International Journal of Scientific Research and Modern Education (IJSRME)
International Peer Reviewed - Refereed Research Journal, Website: www.rdmodernresearch.com
Impact Factor: 7.137, ISSN (Online): 2455 - 5630, Volume 9, Issue 2, July - December, 2024

CLASSIFICATION OF DEFECTIVE PRODUCT FOR SMART FACTORY THROUGH DEEP LEARNING METHOD

K. Vinayakan* & A. Dinesh Kumar**
* PG & Research Department of Computer Science, Khadir Mohideen College (Affiliated to Bharathidasan University), Adirampattinam, Tamil Nadu
** PG & Research Department of Mathematics, Khadir Mohideen College (Affiliated to Bharathidasan University), Adirampattinam, Tamil Nadu

Cite This Article: K. Vinayakan & A. Dinesh Kumar, "Classification of Defective Product for Smart Factory through Deep Learning Method", International Journal of Scientific Research and Modern Education, Volume 9, Issue 2, July - December, Page Number 10-15, 2024.

DOI: https://doi.org/10.5281/zenodo.12807405

Abstract:

Smart manufacturing entails combining modern data analytics with physical science to improve decision-making and system performance. This decade, researchers have been paying close attention to smart factories. The smart factory utilizes IoT technology to intelligently monitor automated manufacturing and defect detection equipment, substantially enhancing production quality and efficiency. The smart factory uses production equipment, functional testing equipment, and fault detection equipment. Therefore, the spared defective product from subsequent operations could potentially reduce and increase the yield of the shipped final product. This high accuracy rate demonstrates the effectiveness of using advanced technology like CNNs in quality control processes. Implementing this system in the smart factory can significantly improve overall production efficiency and reduce costs associated with defective products.

Key Words: Smart Factory, Defect Detection, Image Processing, Convolution Neural Network (CNN), VGG16, Classification

1. Introduction:

In recent years, the world has embraced the Fourth Industrial Revolution, or Industry 4.0, characterized by boosted productivity and increased efficiency through modern technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), robotics, Cloud Computing (CC), sensors, and integrated systems [1]. Industry 4.0 integrates large-scale IoT systems and machine-to-machine (M2M) communications to improve automation, communication, and self-monitoring, reducing the need for human intervention. A comprehensive review of automated vision-based defect detection approaches for materials like ceramics, textiles, and metals is presented in [2].

Smart manufacturing is a modern production paradigm where machines are networked, sensor-monitored, and managed by advanced computer intelligence to improve system efficiency, product quality, and sustainability while reducing costs. Key supporting technologies include IoT, CC, and CPS [3-6]. Accurate product classification is critical for quality control in manufacturing, requiring robust, real-time performance despite noisy factory conditions, such as lighting variations or shadows. The diversity of products also limits the use of feature-based techniques.

Deep learning algorithms have excelled in computer vision applications such as object detection and image classification, showing enormous potential for manufacturing [7]. For instance, a multi-task CNN combines wire defect region identification with faulty product classification [8]. Other CNN-based quality inspection tasks include monitoring PCBs [8, 9], metal surfaces [10], bottled wine, casting products, semiconductor fabrication, LED cup apertures, mobile phone screens, display panel cover glass, bearings, optical film, and leather defects. This research focuses on defect classification using a deep learning model (CNN) with VGG16 to improve classification accuracy.

The article is organized as follows:

- Section 2 discusses defect classification using CNN and other defect identification techniques.
- Section 3 explores the VGG16 architecture with CNN for improved training models and illustrates the use of the SGD optimizer for high learning rates.
- Section 4 compares defect detection classification using a CNN with VGG16 to other CNNs and CNNs with ResNet, looking at things like F1-score, accuracy, precision, and recall.
- Section 5 concludes that CNN with VGG16 achieves better training model accuracy, with an evaluation showing a high accuracy of 97%.

2. Literature Review:

Ren et al. analyzed Electrical Resistance Tomography (ERT) pictures using a color histogram-based evaluation approach [11]. W. Song et al. developed a classification methodology for wood surface defects, focusing on junction-type defects, using percentage color histograms and texture features [12]. Prasitmeeboon presented a two-step process for particle board defects using threshold technology, smoothing, SVM, and color histogram features [13]. J. Ren suggested using a deep neural network with FCN and an autoencoder to tell the difference between keyboard light leakage defects and dust. This cut the number of false positives from 6.27% to 2.37% [14]. C. Sampedro proposed an automatic identification and diagnosis approach for insulator strings with segmentation and diagnostic components [15].

