SINERGI Vol. 26, No. 3, October 2022: 335-342
http://publikasi.mercubuana.ac.id/index.php/sinergi
http://doi.org/10.22441/sinergi.2022.3.009

Defect classification of radius shaping in the tire curing process using Fine-Tuned Deep Neural Network

Zendi Iklima[1], Bugi Nur Rohman[1], Rahmat Muwardi[1], Asif Khan[2], Zody Arifiansyah[3]
[1]Department of Electrical Engineering, Faculty of Engineering, Universitas Mercu Buana, Indonesia
[2]School of Computer Science and Technology, Beijing Institute of Technology, China
[3]Department of Information Technology, Faculty of Computer Science, Universitas Esa Unggul, Indonesia

Abstract
The curing or vulcanization process is the final stage of the tire manufacturing process, where the properties of the tire compound change from rubber-plastic material to become elastic by forming cross-links in its molecular structure. The green tire is formed in the curing process, which is placed on the bottom mould, and the inside of the green tire surrounds the bladder. The top mould will close to carry out the next curing process. In the process of closing the mould, there is a shaping process of forming a green tire placed on the bladder and given a proportional pressure. Improper or abnormal radius shaping results cause seventy percent of product defects. This paper proposed abnormal detection of radius shaping in the curing process using fine-tuned Deep Neural Network. Several DNN models have been examined to analyze an optimized DNN model for abnormal detection of radius shaping in the curing process. The fine-tuned DNN architecture has been exported for the curing system. The DNN was trained with a training accuracy of 97.88%, a validation accuracy of 95%, a testing accuracy of 100%, and a loss of 4.93%.

Keywords:
Abnormal Detection;
Deep Neural Network;
Fined Tuned;
Radius Shaping;

Article History:
Received: January 19, 2022
Revised: February 23, 2022
Accepted: March 11, 2022
Published: October 6, 2022

Corresponding Author:
Zendi Iklima,
Electrical Engineering Department, Universitas Mercu Buana, Indonesia
Email:
zendi.iklima@mercubuana.ac.id

INTRODUCTION

Tire curing is the last stage of the tire manufacturing process. Green tires produced from the assembly process are then fed into the curing area for vulcanization. The curing process is vulcanization with high temperature and pressure [1][2] with polymer (rubber), carbon black, and sulfur with the help of chemical compounds to obtain the required characteristics so that it becomes a quality tire product [3].

In the green tire vulcanization process, defects or products can still reduce product quality. Six factors affect product defects: open mould, blond tread, inner pass, crack bead, mould bead, and under-cure. Under Cure defects in a green tire, production has a percentage of 40%. This is caused by bladder conditions, overload, oil pump drop, pneumatic, and others. This study focuses on classifying under-cured product defects caused by bladder conditions, which are divided into crown bare (CB) defect and cracked inner liner (CIL) defect. A crown bare occurs when the radius or pressure is too high while the delay is too low. A cracked inner liner defect occurs because the radius or pressure is too low while the delay is too high.

Several artificial intelligence algorithms are used to reduce the number of manufacturing defects, especially in the curing process. In a model tire, the capacity of ANN designs to predict optimal curing durations for 11 distinct rubber compositions was investigated. The equivalent cure idea, which has been utilized in the rubber and tire industries for years, was used to predict optimum cure periods for the identical compounds. In order to compare the results of

two methodologies, ANN and analogous cure concept, percentage error criteria were used.

Tobacco curing control technology has a higher labour intensity but cannot be adjusted based on the conditions of different tobacco leaf batches, reducing the tobacco quality. Tobacco quality can be improved by using control technology which can predict the state of the tobacco curing process, accurately predict the tobacco curing state, and make timely adjustments to the curing process. The proposed model of the tobacco curing machine was trained using an Artificial Neural Network (ANN) as a model classifier.

