Data Science and Engineering (DSE) Record, Volume 6, issue 1.

Defect Detection for Electronics Enclosure Using Convolution Neural Network

Atit Luksida[1] and Prompong Sugunnasil[1][0000-0002-5805-0866]

[1] Department of Data Science, Faculty of Engineering, Chiang Mai University, Chiang-Mai 50200, Thailand
{atit l,prompong.sugunnasil}@cmu.ac.th

Abstract. The current challenge facing factories in Thailand is the transition to Industry 4.0. The process of appearance inspection has been transformed from human inspection to a computer-assisted tool. The objective of this process is to improve the accuracy of the inspection by removing human judgment. In this study, we propose a convolution neural network (CNN) to detect the defect of electronic enclosure. Then, we compare the proposed method with several other techniques, including SVM and KNN. The testing dataset comprises 1,190 images captured from a camera oriented in a consistent direction. These images were divided into four balanced classes to mitigate any issues related to class imbalance during model training. Although SVM demonstrated superior accuracy, the substantial time required for training makes it impractical for real-world applications where time efficiency is crucial. In contrast, despite having slightly lower accuracy, CNN showed a beneficial balance between performance and computational efficiency, making it a more pragmatic choice in many real-world scenarios. KNN, although faster than SVM, had the lowest performance in our tests.

Keywords: Defect Detection · Convolution Neural Network · Image Processing.

1 Introduction

Many industries in Thailand are adapting to Industry 4.0. Motor vehicle production grew more than 280% from 2000 – 2007, and we can see that passenger cars were assembled and in very high demand (22). One of the enabling keys is the adoption of new technology. One area where Industry 4.0 has made significant progress is in the field of image classification. By training machine learning models to classify images, industries can automate tasks such as object recognition (20) and defect detection(1), instead of relying on human inspection and decision-making. This saves time and resources and improves the accuracy and reliability of these tasks.

There is a problem that we have identified in the product inspection process. When people conduct inspections using a microscope, mistakes are often made, such as not covering all of the inspection criteria, being unable to identify problems, and forgetting about defective criteria that were previously accepted.

These mistakes lead to lost time, delays in the next production process, and a Proliferation of reducible activities in the company.

In this article, we will explore the concept of image classification using machine learning. We will also delve into the different techniques and approaches that can be used for image classification, including deep learning methods like convolutional neural networks (CNNs) and more traditional approaches like K-nearest neighbors and support vector machines.

The organization of this article is as follows. The next section is the literature review on the methodology. Section 3 provides the details on the proposed method and section 4 explains the experiments and their results. Finally, the conclusion and the future work are discussed in the last section.

2 Literature Review

This section aims to provide the foundation knowledge on the related research. Section sec:cnn provides the theoretical point of view on the convolution neural networks (CNN) and Section 2.2 discusses the existing research on manufacturing defect detection using image processing techniques.

2.1 Convolution Neural Networks

The foundational concept of CNNs was inspired by the notion of self-organization in a multi-layer perceptron. The earliest model embodying this idea was the Neocognitron, introduced by Fukushima in 1980 (5). However, despite setting a conceptual precedent, the Neocognitron was somewhat limited in its practical usability due to the lack of learning algorithms. Addressing this limitation in 1998 (11), developed LeNet-5, a 7-level convolutional network. LeNet-5 was a significant advancement in the field, particularly successful in handwriting and character recognition tasks. Based on LeNet-5, the architecture of a CNN is composed of several components. The convolutional layer is used for local receptive field learning. The pooling layer serves the purpose of down-sampling and providing translation invariance. The fully connected layer integrates the learned features for the ultimate classification task. Subsequently, the learned representations are flattened and passed through a SoftMax function to carry out the final classification (13).

The model was enhanced by adding more layers and incorporating the ReLU(Rectified Linear Unit) as an activation function, as demonstrated in AlexNet. This change was crucial in addressing the vanishing gradient problem that often hinders the training of deep neural networks (10).

