electronics

MDPI

Article

Machine Design Automation Model for Metal Production Defect Recognition with Deep Graph Convolutional Neural Network

Yavuz Selim Balcıoğlu [1,*], Bülent Sezen [2], Ceren Cubukcu Çerasi [1] and Shao Ho Huang [3]

[1] Department of Management Information System, Faculty of Business Administration, Gebze Technical University, 41400 Gebze, Turkey
[2] Department of Business Administration, Faculty of Business Administration, Gebze Technical University, 41400 Gebze, Turkey
[3] Department of Material Engineer, Faculty of Science, Gebze Technical University, 41400 Gebze, Turkey
* Correspondence: ysbalcioglu@gtu.edu.tr

Abstract: Error detection has a vital function in the production stages. Computer-aided error detection applications bring significant technological innovation to the production process to control the quality of products. As a result, the control of product quality has reached an essential point because of computer-aided image processing technologies. Artificial intelligence methods, such as Convolutional Neural Network (CNN), can detect and classify product errors. However, detecting acceptable and small defects on base parts cannot be done with a high rate of accuracy. At this point, it is possible to detect such minor errors with the help of the graph convolutional network, which has emerged as a new method. In this study, the defect elements on the surfaces of metal nut parts are determined through the graph convolutional network, and quality control is ensured. First, the surface images of the metal nut parts are captured. For this, a python-based Raspberry pi card and a modified camera system were installed. Adapters with three different zoom options are used on the camera system, depending on the part to be captured. The images obtained in the second step are sent to the other computer, which is used for image processing via the local server. In the third stage, image transformations are obtained by graphically separating the obtained images in white and black color tones on the second computer, and histogram maps of these images are drawn. Value ranges of these maps are determined and classified according to the value ranges obtained from the images of the defective parts. As a result, nine different models were analyzed. According to the analysis results, the graph convolutional neural network method gives 2.9554% better results than conventional methods.

Keywords: metal; computer vision; neural network; graph; surface defect

Citation: Balcıoğlu, Y.S.; Sezen, B.; Çerasi, C.C.; Huang, S.H. Machine Design Automation Model for Metal Production Defect Recognition with Deep Graph Convolutional Neural Network. *Electronics* **2023**, *12*, 825. https://doi.org/10.3390/electronics12040825

Academic Editor: Stefanos Kollias

Received: 22 September 2022
Accepted: 21 November 2022
Published: 6 February 2023

1. Introduction

One of the most critical issues in the production industry is the production of completed work that accurately meets the company's standards. Establishing the right method and model for the production systems is essential for these standardizations. Afterward, optimized methods should be applied to the production processes. It is seen that traditional methods are still used across the entire production phase. On the other hand, due to technological developments such as industry 4.0, companies need artificial intelligence in addition to computer systems in the production and all other fields [1]. These developments have both advantages and disadvantages. As an example, human error detection is generally effective for detecting errors; however, the use of such a method can create a costly workforce expenditure [2]. The correct method can successfully detect surface defects and prevent hardware system problems. Accordingly, automated surface inspection (A.S.I.) has recently become a popular area of study. The development of technology, particularly

involving big data, has made the collection of product data more accessible. Currently, such data are divided into different classifications, known as data types. Topological data analysis comprises topological images of earth structures, and visuals of microbiological structures are used as graphical representations of traffic density, flow charts, and social networks [3]. Although data is easy to access, it is essential to establish the right model and obtain the right results.

With the ongoing development of technological possibilities, computer system power, and deep learning, the opportunity to obtain unlimited data has made it possible to detect surface defects and classify errors; the detection of such errors is of great importance and significance for quality control. As the major enablers of defect detection, image processing techniques for detection commonly consist of defect localization, recognition, and classification [4]. The convolutional neural network method is shown to be the most important of these developments. It has made profound contributions to machine learning, including speech recognition and font recognition, starting from image processing as a field of use. The surface of each piece has a unique image map, similar to a human fingerprint [5]. This map needs to be expressed graphically to be read and interpreted by artificial intelligence. This method is called the graph convolutional network method. This method aims is to convert the images of surface defects into numerical expressions for parts produced either with errors or without errors. The established model's numerical error values detect small details on the surface, which can be missed by the classical image processing method [6]. Each image is divided into a certain number of frames during the model installation phase, allowing the computer to process the image piece by piece. The larger the image size, the slower the computer's processing of the image [7]. Image analysis-based detection approaches can be quite difficult [8], and are entirely dependent on the quality of photos taken under different situations (e.g., lighting, shadow, etc.). The next sections of our research study are organized as follows: The literature related to our research is examined in Section 2. The proposed research method is reviewed in Section 3. The research model in the context of an image processing system is discussed in Section 4. The results and evaluations of our proposed research study are examined in Sections 5 and 6. Finally, the conclusions of our research study are stated in Section 7.