The ANN classifier performs with 79.8% accuracy, 76.4% of precision, 100% of recall, and 86.6% of F1-Score. Meanwhile, the tobacco curing machine was trained over several deep-learning models. The State Prediction Fusion Model (SPFM) method has been implemented as a control system for tobacco curing. The SPFM improved the model training process was 97.4% of accuracy, 99.7% precision, 99.7% recall, and 99.7% F1-Score. This performance indicates that this model is very good at controlling the tobacco curing process [4].

Tobacco Net was developed based on the convolutional neural network (CNN) architecture. This model classified over 858 datasets containing dry-bulb datasets and wet-bulb datasets. Promising results show that Tobacco Net is an intelligent model for mass tobacco curing processes. The effect of different CNN structures on the prediction accuracy of the dry-bulb label and the wet-bulb label was analyzed from the perspective of computational complexity and prediction performance. Tobacco Net compares with other CNN models, such as the transfer learning and custom CNN models. Tobacco Net performs with dry-blub accuracy and wet-bulb accuracy of 99.65% and 96.83%. The proposed sequential CNN structure is more suitable for analyzing bulk tobacco curing [5].

Physics-informed neural network (PINN) to simulate the thermochemical evolution of composite materials in autoclave curing equipment. The PINN framework for modeling exothermic heat transfer in composite tool systems undergoing a full cure cycle is presented. In addition, a proposed sequential approach to intermittent PINN training is presented, which overcomes the instability in PINN training.

The network parameters are constrained in the training process by introducing a physics-based loss function. Transfer learning demonstrates improvement in PINN training and demonstrates its extension to a surrogate modeling setting by including heat transfer coefficients as input parameters [3].

Based on the literature above, this study develops a classification system for production defects in the curing process using radius shaping parameters, which are produced by installing an ultrasonic sensor in the curing machine. The datasets collected contain three labels: defect crown bare (CB), defect cracked inner liner (CIL), and normal. The proposed model is an optimized deep neural network (DNN) model by configuring the network architecture based on the neural network hyperparameters. This study created a pair of DNN models that are both predictor and classifier models. The proposed model is used to classify the radius shaping parameter in the tire curing process configuration error.

METHOD

Sensor Placement

Figure 1 shows the sensor installation on a tire curing machine. Figure 1(a) is an ultrasonic sensor installed to read the radius when the green tire curing process is carried out. This radius value in Figure 1(c) will be aligned with the pressure value, the actual radius value, and the delay value. Data on the curing machine for green tires cannot be retrieved digitally, so the installation of ultrasonic sensors combined with ESP8266 (in Figure 1(b)) utilizes to obtain the configuration values on the curing machine, which can be converted into trainable parameters. For example, the mechanism in Figure 1 extracts 1500 datasets containing three labels defect crown bare (CB), defect cracked inner liner (CIL), and Normal.

Dataset Representation

The readings of the radius, pressure and delay values can be converted as shown in Figure 2(a). The delay and rim values are shown in Figure 2(b). Figure 3 shows the retrieval of data logs containing data labels: defect crown bare (CB), defect cracked inner liner (CIL), and Normal. When the radius or pressure is too large, and the delay is too short, a crown bare (CB) ensues. The inner liner cracks (CIL) when the radius or pressure is too low, and the delay is too long.