The VGG networks (18) introduced increased depth to the architecture, incorporating up to 19 layers, which found extensive use in the field of computer vision. Furthermore, models such as GoogLeNet or Inception and ResNet incorporated the principles of CNN into their base architectures. These advances strengthened and improved the accuracy of the models, establishing them as foundational architectures for computer vision tasks. The general structure of CNN is as follows.

– Input Layer. This layer receives the image and considers its dimensions, such as width, height, and depth.

– Convolutional Layer. This layer is responsible for extracting features from the given image. The output for each element in the feature map is calculated as.

$$\mathrm{Output}(i, j) = \sum_m \sum_n \mathrm{Input}(i + m, j + n) \times \mathrm{Kernel}(m, n)$$

– Pooling Layer. This layer's primary role is to diminish the spatial dimensions, specifically, the width and height, as shown in the following equation.

$$\mathrm{Result}(i, j) = max\, \mathrm{Input}\,(i + m, j + n)$$

– Output Layer. In the case of classification tasks, this layer often reduces the input dimensions to match the number of target classes.

Subsequently, the application of CNNs extended beyond image recognition. They have been adopted across various domains, including natural language processing (9), medical image analysis (12), and even astronomy for tasks like star-galaxy separation (3).

2.2 Related Researches

Defect detection in industrial products is a critical task aimed at ensuring quality control in manufacturing processes. It involves the identification of anomalies, irregularities, or deviations from the standard specifications in products. The evolution of this field has been marked by the transition from manual inspection to automated systems, leveraging advanced machine learning algorithms to enhance accuracy, efficiency, and speed. Thus, the integration of machine learning techniques in defect detection for industrial products has revolutionized quality control processes.

According to Table 1, the reviewed literature can be broadly categorized into two groups based on the machine learning techniques employed: traditional machine learning algorithms and deep learning methods. Traditional algorithms like KNN and SVM have laid the groundwork, offering robust classification based on feature similarity and high-dimensional data handling. However, the advent of deep learning, especially CNNs, has markedly improved the field, enabling more accurate and efficient defect detection. The use of attention mechanisms in CNNs further underscores the ongoing innovation in this domain, highlighting the potential for even more sophisticated and effective defect detection methods in the future. It becomes clear that CNNs gain a lot of attention in the area of defect detection across diverse manufacturing sectors. It becomes evident that CNNs stand out as a highly appropriate choice for defect detection in the context of electronics enclosure manufacturing. Drawing upon the recent

Table 1. Recent Research on Defect Detection in Manufacturing

Year	Author	Subject of Study	ML Technique	Result Performance
2023	*Saberironaghi, Ren, & El-Gindy (2023)(15)*	Defect Detection in Manufacturing Processes	CNN	Improved accuracy in defect detection.
2020	Saqlain, Abbas, & Lee (2020) (16)	Semiconductor Manufacturing	CNN	Enhanced defect detection in semiconductor images.
2021	Jiang et al. (2021). (8)	Industrial X-ray Images	CNN	Efficient defect detection in X-ray images.
2021	Ortega Sanz et al. (2021). (14)	Automotive Manufacturing	CNN	Successful detection of defects in automotive manufacturing.
2022	Shaikh, Hujare, & Yadav (2022) (17)	Manufacturing Surfaces	CNN	Automation of defect detection in manufacturing surfaces.
2020	Wen et al. (2020) (24)	Semiconductor Images	CNN	Novel CNN-based approach for semiconductor defect detection.
2022	Djavadifar et al. (2022) (4)	Manufacturing Processes	CNN	CNN application for defect detection in manufacturing processes.
2019	Wang & Zhu (2019) (23)	Turbine Blade and Transmission Case	CNN	Improved defect detection using SVM and deep learning.
2012	Jazi, Liu, & Lee (2012) (7)	Glass Substrates	CNN	SVM optimized with simulated annealing (SA).
2016	Yıldız, Buldu & Demetgul (2016)(25)	Texile Fabrics	CNN	KNN-based defect classification.

and reputable academic resources we have referenced, we can discern compelling reasons for this selection. As a consequence, we propose their application as the core methodology in our research for defect detection in electronics enclosure manufacturing. This choice is informed not only by their demonstrated accuracy but also by their adaptability to various manufacturing environments, making CNNs a compelling choice for achieving superior results in defect detection tasks.