Baumgartl et al. created a CNN model for LPBF devices using off-axis in-situ imaging and thermography. This model was 96% accurate at finding flaws in the devices using thermography, but it could only find flaws like splatter and delamination in metal LPBF [16]. It was J. Jayanthi et al. who used Mask R-CNN and Hough Circle Transform (HT) to accurately locate the iris region, beating models such as UniNet V2, AlexNet, Inception, ResNet, DenseNet, and VGGNet [17]. They were able to achieve 99.14% recognition accuracy.

This section discusses various approaches to defect detection in different types of factories, along with analyses from several researchers on machine learning methods and CNN in product defect detection. Research indicates that defect detection through image processing using CNNs can achieve superior accuracy. Therefore, this review advocates for the use of CNN with VGG16, demonstrating its high accuracy in classifying defective products [18-21].

3. Research Methodology:

This research aims to identify and classify defective products from raw materials in manufacturing industries. This study employs real-time datasets gathered from network-captured images, pre-processed using Region of Interest (RoI) extraction. RoI extraction, used for image segmentation, enhances preprocessing performance. Following ROI extraction, we normalize the dataset by dividing each pixel value by 255, thereby representing pixels as values between 0 and 1. This normalization helps to improve model learning. We then split the dataset into training and testing sets.

Normalization is a fundamental process for all neural networks. CNNs, which operate based on deep Forward Neural Networks (FNN), learn invariant feature hierarchies. The distinctive attributes and invariants in CNNs represent consistent transformations applied across various model locations. For training, we use the CNN model with VGG16 architecture, aiming to achieve a well-performing pre-trained model for accurately identifying defective products in a smart manufacturing setting.

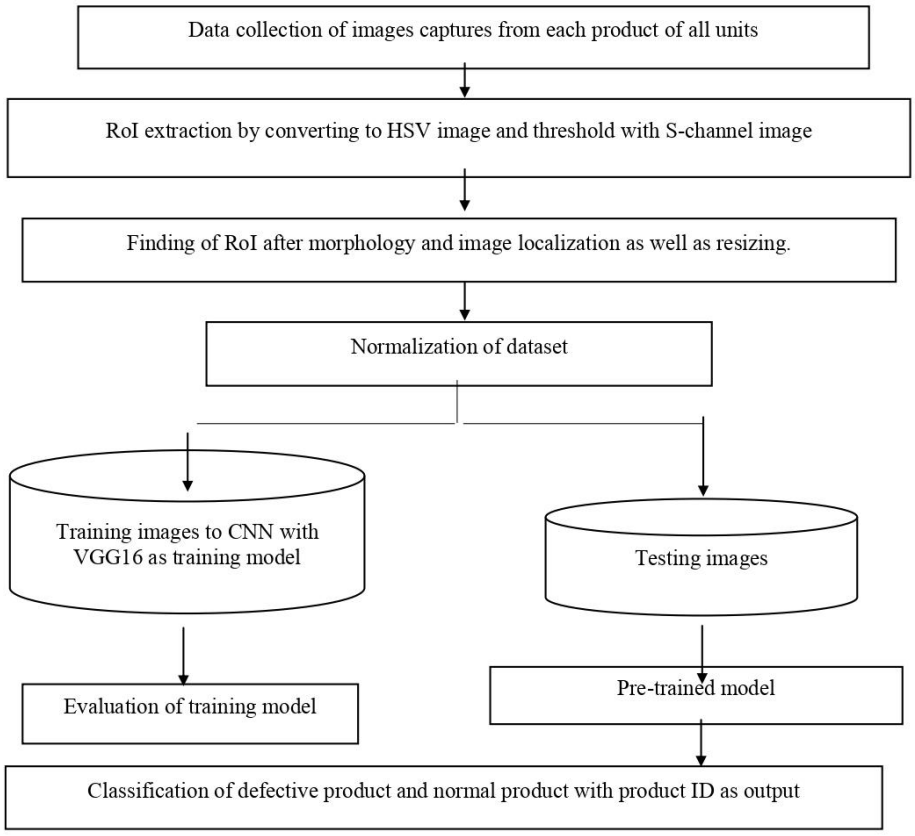

Figure 1: Work flow for the proposed classification method for defective products

3.1 Data Collection:

This study used Peking University's intelligent robot laboratory and a PCB defect dataset with 1245 photos and four different types of defects: false copper, mouse bite, missing hole, and open circuit. To detect and classify defective products, we fed the various defect images into the normal image of the PCB dataset to detect and classify them as defective products. Figure 2 presents a categorized picture dataset, which groups various defect types into distinct groups.