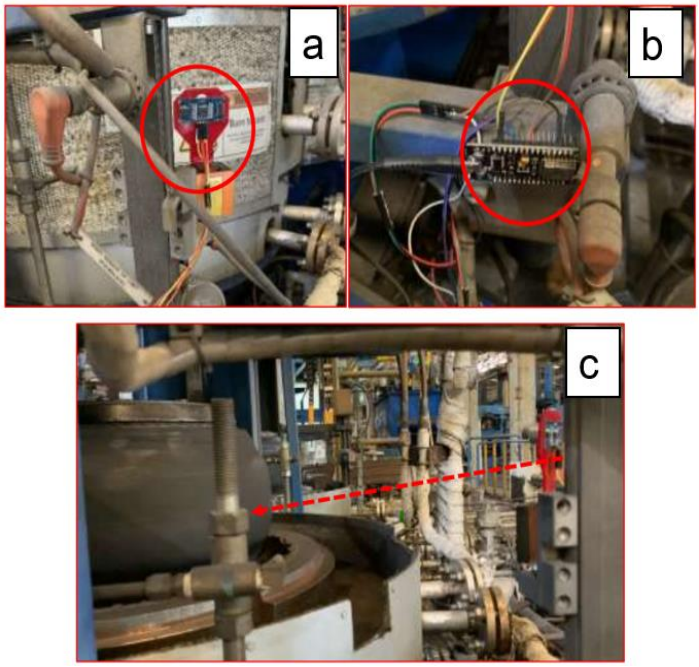
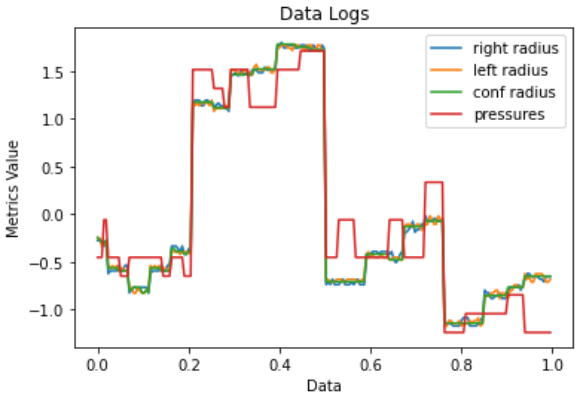
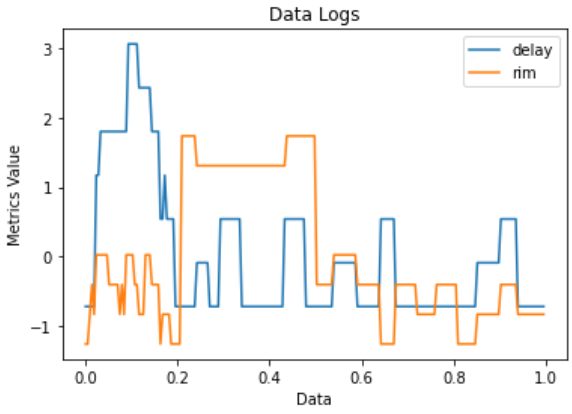


Figure 1. Sensor Installation of (a) ultrasonic sensor, (b) ESP8266 Microcontroller, and (c) sensor placement in Tire Curing Machine



(a)



(b)

Figure 2. Data Logs of (a) Radius and Pressure, (b) Delay and Rim

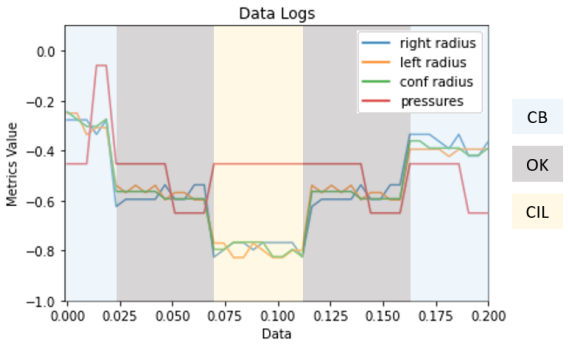


Figure 3. Data Logs and Labels

Research Flow

Figure 4 shows the research flow for defect classification of radius shaping in the tire curing process. This research's first stage is collecting curing machine datasets containing pressures and radius logs. By placing a pair of ultrasonic sensors one meter away from the curing bladder, as shown in Figure 1, Figure 2 represents the sensor placement data logs containing radius, pressure, delay, and rim values. Figure 3 represents the data label of the collected curing machine datasets. The label contains defects: crown bare (CB), cracked inner liner (CIL), and normal.