2. Literature Review

The biggest achievement for the neural network was devised by Krizhevsky et al., using 1.2 million high-resolution images in deep learning studies, which created 650,000 neurons with 60 million parameters. They employed a new regularization approach that proved to be very useful for lowering overfitting in the globally connected layers. They obtained successful findings in their analysis results [9]. In recent years, machine learning has performed well in many fields when applied to defect detection and segmentation. For example, in the field of medical science, the most injured ligament in the human body is the anterior cruciate ligament (ACL). ACL injuries are commonplace among football, basketball, and soccer players. The proposed approach used a customized 14-layer ResNet-14 architecture of a convolutional neural network with six different directions. They could be used to automatically detect and evaluate ACL injuries in athletes [10]. Convolutional Neural Network (CNN) is one of them; they introduce a generalization of CNN from low dimensional grid data. Their proposal was a novel spatial convolution utilizing an aimless walk to uncover the relations within the input. They sought an explanation for the classification and regression problems that generally arise from separating the grid fields by spatial pixel method using 2-dimensional images of low-resolution size as graphical data [11]. Many object detection systems have been applied towards defect detection. The optical experiment was performed on circuit boards' surfaces using CNNs. In this study, a device classification process using CNN, which is a deep learning technique, has been used. Lim et al. used two different models, which were established to determine the best results by comparing models in another study [12]. Yang et al. used this method mainly to improve computing time and achieve a better success rate. In the process, machine

learning was designed to analyze laser sources' visual examinations on the surface using CNNs. Yang et al.'s contributions to this study proved that the correct Visual Geometry Group (V.G.G.) model, which was trained on a large image index, could be used to identify the defect distribution of laser welding. It also proved that the pre-trained V.G.G. model has a small model size, lower error positive rate, and shorter training time. The V.G.G. model was created for these reasons, and a 99.87% success rate was achieved [13]. The deep learning model has gradually become a popular research direction for image detection with CNN, called Deep Convolutional Neural Network (DCNN), which is a novel method for the classification of noisy non-stationary time-series signals based on Cohen's class of their time-frequency representations (TFRs) and deep learning algorithms. They demonstrated the proposed approach through the example of detecting gravitational-wave (GW) signals in intensive real-life noise. The proposed approach can also be a viable solution for deep learning-based analysis of numerous other noisy signals in different practical applications [14].

Error detection has been determined to be an analysis method other than the traditional technique using the deep convolutional neural network method on whether the L.E.D. chips produced are faulty. This study shows that the DCNN method gives more successful results than other methods. It has been reported to be 5.04% more successful than previous methods [15]. Image detection based on deep learning has an accuracy that cannot be achieved by traditional methods. For example, railroad track surfaces were controlled using the deep convolutional neural network method, and these analyses were carried out using video images recorded for hours. They achieved successful results in terms of the method they used [16]. Currently, image surface detection based on deep learning has been applied in almost all disciplines; in the study by Faghih-Roohi et al., a deep convolutional neural network results in the analysis of image data for the exposure of rail surface defects. They compared the results of different network architectures characterized by sizes and activation functions. Their findings were promising and proved the capability of the proposed advance [17]. Additionally, it was used differently in areas such as Bruna et al.'s study, which proposed to use it as a restrictive model, as in a Gibbs distribution, of which tolerable statistics are given by deep convolutional neural networks. They classified the proposed model analytically in the image super-resolution task, but did so generally, in such a way that could be used in other challenging posed issues, such as audio bandwidth [18]. Current state-of-the-art image surface detectors are based on DCNN. The problem is computing analysis takes days to calculate. Today, deep learning, as a long and important past, has evolved into Graphic Convolutional Neural Network (GCNN). The first sign of these methods was used as graphic structure data; Atwood and Towsley created diffusion convolutional neural networks, a new model for graph-structured data. Their research shows how scattering-based representations can be learned from graph-structured data and used as an efficient basis for node classification. They demonstrate that DCNNs can surpass probabilistic relational models and kernels on graph methods at comparative node classification tasks [19]. Moreover, unlabeled data were analyzed using graph-structured data. Significant progress has been made regarding unlabeled data due to this analysis [20]. The graph convolutional neural network method has also been applied in molecular activities and traffic density estimates [21]. They provide an extensive overview of graph neural networks (GNNs) in data mining and machine learning fields. They also discussed the practice of graph neural networks across profuse domains and summarize the open-source codes, benchmark data sets, and model valuation of graph neural networks. First, Bruna et al. used graph-based convolutional architectures in their work. This work, which is included in the Spatial section, is an example of images displayed in different colors, such as cubic [22]. They showed that for low dimensional graphs, it is possible to learn convolutional layers with several specifications independent of the input size, resulting in deep methodical architectures. For an example of GCNN performance research, Li et al. improved traditional graphic convolutional network studies by developing an adaptive graph convolutional network. Li et al. propose a generalized and adjustable graph CNN