3 Defect Detection for Electronics Enclosure Using

This section is dedicated to the explanation of the proposed method. The working strategy of this work is straightforward. Two main stages include the data preparation and the modeling process. The overall concept of this work is illustrated in Figure 1.

Fig. 1. Overall architecture.

The data preparation phase begins with resizing the images to a uniform resolution. We have standardized the input resolution at 150 x 150 pixels to speed up the CNN's training process. Our experimental design has shown that this resolution offers an ideal balance between processing speed and validation accuracy; although smaller sizes could increase speed, they did not significantly improve accuracy. As a result, 150 x 150 pixels has been set as our default resolution. Additionally, we use an RGB color scheme which is essential for detecting specific color-dependent defects in our connectors. Images are then classified into one of four predefined categories, according to general specifications defined by the engineering team. We can detail the defects as follows.

– **Good.** This class refers to a product that has been assembled completely without any errors.
– **Burr.** This class indicates the presence of additional material on the product, typically resulting from the injection molding process.
– **Damage.** This class refers to product areas that do not conform to the specified shape or might be missing some components.
– **Metallic.** This class implies foreign matter, possibly metal debris or particles is attached to the connector.

Then, the data was normalized to standardize these variations by scaling the pixel values to a range of 0 to 1. This was achieved by dividing each pixel value by the maximum possible value, which is 255 for RGB images. Thereby ensuring a consistent level across the dataset, this scaling process enhances the overall effectiveness of the model by facilitating faster convergence during training and reducing the likelihood of certain features overwhelming others due to their scale. Such normalization is crucial for deep learning models that rely heavily on the magnitude and range of the input data.

4 Defect Detection for Electronics Enclosure Using

The objective of this section is to demonstrate the performance of the proposed method. Section 4.1 discusses how the experiment is set up from the data collection to the performance evaluation. Section 4.2 displays the results of the studies, and Section 4.3 discusses the results.

4.1 Related Researches

The dataset utilized for this study was obtained using a digital microscope with a capacity of 5 million pixels. The microscope was utilized to capture images at a magnification range of 10 to 300 times, accurately recording features of the subject to its original size of 1280 x 1024 pixels. The captured images, maintaining a resolution of 60 dpi and a 24-bit depth, offer detailed visual information, enhancing the quality of the dataset. We collected extensive data comprising more than 1,000 images. The number of images for each class is shown in Table 2. These images illustrate both the defective and intact aspects of an actual connector in relation to the predefined class in Section 3. The categorization facilitates an efficient and systematic evaluation, assisting in achieving accurate and meaningful results from the study as per Figure 2 for sample.

Table 2. Number of images for each class.