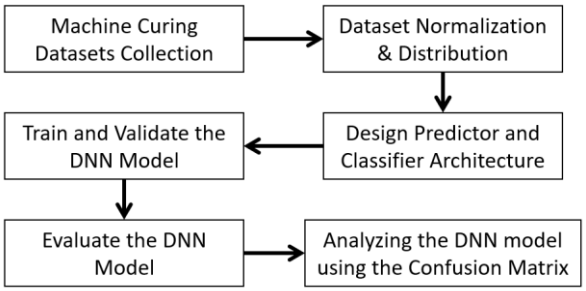


Figure 4. Research Flow

Data normalization is required to avoid overfitting or underfitting DNN models, which means the model works well in the training phase but not in the validation or testing phase. Additionally, the data distribution is important to equilibrate the model performance divided into 70% for the training and 30% for the testing phases. The dataset in the training phase is divided into 90% for the training phase and 10% for the evaluation phase. 30% of 90% in the training phase is the dataset for the validation phase.

The proposed architecture combines two neural networks as the predictor and the classifier. The predictor generates pressure, actual radius, delay, and rim values by using a pair of ultrasonic sensor values as input. The classifier generates the defect label for the green tyre curing process.

After the architecture is designed, the model needs to be trained and validated. The confusion matrix is used to analyze the model fit between the training and validation phases.

Deep Neural Network

Backpropagation is a supervised learning approach for artificial neural networks that uses gradient descent. The approach computes the gradient of the error function with respect to the neural network's weights, given an artificial neural network and an error function. It is a multilayer feedforward neural network extension of the delta rule for perceptron.

Based on Figure 2, our datasets consist of input-output pairs of size N is denoted $X = \{(\vec{x}_i, \vec{y}_i), ..., (\vec{x}_N, \vec{y}_N)\}$. While the error function in classic backpropagation is the mean squared error, denoted as [6, 7, 8].

$$E(X,\theta) = \frac{1}{2N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \quad (1)$$

Where $\hat{y}_i$ denoted as the computed output of the network on input $\vec{x}$, y_i denoted as the target value for input-output pair $(\vec{x}_i, \vec{y}_i)$. the partial derivative of the error function E with respect to a weight in the hidden layers w_{ij}^k for $1 \le k < m$ is [9],

$$\frac{\partial E}{\partial w_{ij}^k} = g'(a_j^k)o_i^{k-1}\sum_{l=1}^{r^{k+1}} w_{jl}^{k+1}\delta_l^{k+1} = \delta_j^k o_i^{k-1} \quad (2)$$

The error δ_j^k at the layer k depends on the error δ_l^{k+1} at the next layer $k+1$. The errors flow backwards to compute the first error terms based on the computed output $\hat{y} = g_o(a_l^m)$ and the target output y. The error terms for the previous layer are computed by performing a product sum (weighted by w_{jl}^{k+1}) of the error terms for the next layer and scaling it by $g'(a_j^k)$, repeated until the input layer is reached. Therefore, the partial derivatives for each input-output pair are denoted as [6][7],

$$\frac{\partial E(X,\theta)}{\partial w_{ij}^k} = \frac{1}{N}\sum_{d=1}^{N}\frac{\partial}{\partial w_{ij}^k}\left(\frac{1}{2}(\hat{y}_d - y_d)^2\right) = \frac{1}{N}\sum_{d=1}^{N}\frac{\partial E_d}{\partial w_{ij}^k} \quad (3)$$

Update the weights according to the learning rate α and total gradient $\frac{\partial E(X,\theta)}{\partial w_{ij}^k}$ formulated as [6][7],

$$\Delta w_{ij}^k = -\alpha\frac{\partial E(X,\theta)}{\partial w_{ij}^k} \quad (4)$$

Proposed Architecture

The DNN Architecture is required to be designed to represent the curing machine data logs shown in Figure 2. This paper proposed a two-DNN architecture that includes both the predictor and classifier models. The predictor model consists of 2 inputs (x_1,x_2) that represent a pair of ultrasonic sensors. The predictor model generates predicted values (y_1,…,y_4) such as pressure, actual radius, delay, and rim values. Based on Figure 3, the classifier generates the label of radius shaping defects (z_1,…,z_3), namely crown bare (CB), cracked inner liner (CIL), and Normal.

