ILKOM Jurnal Ilmiah Vol. 17, No. 1, April 2025, pp. 20-26
Accredited 2nd by RISTEKBRIN No. 10/C/C3/DT.05.00/2025; E-ISSN 2548-7779 | P-ISSN 2087-1716

Research Article

Detection of Back Cover Material Defects Based on Convolutional Neural Network and TensorFlow

Sasa Ani Arnomo [a,1,*]; **Siti Fairuz Nurr Sadikan** [b,2]
[a] *Institut Teknologi Batam, The Vitka City Complex, Tiban, Kota Batam, Kepulauan Riau 29425, Indonesia*
[b] *Universiti Teknologi MARA, Cawangan Melaka, 77300, Melaka, Malaysia*
[1] *sasa@iteba.ac.id;* [2] *fairuznurr@uitm.edu.my*

Article history: Received November 07, 2024; Revised December 26, 2024; Accepted March 03, 2025; Available online April 20, 2025

Abstract

The application of deep learning using TensorFlow to improve the efficiency of material quality control is highly needed. In this case, the material quality inspection process in the form of a back cover is an important phase in the production chain. Damage resulting from industrial machinery reduces product quality. Some common problems occur when the back cover material is damaged, such as scratches, breaks, and cracks. This is caused by impacts or falls during the production process. It takes a long inspection time if done manually by workers one by one. The speed of material inspection is also an important priority for greater efficiency. An approach is proposed that applies deep learning using TensorFlow which focuses on convolutional neural networks (CNN) for image recognition and processing. It is used for image segmentation by detecting and separating objects. This method can classify the image quality of the back cover material. The main objective of this study is to produce a model that can detect material quality accurately. The highest level of accuracy was achieved at epoch 20 of 0.98. The success of this study is that it can reduce the involvement of inspectors. Thus, it can increase the efficiency of the defect detection process in the back cover material.

Keywords: CNN; Back Cover Material; Deep learning; Defect; Quality Control.

Introduction

Quality in general means making a product or service in a timely manner and suitable for use in the environment, without defects and without disappointing consumers. Quality control is a very important process to ensure that the product is ready to be used in production and plays an important role in ensuring the materials that will enter the production process [1], [2]. Material inspection according to standards is carried out during the material inspection process to detect the final quality of the product. In this case, it is very important to decide not to accept defective material from the start of production. Real-time quality control in smart manufacturing through wearables and deep learning is a significant breakthrough [3]. Combining these technologies allows companies to improve product quality, increase efficiency, reduce costs, and increase customer satisfaction. As technology advances, more innovations in this area are expected in the future. Manufacturing companies need not only industrial equipment but also industrial data science. Industrial Engineers develop and implement quality control systems to ensure that products meet established quality standards [4]. They use a variety of statistical techniques and quality control methods to identify and prevent product defects. This allows it to support production and digitalization priorities for industrial management [5], [6]. Data visualization is also needed by the industry to make it easier to see conditions [7]. The quality control (QC) department has a primary role in ensuring materials or services meet established quality standards. The quality control process in the QC department involves strict inspections so that later the goods can be marketed more widely. Meanwhile, high-quality data requires good digital infrastructure [8], [9]. Despite advances in quality control, significant challenges remain in accurately detecting defects in back cover materials. These are mainly caused by friction with other materials or the production machine.

The problem found was how to ensure that the back-cover material meets the quality standards that have been set. Damage to the back-cover material is a common problem, such as scratches, breaks, and cracks. Factors that can cause this damage include friction with other materials. Quality control requires indicators for measurement [10], [11] and is also able to identify the causes of weak-quality [12]. There is a kitchen equipment company in Turkey that controls quality in manufacturing using the multilayer perceptron algorithm for the classification of product quality levels. The result is 96% accuracy [13]. Meanwhile, there is research such as X-ray imaging defect detection in electronic circuit production being further investigated using a machine learning approach. The results obtained from recall ranged from 0.90 to 0.95 in several models that were implemented [14]. Detecting surface anomalies has been researched using

http://dx.doi.org/10.33096/ilkom.v17i1.2259.20-26

mixed monitoring. It is an architecture with two sub-networks namely the classification sub-network that learns from high pixel level and the classification sub-network that learns from weak pixel level [15].

