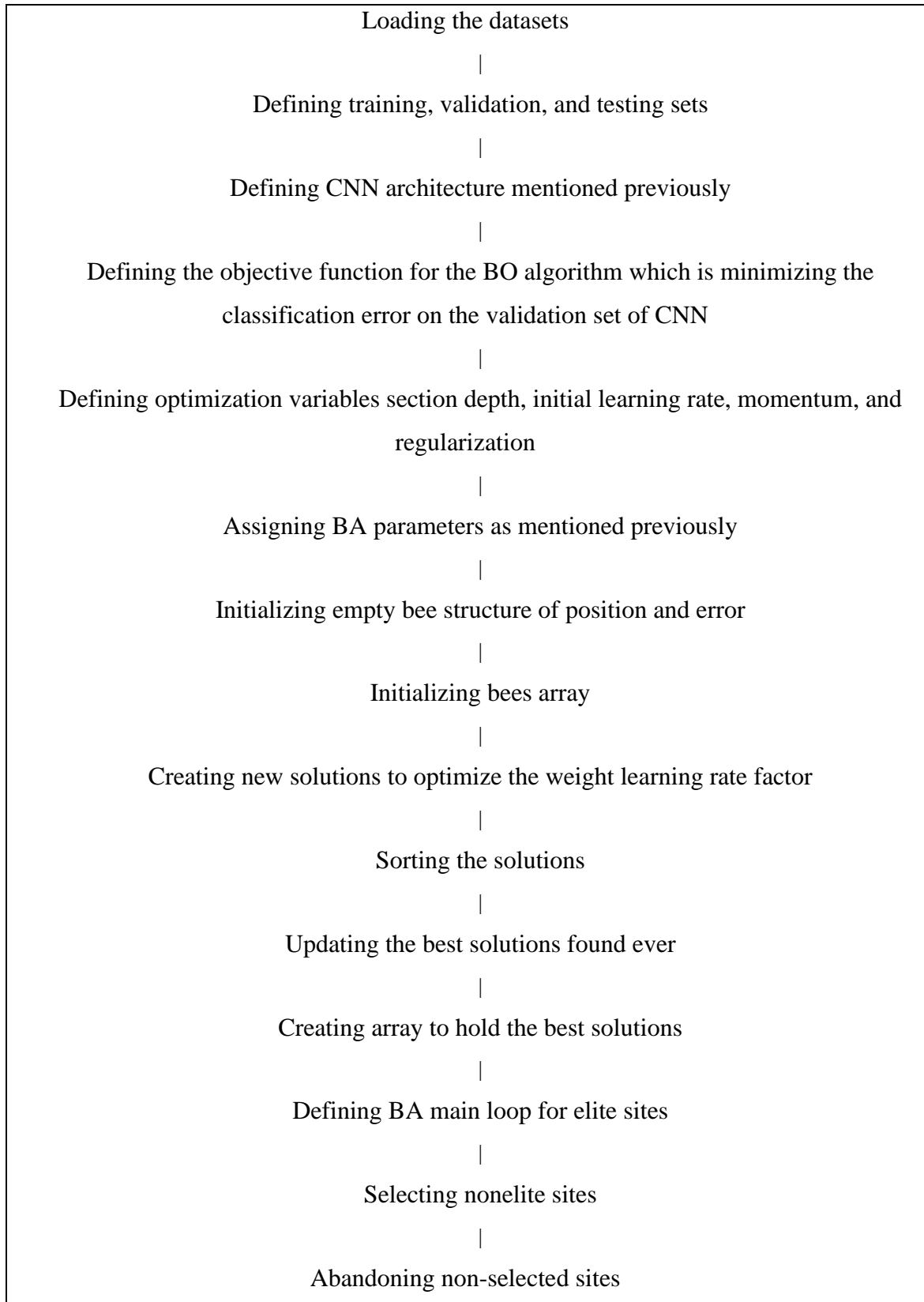


Figure 3.6: The Steps of the MATLAB Code for BA-BO-CNN

In addition, the following figure 3.7 shows the workflow diagram that explains the way of working for the BA-BO scheme to optimize CNN parameters:

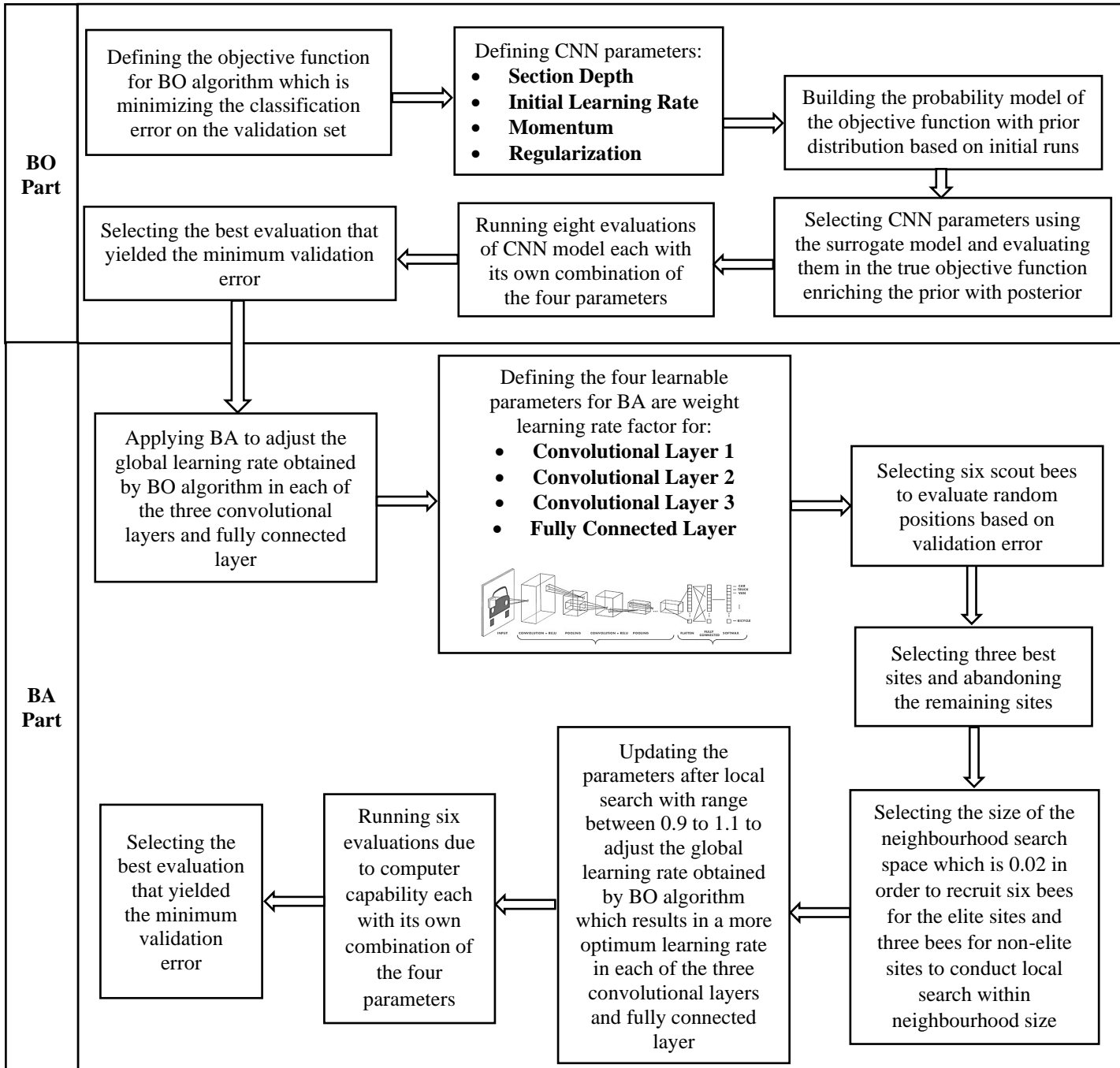


Figure 3.7: Workflow Diagram for the Hybrid BA-BO-CNN Algorithm

3.2 Results and Discussion of the Proposed BA-CNN & BA-BO-CNN Algorithms

MATLAB software is used to apply the hybrid BA-CNN and BA-BO-CNN algorithms on four benchmark datasets taken from (Machine Learning Repository). The first one is ‘Cifar10DataDir’ benchmark image data that consists of 60,000 images classified evenly into 10 classes airplane, automobile, bird, cat deer, dog, frog, horse, ship, and truck. The second dataset is handwritten digits with ten classes from 0 to 9 with 1,000 images in each class. The third set of data is concrete crack images with two classes, 5,000 negative images without cracks present in the road and 5,000 positive images with cracks. Finally, the algorithms are applied to 162 human Electrocardiogram (ECG) signals with three classes, 96 observations from persons with arrhythmia, 30 recordings from people with congestive heart failure, and 36 images from persons with normal sinus rhythms. For the last dataset, pre-trained CNN called ‘SqueezeNet’ (MathWorks-7) is used to classify ECG images since the available data is not large enough to train CNN from scratch, this is one of the threats in using CNN that it needs a large sample to train it, but it is the most powerful image classification tool as stated in (Elngar et al., 2021) and (Chaganti et al., 2020). The system configuration consists of a single CPU with a memory of 256 GB. As the DL algorithms require high computations, limited BA evaluations are applied to optimize CNN parameters. The datasets are shuffled at every epoch and the training, validation and testing sets are selected randomly after each shuffle in order to minimize the data biases and improve the validity of the experiments. The 60,000 images in the ‘Cifar10DataDir’ are divided into 50,000 images for training and 5,000 for each of the validation and testing sets. The handwritten digits and concrete cracks images are divided into 8,000 images for training and 1,000 for validation, and the same for the testing set. Finally, the number of samples for training set in the ECG images is 81 images, while the validation set consists of 41 observations, and the testing set has 40 samples.

In addition to existing CNN and BO-CNN, the novel hybrid BA-CNN is applied to the ‘Cifar10DataDir’ benchmark image dataset. BA is used to find the optimal parameter values for section depth, initial learning rate, momentum, and regularization that yielded the minimum classification error on the validation set, they are the same four parameters

that were optimized using the BO technique in BO-CNN algorithm. Table 3.5 shows the optimal parameters optimized by BA along with classification error on the validation set:

Table 3.5: Optimal CNN Parameters Values for BA-CNN

Optimized Variable using BA	Value
Optimal Section Depth	3
Optimal Initial Learning Rate	0.0289
Optimal Momentum	0.8734
Optimal Regularization	0.0049
Classification Error on the Validation Set	0.1928

The minimum classification error value is 0.1928, it is a result of a section depth value of 3, an initial learning rate of 0.0289, a momentum of 0.8734, and a regularization of 0.0049. The following figure 3.8 shows the training progress in blue line and validation accuracy in black line for BA-CNN:

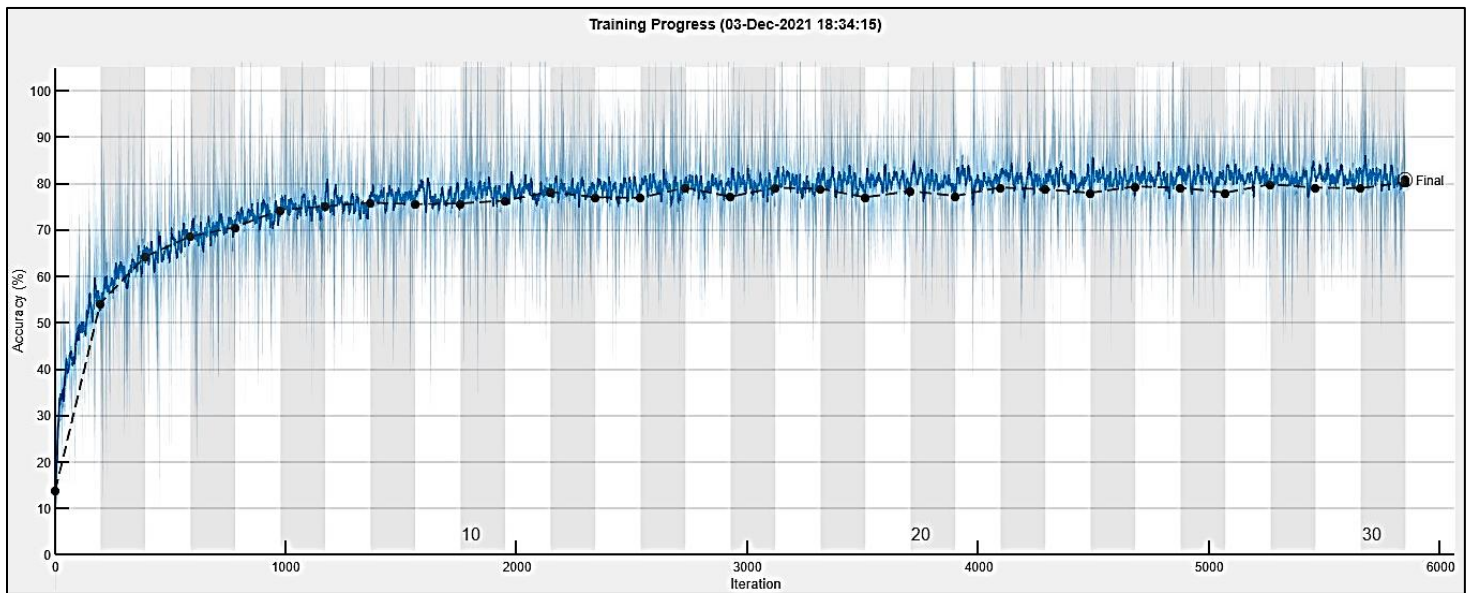


Figure 3.8: Training Progress for BA-CNN

The following three figures 3.9, 3.10, and 3.11 present the confusion matrix for training, validation and testing set for the BA-CNN algorithm showing precision and false alarm in the blue and red rows, and recall and miss out in the blue and red columns respectively:

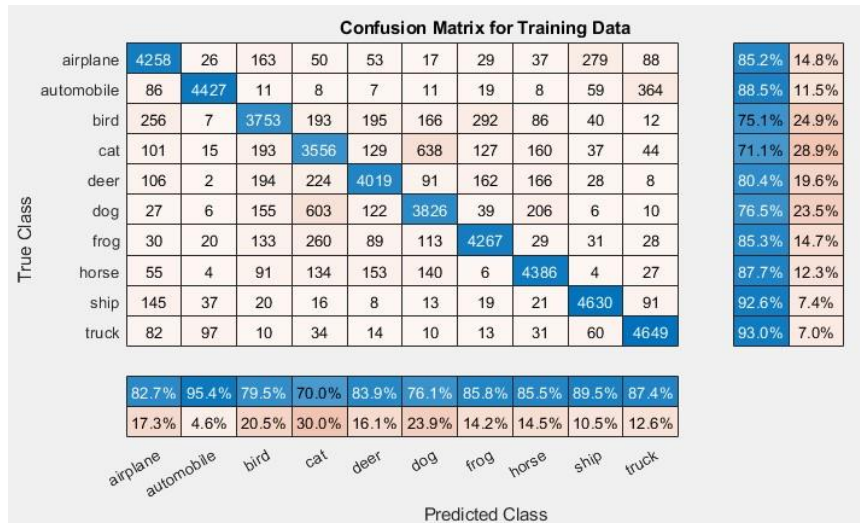


Figure 3.9: Confusion Matrix for Training Data for BA-CNN

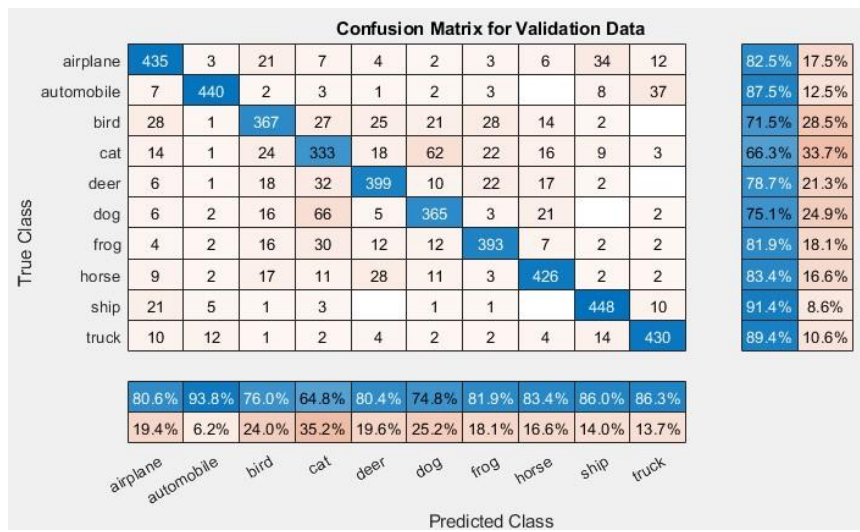


Figure 3.10: Confusion Matrix for Validation Data for BA-CNN

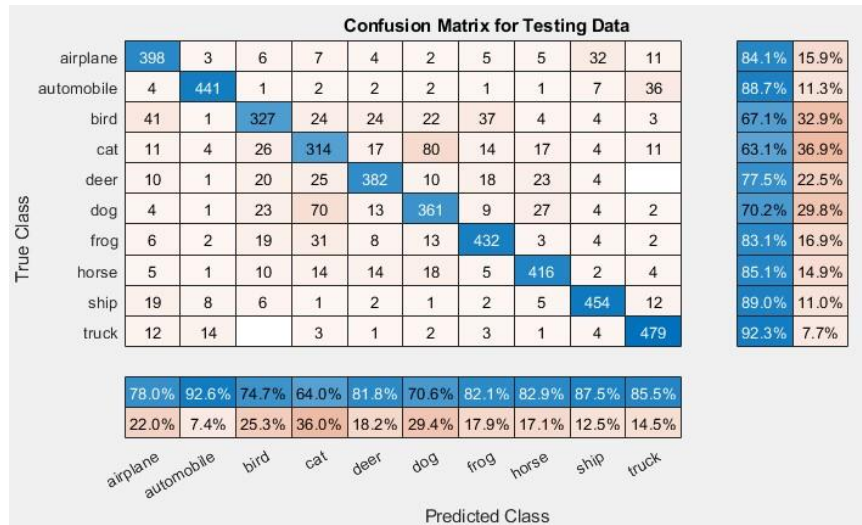


Figure 3.11: Confusion Matrix for Testing Data for BA-CNN

In the other hybrid BA-BO-CNN, the BO technique is applied to ‘Cifar10DataDir’ benchmark image data to find the optimum CNN parameters, figure 3.12 shows the minimum observed objective function and estimated minimum objective function:

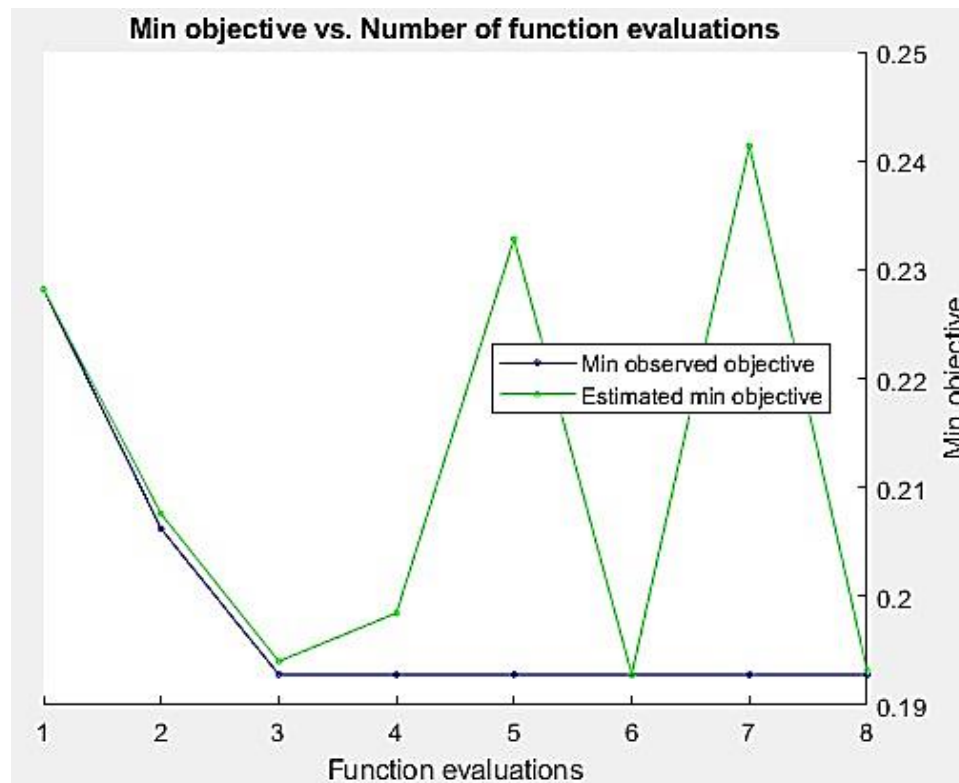


Figure 3.12: Number of Functions Evaluations

Table 3.6 shows the optimum parameters values for section depth, initial learning rate, momentum, and regularization that yielded the minimum classification error obtained by the BO technique after 8 evaluations for approximately 4 hours:

Table 3.6: Optimal CNN Parameters Values for BO-CNN

Optimized Variable using BO	Value
Optimal Section Depth	3
Optimal Initial Learning Rate	0.6893
Optimal Momentum	0.8407
Optimal Regularization	4.5857e-05
Classification Error on the Validation Set	0.1928

The minimum classification error value is 0.1928, it is a result of section depth value of 3, initial learning rate of 0.6893, momentum of 0.8407 and regularization of 4.5857e-05.

Then, BA is added to adjust the global learning rate in each of the three convolutional layers and fully connected layer. Table 3.7 shows the optimal weight learning rate factors optimized by BA along with classification error on the validation set:

Table 3.7: Optimal Weight Learning Rate Factors for BA-BO-CNN

Optimized Variable using BA	Value
Optimal Weight Learning Rate Factor for Convolutional Layer 1	0.9308
Optimal Weight Learning Rate Factor for Convolutional Layer 2	1.0924
Optimal Weight Learning Rate Factor for Convolutional Layer 3	1.0753
Optimal Weight Learning Rate Factor for Fully Connected Layer	0.9977
Classification Error on the Validation Set	0.1778

So, the global learning rate of 0.68934 obtained by BO is adjusted by multiplying it by 0.9308 in the first convolutional layer to become 0.6416. In the second convolutional layer, the adjustment factor is 1.0924 resulting in a learning rate value of 0.7530, the factor for the third convolutional layer is 1.0753, so the adjusted value is 0.7412. Finally, the new learning rate value for the fully connected layer is 0.6877 after adjusting it by a factor of 0.9977. Figure 3.13 shows the training progress in blue line and validation accuracy in black line after adding BA, the training progress curve for CNN, BO-CNN and BA-BO-CNN are similar.

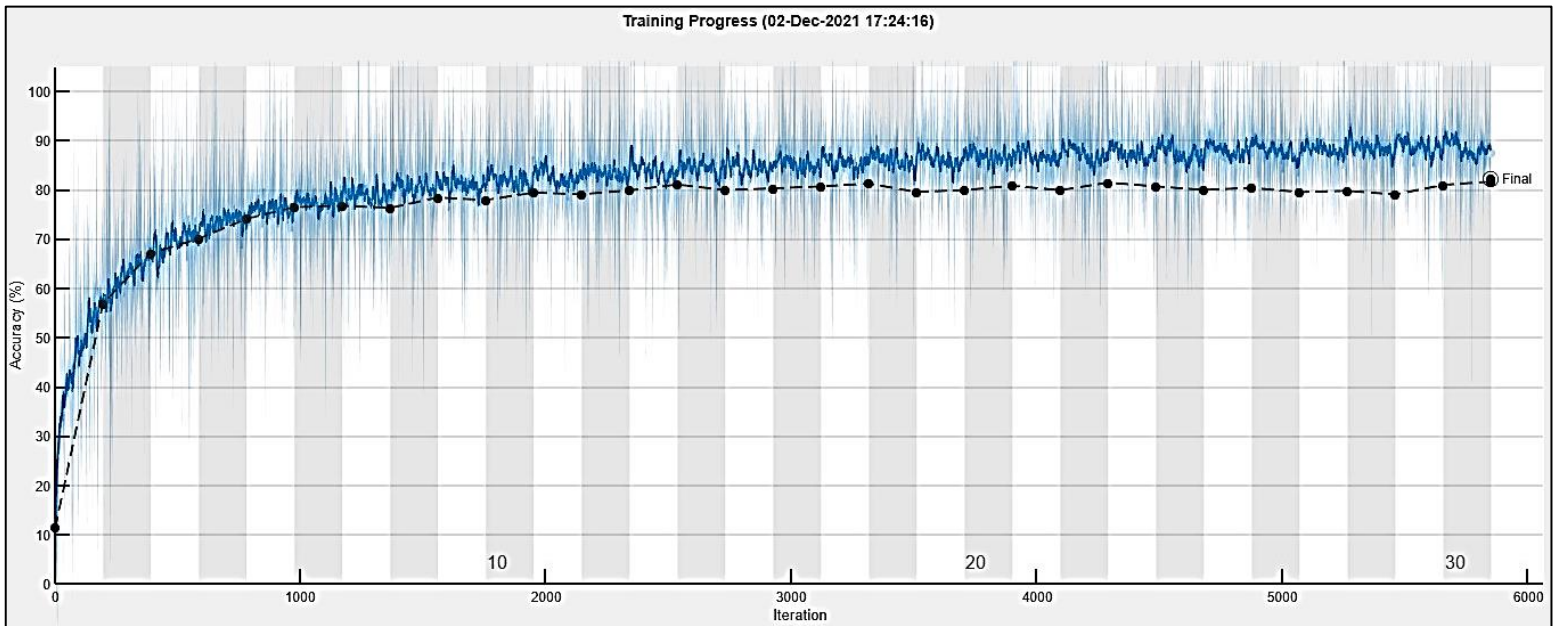


Figure 3.13: Training Progress for BA-BO-CNN

The following three figures 3.14, 3.15, and 3.16 present the confusion matrix for training, validation and testing set for the BA-BO-CNN algorithm showing precision and false alarm in the blue and red rows, and recall and miss out in the blue and red columns respectively:

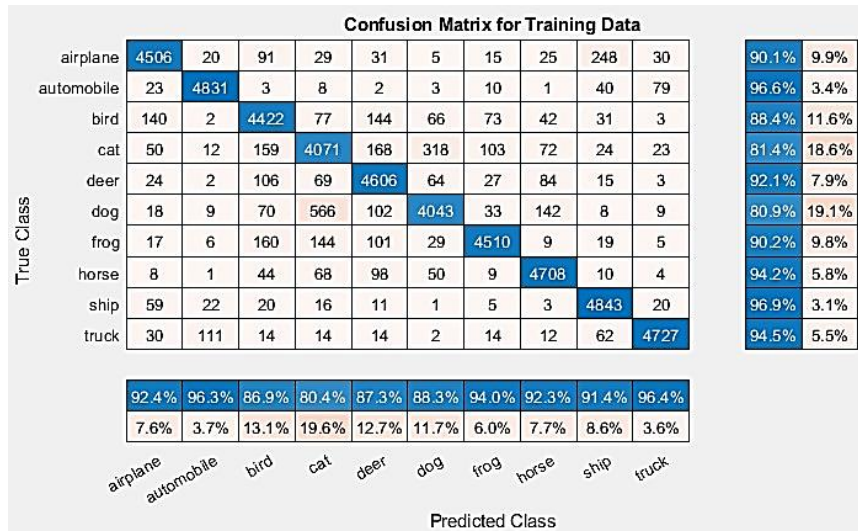


Figure 3.14: Confusion Matrix for Training Data for BA-BO-CNN

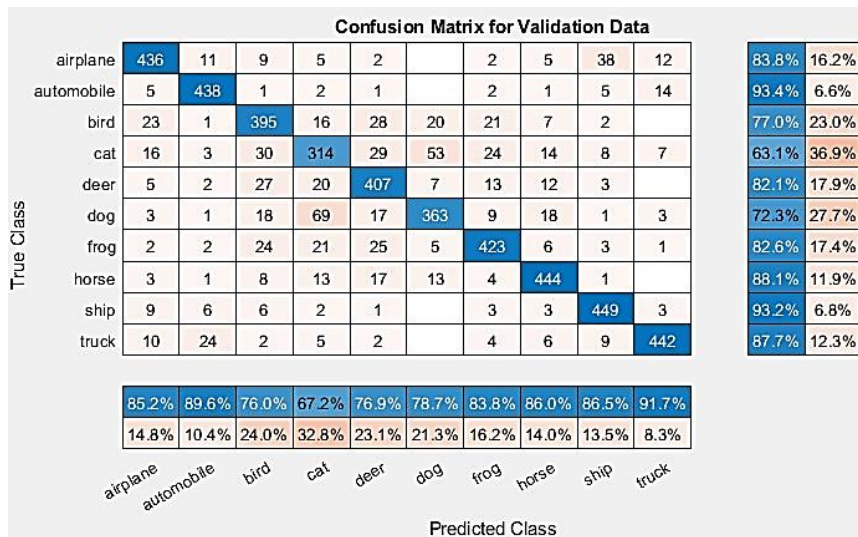


Figure 3.15: Confusion Matrix for Validation Data for BA-BO-CNN

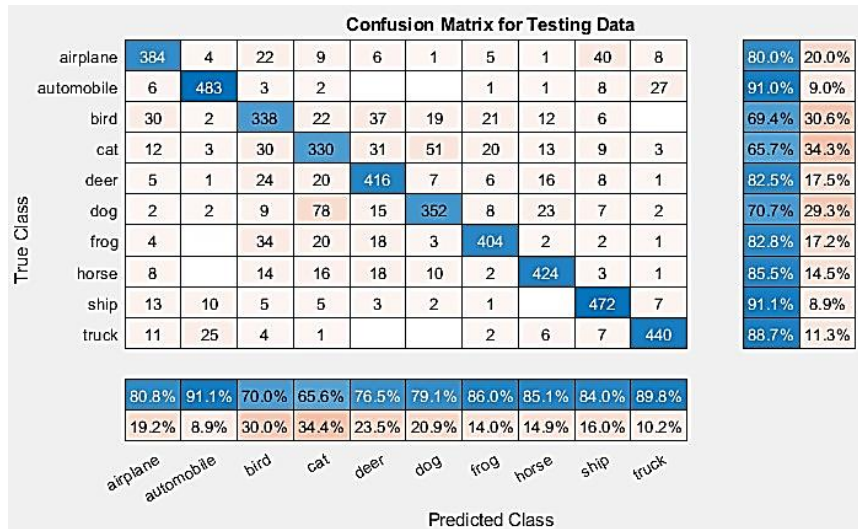


Figure 3.16: Confusion Matrix for Testing Data for BA-BO-CNN

The following table 3.8 shows the training, validation, and testing accuracy along with computational time for the best iteration for the original CNN, BO-CNN, hybrid BA-CNN, and BA-BO-CNN:

Table 3.8: Classification Accuracy and Computational Time for Algorithms (Cifar10DataDir Dataset)

	Existing Original CNN	Existing BO-CNN	Novel Hybrid BA-CNN	Novel Hybrid BA-BO-CNN
Training Accuracy	92.68%	90.27%	83.54%	90.53%
Validation Accuracy	80.34%	80.72%	80.72%	82.22%
Testing Accuracy	80.54%	80.69%	80.02%	80.74%
Computational Time	42 Min 56 Sec	43 Min 24 Sec	41 Min 46 Sec	41 Min 13 Sec

It is seen that the BA-CNN algorithm performs slightly better than the original CNN in terms of validation accuracy, but it is the same as the existing BO-CNN. In addition, it has better performance than the EA-CNN model designed in (Badan, 2019) that achieved an accuracy of 62.37% on the same ‘Cifar10DataDir’ dataset as was shown in table 2.2. The hybrid BA-BO-CNN algorithm has the best validation and testing accuracy, so it has the best classification performance and generalization capability to unseen data. In addition, it is the best algorithm in terms of cost-effectiveness since it achieved lower computational time than the BO-CNN algorithm by 2 minutes and 11 seconds.

The same procedure is followed to apply the original CNN, BO-CNN, hybrid BA-CNN, and BA-BO-CNN on the other three benchmark datasets, the results are shown in the following three tables 3.9, 3.10, and 3.11:

Table 3.9: Classification Accuracy and Computational Time for Algorithms (Digits Dataset)

	Existing Original CNN	Existing BO-CNN	Novel Hybrid BA-CNN	Novel Hybrid BA-BO-CNN
Training Accuracy	99.94%	100%	100%	100%
Validation Accuracy	98.80%	100%	99.90%	100%
Testing Accuracy	99.20%	99.99%	100%	99.99%
Computational Time	2 Min 35 Sec	6 Min 42 Sec	2 Min 28 Sec	3 Min 30 Sec

The table shows that the novel hybrid BA-BO-CNN algorithm produces the same accuracy as the BO-CNN algorithm due to simple features in the digits images which can be classified with high accuracy using existing algorithms, but the computational time in the hybrid algorithm is better by 3 minutes and 12 seconds reduction, so it is better in terms of cost-effectiveness. The novel hybrid BA-CNN algorithm produces almost similar results to existing BO-CNN in terms of accuracy, but it has the lowest computational time with 4 minutes and 14 seconds reduction compared to the BO-CNN model, so it is the best algorithm in terms of cost-effectiveness.

Table 3.10: Classification Accuracy and Computational Time for Algorithms (Concrete Cracks Dataset)

	Existing Original CNN	Existing BO-CNN	Novel Hybrid BA-CNN	Novel Hybrid BA-BO-CNN
Training Accuracy	99.96%	99.90%	98.90%	99.95%
Validation Accuracy	99.50%	99.67%	98.50%	99.67%
Testing Accuracy	99.83%	99.30%	98.85%	99.35%
Computational Time	3 Min 20 Sec	2 Min 46 Sec	2 Min 22 Sec	2 Min 50 Sec

In this dataset, the new hybrid algorithms have almost similar results to the existing original CNN and BO-CNN models since the dataset consists of only two classes, so applying the existing algorithms produced high accuracy.

Table 3.11: Classification Accuracy and Computational Time for Algorithms (ECG Dataset)

	Existing Original CNN	Existing BO-CNN	Novel Hybrid BA-CNN	Novel Hybrid BA-BO-CNN
Training Accuracy	100%	97.53%	100%	96.29%
Validation Accuracy	87.80%	90.24%	87.80%	90.24%
Testing Accuracy	92.50%	90%	92.50%	95%
Computational Time	1 Min 20 Sec	1 Min 20 Sec	1 Min 29 Sec	1 Min 20 Sec

It is seen that the best algorithms in terms of validation accuracy are BO-CNN and BA-BO-CNN algorithms with a value of 90.24%, while the best testing accuracy of 95% comes with a hybrid BA-BO-CNN model. So, it is concluded that the hybrid BA-BO-CNN algorithm has the best classification performance and generalization capability to unseen data, the computational time is almost similar for all algorithms. The hybrid BA-CNN model performs the same as the original CNN, but it is better than the BO-CNN algorithm in terms of testing accuracy. The computational time is almost similar for all algorithms, it is assumed that the computational time will be higher if a large dataset is available to train CNN from scratch instead of using pre-trained CNN ‘SqueezeNet’.

The developed hybrid CNN algorithms can be applied in the manufacturing context, particularly to analyse porosity images of parts manufactured by the SLM process, so the following chapter will create artificial porosity images mimicking the real CT scan of the finished SLM part which will be used to predict the percent of porosity in the finished SLM parts using the developed hybrid CNN algorithms in this chapter.

3.3 Summary

This chapter proposed a novel hybrid Bees Convolutional Neural Network (BA-CNN) algorithm which uses the BA to optimize CNN's four parameters (section depth, initial learning rate, momentum, and regularization) in order to increase the classification accuracy of the network. In addition, another novel nature inspired hybrid algorithm was proposed which combines BO with BA in order to increase the overall performance of CNN which is referred to as the BA-BO-CNN algorithm. BO was used to optimize four CNN parameters section depth, initial learning rate, momentum, and regularization while BA was applied to optimize the weight learning rate factor to adjust the global learning rate obtained by BO algorithm in each convolutional layer and fully connected layer.

Applying the hybrid BA-CNN algorithm to the 'Cifar10DataDir' benchmark image data performed slightly better than original CNN in terms of validation accuracy, but it is the same as existing BO-CNN. In addition, it has better performance than the EA-CNN model designed in (Badan, 2019) that achieved an accuracy of 62.37% on the same 'Cifar10DataDir' dataset as was shown in table 2.2. Applying it to the digit dataset produced the lowest computational time with 4 minutes and 14 seconds reduction compared to BO-CNN, thus showing that it is the best algorithm in terms of cost-effectiveness. However, applying it to concrete cracks images and artificial porosity images produced almost similar results to existing algorithms. Finally, applying it to ECG images improved the testing accuracy from 90% for BO-CNN to 92.50% for the BA-CNN algorithm.

Furthermore, applying the hybrid BA-BO-CNN algorithm to 'Cifar10DataDir' benchmark image data yielded an increase in the validation accuracy from 80.72% to 82.22%, while applying it to the digit dataset showed the same accuracy as the existing original CNN and BO-CNN models, but with an improvement in the computational time by 3 minutes and 12 seconds reduction. Applying it to concrete cracks images produced almost similar results to the existing algorithms, and finally applying it to human Electrocardiogram (ECG) signals improved the testing accuracy from 92.50% for the original CNN to 95% for the BA-BO-CNN algorithm.

The contributions of this chapter are:

- Developing a novel hybrid Bees Convolutional Neural Network (BA-CNN) algorithm in order to improve the performance of CNN.
- Developing a novel hybrid Bees Bayesian Convolutional Neural Network (BA-BO-CNN) algorithm in order to improve the performance of CNN.

Chapter 4: Artificial Porosity Images Creation for Selective Laser Melting Parts

4.1 Artificial Porosity Images Creation Process

Training CNN algorithms would require a large amount of experimental data, which is expensive for SLM parts (www.facfox.com, 2022) since there are many types of production cost for pre-processing, processing, and post-processing cost including preparing geometry data, CAD model, machine setup, material cost, building up the part, and post-processing cost (Rickenbacher, 2013), so producing a large amount of porosity images to train the RCNN is not cost-effective.

This chapter proposes a new efficient approach of creating artificial keyhole porosity images mimicking the real CT scan slices of the finished SLM parts that can be used in the research environment effectively and efficiently. In particular, the artificial images are used to validate the training of an accurate RCNN for the automatic prediction of porosity in CT scans of SLM parts. The steps for creating the artificial porosity images are:

- Establishing regression equations
- Generating pores number and diameter
- Creating 3D cubes
- Slicing 3D cubes into 2D images
- Labelling 2D slices
- Adding noisy background

They are shown in the following figure 4.1 which is followed by subsections explaining each step in detail.

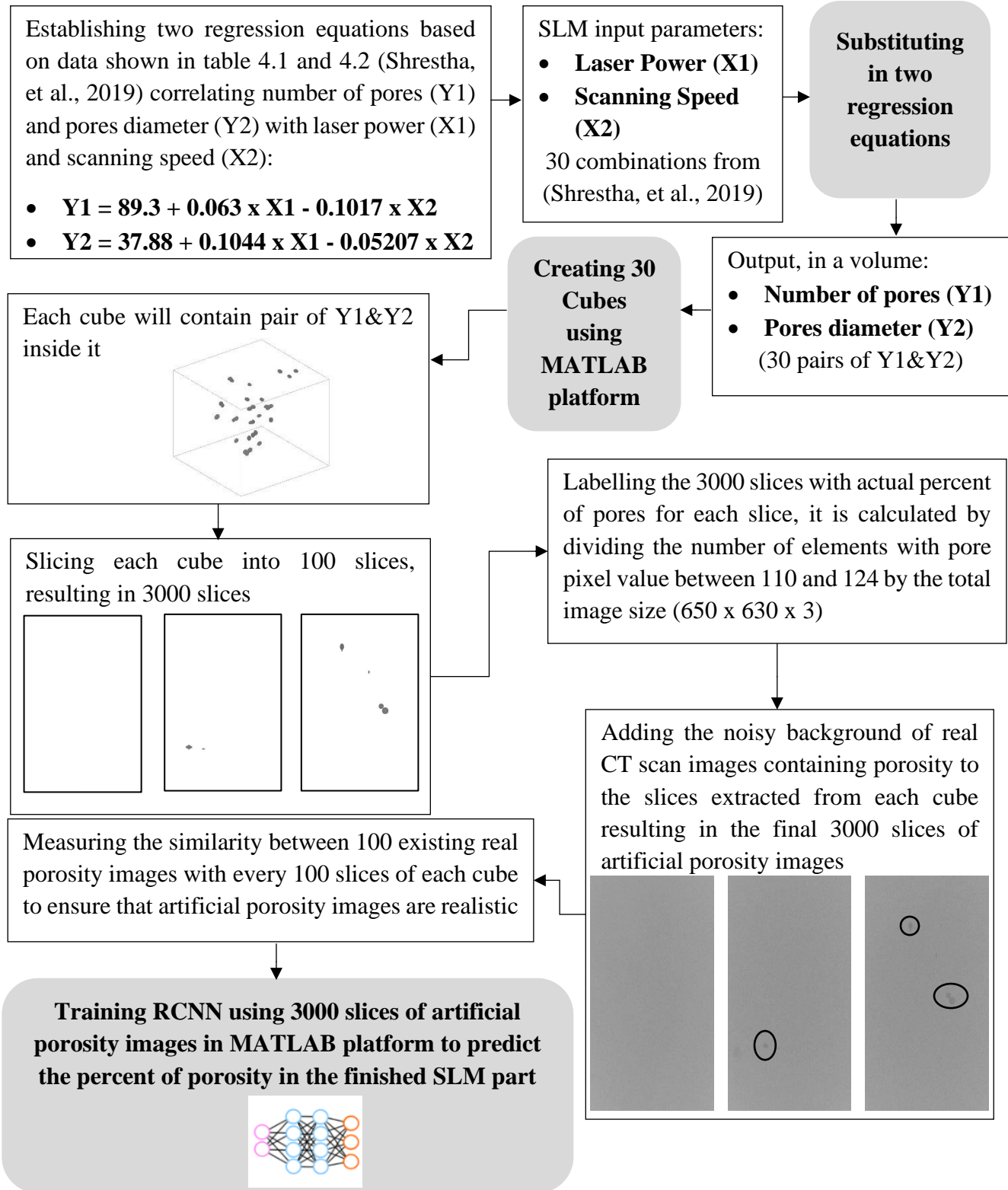


Figure 4.1: Flowchart for the Steps of Creating Artificial Porosity Images

4.1.1 Establishing Regression Equations

The formulation of two regression equations is based on laser power and scanning speed data found in (Shrestha, et al., 2019), the first one correlates the number of pores (Y1) with laser power (X1) and scanning speed (X2) and other equation correlates pores diameter (Y2) with the same parameters (X1 and X2). The data used to establish the two equations are related to keyhole porosity and they will be used as a demonstration for an example of pore types formation.

The following table 4.1 shows the corresponding number of pores for each combination of laser power and scanning speed:

Table 4.1: Number of Pores Data (Shrestha, et al., 2019)

#	Laser Power (W)	Scanning Speed (mm/s)	Number of Pores in 2.6 mm³ Volume
1	195	200	101
2	175	200	121
3	150	200	121
4	125	200	101
5	195	400	41
6	175	400	75
7	150	400	55
8	195	600	4
9	125	400	30
10	175	600	19
11	150	600	4
12	195	800	0
13	175	800	1
14	125	600	2
15	150	800	0
16	195	1000	0
17	175	1000	0
18	125	800	0
19	195	1200	0
20	150	1000	1
21	175	1200	0
22	125	1000	0
23	150	1200	0
24	125	1200	0

The Minitab software is used to establish the first regression equation that fits the 24 observations (Shrestha, et al., 2019) mentioned in table 4.1, thus correlating the number of pores with the laser power and the scanning speed which result in the following equation:

$$\text{Number of Pores} = 89.3 + 0.063 \times \text{Laser Power} - 0.1017 \times \text{Scanning Speed} \quad \text{(Equation 4.1)}$$

Similarly, another regression equation is established to fit pores diameter with laser power and scanning speed data found in (Shrestha, et al., 2019), the dataset is shown in the following table 4.2:

Table 4.2: Pores Diameter Data (Shrestha, et al., 2019)

#	Laser Power (W)	Scanning Speed (mm/s)	Pores Diameter (µm)
1	195	200	47
2	175	200	46
3	150	200	42
4	125	200	45
5	195	400	36
6	175	400	41
7	150	400	27
8	125	400	26
9	195	600	25
10	175	600	27
11	150	600	25

Once again, the Minitab software is used to establish the regression equation that fits the 11 observations (Shrestha, et al., 2019) mentioned in table 4.2 correlating the pores diameter with the laser power and the scanning speed, which results in the following equation:

$$\text{Pores Diameter} = 37.88 + 0.1044 \times \text{Laser Power} - 0.05207 \times \text{Scanning Speed} \quad \text{(Equation 4.2)}$$

4.1.2 Generating Pores Number and Diameter

The two regression equations established in the previous step are used to generate 30 values for the number of pores and pores diameter by substituting 30 combinations of laser power and scanning speed found in (Shrestha, et al., 2019). The following is an illustrative example with a laser power of 15 W and a scanning speed of 50 mm/s:

$$\text{Number of Pores} = 89.3 + 0.063 \times (15) - 0.1017 \times (50) = 85 \text{ Pores}$$

$$\text{Pores Diameter} = 37.88 + 0.1044 \times (15) - 0.05207 \times (50) = 36.94 \mu\text{m}$$

The following table 4.3 shows 30 combinations of laser power and scanning speed found in (Shrestha, et al., 2019) along with the corresponding number of pores after scaling down from a 2.6 mm³ to a 1 mm³ volume, the volume of the cube in which the pores will be positioned and used to train the RCNN. Variations of +/- 10 μm are applied randomly to the generated pore diameters mimicking variations occurring in real porosity images:

Table 4.3: Number of Pores and Pores Diameter for Combined Laser Power and Scanning Speed

#	Laser Power (W)	Scanning Speed (mm/s)	Number of Pores in 2.6 mm ³ volume	Number of Pores After Scaling (Divided by 2.6 to scale it to 1 mm ³ volume)	Average Pores Diameter (μm)	Maximum Pores Diameter (μm)	Minimum Pores Diameter (μm)
1	16	50	85	33	36.9469	46.9469	26.9469
2	20	50	85	33	37.3645	47.3645	27.3645
3	24	50	86	33	37.7821	47.7821	27.7821
4	32	100	81	31	36.0138	46.0138	26.0138
5	40	100	82	32	36.8490	46.8490	26.8490
6	48	100	82	32	37.6842	47.6842	27.6842
7	48	150	77	30	35.0807	45.0807	25.0807
8	60	150	78	30	36.3335	46.3335	26.3335
9	72	150	79	30	37.5863	47.5863	27.5863
10	64	200	73	28	34.1476	44.1476	24.1476

11	80	200	74	28	35.818	45.818	25.818
12	96	200	75	29	37.4884	47.4884	27.4884
13	80	250	69	27	33.2145	43.2145	23.2145
14	100	250	70	27	35.3025	45.3025	25.3025
15	120	250	71	27	37.3905	47.3905	27.3905
16	96	300	65	25	32.2814	42.2814	22.2814
17	120	300	66	25	34.7870	44.7870	24.7870
18	144	300	68	26	37.2926	47.2926	27.2926
19	112	350	61	23	31.3483	41.3483	21.3483
20	140	350	63	24	34.2715	44.2715	24.2715
21	168	350	64	25	37.1947	47.1947	27.1947
22	130	406.2	56	22	30.3011	40.3011	20.3011
23	162.5	406.2	58	22	33.6941	43.6941	23.6941
24	195	406.2	60	23	37.0871	47.0871	27.0871
25	144	450	53	20	29.4821	39.4821	19.4821
26	180	450	55	21	33.2405	43.2405	23.2405
27	156	487.5	50	19	28.7822	38.7822	18.7822
28	195	487.5	52	20	32.8538	42.8538	22.8538
29	176	550	44	17	27.6159	37.6159	17.6159
30	195	609.4	40	15	26.5065	36.5065	16.5065

4.1.3 Creating 3D Cubes

30 samples of 3D cubes are created with a volume of 1 mm³ for each cube, they contain the number of pores and the pores diameter generated in the previous section (table 4.3), Thus, the first cube has 33 pores with an average diameter of 36.94 μm. In (Maskery et al., 2016), the position of pores was considered by analysing the porosity distribution in real images of pores found in alloy Al–Si10–Mg parts produced by SLM process with laser power of 200 W, scanning speed 318 mm/s, and hatch spacing of 80 μm. They observed 975 porosity positions in X and Y. In this research, the Minitab platform is used to conduct

a normality test on these observations to check if their statistical distribution is normal or not, as shown in the following figures 4.2 and 4.3:

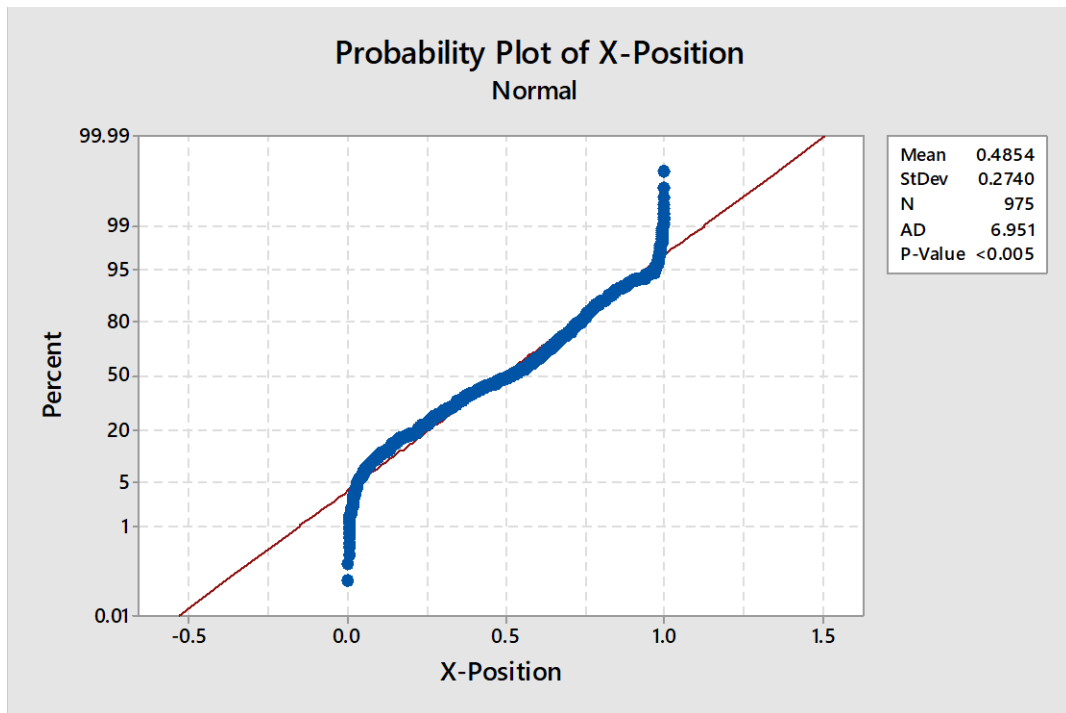


Figure 4.2: Probability Plot for X-Position

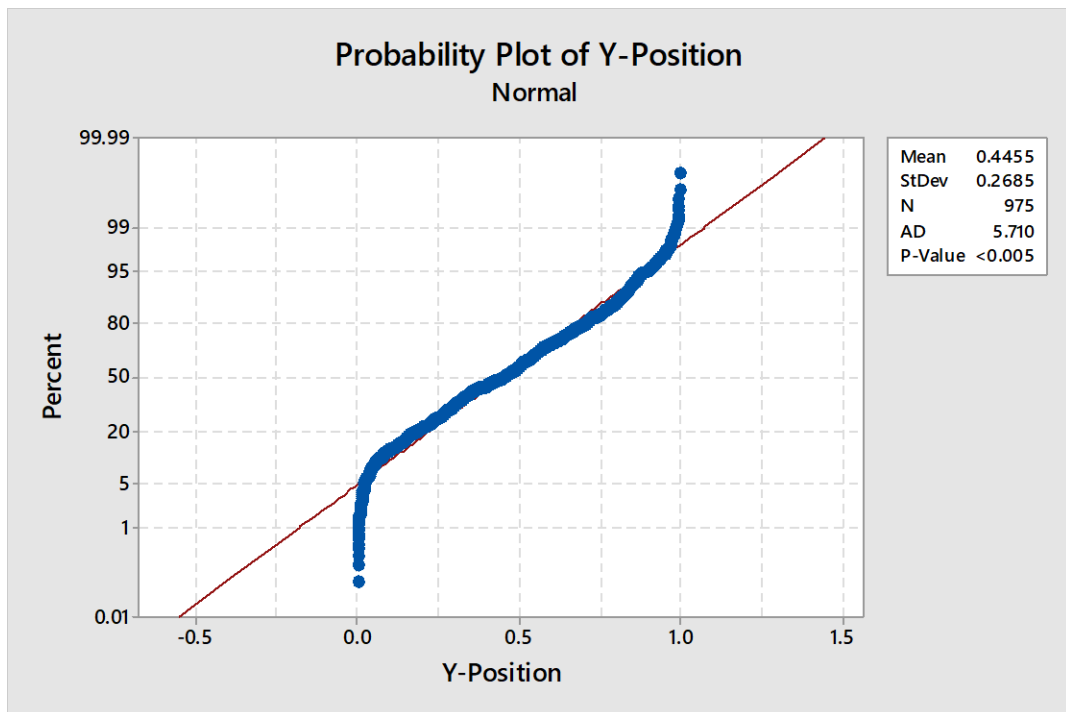


Figure 4.3: Probability Plot for Y-Position

With a P-value of less than 0.005, the probability that these data are not coming from a normal distribution is very low, so it is concluded that the observed porosity positions are normally distributed, with a mean and a standard deviation of 0.48 and 0.27 for X-positions and 0.44 and 0.26 for Y- positions. Therefore, in this research, it is decided to produce porosity positions with the same statistical distribution as observed in (Maskery et al., 2016).