by taking data from a random graph design as input. Accordingly, a task-forced adaptive graph is accomplished for each graph's data while training. Li et al.'s research also shows that comprehensive experiments on nine graph-structured datasets have validated the superior performance advancement regarding both concurrence speed and predictive accuracy [23]. As an example of supervised learning, which is labeled data; in Zhuang and Ma's study, they used a simple and scalable semi-supervised learning method for graph-structured data. Their method aimed to achieve graph convolution from different views of the crude data. In their experiment, using both unsupervised and supervised loss functions, their method exceeded the state-of-the-art techniques from different datasets [24]. Supervised learning models generally require huge amounts of datasets to achieve good performance. Villalba-Diez et al. built an application to evaluate the means by which a Deep Learning soft sensor function can be combined with a high-resolution optical quality control camera to improve the accuracy and lower the cost of an industrial visual analysis process in the Printing Industry 4.0. An accuracy rate of 98.4% was achieved in this study using deep learning methods for printing technologies' quality control systems. Graph-based methods, such as graph convolution network (GCN), can solve this problem well. However, semi-supervised learning, which learns from both labeled and unlabeled samples, is more suitable for this task. Their paper proposes a new method [25] called Multiple Micrographs Graph Convolutional Network (MMGCN). The proposed MMGCN can achieve better computation complexity and practicality than GCN. For accuracy, it can also obtain the best performance and the best class separation. Examining the electrostatic potential (ESP) of the surfaces of ligands and proteins is an essential step in the drug design process. High level quantum mechanics calculations often take a significant amount of time. However, ESP surfaces are required to accurately depict the real electrostatic nature of the molecules. They offer a graph convolutional deep neural network (GCDNN) model that, in a fraction of a second [26], creates electrostatic potential (ESP) surfaces for ligands. The class imbalance in surface defect recognition greatly influences its performance. Some deep learning-based methods have been developed to address the class imbalance. They proposed a new graph-based method, named anchor-based class-balanced GCN (ACB-GCN) [27]. The proposed method achieves better performance than traditional methods and the original GCN, especially in minority classes.

In this paper, our purpose is to shorten the time and also build a low-cost system; this study's primary purpose was to use high-resolution images to increase the accuracy rate and keep the cost low [28].

3. Proposed Method

Our goal is to build a low-cost system for the production line by taking product images, analyzing them as a graph, and performing reasoning on the graph convolutional neural network (GCNN) for spatial classification. GCNN has been chosen because the classification accuracy of GCNN was significantly higher than a two-dimensional CNN for a thickness spatial-based map generated by a graph projection. GCNN also proved position invariance of regional features that renders restate of its pre-trained model for applications without significant distortions in the outcome [29]. The superior performance of GCNN is attributable to the CNN's properties, which are derived from its brain-like architecture and its surface-based representation of the cortical neuron's information.

In this study, firstly, the production model was established in real-time to obtain image data. It aimed to apply supervised surface defect classification to the images obtained through this system with the graph convolutional neural network method. It aimed to apply the spatial resolution method to each image obtained from the product and to create cellular areas. However, each image must be converted to color format from grayscale. As the next process, a 12×12 grid area was created on the image. It was divided into squares to determine whether there were any defects on the surface based on the color value (k) in each square. The error value determined for the model indicates 90% and above black color tone as "k" values. As a result of the graph convolutional neural network method,

with the average value range calculated by computing these values, the training phase was completed. In the last phase, the real-time product images' estimation stage was analyzed over the graph convolutional neural network.

Graph Convolutional Neural Network

One sort of neural network is known as a graph neural network (GCNN), and it is designed to function naturally with data that is organized as a graph. GCNNs have the potential to create more informed predictions about entities in these interactions when compared to models that evaluate individual entities in isolation. This is accomplished by extracting and using attributes from the underlying graph. Graph Convolutional Neural Network (GCNN) is one of these neural network method types. Its difference from other neural networks is that it has a graphics-based structure. It is a natural method to represent information that comes from the current world, and graphs are a basic data structure that can be found in a variety of application disciplines. Graphs may be used to represent a wide variety of systems and interactions, including social networks, molecules, organizations, citations, physical models, and transactions, to name a few examples. Nevertheless, it is significantly difficult to perform calculations using a graph. Graphs are mathematical models that are very adaptable; as a result, there is no consistency in their structure from instance to instance. It is not a simple task to describe graphs in a way that allows processors to use them for computation, and the success of the endeavor is greatly dependent on the issue at hand. In many graphs, the relationships between the nodes do not have any inherent ordering. Because graphs do not have a structure like a grid, it may be challenging to determine the correct order for the nodes inside the graph. This contrasts with images, where each pixel can be recognized individually based on its absolute location. Graphs make it possible to collect relations between data and allow for the collection of data. Traditional methods of machine learning and deep learning are often limited to standard data structures like grids or sequences and can not handle the complexity of graphs. Steamrolling the input, discarding all the structural information, and processing it in a lower dimensional representation is the only method available to explore graphs when using a conventional deep learner. During the last 10 years, several varieties of graph neural networks have emerged, including graph auto-encoder, representation learning, and graph attention. A differentiable model that serves as the update function is often the component of a GCNN that makes up the simplest and most basic design. General neural networks (GCNNs) are built on top of a specific kind of structure that is known as the "message passing neural network", which was introduced by Gilmer et al. [30]. A learned embedding is produced by this framework when the differentiable update function is applied to each node vector. It is possible for edges and global-context vectors to have features in the same way that nodes do. To get embeddings for the edges of the graph, as well as a single embedding for the whole graph, the procedure described above is performed for each edge and each global-context vector. A convolutional pooling function is often included in the most popular design for Graph Neural Networks. This is because the parameters that are utilized are typically the same across all the places in the graph. On the other hand, performing convolutions on graphs is more complicated than doing so on photos. Consequently, Kipf and colleagues developed what they call a "rapid approximation convolution" on graphs, which achieves the same goal as the conventional method of convolution [31].