However, an innovative approach is still needed to increase efficiency and accuracy in quality the detection of back cover material defects. This is through the implementation of deep learning using convolutional neural systems (CNN) and a combination of Tensorflow. Tensorflow is a library using the Python language for machine learning [16]. By using deep learning, systems can learn to represent complex and abstract features of information. It enables more accurate and adaptive quality detection. Industry nowadays requires Artificial Intelligence Applications [17], [18], [19]. Implementing deep learning in the context of quality detection can involve the use of demonstration neural systems, such as convolutional neural systems (CNN). This is used to obtain material quality information presented in the dataset in the form of images. Tensorflow provides powerful tools and systems to train machine learning models efficiently. By using Tensorflow, QC departments can develop systems that can learn from historical data to recognize complex quality patterns. It can then provide feedback automatically. This helps improve the quality control process by reducing human involvement and increasing accuracy in detecting nonconformities or material defects. By continuously improving through continuous learning, these systems can become more adaptive to changes in material characteristics.

Method

This research adopts an artificial intelligence approach. The research method chosen deliberately by researchers to detail the effectiveness of the application is deep learning. The algorithm has the advantage of quality accuracy [20]. The emphasis is on Convolutional neural networks (CNN) and TensorFlow in the context of detecting material quality. TensorFlow provides a powerful framework for building and training machine learning models, especially neural networks [21]–[23]. It is often used for IoT development [24]. The stages in this research are shown as **Figure 1**.

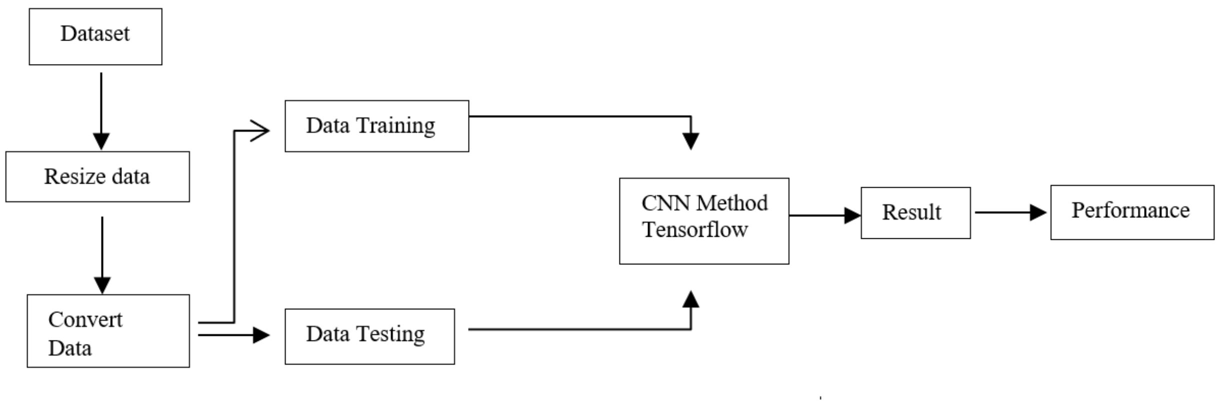

Figure 1. Research Stages

Figure 1 illustrates the sequence of research stages. The main purpose of a dataset is to provide a basis for analysis, modeling, or decision making [25]. As in the beginning, the ML model requires data preprocessing [26]. The data used in this research consists of images of the back cover material. High-quality training data is essential for training accurate models [27]. Testing data is a part of the dataset used to evaluate the performance of the trained model [28]. The data were obtained from a company's quality control department. The original size of this data reaches 800×700 pixels with a resolution of 500 dpi, consisting of 800 images of the back cover material. These images represent each type of back cover material quality. The data that has been collected will then be adjusted to size (resize) because the initial dimensions of the data are too large. The workflow of the convolutional neural network algorithm to detect the back cover material, which is pass and defective is shown **Figure 2**.