Thus, the MATLAB software is used to generate normally distributed porosity positions inside each of the 30 cubes (Appendix 1.5) with the mean and the standard deviation mentioned previously, in the X and Y positions, while the Z position is arbitrarily given the average between X and Y to produce normal distribution pores in a volume. The pores morphology is created to be similar to the 3D view of the pores shown in figure 2.10 in section 2.8.6 (Shrestha et al., 2019). The following figure 4.4 shows an illustrative example of the cube with 33 pores and an average diameter of 36.94 μm as mentioned in the first combination of table 4.3:

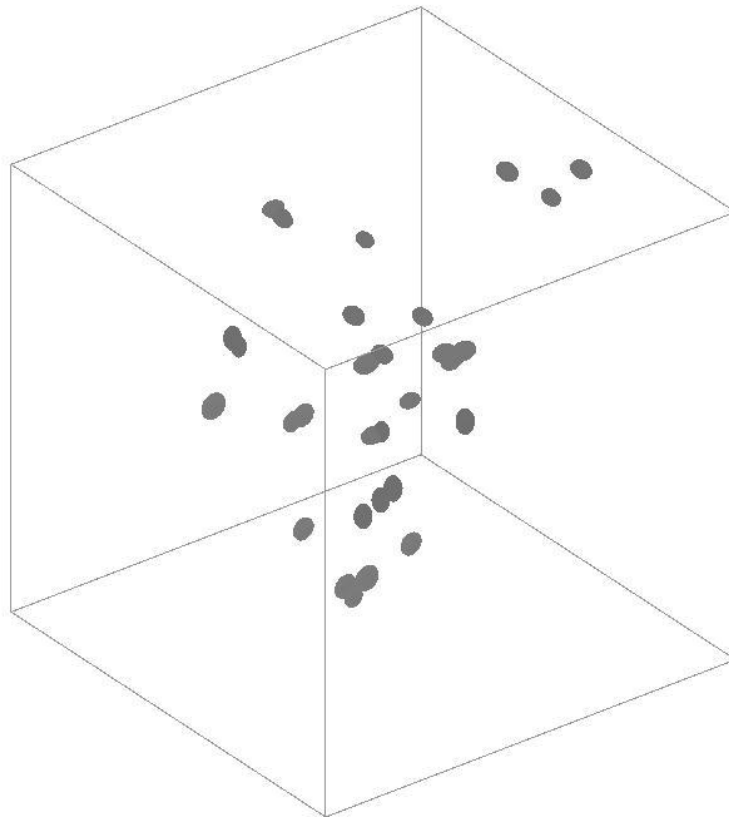


Figure 4.4: 3D Cube with 33 Pores and Average Pores Diameter of 36.94 μm

4.1.4 Slicing 3D Cubes into 2D Images

Each cube created in the previous step is sliced into 100 slices using the MATLAB software (Appendix 1.5) with a thickness of 0.01 mm resulting in 3000 slices of 2D images, an illustrative example for the cube and three not sequential slices is shown in the following figure 4.5:

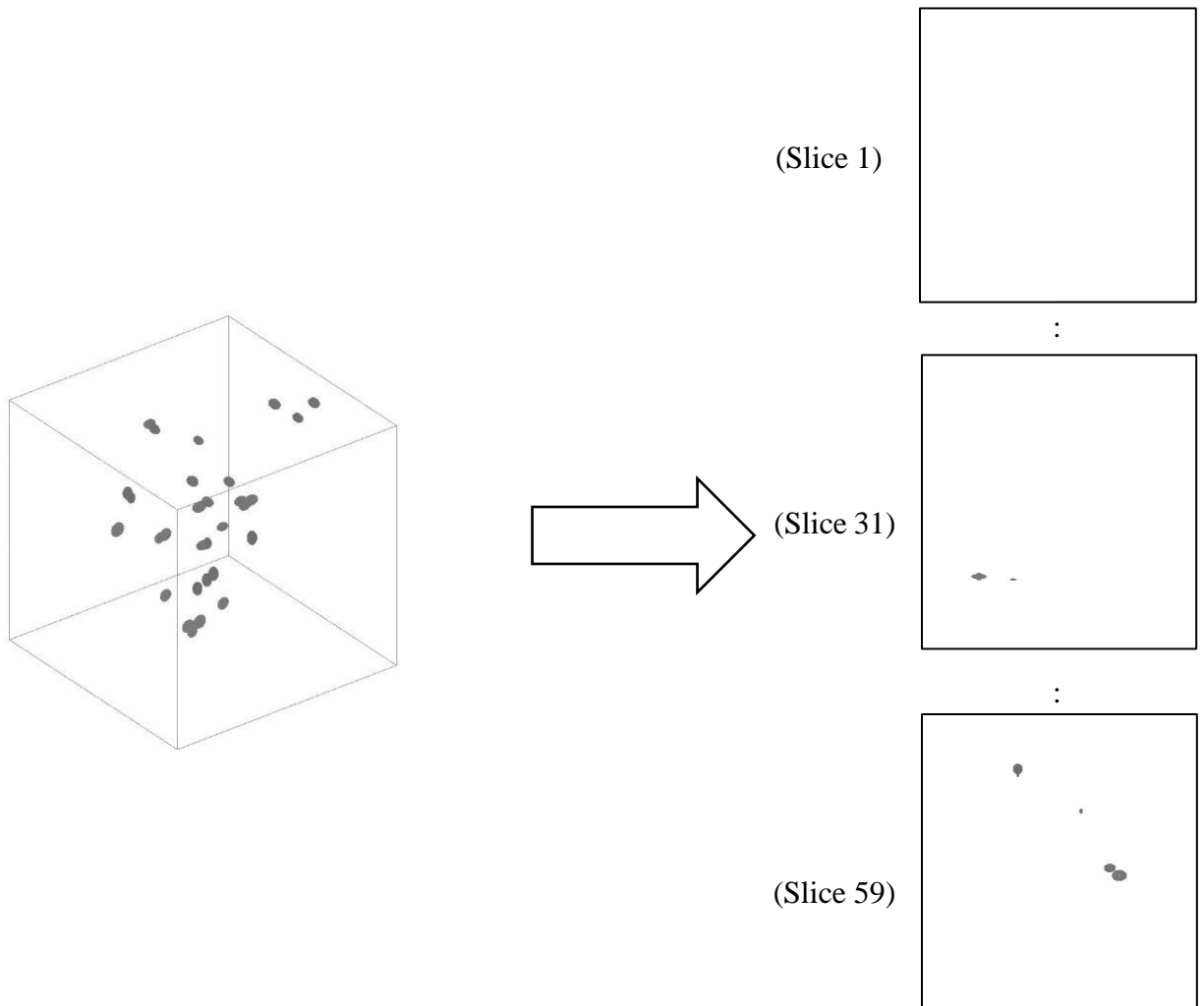


Figure 4.5: An Illustrative Example of the Cube Slicing

The first slices are expected to be with no pore as shown at the bottom of the cube, thereafter part of a pore is shown in some slices and parts of many pores are shown in other slices. The pore morphology tended to be near-spherical in shape as described in section 2.8.6 (Tan et al., 2020).

4.1.5 Labelling 2D Slices

The slices are labelled with the actual percent of porosity for each slice using the MATLAB software (Appendix 1.5), it is calculated by dividing the number of elements with specific pore unique pixel value by the total image size (650 x 630 x 3). The pixel values for the pores are determined by inspecting the pixel values of porosity seen in real CT Scan images (figure 4.9) (Feng et al., 2022), they are between 110 and 124. The first slice has no elements with a pixel value in the range between 110 and 124, so the actual percent of the pore is 0. In slice 31, there are 258 elements with pixel values in the range specified, so the actual percent of the pore is $(258 / (650 \times 630 \times 3)) \times 100 = 0.0210$. Similarly for slice 59 with 2193 elements of pixel values between 110 and 124, the actual percent of the pore is $(2193 / (650 \times 630 \times 3)) \times 100 = 0.1785$, the average actual percent of the pore for the 3000 slices is 0.0134.

4.1.6 Adding Noisy Background

First, image processing in the MATLAB platform is conducted to reduce the pixel brightness values of the background to different gray values between 180 and 150 resulting in the first version of the artificial images shown in the following figure 4.6:

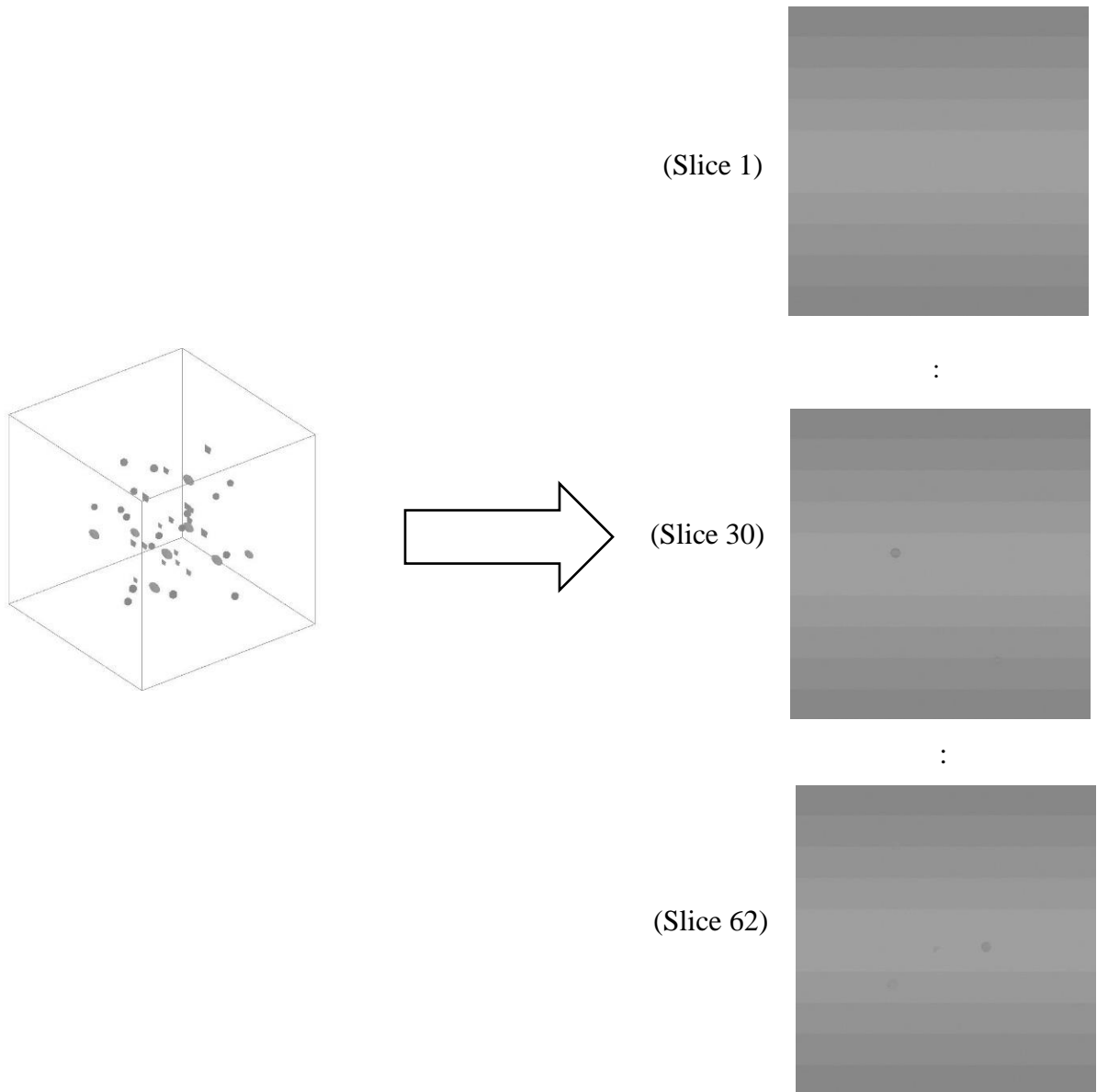


Figure 4.6: An Illustrative Example of the First Version of Artificial Porosity Images

As shown in the previous slices, the background is clean and does not mimic the real CT scan slices of the finished SLM part.

The noisy background of 100 existing real porosity images made in (Feng et al., 2022) are fused with every 100 slices of each cube resulting in the following second version of artificial porosity images shown in figure 4.7:

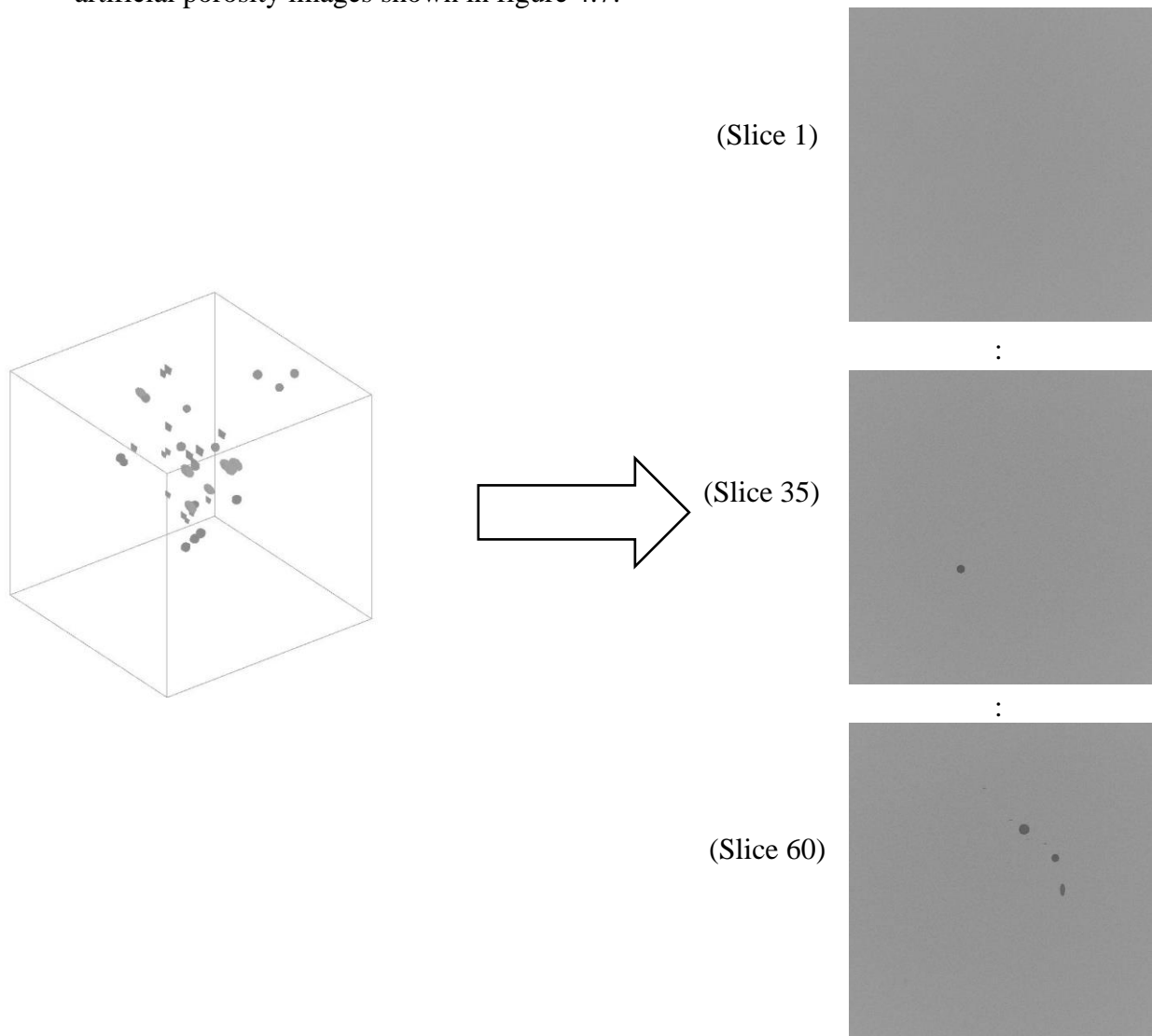


Figure 4.7: An Illustrative Example of the Second Version of Artificial Porosity Images

In the artificial images, the edges of the pores have a uniform shading making them clearly distinguishable from the background. This is not a true representation of porosity seen on real CT scans, where shade variations make it difficult to properly distinguish pores' edges. Thus, in a third version, once again image processing in the MATLAB platform is conducted to further filter the artificial porosity images by creating similar noise, so the background of a sample of 100 existing images is inserted into 100 pores slices extracted from 3D cube after reducing the pixel value for its white background from 255 to

155 to convert it to gray background. The artificial images are overlaid with the noisy background of real images. The degree of overlaying is determined by a factor (between 0 and 1) following the method described in (MathWorks-8) (Appendix 1.5). Thus, with a factor of 1, the pore image is shown without noisy background and with a factor of 0, the noisy background is shown without pores. After many trial and error experiments, a factor of 0.125 is considered to visually be the best value that combined both pore and noisy background images as shown in the following figure 4.8:

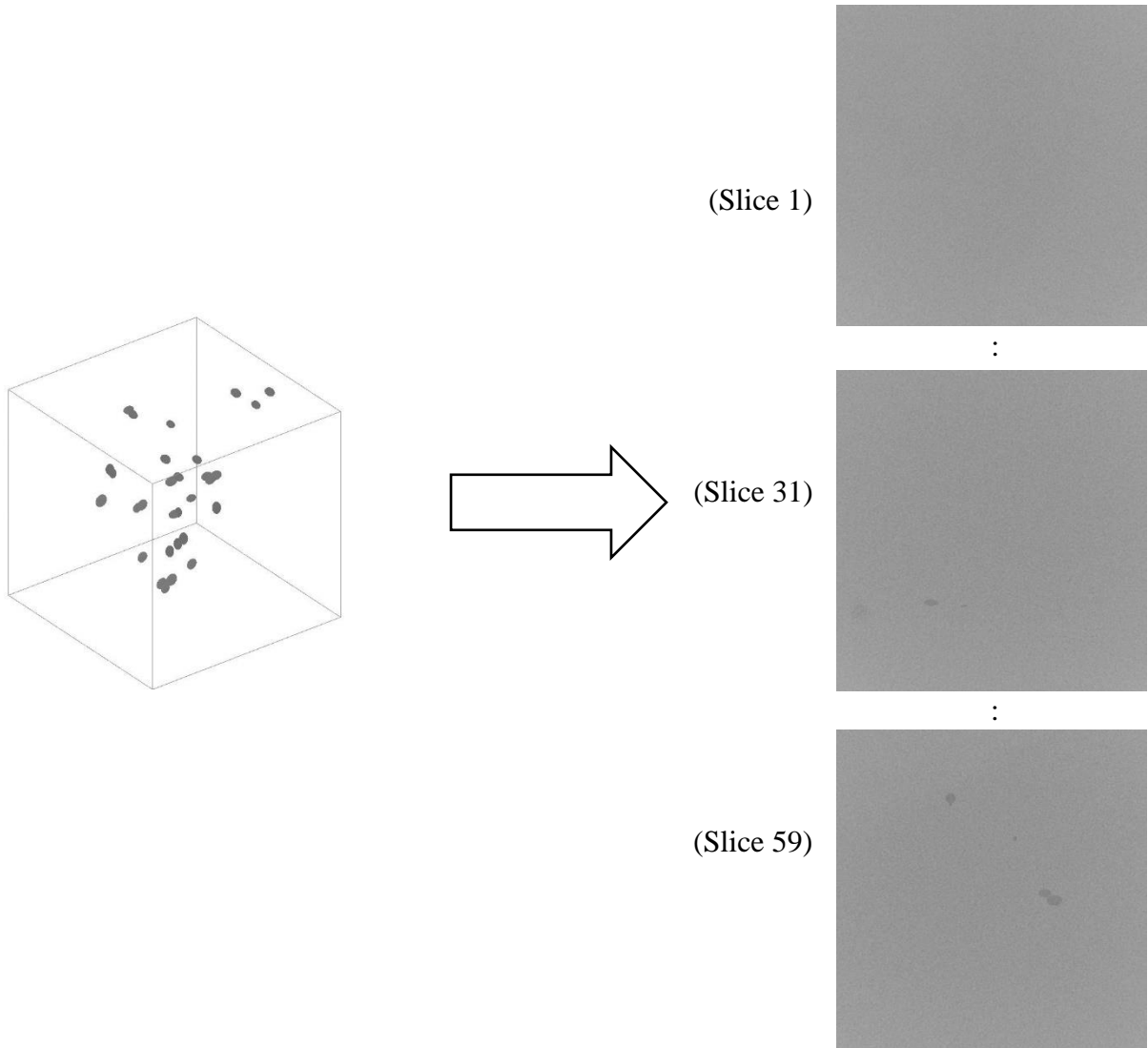


Figure 4.8: An Illustrative Example of the Final Version of Artificial Porosity Images

As shown in the slices, the noise in the background and the pores appear more like real porosity images as the edges are not as clearly defined, but this is a subjective evaluation.

The following section 4.2 will measure the similarity between real and artificial porosity images to make sure the created artificial images are realistic.

4.2 Results and Discussion of the Similarity between Artificial and Real Porosity Images

The created artificial porosity images are compared with real CT scan slices of finished SLM parts (Feng et al., 2022) to verify that the simulated images are close to the reality. The real images are only used as a demonstration and are not directly linked with the pore formation equations mentioned in section 4.1.1. The following figure 4.9 shows an illustrative example of the real slices before and after processing, which was required to focus on the bulk of the material and to obtain images of the same size as the artificial images.

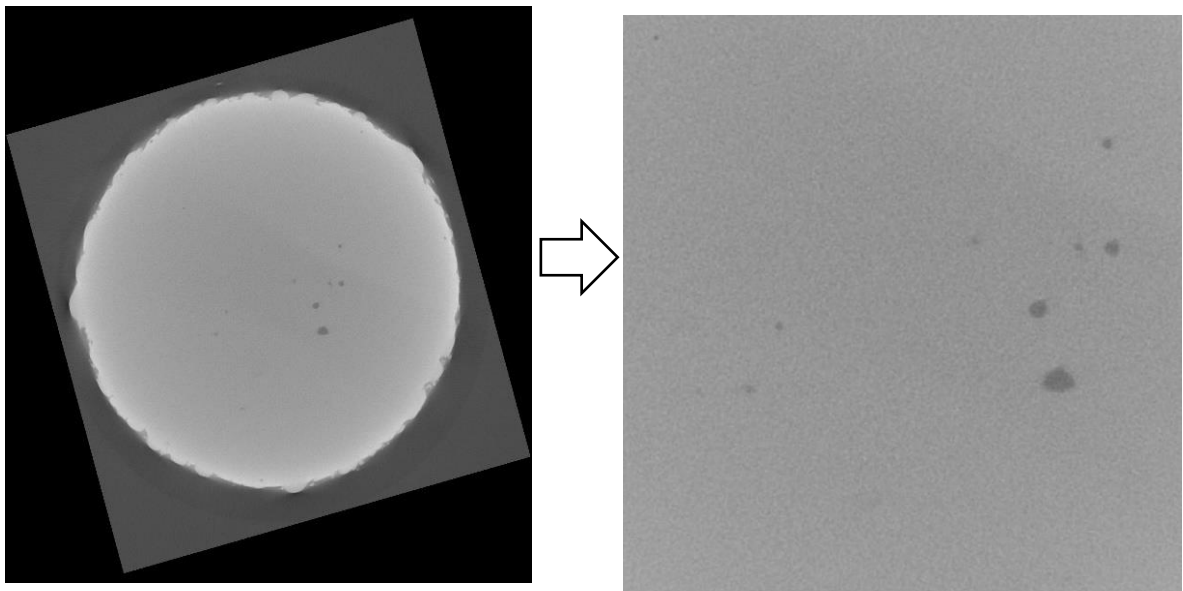


Figure 4.9: An Illustrative Example of the Real Existing Porosity Images (Feng et al., 2022)

A sample of 100 slices of the real images is taken and compared with the created artificial porosity images in the previous section using a quantitative method called Structural Similarity Index (SSI). It is a measure that assesses the images based on three computational terms: the luminance (l), the contrast (c), and the structural terms (s), multiplying three terms results in the overall index. If two images are exactly the same, the index is 1 and if they are totally different the index is close to zero. (MathWorks-9) described the mathematical equations to calculate SSI for images x and y:

$$SSI(x,y) = [l(x,y)]^\alpha \cdot [c(x,y)]^\beta \cdot [s(x,y)]^\gamma \quad \text{(Equation 4.3)}$$

Where:

$$l(x,y) = (2\mu_x \mu_y + C1) / (\mu_x^2 + \mu_y^2 + C1) \quad \text{(Equation 4.4)}$$

$$c(x,y) = (2\sigma_x \sigma_y + C2) / (\sigma_x^2 + \sigma_y^2 + C2) \quad \text{(Equation 4.5)}$$

$$s(x,y) = (\sigma_{xy} + C3) / (\sigma_x \sigma_y + C3) \quad \text{(Equation 4.6)}$$

where μ_x and μ_y are local means, σ_x and σ_y are the standard deviations for images x and y, and σ_{xy} is cross-covariance for images x and y. The exponents for luminance, contrast and structural are alpha(α), beta(β), and gamma(γ), while C1, C2, and C3 are constants added to avoid instability for image regions where the local mean or standard deviation is close to zero.

MATLAB platform is used to calculate the structural similarity index between the real and first version of artificial images (Appendix 1.5), the first index is 0.7085 for the following two images in figure 4.10:

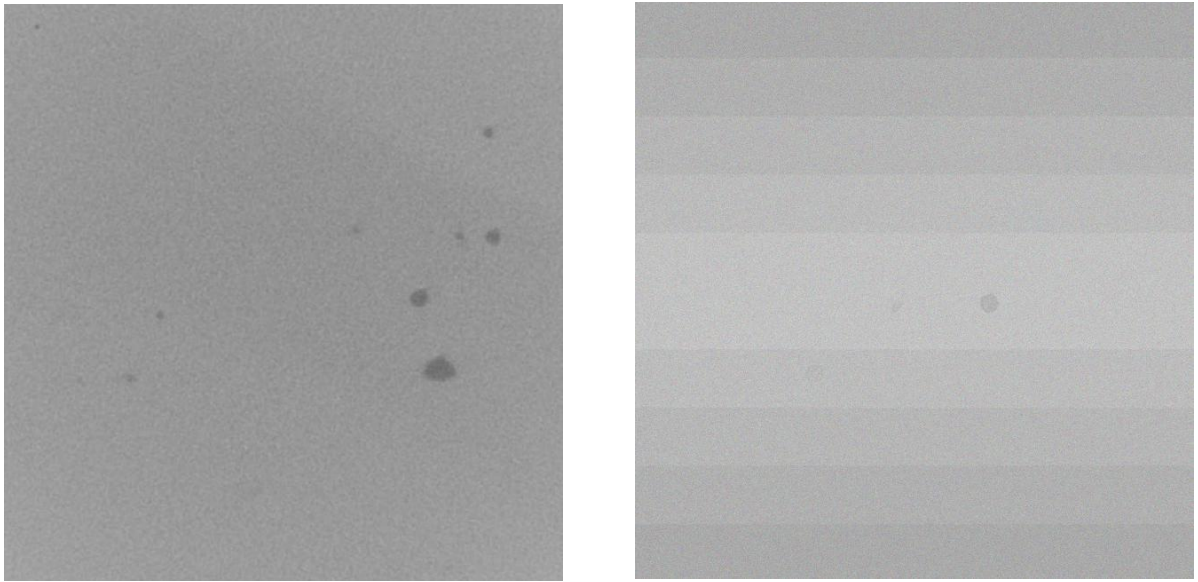


Figure 4.10: Real Image (Left) Vs First Version of Artificial Image (Right)

In order to improve the SSI, the average grayscale value of the artificial image is reduced to 152 instead of 182 since the average grayscale value for the real image is 152, in addition, the mean and variance of gaussian noise were reduced from 0.001 to 0.00005

after many trials to reach the best SSI value which becomes 0.8764 for the following two images in figure 4.11:

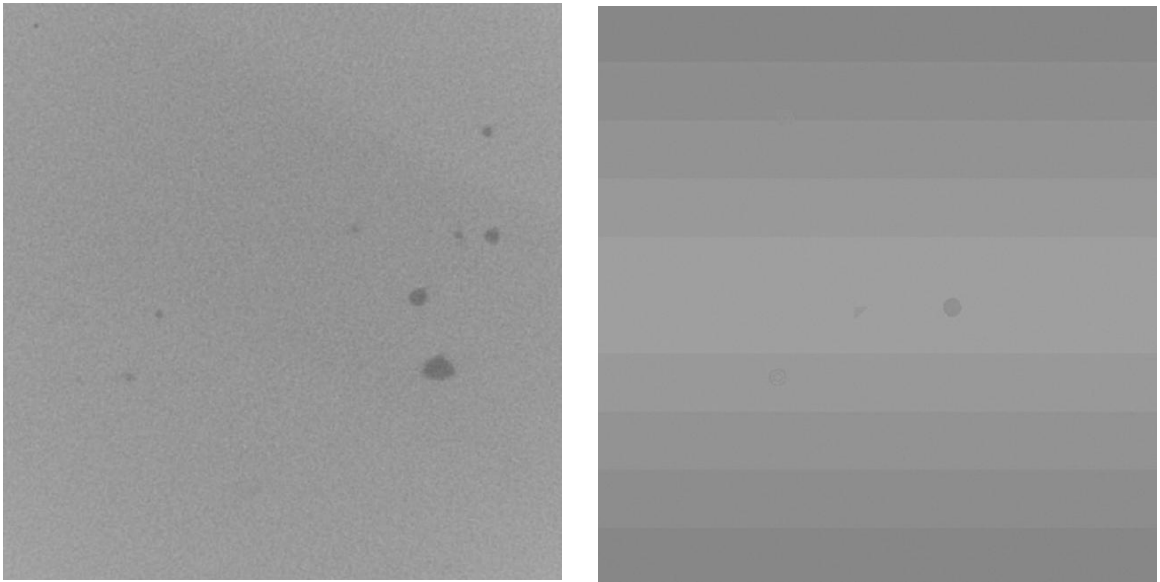


Figure 4.11: Real Image (Left) Vs Improved First Version of Artificial Image (Right)

Furthermore, the SSI algorithm is applied to the second version of artificial porosity images with normally distributed porosity positions and backgrounds of real images, the following figure 4.12 shows real and artificial images:

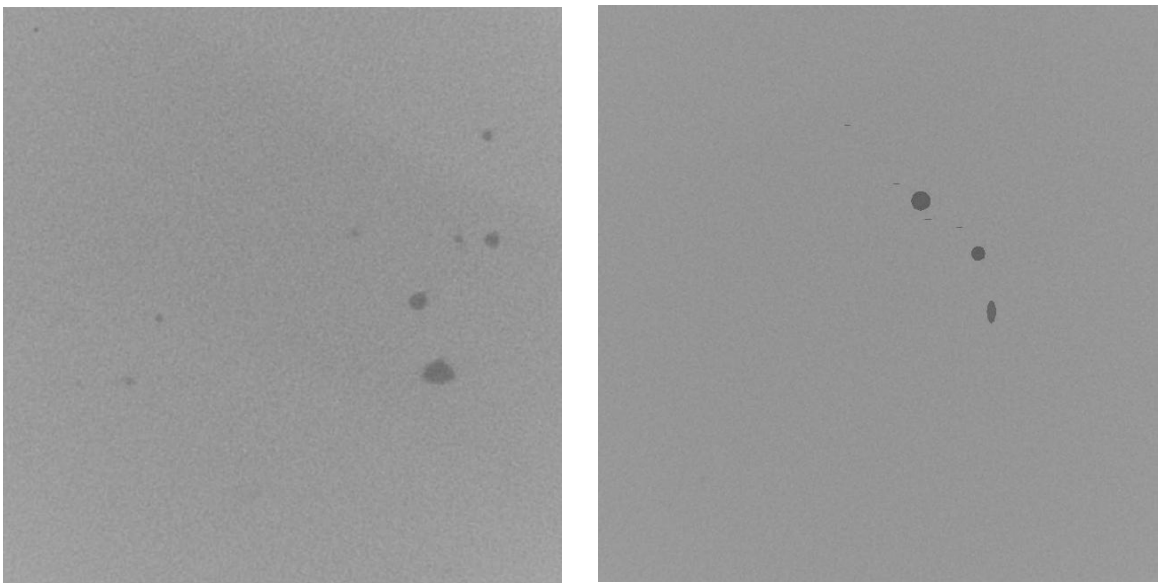


Figure 4.12: Real Image (Left) Vs Second Version of Artificial Image (Right)

A sample of 100 real images and 100 slices for each of the 30 cubes are compared, the average similarity index for 3000 slices is 0.9586. The same sample of 100 real images are used to compare them with the final version of artificial porosity images shown in the following figure 4.13:

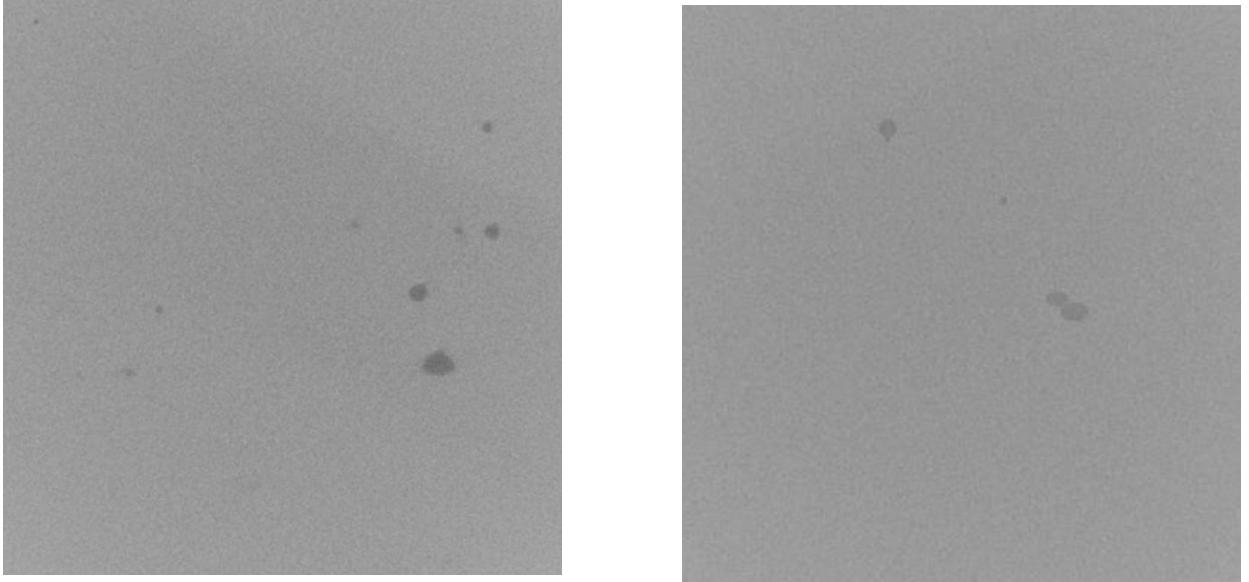


Figure 4.13: Real Image (Left) Vs Final Version of Artificial Image (Right)

The average similarity index for 3000 slices is improved to 0.9967 which means that the final version of artificial porosity images mimics better real images since the index is closer to 1.

The created artificial porosity images in this chapter will be used only to test the proposed RCNN algorithms that will be developed in the next chapter 5. The study is not aimed to study the porosities in depth, so no experiments have been conducted, but the next chapter proposes a method that will enhance such studies, particularly in predicting the percent of porosity as will be demonstrated in the next chapter. In the future, real experiments can be conducted in order to produce real porosity images.

4.3 Summary

This chapter proposed a new efficient approach of creating artificial keyhole porosity images mimicking the real CT scan slices of finished SLM parts that can be used in the research environment effectively and efficiently to validate the training of an accurate RCNN for the automatic prediction of porosity in CT scans of SLM parts. Training CNN

algorithms would require a large amount of experimental data which can be expensive for SLM parts (www.facfox.com, 2022) since there are many types of production costs for pre-processing, processing and post processing including preparing geometry data, CAD model, machine setup, material cost, and building up the part (Rickenbacher, 2013). Thus, producing large amount of porosity images to train RCNN is not cost-effective.

The steps for creating the artificial porosity images are:

- Establishing regression equations
- Generating pores number and diameter
- Creating 3D cubes
- Slicing 3D cubes into 2D images
- Labelling 2D slices
- Adding noisy background

This chapter's contribution which will be continued in the following chapter is:

- Developing a new approach to create a large amount of experimental artificial porosity images similar to real images by a similarity index of 0.9967 which can be used in the research environment efficiently and effectively in order to predict the percent of porosity in the finished SLM part using the developed hybrid CNN algorithms as will be described in the following chapter.

**Chapter 5: Hybrid Regression Convolutional Neural
Network for Predicting the Percent of Porosity in
Selective Laser Melting Parts**

5.1 Predicting the Porosity using the Existing Image Binarization Method

Using the artificial porosity images created in chapter 4, this chapter shows three methods for predicting the percent of porosity in the finished SLM parts. The first one is the existing image binarization method which is one of the main methods for measuring the porosity along with the Archimedes method (Arvieu et al., 2020). Archimedes's principle might be used in the case of producing real SLM parts, as this study uses artificial porosity images, so the image binarization method will be conducted and explained in this section. The second and third methods are Regression Convolutional Neural Network (RCNN) and hybrid Bees Regression Convolutional Neural Network (BA-RCNN) which will be explained in detail in section 5.2 and their results will be shown in section 5.3.

5.1.1 Porosity Assessment Gap using the Image Binarization Method

(Gong et al., 2019) mentioned that using CT scans of sample parts, there is a problem related to assessing accurately the porosity in the SLM parts. One main drawback is when using gray value analysis to assess the porosity of SLM parts visible in CT scan slices. The difficulty is the subjectivity in selecting an appropriate grayscale threshold that would convert a single slice into binary images highlighting defective regions, as well as determining the true level of porosity. For example, in figure 5.1 when an inappropriately low grayscale threshold is applied to the original slice image for binary image conversion, a certain amount of tiny undesired white spots are not filtered as shown in 5.1b. However, if a higher grayscale threshold is adopted, the morphological features of the defective area, specifically near the boundary, are altered dramatically as shown in figure 5.1c. These thresholds would result in significantly different predictions of porosity levels.

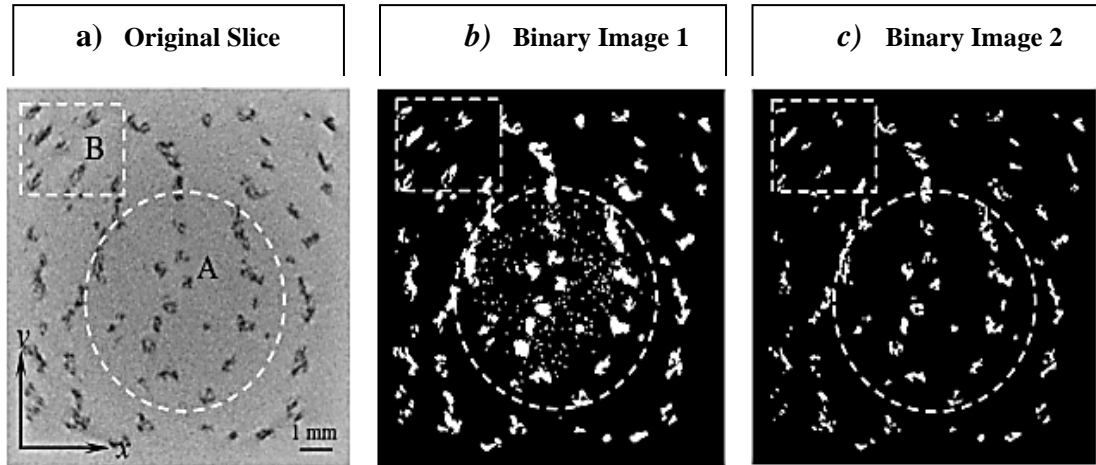


Figure 5.1: Original Slice and Binary Images (Gong et al., 2019)

To evaluate this issue further, the authors of this study (Gong et al., 2019) proposed an empirical method to estimate the porosity in the finished SLM parts using a naturalized threshold for CT scan slices. When compared with the Archimedes method, they found similar increasing or decreasing trends in the predicted percentage of porosity and concluded that the naturalized grayscale threshold is not the best solution to estimate the porosity in CT scan slices due to radiodensity variation and mutual influence of CT setup and it is difficult to compensate these variations with a unique grayscale (Gong et al., 2019). This limitation is found in one of the two main existing methods used in porosity estimation along with the Archimedes method as mentioned in (Arvieu et al., 2020).

5.1.2 Porosity Assessment Gap Validation

The second version of artificial porosity images created in the previous chapter are binarized using an adaptive thresholding algorithm in the MATLAB platform that selects the threshold based on local mean intensity in the pixel neighbourhood with a sensitivity factor between 0 and 1 that indicates the sensitivity toward thresholding more pixels as a foreground (MathWorks-10). The following figure 5.2 shows the artificial image along with the binarized image with sensitivity factor of 0.66:

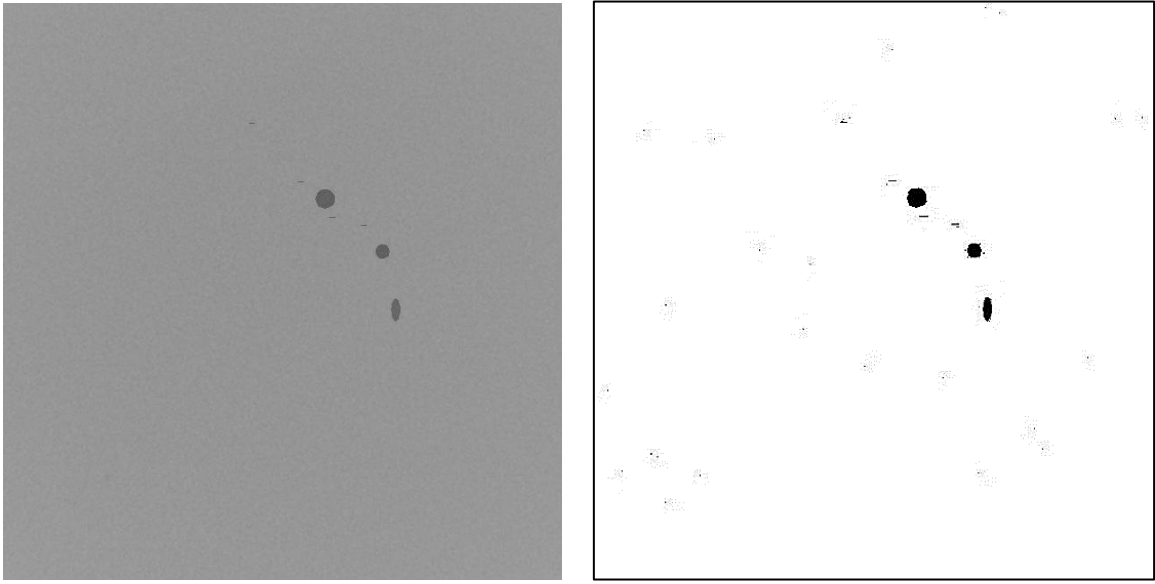


Figure 5.2: Second Version of Artificial Image (Left) Vs Binarized Artificial Image (Right)

As can be seen in figure 5.2, there are tiny undesired black spots in different positions in the binarized image, they appear also in slices with no pores as shown in the following figure 5.3:

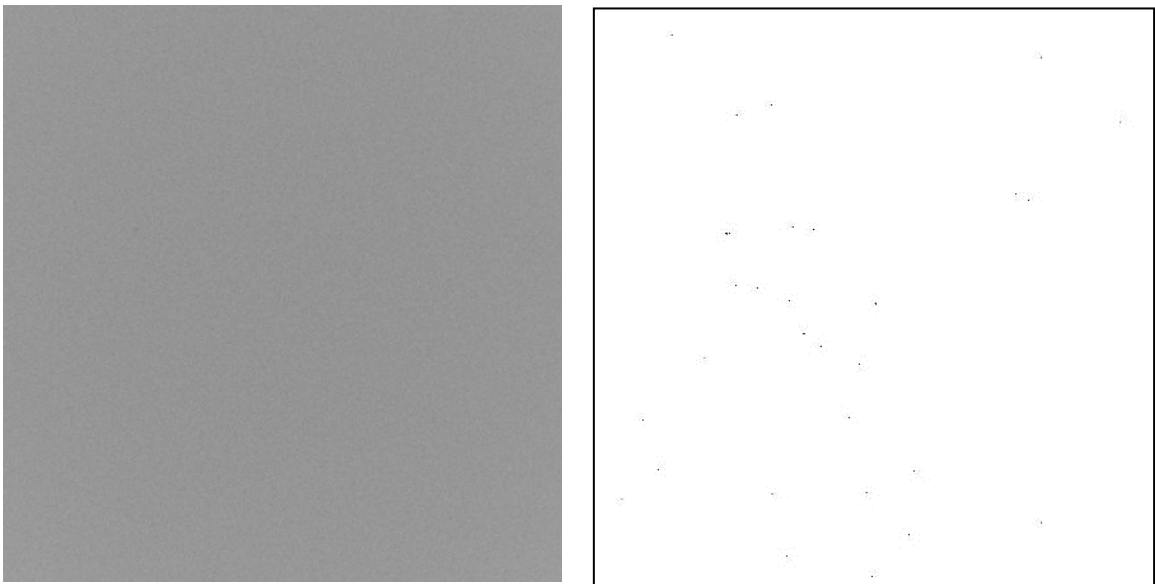


Figure 5.3: Second Version of Artificial Image with No Pore (Left) Vs Binarized Artificial Image (Right)

The actual percent of porosity for artificial images is calculated during the creation process by dividing the number of elements with specific pore unique pixel value by the total image size as mentioned previously in section 4.1.5, the average percent of the pore for 3000 slices of the second version is 0.0134 while the average percent of the pore for the 3000 binarized images is overestimated with a value of 0.0578, so the absolute error is 0.0444. The prediction accuracy with a difference less than the threshold of 0.02 (percent of observations with error less than 0.02) is 64.47%. The same approach is followed for the final version of artificial porosity images, the following figure 5.4 shows the artificial image along with the binarized image with sensitivity factor of 0.66:

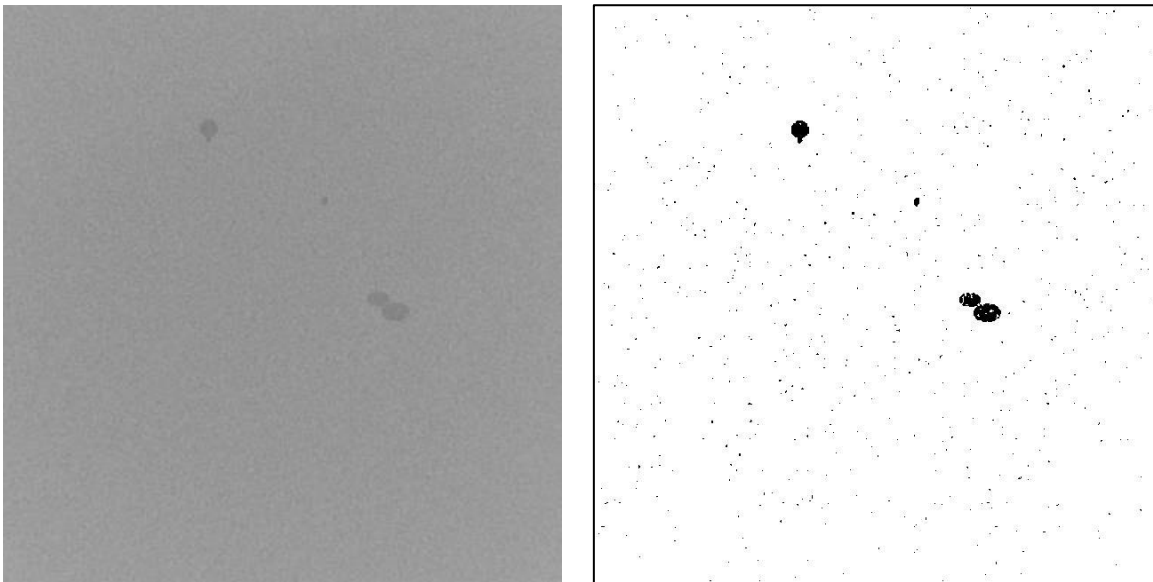


Figure 5.4: Final Version of Artificial Image (Left) Vs Binarized Artificial Image (Right)

As can be seen in figure 5.4, again there are tiny undesired black spots in different positions in the binarized image confirming the problem stated in (Gong et al., 2019), it is a result of inconsistent grayscale in CT scan slices because of the mutual influence of CT setup and radiodensity variation. Removing these spots using image processing is time consuming since they appear in different positions in each slice without a simple pattern, so the processing needs to be performed image by image to distinguish between the undesired black spots and the pores first. Then, unwanted spots need to be replaced with a white background because they are counted in the porosity calculation which overestimates the percent of porosity. Such an issue also occurs in the pores, as tiny white spots would

alter their morphological feature and slightly reduce the percent of porosity. However, the effect of this on porosity calculations is negligible when compared to the effect of the small black spots occurring in the large background. Depending on the sensitivity factor used, the black spots also appear in slices with no pores as shown in the following figure 5.5:

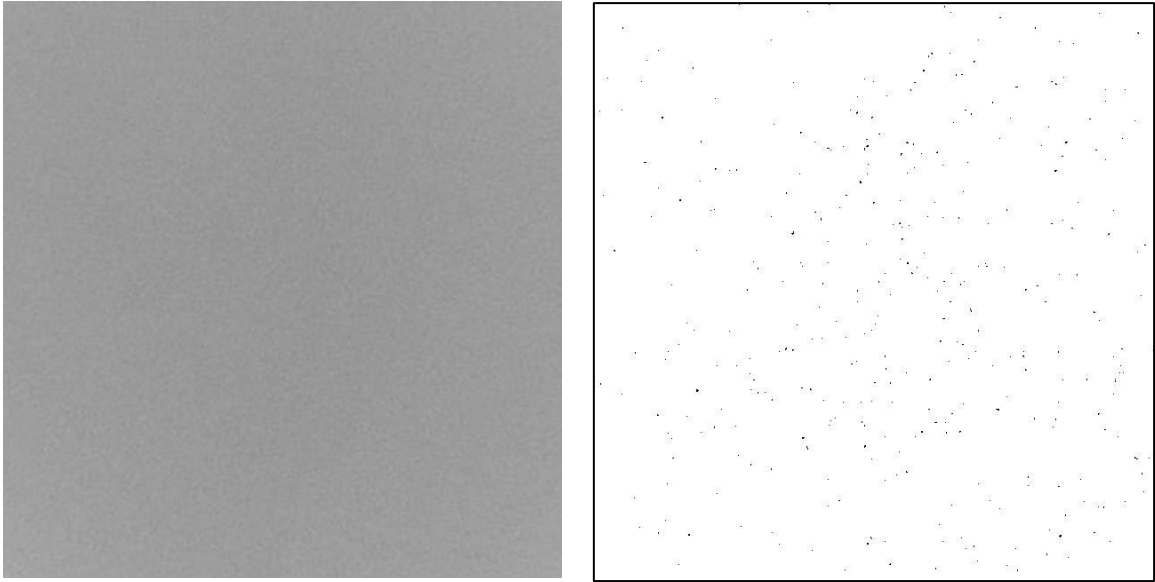
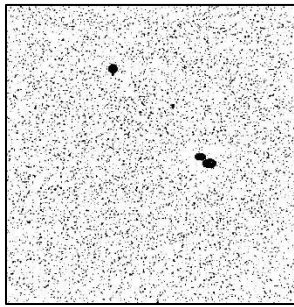
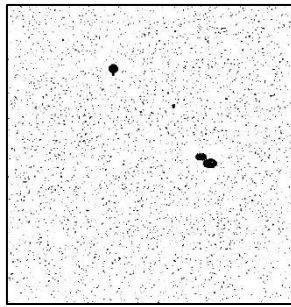


Figure 5.5: Final Version of Artificial Image with No Pore (Left) Vs Binarized Artificial Image (Right)

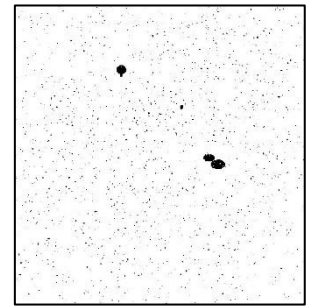
Experimenting different sensitivity factors between 0.63 to 0.74 resulted in the following binary images with different percent of porosity as shown in figure 5.6:



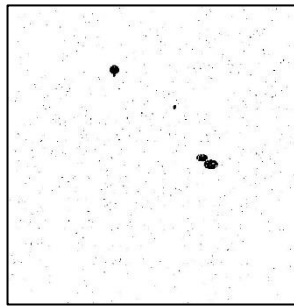
(a) 0.63, 7.89%



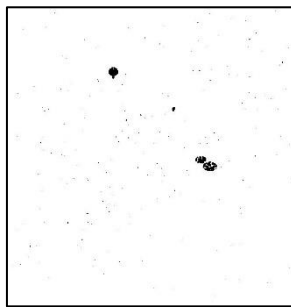
(b) 0.64, 3.21%



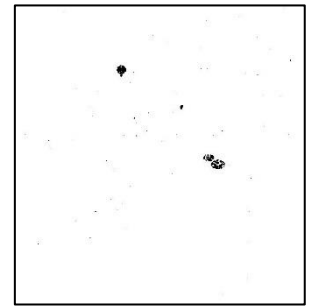
(c) 0.65, 1.26%



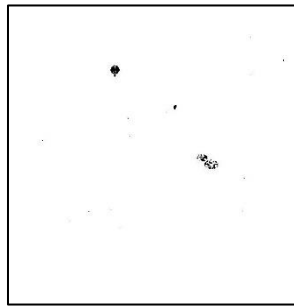
(d) 0.66, 0.56%



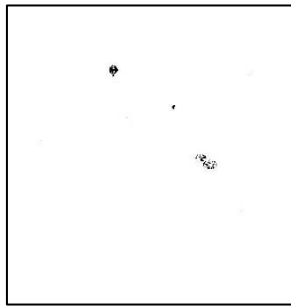
(e) 0.67, 0.32%



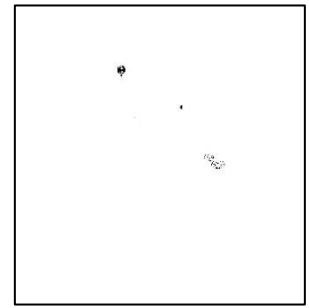
(f) 0.68, 0.24%



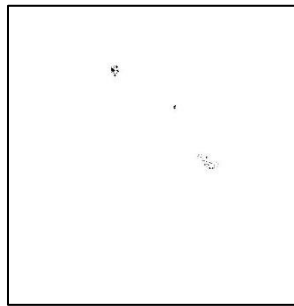
(g) 0.69, 0.17%



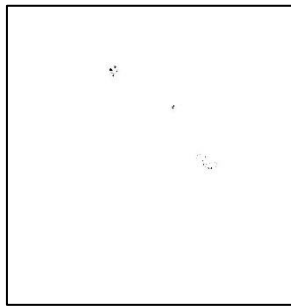
(h) 0.70, 0.13%



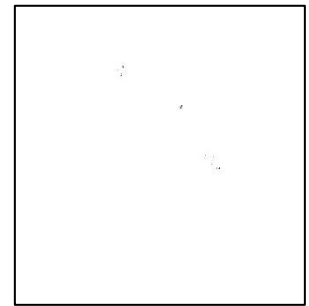
(i) 0.71, 0.09%



(j) 0.72, 0.05%



(k) 0.73, 0.03%



(l) 0.74, 0.01%

Figure 5.6: Binary Images with Different Sensitivity Factors and Percent of Porosity

As can be seen from the images, increasing the sensitivity factors reduces the black spots resulting from the noisy background, but it alters the pores' morphological feature dramatically. The sensitivity factor of 0.69 is relatively better and it will be selected to binarize all 3000 slices since it yielded the most accurate percent of porosity of 0.17% which is close to the actual percent of the pore for the original slice with a value of 0.1785%. It is worth noting that in the case of producing the real porosity images, sensitivity factor selection will be subjective and difficult since the actual percent of porosity will be unknown. The average percent of the pore for 3000 slices of the final version is 0.0203 while the average percent of the pore for 3000 binarized images is overestimated with a value of 0.0424, so the absolute error is 0.0221. The prediction accuracy with a difference of less than a threshold value of 0.02 (percent of observations with error less than 0.02) is 68.60%.

5.2 Predicting the Porosity using Hybrid Regression Convolutional Neural Networks

CNN can be used for regression problems to predict numerical values based on images, so regression CNN will be used to predict the percent of porosity based on the created artificial porosity images in the previous chapter. As mentioned previously (in section 4.1.5), the images are labelled using the actual percent of porosity calculated during the creation process by dividing the number of elements with specific pore unique pixel value by the total image size. CNN learns the porosity pattern in artificial porosity images that mimic CT scan images of the finished SLM part and predict the percent of porosity without the need for subjective difficult thresholding determination to convert a single slice into binary images which ultimately achieves automated quality assessment.

5.2.1 Regression Convolutional Neural Network Architecture

The CNN architecture for predicting the percent of porosity composes of 22 layers of which one input layer, five convolutional layers, five rectified linear unit layers, five batch normalization layers, four average pooling layers, one fully connected layer, and one regression layer.

The input layer is represented by a matrix of size height by width (650 x 630) that consists of pixel brightness numbers between 0 to 255 (0 for black and 255 for white). The

convolutional layers contain filters represented by matrix of weights that slide along the pixel brightness input matrix to create feature map matrix using special dot product as mentioned in (Hui, 2017). Rectified linear unit layers are used after each convolutional layer to increase the speed and effectiveness of the training by mapping negative values to zero and maintaining positive values as mentioned in (MathWorks-1). In addition, batch normalization layers are used as supplement layers after each convolutional layer to mitigate the risk of overfitting by normalizing the input values of the following layers (Yamashita et al., 2018). Pooling layers are used between the convolutional layers to reduce the dimensionality of the output volume (McDermott, 2021) without losing the important features which contribute to minimize the computational cost. Regression and fully connected layers are the output layers that show the predicted percent of the pores.