Figure 5 represents the proposed DNN architecture.

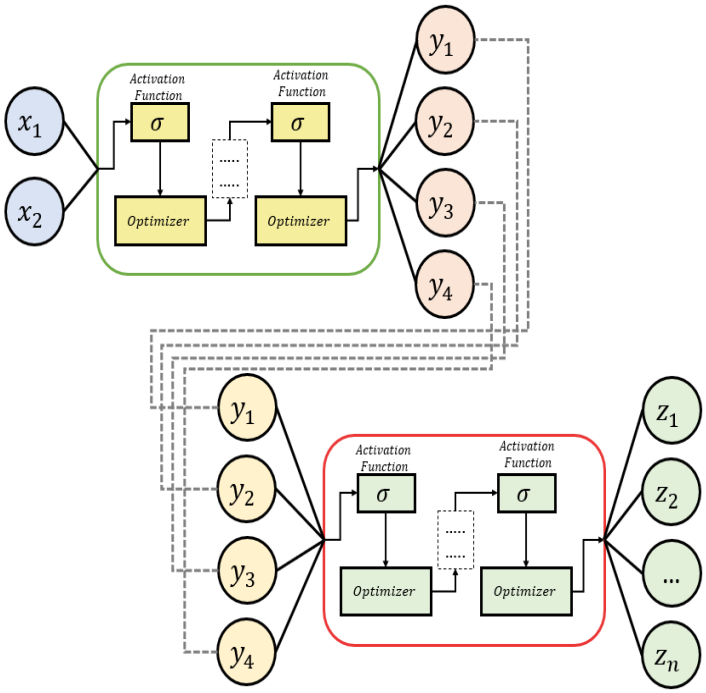


Figure 5. The Proposed DNN Architecture

Table 1 shows the DNN architecture hyperparameters [8][9]. The hyperparameters are designed to build DNN architecture to find the optimum values of several layers, several hidden layers, activation functions, an optimizer, and the learning rate [12].

Based on the Network Hyperparameters, Table 2 shows several DNN models constructed by num of layers (nL), num of neurons (nN), batch size (bS), and activation function.

Table 1. Network Hyperparameters

Hyperparameters	Configurations
Num of Layers	[4, 8, 16, 32, 64] [8]
Batch Sizes	[4, 8, 16, 32, 64] [8]
Num of Neurons	[4, 8, 16, 32, 64] [8]
Activation Functions	Sigmoid, LeakyReLu
Optimizers	Adam [10, 11, 13]
Loss Function	Cross-Entropy [14]
Learning rate	0.0001

Table 2. DNN Models

Model No	Layer Config (nL/nN/bS)	Activation Function
1	4 / 8 / 4	Sigmoid
2	4 / 8 / 4	LeakyReLu
3	4 / 16 / 4	LeakyReLu
4	8 / 16 / 4	LeakyReLu
5	8 / 32 / 4	LeakyReLu
6	16 / 32 / 4	LeakyReLu
7	8 / 32 / 8	LeakyReLu
8	8 / 32 / 16	LeakyReLu
9	8 / 32 / 32	LeakyReLu [13]
10	8 / 32 / 64	LeakyReLu [13]
11	8 / 32 / 128	LeakyReLu
12	8 / 64 / 8	LeakyReLu

RESULTS AND DISCUSSION

The DNN models were performed using Intel(R) Core i5-6300HQ CPU@2.30GHz, 16GB RAM, and Nvidia GeForce GTX 960M 4GB VRAM [13]. Table 3 compares the performance of the DNN models in terms of training accuracy, validation accuracy, training accuracy, training loss, validation loss, and execution time.