Class	Number of images
Good	300
Burr	290
Damage	280
Metallic	320
Total	1,190

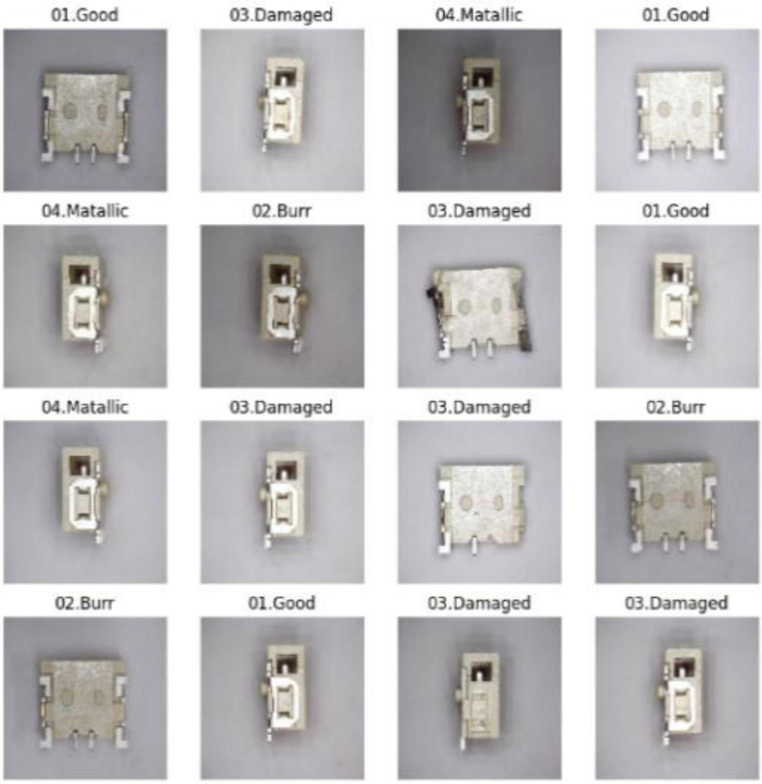

Fig. 2. Example of raw images from each classes.

The performance evaluation of the proposed method includes precision, recall, F1-Score, and accuracy.

– **Precision.** Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. It is defined as:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

High precision indicates a low rate of false positives (19).

– **Recall.** Recall, also known as sensitivity, is the ratio of correctly predicted positive observations to all observations in the actual class. It is defined as:

$$\text{Recall} = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

High recall indicates a low rate of false negatives (2).

– **F1-Score.** The F1-Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. It is defined as:

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-Score is especially useful when seeking a balance between Precision and Recall (21).

– **Accuracy**. Accuracy is the ratio of correctly predicted observations to the total observations. It is defined as:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Observations}}$$

Accuracy is a useful measure when the target classes are well balanced (6).

Moreover, we also studied the training time of the model. Two well-known methods challenge the proposed method: K-nearest neighbors (KNN) and Support Vector Machine (SVM). All procedures were executed using Python in a Jupyter Notebook within the Anaconda environment. The computations were conducted on a machine equipped with a Ryzen 7 5800 CPU running at 3.8 GHz, 16 GB DDR4 RAM, and an Nvidia RTX 3050 4GB GPU. The CNN model was constructed using the Keras library, while the scikit-learn (SKlearn) library was utilized for the development of the SVM and KNN models. Our dataset comprised 1,190 images, split into four balanced classes to avoid class imbalance problems during training. We used 80% of the images for training and the remaining 20% for validation. For the CNN model, we adopted a conventional structure, which was refined through a systematic evaluation of performance metrics against various configurations, ultimately leading us to the model outlined in Figure 3.

4.2 Experimental Results

The experimental results presented in Table 3 demonstrate the performance of various machine learning models in defect detection for electronics enclosures. The proposed CNN, SVM, and KNN were evaluated based on precision, recall, F1-score, and accuracy metrics. The CNN model shows a higher degree of effectiveness with a balanced performance across all metrics, achieving an accuracy of 89.5%. It indicates a high level of reliability in both positive defect detection and the ability to classify non-defective cases correctly. The SVM outperforms the other models, with an impressive accuracy of 98.3%. This high performance suggests that the SVM is particularly well-suited for the high-dimensional space typical of image data in electronic enclosures. Its precision and recall are equal at 98.0%, indicating an excellent balance between sensitivity and specificity. The KNN model demonstrates a lower performance compared to the other models, with an accuracy of 85.0%. This may suggest that the KNN algorithm while being a simpler and more interpretable model, is less capable of handling the complexity of the defect detection task in the given context.

Table 3. Comparison of the precision, recall, F1-score, and accuracy of CNN, SVM and KNN.