The normal number of convolutional layers to start is between two to three layers with a filter size of 3x3 or 5x5 as advised in (Hui, 2017), CNN is designed with five convolutional layers with four average pooling layers in between, the number of filters ranges between 8 for the first layer and 128 for the last one, each layer has twice number of filters of the previous layer (Brownlee, 2019). The filter size is 5x5, the section depth is 3, and the padding that helps to detect the edges of the images is set as 'same' so that the software calculates the size of the padding at the training time automatically and produce output size equal to the input size if the stride (number of pixel shift) is one.

The pooling type is 'average' which takes the average presence of the feature while the max pooling layer takes the most activated feature, so the average pooling is better with light background and max pooling is better with dark background as mentioned in (Ouf, 2017), the default size of pooling layer is 2x2 (Hui, 2017), but it yielded high computational cost since the image input size is big (650 x 630 x 3), the size was changed to 4x4 with stride value of 4 to minimize the training time.

The CNN is trained using SGDM which is the most common training algorithm (Yamashita et al., 2018), the default values for this algorithm are an initial learning rate of 0.01, a momentum of 0.9, a regularization of 1e-04, and the maximum number of epochs is 20 as presented in (MathWorks-6). After applying some experiments and monitoring the

validation accuracy, the momentum was changed to 0.8 and the regularization value became $1e-10$.

5.2.2 The Proposed Hybrid BA-BO-RCNN & BA-RCNN Algorithms

The hybrid regression CNN (BA-BO-RCNN and BA-RCNN) algorithms will train the artificial porosity images of the SLM part to predict the percent of porosity without the need for subjective difficult threshold determination to convert a single slice into binary images as mentioned in (Gong et al., 2019).

The following are the values of BA parameters which are assigned based on the computer capability of a single GPU with 256 GP and using the equations stated in (MathWorks-4).

- Maximum number of iterations = 1
- Number of scout bees (n) = 4
- Number of selected bees (m) = $0.5 \times n = 0.5 \times 4 = 2$
- Number of elite bees (e) = 1
- Number of recruited bees for elite (e) sites (nep) = $2 \times m = 2 \times 2 = 4$
- Number of recruited bees for other best (m-e) sites (nsp) = $0.5 \times n = 0.5 \times 4 = 2$
- Neighbourhood size for each selected patch (local search) (ngh) = $0.1 \times (\text{Var max} - \text{Var min})$
 - Section depth: $0.1 \times (3 - 1) = 0.1 \times 2 = 0.2$
 - Initial learning rate: $0.1 \times (1 - 1e-2) = 0.1 \times 0.99 = 0.099$
 - Momentum: $0.1 \times (0.98 - 0.8) = 0.1 \times 0.18 = 0.018$
 - Regularization: $0.1 \times (1e-2 - 1e-10) = 0.1 \times 0.01 = 0.001$
 - Weight learning rate factor: $0.1 \times (1.1 - 0.9) = 0.1 \times 0.2 = 0.02$

The following figures 5.7 and 5.8 summarizes the task for the hybrid algorithms:

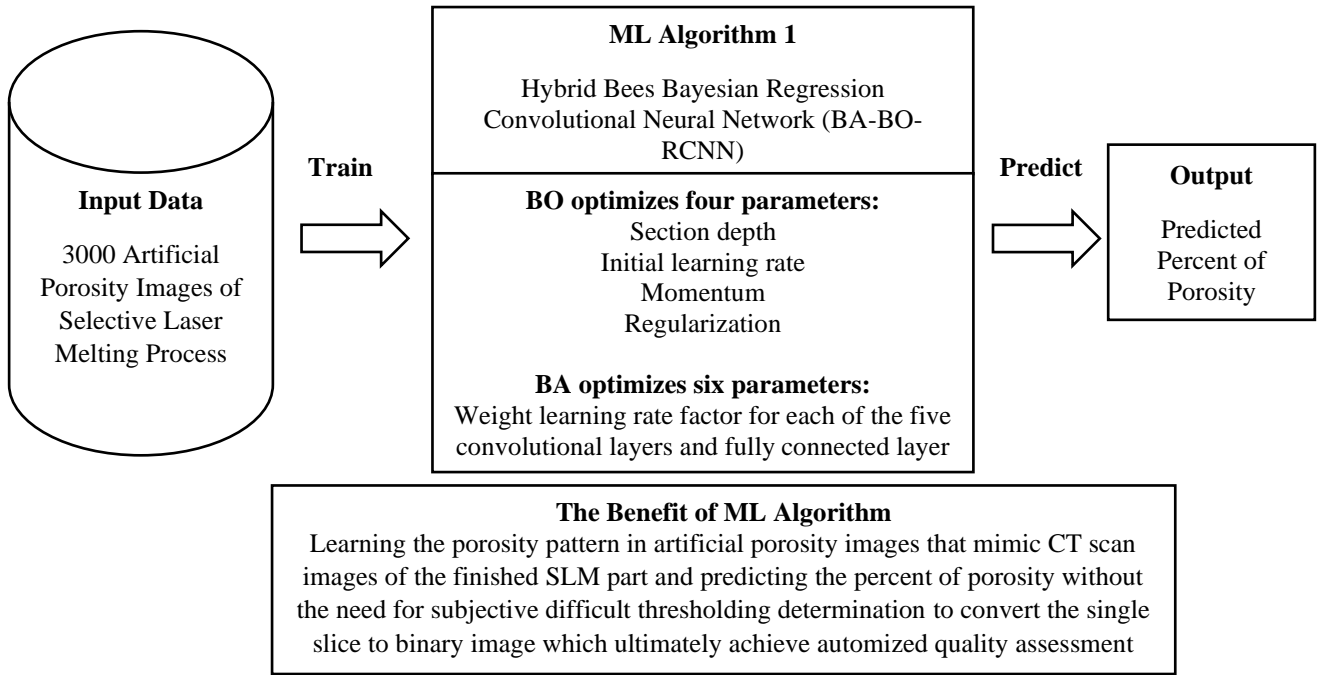


Figure 5.7: An Illustrative Diagram for Hybrid BA-BO-RCNN Algorithm

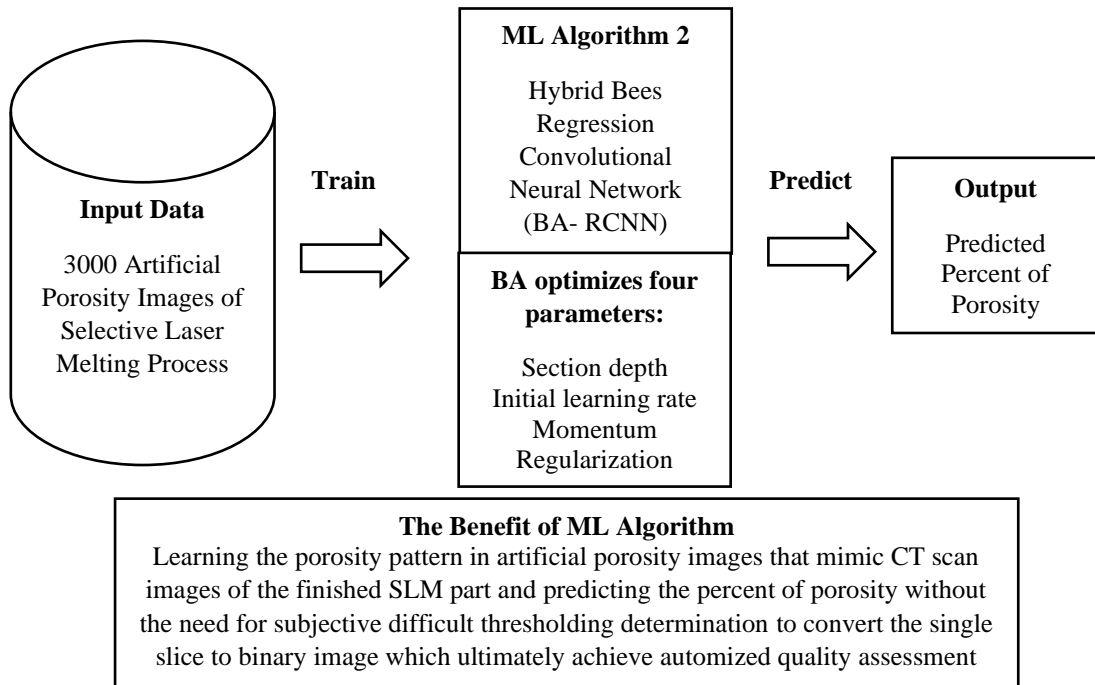


Figure 5.8: An Illustrative Diagram for Hybrid BA-RCNN Algorithm

The steps of MATLAB code for the hybrid regression CNN algorithms are shown in the following figure 5.9:

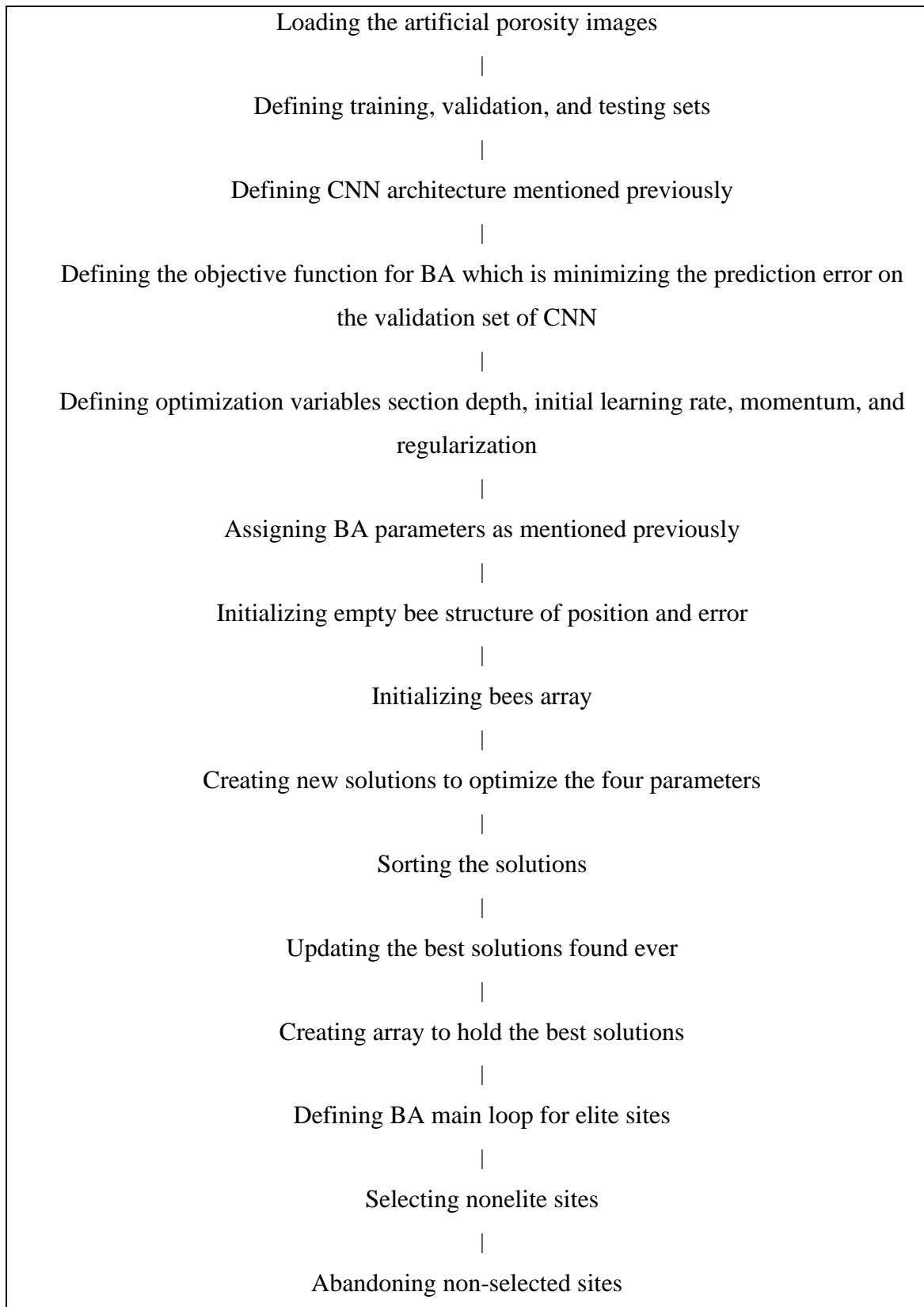


Figure 5.9: The Steps of MATLAB Code for Hybrid Regression CNN Algorithms

5.3 Results and Discussion for Hybrid Regression Convolutional Neural Networks

MATLAB software is used to apply the original Regression CNN (RCNN), hybrid Bayesian Regression CNN (BO-RCNN), hybrid Bees Bayesian Regression CNN (BA-BO-RCNN), and hybrid Bees Regression CNN (BA-RCNN) to the newly created 3,000 artificial porosity images of which 1800 images for training, 600 for each of the validation and testing sets to predict the percent of porosity without the need for subjective difficult thresholding determination to convert a single slice into binary images which ultimately achieve automatized quality assessment. The system configuration consists of a single GPU with a memory of 256 GB to be able to handle the 3000 artificial porosity images of size (650 x 630 x 3). As the DL algorithms require high computations, limited BA evaluations are applied to optimize CNN parameters. The datasets are shuffled at every epoch and the training, validation and testing sets are selected randomly after each shuffle in order to minimize the data biases and improve the validity of the experiments.

5.3.1 Results of Applying the Proposed Hybrid BA-BO-RCNN & BA-RCNN Algorithms to the Second Version of Artificial Porosity Images

First, the second version of artificial porosity images created in the previous chapter will be the input of the hybrid BA-BO-RCNN and BA-RCNN algorithms. The following table 5.1 shows the prediction accuracy with a difference less than a threshold of 0.02 (percent of observations with error less than 0.02) for training, validation, and testing sets along with the computational time for the best iteration for all algorithms:

Table 5.1: Prediction Accuracy and Computational Time for Algorithms (Second Version)

	Existing Original RCNN	Existing BO-RCNN	Novel Hybrid BA-BO-RCNN	Novel Hybrid BA-RCNN
Training Accuracy	81.22%	78.83%	80.11%	82.06%
Validation Accuracy	81.50%	79.67%	82.50%	81.67%
Testing Accuracy	78.83%	76.83%	80.17%	80.67%
Computational Time	18 Min 17 Sec	18 Min 14 Sec	15 Min 23 Sec	18 Min 13 Sec

The existing original RCNN produced 78.83% prediction accuracy in the testing set which is approximately 14% better than the accuracy resulting from the image binarization method with a value of 64.47% for the second version of artificial porosity images. The hybrid BO-RCNN has lower accuracy than the original RCNN with a value of 76.83%, so BO is not recommended to be used in this case as it performs poorly with a high dimensional objective function of more than 20 dimensions (www.stackexchange.com).

Adding BA to BO-RCNN improved the testing accuracy to 80.17% and it is the most efficient algorithm since it has lower computational time than other algorithms by approximately 3 minutes. The following table 5.2 shows the optimal weight learning rate factors optimized by BA:

Table 5.2: Optimal Weight Learning Rate Factors for BA-BO-RCNN (Second Version)

Optimized Variable using BA	Value
Optimal Weight Learning Rate Factor for Convolutional Layer 1	1.0629
Optimal Weight Learning Rate Factor for Convolutional Layer 2	1.0812
Optimal Weight Learning Rate Factor for Convolutional Layer 3	0.9254
Optimal Weight Learning Rate Factor for Convolutional Layer 4	1.0963
Optimal Weight Learning Rate Factor for Convolutional Layer 5	1.0265
Optimal Weight Learning Rate Factor for Fully Connected Layer	0.9195

The improved testing accuracy is a result of adjusting the global learning rate of 0.010041 obtained by BO by multiplying it by 1.0629 in the first convolutional layer to become 0.0106. In the second convolutional layer, the adjustment factor is 1.0812 resulting in a learning rate value of 0.008, the factors for the third, fourth and fifth convolutional layers are 0.9254, 1.0827, and 1.0265, so the adjusted values are 0.0092, 0.0108, and 0.0103

respectively. Finally, the new learning rate value for the fully connected layer is 0.0092 after adjusting it by a factor of 0.9195.

The novel hybrid BA-RCNN produced the best prediction accuracy with a value of 80.67% in the testing set. The following table 5.3 shows the optimal CNN parameters values optimized by BA:

Table 5.3: Optimal CNN Parameters Values for BA-RCNN (Second Version)

Optimized Variable using BA	Value
Optimal Section Depth	3
Optimal Initial Learning Rate	0.0118
Optimal Momentum	0.8229
Optimal Regularization	0.0091

The best-achieved prediction accuracy of 80.67% is a result of a section depth value of 3, an initial learning rate of 0.0118, a momentum of 0.8229 and a regularization of 0.0091.

5.3.2 Results of Applying the Proposed Hybrid BA-BO-RCNN & BA-RCNN Algorithms to the Final Version of Artificial Porosity Images

The same approach is followed to predict the percent of porosity for the final version of artificial porosity images with a difference less than a threshold value of 0.02 (percent of observations with error less than 0.02), the following table 5.4 shows the prediction accuracy for training, validation, and testing sets along with computational time for best iteration for all algorithms:

Table 5.4: Prediction Accuracy and Computational Time for Algorithms (Final Version)

	Existing Original RCNN	Existing BO-RCNN	Novel Hybrid BA-BO-RCNN	Novel Hybrid BA-RCNN
Training Accuracy	74.94%	67.50%	82.61%	85.94%
Validation Accuracy	76%	66.67%	84.17%	87.33%
Testing Accuracy	75.50%	67.50%	83%	85.33%
Computational Time	15 Min 43 Sec	15 Min 38 Sec	15 Min 30 Sec	13 Min 40 Sec

The existing original RCNN produced a prediction accuracy of 75.50% in the testing set which is approximately 7% better than the accuracy resulted from the image binarization method with a value of 68.60% for the final version of artificial porosity images. The hybrid BO-RCNN has lower accuracy than original RCNN with a value of 67.50%, so it confirms the conclusion that BO is not recommended to be used in this objective function as it has more than 20 dimensions where BO performs poorly (www.stackexchange.com).

Adding BA to BO-RCNN improved the accuracy to 83%, the following table 5.5 shows the optimal weight learning rate factors optimized by BA:

Table 5.5: Optimal Weight Learning Rate Factors for BA-BO-RCNN (Final Version)

Optimized Variable using BA	Value
Optimal Weight Learning Rate Factor for Convolutional Layer 1	0.9703
Optimal Weight Learning Rate Factor for Convolutional Layer 2	0.9376
Optimal Weight Learning Rate Factor for Convolutional Layer 3	1.0859
Optimal Weight Learning Rate Factor for Convolutional Layer 4	0.9813
Optimal Weight Learning Rate Factor for Convolutional Layer 5	0.9993
Optimal Weight Learning Rate Factor for Fully Connected Layer	1.0982

The improved testing accuracy is a result of adjusting the global learning rate of 0.011964 obtained by BO by multiplying it by 0.9703 in the first convolutional layer to become 0.0116. In the second convolutional layer, the adjustment factor is 0.9376 resulting in a learning rate value of 0.0112, the factors for third, fourth and fifth convolutional layers are 1.0859, 0.9813, and 0.9993, so the adjusted values are 0.0123, 0.0117, and 0.0112 respectively. Finally, the new learning rate value for fully connected layer is 0.0131 after adjusting it by a factor of 1.0982.

The novel hybrid BA-RCNN produced the best prediction accuracy with a value of 85.33% in the testing set. In addition, the hybrid BA-RCNN is better as well in terms of cost-effectiveness since it has a lower computational time by approximately 2 minutes. The following table 5.6 shows the optimal CNN parameters values optimized by BA:

Table 5.6: Optimal CNN Parameters Values for BA-RCNN (Final Version)

Optimized Variable using BA	Value
Optimal Section Depth	1
Optimal Initial Learning Rate	0.0101
Optimal Momentum	0.9568
Optimal Regularization	0.0097

The best achieved prediction accuracy of 85.33% is a result of a section depth value of 1, initial learning rate of 0.0101, momentum of 0.9568 and regularization of 0.0097.

5.3.3 Sensitivity to Noise in Artificial Porosity Images

This section compares between the image binarization method and the hybrid BA-RCNN algorithm in terms of sensitivity to noise in the artificial porosity images. Two different levels of noise are added to the final version of created artificial porosity images. As described in section 4.1.6, the artificial images were overlaid with the noisy background of real images. So, the noise is determined based on the degree of overlaying which is a factor with a value between 0 and 1 (MathWorks-8). Thus, with a factor of 1, the pore image is shown without noisy background and with a factor of 0, the noisy background is shown without pores. A factor of 0.125 was selected in section 4.1.6 to visually be the best value that combined both pore and noisy background. In this section, two more factors are arbitrarily selected which are 0.2 and 0.25 resulting in more noisy images. The following two figures 5.10 and 5.11 show illustrative examples of the noisy slices along with the binarized images:

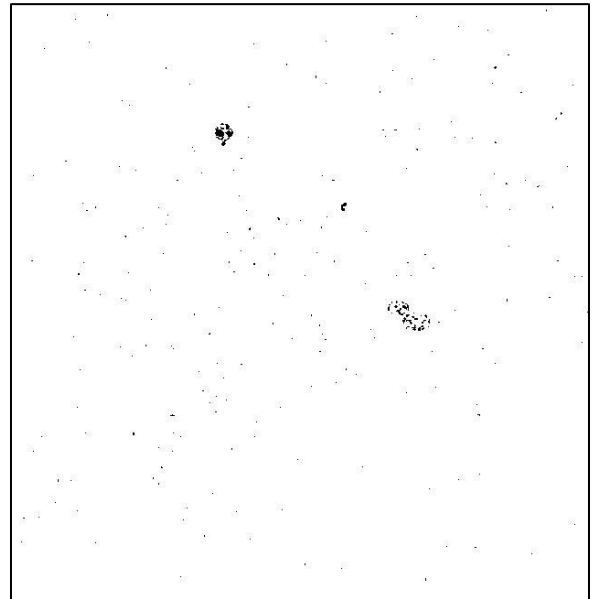
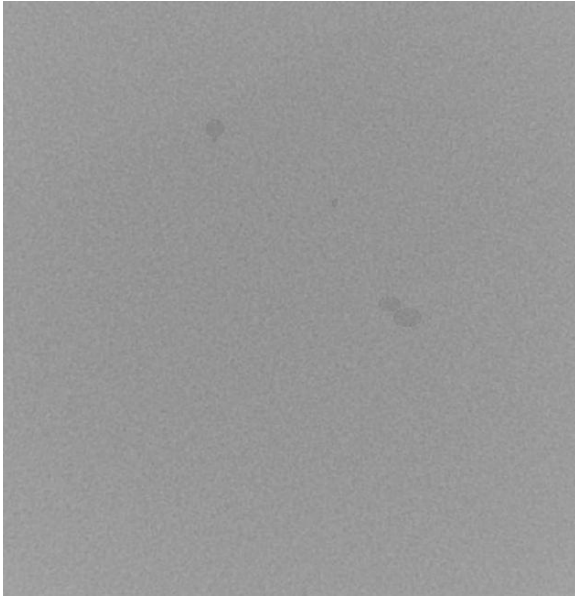


Figure 5.10: Artificial Image with Noise of 0.2 (Left) Vs Binarized Artificial Image (Right)

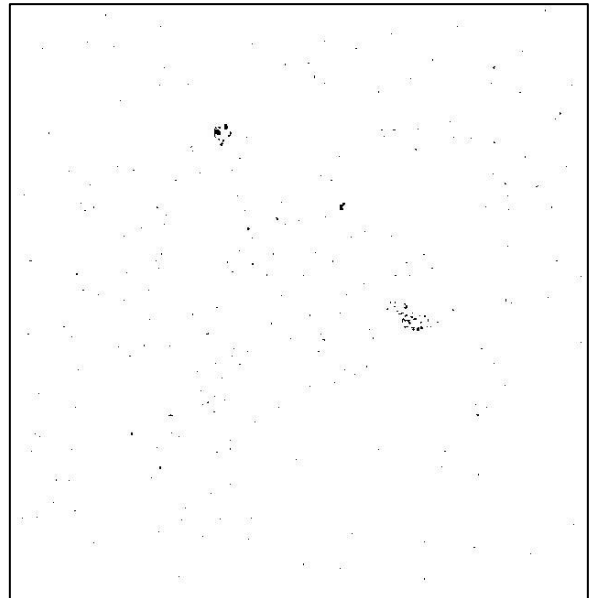
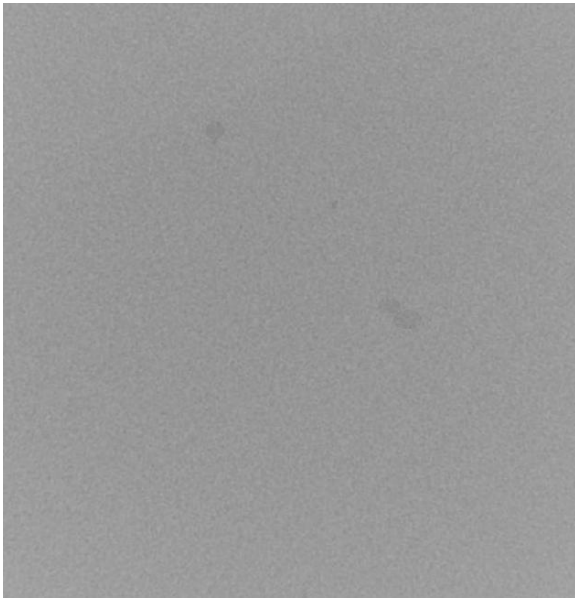


Figure 5.11: Artificial Image with Noise of 0.25 (Left) Vs Binarized Artificial Image (Right)

Applying the image binarization method to 3000 slices with three different levels of noise (0.125, 0.2 and 0.25) results in the following table 5.7 that shows the prediction accuracy for the percent of porosity:

Table 5.7: Prediction Accuracy using Image Binarization Method with Different Levels of Noise

Level of Noise	Prediction Accuracy
0.125	68.60%
0.2	40.73%
0.25	26.6%

Similarly, the hybrid BA-RCNN algorithm is applied to artificial porosity images with the same three levels of noise. The following table 5.8 shows the testing accuracy for predicting the percent of porosity:

Table 5.8: Prediction Accuracy using BA-RCNN Algorithm with Different Levels of Noise

Level of Noise	Prediction Accuracy
0.125	85.33%
0.2	79.21%
0.25	77.33%

To clearly investigate the variations in the prediction accuracy of both techniques, the following figure 5.12 is created which presents the accuracies at different levels of noise:

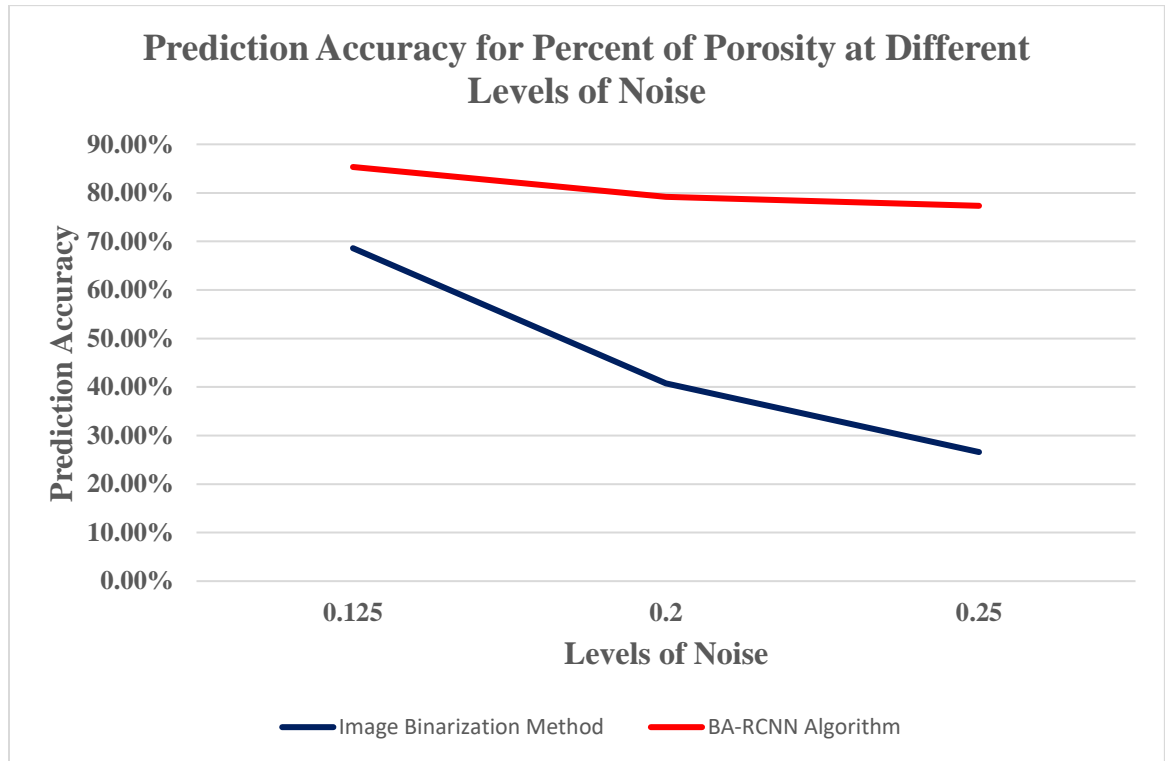


Figure 5.12: Prediction Accuracy for Percent of Porosity at Different Levels of Noise

As can be seen in figure 5.12, the difference between the minimum and maximum accuracy in the BA-RCNN algorithm is less than the range of accuracies in the image binarization method, so it is concluded that the BA-RCNN algorithm is less sensitive to noise. The convolutional filters in CNN that slide along the pixel brightness input matrix can deal with the level of noise in the artificial porosity images so that the creation of feature map matrix is not affected (Hui, 2017). On the other hand, the image binarization method is affected by the level of noise as increasing the sensitivity factors to binarize the images reduces the black spots resulted from the noisy background, but it alters the pores morphological feature dramatically resulting in underestimating the percent of porosity. If the sensitivity factor is decreased, the pore morphology is not significantly changed, but the black spots appear more in the binarized images which overestimate the percent of porosity. So, these thresholds would result in significantly different predictions of porosity levels.

The following chapter will show a better DL algorithm in dealing with sequential layers of the SLM part, which is the LSTM network, it will be hybridized with CNN that extracts

the features in the artificial porosity images for feeding it into LSTM network that considers the long-term dependencies.

5.4 Summary

Using the artificial images created in chapter 4, this chapter showed three methods for predicting the percent of porosity in SLM parts. The first one is the existing image binarization method which is one of the main methods for measuring the porosity along with the Archimedes method (Arvieu et al., 2020). Archimedes's principle might be used in the case of producing real SLM parts, as this study used artificial porosity images, so the image binarization method was conducted. The second and third methods were RCNN and BA-RCNN.

(Gong et al., 2019) mentioned that using CT scans of sample parts, there is a problem related to assessing accurately the porosity in SLM parts. One main drawback is when using gray value analysis to assess the porosity of SLM parts visible in CT scan slices. The difficulty is the subjectivity in selecting an appropriate grayscale threshold that would convert a single slice into binary images highlighting defective regions, as well as determining the true level of porosity. When an inappropriately low grayscale threshold is applied to the original slice for binary image conversion, a certain amount of tiny undesired white spots is not filtered. However, if a higher grayscale threshold is adopted, the morphological features of the defective area, specifically near the boundary, are altered dramatically. These thresholds would result in significantly different predictions of porosity levels.

So, in this chapter, an intelligent method based on CNN was used for the regression problem to predict the percent of porosity in the finished SLM part without the need for subjective difficult threshold determination to convert the single slice to binary image. Applying RCNN on 3000 slices of the second version of artificial porosity images, similar to real CT-scan images by a similarity index of 0.9586, improved porosity prediction accuracy from 64.47% for image binarization method to 78.83% for RCNN, while integrating BA produced the best prediction accuracy with a value of 80.67% which is approximately 16% better than existing image binarization method. The following table 5.9 summarizes the prediction accuracy:

Table 5.9: Porosity Prediction Accuracy (Second Version)

	Image Binarization	Original RCNN	Novel Hybrid BA-RCNN
Porosity Prediction Accuracy	64.47%	78.83%	80.67%

Applying RCNN on 3000 slices of the final version of artificial porosity images, similar to real CT-scan images by a similarity index of 0.9967, improved porosity prediction accuracy from 68.60% for image binarization method to 75.50% for RCNN, while integrating BA produced the best prediction accuracy with a value of 85.33% which is approximately 17% better than existing image binarization method. The following table 5.10 summarizes the prediction accuracy:

Table 5.10: Porosity Prediction Accuracy (Final Version)

	Image Binarization	Original RCNN	Novel Hybrid BA-RCNN
Porosity Prediction Accuracy	68.60%	75.50%	85.33%

The contribution made in chapter 5 by combining the outcome of chapter 4 is :

- Proposing and validating a new approach for predicting the percent of porosity in the finished SLM parts, using hybrid Bees Regression Convolutional Neural Network (BA-RCNN). It was demonstrated that a better accuracy than the existing image binarization method could be achieved (approximately 17% improvement with the data set used). In order to test the algorithm, as the training of the RCNN would require a large amount of experimental data, artificial porosity images mimicking real CT scan slices of the finished SLM part were created with a similarity index of 0.9976 with real images.

**Chapter 6: Hybrid Long Short-Term Memory
Network with Bees Algorithm for Enhancing the
Porosity Prediction in Selective Laser Melting Parts**

6.1 Proposed Novel Hybrid BA-CNN-LSTM Algorithm

Improving the performance of the LSTM network is an ongoing challenge, the traditional approach of assigning the parameters using trial and error is less accurate since it depends on the user experience. However, using a nature inspired algorithm to automatically select the optimum parameters values may improve the performance of the network (Panwar et al., 2017). The existing studies addressed optimizing hidden layers, number of neurons, activation function, loss function, optimizer, batch size, and number of epochs using different nature inspired algorithms as was shown in section 2.4.1. This chapter addresses optimizing the learning rate in the forget, input, and output gates in addition to the cell candidate and fully connected layer so that each step has a customized learning rate for more optimum performance.

In particular, the weights update in each of the three gates, cell candidate, and fully connected layer depends on the learning rate value, so using a customized learning rate for each part would result in a more optimum update for the weights (Brownlee, 2020). This chapter addresses this issue by proposing a new approach that improves the performance of the LSTM network by optimizing the parameters related to the gates, cell state, and fully connected layer. The novel hybrid nature inspired algorithm adopts the BA to improve the performance of three gates and cell state, specifically by optimizing the learning rate factor so that each part has its learning rate determined based on the global learning rate. Having a more optimum learning rate means more optimum updates for the network weight (Brownlee, 2020).

As the bidirectional LSTM layer is used as will be explained later in section 6.1.2, which trains the input as is and on the reverse copy of the input (Brownlee, 2021), so four parameters are related for the forward side and the other four parameters for the backwards and one parameter related to fully connected layer. The following figure 6.1 illustrates the general framework for the proposed BA-CNN-LSTM algorithm, followed by subsections explaining each part in detail.

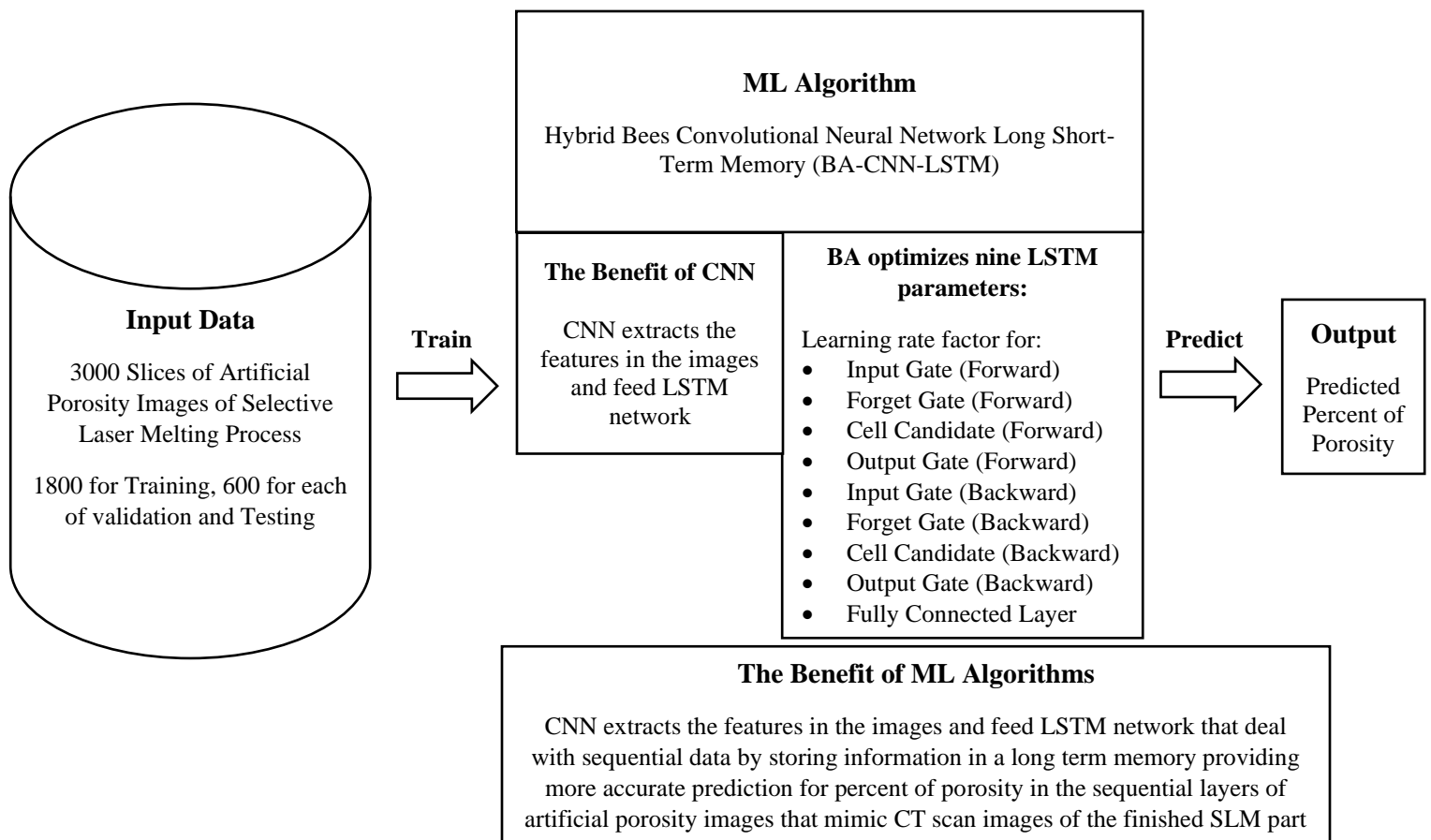


Figure 6.1: The General Framework for the Proposed BA-CNN-LSTM Algorithm

6.1.1 Convolutional Neural Network Architecture

The features in the artificial porosity images are extracted using CNN which uses filters in the convolutional layer to activate some features in the images. The architecture of CNN consists of 19 layers starting with five convolutional layers each accompanied by a rectified linear unit layer and batch normalization layer with four average pooling layers in between.

The filter in the convolutional layers is a matrix that contains weights, this matrix is slid along the input matrix that will be described in the next section and performs a special dot product which results in a feature map matrix (Hui, 2017). Each convolutional layer is accompanied by a rectified linear unit layer that maps the negative values to zero and keeps the positive values for faster effective training (MathWorks-1). Also, each convolutional layer is followed by a batch normalization layer to minimize the overfitting risk (Yamashita et al., 2018). The pooling layers are used between the convolutional layers to reduce the output dimensions with keeping the features of interest in the images (McDermott, 2021), this reduction contributes to minimizing the computational cost.

The normal filter size is 3 x 3 or 5 x 5 as advised in (MathWorks-1), after trying both sizes, the 5 x 5 size is selected in all five convolutional layers, and the numbers of filters are 8, 16, 32, 64 and 128 so each layer has twice the filter number of the previous layer as advised in (Brownlee, 2020). The edges of the images are detected using padding with a stride value of one, so the pixel shift is one cell. The pooling size is 4 x 4 with a stride value of 4 to minimize the computational cost since the image size is big (650 x 630 x 3). There are two types of pooling, average pooling calculates the average pixel brightness value of the four numbers in the pooling matrix. It is compatible with the lighter background, so it is used in the architecture rather than the max pooling type that selects the maximum pixel value taking the most activated feature making it better with the dark background (Ouf, 2017).

The most popular training algorithm is SGDM (Yamashita et al., 2018), so it is used to train on the artificial porosity images with 20 epochs (MathWorks-6), a section depth value of 1 to control the depth of the network, an initial learning rate of 0.0101 that allows for feature learning, a momentum of 0.9568 for parameters updating, and a regularization value of 0.0097 to mitigate the overfitting risk (MathWorks-3). They are the same set of parameters found using BA in the hybrid BA-RCNN algorithm developed in chapter 5.

The hybrid BA-RCNN algorithm was used to predict the percent of porosity in each layer of the SLM part. Since the layers are sequential, LSTM is added after the CNN to deal with sequential data for better prediction accuracy as will be described in the following subsection.

6.1.2 Long Short-Term Memory Architecture

The features extracted from the artificial porosity images using CNN feed the LSTM network that persists the information for a long period so that it can remember long-term dependencies (the-learning-machine). The mathematical equations for the three gates are shown in the following:

- Forget Gate

The first step in the LSTM cell is to decide if the information coming from the previous time scale is relevant to be remembered or irrelevant to be forgotten, it is based on the following forget gate equation (www.analyticsvidhya.com):

$$f_t = \sigma \times (X_t \times U_f + H_{t-1} \times W_f) \quad \text{(Equation 6.1)}$$

Where:

X_t is the current time cycle input

U_f is the input weight matrix

H_{t-1} is the previous time cycle hidden state

W_f is the hidden state weight matrix

Then, the sigmoid function shown in the following is applied resulting in a f_t value between 0 and 1 (MathWorks-2).

$$\sigma(x) = (1 + e^{-x})^{-1} \quad \text{(Equation 6.2)}$$

The f_t is multiplied by the cell state of the previous time cycle:

$$C_{t-1} \times f_t = 0 \text{ (Forget everything)} \quad \text{(Equation 6.3)}$$

$$C_{t-1} \times f_t = C_{t-1} \text{ (Forget nothing)} \quad \text{(Equation 6.4)}$$

- Input Gate

This gate is used to quantify the importance of the new information, it is based on the following input gate equation (www.analyticsvidhya.com):

$$i_t = \sigma \times (X_t \times U_i + H_{t-1} \times W_i) \quad \text{(Equation 6.5)}$$

Where:

X_t is the current time cycle input

U_i is the input weight matrix

H_{t-1} is the previous time cycle hidden state

W_i is the hidden state weight matrix

Similarly, the sigmoid function is applied resulting in i_t value between 0 and 1. Passing the information to the cell state is based on a function of the hidden state of the previous time cycle:

$$N_t = \tanh(X_t \times U_c + H_{t-1} \times W_c) \text{ (new information)} \quad \text{(Equation 6.6)}$$

Using the tanh function results in an N_t value between -1 and 1. If it is positive, it will be added to the cell state, and if it is negative, the information will be subtracted from the cell state as shown in the following equation:

$$C_t = f_t \times C_{t-1} + i_t \times N_t \text{ (updating cell state)} \quad \text{(Equation 6.7)}$$

- Output Gate

In this gate, the cell transfers the updated information from the previous time cycle to the next time cycle, it is based on the following output gate equation (www.analyticsvidhya.com):

$$O_t = \sigma \times (X_t \times U_o + H_{t-1} \times W_o) \quad \text{(Equation 6.8)}$$

Also, the O_t value is between 0 and 1 because of applying the sigmoid function. The current hidden state is a function of long-term memory and the output and it is calculated based on the following equation:

$$H_t = O_t \times \tanh(C_t) \quad \text{(Equation 6.9)}$$

The output of the current time cycle is found using the SoftMax function as shown in the following equation:

$$\text{Output} = \text{SoftMax}(H_t) \quad \text{(Equation 6.10)}$$

The output with the maximum score is the predicted value. (Thakur, 2018) presented a more intuitive architecture of LSTM as shown in figure 6.2:

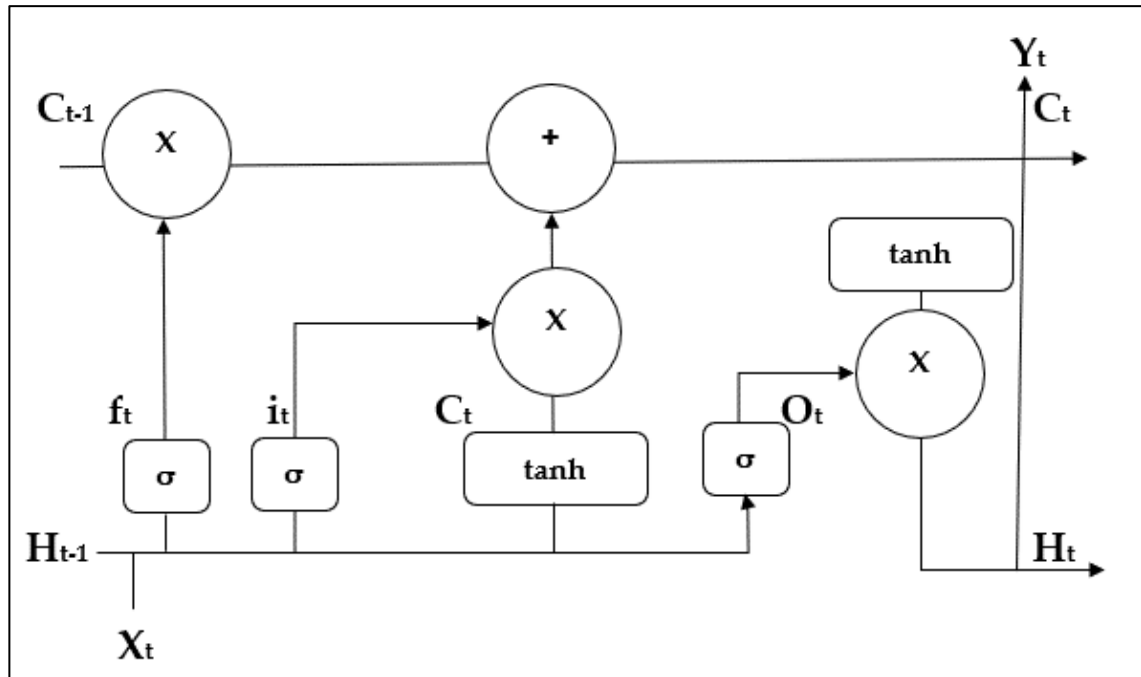


Figure 6.2: Intuitive Diagram of LSTM (Thakur, 2018)

The created artificial porosity images in chapter 4 consist of 30 3D cubes, each one was sliced into 100 2D sequential slices resulting in 3000 slices. So, there are 30 sequences with 100 layers in each sequence, 18 sequences are used for training and 6 sequences for each of the validation and testing sets.

The design of LSTM architecture consists of 8 layers, it starts with a sequence input layer that inputs the sequential data to the network, followed by a sequence folding layer that converts the image sequences to a batch of images so that the convolution operation described in the previous section can be performed on the layers independently (MathWorks-2). After folding, the convolution is applied to input data which is a matrix that contains the pixel value with a range between 0 – 255 where 0 represents the black regions and 255 is for the white regions, the size of this matrix depends on the image size which is 650 x 630 (Hui, 2017). After performing the convolution operations to extract the image features, the sequence unfolding layer is added to restore the sequence structure of the input data followed by a flatten layer that make collapsing for the input spatial dimensions to the channel dimension (MathWorks-2).

Then, the bidirectional LSTM layer is added which takes the input from both directions (forward and backward). The output of both the forward and backward at each stage is transferred to an activation layer (neural network), the output from this activation layer considers the relationship between the past and future layers (Mungalpara, 2021) which increases the prediction accuracy of the percent of porosity in each layer. The following figure 6.3 illustrates the way of working for the bidirectional LSTM layer:

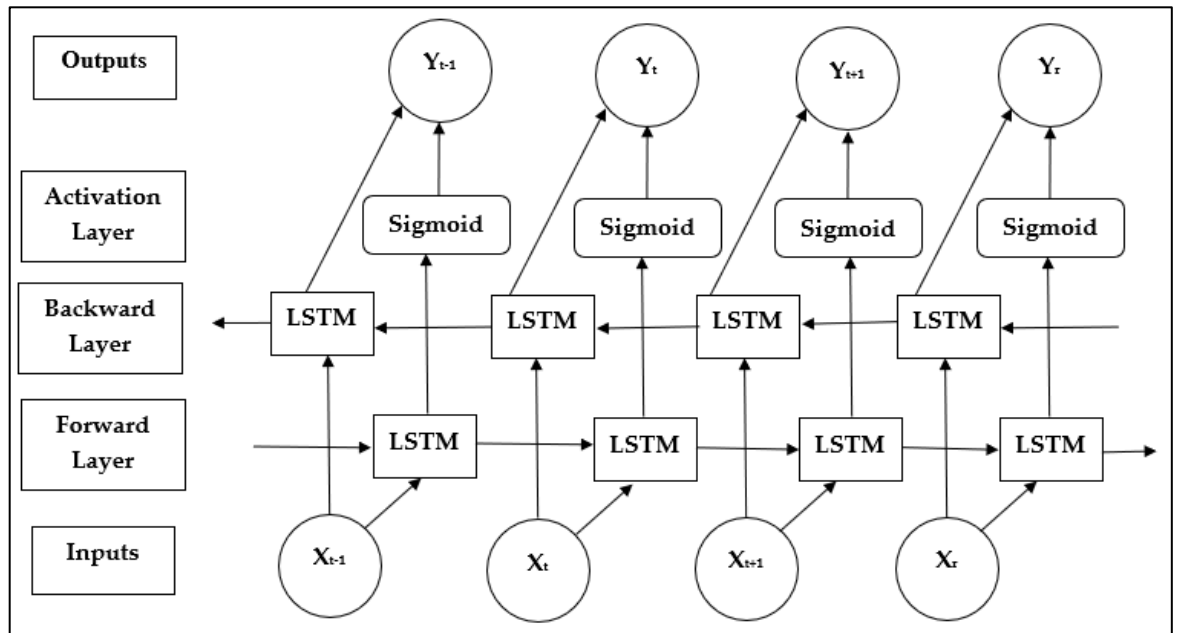


Figure 6.3: The Way of Working for the Bidirectional LSTM Layer (Newman, 2020)

The number of hidden units is 200, having more hidden units increase the computation cost without an improvement in the prediction accuracy, and also it increases the probability of overfitting (www.machinelearningmastery.com). The bidirectional LSTM layer is followed by a dropout layer which minimizes the risk of overfitting (www.machinelearningmastery.com), a fully connected layer is added with one predictor (percent of porosity) and the last layer is the regression layer.

6.1.3 The Novel Hybrid BA-CNN-LSTM Architecture Algorithm

BA is a swarm-based optimization technique that performs global and local searches to find the optimal solution. The following are the values of BA parameters which are assigned based on the capability of the computer and using the equations stated in (MathWorks-4).

- Maximum number of iterations = 1
- Number of scout bees (n) = 4
- Number of selected bees (m) = $0.5 \times n = 0.5 \times 4 = 2$
- Number of elite bees (e) = 1
- Number of recruited bees for elite (e) sites (nep) = $2 \times m = 2 \times 2 = 4$
- Number of recruited bees for other best (m-e) sites (nsp) = $0.5 \times n = 0.5 \times 4 = 2$
- Neighbourhood size for each selected patch (local search) (ngh) = $0.1 \times (\text{Var max} - \text{Var min}) = 0.1 \times (1.1 - 0.9) = 0.1 \times 0.2 = 0.02$

The BA process is used to find the optimal values for nine LSTM parameters in order to improve the performance of three gates, cell state and fully connected layer, particularly by optimizing the learning rate factor so that each part has its own learning rate determined based on the global learning rate. Having a more optimum learning rate means a more optimum update for the network weights described in the previous section (Brownlee, 2020). As the bidirectional LSTM layer is used, which trains the input as is and on the reverse copy of the input (Brownlee, 2021), so four parameters are related for the forward side and the other four parameters for the backwards and one parameter is related to the fully connected layer. So, the nine optimization variables are:

- Learning rate factor for input gate (Forward)
- Learning rate factor for forget gate (Forward)
- Learning rate factor for cell candidate (Forward)
- Learning rate factor for output gate (Forward)
- Learning rate factor for input gate (Backward)
- Learning rate factor for forget gate (Backward)
- Learning rate factor for cell candidate (Backward)
- Learning rate factor for output gate (Backward)
- Learning rate factor for fully connected layer

The optimal adjustment factors obtained by BA (ranging between 0.9 and 1.1) are multiplied by the global learning rate of 0.0101 resulting in a more optimum learning rate for each part stated above so that the network weights described in the previous section are updated with more optimum values in order to improve the performance of LSTM.

The steps of MATLAB code for the proposed hybrid BA-CNN-LSTM algorithm are presented in the following figure 6.4:

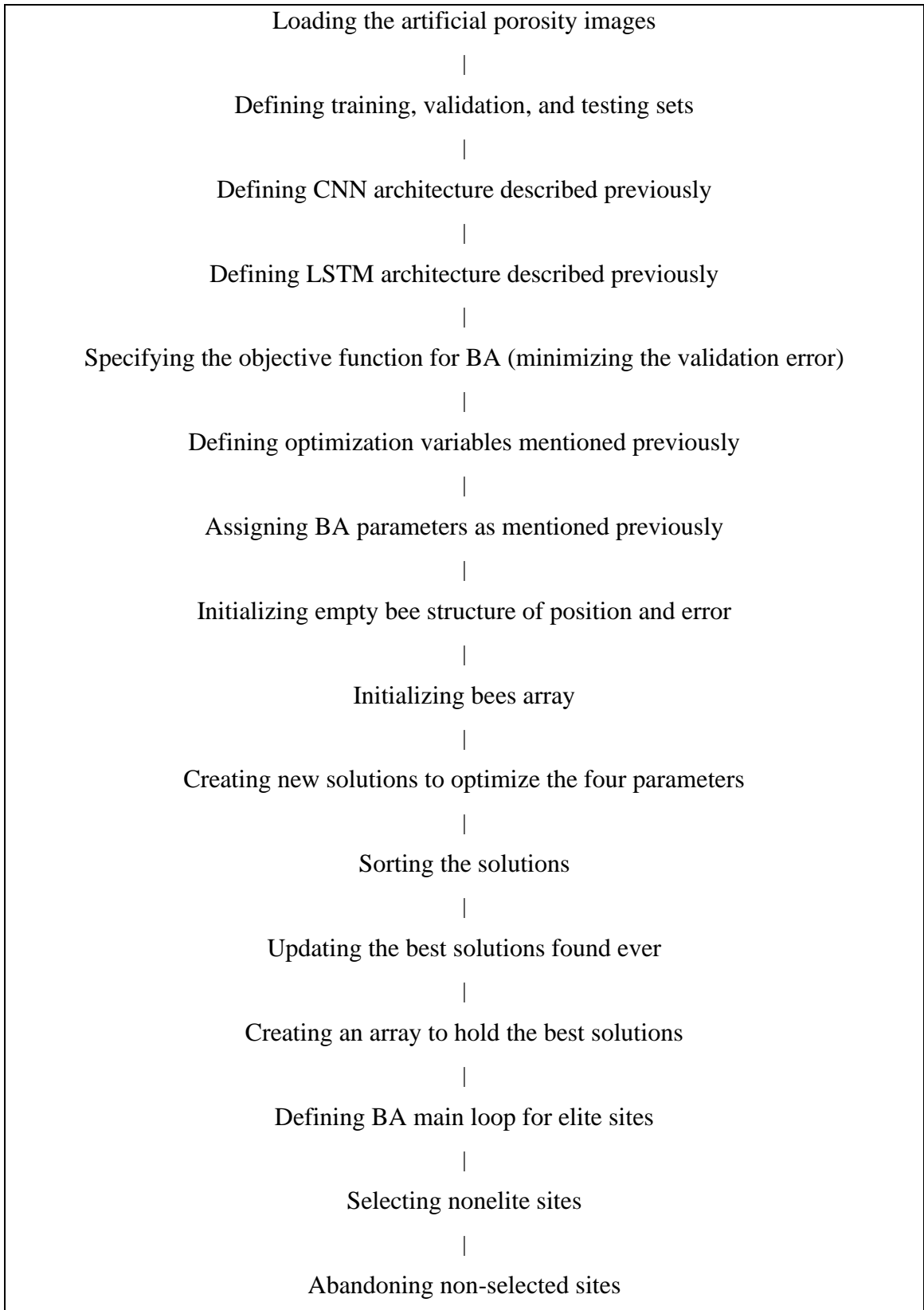


Figure 6.4: The Steps of MATLAB Code for the Proposed Hybrid BA-CNN-LSTM Algorithm

In addition, the following figure 6.5 shows a flow chart for the workflow diagram for the proposed hybrid BA-CNN-LSTM algorithm.