In order to use the approach that they provide, you will need to enter the node characteristics (x_i) and the adjacency matrix (A). In the first step of the process, the adjacency matrix is altered by the addition of an identity matrix. This ensures that the information from the self-node is preserved, even after the feature vectors of the nearby nodes are added together. This modification to the graph, shown by the notation $\forall = A + I$, is known as introducing a self-loop. Because the feature vectors of the initial graph will be scaled out of proportion if they are multiplied directly, another modification that they apply addresses the fact that that the adjacency matrix of a graph requires normalization before applying

it directly. This is because if it is multiplied directly, the adjacency matrix will be applied directly. In order to address the problem described by $D^{-1}\forall$, the adjacency matrix has been normalized using the inverse of the degree matrix. In fact, symmetric normalization is used rather than naïve averaging of the values of the nodes in the immediate vicinity of $D^{-1/2}\forall D^{-1/2}$. In this scenario, the final aggregation function is denoted as:

$$f\left(H^{(I)}, \forall\right) = \sigma\left(D^{-\frac{1}{2}}\forall D^{-\frac{1}{2}}H^{(I)}W^{(I)}\right) \quad (1)$$

whereby, $H^{(I)}$ and $W^{(I)}$ are the multidimensional arrays and the network parameters at the I^{th} layer of the network, and σ is a non-linear perceptron, often a ReLU. A schematic of the GCNN architecture can be seen in Figure 1. This design aggregates the surrounding nodes to update the presentations of individual nodes in a graph.

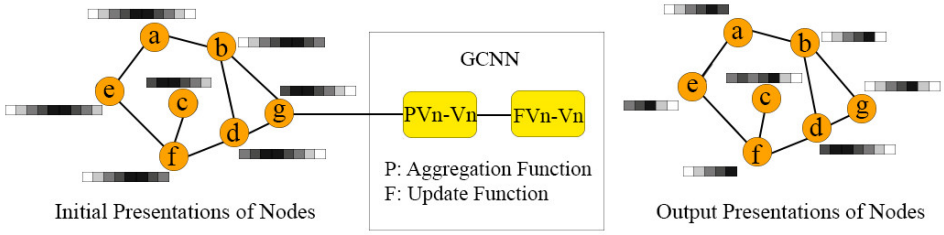

Figure 1. A GCNN architecture diagram with node presentations.

Concerning convolution, many academics have focused on generalizing convolution neural models so that they could function on structured graphs. As a result of their efforts, two approaches have been offered: spectral and spatial graph convolution neural networks. It is understood from the studies that the usage areas of the GCNN method are spread over a wide variety of areas. Examples of these are font recognition, graphical analysis of molecular structures, and traffic density estimates [32], they are divided into two different methods as a working principle. One is the spectral-based, and the other is the spatial-based method. This study focuses on the spatial-based method.

The GCNN method will be used on tagged image data in this study. Apart from traditional application methods, it aims to achieve faster results by converting graphical data into spatial structures. The reason for this preference is that the analysis stages of graphic data require a lot of computer power, and the costs for such processes are high for the sector, which is this paper's application area.

4. Research Model

4.1. Image Processing

Image processing is generally a method of signal processing. Images are used as data entry. The image expresses a two-dimensional x and y function. This x and y function is expressed in planes and coordinates [33]. There are two different shapes. The first of these is separated in itself as the gray image, aka black and white, and the other in color. As seen in Figure 2, the image processing process consists of 4 stages. These are, respectively, input, storage, processing (analysis), and output. The system as a whole is represented by a rectangle. The rectangle divided into 4 sub-systems represents the information flow between them. Raspberry Pi model 4; manufactured on 24 June 2019 with a Codex-A72 64 bit (ARMv8) core structure on Broadcom BCM2711 processor was used in this study [34]. It works at a speed of 1.5 GHz. There are 1, 2, 4, and 8 G.B. models. The 8 G.B. model was used in this project. The operating system is Raspbian.

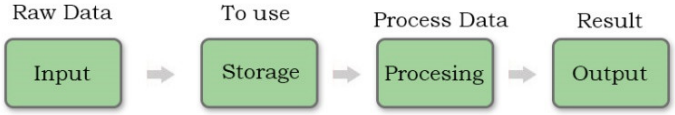

Figure 2. Information Process Cycle.