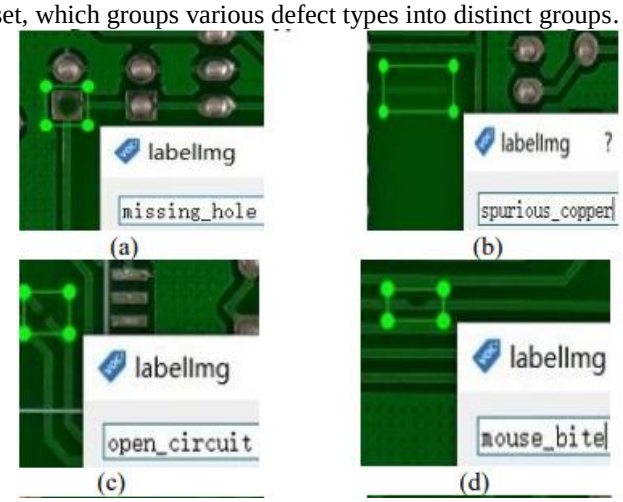

Figure 2 displays input images of various defects on the PCB. The defects on the PCB are as follows: (a) missing hole; (b) spurious copper; (c) open circuit; and (d) mouse bite

Based on the measurements, the callout box average size for the defect available in the data set is about 8×8 pixels, as shown in figure 2. The program allows for the rotation of the images to a specific angle, enhancing the model's robustness and confirming detection accuracy, even if the rotation occurs during the detection process.

3.2 Data Preprocessing:

The casting images capture using special arrangements to ensure stable lighting conditions. The RoI extraction from the captured image is based on saturation channel details and the Saturation Value (HSV), utilizing the fact that the PCB color level differs from the background. The first step involves converting the Red Green Blue (RGB) image to HSV color space using the following general equations: Let R, G, B ∈ [0, 1], the maximum region as MAX = max(R, G, B), and the minimum region as MIN = min(R, G, B).

We express the equations for H, S, and V as follows:

$$H: \begin{cases} 0, \ if\ R = B = G \\ 60° \times \left(0 + \frac{B-G}{MAX-MIN}\right), if\ MAX = R \\ 60° \times \left(2 + \frac{G-R}{MAX-MIN}\right), if\ MAX = B \\ 60° \times \left(4 + \frac{R-B}{MAX-MIN}\right), if\ MAX = G \end{cases} \quad (1)$$

$$S: \begin{cases} 0, \ if\ R = B = G \\ \frac{MAX-MIN}{MAX}, else \end{cases} \quad (2)$$

$$V := MAX \quad (3)$$

The HSV threshold selection is based on the HSV color spacing's pixel range values with respect to each histogram channel. We subsequently apply morphological operations to remove noise from the threshold images. Thus, the RoI extraction focuses on the region of the PCB, which is then located and cropped from the original images.

3.3 CNN uses VGG16 to Classify Defective Products:

The research focuses on classifying the retrieved ROI picture. The defective product classification training method involves two major steps:

- Training CNN with VGG16
- Optimizing and Retraining the Network

Step 1: Training CNN with VGG16

Embedded systems deploy CNN methods to achieve better performance. Training CNNs from scratch requires a dataset with many labelled entries, which is time-consuming and computationally expensive. We introduce Transfer Learning (TL) to address these issues. TL involves re-training a CNN model on a new, small dataset by reprocessing its feature extractor portion. It uses an existing large dataset, such as the ImageNet dataset, and re-trains only the classification functionality, saving training time. TL fine-tunes the pre-trained model weights for classifying a new dataset, significantly improving learning performance without extensive data labeling efforts.

The TL approach has achieved advanced results on many datasets with less training time. We prefer the proposed model, VGG16, over AlexNet due to its deeper architecture and smaller kernel sizes, making it more effective for this application.