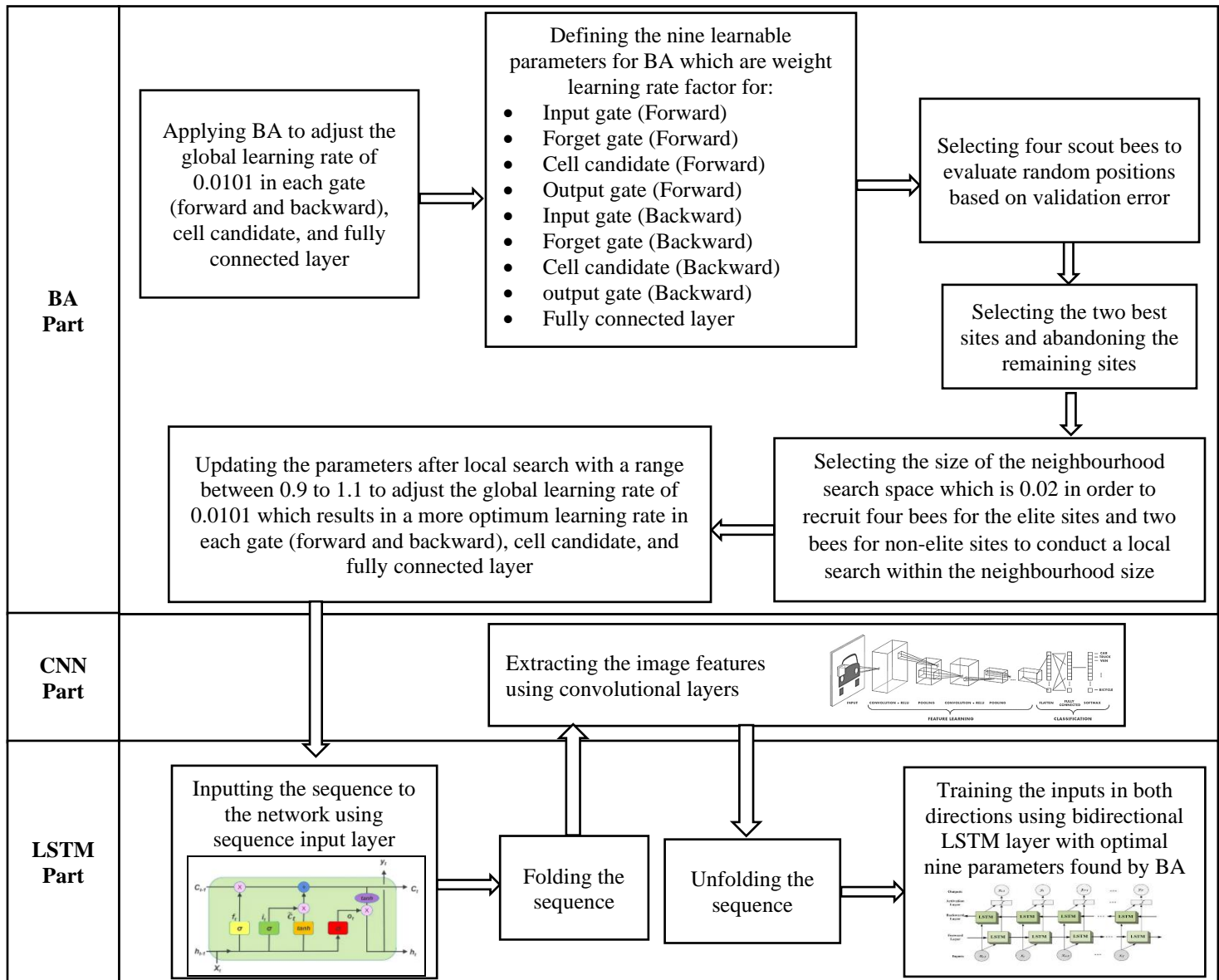


Figure 6.5: Workflow Diagram for the Proposed Hybrid BA-CNN-LSTM Algorithm

6.2 Results and Discussion

This section presents the results of applying the LSTM network on three sets of data, the first one is the artificial porosity images to predict the percent of porosity in the SLM part as mentioned previously. Also, it is applied in the signal processing context to classify Electrocardiogram (ECG) benchmark image data (MathWorks-7). In addition, the turbofan

engine degradation simulation dataset (MathWorks-11) is used to predict the remaining useful life (RUL) of engines using LSTM.

MATLAB platform is used to design the LSTM network, CNN and BA and apply the hybrid algorithms to three benchmark datasets. The system configuration consists of a single GPU with a memory of 256 GB to be able to handle the 3000 artificial porosity images of size (650 x 630 x 3). As the DL algorithms require high computations, limited BA evaluations are applied to optimize LSTM parameters. The datasets are shuffled at every epoch and the training, validation and testing sets are selected randomly after each shuffle in order to minimize the data biases and improve the validity of the experiments. The 3000 slices are divided into 1800 images used for training, 600 images for validation, and the same for the testing set. The number of samples for training set in the ECG images is 81 images, while the validation set consists of 41 observations, and the testing set has 40 samples. For the turbofan engine degradation simulation dataset, 100 samples are used for training, validation, and testing sets.

6.2.1 Results of Applying BA-CNN-LSTM on Artificial Porosity Images

The novel hybrid BA-CNN-LSTM algorithm is developed using the MATLAB platform, it is applied to the created artificial porosity images described in chapter 4 to predict the percent of porosity in sequential layers of SLM parts. The 30 sequences are divided into 18 sequences for training and 6 sequences for each of the validation and testing sets, since each sequence has 100 layers, so 1800 slices are used for training and 600 slices for each of the validation and testing sets. The following table 6.1 shows the values of LSTM parameters for the four evaluations of BA:

Table 6.1: The Values of LSTM Parameters in the Four Evaluations of BA (Artificial Porosity Images)

LSTM Parameter	1	2	3	4
Learning rate factor for input gate (Forward)	1.0618	0.9737	1.0629	1.0810
Learning rate factor for forget gate (Forward)	0.9472	0.9706	1.0812	1.0177
Learning rate factor for cell candidate (Forward)	0.9151	1.0451	0.9254	0.9291
Learning rate factor for output gate (Forward)	1.0349	1.0273	1.0827	1.0300
Learning rate factor for input gate (Backward)	1.0684	1.0604	1.0265	0.9429
Learning rate factor for forget gate (Backward)	1.0258	1.0236	0.9195	1.0386
Learning rate factor for cell candidate (Backward)	0.9749	0.9621	0.9557	1.0006
Learning rate factor for output gate (Backward)	1.0253	0.9779	1.0094	0.9224
Learning rate factor for fully connected layer	0.9623	0.9477	1.0915	0.9931
Prediction error on the validation set	0.0115	0.0120	0.0134	0.0135

As can be seen in table 6.1, the first evaluation yielded the minimum prediction error on the validation set with a value of 0.0115, so the global learning rate of 0.0101 is adjusted in the forward side of the input gate by multiplying it by 1.0618 resulting in a more optimum learning rate value of 0.0107. Similarly, the learning rate in the forward side of forget gate is improved to 0.0095 using an adjustment factor of 0.9472. The new learning rate value for the forward cell candidate is 0.0092 after multiplying the global learning rate by 0.9151. The adjustment factor for the forward output gate is 1.0348 which results in a learning rate value of 0.0104. The new values for four backward parameters are 0.0108, 0.0103, 0.0098, and 0.0103 for input gate, forget gate, cell candidate and output gate respectively, they are

adjusted using factors of 1.0684, 1.0258, 0.9749, and 1.0253. Finally, the performance of the fully connected layer is improved as well by specifying a customized learning rate of 0.0097 after multiplying the global learning rate by 0.9623. The following table 6.2 summarizes the new learning rate values for LSTM parameters:

Table 6.2: The New Learning Rate Values of LSTM Parameters (Artificial Porosity Images)

LSTM Parameter	Adjusted Learning Rate Value
Input gate (Forward)	0.0107
Forget gate (Forward)	0.0095
Cell candidate (Forward)	0.0092
Output gate (Forward)	0.0104
Input gate (Backward)	0.0108
Forget gate (Backward)	0.0103
Cell candidate (Backward)	0.0098
Output gate (Backward)	0.0103
Fully connected layer	0.0097

The following figure 6.6 shows the training progress for the proposed BA-CNN-LSTM algorithm using the new learning rate values stated above. The blue line represents the training progress and the black line for the validation set.

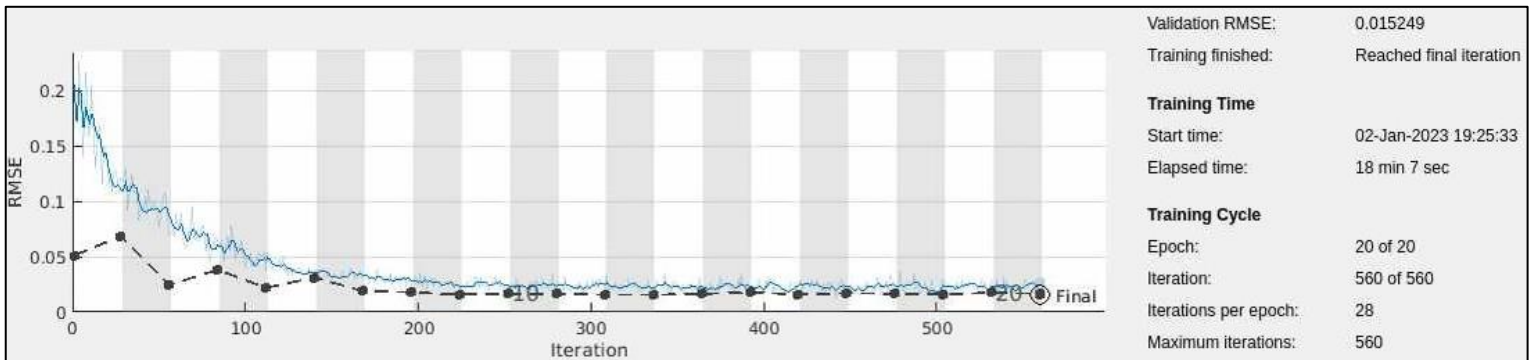


Figure 6.6: Training Progress for the Proposed Hybrid BA-CNN-LSTM Algorithm (Artificial Porosity Images)

As can be seen in figure 6.6, the training starts with RMSE of 0.2 and decreased significantly in the first 200 iterations and then the chart experienced a steady state around RMSE value of 0.02. In the validation set, the chart starts with RMSE value of 0.05 and it

is alternating in the first 150 iterations reaching to RMSE value of 0.0152 at the end of the chart.

The following table 6.3 shows the average porosity error (the difference between the actual and predicted percent of porosity) in training, validation and testing sets. The novel hybrid BA-CNN-LSTM algorithm is compared with an existing algorithm that uses Bayesian Optimization (BO) to optimize the same LSTM parameters (BO-CNN-LSTM) (MathWorks-12), also it is compared with the CNN-LSTM algorithm without BA and with the BA-RCNN algorithm developed in chapter 5 using the same CNN structure.

Table 6.3: The Average Error for Percent of Porosity (Artificial Porosity Images)

	BA-RCNN	CNN-LSTM	BO-CNN-LSTM	BA-CNN-LSTM
Average Error for the Percent of Porosity in the Training Data	0.0230	0.0157	0.0166	0.0159
Average Error for the Percent of Porosity in the Validation Data	0.0214	0.0121	0.0128	0.0115
Average Error for the Percent of Porosity in the Testing Data	0.0234	0.0131	0.0133	0.0122

Adding the LSTM network to CNN reduced the prediction error in all training, validation and testing sets. The hybrid BO-CNN-LSTM did not perform better than the original CNN-LSTM, so BO is not recommended to be used in regression problems as it performs poorly with a high dimensional objective function of more than 20 dimensions (www.stackexchange.com). The CNN-LSTM algorithm is further developed by adding BA to optimize the LSTM parameters which reduced the prediction error further in the validation and testing sets reaching the minimum error value of 0.0115 that comes in the validation set of the novel hybrid BA-CNN-LSTM algorithm.

The following table 6.4 presents the training, validation and testing prediction accuracy for all three algorithms within a 0.02 threshold (the acceptable difference between the actual and predicted percent of porosity), in addition to the time taken for computations in the best iteration.

Table 6.4: The Prediction Accuracy and Time for Percent of Porosity (Artificial Porosity Images)

	BA-RCNN	CNN-LSTM	BO-CNN-LSTM	BA-CNN-LSTM
Training Accuracy	85.94%	88.33%	88.33%	88.33%
Validation Accuracy	87.33%	95.67%	95.33%	96.33%
Testing Accuracy	85.33%	93.17%	93.13%	95.50%
Computational Time	13 Min 40 Sec	17 Min 57 Sec	18 Min 22 Sec	18 Min 7 Sec

As can be seen from table 6.4, adding the LSTM network to CNN improved the prediction accuracy in all training, validation and testing sets. The testing accuracy is increased by 8% in the testing set to be 93.17%. Optimizing LSTM parameters using BO did not improve the testing accuracy and it is almost the same with a value of 93.13%. The CNN-LSTM algorithm is further developed by adding BA to optimize LSTM parameters which increased the prediction accuracy in the validation from 95.67% to 96.33% and in the testing sets from 93.17% to 95.50%, so the improvement in the testing set is approximately 2% from the CNN-LSTM algorithm and 10% from BA-RCNN algorithm. The improvement is similar to the improvement discussed in section 2.4.1 as the hybrid ABC-CNN-LSTM algorithm used to detect the fake reviews of the product yielded an accuracy of 97% compared to 95% for CNN-LSTM algorithm (Jacob et al., 2022). Although the newly developed algorithm requires a longer time for training due to adding another deep learning algorithm (LSTM), but it is more accurate in predicting the sequential data.

So, the performance of the LSTM network in predicting the porosity in the sequential layers is improved after using BA as the hybrid BA-CNN-LSTM provided a more accurate prediction by improving it by 10% for the percent of porosity in sequential layers of artificial porosity images that mimic CT scan images of parts manufactured by the SLM process.

6.2.2 Results of Applying BA-CNN-LSTM on Electrocardiogram (ECG) Dataset

The novel hybrid BA-CNN-LSTM algorithm developed using the MATLAB platform can be designed to deal with classification problems as well. It is applied to Electrocardiogram (ECG) benchmark images described in (MathWorks-7) to classify human ECG time series signals into three classes cardiac arrhythmia (ARR), congestive heart failure (CHF), and normal sinus rhythms (NSR). The following figure 6.7 is an illustrative example of the three classes of ECG time series signals:

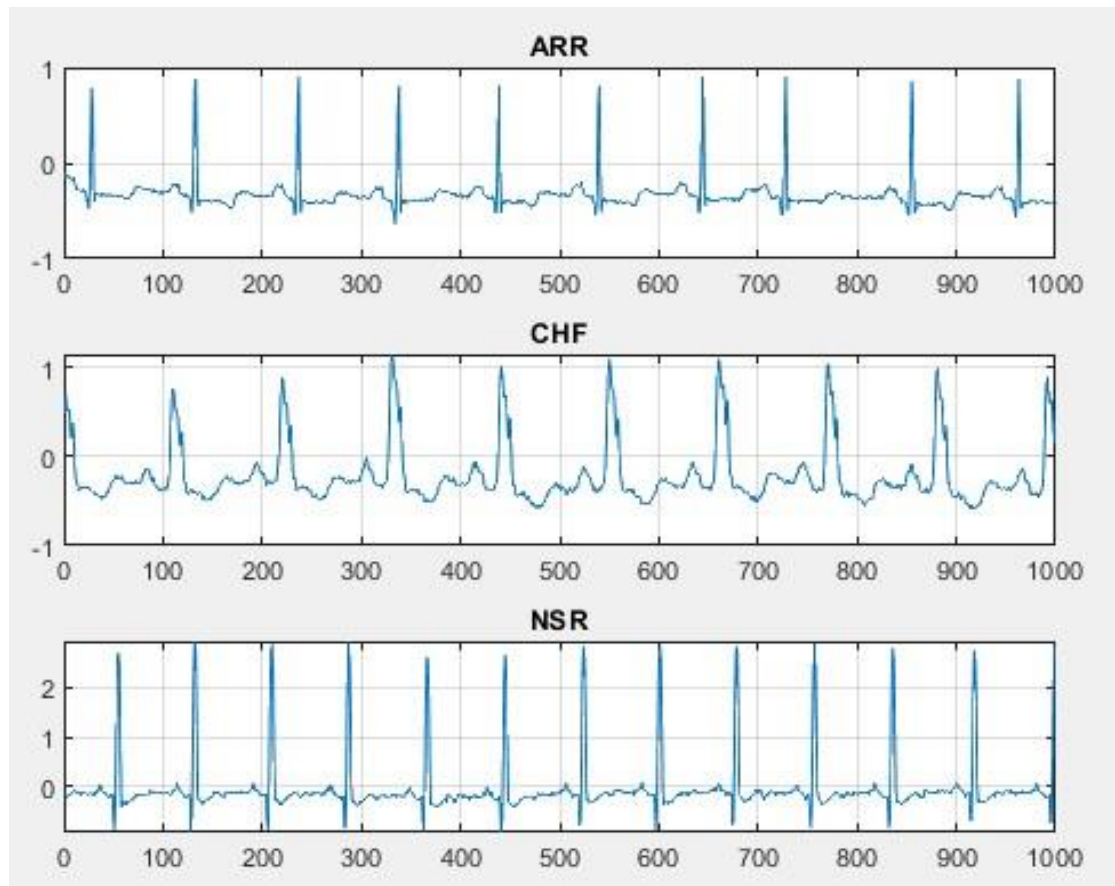


Figure 6.7: An Illustrative Example of the Three Classes of ECG Time Series Signals

There are 162 recordings with 96 observations from ARR, 30 recordings from CHF, and 36 from NSR. As the dataset is not large enough, the “SqueezeNet” pre-trained CNN network is used to extract the features in the images. The data are divided into 81 for training and 41 recordings for the validation set and 40 observations for the testing set. The following table 6.5 shows the values of LSTM parameters for the evaluations of BA:

Table 6.5: The Values of LSTM Parameters in the Evaluations of BA (ECG Dataset)

LSTM Parameter	1	2
Learning rate factor for input gate (Forward)	0.9483	1.0422
Learning rate factor for forget gate (Forward)	0.9808	1.0948
Learning rate factor for cell candidate (Forward)	0.9193	0.9703
Learning rate factor for output gate (Forward)	0.9264	1.0198
Learning rate factor for input gate (Backward)	1.0884	1.0804
Learning rate factor for forget gate (Backward)	1.0912	1.0722
Learning rate factor for cell candidate (Backward)	1.0150	0.9603
Learning rate factor for output gate (Backward)	0.9120	1.0578
Learning rate factor for fully connected layer	0.9470	0.9413
Classification error on the validation set	0.1463	0.1220

As can be seen in table 6.5, the second evaluation yielded the minimum classification error on the validation set with a value of 0.1220, so the global learning rate of 0.00029 is adjusted in the forward side of the input gate by multiplying it by 1.0422 resulting in a more optimum learning rate value of 0.0003. Similarly, the learning rate in the forward side of forget gate is improved to 0.00031 using an adjustment factor of 1.0984. The new learning rate value for the forward cell candidate is 0.00028 after multiplying the global learning rate by 0.9703. The adjustment factor for the forward output gate is 1.0198 which results in a learning rate value of 0.00029. The new values for four backward parameters are 0.00031, 0.00031, 0.00027, and 0.0003 for input gate, forget gate, cell candidate and output

gate respectively, they are adjusted using factors of 1.0804, 1.0722, 0.9603, and 1.0578. Finally, the performance of the fully connected layer is improved as well by specifying a customized learning rate of 0.00027 after multiplying the global learning rate by 0.9413. The following table 6.6 summarizes the new learning rate values for LSTM parameters:

Table 6.6: The New Learning Rate Values of LSTM Parameters (ECG Dataset)

LSTM Parameter	Adjusted Learning Rate Value
Input gate (Forward)	0.000302238
Forget gate (Forward)	0.000317492
Cell candidate (Forward)	0.000281387
Output gate (Forward)	0.000295742
Input gate (Backward)	0.000313316
Forget gate (Backward)	0.000310938
Cell candidate (Backward)	0.000278487
Output gate (Backward)	0.000306762
Fully connected layer	0.000272977

The following figure 6.8 shows the training progress for the proposed BA-CNN-LSTM algorithm using the new learning rate values stated above. The blue line represents the training progress and the black line for the validation set.

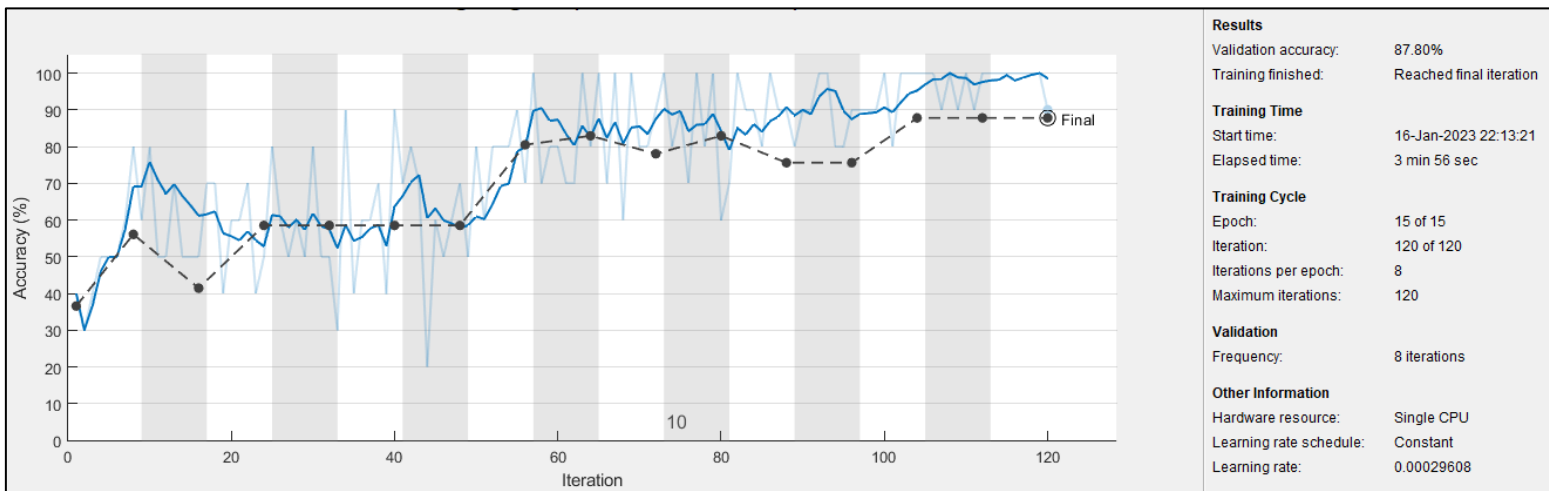


Figure 6.8: Training Progress for the Proposed Hybrid BA-CNN-LSTM Algorithm (ECG Dataset)

As can be seen in figure 6.8, the training starts with a low classification accuracy and increased steadily reaching a percentage value of 100%. In the validation set, the chart has almost the same pattern reaching to validation accuracy of 87.80% at the end of the chart. The following three figures 6.9, 6.10, and 6.11 show the confusion matrix for the BA-CNN-LSTM algorithm for all three sets. Recall and miss out in the blue and red columns while precision and false alarm in the blue and red rows respectively:

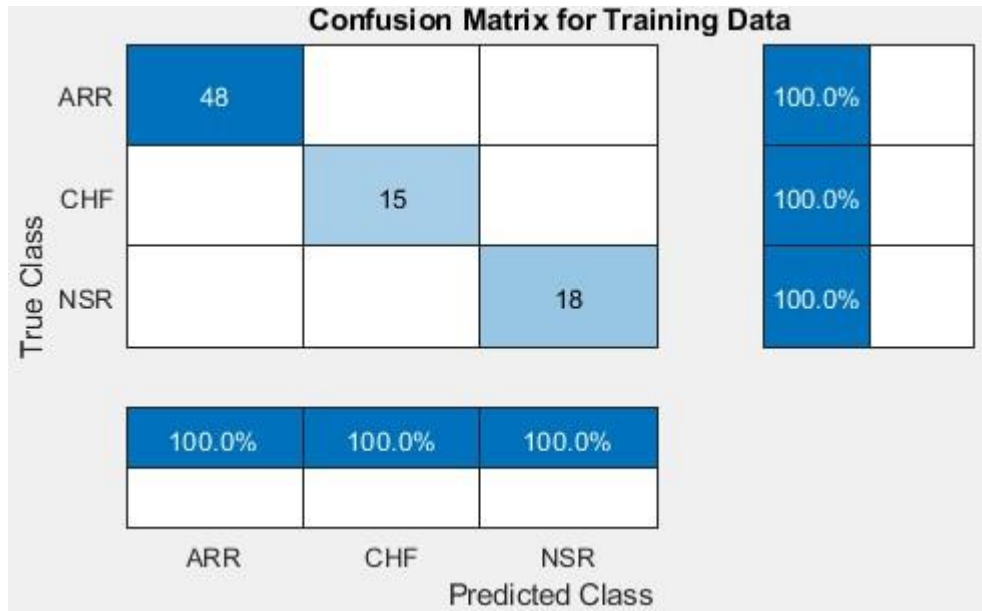


Figure 6.9: Confusion Matrix for the Training Set of the Proposed Hybrid BA-CNN-LSTM Algorithm

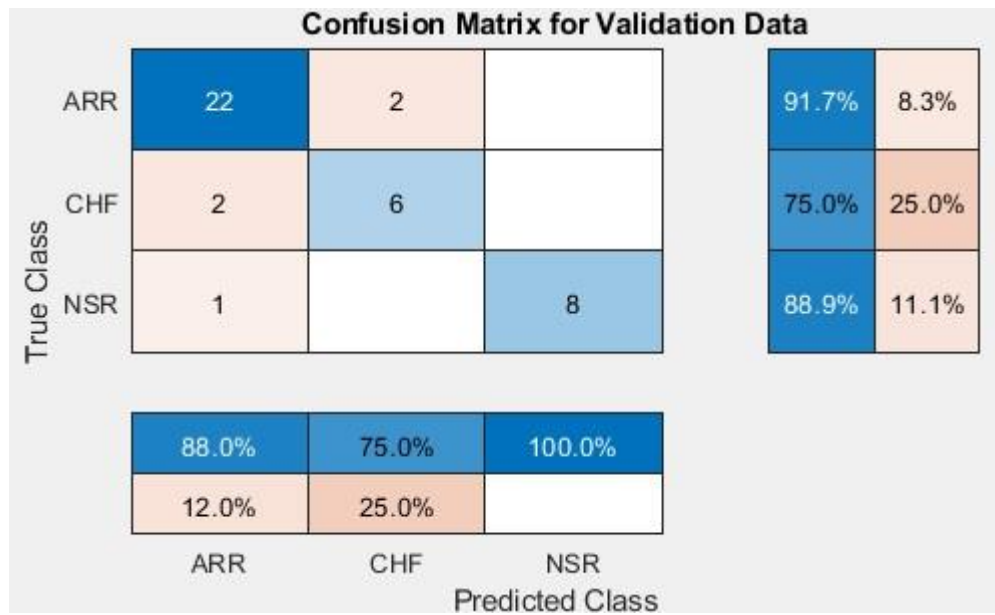


Figure 6.10: Confusion Matrix for the Validation Set of the Proposed Hybrid BA-CNN-LSTM Algorithm

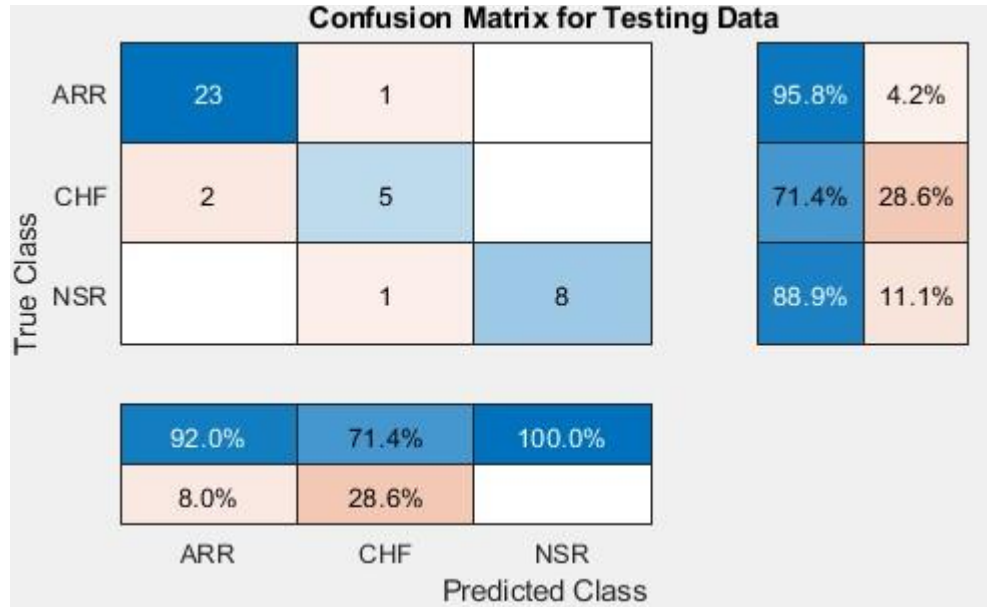


Figure 6.11: Confusion Matrix for the Testing Set of the Proposed Hybrid BA-CNN-LSTM Algorithm

The following table 6.7 presents the training, validation and testing classification accuracy for CNN, CNN-LSTM, BO-CNN-LSTM, and BA-CNN-LSTM algorithms, in addition to the time taken for computations in the best iteration.

Table 6.7: The Classification Accuracy and Time (ECG Dataset)

	CNN	CNN-LSTM	BO-CNN-LSTM	BA-CNN-LSTM
Training Accuracy	100%	100%	100%	100%
Validation Accuracy	85.37%	87.80%	87.80%	87.80%
Testing Accuracy	92.50%	92.50%	95%	95%
Computational Time	3 Min 43 Sec	3 Min 56 Sec	4 Min 5 Sec	3 Min 56 Sec

As can be seen in table 6.7, adding the LSTM network to CNN improved the validation accuracy in the validation set by 2.5%. Optimizing the LSTM parameters using BO or BA improved the testing accuracy from 92.50% to 95%, so both algorithms perform well with classification problems. The computational time is almost similar in all algorithms.

6.2.3 Results of Applying BA-CNN-LSTM on Turbofan Engine Degradation Simulation Dataset

The dataset consists of time series data of 100 engines that starts normally at the beginning and then some faults appear during the series. The numerical data contain 26 columns starting with a unit number, time in cycles, three operational settings, and 21 sensor measurements (Saxena et al., 2008). The dataset contains 100 sequences as each engine represents a sequence that varies in length. There are 100 observations for each of the training, validation, and testing. As the dataset is numerical, CNN is not needed to extract the features, so the hybrid BA-LSTM algorithm is applied using the MATLAB platform to predict the remaining operational cycles before engine failure. As forward LSTM is used in developing the LSTM architecture (MathWorks-11), only four parameters related to this layer are optimized using BA which are the learning rate factor for input gate, forget gate, cell candidate, and fully connected layer. In addition to fully connected layer. The following table 6.8 shows the values of LSTM parameters for the four evaluations of BA:

Table 6.8: The Values of LSTM Parameters in the Four Evaluations of BA (Engine Dataset)

LSTM Parameter	1	2	3	4
Learning rate factor for input gate	1.0712	0.9557	1.0413	0.9048
Learning rate factor for forget gate	1.0167	0.9147	1.0244	0.9898
Learning rate factor for cell candidate	1.0747	1.0939	1.0716	1.0878
Learning rate factor for output gate	1.0526	0.9989	0.9378	1.0041
Learning rate factor for fully connected layer	0.9915	1.0902	0.9613	1.0519
Prediction error on the validation set	4.4272	5.2740	6.7158	11.6459

As can be seen in table 6.8, the first evaluation yielded the minimum prediction error on the validation set with a value of 4.4272, so the global learning rate of 0.01 (MathWorks-11) is adjusted in the input gate by multiplying it by 1.0712 resulting in a more optimum learning rate value of 0.0107. Similarly, the learning rate in the forget gate is improved to 0.0101 using an adjustment factor of 1.0167. The new learning rate value for the cell candidate is 0.0107 after multiplying the global learning rate by 1.0747. The adjustment factor for the output gate is 1.0526 which results in a learning rate value of 0.0105. Finally,

the performance of the fully connected layer is improved as well by specifying a customized learning rate of 0.0099 after multiplying the global learning rate by 0.9915. The following table 6.9 summarizes the new learning rate values for LSTM parameters:

Table 6.9: The New Learning Rate Values of LSTM Parameters (Engine Dataset)

LSTM Parameter	Adjusted Learning Rate Value
Input gate (Forward)	0.0107
Forget gate (Forward)	0.0101
Cell candidate (Forward)	0.0107
Output gate (Forward)	0.0105
Fully connected layer	0.0099

The following figure 6.12 shows the training progress for the BA -LSTM algorithm using the new learning rate values stated above. The blue line represents the training progress and the black line for the validation set.

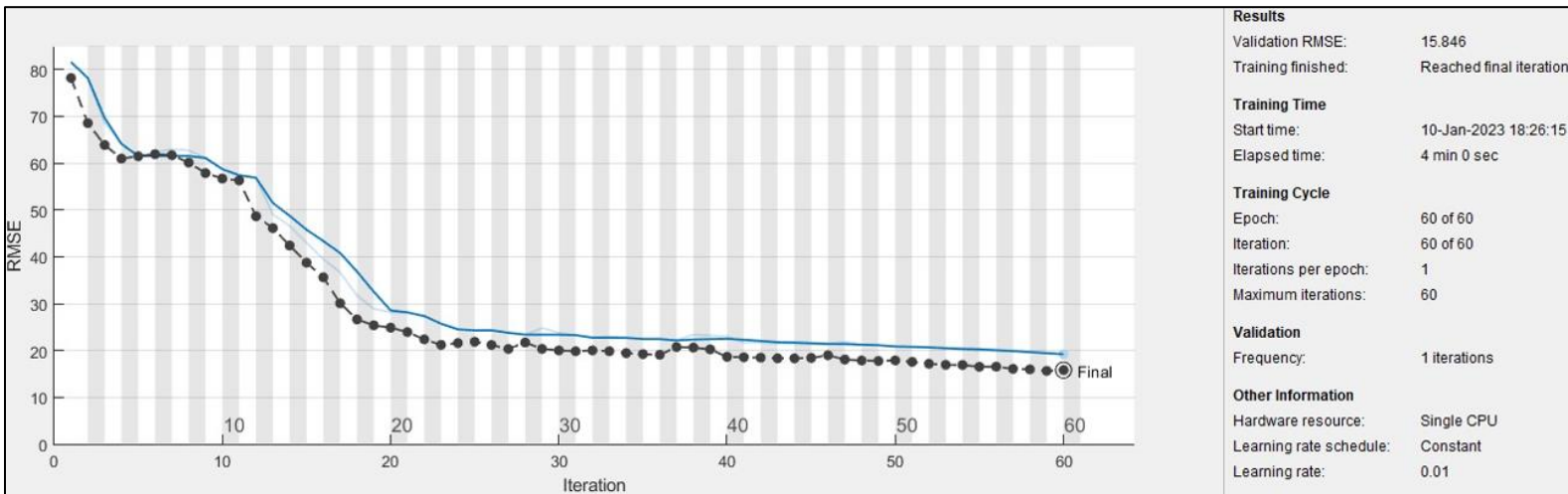


Figure 6.12: Training Progress for the Proposed Hybrid BA-LSTM Algorithm (Engine Dataset)

As can be seen in figure 6.12, the training starts with a root mean square error (RMSE) of almost 80 and decreased significantly in the first 20 iterations and then the chart experienced a steady state around RMSE value of 20. In the validation set, the chart followed the same pattern reaching RMSE value of 15.846 at the end of the chart.

The following table 6.10 presents the training, validation and testing prediction accuracy for LSTM, BO-LSTM and BA-LSTM algorithms within 20 cycles threshold (the acceptable difference between the actual and predicted RUL), in addition to the time taken for computations in the best iteration.

Table 6.10: The Prediction Accuracy and Time (Engine Dataset)

	LSTM	BO-LSTM	BA-LSTM
Training Accuracy	72%	72%	73%
Validation Accuracy	74%	74%	76%
Testing Accuracy	74%	74%	77%
Computational Time	4 Min 10 Sec	4 Min 10 Sec	4 Min

As can be seen in table 6.10, using BO to optimize LSTM parameters did not improve the performance of LSTM, so it confirms the conclusion that it is not recommended to use BO in regression problems as it performs poorly with a high dimensional objective function of more than 20 dimensions (www.stackexchange.com). Adding the BA to optimize LSTM parameters improved the prediction accuracy in all training, validation and testing sets. The testing accuracy is increased by 3% in the testing set to be 77% in BA-LSTM. The computational time is almost similar in both algorithms.

6.3 Summary

Improving the performance of DL algorithms is an ongoing challenge where LSTM which is one of the DL networks that deal with time series or sequential data can be of use. This chapter addressed designing better LSTM topology by optimizing its parameters using one of the most popular nature inspired algorithms that mimic the honey bees' behaviour which is the BA. Artificial porosity images were used for testing the algorithms. Since the input data are images, CNN was added in order to extract the features in the images and to feed them into LSTM to predict the percent of porosity in sequential layers of artificial porosity images that mimic real CT scan images of products manufactured by selective laser melting process. Applying Convolutional Neural Network Long Short-Term Memory (CNN-LSTM) yielded a porosity prediction accuracy of 93.17%. Although using BO to optimize LSTM parameters did not improve the performance of the LSTM as BO performs poorly with a high dimensional objective function of more than 20 dimensions which is the

case in regression problems. Adding BA to optimize LSTM parameters improved its performance in predicting the porosity with an accuracy of 95.50% using hybrid Bees Convolutional Neural Network Long Short-Term Memory (BA-CNN-LSTM). Furthermore, the hybrid BA-CNN-LSTM algorithm can be designed to deal with classification problems as well. Applying it to Electrocardiogram (ECG) benchmark images improved the classification accuracy on the testing set from 92% to 95%. In addition, the turbofan engine degradation simulation numerical dataset was used to predict the remaining useful life (RUL) of engines using LSTM. CNN is not needed in this case as there is no feature extraction for the images, adding the BA to optimize LSTM parameters improved the prediction accuracy by 3% in the testing set to be 77% in the hybrid BA-LSTM algorithm.

The outcome of this chapter is improving the performance of the LSTM network in predicting sequential data using BA. As the input data are images, CNN is added to extract the image features yielding a hybrid algorithm (BA-CNN-LSTM) that provides a 10% more accurate prediction of porosity percentage appearing in sequential layers of artificial porosity images that mimic CT scan images of parts manufactured by SLM process. The following table 6.11 compares the prediction accuracy of the algorithms developed in this chapter and the previous chapter:

Table 6.11: Comparison between All Algorithms Used for Porosity Prediction Accuracy (Final Version)

Prediction Method	Porosity Prediction Accuracy
Image Binarization	68.60%
Original RCNN	75.50%
Hybrid BA-RCNN	85.33%
Hybrid CNN-LSTM	93.17%
Hybrid BA-CNN-LSTM	95.50%

The contribution of this chapter is:

- Improving the performance of LSTM network in predicting sequential data using BA. As the input data are images, CNN is added to extract the image features yielding a hybrid algorithm (BA-CNN-LSTM) that provides a 10% more accurate prediction of porosity percentage appearing in the sequential layers of artificial porosity images that mimic CT scan images of parts manufactured by SLM process.

Chapter 7: Conclusion

7.1 Conclusion

Artificial Neural Network (ANN) has the ability to handle high dimensional real-time data and extracts implicit meaningful patterns that can be used to predict the future state of complex systems. The performance of ANNs depends on the significance of the features extracted from the training data used in training which is often a time consuming process leading to sub-optimal solutions. However, this problem could be overcome by Deep Learning (DL) which is a type of machine learning technique. DL has better learning capability as it has the advantage of automatic feature extraction by learning a large number of nonlinear filters before making decisions.

In this work, a review of DL has been conducted showing all its networks along with their advantages, limitations, and applications. The most popular DL network is the Convolutional Neural Network (CNN) which uses feature learning layers namely convolution, rectified linear unit, batch normalization, and pooling layers in addition to two classification layers one of which is the fully connected layer, and the other is a SoftMax layer used mainly for multiclass image classification. It has been found from the review that there are five open issues namely overfitting, exploding gradient problem, training the model using imbalanced classes, the convergence speed and designing better CNN topology, all of which need to be addressed in order to improve the performance of CNN models further. The most challenging aspect in training CNN is designing better topology where the traditional heuristic approach of using trial and error will result in a less accurate model depending on user experience. In case of applying optimization techniques such as nature inspired algorithms to optimize the parameters of CNN can improve the performance of the model. However, designing better CNN topology is still an open issue where no approach has been found yet that can give the best CNN topology. In addition, nature inspired algorithms have been included showing their contribution to increase the classification performance of CNN models.

This work addressed the most challenging issue stated above which is designing a better CNN topology by proposing a novel nature inspired hybrid algorithm that uses the Bees Algorithm (BA) which is known to mimic the behaviour of honeybees, to optimize four CNN parameters namely section depth, initial learning rate, momentum, and regularization. The proposed novel hybrid algorithm is called Bees Convolutional Neural Network (BA-

CNN) algorithm. Furthermore, another nature inspired hybrid algorithm is proposed which combines the BA with Bayesian Optimization (BO) to increase the performance of CNN which is referred to as BA-BO-CNN. BO was used to optimize the same four parameters while BA was applied to optimize the weight learning rate factor to adjust the global learning rate obtained by BO algorithm in each convolutional layer and fully connected layer. Different data sets have been used to test the proposed novel algorithms.

1. Applying the hybrid BA-CNN algorithm to the 'Cifar10DataDir' benchmark image data did not improve the validation and testing accuracy compared to the existing CNN and BO-CNN, but it yielded better performance than EA-CNN model designed in (Badan, 2019) that achieved an accuracy of 62.37% on the same 'Cifar10DataDir' dataset as was shown in table 2.2 while BA-CNN achieved testing accuracy of 80.02%. Applying it to the digits dataset yielded similar accuracy to the existing algorithms, but it produced the lowest computational time with 4 minutes and 14 seconds reduction compared to the BO-CNN, so it is the best algorithm in terms of cost-effectiveness. Applying it to concrete cracks images produced almost similar results to existing algorithms. Finally, applying it to the ECG images improved the testing accuracy from 90% for the BO-CNN to 92.50% for the novel hybrid BA-CNN algorithm with similar validation accuracy and computational time.
2. In addition, applying the other hybrid BA-BO-CNN algorithm to the 'Cifar10DataDir' benchmark image data produced validation accuracy better than the existing algorithms with a validation accuracy of 82.22% compared to 80.34% and 80.72% for the CNN and BO-CNN respectively, also it has a better testing accuracy of 80.74% compared to 80.52% and 80.69% for the CNN and BO-CNN respectively. Applying it to the digits dataset showed the same accuracy as the existing original CNN and BO-CNN, but with an improvement in the computational time by 3 minutes and 12 seconds reduction. Applying it to concrete cracks images produced almost similar results to the existing algorithms. Finally, applying it to human Electrocardiogram (ECG) signals improved the testing accuracy from 92.50% for the original CNN to 95% for

the novel hybrid BA-BO-CNN algorithm with similar validation accuracy and computational time.

Furthermore, this work presented Additive Manufacturing (AM) processes, applications, advantages, and gaps and open issues that needed to be addressed. It focused on powder bed fusion process showing the way of working, thermodynamical phenomena, parameters, open issues, three porosity types, and state of the art studies about adopting DL techniques to improve the performance of SLM process. It was found that there is a problem related to assessing accurately the porosity in SLM parts when using CT scans of sample parts. One main drawback with the conventional image binarization method has been when using gray value analysis to assess the porosity of SLM parts visible in CT scan slices. The difficulty is the subjectivity in selecting an appropriate grayscale threshold that would convert a single slice into binary images highlighting defective regions, as well as determining the true level of porosity. When an inappropriately low grayscale threshold was applied to the original slice image for binary image conversion, a certain amount of tiny undesired white spots were not filtered. However, if a higher grayscale threshold was adopted, the morphological features of the defective area, specifically near the boundary, were altered dramatically. These thresholds resulted in significantly different predictions of porosity levels.

In addition to classification, CNN has been used for regression problems to predict the percent of porosity in the finished SLM part without the need for subjective difficult threshold determination to convert the single slice to binary image. Applying Regression CNN (RCNN) on 3000 artificially created porosity images, similar to real CT-scan images by a similarity index of 0.9967, improved porosity prediction accuracy from 68.60% for the image binarization method to 75.50% for RCNN, while integrating the BA with the last algorithm produced the best prediction accuracy with a value of 85.33% for Bees Regression Convolutional Neural Network (BA-RCNN).

In addition, the MATLAB platform was used to develop and apply Convolutional Neural Network Long Short-Term Memory (CNN-LSTM) yielding better porosity prediction accuracy of 93.17%. Although using BO to optimize LSTM parameters did not improve the performance of LSTM as BO performs poorly with a high dimensional

objective function of more than 20 dimensions which is the case in regression problems. Adding BA to optimize LSTM parameters improved its performance in predicting the porosity with an accuracy of 95.50% using hybrid Bees Convolutional Neural Network Long Short-Term Memory (BA-CNN-LSTM).

Furthermore, the hybrid BA-CNN-LSTM algorithm was designed to deal with classification problems as well. Applying it to ECG benchmark images improved the classification accuracy on the testing set from 92% to 95%. In addition, the turbofan engine degradation simulation dataset was used to predict the remaining useful life (RUL) of engines using LSTM, adding the BA to optimize LSTM parameters improved the prediction accuracy by 3% in the testing set to be 77% in the hybrid BA-LSTM algorithm.

7.2 Contributions to Knowledge

In summary, the following are the four contributions to knowledge:

1. Developing a novel hybrid Bees Convolutional Neural Network (BA-CNN) algorithm in order to improve the performance of CNN.
2. Developing a novel hybrid Bees Bayesian Convolutional Neural Network (BA-BO-CNN) algorithm in order to improve the performance of CNN.
3. Proposing and validating a new approach for predicting the percent of porosity in the finished SLM parts, using hybrid Bees Regression Convolutional Neural Network (BA-RCNN). It was demonstrated that a better accuracy than the existing image binarization method could be achieved (approximately 17% improvement with the data set used). In order to test the algorithm, as the training of the RCNN would require a large amount of experimental data, artificial porosity images mimicking real CT scan slices of the finished SLM part were created with a similarity index of 0.9976 with real images.
4. Improving the performance of LSTM network in predicting sequential data using BA. As the input data are images, CNN was added to extract the features in the images yielding a hybrid algorithm (BA-CNN-LSTM) that provided a more accurate prediction by improving it by 10% for the percent of porosity in the sequential layers of artificial porosity images that mimic CT scan images of parts manufactured by SLM process.

The first two contributions address the first research question Q1 mentioned in section 1.3 which is about the way of designing the optimum topology for CNN using the BA. The third contribution considers the second research question Q2 which deals with the way of analysing the porosity in the parts produced by the SLM process using novel CNN. Finally, the fourth contribution to the knowledge addresses the last research question Q3 for improving the performance of LSTM network in dealing with sequential data using the BA.

7.3 Study Limitations

The study can be done with more number of evaluations used in conducting BA to optimize DL parameters. In this study, they were limited to the computer capability in advanced research computing at Cardiff University. As DL networks require high computations, the number of evaluations was limited. In addition, the BA parameters of the number of iterations, scout bees, and elite sites were also assigned based on the computer capability in advanced research computing at Cardiff University.

This research study was not aimed to study the porosities in depth, so no experiments have been conducted, but the work proposed DL methods that enhanced such studies, particularly in predicting the percent of porosity in the finished SLM part. So, the created artificial porosity images in chapter 4 were used only to test the proposed DL algorithms developed in chapters 5 and 6.

The creation of artificial porosity images were only based on the laser power and scanning speed parameters as the dataset used to create the images presents only data about laser power and scanning speed. It is assumed that the pores are created in 3D cube with a volume of 1 mm^3 with Z-position represents the average between the X-position and Y-position as Z coordinates are not available in the study used for porosity position analysis. Finally, the created artificial porosity images illustrated only one type of pore which is keyhole porosity considering one shape of this type which is the nearly spherical shape.

7.4 Future Work

The newly developed hybrid CNN algorithms can be developed further by optimizing the weight regularization factor in the convolutional layers and fully connected layer using the BA in order to improve the performance of the algorithms, particularly in preventing the network overfitting. For LSTM network, the BA can be used to optimize the adjustment factor for the regularization in the forget, input, and output gates in both forward and backward directions in addition to the cell candidate. Having more optimum regularization value, reduce the probability of overfitting. The proposed CNN and LSTM algorithms can be used in different contexts such as in the robotics field to build robots with obstacles sense in their path.

In addition, further work is required to tune the artificial porosity images creation approach to be mimicking a wider range of porosity types with more complex shapes closer to the cavities than the spheres and considering all the influential factors that affect the formation of keyhole and lack of fusion porosity, not only laser power and scanning speed. If the research fund can cover the cost of producing real porosity images enough for training DL techniques, then real experiments can be conducted in order to produce real porosity images giving more accurate prediction results, this is helpful in case of assessing the gas porosity as it is not linked directly with the laser power and scanning speed.

References

Abdulhameed, O., Al-Ahmari, A., Ameen, W. and Mian, S.H. (2019). Additive manufacturing: Challenges, trends, and applications. *Advances in Mechanical Engineering*, 11(2), p.168781401882288.

Al-Musawi, A. (2019). The development of new artificial intelligence based hybrid techniques combining bees algorithm, data mining and genetic algorithm for detection, classification and prediction of faults in induction motors. (*Doctoral dissertation, Cardiff University*).

Arvieu, C., Galy, C., Le Guen, E. and Lacoste, E. (2020). Relative Density of SLM-Produced Aluminum Alloy Parts: Interpretation of Results. *Journal of Manufacturing and Materials Processing*, 4(3), p.83.

Autodesk.com. (2020). *Smart Manufacturing Technology Software | Autodesk*. [online] Available at: <https://www.autodesk.com/solutions/smart-manufacturing> [Accessed: 08 September 2020].

Badan, F. (2019). *Evolutionary Algorithms in Convolutional Neural Network Design*. [online] Available at: <http://excel.fit.vutbr.cz/submissions/2019/033/33.pdf> [Accessed: 16 March 2020].

Badem, H., Basturk, A., Caliskan, A. and Yuksel, M.E. (2017). A new efficient training strategy for deep neural networks by hybridization of artificial bee colony and limited-memory BFGS optimization algorithms. *Neurocomputing*, 266, pp.506–526.

Baldominos, A., Saez, Y. and Isasi, P. (2018). Evolutionary convolutional neural networks: An application to handwriting recognition. *Neurocomputing*, 283, pp.38–52.

Bauer, T. (2021). Prediction of process parameters in selective laser melting. (*Doctoral dissertation, ETH Zurich*).

Bernard, N., & Leprevost, F. (2018, September). Evolutionary Algorithms for Convolutional Neural Network Visualisation. In *Latin American High Performance Computing Conference*, Springer, Cham, pp. 18-32.

Brownlee, J. (2020). *Understand the Impact of Learning Rate on Neural Network Performance*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/> [Accessed: 29 November 2021].

Brownlee, J. (2017). *How to Develop a Bidirectional LSTM For Sequence Classification in Python with Keras*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python-keras/> [Accessed: 7 January 2023].

Bullinaria, J.A. and AlYahya, K. (2014). Artificial Bee Colony training of neural networks: comparison with back-propagation. *Memetic Computing*, 6(3), pp.171–182.

Chaganti, S.Y., Nanda, I., Pandi, K.R., Prudhvith, T.G. and Kumar, N. (2020, March). Image Classification using SVM and CNN. In *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, IEEE, pp.1-5.

Chandrayan, P. (2017). *Deep Learning: Deep Belief Network Fundamentals*. [Online] Code Burst. Available at: <https://codeburst.io/deep-learning-deep-belief-network-fundamentals-d0dcfd80d7d4> [Accessed: 8 October 2019].

Chiroma, H., Gital, A. Y. U., Rana, N., Shafi'i, M. A., Muhammad, A. N., Umar, A. Y., and Abubakar, A. I. (2019, April). Nature Inspired Meta-heuristic Algorithms for Deep Learning: Recent Progress and Novel Perspective. *In Science and Information Conference*, Springer, Cham, pp.59-70.

Chung, H. and Shin, K. (2019). Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction. *Neural Computing and Applications*, pp.1-18.

Ciorcea, E. M. (2018, August). IoT analysis of manufacturing using Petri Nets. *In IOP Conference Series: Materials Science and Engineering* (Vol. 400, No. 4, p. 042010). IOP Publishing.

Coeck, S., Bisht, M., Plas, J. and Verbist, F. (2019). Prediction of lack of fusion porosity in selective laser melting based on melt pool monitoring data. *Additive Manufacturing*, 25, pp.347–356.

Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., & Batra, D. (2015). Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv,1511.06068*.

Davydova, O. (2017). *7 types of Artificial Neural Networks for Natural Language Processing*. [online] Medium. Available at: <https://medium.com/@datamonsters/artificial-neural-networks-for-natural-language-processing-part-1-64ca9ebfa3b2> [Accessed: 8 October 2019].

De Filippis, L. A. C., Serio, L. M., Facchini, F., and Mummolo, G. (2017). ANN Modelling to Optimize Manufacturing Process. *In Advanced Applications for Artificial Neural Networks*. IntechOpen.

Ding, L., Fang, W., Luo, H., Love, P. E., Zhong, B., and Ouyang, X. (2018). A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory. *Automation in construction*, 86, pp.118-124.

Ekins, S. (2016). The next era: deep learning in pharmaceutical research. *Pharmaceutical research*, 33(11), pp.2594-2603.

Elngar, A.A., Arafa, M., Fathy, A., Moustafa, B., Mahmoudm, O., Shaban, M. and Fawzy, N. (2021). Image classification based on CNN: a survey. *J. Cybersecurity Inf. Manag.(JCIM)*, 6(1), pp.18-50.

Engati. (n.d.). *Vanishing gradient problem*. [online] Available at: <https://www.engati.com/glossary/vanishing-gradient-problem#:~:text=The%20vanishing%20gradient%20problem%20is%20essentially%20a%20situation> [Accessed: 4 January 2023].

FacFox Docs. (n.d.). *Selective Laser Melting – The Ultimate Guide*. [online] Available at: <https://www.facfox.com/docs/kb/selective-laser-melting-the-ultimate-guide> [Accessed: 2 September 2022].

Farinia Group. (2018). *What is Additive Layer Manufacturing (ALM)?* [online] Available at: <https://www.farinia.com/additive-manufacturing/3d-technique/additive-layer-manufacturing> [Accessed: 4 January 2020].

Feng, S., Chen, Z., Bircher, B., Ji, Z., Nyborg, L. and Bigot, S. (2022). Predicting laser powder bed fusion defects through in-process monitoring data and machine learning. *Materials & Design*, 222, p.111115.