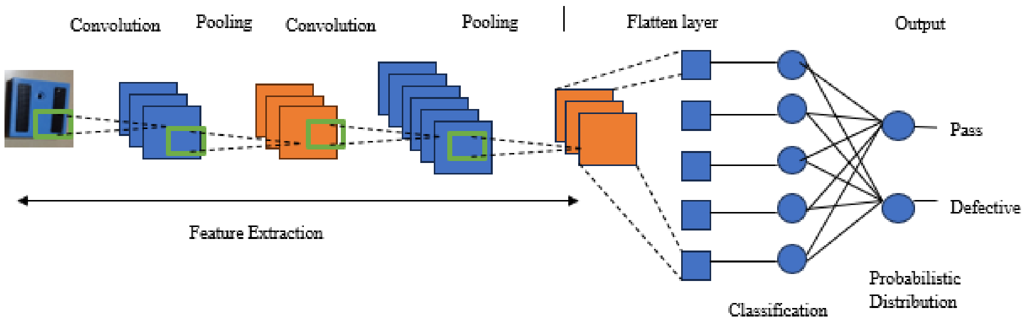

Figure 2. The workflow of the convolutional neural network algorithm

Figure 2 describes the input to output process using the convolutional neural network algorithm. Convolution is used to extract important features from input data [29]. Pooling reduces the spatial size (width and height) of the feature maps produced by the convolutional layers [30], [31]. In this case, it is hoped that it can optimize system performance when carrying out the training process. Each back cover material in the dataset has an image of the back cover material with twelve different back cover material positions. After the back cover material image has been resized, the data is then converted into XML format, then converted into CSV format, and finally in Tfrecord format. These steps are necessary so that the data can be integrated with the Tensorflow library, simplifying the training process and use in the context of model development. The purpose of labeling this image is to store image data which will be arranged in XML format with the Pascal VOC standard. This labeling step is the initial stage for compiling the dataset. The labeling process will be carried out manually on all back cover material images. **Figure 3** is an example of the labeling process which is shown as.

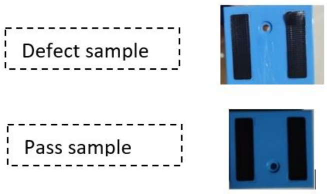

Figure 3. Image Labeling

Data preprocessing is the process of preparing training data for a deep learning model. This process is important to ensure the data used for training is of good quality and can be used by the model for more effective learning. Data preprocessing carried out includes data cleaning, data normalization, and data augmentation.

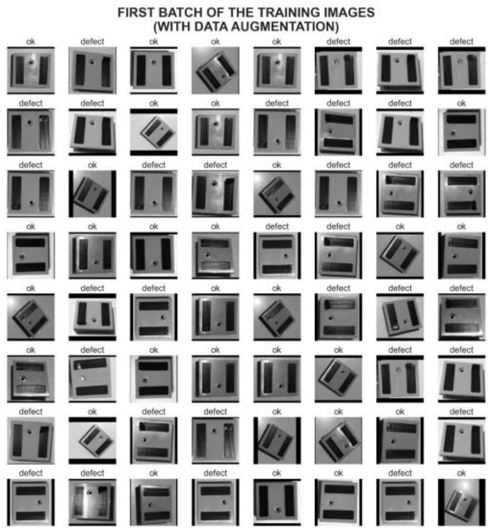

Figure 4. The Training Images with Data Augmentation

Figure 4 shows data augmentation which is carried out to increase the amount of training data and reduce overfitting. Model architecture is the structure of a deep learning model. Model architecture determines how data will be processed by the model. To detect material quality with deep learning using TensorFlow in the QC department, the appropriate model architecture is the Convolutional neural network (CNN). CNN is a model architecture designed for visual detection tasks. CNN works by learning patterns in visual data, such as edges, lines, and shapes. The choice of model architecture should be based on the complexity of the problem and the size of the dataset. For complex problems, such as material quality detection, complex model architectures are required. For large datasets, a model architecture is needed that can handle large amounts of data. The prepared dataset will be trained using convolutional neural network techniques. During this training process, the convolutional neural network method will be applied to achieve a high level of accuracy in carrying out classification. In the final phase, Tensorflow performance results will be presented, showing the level of accuracy of the dataset that has been tested.