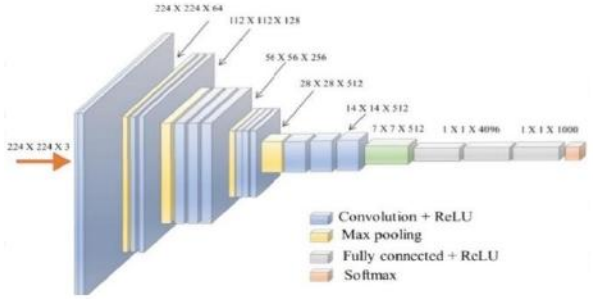

Figure 3: VGG16 architecture with CNN

VGG16 has improved upon AlexNet by offering a deeper learning model with more layers. The proposed VGG16 with CNN model addresses earlier model inadequacies and focuses on achieving better accuracy and efficiency. This research has implemented six variants with similar architectures, ranging from layer numbers 11 to 19. The proposed VGG16 architecture consists of 16 layers: 13 convolutional layers with max pooling layers and 3 fully connected layers, as shown in Figure 3.

VGG16 vs. AlexNet: Key Advantages

The VGG16 architecture's main advantage over AlexNet is that smaller filters are placed on top of each other rather than one large filter. The receptive fields of convolutional filters closely resemble this approach. Through a 7x7 kernel, each pixel receives information from the previous layer's 49 pixels. Higher receptive fields can capture more patterns over a large area, whereas smaller receptive fields might miss these details. Therefore, VGG16's architecture, which includes convolutional layers, max pooling layers, and fully connected layers, is more effective. The model uses 3x3 and 2x2 convolutional filters.

The feature extractor module's backbone is made up of adjustable weights and pre-trained frameworks, which can acquire certain layers or all layers from these networks, forms the backbone of the feature extractor module. This method, known as Transfer Learning (TL), speeds up deep learning models by eliminating the need to train large models from scratch.

Step 2: Optimize and Retrain the Network

Tensor RT-based applications perform 40 times faster than CPU-only platforms during inference. Using TensorRT, researchers can focus on generating new AI-powered applications without performance tuning for inference deployment. For real-time applications, executing pre-trained models with Tensor RT on the PCB helps convert pre-trained models to the Open Neural Network Exchange (ONNX) format.

This research uses a minibatch approach, drawing consistently from the training set. The minibatch size is generally small, ranging from one to a few hundred instances. The Stochastic Gradient Descent (SGD) algorithm performs better as it trains, reaching the local minimum in a reasonable amount of time. The learning rate, represented by iteration k, is a significant parameter for SGD. It needs to decrease over time.

Furthermore, the pre-trained model helps classify images through its loss functions and learning rates, improving the classifier model's performance. Thus, the classifier CNN with the VGG16 model achieves better accuracy in identifying and classifying defective products from normal ones, including the manufacturing shop ID.

4. Result and Discussion:

We conducted this experimental research using a high-performance server, configured with an Intel Core i7 DMI2 CPU, 16GB RAM, and a Quadro K600 GPU. Ubuntu 16.04.6 LTS served as the operating system, crucial for the GPU's execution during the training of the dataset images. The optimizer and loss function utilized in this proposed model were Stochastic Gradient Descent (SGD) and cross-entropy, respectively. We set the learning rate at 0.01, and the training epoch at 50 for improved iteration.

The PCB product data consisted of 3175 augmented, 300x300-pixel RGB images used for training and testing the system. These 3175 images included both normal and defective product images, as shown in Table 1.

Product Dataset	Product Class Status	Image Count	Training Images		Testing Images
			Training	Validating	
PCB Dataset	Normal	1937	1317	232	387
	Defective	1238	842	149	248

Table 1: Dataset distributions for training, validation, and testing

Additionally, the study used original and cropped electrical wire datasets to assess the impact of applying the ROI extraction step. Instances where the ROI extraction process failed led to a reduction in the final number of cropped images compared to the original electrical wire images.

We applied an 80% split for training (70% for actual training and 10% for validation) to all datasets, reserving the remaining 20% for testing. We provide detailed distributions for each dataset, detailing the allocation of data for training, validation, and testing, to ensure a comprehensive evaluation and validation of the developed models.

Figure 4 depicts the training model's accuracy trajectory, showing an initial increase starting around epoch two and reaching 94.8% by the tenth epoch. There is a slight fluctuation between epochs 10 and 30, followed by stabilization at 96.2% accuracy until the 50th epoch. In contrast, the validation model starts with lower accuracy and exhibits variation for the first 10 epochs. Subsequently, it stabilizes around 97.4% accuracy from epoch 10 to epoch 50. This shows how well the optimizer and learning rate work to lower the loss function and improve transfer learning (TL) by using the information stored in a model that has already been trained.