Table 3 shows the proposed model was trained using the Adam [10][11] optimizer in 200 epochs. Models 4, 5, and 6 have good model performance, resulting in training accuracy above 97%. However, the trained model needs to be validated to show that it can classify well, given the distribution of different datasets. Figure 6 shows a summary of the proposed model training process.

Table 3. DNN Models Performance

Model No	Acc (Tr/Val/Ts) (%)	Loss (Tr/Val) (%)	Exec. Time (s)
1	72.13 / 63.36 / 77.27	62.25 / 71.30	20.5
2	93.78 / 90.63 / 95.45	15.98 / 20.50	20.2
3	95.07 / 93.22 / 95.45	11.62 / 16.33	21
4	**97.58 / 95.03 / 100**	**6.21 / 13.78**	**23.5**
5	**97.88 / 95.00 / 100**	**4.93 / 20.63**	**28.4**
6	94.54 / 92.63 / 95.45	10.56 / 21.43	37.4
7	95.68 / 93.68 / 95.45	9.11 / 19.71	17.7
8	95.52 / 91.01 / 95.45	9.89 / 26.51	11.4
9	95.70 / 92.53 / 95.45	10.42 / 23.93	8.54
10	91.05 / 87.37 / 95.45	19.59 / 28.99	7.04
11	91.55 / 88.00 / 95.45	20.07 / 26.11	6.46
12	**97.07 / 96.85 / 100**	**6.44 / 7.16**	**18.2**

Figure 7 shows the DNN Model (5) with the layer configuration num of layers (nL=8), num of neurons (nN=32), batch size (bS=4), and Activation Function ("LeakyReLu").

This DNN model shows a fit representation of training and validation values, which means the trained DNN model has a good data distribution for training dan validation. Moreover, this DNN model can predict and classify the defects of the radius shaping process in data test distribution shown in Figure 8. In order to represent the performance of particular DNN models, this paper assigned the precision, recall, and F1-score, which formulated as [15, 16, 17],

$$recall(TPR) = \frac{(TP)}{(TP+FN)} \quad (6)$$

$$precision = \frac{(TP)}{(TP+FP)} \quad (7)$$

$$f1-score = 2\frac{recall \times precision}{recall + precision} \quad (8)$$

Recall (True Positive Rate/TPR) is defined as a correct predicted positive observation over all label distributions in the actual label. Precision is defined as the correct predicted observations of the total predicted positive observations. The F1-score is a weighted average of precision and recall, which takes both false positive and false negative observations into account.

Based on Figure 8, the confusion matrix in the DNN Model (5), which produces precision, recall, and F1-score, Label CB produces 94% precision, 94% recall, and 94% F1-score. Label CIL produces 94% precision, 95% recall, and 94% F1-score. Label Normal produces 93% precision, 92% recall, and 93% F1-score.