Foote, K. (2022). *A Brief History of Deep Learning*. [Online] Available at: <https://www.dataversity.net/brief-history-deep-learning/#> [Accessed: 26 October 2022].

Fraley, S., Zalewski, J., Oom, M., and Terrien, B. (2020). *14.1: Design of Experiments via Taguchi Methods - Orthogonal Arrays*. [Online] Available at: [https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Book%3A_Chemical_Process_Dynamics_and_Controls_\(Under_Construction\)/14%3A_Design_of_Experiments/14.01%3A_Design_of_Experiments_via_Taguchi_Methods_Orthogonal_Arrays](https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Book%3A_Chemical_Process_Dynamics_and_Controls_(Under_Construction)/14%3A_Design_of_Experiments/14.01%3A_Design_of_Experiments_via_Taguchi_Methods_Orthogonal_Arrays). [Accessed: 29 July 2020].

Fu, K., Cheng, D., Tu, Y., and Zhang, L. (2016, October). Credit card fraud detection using convolutional neural networks. *In International Conference on Neural Information Processing*, Springer, Cham., pp.483-490.

Geissbauer, R., Vedso, J., and Schrauf, S. (2016). Global Industry 4.0 Survey: Building the digital enterprise. PwC.

Gong, H., Nadimpalli, V. K., Rafi, K., Starr, T., and Stucker, B. (2019). Micro-CT evaluation of defects in Ti-6Al-4V parts fabricated by metal additive manufacturing. *Technologies*, 7(2), 44.

Gorgolis, N., Hatzilygeroudis, I., Istenes, Z., and Gyenne, L. G. (2019, July). Hyperparameter optimization of LSTM network models through genetic algorithm. *In 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, IEEE, pp.1-4.

Gu, D., Ma, C., Xia, M., Dai, D., and Shi, Q. (2017). A multiscale understanding of the thermodynamic and kinetic mechanisms of laser additive manufacturing. *Engineering*, 3(5), pp.675-684.

Hui, J. (2017). *Convolutional neural networks (CNN) tutorial*. [Online] Available at: <https://jhui.github.io/2017/03/16/CNN-Convolutional-neural-network> [Accessed: 15 November 2021].

Imanguliyev, A. (2013). Enhancements for the Bees Algorithm. (*Doctoral dissertation, Cardiff University*).

(ISO/ASTM 52900, 2021), *Additive manufacturing. General principles. Fundamentals and vocabulary*. [Online] Available at: <https://www.iso.org/obp/ui/#iso:std:iso-astm:52900:ed-2:v1:en> [Accessed: 15 November 2022].

Jacob, M. S., & Selvi Rajendran, P. (2022). Fuzzy artificial bee colony-based CNN-LSTM and semantic feature for fake product review classification. *Concurrency and Computation: Practice and Experience*, 34(1), e6539.

Joshi, S., Verma, D. K., Saxena, G., & Paraye, A. (2019, April). Issues in Training a Convolutional Neural Network Model for Image Classification. In *International Conference on Advances in Computing and Data Sciences*, Springer, Singapore, pp.282-293.

Kamalika, S. (2018). *The History, Evolution and Growth of Deep Learning*. [Online] Available at: <https://www.analyticsinsight.net/the-history-evolution-and-growth-of-deep-learning/> [Accessed: 26 October 2022].

Kim, D. G., & Choi, J. Y. (2021). Optimization of Design Parameters in LSTM Model for Predictive Maintenance. *Applied Sciences*, 11(14), 6450.

Kim, H., Lin, Y., and Tseng, T. L. B. (2018). A review on quality control in additive manufacturing. *Rapid Prototyping Journal*.

Koc, E. (2010). The Bees Algorithm Theory, Improvements and Applications. *Manufacturing Engineering Centre, School of Engineering, University of Wales, Cardiff, United Kingdom*.

Korshunova, K. P. (2018, August). A convolutional fuzzy neural network for image classification. In *2018 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC)* (pp. 1-4). IEEE.

Kumar, R., Kumar, P., & Kumar, Y. (2022). Integrating big data driven sentiments polarity and ABC-optimized LSTM for time series forecasting. *Multimedia Tools and Applications*, *81*(24), pp.34595-34614.

Lastra, R., Pereira, A., Díaz-Cacho, M., Acevedo, J., and Collazo, A. (2022). Spare Parts Made by Additive Manufacturing to Improve Preventive Maintenance. *Applied Sciences*, *12*(20), 10564.

Le, Q. V. (2015). A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. *Google Brain*, pp.1-20.

Lee, W. Y., Park, S. M., & Sim, K. B. (2018). Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm. *Optik*, *172*, pp.359-367.

Li, B. H., Hou, B. C., Yu, W. T., Lu, X. B., and Yang, C. W. (2017). Applications of artificial intelligence in intelligent manufacturing: a review. *Frontiers of Information Technology & Electronic Engineering*, *18*(1), pp.86-96.

Liang, J., & Liu, R. (2015, October). Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network. In *2015 8th International Congress on Image and Signal Processing (CISP)*, IEEE, pp.697-701.

Lindfield, G., & Penny, J. (2017). *Introduction to Nature-Inspired Optimization*. Academic Press.

(Machine Learning Knowledge) (2019). *Brief History of Deep Learning from 1943-2019 [Timeline]*. [Online] Available at: <https://machinelearningknowledge.ai/brief-history-of-deep-learning/> [Accessed: 26 October 2022].

(Machinelearningmastery), *How to Diagnose Overfitting and Underfitting of LSTM Models*. [Online] Available at: <https://machinelearningmastery.com/diagnose-overfitting-underfitting-lstm-models/> [Accessed: 4 January 2023].

Ma, N., Zhang, X., Zheng, H. T., and Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. *In Proceedings of the European Conference on Computer Vision (ECCV)*, pp.116-131.

(Machine Learning Repository). [Online] Available at: <https://archive.ics.uci.edu/ml/index.php> [Accessed: 2 December 2021].

Mallawaarachchi, V. (2017). *Introduction to Genetic Algorithms*. [Online] Available at: <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3> [Accessed: 2 April 2020].

(Markforged), *Additive Manufacturing History: From the 1980s to Now* [Online] Available at: <https://markforged.com/resources/blog/additive-manufacturing-history> [Accessed: 25 October 2022].

Maskery, I., Aboulkhair, N.T., Corfield, M.R., Tuck, C., Clare, A.T., Leach, R.K., Wildman, R.D., Ashcroft, I.A. and Hague, R.J. (2016). Quantification and characterisation of porosity in selectively laser melted Al–Si10–Mg using X-ray computed tomography. *Materials Characterization*, 111, pp.193-204.

Masko, D., & Hensman, P. (2015). The impact of imbalanced training data for convolutional neural networks.

(MathWorks-1), *Convolutional Neural Network*. [Online] Available at: <https://uk.mathworks.com/solutions/deep-learning/convolutional-neural-network.html> [Accessed: 7 October 2019].

(MathWorks-2), *Long Short-Term Memory Networks*. [Online] Available at: <https://www.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html>. [Accessed: 8 January 2023].

(MathWorks-3), *Deep Learning Using Bayesian Optimization*. [Online] Available at: <https://www.mathworks.com/help/deeplearning/ug/deep-learning-using-bayesian-optimization.html> [Accessed: 4 April 2020].

(MathWorks-4), *Bees Algorithm (BeA) in MATLAB*. [Online] Available at: <https://uk.mathworks.com/matlabcentral/fileexchange/52967-bees-algorithm-bea-in-matlab> [Accessed: 9 April 2020].

(MathWorks-5), *How to draw random-size circles in square - non overlap?* [Online] Available at: <https://uk.mathworks.com/matlabcentral/answers/441380-how-to-draw-random-size-circles-in-square-non-overlap> [Accessed: 17 June 2021].

(MathWorks-6), *TrainingOptionsSGDM*. [Online] Available at: <https://uk.mathworks.com/help/deeplearning/ref/nnet.cnn.trainingoptionssgdm.html> [Accessed: 15 November 2021].

(MathWorks-7), *Classify Time Series Using Wavelet Analysis and Deep Learning*. [Online] Available at: <https://uk.mathworks.com/help/signal/ug/signal-classification-with-wavelet-analysis-and-convolutional-neural-networks.html> [Accessed: 31 August 2022].

(MathWorks-8), *Overlay image on top of image*. [Online] Available at: <https://www.mathworks.com/matlabcentral/answers/414351-overlay-image-on-top-of-image> [Accessed: 4 October 2022].

(MathWorks-9), *Structural similarity (SSIM) index for measuring image quality*. [Online] Available at: <https://uk.mathworks.com/help/images/ref/ssim.html> [Accessed: 27 July 2022].

(MathWorks-10), *adaptthresh*. [Online] Available at: <https://uk.mathworks.com/help/images/ref/adaptthresh.html>. [Accessed: 2 September 2022].

(MathWorks-11), *Sequence-to-Sequence Regression Using Deep Learning*. [Online] Available at: <https://www.mathworks.com/help/deeplearning/ug/sequence-to-sequence-regression-using-deep-learning.html> [Accessed: 10 January 2023].

(MathWorks-12), *Choose Training configurations for LSTM using Bayesian Optimization*. [Online] Available at: <https://www.mathworks.com/help/deeplearning/ug/exp-mgr-sequence-regression-example.html> [Accessed: 28 January 2023].

Mattioli, F.E., Caetano, D.J., Cardoso, A., Naves, E.L., and Lamounier, E.A. (2019). An Experiment on the Use of Genetic Algorithms for Topology Selection in Deep Learning. *J. Electr. Comput. Eng.*, pp.3217542:1-3217542:12.

McDermott, J. (2021). *Convolutional Neural Networks — Image Classification w. Keras*. [Online] Available at: <https://www.learn datasci.com/tutorials/convolutional-neural-networks-image-classification> [Accessed: 15 November 2021].

(Metal AM), *Applications for metal Additive Manufacturing technology*. [Online] Available at: <https://www.metal-am.com/introduction-to-metal-additive-manufacturing-and-3d-printing/applications-for-additive-manufacturing-technology/> [Accessed: 10 January 2020].

Motepe, S., Hasan, A. N., and Stopforth, R. (2019). Improving Load Forecasting Process for a Power Distribution Network Using Hybrid AI and Deep Learning Algorithms. *IEEE Access*, 7, pp.82584-82598.

Moyne, J., & Iskandar, J. (2017). Big data analytics for smart manufacturing: Case studies in semiconductor manufacturing. *Processes*, 5(3), 39.

Mungalpara, J. (2021). *What does it mean by Bidirectional LSTM?* [Online] Available at: <https://medium.com/analytics-vidhya/what-does-it-mean-by-bidirectional-lstm-63d6838e34d9> [Accessed: 7 January 2023].

Nag, A. (2021). *Bayesian Optimization: A step by step approach* [Online] Available at: <https://towardsdatascience.com/bayesian-optimization-a-step-by-step-approach-a1cb678dd2ec> [Accessed: 9 August 2022].

Newman, L. (2020). *Classifying Toxicity in Online Comment forums: End-to-End Project*. [Online] Available at: <https://towardsdatascience.com/classifying-toxicity-in-online-comment-forums-end-to-end-project-57720af39d0b> [Accessed: 26 January 2023].

(Office of Energy Efficiency and Renewable Energy) (2017). *What is Additive Manufacturing?* [Online] Available at: <https://www.energy.gov/eere/articles/what-additive-manufacturing> [Accessed: 25 October 2022].

Ouf, H. (2017). *Maxpooling vs minpooling vs average pooling*. [Online] Available at: <https://hany-ouf.blogspot.com/2020/08/maxpooling-vs-minpooling-vs-average.html> [Accessed: 15 November 2021].

Packianather, M. S., Al-Musawi, A. K., and Anayi, F. (2019). Bee for mining (B4M)—A novel rule discovery method using the Bees algorithm with quality-weight and coverage-weight. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 233(14), pp.5101-5112.

Packianather, M. S., Yuce, B., Mastrocinque, E., Fruggiero, F., Pham, D. T., and Lambiase, A. (2014, August). Novel Genetic Bees Algorithm applied to single machine scheduling problem. In *2014 World Automation Congress (WAC)*, IEEE , pp.906-911.

Pan, H. (2017). *A Study on Deep Learning: Training, Models and Applications*.

Panwar, M., Padmini, J., Acharyya, A., and Biswas, D. (2017, September). Modified distributed arithmetic based low complexity CNN architecture design methodology. In *2017 European Conference on Circuit Theory and Design (ECCTD)*, IEEE, pp.1-4.

Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S. and Zaidi, M., 2005. The bees algorithm. *Technical Note, Manufacturing Engineering Centre, Cardiff University*, UK, pp.44-48.

Popko, E. A., & Weinstein, I. A. (2016, August). Fuzzy logic module of convolutional neural network for handwritten digits recognition. In *Journal of Physics: Conference Series* (Vol. 738, No. 1, p. 012123). IOP Publishing.

Pranolo, A., Mao, Y., Wibawa, A. P., Utama, A. B. P., & Dwiyanto, F. A. (2022). Optimized Three Deep Learning Models Based-PSO Hyperparameters for Beijing PM2.5 Prediction. *Knowledge Engineering and Data Science*, 5(1), pp.53-66.

Qolomany, B., Maabreh, M., Al-Fuqaha, A., Gupta, A., and Benhaddou, D. (2017, June). Parameters optimization of deep learning models using particle swarm optimization. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, pp.1285-1290.

Qureshi, A. U. H., Larijani, H., Mtetwa, N., Javed, A., & Ahmad, J. (2019). RNN-ABC: A new swarm optimization based technique for anomaly detection. *Computers*, 8(3), 59.

Rashid, T. A., & Abdullah, S. M. (2018). A Hybrid of Artificial Bee Colony, Genetic Algorithm, and Neural Network for Diabetic Mellitus Diagnosing. *ARO-The Scientific Journal of Koya University*, 6(1), pp.55-64.

Ratna, D. (2022). *2.17.1.5: Selective Laser Melting Technology*. [Online] Available at: <https://www.sciencedirect.com/topics/materials-science/selective-laser-melting>. [Accessed: 25 October 2022].

(Redcrab), *SoftMax Functioning*. [Online] Available at: <https://www.redcrab-software.com/en/Calculator/Softmax> [Accessed: 12 January 2023].

Rickenbacher, L., Spierings, A. and Wegener, K. (2013). An integrated cost-model for selective laser melting (SLM). *Rapid Prototyping Journal*.

Saric, T., Simunovic, G., and Simunovic, K. (2013). Use of neural networks in prediction and simulation of steel surface roughness. *International Journal of Simulation Modelling*, 12(4), pp.225-236.

Saxena, A., Goebel, K., Simon, D., and Eklund, N. (2008, October). Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation. *In the Proceedings of the 1st International Conference on Prognostics and Health Management (PHM08)*, Denver CO.

Saxena, S. (2021). *LSTM | Introduction to LSTM | Long Short Term Memory*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/> [Accessed: 4 January 2023].

Shah, A., Kadam, E., Shah, H., Shinde, S., and Shingade, S. (2016, September). Deep residual networks with exponential linear unit. *In Proceedings of the Third International Symposium on Computer Vision and the Internet*, pp.59-65.

Shrestha, S., Starr, T., and Chou, K. (2019). A study of keyhole porosity in selective laser melting: single-track scanning with micro-CT analysis. *Journal of Manufacturing Science and Engineering*, 141(7).

Singh, A. K., Ganapathysubramanian, B., Sarkar, S., and Singh, A. (2018). Deep learning for plant stress phenotyping: trends and future perspectives. *Trends in plant science*.

Sinha, T., Verma, B., and Haidar, A. (2017). Optimization of convolutional neural network parameters for image classification. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, pp.1-7.

Snell, R., Tammam-Williams, S., Chechik, L., Lyle, A., Hernández-Nava, E., Boig, C., and Todd, I. (2020). Methods for rapid pore classification in metal additive manufacturing. *JOM*, 72(1), pp.101-109.

(Stackexchange), *Why does Bayesian Optimization perform poorly in more than 20 Dimensions?* [Online] Available at: <https://stats.stackexchange.com/questions/564528/why-does-bayesian-optimization-perform-poorly-in-more-than-20-dimensions#:~:text=Disadvantages%20of%20Bayesian%20Optimization%3A%201%20Requires%20the%20true,20%20di> [Accessed: 17 November 2022].

Sun, S., Brandt, M., and Easton, M. (2017). Powder bed fusion processes: An overview. *Laser additive manufacturing*, pp.55-77.

Tan, Q., Liu, Y., Fan, Z., Zhang, J., Yin, Y., and Zhang, M. X. (2020). Effect of processing parameters on the densification of an additively manufactured 2024 Al alloy. *Journal of Materials Science & Technology*, 58, pp.34-45.

Tch, A. (2017). *The mostly complete chart of Neural Networks, explained*. [Online] Available at: <https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464> [Accessed: 7 October 2019].

Thakur, D. (2018). *LSTM and its equations*. [Online] Available at: <https://medium.com/@divyanshu132/lstm-and-its-equations-5ee9246d04af> [Accessed: 4 January 2023].

(The learning machine), *Long Short-Term Memory (LSTM) Deep Learning*. [Online] Available at: <https://the-learning-machine.com/article/dl/long-short-term-memory> [Accessed: 4 January 2023].

(TriMech), *A Brief History of Additive Manufacturing* . [Online] Available at: <https://trimech.com/blog/a-brief-history-of-additive-manufacturing> [Accessed: 25 October 2022].

Varikuti, M. (2021). *LSTM Network*. [Online] Available at: <https://medium.com/mllearning-ai/lstm-networks-75d44ac8280f> [Accessed: 4 January 2023].

Wang, C., Tan, X. P., Tor, S. B., and Lim, C. S. (2020). Machine learning in additive manufacturing: State-of-the-art and perspectives. *Additive Manufacturing*, 36, 101538.

Wang, J., Ma, Y., Zhang, L., Gao, R. X., and Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, pp.144-156.

Wood, T. *SoftMax Functioning*. [Online] Available at: <https://deepai.org/machine-learning-glossary-and-terms/softmax-layer> [Accessed: 12 January 2023].

Wu, B., Liu, Z., Yuan, Z., Sun, G., and Wu, C. (2017, September). Reducing overfitting in deep convolutional neural networks using redundancy regularizer. In *International Conference on Artificial Neural Networks*, Springer, Cham, pp.49-55.

Wu, J. (2017). Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5, 23.

Wuest, T., Weimer, D., Irgens, C., and Thoben, K. D. (2016). Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1), pp.23-45.

Xu, F., Pun, C. M., Li, H., Zhang, Y., Song, Y., and Gao, H. (2019). Training feed-forward artificial neural networks with a modified artificial bee colony algorithm. *Neurocomputing*.

Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4), pp.611-629.

Zeybek, S., Pham, D. T., Koc, E., and Secer, A. (2021). An Improved Bees Algorithm for Training Deep Recurrent Networks for Sentiment Classification. *Symmetry*, 13(8), 1347.

Zhang, B., Liu, S., and Shin, Y. C. (2019). In-Process monitoring of porosity during laser additive manufacturing process. *Additive Manufacturing*, 28, pp.497-505.

Zhang, H., Kiranyaz, S., and Gabbouj, M. (2018). Finding better topologies for deep convolutional neural networks by evolution. arXiv preprint arXiv:1809.03242.

Appendix 1: MATLAB Codes

1.1 Convolutional Neural Network (CNN)

```
clc;  
clear;  
close all;
```

%Load and Explore Image Data

```
url = 'https://www.cs.toronto.edu/~kriz/cifar-10-matlab.tar.gz';  
cifar10DataDir = tempdir;  
filename = fullfile(cifar10DataDir,'cifar-10-matlab.tar.gz');  
  
dataFolder = fullfile(cifar10DataDir,'cifar-10-batches-mat');  
if ~exist(dataFolder,'dir')  
    fprintf("Downloading CIFAR-10 dataset (175 MB)... ");  
    websave(filename,url);  
    untar(filename,cifar10DataDir);  
    fprintf("Done.\n")  
end  
oldpath = addpath(fullfile(matlabroot,'examples','nnet','main'));  
  
[XTraining,YTraining,XTesting,YTesting] = loadCIFARData(cifar10DataDir);  
idx = randperm(numel(YTesting),5000);  
XValidation = XTesting(:, :, :, idx);  
XTesting(:, :, :, idx) = [];  
YValidation = YTesting(idx);  
YTesting(idx) = [];
```

%Specify Convolutional Neural Network Architecture

```
imageSize = [32 32 3];
numClasses = numel(unique(YTraining));

layers = [
    imageInputLayer(imageSize)

    convBlock(3,11,3)

    maxPooling2dLayer(3,'Stride',2,'Padding','same')

    convBlock(3,22,3)

    maxPooling2dLayer(3,'Stride',2,'Padding','same')

    convBlock(3,44,3)

    averagePooling2dLayer(8)

    fullyConnectedLayer(numClasses)
    softmaxLayer
    classificationLayer];

miniBatchSize = 256;
validationFrequency = floor(numel(YTraining)/miniBatchSize);
options = trainingOptions('sgdm', ...
    'InitialLearnRate', 0.1, ...
    'Momentum', 0.9, ...
    'L2Regularization', 1e-10, ...
    'MaxEpochs',30, ...
    'MiniBatchSize',miniBatchSize, ...
```

```
'ValidationFrequency',validationFrequency, ...
'ValidationData',{XValidation,YValidation}, ...
'Shuffle','every-epoch', ...
'Plots','training-progress');
```

%Train Network using Training Data

```
net = trainNetwork(XTraining,YTraining,layers,options);
```

%Classify Images and Compute Accuracy and Error

```
PredictedTraining = classify(net,XTraining);
```

```
TrainingAccuracy = sum(PredictedTraining == YTraining)/numel(YTraining);
```

```
TrainingError = 1 - TrainingAccuracy;
```

```
PredictedValidation = classify(net,XValidation);
```

```
ValidationAccuracy = sum(PredictedValidation == YValidation)/numel(YValidation);
```

```
ValidationError = 1 - ValidationAccuracy;
```

```
PredictedTesting = classify(net,XTesting);
```

```
TestingAccuracy = sum(PredictedTesting == YTesting)/numel(YTesting);
```

```
TestingError = 1 - TestingAccuracy;
```

%Create Confusion Matrix

```
figure('Units','normalized','Position',[0.2 0.2 0.4 0.4]);
```

```
    TrainingCM = confusionchart(YTraining,PredictedTraining);
```

```
    TrainingCM.Title = 'Confusion Matrix for Training Data';
```

```
    TrainingCM.ColumnSummary = 'column-normalized';
```

```
    TrainingCM.RowSummary = 'row-normalized';
```

```
figure('Units','normalized','Position',[0.2 0.2 0.4 0.4]);
```

```
    ValidationCM = confusionchart(YValidation,PredictedValidation);
```

```
    ValidationCM.Title = 'Confusion Matrix for Validation Data';
```

```

ValidationCM.ColumnSummary = 'column-normalized';
ValidationCM.RowSummary = 'row-normalized';

figure('Units','normalized','Position',[0.2 0.2 0.4 0.4]);
TestingCM = confusionchart(YTesting,PredictedTesting);
TestingCM.Title = 'Confusion Matrix for Testing Data';
TestingCM.ColumnSummary = 'column-normalized';
TestingCM.RowSummary = 'row-normalized';

function layers = convBlock(FilterSize,NumberofFilters,SectionDepth)
layers = [
    convolution2dLayer(FilterSize,NumberofFilters,'Padding','same')
    batchNormalizationLayer
    reluLayer];
layers = repmat(layers,SectionDepth,1);
end

```

1.2 Bayesian Convolutional Neural Network (BO-CNN)

```

clc;
clear;
close all;

%Load and Explore Image Data
url = 'https://www.cs.toronto.edu/~kriz/cifar-10-matlab.tar.gz';
cifar10DataDir = tempdir;
filename = fullfile(cifar10DataDir,'cifar-10-matlab.tar.gz');

dataFolder = fullfile(cifar10DataDir,'cifar-10-batches-mat');
if ~exist(dataFolder,'dir')
    fprintf("Downloading CIFAR-10 dataset (175 MB)... ");
    websave(filename,url);
    untar(filename,cifar10DataDir);

```

```

    fprintf("Done.\n")
end
oldpath = addpath(fullfile(matlabroot,'examples','nnet','main'));

[XTraining, YTraining, XTesting, YTesting] = loadCIFARData(cifar10DataDir);
idx = randperm(numel(YTesting),5000);
XValidation = XTesting(:, :, idx);
XTesting(:, :, idx) = [];
YValidation = YTesting(idx);
YTesting(idx) = [];

%Define the Problem (Objective Function)
ObjFcn =
makeObjFcn(XTraining, YTraining, XValidation, YValidation, XTesting, YTesting);
optimVars = [
    optimizableVariable('SectionDepth',[1 3], 'Type', 'integer')
    optimizableVariable('InitialLearnRate',[1e-2 1], 'Transform', 'log')
    optimizableVariable('Momentum',[0.8 0.98])
    optimizableVariable('L2Regularization',[1e-10 1e-2], 'Transform', 'log')];

%Optimize Variables
BayesObject = bayesopt(ObjFcn, optimVars, ...
    'MaxTime', 4*60*60, ...
    'IsObjectiveDeterministic', false, ...
    'UseParallel', false);

function ObjFcn =
makeObjFcn(XTraining, YTraining, XValidation, YValidation, XTesting, YTesting)

ObjFcn = @ValidationErrorFunction;
function ValidationError = ValidationErrorFunction(optimVars)

```

%Specify Convolutional Neural Network Architecture

```
imageSize = [32 32 3];
```

```
numClasses = numel(unique(YTraining));
```

```
layers = [
```

```
    imageInputLayer(imageSize)
```

```
        convBlock(3,11,optimVars.SectionDepth)
```

```
        maxPooling2dLayer(3,'Stride',2,'Padding','same')
```

```
        convBlock(3,22,optimVars.SectionDepth)
```

```
        maxPooling2dLayer(3,'Stride',2,'Padding','same')
```

```
        convBlock(3,44,optimVars.SectionDepth)
```

```
        averagePooling2dLayer(8)
```

```
        fullyConnectedLayer(numClasses)
```

```
        softmaxLayer
```

```
        classificationLayer];
```

```
miniBatchSize = 256;
```

```
validationFrequency = floor(numel(YTraining)/miniBatchSize);
```

```
options = trainingOptions('sgdm', ...
```

```
    'InitialLearnRate',optimVars.InitialLearnRate, ...
```

```
    'Momentum',optimVars.Momentum, ...
```

```
    'L2Regularization',optimVars.L2Regularization, ...
```

```
    'MaxEpochs',30, ...
```



```
'MiniBatchSize',miniBatchSize, ...  
'ValidationFrequency',validationFrequency, ...  
'ValidationData',{XValidation,YValidation}, ...  
'Shuffle','every-epoch', ...  
'Plots','training-progress');
```

%Train Network using Training Data

```
net = trainNetwork(XTraining,YTraining,layers,options);
```

%Classify Images and Compute Accuracy and Error

```
PredictedTraining = classify(net,XTraining);
```

```
TrainingAccuracy = sum(PredictedTraining == YTraining)/numel(YTraining);
```

```
TrainingError = 1 - TrainingAccuracy;
```

```
PredictedValidation = classify(net,XValidation);
```

```
ValidationAccuracy = sum(PredictedValidation == YValidation)/numel(YValidation);
```

```
ValidationError = 1 - ValidationAccuracy;
```

```
PredictedTesting = classify(net,XTesting);
```

```
TestingAccuracy = sum(PredictedTesting == YTesting)/numel(YTesting);
```

```
TestingError = 1 - TestingAccuracy;
```

%Create Confusion Matrix

```
figure('Units','normalized','Position',[0.2 0.2 0.4 0.4]);
```

```
    TrainingCM = confusionchart(YTraining,PredictedTraining);
```

```
    TrainingCM.Title = 'Confusion Matrix for Training Data';
```

```
    TrainingCM.ColumnSummary = 'column-normalized';
```

```
    TrainingCM.RowSummary = 'row-normalized';
```

```
figure('Units','normalized','Position',[0.2 0.2 0.4 0.4]);
```

```
    ValidationCM = confusionchart(YValidation,PredictedValidation);
```

```

ValidationCM.Title = 'Confusion Matrix for Validation Data';
ValidationCM.ColumnSummary = 'column-normalized';
ValidationCM.RowSummary = 'row-normalized';

figure('Units','normalized','Position',[0.2 0.2 0.4 0.4]);
TestingCM = confusionchart(YTesting,PredictedTesting);
TestingCM.Title = 'Confusion Matrix for Testing Data';
TestingCM.ColumnSummary = 'column-normalized';
TestingCM.RowSummary = 'row-normalized';

end

end

function layers = convBlock(FilterSize,NumberOfFilters,SectionDepth)
layers = [
    convolution2dLayer(FilterSize,NumberOfFilters,'Padding','same')
    batchNormalizationLayer
    reluLayer];
layers = repmat(layers,SectionDepth,1);
end

```

1.3 Bees Convolutional Neural Network (BA-CNN)

```

clc;
clear;
close all;

```

%Load and Explore Image Data

```

url = 'https://www.cs.toronto.edu/~kriz/cifar-10-matlab.tar.gz';
cifar10DataDir = tempdir;
filename = fullfile(cifar10DataDir,'cifar-10-matlab.tar.gz');

dataFolder = fullfile(cifar10DataDir,'cifar-10-batches-mat');

```

```

if ~exist(dataFolder,'dir')
    fprintf("Downloading CIFAR-10 dataset (175 MB)... ");
    websave(filename,url);
    untar(filename,cifar10DataDir);
    fprintf("Done.\n")
end
oldpath = addpath(fullfile(matlabroot,'examples','nnet','main'));

[XTraining,YTraining,XTesting,YTesting] = loadCIFARData(cifar10DataDir);
idx = randperm(numel(YTesting),5000);
XValidation = XTesting(:, :, idx);
XTesting(:, :, idx) = [];
YValidation = YTesting(idx);
YTesting(idx) = [];

%Define the Problem (Objective Function)
ObjFcn =
makeObjFcn(XTraining,YTraining,XValidation,YValidation,XTesting,YTesting);

nVar=4; %Number of Variables
VarSize=[1 nVar]; %Size of Variables
VarMin=[1 1e-2 0.8 1e-10]; %Lower Bound of Variables
VarMax=[3 1 0.98 1e-2]; %Upper Bound of Variables

%Define Bees Algorithm Parameters
MaxIt=1; %Maximum Number of Iterations
nScoutBee=6; %Scout Bees
nSelectedSite=round(0.5*nScoutBee); %Selected Sites
nEliteSite=1; %Selected Elite Sites
nSelectedSiteBee=round(0.5*nScoutBee); %Recruited Bees for Selected Sites

```

nEliteSiteBee=2*nSelectedSiteBee; %Recruited Bees for Elite Sites

r=0.1*(VarMax-VarMin); %Neighbourhood Radius

rdamp=0.95; %Neighbourhood Radius Damp Rate

%Initialize Empty Bee Structure

empty_bee.Position=[];

empty_bee.Error=[];

%Initialize Bees Array

bee= repmat(empty_bee,nScoutBee,1);

%Create New Solutions

for i=1:nScoutBee

 bee(i).Position=unifrnd(VarMin,VarMax,VarSize);

 bee(i).Error=ObjFcn(bee(i).Position);

end

%Sort the Solution

[~, SortOrder]=sort([bee.Error]);

bee=bee(SortOrder);

%Update Best Solution Ever Found

BestSol=bee(1);

%Create Array to Hold Best Error Values

LowestError=zeros(MaxIt,1);

%Create Bees Algorithm Main Loop

```
for it=1:MaxIt
```

%Elite Sites

```
for i=1:nEliteSite
```

```
    bestnewbee.Error=inf;
```

```
    for j=1:nEliteSiteBee
```

```
        newbee.Position=PerformBeeDance(bee(i).Position,r);
```

```
        newbee.Error=ObjFcn(newbee.Position);
```

```
        if newbee.Error<bestnewbee.Error
```

```
            bestnewbee=newbee;
```

```
        end
```

```
    end
```

```
    if bestnewbee.Error<bee(i).Error
```

```
        bee(i)=bestnewbee;
```

```
    end
```

```
end
```

%Selected Non-Elite Sites

```
for i=nEliteSite+1:nSelectedSite
```

```
    bestnewbee.Error=inf;
```

```
    for j=1:nSelectedSiteBee
```

```
        newbee.Position=PerformBeeDance(bee(i).Position,r);
```

```
        newbee.Error=ObjFcn(newbee.Position);
```

```
        if newbee.Error<bestnewbee.Error
```

```
            bestnewbee=newbee;
```

```
        end
```

```
    end
```

```

    if bestnewbee.Error<bee(i).Error
        bee(i)=bestnewbee;
    end
end

%Non-Selected Sites
for i=nSelectedSite+1:nScoutBee
    bee(i).Position=unifrnd(VarMin,VarMax,VarSize);
    bee(i).Error=ObjFcn(bee(i).Position);
end

%Sort the Solution
[~, SortOrder]=sort([bee.Error]);
bee=bee(SortOrder);

%Update Best Solution Ever Found
BestSol=bee(1);

%Store Best Error Ever Found
LowestError(it)=BestSol.Error;
OptimalSolution=BestSol.Position;

%Display Iteration Information
disp(['Iteration ' num2str(it) ': Lowest Error = ' num2str(LowestError(it))]);

%Define Damp Neighborhood Radius
r=r*rdamp;
end

```

%Display the Results

```
figure;
```

%Make a Plot for Lowest Error

```
semilogy(LowestError,'LineWidth',2);
```

```
xlabel('Iteration');
```

```
ylabel('Lowest Error');
```

```
function ObjFcn =
```

```
makeObjFcn(XTraining,YTraining,XValidation,YValidation,XTesting,YTesting)
```

```
ObjFcn = @ValidationErrorFunction;
```

```
function ValidationError = ValidationErrorFunction(OptimalSolution)
```

%Specify Convolutional Neural Network Architecture

```
imageSize = [32 32 3];
```

```
numClasses = numel(unique(YTraining));
```

```
layers = [
```

```
    imageInputLayer(imageSize)
```

```
        convBlock(3,11,round(OptimalSolution(1,1)))
```

```
        maxPooling2dLayer(3,'Stride',2,'Padding','same')
```

```
        convBlock(3,22,round(OptimalSolution(1,1)))
```

```
        maxPooling2dLayer(3,'Stride',2,'Padding','same')
```

```
        convBlock(3,44,round(OptimalSolution(1,1)))
```

```
        averagePooling2dLayer(8)
```

```
fullyConnectedLayer(numClasses)
softmaxLayer
classificationLayer];
```

```
miniBatchSize = 256;
validationFrequency = floor(numel(YTraining)/miniBatchSize);
options = trainingOptions('sgdm', ...
    'InitialLearnRate', OptimalSolution(1,2), ...
    'Momentum', OptimalSolution(1,3), ...
    'L2Regularization', OptimalSolution(1,4), ...
    'MaxEpochs', 30, ...
    'MiniBatchSize', miniBatchSize, ...
    'ValidationFrequency', validationFrequency, ...
    'ValidationData', {XValidation, YValidation}, ...
    'Shuffle', 'every-epoch', ...
    'Plots', 'training-progress');
```

%Train Network using Training Data

```
net = trainNetwork(XTraining, YTraining, layers, options);
```

%Classify Images and Compute Accuracy and Error

```
PredictedTraining = classify(net, XTraining);
```

```
TrainingAccuracy = sum(PredictedTraining == YTraining)/numel(YTraining);
```

```
TrainingError = 1 - TrainingAccuracy;
```

```
PredictedValidation = classify(net, XValidation);
```

```
ValidationAccuracy = sum(PredictedValidation == YValidation)/numel(YValidation);
```

```
ValidationError = 1 - ValidationAccuracy;
```



```

PredictedTesting = classify(net,XTesting);
TestingAccuracy = sum(PredictedTesting == YTesting)/numel(YTesting);
TestingError = 1 - TestingAccuracy;

```

%Create Confusion Matrix

```

figure('Units','normalized','Position',[0.2 0.2 0.4 0.4]);
    TrainingCM = confusionchart(YTraining,PredictedTraining);
    TrainingCM.Title = 'Confusion Matrix for Training Data';
    TrainingCM.ColumnSummary = 'column-normalized';
    TrainingCM.RowSummary = 'row-normalized';

figure('Units','normalized','Position',[0.2 0.2 0.4 0.4]);
    ValidationCM = confusionchart(YValidation,PredictedValidation);
    ValidationCM.Title = 'Confusion Matrix for Validation Data';
    ValidationCM.ColumnSummary = 'column-normalized';
    ValidationCM.RowSummary = 'row-normalized';

figure('Units','normalized','Position',[0.2 0.2 0.4 0.4]);
    TestingCM = confusionchart(YTesting,PredictedTesting);
    TestingCM.Title = 'Confusion Matrix for Testing Data';
    TestingCM.ColumnSummary = 'column-normalized';
    TestingCM.RowSummary = 'row-normalized';

end

end

```

```

function layers = convBlock(FilterSize,NumberofFilters,SectionDepth)
layers = [
    convolution2dLayer(FilterSize,NumberofFilters,'Padding','same')
    batchNormalizationLayer
    reluLayer];
layers = repmat(layers,SectionDepth,1);

```

```
end
```

1.4 Bees Bayesian Convolutional Neural Network (BA-BO-CNN)

```
clc;
```

```
clear;
```

```
close all;
```

%Load and Explore Image Data

```
url = 'https://www.cs.toronto.edu/~kriz/cifar-10-matlab.tar.gz';
```

```
cifar10DataDir = tempdir;
```

```
filename = fullfile(cifar10DataDir,'cifar-10-matlab.tar.gz');
```

```
dataFolder = fullfile(cifar10DataDir,'cifar-10-batches-mat');
```

```
if ~exist(dataFolder,'dir')
```

```
    fprintf("Downloading CIFAR-10 dataset (175 MB)... ");
```

```
    websave(filename,url);
```

```
    untar(filename,cifar10DataDir);
```

```
    fprintf("Done.\n")
```

```
end
```

```
oldpath = addpath(fullfile(matlabroot,'examples','nnet','main'));
```

```
[XTraining,YTraining,XTesting,YTesting] = loadCIFARData(cifar10DataDir);
```

```
idx = randperm(numel(YTesting),5000);
```

```
XValidation = XTesting(:,:,idx);
```

```
XTesting(:,:,idx) = [];
```

```
YValidation = YTesting(idx);
```

```
YTesting(idx) = [];
```

%Define the Problem (Objective Function)

```
ObjFcn = makeObjFcn(XTraining,YTraining,XValidation,YValidation,  
XTesting,YTesting);
```

```

nVar=4; %Number of Weight Learning Rate Factors
VarSize=[1 nVar]; %Matrix Size for Factors
VarMin=0.9; %Lower Bound for Factors
VarMax=1.1; %Upper Bound for Factors

%Define Bees Algorithm Parameters
MaxIt=1; %Maximum Number of Iterations
nScoutBee=6; %Scout Bees
nSelectedSite=round(0.5*nScoutBee); %Selected Sites
nEliteSite=1; %Selected Elite Sites
nSelectedSiteBee=round(0.5*nScoutBee); %Recruited Bees for Selected Sites
nEliteSiteBee=2*nSelectedSiteBee; %Recruited Bees for Elite Sites
r=0.1*(VarMax-VarMin); %Neighbourhood Radius
rdamp=0.95; %Neighbourhood Radius Damp Rate

%Initialize Empty Bee Structure
empty_bee.Position=[];
empty_bee.Error=[];

%Initialize Bees Array
bee= repmat(empty_bee,nScoutBee,1);

%Create New Solutions
for i=1:nScoutBee
    bee(i).Position=unifrnd(VarMin,VarMax,VarSize);
    bee(i).Error=ObjFcn(bee(i).Position);
end

%Sort the Solution
[~, SortOrder]=sort([bee.Error]);
bee=bee(SortOrder);

```

%Update Best Solution Ever Found

```
BestSol=bee(1);
```

%Create Array to Hold Best Error Values

```
LowestError=zeros(MaxIt,1);
```

%Create Bees Algorithm Main Loop

```
for it=1:MaxIt
```

%Elite Sites

```
for i=1:nEliteSite
```

```
    bestnewbee.Error=inf;
```

```
    for j=1:nEliteSiteBee
```

```
        newbee.Position=PerformBeeDance(bee(i).Position,r);
```

```
        newbee.Error=ObjFcn(newbee.Position);
```

```
        if newbee.Error<bestnewbee.Error
```

```
            bestnewbee=newbee;
```

```
        end
```

```
    end
```

```
    if bestnewbee.Error<bee(i).Error
```

```
        bee(i)=bestnewbee;
```

```
    end
```

```
end
```

%Selected Non-Elite Sites

```
for i=nEliteSite+1:nSelectedSite
```

```
    bestnewbee.Error=inf;
```

```

for j=1:nSelectedSiteBee
    newbee.Position=PerformBeeDance(bee(i).Position,r);
    newbee.Error=ObjFcn(newbee.Position);
    if newbee.Error<bestnewbee.Error
        bestnewbee=newbee;
    end
end

if bestnewbee.Error<bee(i).Error
    bee(i)=bestnewbee;
end
end

%Non-Selected Sites
for i=nSelectedSite+1:nScoutBee
    bee(i).Position=unifrnd(VarMin,VarMax,VarSize);
    bee(i).Error=ObjFcn(bee(i).Position);
end

%Sort the Solution
[~, SortOrder]=sort([bee.Error]);
bee=bee(SortOrder);

%Update Best Solution Ever Found
BestSol=bee(1);

%Store Best Error Ever Found
LowestError(it)=BestSol.Error;
OptimalWeightLearningRateFactor=BestSol.Position;

```

```

%Display Iteration Information
disp(['Iteration ' num2str(it) ': Lowest Error = ' num2str(LowestError(it))]);

%Define Damp Neighborhood Radius
r=r*rdamp;
end

%Display the Results
figure;

%Make a Plot for Lowest Error
semilogy(LowestError,'LineWidth',2);
xlabel('Iteration');
ylabel('Lowest Error');

function ObjFcn = makeObjFcn(XTraining,YTraining,XValidation,YValidation,
XTesting,YTesting)

ObjFcn = @ValidationErrorFunction;
function ValidationError =
ValidationErrorFunction(OptimalWeightLearningRateFactor)

%Specify Convolutional Neural Network Architecture
imageSize = [32 32 3];
numClasses = numel(unique(YTraining));

layers = [
    imageInputLayer(imageSize)

    convBlock(3,11,3,OptimalWeightLearningRateFactor(1,1))

```

```

maxPooling2dLayer(3,'Stride',2,'Padding','same')

convBlock(3,22,3,OptimalWeightLearningRateFactor(1,2))

maxPooling2dLayer(3,'Stride',2,'Padding','same')

convBlock(3,44,3,OptimalWeightLearningRateFactor(1,3))

averagePooling2dLayer(8)

fullyConnectedLayer(numClasses,'WeightLearnRateFactor',OptimalWeightLearningRate
Factor(1,4))
    softmaxLayer
    classificationLayer];

miniBatchSize = 256;
validationFrequency = floor(numel(YTraining)/miniBatchSize);
options = trainingOptions('sgdm', ...
    'InitialLearnRate',0.68934, ...
    'Momentum',0.84074, ...
    'L2Regularization',4.5857e-05, ...
    'MaxEpochs',30, ...
    'MiniBatchSize',miniBatchSize, ...
    'ValidationFrequency',validationFrequency, ...
    'ValidationData',{XValidation,YValidation}, ...
    'Shuffle','every-epoch', ...
    'Plots','training-progress');

%Train Network using Training Data
net = trainNetwork(XTraining,YTraining,layers,options);

```

%Classify Images and Compute Accuracy and Error

```
PredictedTraining = classify(net,XTraining);
TrainingAccuracy = sum(PredictedTraining == YTraining)/numel(YTraining);
TrainingError = 1 - TrainingAccuracy;

PredictedValidation = classify(net,XValidation);
ValidationAccuracy = sum(PredictedValidation == YValidation)/numel(YValidation);
ValidationError = 1 - ValidationAccuracy;

PredictedTesting = classify(net,XTesting);
TestingAccuracy = sum(PredictedTesting == YTesting)/numel(YTesting);
TestingError = 1 - TestingAccuracy;
```

%Create Confusion Matrix

```
figure('Units','normalized','Position',[0.2 0.2 0.4 0.4]);
    TrainingCM = confusionchart(YTraining,PredictedTraining);
    TrainingCM.Title = 'Confusion Matrix for Training Data';
    TrainingCM.ColumnSummary = 'column-normalized';
    TrainingCM.RowSummary = 'row-normalized';

figure('Units','normalized','Position',[0.2 0.2 0.4 0.4]);
    ValidationCM = confusionchart(YValidation,PredictedValidation);
    ValidationCM.Title = 'Confusion Matrix for Validation Data';
    ValidationCM.ColumnSummary = 'column-normalized';
    ValidationCM.RowSummary = 'row-normalized';

figure('Units','normalized','Position',[0.2 0.2 0.4 0.4]);
    TestingCM = confusionchart(YTesting,PredictedTesting);
    TestingCM.Title = 'Confusion Matrix for Testing Data';
    TestingCM.ColumnSummary = 'column-normalized';
    TestingCM.RowSummary = 'row-normalized';
```



```

end
end

function layers = convBlock(FilterSize,NumberOfFilters,SectionDepth,
OptimalWeightLearningRateFactor)
layers = [

convolution2dLayer(FilterSize,NumberOfFilters,'Padding','same','WeightLearnRateFactor'
,OptimalWeightLearningRateFactor)
    batchNormalizationLayer
    reluLayer];
layers = repmat(layers,SectionDepth,1);
end

```

1.5 Artificial Porosity Images Creation and Labelling

This code creates artificial porosity images with different shapes of the pores and different pixel values for the background and more importantly for pore itself, in addition it calculates the actual percent of porosity by identifying the ratio of the area of pores to the total area of the surface, (MathWorks-5) was the guide in writing this code.

```
clear ; clc ; clf ;
```

%Define the Main Loop

```
for x = 1:30
```

%Set the Number of Pores to Plot

```
n = [33; 33; 33; 31; 32; 32; 30; 30; 30; 28; 28; 29; 27; 27; 27; 25; 25; 26; 23; 24; 25; 22;
22; 23; 20; 21; 19; 20; 17; 15];
```

```
Radius = zeros(n(x, 1), 1);
```

```
Position = zeros(n(x, 1), 3);
```

```
Random_X_Position = randn(n(x, 1),1);
```

```
Random_X_Position(Random_X_Position<0) = 0;
```

```
Normalized_X_Position = Random_X_Position / max(Random_X_Position);
```

```
X_Position = 0.4854*(Normalized_X_Position)+ 0.2742;
```

```
Random_Y_Position = randn(n(x, 1),1);
```

```
Random_Y_Position(Random_Y_Position<0) = 0;
```

```
Normalized_Y_Position = Random_Y_Position / max(Random_Y_Position);
```

```
Y_Position = 0.4455*(Normalized_Y_Position)+ 0.2685;
```

```
Z_Position = (X_Position + Y_Position) / 2;
```

```
r = [0.0369469; 0.0373645; 0.0377821; 0.0360138; 0.036849; 0.0376842; 0.0350807;  
0.0363335; 0.0375863; 0.0341476; 0.035818; 0.0374884; 0.0332145; 0.0353025;  
0.0373905; 0.0322814; 0.034787; 0.0372926; 0.0313483; 0.0342715; 0.0371947;  
0.030301166; 0.033694166; 0.037087166; 0.0294821; 0.0332405; 0.028782275;  
0.032853875; 0.0276159; 0.026506542];
```

```
for idx = 1:n(x, 1)
```

```
    is_good = false;
```

```
    %Generate Positions and Radius
```

```
        Radius(idx) = ((r(x, 1) + rand * (-0.01:0.01))/ 2);
```

```
        Position(idx, :) = [X_Position(idx) Y_Position(idx) Z_Position(idx)];
```

```
    %Ensure we are inside the big surface
```

```
        if ((sqrt(sum(Position(idx, :).^2)) + Radius(idx)) < 0.7)
```

```
            && ((idx == 1) || ...
```

```
                all(sqrt(sum((Position(1:(idx-1), :) - repmat(Position(idx, :), idx-1, 1)).^2, 2)) >
```

```
                    Radius(1:(idx-1))+Radius(idx))) %All distances are bigger than the pores radius sum
```

```
            end
```

```
        end
```

%Generate Surface

```
clf;
```

```
hold on
```

```
daspect([1, 1, 1])
```

```
set(gca,'XColor','none','YColor','none','ZColor','none');
```

```
h(1) = axes('Position',[0 0 1 1]);
```

```
vert = [1 1 0;
```

```
0 1 0;
```

```
0 1 1;
```

```
1 1 1;
```

```
0 0 1;
```

```
1 0 1;
```

```
1 0 0;
```

```
0 0 0];
```

```
fac = [1 2 3 4;
```

```
4 3 5 6;
```

```
6 7 8 5;
```

```
1 2 8 7;
```

```
6 7 1 4;
```

```
2 3 5 8];
```

```
patch('Faces',fac,'Vertices',vert,'FaceColor', uint8([150 150 150]),'EdgeColor', uint8([150  
150 150]));
```

```
material metal;
```

```
alpha('color');
```

```
axis([0, 1, 0, 1, 0, 1]);
```

```
axis equal;
```

```
set(gca,'XColor','none','YColor','none','ZColor','none');
```

hold on;

%Generate Pores

for idx = 1:round(n(x, 1)/4)

[x1,x2,x3] = sphere(24);

surf((x1*Radius(idx))+0.1+Position(idx,1), (x2*Radius(idx))+0.1+Position(idx,2),
(x3*Radius(idx))+0.1+Position(idx,3),'FaceColor', uint8([110 110 110]),'EdgeColor',
uint8([110 110 110]));

surf((x1*Radius(idx))+0.1+Position(idx,1), (x2*Radius(idx))+0.1+Position(idx,2),
(x3*1.5*Radius(idx))+0.1+Position(idx,3),'FaceColor', uint8([112 112 112]),'EdgeColor',
uint8([112 112 112]));

material metal;

alphamap('rampdown');

axis([0, 1, 0, 1, 0, 1]);

axis equal;

set(gca,'XColor','none','YColor','none','ZColor','none');

end

for idx = round((n(x, 1)/4)+1):round(n(x, 1)/2)

[x1,x2,x3] = sphere(24);

surf((x1*Radius(idx))+0.3+Position(idx,1), (x2*Radius(idx))+0.3+Position(idx,2),
(x3*Radius(idx))+0.3+Position(idx,3),'FaceColor', uint8([114 114 114]),'EdgeColor',
uint8([114 114 114]));

surf((x1*Radius(idx))+0.3+Position(idx,1), (x2*1.5*Radius(idx))+0.3+Position(idx,2),
(x3*Radius(idx))+0.3+Position(idx,3),'FaceColor', uint8([116 116 116]),'EdgeColor',
uint8([116 116 116]));

material metal;

alphamap('rampdown');

axis([0, 1, 0, 1, 0, 1]);

```

axis equal;
set(gca,'XColor','none','YColor','none','ZColor','none');

end

for idx = round((n(x, 1)/2)+1):round(3*n(x, 1)/4)
[x1,x2,x3] = sphere(24);
surf((x1*Radius(idx))+0.2+Position(idx,1), (x2*Radius(idx))+0.2+Position(idx,2),
(x3*Radius(idx))+0.2+Position(idx,3),'FaceColor', uint8([118 118 118]),'EdgeColor',
uint8([118 118 118]));
surf((x1*1.5*Radius(idx))+0.2+Position(idx,1), (x2*Radius(idx))+0.2+Position(idx,2),
(x3*Radius(idx))+0.2+Position(idx,3),'FaceColor', uint8([120 120 120]),'EdgeColor',
uint8([120 120 120]));
material metal;
alphamap('rampdown');
axis([0, 1, 0, 1, 0, 1]);
axis equal;
set(gca,'XColor','none','YColor','none','ZColor','none');

end

for idx = round((3*n(x, 1)/4)+1):n(x, 1)
[x1,x2,x3] = sphere(24);
surf((x1*Radius(idx))+Position(idx,1), (x2*Radius(idx))+Position(idx,2),
(x3*Radius(idx))+Position(idx,3),'FaceColor', uint8([122 122 122]),'EdgeColor',
uint8([122 122 122]));
surf((x1*1.5*Radius(idx))+Position(idx,1), (x2*0.5*Radius(idx))+Position(idx,2),
(x3*1.5*Radius(idx))+Position(idx,3),'FaceColor', uint8([124 124 124]),'EdgeColor',
uint8([124 124 124]));
material metal;
alphamap('rampdown');

```

```

axis([0, 1, 0, 1, 0, 1]);
axis equal;
set(gca,'XColor','none','YColor','none','ZColor','none');

end

DiameterandPosition(1:n(x),(4*x)-3) = 2 * Radius;
DiameterandPosition(1:n(x),((4*x)-2):4*x) = Position;

view(3)

saveas(gca, "LP-SS" + x + " 3D.jpg")

for i = 0.01:0.01:1

S = surface([0 1; 0 1], [0 0; 1 1], [i i; i i],'FaceColor', uint8([255 255 255]), 'EdgeColor',
uint8([255 255 255]));

axis([0, 1, 0, 1, 0, i+0.0099]);

set(gca,'XColor','none','YColor','none','ZColor','none');

SN = 100*i;
view(2);

saveas(gca, "LP-SS" + x + "." + SN + " White Background.jpg")

Im = imread("LP-SS" + x + "." + SN + " White Background.jpg");
Imc = imcrop(Im,centerCropWindow2d(size(Im),[650 630]));

```

```

Actual_Percent_of_Pores(round(SN + ((x-1)*100)), 1) = ((numel(find(Imc(:,:,) == 110))
+numel(find(Imc(:,:,) == 112)) + numel(find(Imc(:,:,) == 114)) + numel(find(Imc(:,:,)
== 116)) + numel(find(Imc(:,:,) == 118)) + numel(find(Imc(:,:,) == 120)) +
numel(find(Imc(:,:,) == 122)) + numel(find(Imc(:,:,) == 124))) / (650*630*3)) * 100;

```

```

Background = imread("Background (" + SN + ").jpg");

```

```

Imcd = Imc - 100;

```

```

alph = 0.125;

```

```

sza = size(Imcd);

```

```

ArtificialImages = Background;

```

```

Croi = ArtificialImages(end-sza(1)+1:end,end-sza(2)+1:end,:);

```

```

ArtificialImages = Imcd*alph + Croi*(1-alph);

```

```

imwrite(ArtificialImages, "Slice LP-SS" + x + "." + SN + ".jpg")

```

```

end

```

```

end

```

```

for Cube = 1:30

```

```

for Slice = 1:100

```

```

Ref = imread("Ref (" + Slice + ").jpg");

```

```

Art = imread("Slice LP-SS" + Cube + "." + Slice + ".jpg");

```

```

[ssimval(round(Slice + ((Cube-1)*100)),1),ssimmap] = ssim(Art,Ref);

```

```

AvgSSI = sum(ssimval,'all') / 3000;

```

```

imshow(ssimmap,[])

```

```

AbsDiffImage = imabsdiff(Art,Ref);

```

```

AvgDiffImage(round(Slice + ((Cube-1)*100)),1) = sum(AbsDiffImage,'all') /
(650*630*3);

```

```

AvgofAvgDiffImage = sum(AvgDiffImage,'all') / 3000;

```

```

T = adaptthresh(Art, 0.69);
IB = imbinarize(Art,T);
imshowpair(Art, IB, 'montage')

Binary_Percent_of_Pores(round(Slice + ((Cube-1)*100), 1) = ((numel(find(IB(:,:,.) ==
0))) / (650*630*3)) * 100;

imwrite(IB, "LP-SS" + Cube + "." + Slice + " After Binarization.jpg")

end
end

```

1.6 Actual Percent of Porosity Dataset

The actual percent of porosity calculated in the previous code by identifying the ratio of the area of pores to the total area of the surface is shown in the following table appendix 1.1:

Table Appendix 1.1: Actual Percent of Porosity Dataset

#	Actual Percent of Pores (%)	#	Actual Percent of Pores (%)	#	Actual Percent of Pores (%)
1	0.0000	1001	0.0000	2001	0.0000
2	0.0000	1002	0.0000	2002	0.0000
3	0.0000	1003	0.0000	2003	0.0000
4	0.0000	1004	0.0000	2004	0.0000
5	0.0000	1005	0.0000	2005	0.0000
6	0.0000	1006	0.0000	2006	0.0000
7	0.0000	1007	0.0000	2007	0.0000
8	0.0000	1008	0.0000	2008	0.0000
9	0.0000	1009	0.0000	2009	0.0000
10	0.0000	1010	0.0000	2010	0.0000
11	0.0000	1011	0.0000	2011	0.0000
12	0.0000	1012	0.0000	2012	0.0000
13	0.0000	1013	0.0000	2013	0.0000
14	0.0000	1014	0.0000	2014	0.0000
15	0.0000	1015	0.0000	2015	0.0000

16	0.0000	1016	0.0000	2016	0.0000
17	0.0000	1017	0.0000	2017	0.0000
18	0.0000	1018	0.0000	2018	0.0000
19	0.0000	1019	0.0000	2019	0.0000
20	0.0000	1020	0.0000	2020	0.0000
21	0.0000	1021	0.0000	2021	0.0000
22	0.0000	1022	0.0000	2022	0.0000
23	0.0000	1023	0.0239	2023	0.0000
24	0.0000	1024	0.0110	2024	0.0259
25	0.0000	1025	0.0154	2025	0.0266
26	0.0000	1026	0.0000	2026	0.0129
27	0.0000	1027	0.0000	2027	0.0840
28	0.0100	1028	0.0281	2028	0.0093
29	0.0132	1029	0.0276	2029	0.0000
30	0.0173	1030	0.0200	2030	0.0496
31	0.0000	1031	0.0449	2031	0.0125
32	0.0000	1032	0.0147	2032	0.0393
33	0.0000	1033	0.0063	2033	0.0105
34	0.0105	1034	0.0256	2034	0.0000
35	0.0327	1035	0.0190	2035	0.0000
36	0.0227	1036	0.0440	2036	0.0000
37	0.0227	1037	0.0623	2037	0.0000
38	0.0098	1038	0.0889	2038	0.0000
39	0.0295	1039	0.0422	2039	0.0000
40	0.0142	1040	0.0808	2040	0.0000
41	0.0264	1041	0.0549	2041	0.0000
42	0.0000	1042	0.0449	2042	0.0000
43	0.0332	1043	0.0476	2043	0.0000
44	0.0252	1044	0.0256	2044	0.0000
45	0.0322	1045	0.0376	2045	0.0000
46	0.0054	1046	0.0232	2046	0.0000
47	0.0000	1047	0.0000	2047	0.0000
48	0.0032	1048	0.0005	2048	0.0000
49	0.0144	1049	0.0110	2049	0.0000
50	0.0291	1050	0.0151	2050	0.0000
51	0.0613	1051	0.0166	2051	0.0000
52	0.0195	1052	0.0374	2052	0.0000
53	0.0286	1053	0.0718	2053	0.0000
54	0.0107	1054	0.0476	2054	0.0000
55	0.0154	1055	0.0869	2055	0.0000

56	0.0000	1056	0.0381	2056	0.0000
57	0.0000	1057	0.0171	2057	0.0000
58	0.0000	1058	0.0525	2058	0.0000
59	0.0198	1059	0.0249	2059	0.0000
60	0.0117	1060	0.0122	2060	0.0000
61	0.0176	1061	0.0313	2061	0.0000
62	0.0227	1062	0.0261	2062	0.0000
63	0.0085	1063	0.0122	2063	0.0000
64	0.0017	1064	0.0071	2064	0.0000
65	0.0000	1065	0.0244	2065	0.0000
66	0.0000	1066	0.0049	2066	0.0000
67	0.0000	1067	0.0000	2067	0.0024
68	0.0000	1068	0.0032	2068	0.0000
69	0.0000	1069	0.0183	2069	0.0000
70	0.0000	1070	0.0190	2070	0.0054
71	0.0000	1071	0.0335	2071	0.0271
72	0.0010	1072	0.0205	2072	0.0320
73	0.0103	1073	0.0386	2073	0.0505
74	0.0151	1074	0.0215	2074	0.0694
75	0.0161	1075	0.0222	2075	0.0252
76	0.0071	1076	0.0107	2076	0.0161
77	0.0000	1077	0.0005	2077	0.0054
78	0.0000	1078	0.0000	2078	0.0000
79	0.0000	1079	0.0000	2079	0.0000
80	0.0000	1080	0.0000	2080	0.0000
81	0.0000	1081	0.0000	2081	0.0000
82	0.0000	1082	0.0000	2082	0.0000
83	0.0000	1083	0.0000	2083	0.0000
84	0.0000	1084	0.0000	2084	0.0000
85	0.0000	1085	0.0000	2085	0.0000
86	0.0000	1086	0.0000	2086	0.0000
87	0.0000	1087	0.0000	2087	0.0000
88	0.0000	1088	0.0000	2088	0.0000
89	0.0000	1089	0.0000	2089	0.0000
90	0.0000	1090	0.0000	2090	0.0000
91	0.0000	1091	0.0000	2091	0.0000
92	0.0000	1092	0.0000	2092	0.0000
93	0.0000	1093	0.0000	2093	0.0000
94	0.0000	1094	0.0000	2094	0.0000
95	0.0000	1095	0.0000	2095	0.0000

96	0.0000	1096	0.0000	2096	0.0000
97	0.0000	1097	0.0000	2097	0.0000
98	0.0000	1098	0.0000	2098	0.0000
99	0.0000	1099	0.0000	2099	0.0000
100	0.0000	1100	0.0000	2100	0.0000
101	0.0000	1101	0.0000	2101	0.0000
102	0.0000	1102	0.0000	2102	0.0000
103	0.0000	1103	0.0000	2103	0.0000
104	0.0000	1104	0.0000	2104	0.0000
105	0.0000	1105	0.0000	2105	0.0000
106	0.0000	1106	0.0000	2106	0.0000
107	0.0000	1107	0.0000	2107	0.0000
108	0.0000	1108	0.0000	2108	0.0000
109	0.0000	1109	0.0000	2109	0.0000
110	0.0000	1110	0.0000	2110	0.0000
111	0.0000	1111	0.0000	2111	0.0000
112	0.0000	1112	0.0000	2112	0.0000
113	0.0000	1113	0.0000	2113	0.0000
114	0.0000	1114	0.0000	2114	0.0000
115	0.0000	1115	0.0000	2115	0.0000
116	0.0000	1116	0.0000	2116	0.0000
117	0.0000	1117	0.0000	2117	0.0000
118	0.0000	1118	0.0000	2118	0.0000
119	0.0000	1119	0.0000	2119	0.0000
120	0.0000	1120	0.0000	2120	0.0000
121	0.0000	1121	0.0000	2121	0.0015
122	0.0000	1122	0.0088	2122	0.0020
123	0.0000	1123	0.0295	2123	0.0017
124	0.0000	1124	0.0125	2124	0.0000
125	0.0000	1125	0.0371	2125	0.0000
126	0.0015	1126	0.0000	2126	0.0000
127	0.0090	1127	0.0000	2127	0.0000
128	0.0049	1128	0.0000	2128	0.0000
129	0.0347	1129	0.0000	2129	0.0000
130	0.0164	1130	0.0000	2130	0.0000
131	0.0562	1131	0.0269	2131	0.0000
132	0.0225	1132	0.0129	2132	0.0000
133	0.0488	1133	0.0232	2133	0.0000
134	0.0295	1134	0.0310	2134	0.0125
135	0.0259	1135	0.0117	2135	0.0000

136	0.0122	1136	0.0752	2136	0.0000
137	0.0286	1137	0.0376	2137	0.0000
138	0.0151	1138	0.0840	2138	0.0000
139	0.0122	1139	0.0554	2139	0.0000
140	0.0000	1140	0.0759	2140	0.0000
141	0.0000	1141	0.0698	2141	0.0000
142	0.0000	1142	0.0510	2142	0.0000
143	0.0000	1143	0.0269	2143	0.0000
144	0.0000	1144	0.0144	2144	0.0000
145	0.0000	1145	0.0259	2145	0.0000
146	0.0000	1146	0.0325	2146	0.0000
147	0.0000	1147	0.0264	2147	0.0000
148	0.0000	1148	0.0581	2148	0.0000
149	0.0000	1149	0.0591	2149	0.0000
150	0.0000	1150	0.0596	2150	0.0000
151	0.0335	1151	0.0303	2151	0.0000
152	0.0171	1152	0.0371	2152	0.0000
153	0.0593	1153	0.0706	2153	0.0000
154	0.0300	1154	0.0388	2154	0.0000
155	0.0308	1155	0.0444	2155	0.0000
156	0.0205	1156	0.0295	2156	0.0000
157	0.0151	1157	0.0139	2157	0.0000
158	0.0335	1158	0.0186	2158	0.0000
159	0.0347	1159	0.0103	2159	0.0000
160	0.0237	1160	0.0061	2160	0.0000
161	0.0063	1161	0.0098	2161	0.0000
162	0.0000	1162	0.0134	2162	0.0000
163	0.0000	1163	0.0315	2163	0.0000
164	0.0000	1164	0.0083	2164	0.0000
165	0.0000	1165	0.0129	2165	0.0000
166	0.0000	1166	0.0212	2166	0.0283
167	0.0115	1167	0.0188	2167	0.0391
168	0.0164	1168	0.0088	2168	0.0276
169	0.0142	1169	0.0325	2169	0.0525
170	0.0151	1170	0.0078	2170	0.0486
171	0.0020	1171	0.0000	2171	0.0364
172	0.0000	1172	0.0027	2172	0.0357
173	0.0061	1173	0.0149	2173	0.0642
174	0.0149	1174	0.0117	2174	0.0212
175	0.0151	1175	0.0166	2175	0.0217

176	0.0063	1176	0.0100	2176	0.0000
177	0.0000	1177	0.0000	2177	0.0054
178	0.0000	1178	0.0000	2178	0.0000
179	0.0000	1179	0.0000	2179	0.0000
180	0.0000	1180	0.0000	2180	0.0000
181	0.0000	1181	0.0000	2181	0.0000
182	0.0000	1182	0.0000	2182	0.0000
183	0.0000	1183	0.0000	2183	0.0000
184	0.0000	1184	0.0000	2184	0.0000
185	0.0000	1185	0.0000	2185	0.0000
186	0.0000	1186	0.0000	2186	0.0000
187	0.0000	1187	0.0000	2187	0.0000
188	0.0000	1188	0.0000	2188	0.0000
189	0.0000	1189	0.0000	2189	0.0000
190	0.0000	1190	0.0000	2190	0.0000
191	0.0000	1191	0.0000	2191	0.0000
192	0.0000	1192	0.0000	2192	0.0000
193	0.0000	1193	0.0000	2193	0.0000
194	0.0000	1194	0.0000	2194	0.0000
195	0.0000	1195	0.0000	2195	0.0000
196	0.0000	1196	0.0000	2196	0.0000
197	0.0000	1197	0.0000	2197	0.0000
198	0.0000	1198	0.0000	2198	0.0000
199	0.0000	1199	0.0000	2199	0.0000
200	0.0000	1200	0.0000	2200	0.0000
201	0.0000	1201	0.0000	2201	0.0000
202	0.0000	1202	0.0000	2202	0.0000
203	0.0000	1203	0.0000	2203	0.0000
204	0.0000	1204	0.0000	2204	0.0000
205	0.0000	1205	0.0000	2205	0.0000
206	0.0000	1206	0.0000	2206	0.0000
207	0.0000	1207	0.0000	2207	0.0000
208	0.0000	1208	0.0000	2208	0.0000
209	0.0000	1209	0.0000	2209	0.0000
210	0.0000	1210	0.0000	2210	0.0000
211	0.0000	1211	0.0000	2211	0.0000
212	0.0000	1212	0.0000	2212	0.0000
213	0.0000	1213	0.0107	2213	0.0000
214	0.0000	1214	0.0266	2214	0.0000
215	0.0000	1215	0.0147	2215	0.0000

216	0.0000	1216	0.0349	2216	0.0000
217	0.0000	1217	0.0000	2217	0.0000
218	0.0000	1218	0.0000	2218	0.0000
219	0.0000	1219	0.0000	2219	0.0000
220	0.0000	1220	0.0000	2220	0.0000
221	0.0000	1221	0.0000	2221	0.0000
222	0.0000	1222	0.0000	2222	0.0000
223	0.0000	1223	0.0000	2223	0.0000
224	0.0000	1224	0.0000	2224	0.0000
225	0.0032	1225	0.0000	2225	0.0000
226	0.0440	1226	0.0000	2226	0.0454
227	0.0134	1227	0.0000	2227	0.0386
228	0.0542	1228	0.0000	2228	0.0198
229	0.0398	1229	0.0000	2229	0.0000
230	0.0139	1230	0.0000	2230	0.0195
231	0.0728	1231	0.0388	2231	0.0000
232	0.0098	1232	0.0129	2232	0.0000
233	0.0234	1233	0.0291	2233	0.1101
234	0.0000	1234	0.0059	2234	0.0657
235	0.0000	1235	0.0000	2235	0.0466
236	0.0000	1236	0.0000	2236	0.0000
237	0.0000	1237	0.0242	2237	0.0000
238	0.0095	1238	0.0803	2238	0.0000
239	0.0381	1239	0.0906	2239	0.0000
240	0.0547	1240	0.0716	2240	0.0000
241	0.0637	1241	0.0972	2241	0.0000
242	0.0513	1242	0.0464	2242	0.0000
243	0.0615	1243	0.0505	2243	0.0000
244	0.0694	1244	0.0591	2244	0.0000
245	0.0549	1245	0.0520	2245	0.0000
246	0.0894	1246	0.0579	2246	0.0000
247	0.0281	1247	0.0161	2247	0.0000
248	0.1451	1248	0.0488	2248	0.0000
249	0.0359	1249	0.0081	2249	0.0000
250	0.0559	1250	0.0000	2250	0.0000
251	0.0200	1251	0.0632	2251	0.0000
252	0.0168	1252	0.0757	2252	0.0000
253	0.0139	1253	0.1092	2253	0.0000
254	0.0117	1254	0.1541	2254	0.0000
255	0.0159	1255	0.1153	2255	0.0000

256	0.0234	1256	0.0618	2256	0.0000
257	0.0530	1257	0.0383	2257	0.0000
258	0.0310	1258	0.0518	2258	0.0000
259	0.0525	1259	0.0422	2259	0.0000
260	0.0486	1260	0.0105	2260	0.0000
261	0.0274	1261	0.0237	2261	0.0000
262	0.0122	1262	0.1189	2262	0.0000
263	0.0303	1263	0.0918	2263	0.0000
264	0.0088	1264	0.0784	2264	0.0000
265	0.0115	1265	0.0945	2265	0.0000
266	0.0286	1266	0.0994	2266	0.0584
267	0.0037	1267	0.0469	2267	0.0173
268	0.0000	1268	0.0261	2268	0.0188
269	0.0000	1269	0.0176	2269	0.0493
270	0.0000	1270	0.0220	2270	0.0359
271	0.0000	1271	0.0173	2271	0.0120
272	0.0117	1272	0.0149	2272	0.0293
273	0.0293	1273	0.0444	2273	0.0562
274	0.0234	1274	0.0066	2274	0.0335
275	0.0266	1275	0.0000	2275	0.0300
276	0.0134	1276	0.0000	2276	0.0234
277	0.0005	1277	0.0000	2277	0.0066
278	0.0000	1278	0.0000	2278	0.0000
279	0.0000	1279	0.0000	2279	0.0000
280	0.0000	1280	0.0000	2280	0.0000
281	0.0000	1281	0.0000	2281	0.0000
282	0.0000	1282	0.0000	2282	0.0000
283	0.0000	1283	0.0000	2283	0.0000
284	0.0000	1284	0.0000	2284	0.0000
285	0.0000	1285	0.0000	2285	0.0000
286	0.0000	1286	0.0000	2286	0.0000
287	0.0000	1287	0.0000	2287	0.0000
288	0.0000	1288	0.0000	2288	0.0000
289	0.0000	1289	0.0000	2289	0.0000
290	0.0000	1290	0.0000	2290	0.0000
291	0.0000	1291	0.0000	2291	0.0000
292	0.0000	1292	0.0000	2292	0.0000
293	0.0000	1293	0.0000	2293	0.0000
294	0.0000	1294	0.0000	2294	0.0000
295	0.0000	1295	0.0000	2295	0.0000

296	0.0000	1296	0.0000	2296	0.0000
297	0.0000	1297	0.0000	2297	0.0000
298	0.0000	1298	0.0000	2298	0.0000
299	0.0000	1299	0.0000	2299	0.0000
300	0.0000	1300	0.0000	2300	0.0000
301	0.0000	1301	0.0000	2301	0.0000
302	0.0005	1302	0.0000	2302	0.0000
303	0.0007	1303	0.0000	2303	0.0000
304	0.0000	1304	0.0000	2304	0.0000
305	0.0000	1305	0.0000	2305	0.0000
306	0.0000	1306	0.0000	2306	0.0000
307	0.0000	1307	0.0000	2307	0.0000
308	0.0000	1308	0.0000	2308	0.0000
309	0.0000	1309	0.0000	2309	0.0000
310	0.0000	1310	0.0000	2310	0.0000
311	0.0000	1311	0.0000	2311	0.0000
312	0.0000	1312	0.0000	2312	0.0000
313	0.0000	1313	0.0000	2313	0.0000
314	0.0000	1314	0.0000	2314	0.0000
315	0.0000	1315	0.0000	2315	0.0000
316	0.0000	1316	0.0000	2316	0.0000
317	0.0000	1317	0.0000	2317	0.0000
318	0.0000	1318	0.0000	2318	0.0000
319	0.0000	1319	0.0000	2319	0.0000
320	0.0000	1320	0.0000	2320	0.0000
321	0.0186	1321	0.0000	2321	0.0017
322	0.0073	1322	0.0000	2322	0.0007
323	0.0112	1323	0.0000	2323	0.0010
324	0.0000	1324	0.0076	2324	0.0264
325	0.0000	1325	0.0361	2325	0.0125
326	0.0000	1326	0.0823	2326	0.0630
327	0.0000	1327	0.0596	2327	0.0000
328	0.0000	1328	0.0327	2328	0.0281
329	0.0000	1329	0.0342	2329	0.0000
330	0.0000	1330	0.0000	2330	0.0259
331	0.0002	1331	0.0005	2331	0.0256
332	0.0093	1332	0.0955	2332	0.0164
333	0.0090	1333	0.0413	2333	0.1140
334	0.0093	1334	0.0320	2334	0.0176
335	0.0017	1335	0.0989	2335	0.0308

336	0.0000	1336	0.0486	2336	0.0000
337	0.0000	1337	0.0147	2337	0.0000
338	0.0000	1338	0.0425	2338	0.0000
339	0.0000	1339	0.0022	2339	0.0000
340	0.0002	1340	0.0103	2340	0.0000
341	0.0137	1341	0.0833	2341	0.0000
342	0.0173	1342	0.0386	2342	0.0000
343	0.0581	1343	0.1160	2343	0.0000
344	0.0574	1344	0.0427	2344	0.0000
345	0.0466	1345	0.0969	2345	0.0000
346	0.0269	1346	0.0684	2346	0.0000
347	0.0295	1347	0.0513	2347	0.0000
348	0.0557	1348	0.0252	2348	0.0000
349	0.0232	1349	0.0288	2349	0.0000
350	0.0049	1350	0.0046	2350	0.0000
351	0.0339	1351	0.0000	2351	0.0000
352	0.0203	1352	0.0032	2352	0.0000
353	0.0584	1353	0.0716	2353	0.0000
354	0.0166	1354	0.0313	2354	0.0000
355	0.0339	1355	0.0513	2355	0.0000
356	0.0002	1356	0.0593	2356	0.0000
357	0.0105	1357	0.0054	2357	0.0000
358	0.0181	1358	0.0156	2358	0.0000
359	0.0110	1359	0.0371	2359	0.0000
360	0.0242	1360	0.0325	2360	0.0000
361	0.0044	1361	0.0168	2361	0.0000
362	0.0161	1362	0.0576	2362	0.0000
363	0.0044	1363	0.0371	2363	0.0000
364	0.0239	1364	0.0725	2364	0.0000
365	0.0044	1365	0.0647	2365	0.0000
366	0.0000	1366	0.0540	2366	0.0264
367	0.0000	1367	0.0078	2367	0.0059
368	0.0000	1368	0.0000	2368	0.0000
369	0.0000	1369	0.0071	2369	0.0000
370	0.0000	1370	0.0403	2370	0.0046
371	0.0037	1371	0.0164	2371	0.0161
372	0.0120	1372	0.0459	2372	0.0159
373	0.0227	1373	0.0420	2373	0.0342
374	0.0134	1374	0.0305	2374	0.0188
375	0.0151	1375	0.0364	2375	0.0176

376	0.0093	1376	0.0144	2376	0.0105
377	0.0000	1377	0.0012	2377	0.0007
378	0.0000	1378	0.0000	2378	0.0000
379	0.0000	1379	0.0000	2379	0.0000
380	0.0000	1380	0.0000	2380	0.0000
381	0.0000	1381	0.0000	2381	0.0000
382	0.0000	1382	0.0000	2382	0.0000
383	0.0000	1383	0.0000	2383	0.0000
384	0.0000	1384	0.0000	2384	0.0000
385	0.0000	1385	0.0000	2385	0.0000
386	0.0000	1386	0.0000	2386	0.0000
387	0.0000	1387	0.0000	2387	0.0000
388	0.0000	1388	0.0000	2388	0.0000
389	0.0000	1389	0.0000	2389	0.0000
390	0.0000	1390	0.0000	2390	0.0000
391	0.0000	1391	0.0000	2391	0.0000
392	0.0000	1392	0.0000	2392	0.0000
393	0.0000	1393	0.0000	2393	0.0000
394	0.0000	1394	0.0000	2394	0.0000
395	0.0000	1395	0.0000	2395	0.0000
396	0.0000	1396	0.0000	2396	0.0000
397	0.0000	1397	0.0000	2397	0.0000
398	0.0000	1398	0.0000	2398	0.0000
399	0.0000	1399	0.0000	2399	0.0000
400	0.0000	1400	0.0000	2400	0.0000
401	0.0000	1401	0.0000	2401	0.0000
402	0.0000	1402	0.0000	2402	0.0000
403	0.0000	1403	0.0000	2403	0.0000
404	0.0000	1404	0.0000	2404	0.0000
405	0.0000	1405	0.0000	2405	0.0000
406	0.0000	1406	0.0000	2406	0.0000
407	0.0000	1407	0.0000	2407	0.0010
408	0.0000	1408	0.0000	2408	0.0000
409	0.0000	1409	0.0000	2409	0.0000
410	0.0000	1410	0.0000	2410	0.0000
411	0.0000	1411	0.0000	2411	0.0000
412	0.0000	1412	0.0000	2412	0.0000
413	0.0000	1413	0.0000	2413	0.0000
414	0.0000	1414	0.0000	2414	0.0000
415	0.0000	1415	0.0000	2415	0.0000

416	0.0000	1416	0.0000	2416	0.0000
417	0.0000	1417	0.0000	2417	0.0000
418	0.0000	1418	0.0000	2418	0.0000
419	0.0000	1419	0.0000	2419	0.0000
420	0.0000	1420	0.0000	2420	0.0000
421	0.0000	1421	0.0000	2421	0.0000
422	0.0000	1422	0.0000	2422	0.0000
423	0.0000	1423	0.0115	2423	0.0000
424	0.0000	1424	0.0203	2424	0.0000
425	0.0000	1425	0.0176	2425	0.0000
426	0.0000	1426	0.0120	2426	0.0000
427	0.0364	1427	0.0098	2427	0.0000
428	0.0112	1428	0.0134	2428	0.0000
429	0.0398	1429	0.0076	2429	0.0491
430	0.0005	1430	0.0125	2430	0.0000
431	0.0105	1431	0.0139	2431	0.0000
432	0.0156	1432	0.0200	2432	0.0000
433	0.0447	1433	0.0271	2433	0.0398
434	0.0779	1434	0.0447	2434	0.0112
435	0.0816	1435	0.0405	2435	0.0000
436	0.0579	1436	0.0408	2436	0.0000
437	0.0728	1437	0.0222	2437	0.0000
438	0.0027	1438	0.0269	2438	0.0000
439	0.0000	1439	0.0037	2439	0.0000
440	0.0000	1440	0.0200	2440	0.0000
441	0.0000	1441	0.0664	2441	0.0000
442	0.0000	1442	0.0808	2442	0.0000
443	0.0364	1443	0.0608	2443	0.0000
444	0.0217	1444	0.0730	2444	0.0000
445	0.0303	1445	0.0611	2445	0.0000
446	0.0442	1446	0.0391	2446	0.0000
447	0.0198	1447	0.0198	2447	0.0000
448	0.0911	1448	0.0596	2448	0.0000
449	0.0559	1449	0.0230	2449	0.0000
450	0.1287	1450	0.0276	2450	0.0000
451	0.0501	1451	0.0317	2451	0.0000
452	0.0186	1452	0.0000	2452	0.0000
453	0.0987	1453	0.0484	2453	0.0000
454	0.0361	1454	0.0413	2454	0.0000
455	0.0427	1455	0.1045	2455	0.0000

456	0.0335	1456	0.0650	2456	0.0000
457	0.0032	1457	0.0759	2457	0.0000
458	0.0071	1458	0.0618	2458	0.0000
459	0.0342	1459	0.0571	2459	0.0000
460	0.0186	1460	0.0759	2460	0.0010
461	0.0562	1461	0.0400	2461	0.0000
462	0.0447	1462	0.0672	2462	0.0000
463	0.0264	1463	0.0723	2463	0.0000
464	0.0408	1464	0.0310	2464	0.0000
465	0.0215	1465	0.0310	2465	0.0000
466	0.0066	1466	0.0188	2466	0.0339
467	0.0147	1467	0.0000	2467	0.0327
468	0.0125	1468	0.0000	2468	0.0217
469	0.0103	1469	0.0000	2469	0.0464
470	0.0000	1470	0.0000	2470	0.0000
471	0.0000	1471	0.0000	2471	0.0256
472	0.0000	1472	0.0078	2472	0.0427
473	0.0000	1473	0.0383	2473	0.0515
474	0.0000	1474	0.0154	2474	0.0291
475	0.0000	1475	0.0188	2475	0.0320
476	0.0000	1476	0.0142	2476	0.0232
477	0.0000	1477	0.0017	2477	0.0066
478	0.0000	1478	0.0000	2478	0.0000
479	0.0000	1479	0.0000	2479	0.0000
480	0.0000	1480	0.0000	2480	0.0000
481	0.0000	1481	0.0000	2481	0.0000
482	0.0000	1482	0.0000	2482	0.0000
483	0.0000	1483	0.0000	2483	0.0000
484	0.0000	1484	0.0000	2484	0.0000
485	0.0000	1485	0.0000	2485	0.0000
486	0.0000	1486	0.0000	2486	0.0000
487	0.0000	1487	0.0000	2487	0.0000
488	0.0000	1488	0.0000	2488	0.0000
489	0.0000	1489	0.0000	2489	0.0000
490	0.0000	1490	0.0000	2490	0.0000
491	0.0000	1491	0.0000	2491	0.0000
492	0.0000	1492	0.0000	2492	0.0000
493	0.0000	1493	0.0000	2493	0.0000
494	0.0000	1494	0.0000	2494	0.0000
495	0.0000	1495	0.0000	2495	0.0000

496	0.0000	1496	0.0000	2496	0.0000
497	0.0000	1497	0.0000	2497	0.0000
498	0.0000	1498	0.0000	2498	0.0000
499	0.0000	1499	0.0000	2499	0.0000
500	0.0000	1500	0.0000	2500	0.0000
501	0.0000	1501	0.0000	2501	0.0000
502	0.0000	1502	0.0000	2502	0.0000
503	0.0000	1503	0.0000	2503	0.0000
504	0.0000	1504	0.0000	2504	0.0000
505	0.0000	1505	0.0000	2505	0.0000
506	0.0000	1506	0.0000	2506	0.0000
507	0.0000	1507	0.0000	2507	0.0000
508	0.0000	1508	0.0000	2508	0.0000
509	0.0000	1509	0.0000	2509	0.0000
510	0.0000	1510	0.0000	2510	0.0000
511	0.0000	1511	0.0000	2511	0.0000
512	0.0000	1512	0.0000	2512	0.0000
513	0.0000	1513	0.0000	2513	0.0242
514	0.0000	1514	0.0000	2514	0.0266
515	0.0000	1515	0.0000	2515	0.0195
516	0.0000	1516	0.0000	2516	0.0332
517	0.0000	1517	0.0000	2517	0.0000
518	0.0000	1518	0.0000	2518	0.0000
519	0.0000	1519	0.0000	2519	0.0000
520	0.0000	1520	0.0000	2520	0.0137
521	0.0000	1521	0.0000	2521	0.0349
522	0.0000	1522	0.0000	2522	0.0068
523	0.0000	1523	0.0044	2523	0.0000
524	0.0000	1524	0.0325	2524	0.0000
525	0.0000	1525	0.1309	2525	0.0000
526	0.0000	1526	0.0906	2526	0.0000
527	0.0000	1527	0.0999	2527	0.0288
528	0.0000	1528	0.0430	2528	0.0190
529	0.0000	1529	0.0298	2529	0.0466
530	0.0000	1530	0.1197	2530	0.0926
531	0.0220	1531	0.0313	2531	0.0894
532	0.0584	1532	0.0442	2532	0.0945
533	0.0669	1533	0.0391	2533	0.1238
534	0.0606	1534	0.0195	2534	0.0271
535	0.0190	1535	0.0007	2535	0.0000

536	0.0107	1536	0.0381	2536	0.0000
537	0.0354	1537	0.0173	2537	0.0000
538	0.0286	1538	0.0293	2538	0.0000
539	0.0469	1539	0.0061	2539	0.0000
540	0.0598	1540	0.0076	2540	0.0000
541	0.0608	1541	0.1031	2541	0.0000
542	0.0369	1542	0.0444	2542	0.0000
543	0.0095	1543	0.1739	2543	0.0000
544	0.0000	1544	0.0344	2544	0.0000
545	0.0000	1545	0.0630	2545	0.0000
546	0.0000	1546	0.0647	2546	0.0000
547	0.0000	1547	0.0708	2547	0.0137
548	0.0000	1548	0.0737	2548	0.0000
549	0.0000	1549	0.0913	2549	0.0000
550	0.0000	1550	0.0540	2550	0.0000
551	0.0000	1551	0.0542	2551	0.0332
552	0.0000	1552	0.0181	2552	0.0195
553	0.0488	1553	0.0466	2553	0.0266
554	0.0249	1554	0.1177	2554	0.0242
555	0.0471	1555	0.0657	2555	0.0000
556	0.0093	1556	0.0432	2556	0.0000
557	0.0107	1557	0.1140	2557	0.0000
558	0.0103	1558	0.0879	2558	0.0000
559	0.0000	1559	0.0369	2559	0.0000
560	0.0039	1560	0.0510	2560	0.0000
561	0.0110	1561	0.1074	2561	0.0000
562	0.0230	1562	0.0440	2562	0.0000
563	0.0210	1563	0.0579	2563	0.0000
564	0.0242	1564	0.0391	2564	0.0000
565	0.0352	1565	0.0198	2565	0.0000
566	0.0181	1566	0.0708	2566	0.0642
567	0.0090	1567	0.0598	2567	0.0171
568	0.0083	1568	0.0464	2568	0.0242
569	0.0129	1569	0.0574	2569	0.0217
570	0.0132	1570	0.0496	2570	0.0276
571	0.0210	1571	0.0488	2571	0.0137
572	0.0076	1572	0.0147	2572	0.0227
573	0.0168	1573	0.0017	2573	0.0444
574	0.0107	1574	0.0000	2574	0.0237
575	0.0129	1575	0.0000	2575	0.0208

576	0.0037	1576	0.0000	2576	0.0154
577	0.0000	1577	0.0000	2577	0.0063
578	0.0000	1578	0.0000	2578	0.0000
579	0.0000	1579	0.0000	2579	0.0000
580	0.0000	1580	0.0000	2580	0.0000
581	0.0000	1581	0.0000	2581	0.0000
582	0.0000	1582	0.0000	2582	0.0000
583	0.0000	1583	0.0000	2583	0.0000
584	0.0000	1584	0.0000	2584	0.0000
585	0.0000	1585	0.0000	2585	0.0000
586	0.0000	1586	0.0000	2586	0.0000
587	0.0000	1587	0.0000	2587	0.0000
588	0.0000	1588	0.0000	2588	0.0000
589	0.0000	1589	0.0000	2589	0.0000
590	0.0000	1590	0.0000	2590	0.0000
591	0.0000	1591	0.0000	2591	0.0000
592	0.0000	1592	0.0000	2592	0.0000
593	0.0000	1593	0.0000	2593	0.0000
594	0.0000	1594	0.0000	2594	0.0000
595	0.0000	1595	0.0000	2595	0.0000
596	0.0000	1596	0.0000	2596	0.0000
597	0.0000	1597	0.0000	2597	0.0000
598	0.0000	1598	0.0000	2598	0.0000
599	0.0000	1599	0.0000	2599	0.0000
600	0.0000	1600	0.0000	2600	0.0000
601	0.0000	1601	0.0000	2601	0.0000
602	0.0000	1602	0.0000	2602	0.0000
603	0.0000	1603	0.0000	2603	0.0000
604	0.0000	1604	0.0000	2604	0.0000
605	0.0000	1605	0.0000	2605	0.0000
606	0.0000	1606	0.0000	2606	0.0000
607	0.0000	1607	0.0000	2607	0.0000
608	0.0000	1608	0.0000	2608	0.0000
609	0.0000	1609	0.0000	2609	0.0000
610	0.0000	1610	0.0000	2610	0.0000
611	0.0000	1611	0.0000	2611	0.0000
612	0.0000	1612	0.0000	2612	0.0000
613	0.0000	1613	0.0000	2613	0.0000
614	0.0000	1614	0.0000	2614	0.0000
615	0.0000	1615	0.0000	2615	0.0000

616	0.0000	1616	0.0000	2616	0.0000
617	0.0000	1617	0.0000	2617	0.0000
618	0.0000	1618	0.0000	2618	0.0159
619	0.0000	1619	0.0000	2619	0.0371
620	0.0000	1620	0.0000	2620	0.0227
621	0.0039	1621	0.0000	2621	0.0000
622	0.0017	1622	0.0000	2622	0.0315
623	0.0051	1623	0.0000	2623	0.0000
624	0.0000	1624	0.0000	2624	0.0051
625	0.0000	1625	0.0000	2625	0.0066
626	0.0000	1626	0.0120	2626	0.0000
627	0.0000	1627	0.0400	2627	0.0000
628	0.0015	1628	0.0799	2628	0.0068
629	0.1343	1629	0.0674	2629	0.0000
630	0.0725	1630	0.0320	2630	0.0476
631	0.0974	1631	0.0833	2631	0.0000
632	0.0872	1632	0.0105	2632	0.0449
633	0.0000	1633	0.0435	2633	0.0198
634	0.0000	1634	0.0063	2634	0.0000
635	0.0469	1635	0.0737	2635	0.0000
636	0.0095	1636	0.0637	2636	0.0000
637	0.0505	1637	0.0476	2637	0.0000
638	0.0000	1638	0.0569	2638	0.0000
639	0.0000	1639	0.0095	2639	0.0000
640	0.0000	1640	0.0000	2640	0.0000
641	0.0015	1641	0.0481	2641	0.0000
642	0.0339	1642	0.0408	2642	0.0000
643	0.1170	1643	0.0781	2643	0.0000
644	0.0608	1644	0.0410	2644	0.0000
645	0.0818	1645	0.0332	2645	0.0000
646	0.1143	1646	0.0823	2646	0.0000
647	0.0430	1647	0.0523	2647	0.0227
648	0.0090	1648	0.0186	2648	0.0371
649	0.0259	1649	0.0327	2649	0.0159
650	0.0247	1650	0.0230	2650	0.0000
651	0.0706	1651	0.0183	2651	0.0000
652	0.0298	1652	0.0186	2652	0.0000
653	0.0708	1653	0.0864	2653	0.0000
654	0.0698	1654	0.1050	2654	0.0000
655	0.0679	1655	0.0703	2655	0.0000

656	0.0418	1656	0.0908	2656	0.0000
657	0.0398	1657	0.0733	2657	0.0000
658	0.0510	1658	0.0259	2658	0.0000
659	0.0647	1659	0.0586	2659	0.0000
660	0.0623	1660	0.0503	2660	0.0000
661	0.0435	1661	0.0330	2661	0.0000
662	0.0926	1662	0.0520	2662	0.0000
663	0.0188	1663	0.0344	2663	0.0000
664	0.0215	1664	0.0181	2664	0.0000
665	0.0068	1665	0.0347	2665	0.0000
666	0.0071	1666	0.0227	2666	0.0562
667	0.0215	1667	0.0274	2667	0.0457
668	0.0105	1668	0.0078	2668	0.0291
669	0.0200	1669	0.0088	2669	0.0479
670	0.0098	1670	0.0217	2670	0.0879
671	0.0010	1671	0.0274	2671	0.0276
672	0.0078	1672	0.0274	2672	0.0513
673	0.0274	1673	0.0361	2673	0.0505
674	0.0144	1674	0.0190	2674	0.0151
675	0.0161	1675	0.0166	2675	0.0168
676	0.0186	1676	0.0120	2676	0.0137
677	0.0044	1677	0.0005	2677	0.0007
678	0.0000	1678	0.0000	2678	0.0000
679	0.0000	1679	0.0000	2679	0.0000
680	0.0000	1680	0.0000	2680	0.0000
681	0.0000	1681	0.0000	2681	0.0000
682	0.0000	1682	0.0000	2682	0.0000
683	0.0000	1683	0.0000	2683	0.0000
684	0.0000	1684	0.0000	2684	0.0000
685	0.0000	1685	0.0000	2685	0.0000
686	0.0000	1686	0.0000	2686	0.0000
687	0.0000	1687	0.0000	2687	0.0000
688	0.0000	1688	0.0000	2688	0.0000
689	0.0000	1689	0.0000	2689	0.0000
690	0.0000	1690	0.0000	2690	0.0000
691	0.0000	1691	0.0000	2691	0.0000
692	0.0000	1692	0.0000	2692	0.0000
693	0.0000	1693	0.0000	2693	0.0000
694	0.0000	1694	0.0000	2694	0.0000
695	0.0000	1695	0.0000	2695	0.0000

696	0.0000	1696	0.0000	2696	0.0000
697	0.0000	1697	0.0000	2697	0.0000
698	0.0000	1698	0.0000	2698	0.0000
699	0.0000	1699	0.0000	2699	0.0000
700	0.0000	1700	0.0000	2700	0.0000
701	0.0000	1701	0.0000	2701	0.0000
702	0.0000	1702	0.0000	2702	0.0000
703	0.0000	1703	0.0000	2703	0.0000
704	0.0000	1704	0.0000	2704	0.0000
705	0.0000	1705	0.0000	2705	0.0000
706	0.0000	1706	0.0000	2706	0.0000
707	0.0000	1707	0.0000	2707	0.0000
708	0.0000	1708	0.0000	2708	0.0000
709	0.0000	1709	0.0000	2709	0.0000
710	0.0000	1710	0.0000	2710	0.0000
711	0.0000	1711	0.0000	2711	0.0000
712	0.0000	1712	0.0000	2712	0.0000
713	0.0000	1713	0.0000	2713	0.0000
714	0.0000	1714	0.0000	2714	0.0000
715	0.0000	1715	0.0000	2715	0.0000
716	0.0000	1716	0.0000	2716	0.0000
717	0.0000	1717	0.0286	2717	0.0000
718	0.0000	1718	0.0181	2718	0.0000
719	0.0000	1719	0.0230	2719	0.0000
720	0.0000	1720	0.0220	2720	0.0000
721	0.0000	1721	0.0056	2721	0.0000
722	0.0000	1722	0.0093	2722	0.0000
723	0.0000	1723	0.0122	2723	0.0000
724	0.0000	1724	0.0059	2724	0.0000
725	0.0000	1725	0.0000	2725	0.0000
726	0.0000	1726	0.0000	2726	0.0000
727	0.0000	1727	0.0000	2727	0.0906
728	0.0000	1728	0.0000	2728	0.0354
729	0.0000	1729	0.0535	2729	0.0632
730	0.0000	1730	0.0134	2730	0.0960
731	0.0012	1731	0.1106	2731	0.0484
732	0.0249	1732	0.0281	2732	0.0459
733	0.0139	1733	0.0476	2733	0.0625
734	0.0225	1734	0.0024	2734	0.0422
735	0.0254	1735	0.0000	2735	0.0706

736	0.0242	1736	0.0000	2736	0.0000
737	0.0205	1737	0.0364	2737	0.0000
738	0.0860	1738	0.0200	2738	0.0000
739	0.0298	1739	0.0386	2739	0.0000
740	0.0657	1740	0.0567	2740	0.0000
741	0.0352	1741	0.0747	2741	0.0000
742	0.0068	1742	0.0537	2742	0.0000
743	0.0806	1743	0.1426	2743	0.0000
744	0.0657	1744	0.0476	2744	0.0000
745	0.0708	1745	0.0713	2745	0.0000
746	0.1243	1746	0.0281	2746	0.0000
747	0.1314	1747	0.0674	2747	0.0000
748	0.1094	1748	0.0559	2748	0.0000
749	0.0777	1749	0.0650	2749	0.0000
750	0.0571	1750	0.0298	2750	0.0000
751	0.0591	1751	0.0073	2751	0.0000
752	0.0232	1752	0.0200	2752	0.0000
753	0.0676	1753	0.0313	2753	0.0000
754	0.0425	1754	0.0662	2754	0.0000
755	0.0462	1755	0.1045	2755	0.0000
756	0.0210	1756	0.0518	2756	0.0000
757	0.0332	1757	0.0581	2757	0.0000
758	0.0107	1758	0.0247	2758	0.0000
759	0.0000	1759	0.0308	2759	0.0000
760	0.0000	1760	0.0325	2760	0.0000
761	0.0010	1761	0.0234	2761	0.0000
762	0.0217	1762	0.0447	2762	0.0000
763	0.0545	1763	0.0383	2763	0.0000
764	0.0379	1764	0.0203	2764	0.0000
765	0.0632	1765	0.0339	2765	0.0000
766	0.0542	1766	0.0227	2766	0.0457
767	0.0056	1767	0.0393	2767	0.0569
768	0.0000	1768	0.0166	2768	0.0317
769	0.0066	1769	0.0100	2769	0.0552
770	0.0212	1770	0.0000	2770	0.0708
771	0.0078	1771	0.0000	2771	0.0161
772	0.0168	1772	0.0078	2772	0.0327
773	0.0095	1773	0.0195	2773	0.0784
774	0.0000	1774	0.0112	2774	0.0271
775	0.0000	1775	0.0186	2775	0.0315

776	0.0000	1776	0.0147	2776	0.0388
777	0.0000	1777	0.0032	2777	0.0073
778	0.0000	1778	0.0000	2778	0.0000
779	0.0000	1779	0.0000	2779	0.0000
780	0.0000	1780	0.0000	2780	0.0000
781	0.0000	1781	0.0000	2781	0.0000
782	0.0000	1782	0.0000	2782	0.0000
783	0.0000	1783	0.0000	2783	0.0000
784	0.0000	1784	0.0000	2784	0.0000
785	0.0000	1785	0.0000	2785	0.0000
786	0.0000	1786	0.0000	2786	0.0000
787	0.0000	1787	0.0000	2787	0.0000
788	0.0000	1788	0.0000	2788	0.0000
789	0.0000	1789	0.0000	2789	0.0000
790	0.0000	1790	0.0000	2790	0.0000
791	0.0000	1791	0.0000	2791	0.0000
792	0.0000	1792	0.0000	2792	0.0000
793	0.0000	1793	0.0000	2793	0.0000
794	0.0000	1794	0.0000	2794	0.0000
795	0.0000	1795	0.0000	2795	0.0000
796	0.0000	1796	0.0000	2796	0.0000
797	0.0000	1797	0.0000	2797	0.0000
798	0.0000	1798	0.0000	2798	0.0000
799	0.0000	1799	0.0000	2799	0.0000
800	0.0000	1800	0.0000	2800	0.0000
801	0.0000	1801	0.0000	2801	0.0000
802	0.0000	1802	0.0000	2802	0.0000
803	0.0000	1803	0.0000	2803	0.0000
804	0.0000	1804	0.0000	2804	0.0000
805	0.0000	1805	0.0000	2805	0.0000
806	0.0000	1806	0.0000	2806	0.0000
807	0.0000	1807	0.0000	2807	0.0000
808	0.0000	1808	0.0000	2808	0.0000
809	0.0000	1809	0.0000	2809	0.0000
810	0.0000	1810	0.0000	2810	0.0000
811	0.0000	1811	0.0000	2811	0.0000
812	0.0000	1812	0.0000	2812	0.0000
813	0.0000	1813	0.0000	2813	0.0000
814	0.0000	1814	0.0000	2814	0.0000
815	0.0000	1815	0.0000	2815	0.0000

816	0.0039	1816	0.0000	2816	0.0000
817	0.0098	1817	0.0000	2817	0.0000
818	0.0051	1818	0.0000	2818	0.0000
819	0.0068	1819	0.0000	2819	0.0000
820	0.0000	1820	0.0000	2820	0.0000
821	0.0000	1821	0.0000	2821	0.0000
822	0.0000	1822	0.0000	2822	0.0000
823	0.0313	1823	0.0000	2823	0.0000
824	0.0168	1824	0.0000	2824	0.0000
825	0.0230	1825	0.0498	2825	0.0000
826	0.0063	1826	0.0237	2826	0.0000
827	0.0242	1827	0.0684	2827	0.0000
828	0.0117	1828	0.0256	2828	0.0244
829	0.0781	1829	0.0757	2829	0.0000
830	0.0115	1830	0.0618	2830	0.0293
831	0.0515	1831	0.0654	2831	0.0000
832	0.0256	1832	0.0449	2832	0.0000
833	0.0217	1833	0.0950	2833	0.0491
834	0.0205	1834	0.0369	2834	0.0493
835	0.0288	1835	0.0000	2835	0.1380
836	0.0000	1836	0.0000	2836	0.0000
837	0.0000	1837	0.0000	2837	0.0000
838	0.0181	1838	0.0000	2838	0.0000
839	0.0488	1839	0.0000	2839	0.0000
840	0.0308	1840	0.0000	2840	0.0000
841	0.0640	1841	0.0000	2841	0.0000
842	0.0000	1842	0.0000	2842	0.0000
843	0.0063	1843	0.0000	2843	0.0000
844	0.0129	1844	0.0000	2844	0.0000
845	0.0156	1845	0.0000	2845	0.0000
846	0.0037	1846	0.0000	2846	0.0000
847	0.0606	1847	0.0000	2847	0.0000
848	0.0261	1848	0.0000	2848	0.0000
849	0.0364	1849	0.0000	2849	0.0000
850	0.0000	1850	0.0000	2850	0.0000
851	0.0000	1851	0.0000	2851	0.0000
852	0.0000	1852	0.0000	2852	0.0000
853	0.0239	1853	0.0000	2853	0.0000
854	0.0227	1854	0.0000	2854	0.0000
855	0.0525	1855	0.0000	2855	0.0000

856	0.0103	1856	0.0000	2856	0.0000
857	0.0129	1857	0.0000	2857	0.0000
858	0.0112	1858	0.0000	2858	0.0000
859	0.0037	1859	0.0000	2859	0.0000
860	0.0364	1860	0.0000	2860	0.0000
861	0.0222	1861	0.0000	2861	0.0000
862	0.0349	1862	0.0000	2862	0.0000
863	0.0654	1863	0.0000	2863	0.0000
864	0.0217	1864	0.0000	2864	0.0000
865	0.0190	1865	0.0000	2865	0.0000
866	0.0237	1866	0.0359	2866	0.0603
867	0.0007	1867	0.0044	2867	0.0403
868	0.0000	1868	0.0000	2868	0.0186
869	0.0000	1869	0.0000	2869	0.0134
870	0.0000	1870	0.0000	2870	0.0166
871	0.0000	1871	0.0000	2871	0.0066
872	0.0046	1872	0.0171	2872	0.0078
873	0.0151	1873	0.0755	2873	0.0274
874	0.0178	1874	0.0276	2874	0.0200
875	0.0171	1875	0.0366	2875	0.0183
876	0.0098	1876	0.0230	2876	0.0164
877	0.0000	1877	0.0085	2877	0.0044
878	0.0000	1878	0.0000	2878	0.0000
879	0.0000	1879	0.0000	2879	0.0000
880	0.0000	1880	0.0000	2880	0.0000
881	0.0000	1881	0.0000	2881	0.0000
882	0.0000	1882	0.0000	2882	0.0000
883	0.0000	1883	0.0000	2883	0.0000
884	0.0000	1884	0.0000	2884	0.0000
885	0.0000	1885	0.0000	2885	0.0000
886	0.0000	1886	0.0000	2886	0.0000
887	0.0000	1887	0.0000	2887	0.0000
888	0.0000	1888	0.0000	2888	0.0000
889	0.0000	1889	0.0000	2889	0.0000
890	0.0000	1890	0.0000	2890	0.0000
891	0.0000	1891	0.0000	2891	0.0000
892	0.0000	1892	0.0000	2892	0.0000
893	0.0000	1893	0.0000	2893	0.0000
894	0.0000	1894	0.0000	2894	0.0000
895	0.0000	1895	0.0000	2895	0.0000

896	0.0000	1896	0.0000	2896	0.0000
897	0.0000	1897	0.0000	2897	0.0000
898	0.0000	1898	0.0000	2898	0.0000
899	0.0000	1899	0.0000	2899	0.0000
900	0.0000	1900	0.0000	2900	0.0000
901	0.0000	1901	0.0000	2901	0.0000
902	0.0000	1902	0.0000	2902	0.0000
903	0.0000	1903	0.0000	2903	0.0000
904	0.0000	1904	0.0000	2904	0.0000
905	0.0000	1905	0.0000	2905	0.0000
906	0.0000	1906	0.0000	2906	0.0000
907	0.0000	1907	0.0000	2907	0.0000
908	0.0000	1908	0.0000	2908	0.0000
909	0.0000	1909	0.0000	2909	0.0000
910	0.0000	1910	0.0000	2910	0.0000
911	0.0000	1911	0.0000	2911	0.0000
912	0.0000	1912	0.0000	2912	0.0000
913	0.0000	1913	0.0000	2913	0.0000
914	0.0000	1914	0.0000	2914	0.0000
915	0.0000	1915	0.0000	2915	0.0000
916	0.0000	1916	0.0000	2916	0.0000
917	0.0000	1917	0.0000	2917	0.0000
918	0.0000	1918	0.0000	2918	0.0000
919	0.0000	1919	0.0000	2919	0.0254
920	0.0000	1920	0.0000	2920	0.0339
921	0.0000	1921	0.0000	2921	0.0127
922	0.0000	1922	0.0000	2922	0.0442
923	0.0000	1923	0.0000	2923	0.0242
924	0.0000	1924	0.0000	2924	0.0059
925	0.0344	1925	0.0000	2925	0.0076
926	0.0593	1926	0.0000	2926	0.0369
927	0.0383	1927	0.0000	2927	0.0000
928	0.0437	1928	0.0000	2928	0.0466
929	0.0632	1929	0.0000	2929	0.0000
930	0.0696	1930	0.0000	2930	0.0449
931	0.0689	1931	0.0000	2931	0.0249
932	0.0327	1932	0.0000	2932	0.0278
933	0.0855	1933	0.0591	2933	0.0000
934	0.0208	1934	0.0708	2934	0.0000
935	0.0264	1935	0.0249	2935	0.0000

936	0.0269	1936	0.0000	2936	0.0000
937	0.0000	1937	0.0000	2937	0.0000
938	0.0000	1938	0.0000	2938	0.0000
939	0.1033	1939	0.0000	2939	0.0000
940	0.1375	1940	0.0000	2940	0.0000
941	0.1006	1941	0.0000	2941	0.0000
942	0.1114	1942	0.0000	2942	0.0000
943	0.1131	1943	0.0000	2943	0.0000
944	0.0708	1944	0.0000	2944	0.0000
945	0.0823	1945	0.0000	2945	0.0000
946	0.0659	1946	0.0000	2946	0.0000
947	0.0095	1947	0.0000	2947	0.0339
948	0.0142	1948	0.0000	2948	0.0254
949	0.0274	1949	0.0000	2949	0.0000
950	0.0315	1950	0.0000	2950	0.0000
951	0.0405	1951	0.0000	2951	0.0000
952	0.0520	1952	0.0000	2952	0.0000
953	0.1368	1953	0.0000	2953	0.0000
954	0.0611	1954	0.0000	2954	0.0000
955	0.0305	1955	0.0000	2955	0.0000
956	0.0376	1956	0.0000	2956	0.0000
957	0.0000	1957	0.0000	2957	0.0000
958	0.0132	1958	0.0000	2958	0.0000
959	0.0527	1959	0.0000	2959	0.0000
960	0.0882	1960	0.0000	2960	0.0000
961	0.0479	1961	0.0000	2961	0.0000
962	0.1319	1962	0.0000	2962	0.0000
963	0.1038	1963	0.0000	2963	0.0000
964	0.0266	1964	0.0000	2964	0.0000
965	0.0366	1965	0.0000	2965	0.0000
966	0.0171	1966	0.0000	2966	0.0625
967	0.0000	1967	0.0000	2967	0.0339
968	0.0000	1968	0.0000	2968	0.0320
969	0.0000	1969	0.0000	2969	0.0234
970	0.0000	1970	0.0000	2970	0.0210
971	0.0000	1971	0.0134	2971	0.0359
972	0.0081	1972	0.0244	2972	0.0400
973	0.0195	1973	0.0339	2973	0.0447
974	0.0222	1974	0.0354	2974	0.0554
975	0.0203	1975	0.0230	2975	0.0310

976	0.0149	1976	0.0168	2976	0.0103
977	0.0039	1977	0.0034	2977	0.0000
978	0.0000	1978	0.0000	2978	0.0000
979	0.0000	1979	0.0000	2979	0.0000
980	0.0000	1980	0.0000	2980	0.0000
981	0.0000	1981	0.0000	2981	0.0000
982	0.0000	1982	0.0000	2982	0.0000
983	0.0000	1983	0.0000	2983	0.0000
984	0.0000	1984	0.0000	2984	0.0000
985	0.0000	1985	0.0000	2985	0.0000
986	0.0000	1986	0.0000	2986	0.0000
987	0.0000	1987	0.0000	2987	0.0000
988	0.0000	1988	0.0000	2988	0.0000
989	0.0000	1989	0.0000	2989	0.0000
990	0.0000	1990	0.0000	2990	0.0000
991	0.0000	1991	0.0000	2991	0.0000
992	0.0000	1992	0.0000	2992	0.0000
993	0.0000	1993	0.0000	2993	0.0000
994	0.0000	1994	0.0000	2994	0.0000
995	0.0000	1995	0.0000	2995	0.0000
996	0.0000	1996	0.0000	2996	0.0000
997	0.0000	1997	0.0000	2997	0.0000
998	0.0000	1998	0.0000	2998	0.0000
999	0.0000	1999	0.0000	2999	0.0000
1000	0.0000	2000	0.0000	3000	0.0000

1.7 Regression Convolutional Neural Network (RCNN)

clc;

clear;

close all;

for c = 1:1

SD = 3;

ILR = 0.01;

```
M = 0.8;
```

```
R = 1e-10;
```

%Load and Explore Image Data

```
ArtificialPorosityImagesDatasetPath =
```

```
fullfile('/nfs/home', 'store02', 'users', 'c.c1881324', 'Documents', 'MATLAB', '3000
```

```
Slices', '*.jpg');
```

```
[XTraining, YTrainingPoP, XValidation, YValidationPoP, XTesting, YTestingPoP] =
```

```
loadAPIData(ArtificialPorosityImagesDatasetPath);
```

%Specify Convolutional Neural Network Architecture

```
imageSize = [650 630 3];
```

```
layersPoP = [
```

```
    imageInputLayer(imageSize)
```

```
    convBlock(5, 8, SD(c))
```

```
    averagePooling2dLayer(4, 'Stride', 4)
```

```
    convBlock(5, 16, SD(c))
```

```
    averagePooling2dLayer(4, 'Stride', 4)
```

convBlock(5,32, SD(c))

averagePooling2dLayer(4,'Stride',4)

convBlock(5,64, SD(c))

averagePooling2dLayer(4,'Stride',4)

convBlock(5,128, SD(c))

fullyConnectedLayer(1)

regressionLayer];

miniBatchSize = 64;

validationFrequency = floor(1800/miniBatchSize);

optionsPoP = trainingOptions('sgdm', ...