Results and Discussion

There are a series of initial steps including the labeling process, data conversion, and creating a folder for the back cover material. The next step is to train the data that has been collected using the CNN algorithm. The formation of a network model from the CNN algorithm has a significant impact on the accuracy of the model results. In this research, the input image size used is 150×150 pixels.

1. The initial stage of the convolution process uses a 3×3 kernel with 80 filters. This step involves combining two different matrices to produce a new matrix. After the convolution stage is complete, RELU (Retrified Linear Unit) is carried out where new activations are added. The RELU function aims to change negative values to zero, eliminating negative values in the convolution result matrix. The padding value used in the convolution stage is 0, thus the size of the convolution results remains the same, namely 150×150.
2. The pooling process is a step to reduce the dimensions of the matrix using pooling operations. The pooling layer consists of filters whose matrix value sizes will alternately move in the feature maps area. The output from the pooling stage produces a new value matrix because in this study maxpooling activation is applied. The working mechanism of max pooling involves taking the maximum value based on the kernel shift.
3. The second convolution step involves redirecting the output of the first pooling stage, which starts with a 40×40 image matrix and uses 150 filters with 3×3 kernels. At this stage, the RELU activation function is also applied.
4. The next step is to carry out the second pooling, the process stages of which are identical to the first pooling. The difference between them is the final dimensions of the output matrix. The second pooling produces an output of size 75×75.
5. After that, the next step is to enter the flattened or fully connected stage, which consists of only one hidden layer. In the flattening stage, the output of the pooling layer is converted into a vector. Next, the flattening stage will involve the image detection process.
6. The final step involves applying a softmax activation function, which is commonly used in the context of the detection of multiclass linear discriminant analysis and multinomial logistic regression.

The application of the CNN network model above focuses on the training stage. From the training process, the CNN model structure is obtained which is shown below.

Table 1. Structure of CNN Model Results

No	Nama	*Size*	Parameter
0	Input	150x150x3	0
1	conv2d_10 (Conv2D)	(None, 150, 150, 32)	320
2	*max_pooling*2d_10 (*Max*Pooli	(None, 75, 75, 32)	0
3	conv2d_11 (Conv2D)	conv2d_11 (Conv2D)	18496
4	*max_pooling*2d_11 (*MaxPooling*)	(None, 19, 19, 64)	0
5	flatten_5 (Flatten)	(None, 23104)	0
6	dense_10 (Dense)	(None, 128)	2957440
7	dense_11 (Dense)	(None, 1)	(None, 1)
	Total		2976385 (11.35 MB)

Table 1 is the result of training by applying the CNN network model. The total parameters resulting from this modeling reached 2976385 parameters. Determining model parameters is intended to identify optimal parameters in the CNN (Convolutional neural network) model. The parameters to be tested include a number of varying values from the epoch, a number of training data, and learning rate values. This experiment aims to compare the performance between these models, while still paying attention to the optimal values of each parameter. Epoch is a phase where the entire dataset has gone through the training process in the Neural Network (NN) until it returns to the beginning in one round. In the context of deep learning, a single epoch has a significant impact on training, because the entire dataset participates in the training phase, which can take quite a long time. To simplify and speed up training, the dataset is usually divided into batches of a certain size. It is known as Batch Size. Researchers usually determine the Batch Size value by considering the total number of existing datasets. Following are the accuracy results obtained by running several different epochs using Tensorflow.