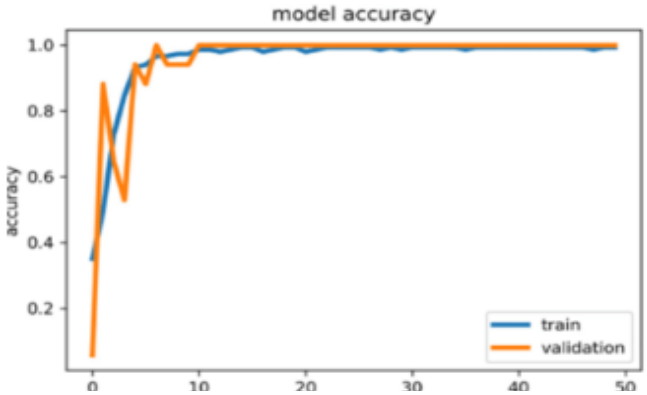

Figure 4: Model accuracy for training and validation

Figure 5 illustrates that the train model's loss begins with 1.56 at 2 epochs, and the losses start decreasing as the epoch increases until 40 epochs. The slight fluctuation occurred from 40 to 50 epochs and became steady with losses less than 0.16 until 50 epochs. In the validation model, the loss begins at 1.52 and fluctuates until 30 epochs, but it gets steady from 30 to 50 epochs with a loss less than 0.1. This shows that learning

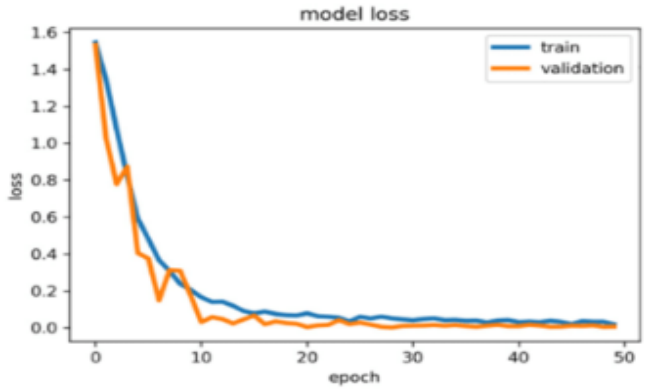

Figure 5: Model loss for training and validation

The rate and optimizer have reduced the loss function and improved the TL through better knowledge transformation using a pre-trained model.

S.No	Algorithm	Confusion Matrix Values			
		True Positive (TP)	True Negative (TN)	False Positive (FP)	False Negative (FN)
1	CNN	337	249	10	39
2	CNN-VGG16	391	225	6	13
3	CNN-ResNet	358	241	8	28

Table 2: Evaluates testing images using a confusion matrix for various algorithms

In this study employed accuracy, precision, recall, and the F1 score as assessment measures for the proposed CNN with VGG16 for product categorization. We computed these four-assessment metrics using True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) outcomes from a confusion matrix between the forecast and reality, as shown in Table 2. The ratio of correct predictions (TP+TN) to the total number of predictions (TP+TN+FP+FN) is known as accuracy. The number of accurately predicted samples (TP) divided by the number of positive samples anticipated (TP+FP) yields precision. The number of properly predicted (TP) samples divided by all actual positive samples (TP+FN) yields recall. The F1 score is defined as the weighted average of accuracy and recall. Defect classification in a product is a binary problem. Therefore, this study adopts a more reliable statistical rate evaluation metric that takes into account both positive and negative elements.

Figure 6 illustrates that CNN with VGG16 has a high accuracy when compared to existing CNN and CNN-ResNet. The performance of the proposed CNN-VGG16 is 97%, indicating that TL is even better for large datasets when compared with CNN and CNN-ResNet.