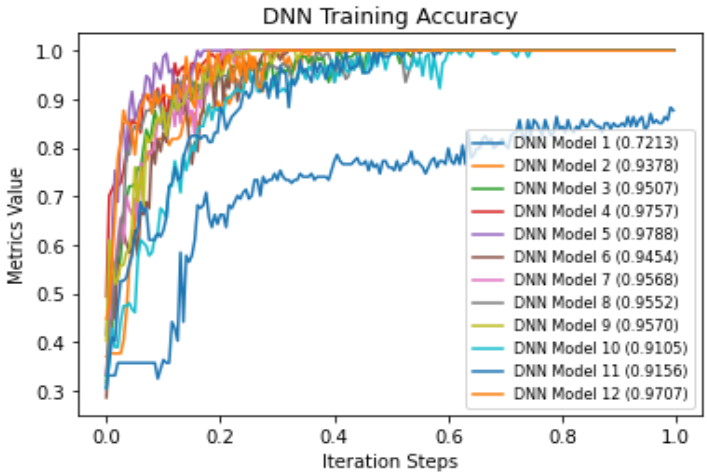


Figure 6. DNN Models Training Performance

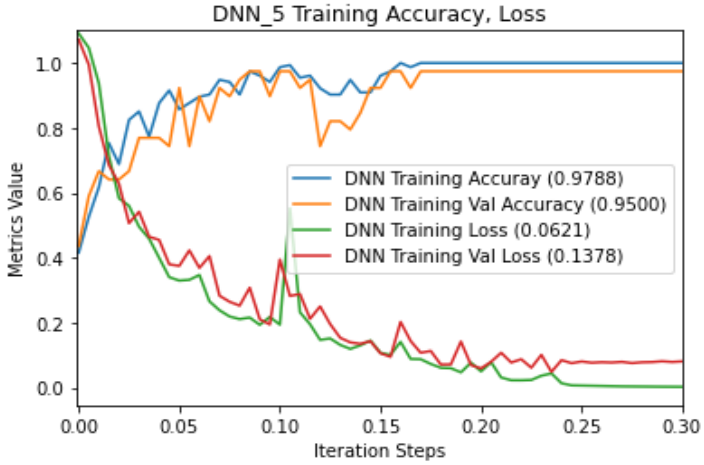


Figure 7. DNN Models (5) Training Performance

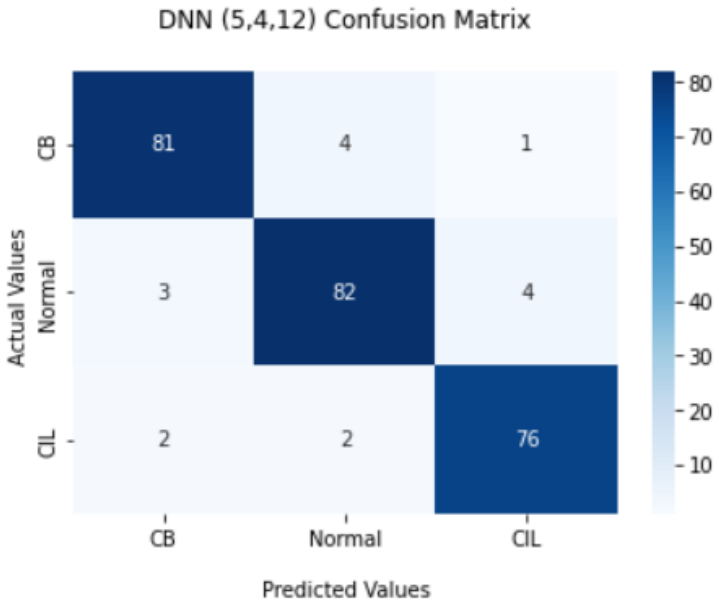


Figure 8. DNN Models (5, 4, and 12) Confusion Matrix

CONCLUSION

This research suggests utilizing a fine-tuned Deep Neural Network to identify aberrant radius shaping during the curing process. Several DNN models were investigated to develop an optimum DNN model for defects detection of radius shaping during the curing process. The proposed 12 DNN models were trained using the Adam optimizer in 200 epochs. Models 4, 5, and 6 have good model performance, resulting in training accuracy above 97%. The trained model needs to be validated to show that it can classify well, given the distribution of different datasets. The dataset distribution must be in equilibrium to generate a fit DNN model performance matrix. Model 1 has a training accuracy of 72.13%, a validation accuracy of 63.36%, and a testing accuracy of 77.27%. However, Model 1 produces a high loss of 62.25% and a validation loss of 71.30%. The fine-tuned DNN architecture has been exported for use in the curing system. The DNN was trained with a training accuracy of 97.88%, a validation accuracy of 95%, a testing accuracy of 100%, and a loss of 4.93%.

ACKNOWLEDGMENT

The author expresses gratitude to the Electrical Engineering Department and Research Center of Mercu Buana University, Jakarta, Indonesia. With this support, the authors have completed this research.