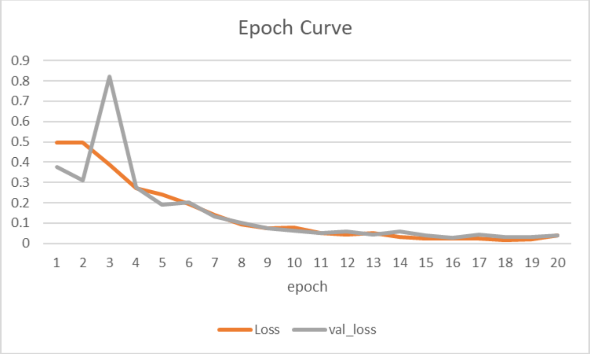

Figure 5. Loss and Val Loss in the Epoch

Figure 5 shows that there are 20 epochs, the loss value from training was reduced from 0.4981 to 0.0383 and the loss value from validation was reduced from 0.3767 to 0.0391. Meanwhile, the accuracy values for both training and validation increased to 0.9891 and 0.9887. The difference is not too far, indicating a low possibility of overfitting. To test this impact, researchers will divide the training dataset into three groups of 200, 400, and 800 data each. From each group of training data. The following are the results of dividing the three amounts of training data:

Table 2. Results of Dividing Three Amounts of Data

Data *training*	Accuracy Validation	Lost Validation	Time (s)
200	90%	0.9256	95s
400	88%	0.2754	109s
800	96%	0.10012	96s

Table 2 concludes that the accuracy rate obtained for various training data ranges from 88-96%. The split ratio used is 0.2 from the dataset. The percentage results achieved show that the Convolutional neural network (CNN) model that has been developed has succeeded in identifying materials effectively. Therefore, the more training data used, the greater the level of accuracy of the results. This is because the system can understand image patterns better. Thus, the accuracy of the detection process will be further improved. Learning rate is a parameter applied in the gradient descent algorithm. It is a method used to find local minimum values in parametric functions. In this research, researchers used three learning rate values, namely 0.1, 0.6, and 1. These learning rate values can be directly implemented in TensorFlow by setting appropriate learning rate parameters. Determining the learning rate value has a significant impact on the accuracy obtained.

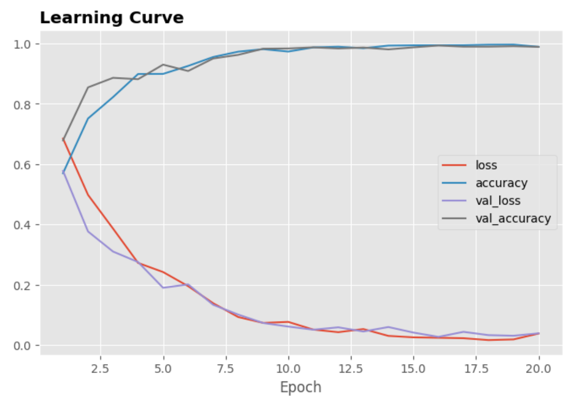

Figure 6. Learning Curve

Figure 6 shows the accuracy results for each learning rate value that has been applied. The gap between the training and test data curve is not large. It is not stated as overfitting. The learning curve is not flat on either the training or test data. This indicates an optimal curve that does not experience underfitting. The model is stated as being able to capture patterns in the data. The learning curve reflects the model's success in detecting and classifying materials that meet certain quality standards. By applying deep learning technology, the model can automatically process information from the training dataset and identify characteristics. This indicates that a material can be categorized as the back-cover material which is pass and defective.

Conclusion

Based on the analysis and discussion that has been carried out in this research, it can be concluded that the Convolutional Neural Network (CNN) model that has been developed shows satisfactory accuracy results. The highest level of accuracy was achieved at epoch 20, namely 98%. This shows that the closer to the epoch value of 20, the level of accuracy in testing will increase. However, it should be noted that the accuracy value can decrease if the epoch value exceeds 20 because the number of excessive epochs is not proportional to the number of existing datasets. The optimal learning rate value to achieve a high level of accuracy is 0.1 with an accuracy of up to 90%. Using a smaller learning rate value tends to produce a lower loss value. However, it should be noted that the smaller the learning rate value, the higher the time required because the system will try to reduce the loss value that occurs. Meanwhile, using different amounts of training data, accuracy values are between 88% and 96%.