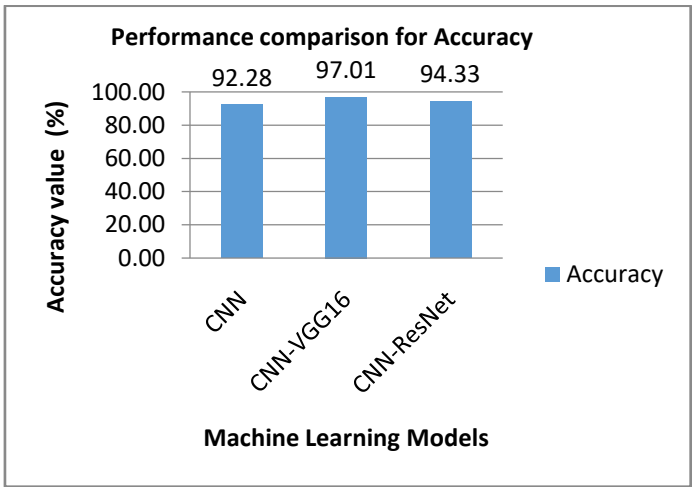

Figure 6: Comparison of accuracy performance for various deep learning models

Figure 7 illustrates the precision, recall, and f1-score performance that describes the proposed model CNN-VGG16. It has high scores in all precision, recall, and f1-scores of 0.98, 0.97, and 0.98, respectively. The higher scores of CNN-VGG16 can be attributed to its improved pre-trained model, which utilizes the SGD optimizer and a cross-entropy loss function. The positive score is high in both actual and prediction, which aids in precision and recall, thereby improving

5. Conclusion:

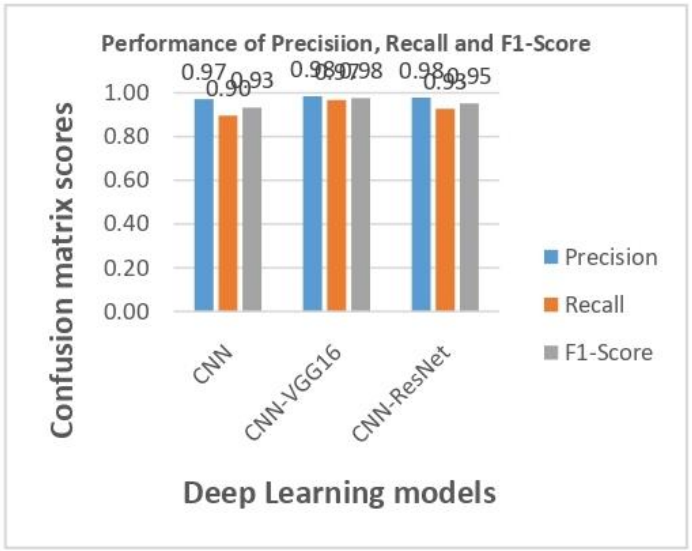

Figure 7: Comparison of precision, recall, and f1-score performance for various deep learning models

This article introduces a CNN-VGG16-based defective product classification method for smart factories. The proposed system focused on layers that assist in capturing large pixels, while the SGD optimizer involved generating a better learning rate for PCB production. To re-train CCN-VGG16 for PCB product dataset ROI extraction based on S-channel data, this pre-training model helps build a better TL model. Each pre-trained model is deployed to determine the defective PCB product with a high accuracy of 97.3% in the training model. Moreover, the proposed CNN-VGG16 is evaluated using various deep learning models,

such as CNN and CNN-ResNet. Therefore, the evaluation results obtained from testing the image dataset of the proposed CNN-VGG16 demonstrate a high accuracy of 97%. Thus, the proposed CNN-VGG16 has aided in identifying defective PCBs before they enter the production unit, potentially reducing waste and minimizing production costs. This is the expectation for smart factories; classifying defective products with high accuracy is the advancement required for Industry 4.0.

The f1-score's weight. When compared to the proposed model CNN-VGG16, CNN and CNN-ResNet performed less in precision, recall, and f1-score. Furthermore, the proposed CNN-VGG16 exhibits superior accuracy in distinguishing defective products from normal ones in smart factories through image