4.2. Hardware Build

4.2.1. Camera Interface

The camera module captures images over the C.S.I. connection. Standard camera types for Raspberry Pi are V2 and H.Q. cameras. These two cameras are the most used models. Apart from the standard cameras, a custom camera with high macro shooting was installed for the project. Besides having three different zoom support features, it is also possible to use the lenses of DSLR cameras with an adapter produced with the help of a 3D printer. Detailed features of this camera are shown in Figures 3 and 4.

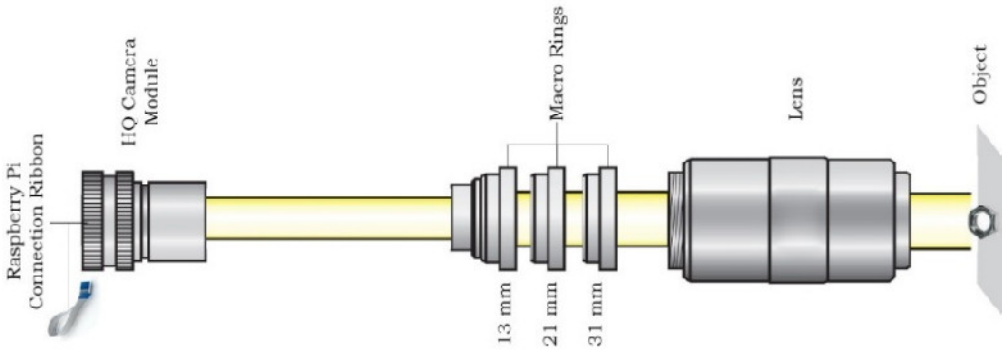

Figure 3. Raspberry Pi custom camera diagram.

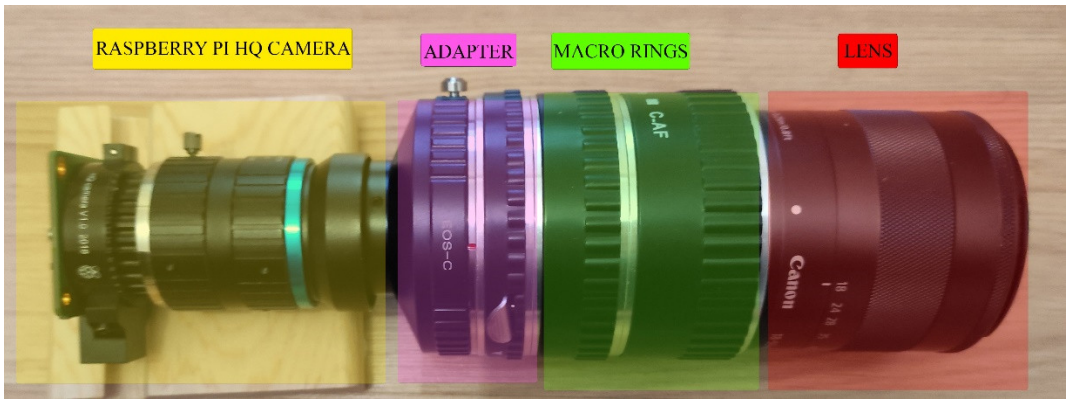

Figure 4. Real version of the camera that we built for our research.

4.2.2. Model Defect Inspections Process

Two different modules are placed on the Raspberry pi 4 in the established model. The first module is the motion sensor. The purpose of this module is to ensure that the part which passes over the production line gets triggered by the motion sensor when it comes under the camera, thereby capturing the product's image. The second module is the installation of a camera and its zoom apparatus specially created for this model. As seen in Figures 2 and 3, three different macro shooting options can be used with this camera system. After the image is captured, the process to transfer the captured image to the cloud or to the second computer begins. Here, the cloud feature is an option that can be applied to companies that produce large scale captured images. The images captured for medium and low-scale production companies will be stored on a local computer. In the third stage, artificial intelligence, which is trained based on previously captured images via DGNN, starts the analysis process to determine whether the instantaneously produced parts are produced with or without errors. According to the result, if the produced part does not have any errors, it will be labeled as good/or correct, and will then be separated from the production line.

5. Procedures

An automation system has been established to obtain the data set. As shown in Figure 5, when every piece passing through the production line instantly enters the darkroom, the surface photographs are taken by the camera whose top angle is adapted to 90 degrees from top to bottom with the motion sensor triggered. The surface photograph of the metal product taken for each piece is transferred to the second computer via raspberry pi via the local server. Later, by performing the analysis process on this computer, it is determined that the metal part produced is either faulty or error-free.