Model	Precision	Recall	F1-Score	Accuracy
CNN	90.0%	89.0%	89.0%	89.5%
SVM	98.0%	98.0%	98.0%	98.3%
KNN	86.0%	85.0%	85.0%	85.0%

The training time for each machine learning model is a critical aspect of resource consumption in the defect detection task for electronics enclosures. Table 4 compares the training times required by the proposed CNN, SVM, and KNN. The CNN model required 108.58 seconds for training, which indicates a high efficiency in terms of computational time. This efficiency makes the CNN model a practical choice for scenarios where quick model deployment is necessary. On the other hand, the SVM model took significantly longer, with a training time of over 10 hours. This substantial increase in training duration may be attributed to the SVM's computational complexity, especially when dealing with large feature spaces commonly present in image data. Lastly, the KNN model's training time was recorded at 189.277 seconds. While not as efficient as the CNN, the KNN training time remains reasonable. However, it is important to note that the KNN algorithm typically has a faster training phase but can be slower during the prediction phase due to its lazy learning nature.

As seen in Figure 4, the training loss decreases sharply within the initial epochs, indicating a rapid learning phase where the model quickly assimilates the patterns within the training data. The subsequent gradual decline suggests that the model continues to learn and improve, albeit at a slower rate, as it begins to converge toward an optimal set of weights. The validation loss, representing the

Table 4. Training time for each model

Model	Training time
CNN	108.58 Sec
SVM	> 10 Hours
KNN	189.277 Sec

model's performance on unseen data, mirrors the training loss closely. This close correspondence suggests that the model is generalizing well and not overfitting to the training data. The validation loss reaches a plateau early, which is an indication that additional training beyond this point does not yield significant improvements in model performance on the validation set. As shown in Figure 5, the training accuracy curve shows a steep ascent within the initial few epochs, reaching a high level of accuracy swiftly. This rapid increase suggests that the CNN model is capable of learning the distinctive features of the dataset effectively. After the sharp rise, the training accuracy plateaus, indicating that the model has nearly optimized its parameters for the training dataset. Conversely, the validation accuracy increases alongside the training accuracy, which is indicative of the model's ability to generalize to new, unseen data. The small gap between the training and validation accuracy implies that the model is not overfitting and has good predictive performance.

4.3 Experimental Results

The experimental results yield insightful implications for applying machine learning models in defect detection for electronic enclosures. The CNN, while slightly less accurate than the SVM, presents a compelling balance between computational efficiency and performance. Its quick training time aligns with industrial needs for rapid deployment. The SVM, despite its high accuracy, is less favorable due to the impractical training duration. The KNN, with lower performance metrics, reinforces the necessity of complex models to handle the intricacies of defect detection tasks. The consistency between training and validation loss and accuracy for the CNN underscores its robustness and potential for real-world applications, showcasing the model's ability to generalize beyond the training data without significant overfitting.

5 Conclusion

This work presented a Convolution Neural Network (CNN) model for defect detection in electronics enclosures, demonstrating its viability against traditional machine learning techniques. The proposed CNN model balanced accuracy and computational efficiency, outperforming the KNN in speed and being more practical than the SVM in training duration. The experimental results highlighted CNN's capability for rapid learning and generalization without significant overfitting, aligning well with the needs of industrial applications in an Industry 4.0 context.

Future research will focus on further optimization of the CNN architecture for defect detection, exploring the effects of varying layer depths and activation functions.

Additional work could also investigate using real-time data streams to enhance the model's predictive capabilities and adaptability to different manufacturing environments. Another promising direction is integrating attention mechanisms and other recent innovations in deep learning to improve defect detection accuracy and computational efficiency. Lastly, extending the application of the proposed model to other areas in manufacturing and beyond presents a significant opportunity for broader impact.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 150, 150, 32)      896

 batch_normalization (BatchN  (None, 150, 150, 32)     128
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 75, 75, 32)       0
 )

 dropout (Dropout)           (None, 75, 75, 32)        0

 conv2d_1 (Conv2D)           (None, 75, 75, 64)        18496

 batch_normalization_1 (Batc  (None, 75, 75, 64)       256
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 37, 37, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 37, 37, 64)        0

 conv2d_2 (Conv2D)           (None, 37, 37, 128)       73856

 batch_normalization_2 (Batc  (None, 37, 37, 128)      512
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 18, 18, 128)      0
 2D)