REFERENCES

[1] M. Shiva et al., “Effects of silicon carbide as a heat conductive filler in butyl rubber for bladder tire curing applications,” *Materials Today Communications*, vol. 29, ID: 102773, 2021, doi: 10.1016/j.mtcomm.2021.102773

[2] J. Wu and S. X.Yang, “Intelligent Control of Bulk Tobacco Curing Schedule Using LS-SVM- and ANFIS-Based Multi-Sensor Data Fusion Approaches,” *Sensors*, vol. 19, no. 8, pp. 1778, 2019, doi: 10.3390/s19081778

[3] D. Pavolo and D. Chikobvu, “Estimating Rubber Covered Conveyor Belting Cure Times Using Multiple Simultaneous Optimizations Ensemble,” *Operational Research in Engineering Sciences: Theory and Applications*, vol. 5, no. 1, 2022, doi: 10.31181/oresta180222016p

[4] Y. Wang and L. Qin, “Research on state prediction method of tobacco curing process based on model fusion,” *Journal of Ambient Intelligence and Humanized Computing*, no.13 pp 2951–2961, 2021, doi: 10.1007/s12652-021-03129-5.

[5] J. Wu and S. X. Yang, “Modeling of the Bulk Tobacco Flue-Curing Process Using a Deep Learning-Based Method,” *IEEE Access*, vol. 9, pp. 140424–140436, 2021, doi: 10.1109/ACCESS.2021.3119544.

[6] H. A. A. Osman and N. Z. Azlan, “Generating images for Supervised Hyperspectral Image Classification with Generative Adversarial Nets,” *Journal of Integrated and Advanced Engineering (JIAE)*, vol. 2, no. 2, pp. 107-112, 2022, doi: 10.51662/jiae.v2i2.80

[7] J. McGonagle et al., “Backpropagation,” *Brilliant Math & Science,* January 20, 2020.

https://brilliant.org/wiki/backpropagation/ (Accessed January 20, 2022).
[8] A. A. Jaber and R. Bicker, "Fault diagnosis of industrial robot bearings based on discrete wavelet transform and artificial neural network," *International Journal of Health Planning and Management*, vol. 7, no. 2, 2016, doi: 10.12691/ajme-4-1-4.
[9] Z. Iklima et al., "Self-Learning of Delta Robot Using Inverse Kinematics and Artificial Neural Network," *SINERGI* vol. 25, no. 3, pp. 237–244, 2021 doi: 10.22441/sinergi. 2021.3.001.
[10] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 2018, pp. 1–2, doi: 10.1109/IWQoS.2018. 8624183.
[11] S. Bock, J. Goppold, and M. Weiß, "An improvement of the convergence proof of the ADAM-Optimizer," *Conference Paper At Oth Clusterkonferenz,* 2018, pp. 1–5.
[12] S. Teerapittayanon, B. McDanel, and H. T. Kung, "Distributed Deep Neural Networks over the Cloud, the Edge and End Devices," *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS),* 2017, pp. 328-339, doi: 10.1109/ICDCS. 2017.226.
[13] I. Zendi, "A Microservices-based for Distributed Deep Neural Network of Delta Robot Control System," *2020 IEEE International Conference on Communication, Networks and Satellite (Comnetsat),* 2020, pp. 218-221, doi: 10.1109/Comnetsat50391.2020.9328936.
[14] R. Almodfer, S. Xiong, M. Mudhsh, and P. Duan, "Enhancing AlexNet for Arabic Handwritten words Recognition Using Incremental Dropout," *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2017, pp. 663-669, doi: 10.1109/ICTAI.2017.00106.
[15] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network," *Appl. Intell.*, vol. 51, no. 2, pp. 854–864, 2021, doi: 10.1007/s10489-020-01829-7.
[16] S. Sukegawa *et al.*, "Deep neural networks for dental implant system classification," *Biomolecules*, vol. 10, no. 7, pp. 1–13, 2020, doi: 10.3390/biom10070984.
[17] A. Thakkar and K. Chaudhari," A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions," *Expert Systems with Applications*, vol. 177, ID: 114800, 2021, doi: 10.1016/j.eswa.2021. 11