Acknowledgement

Thanks, are expressed to Putera Batam University and Universiti Teknologi MARA for supporting the completion of the research. We also express our gratitude to the quality control department which was willing to provide the back-cover material as research material.

References

[1] S. I. Kalogiannidis, “The Effects of Total Quality Management Practices and Marketing on Performance of SMEs. A Case of Selected Manufacturing Industries, Greece,” *Bus. Manag. Strateg.*, vol. 12, no. 1, p. 48, 2021.

[2] R. Nicolas *et al.*, “Assessing the fresh properties of printable cement-based materials: High potential tests for quality control,” *Cem. Concr. Res.*, vol. 158, no. 2022, p. 106836, 2022.

[3] I. M. Sarivan *et al.*, “Enabling real-time quality inspection in smart manufacturing through wearable smart devices and deep learning,” *Procedia Manuf.*, vol. 51, pp. 373–380, 2020.

[4] J. Deuse, N. West, and M. Syberg, “Rediscovering Scientific Management - The Evolution from Industrial Engineering to Industrial Data Science,” *Int. J. Prod. Manag. Eng.*, vol. 10, no. 1, pp. 1–12, 2022.

[5] G. Dutta, R. Kumar, R. Sindhwani, and R. K. Singh, “Digitalization priorities of quality control processes for SMEs: a conceptual study in perspective of Industry 4.0 adoption,” *J. Intell. Manuf.*, vol. 32, no. 6, pp. 1679–1698, 2021.

[6] M. C. Magnanini, M. Colledani, and D. Caputo, “Reference architecture for the industrial implementation of zero-defect manufacturing strategies,” *Procedia CIRP*, vol. 93, no. March, pp. 646–651, 2020.

[7] L. Allen, J. Atkinson, D. Jayasundara, J. Cordiner, and P. Z. Moghadam, “Data visualization for Industry 4.0: A stepping-stone toward a digital future, bridging the gap between academia and industry,” *Patterns*, vol. 2, no. 5, p. 100266, 2021.

[8] A. Ghadge, M. Er Kara, H. Moradlou, and M. Goswami, “The impact of Industry 4.0 implementation on supply chains,” *J. Manuf. Technol. Manag.*, vol. 31, no. 4, pp. 669–686, 2020.

[9] N. N. Misra, Y. Dixit, A. Al-Mallahi, M. S. Bhullar, R. Upadhyay, and A. Martynenko, “IoT, Big Data, and Artificial Intelligence in Agriculture and Food Industry,” *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6305–6324, 2022.

[10] M. R. Baqueta, A. Coqueiro, P. H. Març̧o, and P. Valderrama, “Quality Control Parameters in the Roasted Coffee Industry: a Proposal by Using MicroNIR Spectroscopy and Multivariate Calibration,” *Food Anal. Methods*, vol. 13, no. 1, pp. 50–60, 2020.

[11] N. A. R. Negri, G. M. R. Borille, and V. A. Falcão, “Acceptance of biometric technology in airport check-in,” *J. Air Transp. Manag.*, vol. 81, no. August, p. 101720, 2019.

[12] M. Bhargava and S. Gaur, “Process Improvement Using Six-Sigma (DMAIC Process) in Bearing Manufacturing Industry: A Case Study,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1017, no. 1, 2021.

[13] G. Sariyer, S. K. Mangla, Y. Kazancoglu, C. Ocal Tasar, and S. Luthra, “Data analytics for quality management in Industry 4.0 from a MSME perspective,” *Ann. Oper. Res.*, no. 0123456789, 2021.

[14] Q. Zhang *et al.*, “Deep learning based solder joint defect detection on industrial printed circuit board X-ray images,” *Complex Intell. Syst.*, vol. 8, no. 2, pp. 1525–1537, 2022.

[15] J. Božič, D. Tabernik, and D. Skočaj, “Mixed supervision for surface-defect detection: From weakly to fully supervised learning,” *Comput. Ind.*, vol. 129, 2021.