 dropout_2 (Dropout)         (None, 18, 18, 128)       0

 conv2d_3 (Conv2D)           (None, 18, 18, 128)       147584

 batch_normalization_3 (Batc  (None, 18, 18, 128)      512
 hNormalization)

 max_pooling2d_3 (MaxPooling  (None, 9, 9, 128)        0
 2D)

 dropout_3 (Dropout)         (None, 9, 9, 128)         0

 flatten (Flatten)           (None, 10368)             0

 dense (Dense)               (None, 512)               5308928

 batch_normalization_4 (Batc  (None, 512)              2048
 hNormalization)

 dropout_4 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 4)                 2052

=================================================================
Total params: 5,555,268
Trainable params: 5,553,540
Non-trainable params: 1,728
_________________________________________________________________
```

Fig. 3. Architecture of the proposed CNN.

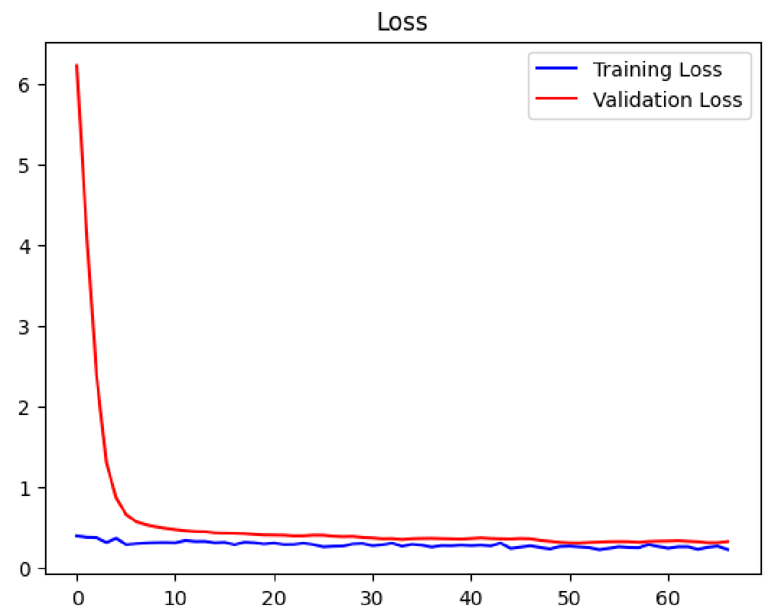

Fig. 4. Training and Validation Loss over Epochs

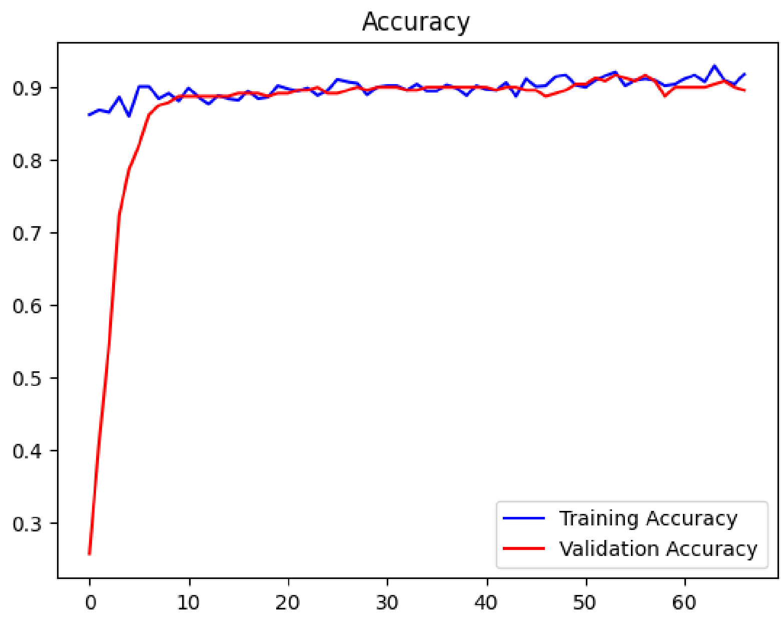

Fig. 5. Training and Validation Accuracy over Epochs

Bibliography

[1] Badmos, O., Kopp, A., Bernthaler, T., Schneider, G.: Image-based defect detec tion in lithium-ion battery electrode using convolutional neural networks.

[2] Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves pp. 233–240 (2006)

[3] Dieleman, S., Willett, K.W., Dambre, J.: Rotation-invariant convolutional neural networks for galaxy morphology prediction. Monthly notices of the royal astronomical society 450(2), 1441–1459 (2015)

[4] Djavadifar, A., Graham-Knight, J.B., Krber, M., Lasserre, P., Najjaran,H.: Automated visual detection of geometrical defects in composite manufacturing processes using deep convolutional neural networks. Journal of Intelligent Manufacturing 33(8), 2257–2275 (2022)

[5] Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics 36(4), 193–202 (1980)

[6] Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent Data Analysis 6(5), 429–449 (2002)

[7] Jazi, A.Y., Liu, J.J., Lee, H.: Automatic inspection of tft-lcd glass substrates using optimized support vector machines. IFAC Proceedings Volumes 45(15), 325–330 (2012)

[8] Jiang, L., Wang, Y., Tang, Z., Miao, Y., Chen, S.: Casting defect detection in x-ray images using convolutional neural networks and attention-guided data augmentation. Measurement 170, 108736 (2021)

[9] Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)

[10] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25 (2012)