6. References:

1. Büchi, G.; Cugno, M.; Castagnoli, R. Smart factory performance and Industry 4.0. Technol. Forecast. Soc. Chang. 2020, 150, 119790.
2. Czimmermann, T.; Ciuti, G.; Milazzo, M.; Chiurazzi, M.; Roccella, S.; Oddo, C.M.; Dario, P. Visual-Based Defect Detection and Classification Approaches for Industrial Applications-A Survey. Sensors 2020, 20, 1459.
3. Wang L, Törngren M, Onori M. Current status and advancement of cyber-physical systems in manufacturing. J Manuf Syst 2015; 37: 517-27.
4. Wang P, Gao RX, Fan Z. Cloud computing for cloud manufacturing: benefits and limitations. J Manuf Sci Eng 2015; 137:1-10.
5. Lu Y, Xu X, Xu J. Development of a hybrid manufacturing cloud. J Manuf Syst 2014; 33(4):551-66.
6. Wang, J.; Ma, Y.; Zhang, L.; Gao, R.X.; Wu, D. Deep learning for smart manufacturing: Methods and applications. J. Manuf. Syst. 2018, 48, 144-156.
7. Tao, X.; Wang, Z.H.; Zhang, Z.T.; Zhang, D.P.; Xu, D.; Gong, X.Y.; Zhang, L. Wire defect recognition of spring-wire socket using multitask convolutional neural networks. IEEE Trans Compon. Packag. Manuf. Technol. 2018, 8, 689-698.
8. Adibhatla, V.A.; Chih, H.-C.; Hsu, C.-C.; Cheng, J.; Abbod, M.F.; Shieh, J.-S. Defect Detection in Printed Circuit Boards Using You-Only-Look-Once Convolutional Neural Networks. Electronics 2020, 9, 1547.
9. Zhang, E.; Li, B.; Li, P.; Chen, Y. A Deep Learning Based Printing Defect Classification Method with Imbalanced Samples. Symmetry 2019, 11, 1440.
10. Yun, J.P.; Shin, W.C.; Koo, G.; Kim, M.S.; Lee, C.; Lee, S.J. Automated defect inspection system for metal surfaces based on deep learning and data augmentation. J. Manuf. Syst. 2020, 55, 317-324.
11. Ren, H.; Tian, K.; Hong, S.; Dong, B.; Xing, F.; Qin, L. Visualized investigation of defect in cementitious materials with electrical resistance tomography. Constr. Build. Mater. 2019, 196, 428-436.
12. Prasitmeeboon, P.; Yau, H. Defect Detection of Particleboards by Visual Analysis and Machine Learning. In Proceedings of the 2019 5th International Conference on Engineering, Applied Sciences and Technology, Luang Prabang, Laos, 2-5 July 2019; pp. 1-4.
13. Ren, J.; Huang, X. Defect Detection Using Combined Deep Autoencoder and Classifier for Small Sample Size. In Proceedings of the 2020 IEEE 6th International Conference on Control Science and Systems Engineering (ICCSSE), Beijing, China, 17-19 July 2020; pp. 32-35.
14. Sampedro, C.; Rodriguez-Vazquez, J.; Rodriguez-Ramos, A.; Carrio, A.; Campoy, P. Deep Learning-Based System for Automatic Recognition and Diagnosis of Electrical Insulator Strings. IEEE Access 2019, 7, 101283-101308. [CrossRef]
15. H. Baumgartl, J. Tomas, R. Buettner, M. Merkel, A deep learning-based model for defect detection in laser-powder bed fusion using in-situ thermographic monitoring, Prog. Addit. Manuf. 5 (2020) 277-285.
16. Jayanthi, J., Lydia, E.L., Krishnaraj, N. et al. An effective deep learning features based integrated framework for iris detection and recognition. J Ambient Intell Human Comput 12, 3271-3281 (2021).
17. K. Vinayakan, M. V. Srinath, A. Adhiselvam, Security for Multipath Routing Protocol using Trust based AOMDV in MANETs, Vol. 2 No. 43 (2022), 1640-1654
18. Vinayakan, K., Srinath, M. V., & Adhiselvam, A. (2022), Reinforced securing of data leakage in Mobile Ad hoc Network (MANET) by hybrid mechanism of identity based encryption (IBE). International Journal of Health Sciences, 6(S8), 3622-3635
19. S. R. Boselin Prabhu, P. Rajeswari, A. Dinesh Kumar, An Analytical Review of Fiber-Optic Sensors and Biosensors, Journal of Engineering, Scientific Research and Applications, Volume 2, No. 1, 2016, 58-61
20. S. R. Boselin Prabhu, N. Balakumar, P. Rajeswari, A. Dinesh Kumar, Wireless Electricity Transfer Methodologies Using Embedded System Technology, Journal of Engineering, Scientific Research and Applications, Vol 2, No. 1, 2016, 81-89
21. S. R. Boselin Prabhu, P. Rajeswari, A. Dinesh Kumar, Analysis of Decentralized Clustering Hierarchy for Highly Distributed WSN, Journal of Engineering, Scientific Research and Applications, Vol 2, No. 2, 2016, 45-49