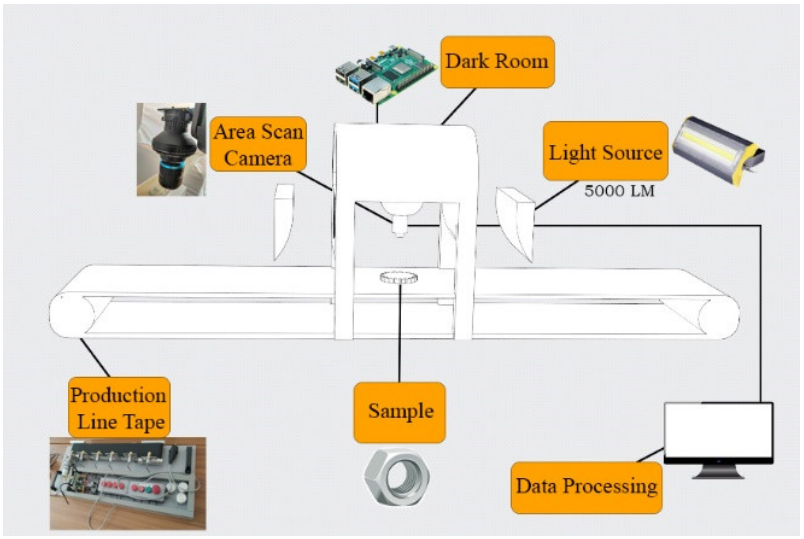

Figure 5. Principle of the system.

Figure 6 shows the system established for the article. Two different movable arms were placed on the production line. There is a moving motor on each arm. Through these motors, close or distant shots can be taken by moving back and forth on the surface of the metal part to be imaged. Due to the L.E.D. screen in the middle, the analysis results from the Raspberry Pi are reflected on the screen. As a result of the two light sources placed on both arms, a stable light image was obtained on the metal part. The robotic arm, controlled by a Raspberry Pi board, has a moving screen made of aluminum. By moving the camera forward and backward on the support pipe, both a close-up and a wide-angle shot can be taken by focusing light into a laser point.

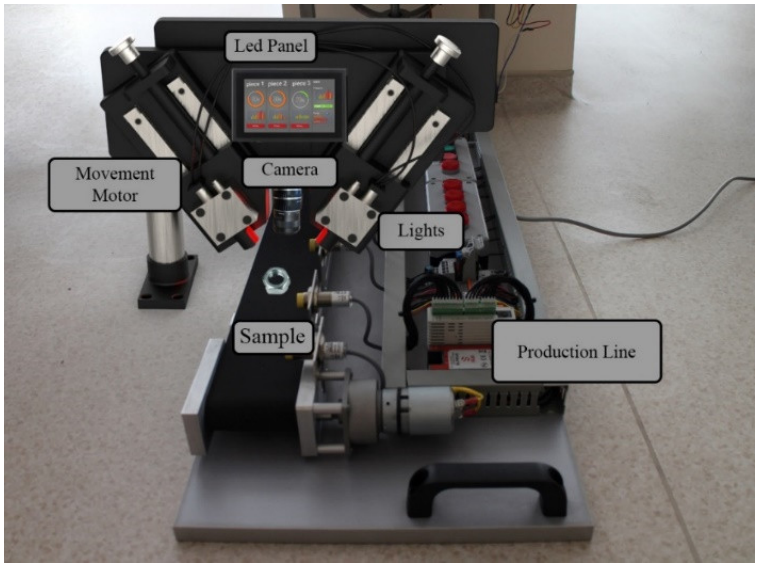

Figure 6. Original system.

A total of 1000 pieces of metal nuts previously produced were used for our data set. Scratches and cuts were made on the surface of 50 metal nuts, they were reserved for test analysis, and were included in the faulty production analysis. The images captured for this data set were recorded as 5184 × 3456 pixels, 18 m/pixels spatial resolution, and four different color channels. In Figure 7, five different metal nuts, which were randomly selected from among 50 different metal nuts, are shown as an example.

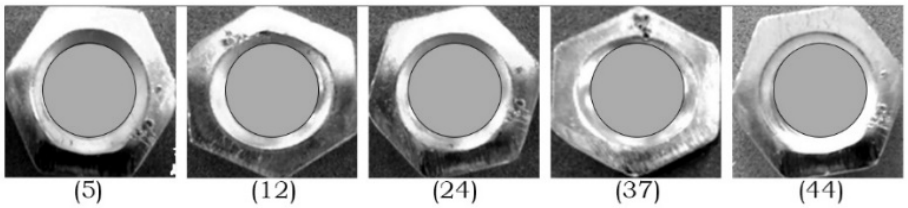

Figure 7. Five different randomly selected metal nuts.

Methodology

Inspired by the spatial domain method [35], the spatial term invokes pixels, which are the smallest elements of an image. All the colors directly execute these values.

The spatial domain method operates directly on pixels, and can be denoted by the expression:

$$g(x, y) = T[f(x, y)] \quad (2)$$

g(x, y): represents the output image after T is applied to the input image. f(x, y): input image, which describes pixel values in specific (x, y) coordinates, as shown in Figure 8.