[16] E. Handoyo, Y. A. A. Soetrisno, E. W. Sinuraya, D. Denis, I. Santoso, and H. M. Irsyad, “Designing a Machine Learning Model Using Tensorflow in the Cato Application to Recognize Human Body Members,” *J. Sains Teknol.*, vol. 5, no. 2, pp. 285–294, 2022.

[17] S. A. Arnomo, Yulia, and Ukas, “Building The Prediction of Sales Evaluation on Exponential Smoothing using The OutSystems Platform,” *Ilk. J. Ilm.*, vol. 15, no. 2, pp. 222–228, 2023.

[18] R. S. Peres, X. Jia, J. Lee, K. Sun, A. W. Colombo, and J. Barata, “Industrial Artificial Intelligence in Industry 4.0 -Systematic Review, Challenges and Outlook,” *IEEE Access*, pp. 220121–220139, 2020.

[19] D. B. Rathore, “Integration of Artificial Intelligence& It’s Practices in Apparel Industry,” *Int. J. New Media Stud.*, vol. 10, no. 01, pp. 25–37, 2023.

[20] C. Wu *et al.*, “SEMG Measurement Position and Feature Optimization Strategy for Gesture Recognition Based

on ANOVA and Neural Networks," *IEEE Access.*, vol. 8, pp. 56290–56299, 2020.

[21] I. Kilic and G. Aydin, "Traffic Sign Detection and Recognition Using TensorFlow' s Object Detection API with A New Benchmark Dataset," *2020 Int. Conf. Electr. Eng. ICEE 2020*, 2020.

[22] A. R. Visalatchi, T. Navasri, P. Ranjanipriya, and R. Yogamathi, "Intelligent Vision with TensorFlow using Neural Network Algorithms," *Proc. 4th Int. Conf. Comput. Methodol. Commun. ICCMC 2020*, no. Iccmc, pp. 944–948, 2020.

[23] Y. Ju, X. Wang, and X. Chen, "Research on OMR recognition based on convolutional neural network tensorflow platform," *Proc. - 2019 11th Int. Conf. Meas. Technol. Mechatronics Autom. ICMTMA 2019*, pp. 688–691, 2019.

[24] T. J. Sheng *et al.*, "An Internet of Things Based Smart Waste Management System Using LoRa and Tensorflow Deep Learning Model," *IEEE Access*, vol. 8, pp. 148793–148811, 2020.

[25] B. A. Alpatov, N. Y. Shubin, and A. V. Yakovlev, "TensorFlow for Generating Edge Detection Dataset," *2020 9th Mediterr. Conf. Embed. Comput. MECO 2020*, pp. 8–11, 2020.

[26] V. Oujezsky, A. Tomasov, M. Holik, V. Skorpil, T. Horvath, and M. Jurcik, "GPON Traffic Analysis with TensorFlow," *2020 43rd Int. Conf. Telecommun. Signal Process. TSP 2020*, pp. 69–72, 2020.

[27] D. Bajpai and L. He, "Custom Dataset Creation with Tensorflow Framework and Image Processing for Google T-Rex," *Proc. - 2020 12th Int. Conf. Comput. Intell. Commun. Networks, CICN 2020*, pp. 45–48, 2020.

[28] O. Kembuan, G. Caren Rorimpandey, and S. Milian Tompunu Tengker, "Convolutional Neural Network (CNN) for Image Classification of Indonesia Sign Language Using Tensorflow," *2020 2nd Int. Conf. Cybern. Intell. Syst. ICORIS 2020*, no. 26, 2020.

[29] H. Y. Suen, K. E. Hung, and C. L. Lin, "TensorFlow-Based Automatic Personality Recognition Used in Asynchronous Video Interviews," *IEEE Access*, vol. 7, no. c, pp. 61018–61023, 2019.

[30] K. Wongsuphasawat *et al.*, "Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 1–12, 2018.

[31] Di. Someshwar, D. Bhanushali, V. Chaudhari, and S. Nadkarni, "Implementation of Virtual Assistant with Sign Language using Deep Learning and TensorFlow," *Proc. 2nd Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2020*, pp. 595–600, 2020.