'InitialLearnRate', ILR(c), ...

'Momentum', M(c), ...

'L2Regularization', R(c), ...

'MaxEpochs',20, ...

'MiniBatchSize',miniBatchSize, ...

'ValidationFrequency',validationFrequency, ...

```
'ValidationData',{XValidation,YValidationPoP(:,1)}, ...
```

```
'Shuffle','every-epoch', ...
```

```
'Plots','training-progress');
```

%Train Network using Training Data

```
netPoP = trainNetwork(XTraining,YTrainingPoP(:,1),layersPoP,optionsPoP);
```

%Predict Responses and Compute Accuracy and Error

```
PredictedTrainingPoP(:,c) = predict(netPoP,XTraining);
```

```
PredictedValidationPoP(:,c) = predict(netPoP,XValidation);
```

```
PredictedTestingPoP(:,c) = predict(netPoP,XTesting);
```

```
TrainingError(:,c) = YTrainingPoP(:,1) - PredictedTrainingPoP(:,c);
```

```
AbsoluteTrainingError(:,c) = abs(TrainingError(:,c));
```

```
AverageTrainingError(:,c) = sum(AbsoluteTrainingError(:,c))/1800;
```

```
numCorrectTrainingPoP(:,c) = sum(abs(TrainingError(:,c)) < 0.05);
```

```
TrainingAccuracyPoP(:,c) = (numCorrectTrainingPoP(:,c))/1800*100;
```

```
ValidationError(:,c) = YValidationPoP(:,1) - PredictedValidationPoP(:,c);
```

```
AbsoluteValidationError(:,c) = abs(ValidationError(:,c));
```

```
AverageValidationError(:,c) = sum(AbsoluteValidationError(:,c))/600;
```

```
numCorrectValidationPoP(:,c) = sum(abs(ValidationError(:,c)) < 0.05);
```

```
ValidationAccuracyPoP(:,c) = (numCorrectValidationPoP(:,c))/600*100;
```

```
TestingError(:,c) = YTestingPoP(:,1) - PredictedTestingPoP(:,c);
```

```
AbsoluteTestingError(:,c) = abs(TestingError(:,c));
```

```
AverageTestingError(:,c) = sum(AbsoluteTestingError(:,c))/600;
```

```
numCorrectTestingPoP(:,c) = sum(abs(TestingError(:,c))) < 0.05);
```

```
TestingAccuracyPoP(:,c) = (numCorrectTestingPoP(:,c))/600*100;
```

```
end
```

```
function layers = convBlock(FilterSize,NumberOfFilters,SectionDepth)
```

```
layers = [
```

```
    convolution2dLayer(FilterSize,NumberOfFilters,'Padding','same')
```

```
    batchNormalizationLayer
```

```
    reluLayer];
```

```
layers = repmat(layers,SectionDepth,1);
```

```
end
```

```
function [XTraining,YTrainingPoP,XValidation,YValidationPoP,XTesting,YTestingPoP]  
= loadAPIData(ArtificialPorosityImagesDatasetPath)
```

```
location = dir(ArtificialPorosityImagesDatasetPath);
```

Percent_of_Porosity = Actual_Percent_of_Porosity;

RN = [1834 919 935 220 2897 1891 32 675 2106 2725 424 1690 900 1548 2596 2947 505
1802 2706 149 2648 2793 165 2726 1859 1769 2022 878 564 2109 646 566 2376 2163
518 2496 898 522 1049 2192 2519 2221 620 1211 2152 2191 2617 618 2877 769 757 1692
164 2155 554 277 947 1509 98 2554 1382 2869 2633 413 2801 2934 2262 1327 2770 1575
288 1926 2749 2573 695 2100 1281 1052 2511 754 279 57 832 656 1446 2371 2092 2288
1097 1545 1474 2649 2293 1232 787 2986 745 1706 1868 2685 2539 124 130 2184 2838
893 2409 1259 348 1095 481 714 2359 1082 1387 605 729 2211 1343 1030 693 120 1621
245 1609 99 1990 2413 188 1210 558 369 1809 2954 1917 374 1463 1083 2993 2279 2052
574 1873 1318 1157 863 1921 2872 1835 1501 71 603 2622 150 181 1804 560 1783 2001
1262 1331 1003 1881 858 525 2411 191 2623 2704 1141 2873 2 1709 2004 93 551 136
1511 2777 1819 2333 658 2943 2720 1845 1160 1676 1849 2882 1889 425 1848 72 2738
2828 154 2448 1775 2007 2810 1672 1987 1335 1350 428 1818 1296 2615 2142 170 211
176 772 133 1524 1028 1131 1638 1554 2213 895 546 1078 1058 417 156 701 1340 226
463 306 1712 2951 1862 1231 2991 851 1522 2718 1406 2864 1726 1745 1882 1915 955
734 1488 1847 888 2529 2353 1269 2204 192 1297 1649 2165 1137 2802 2182 743 333
2072 980 299 1563 2942 2248 1753 2964 1999 610 2485 1770 2918 708 591 557 1694
883 723 1193 2681 1734 2158 1453 1960 1552 2318 95 1677 1334 549 478 2789 17 49
2745 1433 1705 1223 680 886 490 14 1724 1654 1627 2917 516 194 2978 790 2616 2299
83 2518 619 219 1197 337 1130 1518 1693 345 2657 1995 2462 407 437 2422 2274 1617
843 614 1055 2097 114 2386 1075 1233 2476 1385 107 20 74 710 1559 2084 1924 2329
1301 2662 382 2354 2205 2095 2740 978 1034 1896 469 258 1154 766 2081 908 1079
1349 1369 2880 2699 30 1739 2317 1644 2561 1088 1470 1564 2670 2003 1271 1902 493
1416 152 1235 2077 1580 1172 2296 1589 1779 1332 48 2860 2196 765 323 2937 1206
738 1442 303 666 1777 1852 2096 2765 361 1346 2562 573 2116 148 167 1831 826 61
233 611 2341 995 474 1152 296 2384 456 781 2636 2316 1814 2028 2507 2688 1898 2432
1407 458 1909 236 609 586 2727 2513 2352 486 1910 2228 2652 1582 108 2217 497 2345
1481 2983 1494 1333 2350 2344 655 2939 771 435 2536 529 483 2672 457 2925 2383
215 360 2788 1830 1539 2874 805 561 2995 1824 1056 1313 945 2578 2415 1648 661
1674 2187 2973 2249 2927 1581 933 2210 706 2754 1550 2368 179 2401 1435 2881 1951

422 531 686 822 2999 406 2336 645 1286 285 733 2957 357 21 2647 807 241 223 1176
700 1065 1089 2375 755 1005 2843 2247 1161 2684 1659 719 468 158 1691 1047 2355
2402 760 2671 948 455 1074 1290 2563 1227 2517 2471 2451 762 877 2166 2521 916
810 2878 1324 1549 2414 859 2151 2803 1476 1033 2427 2128 2817 2540 2180 2218
2576 924 3 2585 524 2020 2159 1912 2183 1576 1808 453 2711 2837 1177 302 1181 1827
2935 1051 2389 352 2849 2275 168 1212 2497 2962 1553 946 2324 2272 1729 2717 2175
209 1345 1963 1437 1168 964 1837 2227 1591 1630 2233 1475 1105 147 2260 1820 2381
1979 334 2464 2840 2858 484 1738 520 2574 2712 1258 349 1142 419 616 2848 990 1742
2093 2549 1937 391 2643 2977 1875 2666 2481 2683 962 2621 2378 1811 33 2531 1147
2178 834 51 86 750 1189 835 317 50 1451 2790 2010 82 2348 2474 1531 956 590 1196
2567 1890 2807 2364 2188 662 2604 1767 1947 2156 1001 1788 1438 1173 556 1050 207
2679 582 1717 585 2656 271 761 1586 1388 2944 1175 290 46 2388 543 2482 2673 1238
1791 177 721 768 1607 785 1200 1935 1117 2695 2575 842 1908 1558 322 257 1561 2012
2261 1062 1749 650 2587 1825 161 1740 2487 1201 545 2239 791 402 2486 2295 1317
42 2989 1266 1110 606 2611 2098 2416 2428 1521 767 427 1532 2741 2524 339 1933
2689 2253 376 1018 1897 1409 1380 1422 1121 638 1460 2038 1300 2797 1505 913 1465
393 587 1179 2598 1316 122 1841 2594 2572 1263 2147 507 439 153 784 1485 2690 2310
1976 2460 2347 2608 1360 1762 2026 2515 1118 364 313 1436 464 1967 1751 526 15
2976 1556 1473 854 2185 1086 1928 1686 2721 2051 1150 2875 559 2759 1366 1678
2809 1592 2311 713 1427 195 1112 112 2137 1796 1708 2328 862 1081 1646 118 1828
813 725 169 415 1039 1765 1445 1741 2914 1975 2177 1945 2066 1795 624 1341 780
2050 932 570 1472 783 634 1573 1663 199 359 1956 162 2323 1728 2186 548 811 612
1365 2553 1480 2551 2438 2454 1396 1537 2921 702 1810 1555 630 672 917 1490 2687
647 1906 1759 2387 1355 10 2888 2372 2291 1045 200 2064 2117 2610 2703 1851 1636
688 2014 1022 597 909 2558 1784 1174 163 2141 602 1668 44 2702 1766 186 891 1042
1533 2207 930 1199 1489 498 2363 1954 1021 2896 1778 657 1356 1378 2061 968 2358
377 208 938 2823 1328 1129 1645 2090 2111 1922 1665 1600 2853 1464 2577 1241 1415
690 703 1253 943 2959 649 2542 890 1123 63 304 837 1651 1391 2928 2305 642 2614
711 2280 1140 1264 523 289 1994 2693 255 2101 318 1577 673 1794 367 2429 936 467
1038 1166 1061 1934 1984 2452 459 1572 426 2550 2568 1988 2757 847 2556 1746 594
2945 401 1608 627 685 1565 1658 1680 2994 36 535 1797 2827 1540 1441 577 1662 2734

482 1225 1418 1877 2039 254 1462 1032 539 67 1919 2144 262 550 2258 599 2811 537
1111 198 477 75 404 1687 1326 2775 229 234 1182 635 1590 269 2085 298 2508 742
2852 2906 506 2024 2581 2640 720 764 1119 1895 267 2960 1911 2400 252 1700 2292
1673 111 2532 975 2040 996 608 2209 1874 950 1615 1424 37 1128 1730 2297 1579 993
2992 2492 1307 2277 989 1239 2343 2664 447 2968 2201 892 216 2300 623 1790 96 2606
2126 430 1846 332 2769 2916 127 263 2920 2729 1151 2110 2360 1936 2733 1578 2265
205 2241 1525 222 1972 1236 1059 1903 2701 292 2013 2285 1655 225 1299 2509 532 7
2708 307 2250 514 1071 2223 866 2533 1980 1107 2886 183 999 2398 2719 1261 22 2047
977 1185 492 991 1713 2172 2861 1815 1216 1983 2488 2599 1535 238 1384 528 1053
2268 2080 1155 1230 2815 2017 931 2555 237 2431 2579 2890 2912 398 1743 1309 480
2651 2694 1885 1681 39 848 2194 1657 613 2170 1426 2825 2222 476 792 876 954 2237
405 872 400 804 231 2714 2799 1940 2709 1641 515 1871 113 1084 2062 1916 1569 80
2326 2294 519 2547 1772 1285 699 776 2772 2123 1833 1727 942 2271 331 1135 203
2924 389 1260 1249 2632 511 2435 259 636 816 1958 1425 239 2956 1183 2940 2779
280 134 2314 1011 1419 2021 2642 691 2031 1861 2478 1955 676 2281 2506 2565 640
305 1792 451 502 2812 232 2910 2602 2893 1133 100 670 1599 2181 914 2493 2659 1008
562 801 969 2121 2083 2526 1526 1138 2950 2339 1876 1188 1023 2904 2168 1986 1860
155 1394 1429 1744 2795 1768 2735 336 1965 1907 2716 592 1613 2814 1068 1314 87
1361 362 2813 2189 2164 2391 2032 12 2132 970 1914 2235 178 2220 2425 1254 1102
1647 1959 2746 604 41 1293 1459 1623 2868 2654 2891 2450 2743 2824 1886 1310 1982
350 1439 2058 2408 2430 2334 527 1311 2938 2677 664 584 1842 2393 479 2417 778
1205 65 227 1939 91 963 684 1203 324 1574 2941 2086 2122 196 1226 1132 2663 1805
106 2069 2143 217 1167 512 2045 358 563 2055 2230 2645 665 2002 2167 576 1992 1099
1634 958 632 1171 1029 749 2254 321 588 833 1640 2595 1392 2146 1106 2489 806 1069
2046 2751 2145 1879 2787 953 751 1718 2286 243 2439 1284 2747 1401 1787 2552 1806
2252 2263 880 1277 1725 1704 1267 242 2713 1092 1397 2500 1421 2162 1840 1370
1486 1255 1251 2197 2972 1594 1163 1368 2629 1763 601 607 204 294 2728 1444 641
1080 929 1221 1389 736 1761 1807 73 2212 1969 1482 2307 2463 1618 185 598 2458
1546 1865 94 707 2538 622 2071 1273 541 1066 440 1500 2118 2443 2830 2053 979 1305
1194 266 1622 146 2075 1799 2016 2784 1782 2120 175 2018 637 261 669 77 2762 1683
2955 1447 944 1517 2107 1101 2436 959 2426 820 625 2974 38 281 1520 2331 596 2456

884 224 1703 1108 501 2755 2900 1786 311 1538 960 1631 366 347 1432 2990 92 827
727 2887 1754 2931 2516 1218 384 2301 1774 2546 861 1624 2761 2236 1516 68 18 197
2027 465 1320 2520 1376 249 1190 865 2796 509 2544 1562 2198 1557 2479 1661 1013
2985 2744 2133 794 648 2475 54 1279 1938 414 677 1776 2466 240 2327 1184 905 2661
1507 2635 1968 143 663 2965 454 534 2477 739 1024 1952 1667 1587 779 2412 2926
1372 853 491 344 1004 1077 855 2124 814 580 375 2088 1637 1856 431 868 2764 9 2987
97 542 671 1040 704 2975 1931 2407 1450 544 1126 569 52 1291 2219 2855 2980 2269
1469 173 218 2054 1853 1639 1270 2200 994 852 1682 1228 1209 740 2665 1601 1780
503 881 1325 2446 166 2984 418 2862 2420 824 1362 2766 1120 2908 2480 796 2742
473 1878 1399 356 2176 2571 2570 187 1998 882 1614 340 643 1588 1510 1925 2961
2981 487 2203 2792 2362 538 11 1477 1985 160 1981 450 353 1420 2930 291 2074 335
1094 2597 2791 284 568 1773 2752 3000 1699 1801 2490 2232 2922 2548 2580 1643 902
775 1016 2349 308 709 1282 1250 593 489 2631 2680 301 230 1771 1063 24 2284 2885
2530 1747 2119 2135 1268 1073 682 2863 1100 1136 1187 81 2834 1109 1977 1793 2768
1217 1688 2322 1434 1404 2369 260 857 2846 2835 16 1515 1015 724 1087 2424 1964
2315 985 247 838 1275 2023 2289 1139 2397 2794 1702 1542 2692 2600 2395 2154 982
2590 2199 1373 1354 1247 2707 1544 2819 2969 1411 1671 2589 2705 2953 1625 2240
2929 911 922 1823 2282 316 2444 668 904 109 737 2879 2157 1060 2634 138 327 1248
1966 1932 1821 581 1957 1220 125 1989 246 2818 521 2264 716 1606 1884 817 1457
1943 1036 786 513 2374 1974 2000 2816 741 2091 1929 2624 1468 726 2225 1495 2758
578 1978 2078 2821 2453 2767 2215 1696 2244 1229 793 1760 1353 151 660 1541 2377
1584 396 66 1813 444 2586 2678 2138 1858 2501 910 2627 387 2785 452 2583 976 1096
1330 2266 2541 2029 1243 1312 2179 381 23 689 172 735 2267 2105 951 315 1405 2820
1566 434 571 1315 1244 1359 2396 1014 1529 2560 2584 1611 379 2628 328 921 1498
2686 1252 330 2312 1893 60 1679 1850 2366 763 906 1054 378 2630 1685 1660 1997
1752 821 84 840 1449 2195 2498 2030 510 2006 2566 2406 320 845 698 1454 2822 2125
1148 433 1478 1035 1043 874 171 2676 1164 2625 869 2842 1461 1027 2094 1410 1204
34 1017 372 2468 2008 370 2682 782 1616 1514 2065 2320 2070 1321 2491 1280 90 934
416 1785 2404 770 2619 2871 131 1789 1950 47 1386 2612 1240 1124 1090 472 264 58
2495 2543 184 174 1597 1832 1570 633 1192 2082 342 812 2982 2449 1946 1866 799
1278 2958 986 1448 628 920 1870 894 2715 2997 2057 1213 274 341 2971 1072 69 1342

1287 1234 59 1568 575 747 310 1748 830 1431 1283 2857 412 692 2035 461 2537 64
2948 28 2739 2043 1721 1245 1143 803 825 411 2238 1894 2911 2202 2246 1145 1948
2231 819 2340 2392 1604 2351 1256 2907 466 583 1867 1012 1844 1159 987 446 2149
1502 2367 2033 652 773 2856 1949 1222 1048 2691 1294 2512 1755 678 722 567 2102
988 2763 2104 202 753 371 2737 13 1186 617 1652 925 2895 536 423 2484 2946 2308
626 530 1491 972 1067 1000 193 1076 2527 388 966 2771 1595 2998 442 2214 1689 1127
2963 758 2805 1020 1707 2722 2588 1122 717 2850 1534 445 19 705 831 836 2304 2774
2073 1452 997 2870 1583 385 1930 2287 867 210 2528 1010 1629 1904 1817 85 110 2902
180 1041 297 850 2298 2698 214 2913 1888 1838 102 1757 2832 1025 287 2136 1091
2276 2970 798 1113 1720 228 2831 2306 2390 1064 121 1471 1603 965 2130 2059 88
2697 1412 1466 697 926 844 1162 1854 1458 409 659 1114 1547 2206 1642 730 1781 5
2569 2748 1612 2724 144 399 115 1816 873 653 213 1295 1530 1363 1497 555 1319 1381
1170 1044 2278 2036 2473 795 1288 504 540 1367 547 940 1337 145 937 2174 2148 135
2894 1336 1207 856 2601 2967 1026 2534 485 1134 2483 2646 2889 1 1598 2559 1628
1304 2884 1149 1169 2173 1116 1208 2979 2437 2696 2042 383 2593 2603 2335 639
2313 829 2019 789 2113 2667 2618 1158 1596 1719 448 1374 2510 1398 1455 351 2208
1344 1527 2309 1513 2273 1593 2903 1224 2419 2445 2966 2257 984 443 2781 1731 903
1602 2641 282 1098 394 1666 1338 2470 2841 2658 928 35 338 2736 62 2668 29 2782
2242 864 600 137 2952 918 2139 674 1855 1246 2638 2901 2582 2892 1836 408 2193
1306 1800 221 2607 2899 1701 140 116 2783 1920 2626 460 1633 2169 142 26 1219 2270
1195 896 2319 889 132 1927 2127 897 248 915 2854 1519 1303 2564 2229 1723 715 1031
2933 2804 343 1756 123 1887 1165 2866 78 679 2342 2675 40 907 2150 2756 2171 517
2447 346 952 2504 2433 1237 1289 981 488 1619 397 1352 2160 295 1722 470 572 128
1493 1467 2034 1585 973 2936 828 2780 629 1104 495 1403 1440 2545 1843 1375 268
1347 1735 25 1864 2370 971 1146 2379 2883 774 129 2836 2440 1551 1697 2216 1125
1257 276 2063 386 438 1178 1803 1536 1395 1242 2467 365 500 998 105 731 818 2996
1417 2845 1492 1085 1496 2060 712 251 1364 939 2337 1798 1635 2919 2905 2441 808
436 1971 432 1669 631 27 565 1872 2302 1632 1650 2461 2079 2605 2786 494 1191 104
1413 1942 2773 1423 410 694 1019 272 373 871 992 2056 2469 1198 354 2898 718 253
746 1610 1408 70 957 2465 846 1698 2330 2674 2876 1046 681 1567 395 1944 2459 2099
589 2356 235 1504 1322 1626 2455 1156 744 1402 403 139 2048 885 2609 728 2723 2015

2613 1620 76 887 475 1202 1002 325 777 1329 2710 1400 2068 2245 326 2806 1733 2131
2140 1996 2778 1953 2025 270 1695 2108 1962 1736 2988 2442 2535 2089 2457 2251
2115 1153 901 2259 1900 2373 2005 244 2226 687 2103 420 31 2283 355 2620 1571 250
2114 2243 499 2932 2865 533 2044 2234 1009 471 2255 300 45 157 55 441 2800 2592
1292 1991 2949 1664 421 141 2829 2041 1918 2730 79 1737 2557 841 283 508 1892 1653
2380 941 2434 449 860 314 126 756 552 839 2346 2760 1428 2076 899 1869 1483 1180
2325 4 1560 1414 2499 1393 189 1543 2129 2867 392 644 182 2753 1499 8 1684 912
1057 368 212 2731 788 2256 1484 1357 961 159 1339 2655 2037 923 1506 2338 800 1115
2049 293 1348 949 2833 103 265 654 2776 2161 732 1508 1715 1905 2750 1675 1272
1503 1605 752 1390 278 2826 275 1298 2591 2418 1214 2502 2637 2405 119 1265 1528
1883 667 2421 1970 2303 429 2385 1007 696 621 206 2153 595 1750 319 2009 1711 1037
1812 2639 2067 2844 1070 1487 312 329 1351 1822 1379 2523 1829 43 2851 2224 53
2190 2011 1523 2923 117 2909 2290 1899 1377 870 1456 1358 615 2839 683 2525 101
2514 1863 2644 823 2087 1710 1880 759 56 802 967 2134 1276 2472 2410 1670 809 1103
2399 2700 815 2403 1479 1323 2321 1308 1714 1901 2522 1993 2732 89 1274 1383 1758
1302 309 1656 2669 273 1732 1371 496 2357 1839 983 2394 1144 553 1093 190 2332
2660 2361 1215 363 849 380 579 2112 974 2494 1006 1973 2423 748 2505 256 286 651
2503 2859 6 2653 797 875 1443 1430 2382 1826 2650 201 1716 2365 2808 462 1913 390
1764 1512 1941 1857 1961 2798 1923 2915 879 2847 927];

```
XTraining = zeros(650,630,3,1800);
```

```
YTrainingPoP = zeros(1800,1);
```

```
for i =1:1800
```

```
XTraining(:,:,,i) = double(imread("S (" + RN(i) + ").jpg"));
```

```
YTrainingPoP(i,1) = Percent_of_Porosity(RN(i));
```

```
end
```

```
XValidation = zeros(650,630,3,600);  
YValidationPoP = zeros(600,1);  
for i = 1801:2400  
XValidation(:,:,,i-1800) = double(imread("S (" + RN(i) + ").jpg"));  
YValidationPoP(i-1800,1) = Percent_of_Porosity(RN(i));  
  
end
```

```
XTesting = zeros(650,630,3,600);  
YTestingPoP = zeros(600,1);  
for i = 2401:3000  
XTesting(:,:,,i-2400) = double(imread("S (" + RN(i) + ").jpg"));  
YTestingPoP(i-2400,1) = Percent_of_Porosity(RN(i));  
  
end
```

```
end
```

1.8 Bayesian Regression Convolutional Neural Network (BO-RCNN)

```
clc;
```

```
clear;
```

```
close all;
```

%Load and Explore Image Data

```
ArtificialPorosityImagesDatasetPath =  
fullfile('/nfshome','store02','users','c.c1881324','Documents','MATLAB','3000  
Slices','*.jpg');
```

```
[XTraining,YTrainingPoP,XValidation,YValidationPoP,XTesting,YTestingPoP] =  
loadAPIData(ArtificialPorosityImagesDatasetPath);
```

%Define the Problem (Objective Function)

```
ObjFcn =  
makeObjFcn(XTraining,YTrainingPoP,XValidation,YValidationPoP,XTesting,YTesting  
PoP);
```

```
optimVars = [  
  
    optimizableVariable('SectionDepth',[1 3],'Type','integer')  
  
    optimizableVariable('InitialLearnRate',[0.01 0.012],'Transform','log')  
  
    optimizableVariable('Momentum',[0.8 0.98])  
  
    optimizableVariable('L2Regularization',[1e-10 1e-2],'Transform','log)];
```

%Optimize Variables

```
BayesObject = bayesopt(ObjFcn,optimVars, ...  
  
    'MaxTime',1*60*60, ...  
  
    'IsObjectiveDeterministic',false, ...  
  
    'UseParallel',false);
```

```
function ObjFcn =  
makeObjFcn(XTraining,YTrainingPoP,XValidation,YValidationPoP,XTesting,YTesting  
PoP)
```

```
ObjFcn = @ValidationErrorFunction;
```

```
function AverageValidationError = ValidationErrorFunction(optimVars)
```

```
%Specify Convolutional Neural Network Architecture
```

```
imageSize = [650 630 3];
```

```
layersPoP = [
```

```
imageInputLayer(imageSize)
```

```
convBlock(5,8, optimVars. SectionDepth)
```

```
averagePooling2dLayer(4,'Stride',4)
```

```
convBlock(5,16, optimVars. SectionDepth)
```

```
averagePooling2dLayer(4,'Stride',4)
```

```
convBlock(5,32, optimVars. SectionDepth)
```

```

averagePooling2dLayer(4,'Stride',4)

convBlock(5,64, optimVars. SectionDepth)

averagePooling2dLayer(4,'Stride',4)

convBlock(5,128, optimVars. SectionDepth)

fullyConnectedLayer(1)
regressionLayer];

miniBatchSize = 64;

validationFrequency = floor(1800/miniBatchSize);

optionsPoP = trainingOptions('sgdm', ...
    'InitialLearnRate', optimVars.InitialLearnRate, ...
    'Momentum', optimVars.Momentum, ...
    'L2Regularization', optimVars.L2Regularization, ...
    'MaxEpochs',20, ...
    'MiniBatchSize',miniBatchSize, ...
    'ValidationFrequency',validationFrequency, ...
    'ValidationData',{XValidation,YValidationPoP(:,1)}, ...
    'Shuffle','every-epoch', ...
    'Plots','training-progress');

```

%Train Network using Training Data

```
netPoP = trainNetwork(XTraining,YTrainingPoP(:,1),layersPoP,optionsPoP);
```

%Predict Responses and Compute Accuracy and Error

```
c = 1;
```

```
PredictedTrainingPoP(:,c) = predict(netPoP,XTraining);
```

```
PredictedValidationPoP(:,c) = predict(netPoP,XValidation);
```

```
PredictedTestingPoP(:,c) = predict(netPoP,XTesting);
```

```
TrainingError(:,c) = YTrainingPoP(:,1) - PredictedTrainingPoP(:,c);
```

```
AbsoluteTrainingError(:,c) = abs(TrainingError(:,c));
```

```
AverageTrainingError(:,c) = sum(AbsoluteTrainingError(:,c))/1800;
```

```
numCorrectTrainingPoP(:,c) = sum(abs(TrainingError(:,c)) < 0.05);
```

```
TrainingAccuracyPoP(:,c) = (numCorrectTrainingPoP(:,c))/1800*100;
```

```
ValidationError(:,c) = YValidationPoP(:,1) - PredictedValidationPoP(:,c);
```

```
AbsoluteValidationError(:,c) = abs(ValidationError(:,c));
```

```
AverageValidationError(:,c) = sum(AbsoluteValidationError(:,c))/600;
```

```
numCorrectValidationPoP(:,c) = sum(abs(ValidationError(:,c)) < 0.05);
```

```
ValidationAccuracyPoP(:,c) = (numCorrectValidationPoP(:,c))/600*100;
```

```
TestingError(:,c) = YTestingPoP(:,1) - PredictedTestingPoP(:,c);
```

```
AbsoluteTestingError(:,c) = abs(TestingError(:,c));
```



```

AverageTestingError(:,c) = sum(AbsoluteTestingError(:,c))/600;
numCorrectTestingPoP(:,c) = sum(abs(TestingError(:,c))) < 0.05);
TestingAccuracyPoP(:,c) = (numCorrectTestingPoP(:,c))/600)*100;

```

```

    end

```

```

end

```

```

function layers = convBlock(FilterSize,NumberOfFilters,SectionDepth)

```

```

layers = [

```

```

    convolution2dLayer(FilterSize,NumberOfFilters,'Padding','same')

```

```

    batchNormalizationLayer

```

```

    reluLayer];

```

```

layers = repmat(layers,SectionDepth,1);

```

```

end

```

```

function [XTraining,YTrainingPoP,XValidation,YValidationPoP,XTesting,YTestingPoP]
= loadAPIData(ArtificialPorosityImagesDatasetPath)

```

```

location = dir(ArtificialPorosityImagesDatasetPath);

```

```

Percent_of_Porosity = Actual_Percent_of_Porosity;

```

RN = [1834 919 935 220 2897 1891 32 675 2106 2725 424 1690 900 1548 2596 2947 505
1802 2706 149 2648 2793 165 2726 1859 1769 2022 878 564 2109 646 566 2376 2163
518 2496 898 522 1049 2192 2519 2221 620 1211 2152 2191 2617 618 2877 769 757 1692
164 2155 554 277 947 1509 98 2554 1382 2869 2633 413 2801 2934 2262 1327 2770 1575
288 1926 2749 2573 695 2100 1281 1052 2511 754 279 57 832 656 1446 2371 2092 2288
1097 1545 1474 2649 2293 1232 787 2986 745 1706 1868 2685 2539 124 130 2184 2838
893 2409 1259 348 1095 481 714 2359 1082 1387 605 729 2211 1343 1030 693 120 1621
245 1609 99 1990 2413 188 1210 558 369 1809 2954 1917 374 1463 1083 2993 2279 2052
574 1873 1318 1157 863 1921 2872 1835 1501 71 603 2622 150 181 1804 560 1783 2001
1262 1331 1003 1881 858 525 2411 191 2623 2704 1141 2873 2 1709 2004 93 551 136
1511 2777 1819 2333 658 2943 2720 1845 1160 1676 1849 2882 1889 425 1848 72 2738
2828 154 2448 1775 2007 2810 1672 1987 1335 1350 428 1818 1296 2615 2142 170 211
176 772 133 1524 1028 1131 1638 1554 2213 895 546 1078 1058 417 156 701 1340 226
463 306 1712 2951 1862 1231 2991 851 1522 2718 1406 2864 1726 1745 1882 1915 955
734 1488 1847 888 2529 2353 1269 2204 192 1297 1649 2165 1137 2802 2182 743 333
2072 980 299 1563 2942 2248 1753 2964 1999 610 2485 1770 2918 708 591 557 1694
883 723 1193 2681 1734 2158 1453 1960 1552 2318 95 1677 1334 549 478 2789 17 49
2745 1433 1705 1223 680 886 490 14 1724 1654 1627 2917 516 194 2978 790 2616 2299
83 2518 619 219 1197 337 1130 1518 1693 345 2657 1995 2462 407 437 2422 2274 1617
843 614 1055 2097 114 2386 1075 1233 2476 1385 107 20 74 710 1559 2084 1924 2329
1301 2662 382 2354 2205 2095 2740 978 1034 1896 469 258 1154 766 2081 908 1079
1349 1369 2880 2699 30 1739 2317 1644 2561 1088 1470 1564 2670 2003 1271 1902 493
1416 152 1235 2077 1580 1172 2296 1589 1779 1332 48 2860 2196 765 323 2937 1206
738 1442 303 666 1777 1852 2096 2765 361 1346 2562 573 2116 148 167 1831 826 61
233 611 2341 995 474 1152 296 2384 456 781 2636 2316 1814 2028 2507 2688 1898 2432
1407 458 1909 236 609 586 2727 2513 2352 486 1910 2228 2652 1582 108 2217 497 2345
1481 2983 1494 1333 2350 2344 655 2939 771 435 2536 529 483 2672 457 2925 2383
215 360 2788 1830 1539 2874 805 561 2995 1824 1056 1313 945 2578 2415 1648 661
1674 2187 2973 2249 2927 1581 933 2210 706 2754 1550 2368 179 2401 1435 2881 1951
422 531 686 822 2999 406 2336 645 1286 285 733 2957 357 21 2647 807 241 223 1176
700 1065 1089 2375 755 1005 2843 2247 1161 2684 1659 719 468 158 1691 1047 2355

2402 760 2671 948 455 1074 1290 2563 1227 2517 2471 2451 762 877 2166 2521 916
810 2878 1324 1549 2414 859 2151 2803 1476 1033 2427 2128 2817 2540 2180 2218
2576 924 3 2585 524 2020 2159 1912 2183 1576 1808 453 2711 2837 1177 302 1181 1827
2935 1051 2389 352 2849 2275 168 1212 2497 2962 1553 946 2324 2272 1729 2717 2175
209 1345 1963 1437 1168 964 1837 2227 1591 1630 2233 1475 1105 147 2260 1820 2381
1979 334 2464 2840 2858 484 1738 520 2574 2712 1258 349 1142 419 616 2848 990 1742
2093 2549 1937 391 2643 2977 1875 2666 2481 2683 962 2621 2378 1811 33 2531 1147
2178 834 51 86 750 1189 835 317 50 1451 2790 2010 82 2348 2474 1531 956 590 1196
2567 1890 2807 2364 2188 662 2604 1767 1947 2156 1001 1788 1438 1173 556 1050 207
2679 582 1717 585 2656 271 761 1586 1388 2944 1175 290 46 2388 543 2482 2673 1238
1791 177 721 768 1607 785 1200 1935 1117 2695 2575 842 1908 1558 322 257 1561 2012
2261 1062 1749 650 2587 1825 161 1740 2487 1201 545 2239 791 402 2486 2295 1317
42 2989 1266 1110 606 2611 2098 2416 2428 1521 767 427 1532 2741 2524 339 1933
2689 2253 376 1018 1897 1409 1380 1422 1121 638 1460 2038 1300 2797 1505 913 1465
393 587 1179 2598 1316 122 1841 2594 2572 1263 2147 507 439 153 784 1485 2690 2310
1976 2460 2347 2608 1360 1762 2026 2515 1118 364 313 1436 464 1967 1751 526 15
2976 1556 1473 854 2185 1086 1928 1686 2721 2051 1150 2875 559 2759 1366 1678
2809 1592 2311 713 1427 195 1112 112 2137 1796 1708 2328 862 1081 1646 118 1828
813 725 169 415 1039 1765 1445 1741 2914 1975 2177 1945 2066 1795 624 1341 780
2050 932 570 1472 783 634 1573 1663 199 359 1956 162 2323 1728 2186 548 811 612
1365 2553 1480 2551 2438 2454 1396 1537 2921 702 1810 1555 630 672 917 1490 2687
647 1906 1759 2387 1355 10 2888 2372 2291 1045 200 2064 2117 2610 2703 1851 1636
688 2014 1022 597 909 2558 1784 1174 163 2141 602 1668 44 2702 1766 186 891 1042
1533 2207 930 1199 1489 498 2363 1954 1021 2896 1778 657 1356 1378 2061 968 2358
377 208 938 2823 1328 1129 1645 2090 2111 1922 1665 1600 2853 1464 2577 1241 1415
690 703 1253 943 2959 649 2542 890 1123 63 304 837 1651 1391 2928 2305 642 2614
711 2280 1140 1264 523 289 1994 2693 255 2101 318 1577 673 1794 367 2429 936 467
1038 1166 1061 1934 1984 2452 459 1572 426 2550 2568 1988 2757 847 2556 1746 594
2945 401 1608 627 685 1565 1658 1680 2994 36 535 1797 2827 1540 1441 577 1662 2734
482 1225 1418 1877 2039 254 1462 1032 539 67 1919 2144 262 550 2258 599 2811 537
1111 198 477 75 404 1687 1326 2775 229 234 1182 635 1590 269 2085 298 2508 742

2852 2906 506 2024 2581 2640 720 764 1119 1895 267 2960 1911 2400 252 1700 2292
1673 111 2532 975 2040 996 608 2209 1874 950 1615 1424 37 1128 1730 2297 1579 993
2992 2492 1307 2277 989 1239 2343 2664 447 2968 2201 892 216 2300 623 1790 96 2606
2126 430 1846 332 2769 2916 127 263 2920 2729 1151 2110 2360 1936 2733 1578 2265
205 2241 1525 222 1972 1236 1059 1903 2701 292 2013 2285 1655 225 1299 2509 532 7
2708 307 2250 514 1071 2223 866 2533 1980 1107 2886 183 999 2398 2719 1261 22 2047
977 1185 492 991 1713 2172 2861 1815 1216 1983 2488 2599 1535 238 1384 528 1053
2268 2080 1155 1230 2815 2017 931 2555 237 2431 2579 2890 2912 398 1743 1309 480
2651 2694 1885 1681 39 848 2194 1657 613 2170 1426 2825 2222 476 792 876 954 2237
405 872 400 804 231 2714 2799 1940 2709 1641 515 1871 113 1084 2062 1916 1569 80
2326 2294 519 2547 1772 1285 699 776 2772 2123 1833 1727 942 2271 331 1135 203
2924 389 1260 1249 2632 511 2435 259 636 816 1958 1425 239 2956 1183 2940 2779
280 134 2314 1011 1419 2021 2642 691 2031 1861 2478 1955 676 2281 2506 2565 640
305 1792 451 502 2812 232 2910 2602 2893 1133 100 670 1599 2181 914 2493 2659 1008
562 801 969 2121 2083 2526 1526 1138 2950 2339 1876 1188 1023 2904 2168 1986 1860
155 1394 1429 1744 2795 1768 2735 336 1965 1907 2716 592 1613 2814 1068 1314 87
1361 362 2813 2189 2164 2391 2032 12 2132 970 1914 2235 178 2220 2425 1254 1102
1647 1959 2746 604 41 1293 1459 1623 2868 2654 2891 2450 2743 2824 1886 1310 1982
350 1439 2058 2408 2430 2334 527 1311 2938 2677 664 584 1842 2393 479 2417 778
1205 65 227 1939 91 963 684 1203 324 1574 2941 2086 2122 196 1226 1132 2663 1805
106 2069 2143 217 1167 512 2045 358 563 2055 2230 2645 665 2002 2167 576 1992 1099
1634 958 632 1171 1029 749 2254 321 588 833 1640 2595 1392 2146 1106 2489 806 1069
2046 2751 2145 1879 2787 953 751 1718 2286 243 2439 1284 2747 1401 1787 2552 1806
2252 2263 880 1277 1725 1704 1267 242 2713 1092 1397 2500 1421 2162 1840 1370
1486 1255 1251 2197 2972 1594 1163 1368 2629 1763 601 607 204 294 2728 1444 641
1080 929 1221 1389 736 1761 1807 73 2212 1969 1482 2307 2463 1618 185 598 2458
1546 1865 94 707 2538 622 2071 1273 541 1066 440 1500 2118 2443 2830 2053 979 1305
1194 266 1622 146 2075 1799 2016 2784 1782 2120 175 2018 637 261 669 77 2762 1683
2955 1447 944 1517 2107 1101 2436 959 2426 820 625 2974 38 281 1520 2331 596 2456
884 224 1703 1108 501 2755 2900 1786 311 1538 960 1631 366 347 1432 2990 92 827
727 2887 1754 2931 2516 1218 384 2301 1774 2546 861 1624 2761 2236 1516 68 18 197

2027 465 1320 2520 1376 249 1190 865 2796 509 2544 1562 2198 1557 2479 1661 1013
2985 2744 2133 794 648 2475 54 1279 1938 414 677 1776 2466 240 2327 1184 905 2661
1507 2635 1968 143 663 2965 454 534 2477 739 1024 1952 1667 1587 779 2412 2926
1372 853 491 344 1004 1077 855 2124 814 580 375 2088 1637 1856 431 868 2764 9 2987
97 542 671 1040 704 2975 1931 2407 1450 544 1126 569 52 1291 2219 2855 2980 2269
1469 173 218 2054 1853 1639 1270 2200 994 852 1682 1228 1209 740 2665 1601 1780
503 881 1325 2446 166 2984 418 2862 2420 824 1362 2766 1120 2908 2480 796 2742
473 1878 1399 356 2176 2571 2570 187 1998 882 1614 340 643 1588 1510 1925 2961
2981 487 2203 2792 2362 538 11 1477 1985 160 1981 450 353 1420 2930 291 2074 335
1094 2597 2791 284 568 1773 2752 3000 1699 1801 2490 2232 2922 2548 2580 1643 902
775 1016 2349 308 709 1282 1250 593 489 2631 2680 301 230 1771 1063 24 2284 2885
2530 1747 2119 2135 1268 1073 682 2863 1100 1136 1187 81 2834 1109 1977 1793 2768
1217 1688 2322 1434 1404 2369 260 857 2846 2835 16 1515 1015 724 1087 2424 1964
2315 985 247 838 1275 2023 2289 1139 2397 2794 1702 1542 2692 2600 2395 2154 982
2590 2199 1373 1354 1247 2707 1544 2819 2969 1411 1671 2589 2705 2953 1625 2240
2929 911 922 1823 2282 316 2444 668 904 109 737 2879 2157 1060 2634 138 327 1248
1966 1932 1821 581 1957 1220 125 1989 246 2818 521 2264 716 1606 1884 817 1457
1943 1036 786 513 2374 1974 2000 2816 741 2091 1929 2624 1468 726 2225 1495 2758
578 1978 2078 2821 2453 2767 2215 1696 2244 1229 793 1760 1353 151 660 1541 2377
1584 396 66 1813 444 2586 2678 2138 1858 2501 910 2627 387 2785 452 2583 976 1096
1330 2266 2541 2029 1243 1312 2179 381 23 689 172 735 2267 2105 951 315 1405 2820
1566 434 571 1315 1244 1359 2396 1014 1529 2560 2584 1611 379 2628 328 921 1498
2686 1252 330 2312 1893 60 1679 1850 2366 763 906 1054 378 2630 1685 1660 1997
1752 821 84 840 1449 2195 2498 2030 510 2006 2566 2406 320 845 698 1454 2822 2125
1148 433 1478 1035 1043 874 171 2676 1164 2625 869 2842 1461 1027 2094 1410 1204
34 1017 372 2468 2008 370 2682 782 1616 1514 2065 2320 2070 1321 2491 1280 90 934
416 1785 2404 770 2619 2871 131 1789 1950 47 1386 2612 1240 1124 1090 472 264 58
2495 2543 184 174 1597 1832 1570 633 1192 2082 342 812 2982 2449 1946 1866 799
1278 2958 986 1448 628 920 1870 894 2715 2997 2057 1213 274 341 2971 1072 69 1342
1287 1234 59 1568 575 747 310 1748 830 1431 1283 2857 412 692 2035 461 2537 64
2948 28 2739 2043 1721 1245 1143 803 825 411 2238 1894 2911 2202 2246 1145 1948

2231 819 2340 2392 1604 2351 1256 2907 466 583 1867 1012 1844 1159 987 446 2149
1502 2367 2033 652 773 2856 1949 1222 1048 2691 1294 2512 1755 678 722 567 2102
988 2763 2104 202 753 371 2737 13 1186 617 1652 925 2895 536 423 2484 2946 2308
626 530 1491 972 1067 1000 193 1076 2527 388 966 2771 1595 2998 442 2214 1689 1127
2963 758 2805 1020 1707 2722 2588 1122 717 2850 1534 445 19 705 831 836 2304 2774
2073 1452 997 2870 1583 385 1930 2287 867 210 2528 1010 1629 1904 1817 85 110 2902
180 1041 297 850 2298 2698 214 2913 1888 1838 102 1757 2832 1025 287 2136 1091
2276 2970 798 1113 1720 228 2831 2306 2390 1064 121 1471 1603 965 2130 2059 88
2697 1412 1466 697 926 844 1162 1854 1458 409 659 1114 1547 2206 1642 730 1781 5
2569 2748 1612 2724 144 399 115 1816 873 653 213 1295 1530 1363 1497 555 1319 1381
1170 1044 2278 2036 2473 795 1288 504 540 1367 547 940 1337 145 937 2174 2148 135
2894 1336 1207 856 2601 2967 1026 2534 485 1134 2483 2646 2889 1 1598 2559 1628
1304 2884 1149 1169 2173 1116 1208 2979 2437 2696 2042 383 2593 2603 2335 639
2313 829 2019 789 2113 2667 2618 1158 1596 1719 448 1374 2510 1398 1455 351 2208
1344 1527 2309 1513 2273 1593 2903 1224 2419 2445 2966 2257 984 443 2781 1731 903
1602 2641 282 1098 394 1666 1338 2470 2841 2658 928 35 338 2736 62 2668 29 2782
2242 864 600 137 2952 918 2139 674 1855 1246 2638 2901 2582 2892 1836 408 2193
1306 1800 221 2607 2899 1701 140 116 2783 1920 2626 460 1633 2169 142 26 1219 2270
1195 896 2319 889 132 1927 2127 897 248 915 2854 1519 1303 2564 2229 1723 715 1031
2933 2804 343 1756 123 1887 1165 2866 78 679 2342 2675 40 907 2150 2756 2171 517
2447 346 952 2504 2433 1237 1289 981 488 1619 397 1352 2160 295 1722 470 572 128
1493 1467 2034 1585 973 2936 828 2780 629 1104 495 1403 1440 2545 1843 1375 268
1347 1735 25 1864 2370 971 1146 2379 2883 774 129 2836 2440 1551 1697 2216 1125
1257 276 2063 386 438 1178 1803 1536 1395 1242 2467 365 500 998 105 731 818 2996
1417 2845 1492 1085 1496 2060 712 251 1364 939 2337 1798 1635 2919 2905 2441 808
436 1971 432 1669 631 27 565 1872 2302 1632 1650 2461 2079 2605 2786 494 1191 104
1413 1942 2773 1423 410 694 1019 272 373 871 992 2056 2469 1198 354 2898 718 253
746 1610 1408 70 957 2465 846 1698 2330 2674 2876 1046 681 1567 395 1944 2459 2099
589 2356 235 1504 1322 1626 2455 1156 744 1402 403 139 2048 885 2609 728 2723 2015
2613 1620 76 887 475 1202 1002 325 777 1329 2710 1400 2068 2245 326 2806 1733 2131
2140 1996 2778 1953 2025 270 1695 2108 1962 1736 2988 2442 2535 2089 2457 2251

2115 1153 901 2259 1900 2373 2005 244 2226 687 2103 420 31 2283 355 2620 1571 250
2114 2243 499 2932 2865 533 2044 2234 1009 471 2255 300 45 157 55 441 2800 2592
1292 1991 2949 1664 421 141 2829 2041 1918 2730 79 1737 2557 841 283 508 1892 1653
2380 941 2434 449 860 314 126 756 552 839 2346 2760 1428 2076 899 1869 1483 1180
2325 4 1560 1414 2499 1393 189 1543 2129 2867 392 644 182 2753 1499 8 1684 912
1057 368 212 2731 788 2256 1484 1357 961 159 1339 2655 2037 923 1506 2338 800 1115
2049 293 1348 949 2833 103 265 654 2776 2161 732 1508 1715 1905 2750 1675 1272
1503 1605 752 1390 278 2826 275 1298 2591 2418 1214 2502 2637 2405 119 1265 1528
1883 667 2421 1970 2303 429 2385 1007 696 621 206 2153 595 1750 319 2009 1711 1037
1812 2639 2067 2844 1070 1487 312 329 1351 1822 1379 2523 1829 43 2851 2224 53
2190 2011 1523 2923 117 2909 2290 1899 1377 870 1456 1358 615 2839 683 2525 101
2514 1863 2644 823 2087 1710 1880 759 56 802 967 2134 1276 2472 2410 1670 809 1103
2399 2700 815 2403 1479 1323 2321 1308 1714 1901 2522 1993 2732 89 1274 1383 1758
1302 309 1656 2669 273 1732 1371 496 2357 1839 983 2394 1144 553 1093 190 2332
2660 2361 1215 363 849 380 579 2112 974 2494 1006 1973 2423 748 2505 256 286 651
2503 2859 6 2653 797 875 1443 1430 2382 1826 2650 201 1716 2365 2808 462 1913 390
1764 1512 1941 1857 1961 2798 1923 2915 879 2847 927];

```
XTraining = zeros(650,630,3,1800);
```

```
YTrainingPoP = zeros(1800,1);
```

```
for i =1:1800
```

```
XTraining(:,:,i) = double(imread("S (" + RN(i) + ").jpg"));
```

```
YTrainingPoP(i,1) = Percent_of_Porosity(RN(i));
```

```
end
```

```
XValidation = zeros(650,630,3,600);  
YValidationPoP = zeros(600,1);  
for i = 1801:2400  
XValidation(:,:,,i-1800) = double(imread("S (" + RN(i) + ").jpg"));  
YValidationPoP(i-1800,1) = Percent_of_Porosity(RN(i));  
  
end
```

```
XTesting = zeros(650,630,3,600);  
YTestingPoP = zeros(600,1);  
for i = 2401:3000  
XTesting(:,:,,i-2400) = double(imread("S (" + RN(i) + ").jpg"));  
YTestingPoP(i-2400,1) = Percent_of_Porosity(RN(i));  
  
end
```

```
end
```

1.9 Bees Bayesian Regression Convolutional Neural Network (BA-BO-RCNN)

```
clc;
```

```
clear;
```

```
close all;
```


%Load and Explore Image Data

```
ArtificialPorosityImagesDatasetPath =  
fullfile('/nfshome','store02','users','c.c1881324','Documents','MATLAB','3000  
Slices','*.jpg');
```

```
[XTraining,YTrainingPoP,XValidation,YValidationPoP,XTesting,YTestingPoP] =  
loadAPIData(ArtificialPorosityImagesDatasetPath);
```

%Define the Problem (Objective Function)

```
ObjFcn =  
makeObjFcn(XTraining,YTrainingPoP,XValidation,YValidationPoP,XTesting,YTesting  
PoP);
```

```
nVar=6; %Number of Weight Learning Rate Factors
```

```
VarSize=[1 nVar]; %Matrix Size for Factors
```

```
VarMin=0.9; %Lower Bound for Factors
```

```
VarMax=1.1; %Upper Bound for Factors
```

%Define Bees Algorithm Parameters

```
MaxIt=1; %Maximum Number of Iterations
```

```
nScoutBee=4; %Scout Bees
```

```
nSelectedSite=round(0.5*nScoutBee); %Selected Sites
```

```
nEliteSite=1; %Selected Elite Sites
```

```
nSelectedSiteBee=round(0.5*nScoutBee); %Recruited Bees for Selected Sites
```

```
nEliteSiteBee=2*nSelectedSiteBee; %Recruited Bees for Elite Sites
```

```
r=0.1*(VarMax-VarMin); %Neighbourhood Radius
```

```
rdamp=0.95; %Neighbourhood Radius Damp Rate
```

%Initialize Empty Bee Structure

```
empty_bee.Position=[];
```

```
empty_bee.Error=[];
```

%Initialize Bees Array

```
bee= repmat(empty_bee,nScoutBee,1);
```

%Create New Solutions

```
for i=1:nScoutBee
```

```
    bee(i).Position=unifrnd(VarMin,VarMax,VarSize);
```

```
    bee(i).Error=ObjFcn(bee(i).Position);
```

```
end
```

%Sort the Solution

```
[~, SortOrder]=sort([bee.Error]);
```

```
bee=bee(SortOrder);
```

%Update Best Solution Ever Found

```
BestSol=bee(1);
```

%Create Array to Hold Best Error Values

```
LowestError=zeros(MaxIt,1);
```

%Create Bees Algorithm Main Loop

```
for it=1:MaxIt
```

%Elite Sites

```
for i=1:nEliteSite
```

```
    bestnewbee.Error=inf;
```

```
    for j=1:nEliteSiteBee
```

```
        newbee.Position=PerformBeeDance(bee(i).Position,r);
```

```
        newbee.Error=ObjFcn(newbee.Position);
```

```
        if newbee.Error<bestnewbee.Error
```

```
            bestnewbee=newbee;
```

```
        end
```

```
    end
```

```
    if bestnewbee.Error<bee(i).Error
```

```
        bee(i)=bestnewbee;
```

```

    end

end

%Selected Non-Elite Sites

for i=nEliteSite+1:nSelectedSite

    bestnewbee.Error=inf;

    for j=1:nSelectedSiteBee

        newbee.Position=PerformBeeDance(bee(i).Position,r);

        newbee.Error=ObjFcn(newbee.Position);

        if newbee.Error<bestnewbee.Error

            bestnewbee=newbee;

        end

    end

end

    if bestnewbee.Error<bee(i).Error

        bee(i)=bestnewbee;

    end

end

%Non-Selected Sites

for i=nSelectedSite+1:nScoutBee

    bee(i).Position=unifrnd(VarMin,VarMax,VarSize);

```

```

    bee(i).Error=ObjFcn(bee(i).Position);
end

%Sort the Solution

[~, SortOrder]=sort([bee.Error]);

bee=bee(SortOrder);

%Update Best Solution Ever Found

BestSol=bee(1);

%Store Best Error Ever Found

LowestError(it)=BestSol.Error;

OptimalWeightLearningRateFactor=BestSol.Position;

%Display Iteration Information

disp(['Iteration ' num2str(it) ': Lowest Error = ' num2str(LowestError(it))]);

%Define Damp Neighborhood Radius

r=r*rdamp;

end

%Display the Results

figure;

```

%Make a Plot for Lowest Error

```
semilogy(LowestError,'LineWidth',2);
```

```
xlabel('Iteration');
```

```
ylabel('Lowest Error');
```

```
function ObjFcn =
```

```
makeObjFcn(XTraining,YTrainingPoP,XValidation,YValidationPoP,XTesting,YTesting  
PoP)
```

```
ObjFcn = @ValidationErrorFunction;
```

```
function AverageValidationError =  
ValidationErrorFunction(OptimalWeightLearningRateFactor)
```

%Specify Convolutional Neural Network Architecture

```
imageSize = [650 630 3];
```

```
layersPoP = [
```

```
imageInputLayer(imageSize)
```

```
convBlock(5,8,3,OptimalWeightLearningRateFactor(1,1))
```

```
averagePooling2dLayer(4,'Stride',4)
```

```
convBlock(5,16,3,OptimalWeightLearningRateFactor(1,2))
```

```

averagePooling2dLayer(4,'Stride',4)

convBlock(5,32,3,OptimalWeightLearningRateFactor(1,3))

averagePooling2dLayer(4,'Stride',4)

convBlock(5,64,3,OptimalWeightLearningRateFactor(1,4))

averagePooling2dLayer(4,'Stride',4)

convBlock(5,128,3,OptimalWeightLearningRateFactor(1,5))

fullyConnectedLayer(1,'WeightLearnRateFactor',OptimalWeightLearningRateFactor(1,6)
)
    regressionLayer];

miniBatchSize = 64;
validationFrequency = floor(1800/miniBatchSize);
optionsPoP = trainingOptions('sgdm', ...
    'InitialLearnRate', 0.011964, ...
    'Momentum', 0.90267, ...
    'L2Regularization', 4.7995e-07, ...