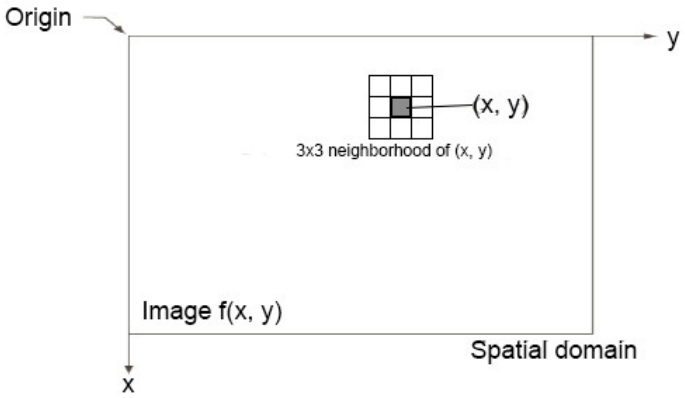

Figure 8. 3 × 3 neighborhood about a point in an image.

T: is an operator on f(x, y) specified over some neighborhood of (x, y) and also shows the operator of set input images, which calculates the sum of the pixels from n images. It can work as a gray-level function.

Normalization: if T(r) has the form shown in Figure 9, the effect of applying the transformation to every pixel of f to generate the corresponding pixels in g would indicate:

The dark area: the levels below (for this paper 90) k in the original image are faulty. The bright area: the levels above (0 < k < 90) k in the original image are good.

The grayscale image is used as the input to the algorithm. RGB (Red Green Blue) images are acquired in the acquisition task. These images need to be converted into grayscale images. The purpose of converting input images to grayscale is to minimize the different image value variances induced by different subject colors. If the algorithm is optimized to the RGB color setting, it will be very difficult to work with grayscale color.

We propose finding defect areas using the "k" parameter with the following formulation. In Figure 9, it is shown that 3 × 3 neighborhoods are divided into a square. However, in our model, each image taken is divided into 12 × 12 squares in the analysis phase through the spatial method, and defective parts on the piece's surface are detected as dark

and black with the light system created using 5000 lm value. All areas aim to have white and relative values. As shown in Figure 10, "k" (black) was determined as 90 percent and above as the reference value. k-value gives us the numerical expression between the black and white color values.

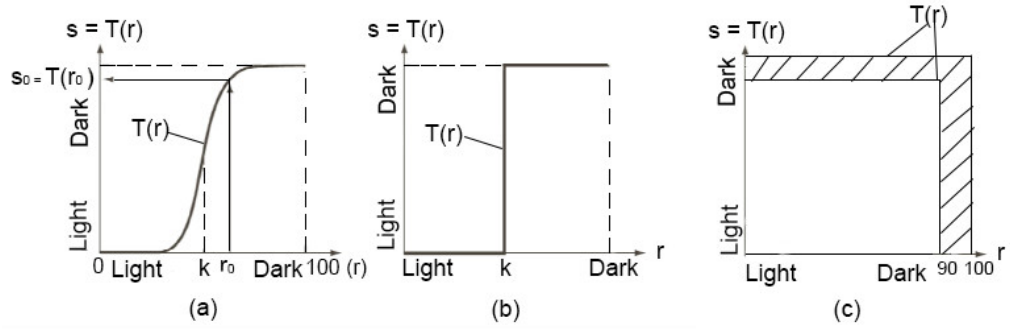

Figure 9. Level transformation functions values for grayscale images (**a**), (**b**)—Our proposed shown (**c**).

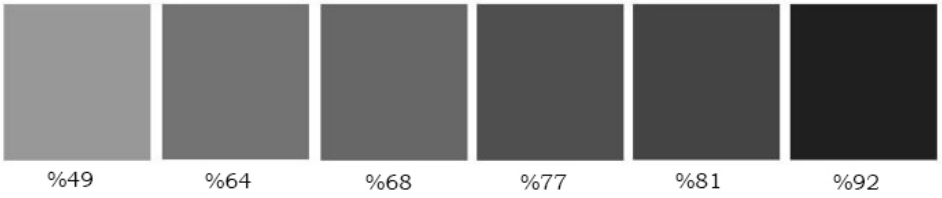

Figure 10. Based on a color palette and K percentages.

For example, the "k" value of a pure white color is "0", while the "k" value of a pure black color is "100". The values between them form gray tones. Due to the lighting provided by the light sources used, flat metal surfaces have a low gray ratio or tones close to white. When any damage, such as scratches or holes, are found on the surface, black and shaded areas appear on these surfaces. The "k" value helps us in terms of the quality of the surface and the detection of defects. The appropriate finding of the "k" is crucial. A high "k" value brings the result of "defect area". Otherwise, bad lights correspond to false defect areas or edges. Through the use of this method, it is possible to minimize the computational time.