[11] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)

[12] Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., S´anchez, C.I.: A survey on deep learning in medical image analysis. Medical image analysis 42, 60–88 (2017)

[13] Momeny, M., Latif, A.M., Sarram, M.A., Sheikhpour, R., Zhang, Y.D.: A noise robust convolutional neural network for image classification. Results in Engineering 10, 100225 (2021)

[14] Ortega Sanz, D., G´omez Mu˜noz, C.Q., Garc´ıa M´arquez, F.P.: Use of convolutional neural networks for quality control in automotive industry. Introduction to Internet of Things in Management Science and Operations Research: Implemented Studies pp. 231–245 (2021)

[15] Saberironaghi, A., Ren, J., El-Gindy, M.: Defect detection methods for industrial products using deep learning techniques: A review. Algorithms 16(2), 95 (2023)

[16] Saqlain, M., Abbas, Q., Lee, J.Y.: A deep convolutional neural network for wafer defect identification on an imbalanced dataset in semiconductor manufacturing processes. IEEE Transactions on Semiconductor Manufacturing 33(3), 436–444 (2020)

[17] Shaikh, D., Hujare, D., Yadav, S.: Surface defect detection using convolutional neural network model architecture. Journal of Engineering Research and Sciences 1(5), 134–144 (2022)

[18] Simonyan, K., Zisserman, A.: Very deep convolutional networks for largescale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[19] Sokolova, M., Lapalme, G.: Systematic analysis of performance measures for classification tasks. Information Processing Management 45(4), 427–437 (2009)
[20] Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision 104, 154–171 (2013)
[21] Van Rijsbergen, C.J.: Information retrieval, 2nd edition (1979)
[22] Wad, P.: The automobile industry of southeast asia: Malaysia and thailand. Journal of the Asia pacific Economy 14(2), 172–193 (2009)
[23] Wang, Z., Zhu, D.: An accurate detection method for surface defects of complex components based on support vector machine and spreading algorithm. Measurement 147, 106886 (2019)
[24] Wen, G., Gao, Z., Cai, Q.,Wang, Y., Mei, S.: A novel method based on deep convolutional neural networks for wafer semiconductor surface defect inspection. IEEE Transactions on Instrumentation and Measurement 69(12), 9668–9680 (2020)
[25] Yıldız, K., Buldu, A., Demetgul, M.: A thermal-based defect classification method in textile fabrics with k-nearest neighbor algorithm. Journal of Industrial Textiles 45(5), 780–795 (2016)