```

```

'MaxEpochs',20, ...
'MiniBatchSize',miniBatchSize, ...
'ValidationFrequency',validationFrequency, ...
'ValidationData',{XValidation,YValidationPoP(:,1)}, ...
'Shuffle','every-epoch', ...
'Plots','training-progress');

```

%Train Network using Training Data

```
netPoP = trainNetwork(XTraining,YTrainingPoP(:,1),layersPoP,optionsPoP);
```

%Predict Responses and Compute Accuracy and Error

```
c = 1;
```

```
PredictedTrainingPoP(:,c) = predict(netPoP,XTraining);
```

```
PredictedValidationPoP(:,c) = predict(netPoP,XValidation);
```

```
PredictedTestingPoP(:,c) = predict(netPoP,XTesting);
```

```
TrainingError(:,c) = YTrainingPoP(:,1) - PredictedTrainingPoP(:,c);
```

```
AbsoluteTrainingError(:,c) = abs(TrainingError(:,c));
```

```
AverageTrainingError(:,c) = sum(AbsoluteTrainingError(:,c))/1800;
```

```
numCorrectTrainingPoP(:,c) = sum(abs(TrainingError(:,c)) < 0.05);
```

```
TrainingAccuracyPoP(:,c) = (numCorrectTrainingPoP(:,c))/1800*100;
```



```

ValidationError(:,c) = YValidationPoP(:,1) - PredictedValidationPoP(:,c);
AbsoluteValidationError(:,c) = abs(ValidationError(:,c));
AverageValidationError(:,c) = sum(AbsoluteValidationError(:,c))/600;
numCorrectValidationPoP(:,c) = sum(abs(ValidationError(:,c)) < 0.05);
ValidationAccuracyPoP(:,c) = (numCorrectValidationPoP(:,c))/600*100;

```

```

TestingError(:,c) = YTestingPoP(:,1) - PredictedTestingPoP(:,c);
AbsoluteTestingError(:,c) = abs(TestingError(:,c));
AverageTestingError(:,c) = sum(AbsoluteTestingError(:,c))/600;
numCorrectTestingPoP(:,c) = sum(abs(TestingError(:,c)) < 0.05);
TestingAccuracyPoP(:,c) = (numCorrectTestingPoP(:,c))/600*100;

```

```

    end

```

```

end

```

%Create Bees Dance Function

```

function y=PerformBeeDance(ValidationErrorFunction,r)

```

```

    nVar=numel(ValidationErrorFunction);
    k=randi([1 nVar]);
    y=ValidationErrorFunction;
    y(k)=ValidationErrorFunction(k)+unifrnd(-r,r);

```

```

end

```

```

function layers =
convBlock(FilterSize,NumberofFilters,SectionDepth,OptimalWeightLearningRateFactor)

layers = [

convolution2dLayer(FilterSize,NumberofFilters,'Padding','same','WeightLearnRateFactor'
,OptimalWeightLearningRateFactor)

    batchNormalizationLayer

    reluLayer];

layers = repmat(layers,SectionDepth,1);

end

```

```

function [XTraining,YTrainingPoP,XValidation,YValidationPoP,XTesting,YTestingPoP]
= loadAPIData(ArtificialPorosityImagesDatasetPath)

```

```

location = dir(ArtificialPorosityImagesDatasetPath);

```

```

Percent_of_Porosity = Actual_Percent_of_Porosity;

```

```

RN = [1834 919 935 220 2897 1891 32 675 2106 2725 424 1690 900 1548 2596 2947 505
1802 2706 149 2648 2793 165 2726 1859 1769 2022 878 564 2109 646 566 2376 2163
518 2496 898 522 1049 2192 2519 2221 620 1211 2152 2191 2617 618 2877 769 757 1692
164 2155 554 277 947 1509 98 2554 1382 2869 2633 413 2801 2934 2262 1327 2770 1575
288 1926 2749 2573 695 2100 1281 1052 2511 754 279 57 832 656 1446 2371 2092 2288
1097 1545 1474 2649 2293 1232 787 2986 745 1706 1868 2685 2539 124 130 2184 2838
893 2409 1259 348 1095 481 714 2359 1082 1387 605 729 2211 1343 1030 693 120 1621
245 1609 99 1990 2413 188 1210 558 369 1809 2954 1917 374 1463 1083 2993 2279 2052

```

574 1873 1318 1157 863 1921 2872 1835 1501 71 603 2622 150 181 1804 560 1783 2001
1262 1331 1003 1881 858 525 2411 191 2623 2704 1141 2873 2 1709 2004 93 551 136
1511 2777 1819 2333 658 2943 2720 1845 1160 1676 1849 2882 1889 425 1848 72 2738
2828 154 2448 1775 2007 2810 1672 1987 1335 1350 428 1818 1296 2615 2142 170 211
176 772 133 1524 1028 1131 1638 1554 2213 895 546 1078 1058 417 156 701 1340 226
463 306 1712 2951 1862 1231 2991 851 1522 2718 1406 2864 1726 1745 1882 1915 955
734 1488 1847 888 2529 2353 1269 2204 192 1297 1649 2165 1137 2802 2182 743 333
2072 980 299 1563 2942 2248 1753 2964 1999 610 2485 1770 2918 708 591 557 1694
883 723 1193 2681 1734 2158 1453 1960 1552 2318 95 1677 1334 549 478 2789 17 49
2745 1433 1705 1223 680 886 490 14 1724 1654 1627 2917 516 194 2978 790 2616 2299
83 2518 619 219 1197 337 1130 1518 1693 345 2657 1995 2462 407 437 2422 2274 1617
843 614 1055 2097 114 2386 1075 1233 2476 1385 107 20 74 710 1559 2084 1924 2329
1301 2662 382 2354 2205 2095 2740 978 1034 1896 469 258 1154 766 2081 908 1079
1349 1369 2880 2699 30 1739 2317 1644 2561 1088 1470 1564 2670 2003 1271 1902 493
1416 152 1235 2077 1580 1172 2296 1589 1779 1332 48 2860 2196 765 323 2937 1206
738 1442 303 666 1777 1852 2096 2765 361 1346 2562 573 2116 148 167 1831 826 61
233 611 2341 995 474 1152 296 2384 456 781 2636 2316 1814 2028 2507 2688 1898 2432
1407 458 1909 236 609 586 2727 2513 2352 486 1910 2228 2652 1582 108 2217 497 2345
1481 2983 1494 1333 2350 2344 655 2939 771 435 2536 529 483 2672 457 2925 2383
215 360 2788 1830 1539 2874 805 561 2995 1824 1056 1313 945 2578 2415 1648 661
1674 2187 2973 2249 2927 1581 933 2210 706 2754 1550 2368 179 2401 1435 2881 1951
422 531 686 822 2999 406 2336 645 1286 285 733 2957 357 21 2647 807 241 223 1176
700 1065 1089 2375 755 1005 2843 2247 1161 2684 1659 719 468 158 1691 1047 2355
2402 760 2671 948 455 1074 1290 2563 1227 2517 2471 2451 762 877 2166 2521 916
810 2878 1324 1549 2414 859 2151 2803 1476 1033 2427 2128 2817 2540 2180 2218
2576 924 3 2585 524 2020 2159 1912 2183 1576 1808 453 2711 2837 1177 302 1181 1827
2935 1051 2389 352 2849 2275 168 1212 2497 2962 1553 946 2324 2272 1729 2717 2175
209 1345 1963 1437 1168 964 1837 2227 1591 1630 2233 1475 1105 147 2260 1820 2381
1979 334 2464 2840 2858 484 1738 520 2574 2712 1258 349 1142 419 616 2848 990 1742
2093 2549 1937 391 2643 2977 1875 2666 2481 2683 962 2621 2378 1811 33 2531 1147
2178 834 51 86 750 1189 835 317 50 1451 2790 2010 82 2348 2474 1531 956 590 1196

2567 1890 2807 2364 2188 662 2604 1767 1947 2156 1001 1788 1438 1173 556 1050 207
2679 582 1717 585 2656 271 761 1586 1388 2944 1175 290 46 2388 543 2482 2673 1238
1791 177 721 768 1607 785 1200 1935 1117 2695 2575 842 1908 1558 322 257 1561 2012
2261 1062 1749 650 2587 1825 161 1740 2487 1201 545 2239 791 402 2486 2295 1317
42 2989 1266 1110 606 2611 2098 2416 2428 1521 767 427 1532 2741 2524 339 1933
2689 2253 376 1018 1897 1409 1380 1422 1121 638 1460 2038 1300 2797 1505 913 1465
393 587 1179 2598 1316 122 1841 2594 2572 1263 2147 507 439 153 784 1485 2690 2310
1976 2460 2347 2608 1360 1762 2026 2515 1118 364 313 1436 464 1967 1751 526 15
2976 1556 1473 854 2185 1086 1928 1686 2721 2051 1150 2875 559 2759 1366 1678
2809 1592 2311 713 1427 195 1112 112 2137 1796 1708 2328 862 1081 1646 118 1828
813 725 169 415 1039 1765 1445 1741 2914 1975 2177 1945 2066 1795 624 1341 780
2050 932 570 1472 783 634 1573 1663 199 359 1956 162 2323 1728 2186 548 811 612
1365 2553 1480 2551 2438 2454 1396 1537 2921 702 1810 1555 630 672 917 1490 2687
647 1906 1759 2387 1355 10 2888 2372 2291 1045 200 2064 2117 2610 2703 1851 1636
688 2014 1022 597 909 2558 1784 1174 163 2141 602 1668 44 2702 1766 186 891 1042
1533 2207 930 1199 1489 498 2363 1954 1021 2896 1778 657 1356 1378 2061 968 2358
377 208 938 2823 1328 1129 1645 2090 2111 1922 1665 1600 2853 1464 2577 1241 1415
690 703 1253 943 2959 649 2542 890 1123 63 304 837 1651 1391 2928 2305 642 2614
711 2280 1140 1264 523 289 1994 2693 255 2101 318 1577 673 1794 367 2429 936 467
1038 1166 1061 1934 1984 2452 459 1572 426 2550 2568 1988 2757 847 2556 1746 594
2945 401 1608 627 685 1565 1658 1680 2994 36 535 1797 2827 1540 1441 577 1662 2734
482 1225 1418 1877 2039 254 1462 1032 539 67 1919 2144 262 550 2258 599 2811 537
1111 198 477 75 404 1687 1326 2775 229 234 1182 635 1590 269 2085 298 2508 742
2852 2906 506 2024 2581 2640 720 764 1119 1895 267 2960 1911 2400 252 1700 2292
1673 111 2532 975 2040 996 608 2209 1874 950 1615 1424 37 1128 1730 2297 1579 993
2992 2492 1307 2277 989 1239 2343 2664 447 2968 2201 892 216 2300 623 1790 96 2606
2126 430 1846 332 2769 2916 127 263 2920 2729 1151 2110 2360 1936 2733 1578 2265
205 2241 1525 222 1972 1236 1059 1903 2701 292 2013 2285 1655 225 1299 2509 532 7
2708 307 2250 514 1071 2223 866 2533 1980 1107 2886 183 999 2398 2719 1261 22 2047
977 1185 492 991 1713 2172 2861 1815 1216 1983 2488 2599 1535 238 1384 528 1053
2268 2080 1155 1230 2815 2017 931 2555 237 2431 2579 2890 2912 398 1743 1309 480

2651 2694 1885 1681 39 848 2194 1657 613 2170 1426 2825 2222 476 792 876 954 2237
405 872 400 804 231 2714 2799 1940 2709 1641 515 1871 113 1084 2062 1916 1569 80
2326 2294 519 2547 1772 1285 699 776 2772 2123 1833 1727 942 2271 331 1135 203
2924 389 1260 1249 2632 511 2435 259 636 816 1958 1425 239 2956 1183 2940 2779
280 134 2314 1011 1419 2021 2642 691 2031 1861 2478 1955 676 2281 2506 2565 640
305 1792 451 502 2812 232 2910 2602 2893 1133 100 670 1599 2181 914 2493 2659 1008
562 801 969 2121 2083 2526 1526 1138 2950 2339 1876 1188 1023 2904 2168 1986 1860
155 1394 1429 1744 2795 1768 2735 336 1965 1907 2716 592 1613 2814 1068 1314 87
1361 362 2813 2189 2164 2391 2032 12 2132 970 1914 2235 178 2220 2425 1254 1102
1647 1959 2746 604 41 1293 1459 1623 2868 2654 2891 2450 2743 2824 1886 1310 1982
350 1439 2058 2408 2430 2334 527 1311 2938 2677 664 584 1842 2393 479 2417 778
1205 65 227 1939 91 963 684 1203 324 1574 2941 2086 2122 196 1226 1132 2663 1805
106 2069 2143 217 1167 512 2045 358 563 2055 2230 2645 665 2002 2167 576 1992 1099
1634 958 632 1171 1029 749 2254 321 588 833 1640 2595 1392 2146 1106 2489 806 1069
2046 2751 2145 1879 2787 953 751 1718 2286 243 2439 1284 2747 1401 1787 2552 1806
2252 2263 880 1277 1725 1704 1267 242 2713 1092 1397 2500 1421 2162 1840 1370
1486 1255 1251 2197 2972 1594 1163 1368 2629 1763 601 607 204 294 2728 1444 641
1080 929 1221 1389 736 1761 1807 73 2212 1969 1482 2307 2463 1618 185 598 2458
1546 1865 94 707 2538 622 2071 1273 541 1066 440 1500 2118 2443 2830 2053 979 1305
1194 266 1622 146 2075 1799 2016 2784 1782 2120 175 2018 637 261 669 77 2762 1683
2955 1447 944 1517 2107 1101 2436 959 2426 820 625 2974 38 281 1520 2331 596 2456
884 224 1703 1108 501 2755 2900 1786 311 1538 960 1631 366 347 1432 2990 92 827
727 2887 1754 2931 2516 1218 384 2301 1774 2546 861 1624 2761 2236 1516 68 18 197
2027 465 1320 2520 1376 249 1190 865 2796 509 2544 1562 2198 1557 2479 1661 1013
2985 2744 2133 794 648 2475 54 1279 1938 414 677 1776 2466 240 2327 1184 905 2661
1507 2635 1968 143 663 2965 454 534 2477 739 1024 1952 1667 1587 779 2412 2926
1372 853 491 344 1004 1077 855 2124 814 580 375 2088 1637 1856 431 868 2764 9 2987
97 542 671 1040 704 2975 1931 2407 1450 544 1126 569 52 1291 2219 2855 2980 2269
1469 173 218 2054 1853 1639 1270 2200 994 852 1682 1228 1209 740 2665 1601 1780
503 881 1325 2446 166 2984 418 2862 2420 824 1362 2766 1120 2908 2480 796 2742
473 1878 1399 356 2176 2571 2570 187 1998 882 1614 340 643 1588 1510 1925 2961

2981 487 2203 2792 2362 538 11 1477 1985 160 1981 450 353 1420 2930 291 2074 335
1094 2597 2791 284 568 1773 2752 3000 1699 1801 2490 2232 2922 2548 2580 1643 902
775 1016 2349 308 709 1282 1250 593 489 2631 2680 301 230 1771 1063 24 2284 2885
2530 1747 2119 2135 1268 1073 682 2863 1100 1136 1187 81 2834 1109 1977 1793 2768
1217 1688 2322 1434 1404 2369 260 857 2846 2835 16 1515 1015 724 1087 2424 1964
2315 985 247 838 1275 2023 2289 1139 2397 2794 1702 1542 2692 2600 2395 2154 982
2590 2199 1373 1354 1247 2707 1544 2819 2969 1411 1671 2589 2705 2953 1625 2240
2929 911 922 1823 2282 316 2444 668 904 109 737 2879 2157 1060 2634 138 327 1248
1966 1932 1821 581 1957 1220 125 1989 246 2818 521 2264 716 1606 1884 817 1457
1943 1036 786 513 2374 1974 2000 2816 741 2091 1929 2624 1468 726 2225 1495 2758
578 1978 2078 2821 2453 2767 2215 1696 2244 1229 793 1760 1353 151 660 1541 2377
1584 396 66 1813 444 2586 2678 2138 1858 2501 910 2627 387 2785 452 2583 976 1096
1330 2266 2541 2029 1243 1312 2179 381 23 689 172 735 2267 2105 951 315 1405 2820
1566 434 571 1315 1244 1359 2396 1014 1529 2560 2584 1611 379 2628 328 921 1498
2686 1252 330 2312 1893 60 1679 1850 2366 763 906 1054 378 2630 1685 1660 1997
1752 821 84 840 1449 2195 2498 2030 510 2006 2566 2406 320 845 698 1454 2822 2125
1148 433 1478 1035 1043 874 171 2676 1164 2625 869 2842 1461 1027 2094 1410 1204
34 1017 372 2468 2008 370 2682 782 1616 1514 2065 2320 2070 1321 2491 1280 90 934
416 1785 2404 770 2619 2871 131 1789 1950 47 1386 2612 1240 1124 1090 472 264 58
2495 2543 184 174 1597 1832 1570 633 1192 2082 342 812 2982 2449 1946 1866 799
1278 2958 986 1448 628 920 1870 894 2715 2997 2057 1213 274 341 2971 1072 69 1342
1287 1234 59 1568 575 747 310 1748 830 1431 1283 2857 412 692 2035 461 2537 64
2948 28 2739 2043 1721 1245 1143 803 825 411 2238 1894 2911 2202 2246 1145 1948
2231 819 2340 2392 1604 2351 1256 2907 466 583 1867 1012 1844 1159 987 446 2149
1502 2367 2033 652 773 2856 1949 1222 1048 2691 1294 2512 1755 678 722 567 2102
988 2763 2104 202 753 371 2737 13 1186 617 1652 925 2895 536 423 2484 2946 2308
626 530 1491 972 1067 1000 193 1076 2527 388 966 2771 1595 2998 442 2214 1689 1127
2963 758 2805 1020 1707 2722 2588 1122 717 2850 1534 445 19 705 831 836 2304 2774
2073 1452 997 2870 1583 385 1930 2287 867 210 2528 1010 1629 1904 1817 85 110 2902
180 1041 297 850 2298 2698 214 2913 1888 1838 102 1757 2832 1025 287 2136 1091
2276 2970 798 1113 1720 228 2831 2306 2390 1064 121 1471 1603 965 2130 2059 88

2697 1412 1466 697 926 844 1162 1854 1458 409 659 1114 1547 2206 1642 730 1781 5
2569 2748 1612 2724 144 399 115 1816 873 653 213 1295 1530 1363 1497 555 1319 1381
1170 1044 2278 2036 2473 795 1288 504 540 1367 547 940 1337 145 937 2174 2148 135
2894 1336 1207 856 2601 2967 1026 2534 485 1134 2483 2646 2889 1 1598 2559 1628
1304 2884 1149 1169 2173 1116 1208 2979 2437 2696 2042 383 2593 2603 2335 639
2313 829 2019 789 2113 2667 2618 1158 1596 1719 448 1374 2510 1398 1455 351 2208
1344 1527 2309 1513 2273 1593 2903 1224 2419 2445 2966 2257 984 443 2781 1731 903
1602 2641 282 1098 394 1666 1338 2470 2841 2658 928 35 338 2736 62 2668 29 2782
2242 864 600 137 2952 918 2139 674 1855 1246 2638 2901 2582 2892 1836 408 2193
1306 1800 221 2607 2899 1701 140 116 2783 1920 2626 460 1633 2169 142 26 1219 2270
1195 896 2319 889 132 1927 2127 897 248 915 2854 1519 1303 2564 2229 1723 715 1031
2933 2804 343 1756 123 1887 1165 2866 78 679 2342 2675 40 907 2150 2756 2171 517
2447 346 952 2504 2433 1237 1289 981 488 1619 397 1352 2160 295 1722 470 572 128
1493 1467 2034 1585 973 2936 828 2780 629 1104 495 1403 1440 2545 1843 1375 268
1347 1735 25 1864 2370 971 1146 2379 2883 774 129 2836 2440 1551 1697 2216 1125
1257 276 2063 386 438 1178 1803 1536 1395 1242 2467 365 500 998 105 731 818 2996
1417 2845 1492 1085 1496 2060 712 251 1364 939 2337 1798 1635 2919 2905 2441 808
436 1971 432 1669 631 27 565 1872 2302 1632 1650 2461 2079 2605 2786 494 1191 104
1413 1942 2773 1423 410 694 1019 272 373 871 992 2056 2469 1198 354 2898 718 253
746 1610 1408 70 957 2465 846 1698 2330 2674 2876 1046 681 1567 395 1944 2459 2099
589 2356 235 1504 1322 1626 2455 1156 744 1402 403 139 2048 885 2609 728 2723 2015
2613 1620 76 887 475 1202 1002 325 777 1329 2710 1400 2068 2245 326 2806 1733 2131
2140 1996 2778 1953 2025 270 1695 2108 1962 1736 2988 2442 2535 2089 2457 2251
2115 1153 901 2259 1900 2373 2005 244 2226 687 2103 420 31 2283 355 2620 1571 250
2114 2243 499 2932 2865 533 2044 2234 1009 471 2255 300 45 157 55 441 2800 2592
1292 1991 2949 1664 421 141 2829 2041 1918 2730 79 1737 2557 841 283 508 1892 1653
2380 941 2434 449 860 314 126 756 552 839 2346 2760 1428 2076 899 1869 1483 1180
2325 4 1560 1414 2499 1393 189 1543 2129 2867 392 644 182 2753 1499 8 1684 912
1057 368 212 2731 788 2256 1484 1357 961 159 1339 2655 2037 923 1506 2338 800 1115
2049 293 1348 949 2833 103 265 654 2776 2161 732 1508 1715 1905 2750 1675 1272
1503 1605 752 1390 278 2826 275 1298 2591 2418 1214 2502 2637 2405 119 1265 1528

```
1883 667 2421 1970 2303 429 2385 1007 696 621 206 2153 595 1750 319 2009 1711 1037
1812 2639 2067 2844 1070 1487 312 329 1351 1822 1379 2523 1829 43 2851 2224 53
2190 2011 1523 2923 117 2909 2290 1899 1377 870 1456 1358 615 2839 683 2525 101
2514 1863 2644 823 2087 1710 1880 759 56 802 967 2134 1276 2472 2410 1670 809 1103
2399 2700 815 2403 1479 1323 2321 1308 1714 1901 2522 1993 2732 89 1274 1383 1758
1302 309 1656 2669 273 1732 1371 496 2357 1839 983 2394 1144 553 1093 190 2332
2660 2361 1215 363 849 380 579 2112 974 2494 1006 1973 2423 748 2505 256 286 651
2503 2859 6 2653 797 875 1443 1430 2382 1826 2650 201 1716 2365 2808 462 1913 390
1764 1512 1941 1857 1961 2798 1923 2915 879 2847 927];
```

```
XTraining = zeros(650,630,3,1800);
```

```
YTrainingPoP = zeros(1800,1);
```

```
for i = 1:1800
```

```
XTraining(:,:,,i) = double(imread("S (" + RN(i) + ").jpg"));
```

```
YTrainingPoP(i,1) = Percent_of_Porosity(RN(i));
```

```
end
```

```
XValidation = zeros(650,630,3,600);
```

```
YValidationPoP = zeros(600,1);
```

```
for i = 1801:2400
```

```
XValidation(:,:,,i-1800) = double(imread("S (" + RN(i) + ").jpg"));
```

```
YValidationPoP(i-1800,1) = Percent_of_Porosity(RN(i));
```

```
end
```



```

XTesting = zeros(650,630,3,600);
YTestingPoP = zeros(600,1);

for i = 2401:3000

XTesting(:, :, :, i-2400) = double(imread("S (" + RN(i) + ").jpg"));
YTestingPoP(i-2400,1) = Percent_of_Porosity(RN(i));

end

end

```

1.10 Bees Regression Convolutional Neural Network (BA-RCNN)

```

clc;

clear;

close all;

```

%Load and Explore Image Data

```

ArtificialPorosityImagesDatasetPath =
fullfile('/nfshome','store02','users','c.c1881324','Documents','MATLAB','3000
Slices','*.jpg');

[XTraining, YTrainingPoP, XValidation, YValidationPoP, XTesting, YTestingPoP] =
loadAPIData(ArtificialPorosityImagesDatasetPath);

```

%Define the Problem (Objective Function)

```
ObjFcn =  
makeObjFcn(XTraining,YTrainingPoP,XValidation,YValidationPoP,XTesting,YTesting  
PoP);
```

```
nVar=4; %Number of Variables
```

```
VarSize=[1 nVar]; %Size of Variables
```

```
VarMin=[1 0.01 0.8 1e-10]; %Lower Bound of Variables
```

```
VarMax=[3 0.012 0.98 1e-2]; %Upper Bound of Variables
```

%Define Bees Algorithm Parameters

```
MaxIt=1; %Maximum Number of Iterations
```

```
nScoutBee=4; %Scout Bees
```

```
nSelectedSite=round(0.5*nScoutBee); %Selected Sites
```

```
nEliteSite=1; %Selected Elite Sites
```

```
nSelectedSiteBee=round(0.5*nScoutBee); %Recruited Bees for Selected Sites
```

```
nEliteSiteBee=2*nSelectedSiteBee; %Recruited Bees for Elite Sites
```

```
r=0.1*(VarMax-VarMin); %Neighbourhood Radius
```

```
rdamp=0.95; %Neighbourhood Radius Damp Rate
```

%Initialize Empty Bee Structure

```
empty_bee.Position=[];
```

```
empty_bee.Error=[];
```

%Initialize Bees Array

```
bee= repmat(empty_bee,nScoutBee,1);
```

%Create New Solutions

```
for i=1:nScoutBee
```

```
    bee(i).Position=unifrnd(VarMin,VarMax,VarSize);
```

```
    bee(i).Error=ObjFcn(bee(i).Position);
```

```
end
```

%Sort the Solution

```
[~, SortOrder]=sort([bee.Error]);
```

```
bee=bee(SortOrder);
```

%Update Best Solution Ever Found

```
BestSol=bee(1);
```

%Create Array to Hold Best Error Values

```
LowestError=zeros(MaxIt,1);
```

%Create Bees Algorithm Main Loop

```
for it=1:MaxIt
```

%Elite Sites

```
for i=1:nEliteSite

    bestnewbee.Error=inf;

    for j=1:nEliteSiteBee

        newbee.Position=PerformBeeDance(bee(i).Position,r);

        newbee.Error=ObjFcn(newbee.Position);

        if newbee.Error<bestnewbee.Error

            bestnewbee=newbee;

        end

    end

end

if bestnewbee.Error<bee(i).Error

    bee(i)=bestnewbee;

end

end
```

%Selected Non-Elite Sites

```
for i=nEliteSite+1:nSelectedSite

    bestnewbee.Error=inf;

    for j=1:nSelectedSiteBee

        newbee.Position=PerformBeeDance(bee(i).Position,r);
```

```

        newbie.Error=ObjFcn(newbee.Position);

        if newbie.Error<bestnewbee.Error

            bestnewbee=newbee;

        end

    end

end

if bestnewbee.Error<bee(i).Error

    bee(i)=bestnewbee;

end

end

%Non-Selected Sites

for i=nSelectedSite+1:nScoutBee

    bee(i).Position=unifrnd(VarMin,VarMax,VarSize);

    bee(i).Error=ObjFcn(bee(i).Position);

end

%Sort the Solution

[~, SortOrder]=sort([bee.Error]);

bee=bee(SortOrder);

%Update Best Solution Ever Found

BestSol=bee(1);

```

%Store Best Error Ever Found

LowestError(it)=BestSol.Error;

OptimalSolution=BestSol.Position;

%Display Iteration Information

disp(['Iteration ' num2str(it) ': Lowest Error = ' num2str(LowestError(it))]);

%Define Damp Neighborhood Radius

r=r*rdamp;

end

%Display the Results

figure;

%Make a Plot for Lowest Error

semilogy(LowestError,'LineWidth',2);

xlabel('Iteration');

ylabel('Lowest Error');

function ObjFcn =

makeObjFcn(XTraining,YTrainingPoP,XValidation,YValidationPoP,XTesting,YTesting
PoP)

```
ObjFcn = @ValidationErrorFunction;
```

```
function AverageValidationError = ValidationErrorFunction(OptimalSolution)
```

%Specify Convolutional Neural Network Architecture

```
imageSize = [650 630 3];
```

```
layersPoP = [
```

```
    imageInputLayer(imageSize)
```

```
        convBlock(5,8, round(OptimalSolution(1,1)))
```

```
        averagePooling2dLayer(4,'Stride',4)
```

```
        convBlock(5,16, round(OptimalSolution(1,1)))
```

```
        averagePooling2dLayer(4,'Stride',4)
```

```
        convBlock(5,32, round(OptimalSolution(1,1)))
```

```
        averagePooling2dLayer(4,'Stride',4)
```

```
        convBlock(5,64, round(OptimalSolution(1,1)))
```

```

averagePooling2dLayer(4,'Stride',4)

convBlock(5,128, round(OptimalSolution(1,1)))

fullyConnectedLayer(1)

regressionLayer];

miniBatchSize = 64;

validationFrequency = floor(1800/miniBatchSize);

optionsPoP = trainingOptions('sgdm', ...
    'InitialLearnRate', OptimalSolution(1,2), ...
    'Momentum', OptimalSolution(1,3), ...
    'L2Regularization', OptimalSolution(1,4), ...
    'MaxEpochs',20, ...
    'MiniBatchSize',miniBatchSize, ...
    'ValidationFrequency',validationFrequency, ...
    'ValidationData',{XValidation,YValidationPoP(:,1)}, ...
    'Shuffle','every-epoch', ...
    'Plots','training-progress');

```

%Train Network using Training Data

```
netPoP = trainNetwork(XTraining,YTrainingPoP(:,1),layersPoP,optionsPoP);
```


%Predict Responses and Compute Accuracy and Error

c = 1;

PredictedTrainingPoP(:,c) = predict(netPoP,XTraining);

PredictedValidationPoP(:,c) = predict(netPoP,XValidation);

PredictedTestingPoP(:,c) = predict(netPoP,XTesting);

TrainingError(:,c) = YTrainingPoP(:,1) - PredictedTrainingPoP(:,c);

AbsoluteTrainingError(:,c) = abs(TrainingError(:,c));

AverageTrainingError(:,c) = sum(AbsoluteTrainingError(:,c))/1800;

numCorrectTrainingPoP(:,c) = sum(abs(TrainingError(:,c)) < 0.05);

TrainingAccuracyPoP(:,c) = (numCorrectTrainingPoP(:,c))/1800*100;

ValidationError(:,c) = YValidationPoP(:,1) - PredictedValidationPoP(:,c);

AbsoluteValidationError(:,c) = abs(ValidationError(:,c));

AverageValidationError(:,c) = sum(AbsoluteValidationError(:,c))/600;

numCorrectValidationPoP(:,c) = sum(abs(ValidationError(:,c)) < 0.05);

ValidationAccuracyPoP(:,c) = (numCorrectValidationPoP(:,c))/600*100;

TestingError(:,c) = YTestingPoP(:,1) - PredictedTestingPoP(:,c);

AbsoluteTestingError(:,c) = abs(TestingError(:,c));

AverageTestingError(:,c) = sum(AbsoluteTestingError(:,c))/600;

numCorrectTestingPoP(:,c) = sum(abs(TestingError(:,c)) < 0.05);

TestingAccuracyPoP(:,c) = (numCorrectTestingPoP(:,c))/600*100;

```
end
```

```
end
```

%Create Bees Dance Function

```
function y=PerformBeeDance(ValidationErrorFunction,r)
```

```
    nVar=numel(ValidationErrorFunction);
```

```
    k=randi([1 nVar]);
```

```
    y=ValidationErrorFunction;
```

```
    y(k)=ValidationErrorFunction(k)+unifrnd(-r,r);
```

```
end
```

```
function layers = convBlock(FilterSize,NumberOfFilters,SectionDepth)
```

```
layers = [
```

```
    convolution2dLayer(FilterSize,NumberOfFilters,'Padding','same')
```

```
    batchNormalizationLayer
```

```
    reluLayer];
```

```
layers = repmat(layers,SectionDepth,1);
```

```
end
```

```
function [XTraining,YTrainingPoP,XValidation,YValidationPoP,XTesting,YTestingPoP]
= loadAPIData(ArtificialPorosityImagesDatasetPath)
```

```
location = dir(ArtificialPorosityImagesDatasetPath);
```

```
Percent_of_Porosity = Actual_Percent_of_Porosity;
```

```
RN = [1834 919 935 220 2897 1891 32 675 2106 2725 424 1690 900 1548 2596 2947 505
1802 2706 149 2648 2793 165 2726 1859 1769 2022 878 564 2109 646 566 2376 2163
518 2496 898 522 1049 2192 2519 2221 620 1211 2152 2191 2617 618 2877 769 757 1692
164 2155 554 277 947 1509 98 2554 1382 2869 2633 413 2801 2934 2262 1327 2770 1575
288 1926 2749 2573 695 2100 1281 1052 2511 754 279 57 832 656 1446 2371 2092 2288
1097 1545 1474 2649 2293 1232 787 2986 745 1706 1868 2685 2539 124 130 2184 2838
893 2409 1259 348 1095 481 714 2359 1082 1387 605 729 2211 1343 1030 693 120 1621
245 1609 99 1990 2413 188 1210 558 369 1809 2954 1917 374 1463 1083 2993 2279 2052
574 1873 1318 1157 863 1921 2872 1835 1501 71 603 2622 150 181 1804 560 1783 2001
1262 1331 1003 1881 858 525 2411 191 2623 2704 1141 2873 2 1709 2004 93 551 136
1511 2777 1819 2333 658 2943 2720 1845 1160 1676 1849 2882 1889 425 1848 72 2738
2828 154 2448 1775 2007 2810 1672 1987 1335 1350 428 1818 1296 2615 2142 170 211
176 772 133 1524 1028 1131 1638 1554 2213 895 546 1078 1058 417 156 701 1340 226
463 306 1712 2951 1862 1231 2991 851 1522 2718 1406 2864 1726 1745 1882 1915 955
734 1488 1847 888 2529 2353 1269 2204 192 1297 1649 2165 1137 2802 2182 743 333
2072 980 299 1563 2942 2248 1753 2964 1999 610 2485 1770 2918 708 591 557 1694
883 723 1193 2681 1734 2158 1453 1960 1552 2318 95 1677 1334 549 478 2789 17 49
2745 1433 1705 1223 680 886 490 14 1724 1654 1627 2917 516 194 2978 790 2616 2299
83 2518 619 219 1197 337 1130 1518 1693 345 2657 1995 2462 407 437 2422 2274 1617
843 614 1055 2097 114 2386 1075 1233 2476 1385 107 20 74 710 1559 2084 1924 2329
1301 2662 382 2354 2205 2095 2740 978 1034 1896 469 258 1154 766 2081 908 1079
1349 1369 2880 2699 30 1739 2317 1644 2561 1088 1470 1564 2670 2003 1271 1902 493
```

1416 152 1235 2077 1580 1172 2296 1589 1779 1332 48 2860 2196 765 323 2937 1206
738 1442 303 666 1777 1852 2096 2765 361 1346 2562 573 2116 148 167 1831 826 61
233 611 2341 995 474 1152 296 2384 456 781 2636 2316 1814 2028 2507 2688 1898 2432
1407 458 1909 236 609 586 2727 2513 2352 486 1910 2228 2652 1582 108 2217 497 2345
1481 2983 1494 1333 2350 2344 655 2939 771 435 2536 529 483 2672 457 2925 2383
215 360 2788 1830 1539 2874 805 561 2995 1824 1056 1313 945 2578 2415 1648 661
1674 2187 2973 2249 2927 1581 933 2210 706 2754 1550 2368 179 2401 1435 2881 1951
422 531 686 822 2999 406 2336 645 1286 285 733 2957 357 21 2647 807 241 223 1176
700 1065 1089 2375 755 1005 2843 2247 1161 2684 1659 719 468 158 1691 1047 2355
2402 760 2671 948 455 1074 1290 2563 1227 2517 2471 2451 762 877 2166 2521 916
810 2878 1324 1549 2414 859 2151 2803 1476 1033 2427 2128 2817 2540 2180 2218
2576 924 3 2585 524 2020 2159 1912 2183 1576 1808 453 2711 2837 1177 302 1181 1827
2935 1051 2389 352 2849 2275 168 1212 2497 2962 1553 946 2324 2272 1729 2717 2175
209 1345 1963 1437 1168 964 1837 2227 1591 1630 2233 1475 1105 147 2260 1820 2381
1979 334 2464 2840 2858 484 1738 520 2574 2712 1258 349 1142 419 616 2848 990 1742
2093 2549 1937 391 2643 2977 1875 2666 2481 2683 962 2621 2378 1811 33 2531 1147
2178 834 51 86 750 1189 835 317 50 1451 2790 2010 82 2348 2474 1531 956 590 1196
2567 1890 2807 2364 2188 662 2604 1767 1947 2156 1001 1788 1438 1173 556 1050 207
2679 582 1717 585 2656 271 761 1586 1388 2944 1175 290 46 2388 543 2482 2673 1238
1791 177 721 768 1607 785 1200 1935 1117 2695 2575 842 1908 1558 322 257 1561 2012
2261 1062 1749 650 2587 1825 161 1740 2487 1201 545 2239 791 402 2486 2295 1317
42 2989 1266 1110 606 2611 2098 2416 2428 1521 767 427 1532 2741 2524 339 1933
2689 2253 376 1018 1897 1409 1380 1422 1121 638 1460 2038 1300 2797 1505 913 1465
393 587 1179 2598 1316 122 1841 2594 2572 1263 2147 507 439 153 784 1485 2690 2310
1976 2460 2347 2608 1360 1762 2026 2515 1118 364 313 1436 464 1967 1751 526 15
2976 1556 1473 854 2185 1086 1928 1686 2721 2051 1150 2875 559 2759 1366 1678
2809 1592 2311 713 1427 195 1112 112 2137 1796 1708 2328 862 1081 1646 118 1828
813 725 169 415 1039 1765 1445 1741 2914 1975 2177 1945 2066 1795 624 1341 780
2050 932 570 1472 783 634 1573 1663 199 359 1956 162 2323 1728 2186 548 811 612
1365 2553 1480 2551 2438 2454 1396 1537 2921 702 1810 1555 630 672 917 1490 2687
647 1906 1759 2387 1355 10 2888 2372 2291 1045 200 2064 2117 2610 2703 1851 1636

688 2014 1022 597 909 2558 1784 1174 163 2141 602 1668 44 2702 1766 186 891 1042
1533 2207 930 1199 1489 498 2363 1954 1021 2896 1778 657 1356 1378 2061 968 2358
377 208 938 2823 1328 1129 1645 2090 2111 1922 1665 1600 2853 1464 2577 1241 1415
690 703 1253 943 2959 649 2542 890 1123 63 304 837 1651 1391 2928 2305 642 2614
711 2280 1140 1264 523 289 1994 2693 255 2101 318 1577 673 1794 367 2429 936 467
1038 1166 1061 1934 1984 2452 459 1572 426 2550 2568 1988 2757 847 2556 1746 594
2945 401 1608 627 685 1565 1658 1680 2994 36 535 1797 2827 1540 1441 577 1662 2734
482 1225 1418 1877 2039 254 1462 1032 539 67 1919 2144 262 550 2258 599 2811 537
1111 198 477 75 404 1687 1326 2775 229 234 1182 635 1590 269 2085 298 2508 742
2852 2906 506 2024 2581 2640 720 764 1119 1895 267 2960 1911 2400 252 1700 2292
1673 111 2532 975 2040 996 608 2209 1874 950 1615 1424 37 1128 1730 2297 1579 993
2992 2492 1307 2277 989 1239 2343 2664 447 2968 2201 892 216 2300 623 1790 96 2606
2126 430 1846 332 2769 2916 127 263 2920 2729 1151 2110 2360 1936 2733 1578 2265
205 2241 1525 222 1972 1236 1059 1903 2701 292 2013 2285 1655 225 1299 2509 532 7
2708 307 2250 514 1071 2223 866 2533 1980 1107 2886 183 999 2398 2719 1261 22 2047
977 1185 492 991 1713 2172 2861 1815 1216 1983 2488 2599 1535 238 1384 528 1053
2268 2080 1155 1230 2815 2017 931 2555 237 2431 2579 2890 2912 398 1743 1309 480
2651 2694 1885 1681 39 848 2194 1657 613 2170 1426 2825 2222 476 792 876 954 2237
405 872 400 804 231 2714 2799 1940 2709 1641 515 1871 113 1084 2062 1916 1569 80
2326 2294 519 2547 1772 1285 699 776 2772 2123 1833 1727 942 2271 331 1135 203
2924 389 1260 1249 2632 511 2435 259 636 816 1958 1425 239 2956 1183 2940 2779
280 134 2314 1011 1419 2021 2642 691 2031 1861 2478 1955 676 2281 2506 2565 640
305 1792 451 502 2812 232 2910 2602 2893 1133 100 670 1599 2181 914 2493 2659 1008
562 801 969 2121 2083 2526 1526 1138 2950 2339 1876 1188 1023 2904 2168 1986 1860
155 1394 1429 1744 2795 1768 2735 336 1965 1907 2716 592 1613 2814 1068 1314 87
1361 362 2813 2189 2164 2391 2032 12 2132 970 1914 2235 178 2220 2425 1254 1102
1647 1959 2746 604 41 1293 1459 1623 2868 2654 2891 2450 2743 2824 1886 1310 1982
350 1439 2058 2408 2430 2334 527 1311 2938 2677 664 584 1842 2393 479 2417 778
1205 65 227 1939 91 963 684 1203 324 1574 2941 2086 2122 196 1226 1132 2663 1805
106 2069 2143 217 1167 512 2045 358 563 2055 2230 2645 665 2002 2167 576 1992 1099
1634 958 632 1171 1029 749 2254 321 588 833 1640 2595 1392 2146 1106 2489 806 1069

2046 2751 2145 1879 2787 953 751 1718 2286 243 2439 1284 2747 1401 1787 2552 1806
2252 2263 880 1277 1725 1704 1267 242 2713 1092 1397 2500 1421 2162 1840 1370
1486 1255 1251 2197 2972 1594 1163 1368 2629 1763 601 607 204 294 2728 1444 641
1080 929 1221 1389 736 1761 1807 73 2212 1969 1482 2307 2463 1618 185 598 2458
1546 1865 94 707 2538 622 2071 1273 541 1066 440 1500 2118 2443 2830 2053 979 1305
1194 266 1622 146 2075 1799 2016 2784 1782 2120 175 2018 637 261 669 77 2762 1683
2955 1447 944 1517 2107 1101 2436 959 2426 820 625 2974 38 281 1520 2331 596 2456
884 224 1703 1108 501 2755 2900 1786 311 1538 960 1631 366 347 1432 2990 92 827
727 2887 1754 2931 2516 1218 384 2301 1774 2546 861 1624 2761 2236 1516 68 18 197
2027 465 1320 2520 1376 249 1190 865 2796 509 2544 1562 2198 1557 2479 1661 1013
2985 2744 2133 794 648 2475 54 1279 1938 414 677 1776 2466 240 2327 1184 905 2661
1507 2635 1968 143 663 2965 454 534 2477 739 1024 1952 1667 1587 779 2412 2926
1372 853 491 344 1004 1077 855 2124 814 580 375 2088 1637 1856 431 868 2764 9 2987
97 542 671 1040 704 2975 1931 2407 1450 544 1126 569 52 1291 2219 2855 2980 2269
1469 173 218 2054 1853 1639 1270 2200 994 852 1682 1228 1209 740 2665 1601 1780
503 881 1325 2446 166 2984 418 2862 2420 824 1362 2766 1120 2908 2480 796 2742
473 1878 1399 356 2176 2571 2570 187 1998 882 1614 340 643 1588 1510 1925 2961
2981 487 2203 2792 2362 538 11 1477 1985 160 1981 450 353 1420 2930 291 2074 335
1094 2597 2791 284 568 1773 2752 3000 1699 1801 2490 2232 2922 2548 2580 1643 902
775 1016 2349 308 709 1282 1250 593 489 2631 2680 301 230 1771 1063 24 2284 2885
2530 1747 2119 2135 1268 1073 682 2863 1100 1136 1187 81 2834 1109 1977 1793 2768
1217 1688 2322 1434 1404 2369 260 857 2846 2835 16 1515 1015 724 1087 2424 1964
2315 985 247 838 1275 2023 2289 1139 2397 2794 1702 1542 2692 2600 2395 2154 982
2590 2199 1373 1354 1247 2707 1544 2819 2969 1411 1671 2589 2705 2953 1625 2240
2929 911 922 1823 2282 316 2444 668 904 109 737 2879 2157 1060 2634 138 327 1248
1966 1932 1821 581 1957 1220 125 1989 246 2818 521 2264 716 1606 1884 817 1457
1943 1036 786 513 2374 1974 2000 2816 741 2091 1929 2624 1468 726 2225 1495 2758
578 1978 2078 2821 2453 2767 2215 1696 2244 1229 793 1760 1353 151 660 1541 2377
1584 396 66 1813 444 2586 2678 2138 1858 2501 910 2627 387 2785 452 2583 976 1096
1330 2266 2541 2029 1243 1312 2179 381 23 689 172 735 2267 2105 951 315 1405 2820
1566 434 571 1315 1244 1359 2396 1014 1529 2560 2584 1611 379 2628 328 921 1498

2686 1252 330 2312 1893 60 1679 1850 2366 763 906 1054 378 2630 1685 1660 1997
1752 821 84 840 1449 2195 2498 2030 510 2006 2566 2406 320 845 698 1454 2822 2125
1148 433 1478 1035 1043 874 171 2676 1164 2625 869 2842 1461 1027 2094 1410 1204
34 1017 372 2468 2008 370 2682 782 1616 1514 2065 2320 2070 1321 2491 1280 90 934
416 1785 2404 770 2619 2871 131 1789 1950 47 1386 2612 1240 1124 1090 472 264 58
2495 2543 184 174 1597 1832 1570 633 1192 2082 342 812 2982 2449 1946 1866 799
1278 2958 986 1448 628 920 1870 894 2715 2997 2057 1213 274 341 2971 1072 69 1342
1287 1234 59 1568 575 747 310 1748 830 1431 1283 2857 412 692 2035 461 2537 64
2948 28 2739 2043 1721 1245 1143 803 825 411 2238 1894 2911 2202 2246 1145 1948
2231 819 2340 2392 1604 2351 1256 2907 466 583 1867 1012 1844 1159 987 446 2149
1502 2367 2033 652 773 2856 1949 1222 1048 2691 1294 2512 1755 678 722 567 2102
988 2763 2104 202 753 371 2737 13 1186 617 1652 925 2895 536 423 2484 2946 2308
626 530 1491 972 1067 1000 193 1076 2527 388 966 2771 1595 2998 442 2214 1689 1127
2963 758 2805 1020 1707 2722 2588 1122 717 2850 1534 445 19 705 831 836 2304 2774
2073 1452 997 2870 1583 385 1930 2287 867 210 2528 1010 1629 1904 1817 85 110 2902
180 1041 297 850 2298 2698 214 2913 1888 1838 102 1757 2832 1025 287 2136 1091
2276 2970 798 1113 1720 228 2831 2306 2390 1064 121 1471 1603 965 2130 2059 88
2697 1412 1466 697 926 844 1162 1854 1458 409 659 1114 1547 2206 1642 730 1781 5
2569 2748 1612 2724 144 399 115 1816 873 653 213 1295 1530 1363 1497 555 1319 1381
1170 1044 2278 2036 2473 795 1288 504 540 1367 547 940 1337 145 937 2174 2148 135
2894 1336 1207 856 2601 2967 1026 2534 485 1134 2483 2646 2889 1 1598 2559 1628
1304 2884 1149 1169 2173 1116 1208 2979 2437 2696 2042 383 2593 2603 2335 639
2313 829 2019 789 2113 2667 2618 1158 1596 1719 448 1374 2510 1398 1455 351 2208
1344 1527 2309 1513 2273 1593 2903 1224 2419 2445 2966 2257 984 443 2781 1731 903
1602 2641 282 1098 394 1666 1338 2470 2841 2658 928 35 338 2736 62 2668 29 2782
2242 864 600 137 2952 918 2139 674 1855 1246 2638 2901 2582 2892 1836 408 2193
1306 1800 221 2607 2899 1701 140 116 2783 1920 2626 460 1633 2169 142 26 1219 2270
1195 896 2319 889 132 1927 2127 897 248 915 2854 1519 1303 2564 2229 1723 715 1031
2933 2804 343 1756 123 1887 1165 2866 78 679 2342 2675 40 907 2150 2756 2171 517
2447 346 952 2504 2433 1237 1289 981 488 1619 397 1352 2160 295 1722 470 572 128
1493 1467 2034 1585 973 2936 828 2780 629 1104 495 1403 1440 2545 1843 1375 268

1347 1735 25 1864 2370 971 1146 2379 2883 774 129 2836 2440 1551 1697 2216 1125
1257 276 2063 386 438 1178 1803 1536 1395 1242 2467 365 500 998 105 731 818 2996
1417 2845 1492 1085 1496 2060 712 251 1364 939 2337 1798 1635 2919 2905 2441 808
436 1971 432 1669 631 27 565 1872 2302 1632 1650 2461 2079 2605 2786 494 1191 104
1413 1942 2773 1423 410 694 1019 272 373 871 992 2056 2469 1198 354 2898 718 253
746 1610 1408 70 957 2465 846 1698 2330 2674 2876 1046 681 1567 395 1944 2459 2099
589 2356 235 1504 1322 1626 2455 1156 744 1402 403 139 2048 885 2609 728 2723 2015
2613 1620 76 887 475 1202 1002 325 777 1329 2710 1400 2068 2245 326 2806 1733 2131
2140 1996 2778 1953 2025 270 1695 2108 1962 1736 2988 2442 2535 2089 2457 2251
2115 1153 901 2259 1900 2373 2005 244 2226 687 2103 420 31 2283 355 2620 1571 250
2114 2243 499 2932 2865 533 2044 2234 1009 471 2255 300 45 157 55 441 2800 2592
1292 1991 2949 1664 421 141 2829 2041 1918 2730 79 1737 2557 841 283 508 1892 1653
2380 941 2434 449 860 314 126 756 552 839 2346 2760 1428 2076 899 1869 1483 1180
2325 4 1560 1414 2499 1393 189 1543 2129 2867 392 644 182 2753 1499 8 1684 912
1057 368 212 2731 788 2256 1484 1357 961 159 1339 2655 2037 923 1506 2338 800 1115
2049 293 1348 949 2833 103 265 654 2776 2161 732 1508 1715 1905 2750 1675 1272
1503 1605 752 1390 278 2826 275 1298 2591 2418 1214 2502 2637 2405 119 1265 1528
1883 667 2421 1970 2303 429 2385 1007 696 621 206 2153 595 1750 319 2009 1711 1037
1812 2639 2067 2844 1070 1487 312 329 1351 1822 1379 2523 1829 43 2851 2224 53
2190 2011 1523 2923 117 2909 2290 1899 1377 870 1456 1358 615 2839 683 2525 101
2514 1863 2644 823 2087 1710 1880 759 56 802 967 2134 1276 2472 2410 1670 809 1103
2399 2700 815 2403 1479 1323 2321 1308 1714 1901 2522 1993 2732 89 1274 1383 1758
1302 309 1656 2669 273 1732 1371 496 2357 1839 983 2394 1144 553 1093 190 2332
2660 2361 1215 363 849 380 579 2112 974 2494 1006 1973 2423 748 2505 256 286 651
2503 2859 6 2653 797 875 1443 1430 2382 1826 2650 201 1716 2365 2808 462 1913 390
1764 1512 1941 1857 1961 2798 1923 2915 879 2847 927];

XTraining = zeros(650,630,3,1800);

YTrainingPoP = zeros(1800,1);


```

for i = 1:1800

XTraining(:,:,,i) = double(imread("S (" + RN(i) + ").jpg"));

YTrainingPoP(i,1) = Percent_of_Porosity(RN(i));

end

XValidation = zeros(650,630,3,600);

YValidationPoP = zeros(600,1);

for i = 1801:2400

XValidation(:,:,,i-1800) = double(imread("S (" + RN(i) + ").jpg"));

YValidationPoP(i-1800,1) = Percent_of_Porosity(RN(i));

end

XTesting = zeros(650,630,3,600);

YTestingPoP = zeros(600,1);

for i = 2401:3000

XTesting(:,:,,i-2400) = double(imread("S (" + RN(i) + ").jpg"));

YTestingPoP(i-2400,1) = Percent_of_Porosity(RN(i));

end

end

```

Appendix 2: Curriculum Vitae

Profile

Name: Nawaf Mohammad H Alamri

Date of Birth: 20/10/1992

Living area: Alrayyan Dist – Alharamain Road – Jeddah – Saudi Arabia - 23741

Mobile No: 07585697168

Email: alamrinm@cardiff.ac.uk

Nationality: Saudi Arabia

Objective

I would like to focus my career on teaching, making research and consulting in industrial engineering cases by using the best practices in a creative way. I also aim to develop new skills and competence and acquire practical experience to be compatible with every changing work demand of the labour market

Education

- Currently, I am studying PhD in Engineering in the field of in Mechanics, Material and Advanced Manufacturing at Cardiff University since 2019
- MSc. in Manufacturing Engineering Innovation and Management at Cardiff University with **Merit Classification** (2018-2020)
- MSc. in Industrial Engineering at King Abdulaziz University with a GPA **4.97 out of 5** (2016-2019)
- BSc. in Industrial Engineering at King Abdulaziz University with a GPA **4.97 out of 5** (2010-2015)
- I awarded Superior certificate in 2010-2011, 2011-2012, 2012-2013, and 2013-2014
- I graduated from high school in 2010 with Grade **99.95%**

Experience

- Currently, I am Teaching Assistant at King Abdulaziz University since 2017 and at Cardiff University since 2019
- 2 years of experience at supply chain department in Abdul Latif Jameel Group (2015-2017)
- Part time job in research and consulting institute at King Abdulaziz University (2013-2015)
- Students' representative in ABET evaluation meetings during the assessment visit
- Teaching assistant (Design of Experiments course) in Industrial Engineering department at King Abdulaziz University
- Grader in Introduction to Engineering Design course at King Abdulaziz University

Training

- Entrepreneurship course at Al-Ahli Bank entitled *How to start your small project* (2013)
- Riyali course (Financial Literacy Program) at King Abdulaziz University (2013)
- English course at New Horizons institute (2010)

Skills

- Developing and analysing Convolutional neural network models using MATLAB
- Developing and analysing neural network models using MATLAB
- Performing statistical analysis using Minitab
- Analyzing engineering experiments using Design-Expert
- Performing data mining using WEKA software
- Designing database system using oracle application
- Simulation and Animation using Arena Package
- Analysis of Supply Chain systems
- Conducting management self-assessment based on EFQM criteria
- Applications of project management tools & techniques

Projects

- **Applying Deep Learning Algorithm to Achieve Automated Porosity Assessment for Additive Layer Manufacturing Process**
Developing convolutional neural network and long short term memory algorithms to predict the percent of porosity in the finished selective laser melting parts achieving automated porosity assessment
- **Applying Machine Learning Algorithm for a Milling Process Simulator for Process Modelling and Optimisation**
Implementing neural network algorithm for a milling process simulator in order to model and optimize the process
- **Water and Electricity Consumption Analysis using DEA and Regression Analysis: Future Prediction Considering Saudi Vision 2030**
Identifying the relative efficiency of water consumption for the five main regions in Saudi Arabia and predicting the future consumption and production of the water and electricity considering strategy in Saudi Vision 2030
- **Research and Consultant Institute Project**
Redesigning business process in this institute using simulation, design of experiment and operation Research to solve this problem
- **Ejector Pump Project**
Applying response surface technique to find the best ejector pump for a system, the project was a part of joint work between King Abdulaziz University and Missouri University in USA
- **Maintenance C-check Project**
Simulating and optimizing the maintenance C-Check process at Saudi Airlines
- **EFQM Project**
Conducting management self-assessment based on EFQM criteria on research and consulting institute
- **Assembly Task Performance**
Identifying the most influential factors on human body during assembly task performance

- **Database Project**
Designing a reservation system for an airline company using oracle applications
- **Reaction Time Project**
Identifying the most influential factors on human reaction time using Design of Experiments technique

Voluntary Work

Member of the organizing committee of the eighth Engineering day (2014) at King Abdulaziz University (Leader of the team in charge of introducing Industrial Engineering field to junior students)

References

- **Prof. Seraj Y. Abed**

Professor of Industrial Engineering Department at King Abdulaziz University,

Mobile No: 00966505601288

Email: sabed@kau.edu.sa
- **Prof. Khalid A. Alghamdi**

Professor of Industrial Engineering Department at King Abdulaziz University,

Mobile No: 00966533333194

Email: kaaalghamdi@kau.edu.sa
- **Prof. Hisham M. Alidrisi**

Professor of Industrial Engineering Department at King Abdulaziz University,

Mobile No: 00966542516516

Email: hmalidrisi@kau.eu.sa