6. Evaluated Algorithm

Before the analysis, 1000 different surface images of the total product were used. Based on the 4 to 1 rule, 800 images were used for the training phase, and 200 were used for the test phase. All analysis steps were completed using the Raspberian operating system and the TensorFlow framework on a Windows-based computer. A 64 GB ram and gtx 2080 graphics card with Core i7 was used as the computer processor. During the analysis, the iteration value was applied as 50, and the batch size was 128. Seven different models were established within the created models' framework. The analysis results made over the data augmentation and grayscale color layer are shown in Table 1. A consistent analysis method was determined by increasing the value ranges determined according to 14 different variables at the same rate. As can be seen in Table 2, 2D-CNN and YOLOv3 were used for model comparisons. Zhang et al. recently attempted to use the YOLOv3 network model for the surface defect detection of steel strips [36]. When Table 1 is examined, characteristics of data augmentation and histogram graphics are drawn to determine the color palette values of each image directly. These affect the analysis results and cause changes to the accuracy of the models. Average data augmentation values used for 2D-CNN and YOLOv3 based are shown in Table 1, which models data augmentation values.

Table 1. Comparison of different GCN models configuration.

	Configuration of GCN Models												
											Gaussian Noise		Speckle Noise
	Monochrome	Histogram Equalization	Max. Rotation	Max. Verical Shift	Max. Horizontal Shift	Min. Scale	Max. Scale	Max. Vertical Shear	Max. Horizontal Shear	Max. Brightness	Min.Std Dev.	Max. Std Dev.	Min. Std Dev.	Max. Std Dev.
Model 1	yes	yes	45	10	10	0.1	0.1	45	10	10	5	10	0.05	0.1
Model 2	yes	yes	90	20	20	0.2	0.2	55	20	20	10	20	0.1	0.15
Model 3	yes	yes	135	30	30	0.3	0.3	65	30	30	15	30	0.15	0.2
Model 4	yes	yes	180	40	40	0.4	0.4	75	40	40	20	40	0.2	0.25
Model 5	yes	yes	225	50	50	0.5	0.5	85	50	50	25	50	0.25	0.3
Model 6	yes	yes	270	60	60	0.6	0.6	95	60	60	30	60	0.3	0.35
Model 7	yes	yes	315	70	70	0.7	0.7	105	70	70	35	70	0.35	0.4

Table 2. Accuracy comparison of different models.

Accuracy Comparisons for Metal Nut Images. Bold Numbers Indicate the Best Performance.										
Class No.	**Class Name**	**Model 1**	**Model 2**	**Model 3**	**Model 4**	**Model 5**	**Model 6**	**Model 7**	**2D CNN**	**YOLOv3**
1	Nut	90.8816	75.9536	76.1821	72.9791	90.2698	88.9597	90.6628	87.9262	90.6145

Sample analysis results of metal nuts were found to be error-free due to the analysis shown in Figure 11. The highest k value determined on the 144 square area divided as 12 × 12 grid areas was 65. Due to this value, 90% and above, which was determined before the analysis, remained below the k value. This value was determined to be the part subject to the test without error.

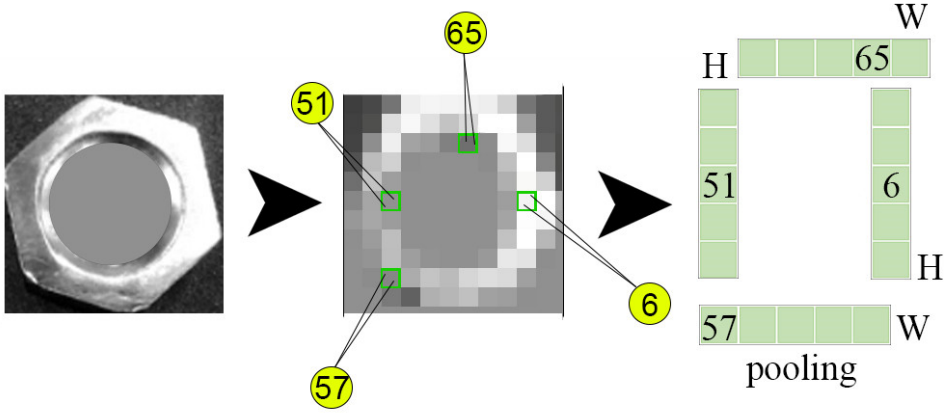

Figure 11. The detected k percentage values of the metal nut piece whose surface image is captured (belongs to the product without error).

The sample study of detecting defective parts in the analysis phase is shown in Figure 12. The error was detected in two different square areas belonging to 90% and above k value. Compared to the original pictures of the product subject to the test, the product's defective parts are seen because of the analysis. The effect of the algorithm on metal nuts is illustrated in Figure 13, and are also shown by images and their 3D surface plot graphics that are highlighted. Table 2 displays the outcomes of the effects that were caused by the numerical values that were established in Table 1. Figure 14 shows the comparison of the

analysis results of ten faulty and eight correct parts selected from a random sample group. Samples with yellow circles were detected without error. Samples with red circles were detected incorrectly by 14% and 39%, respectively.

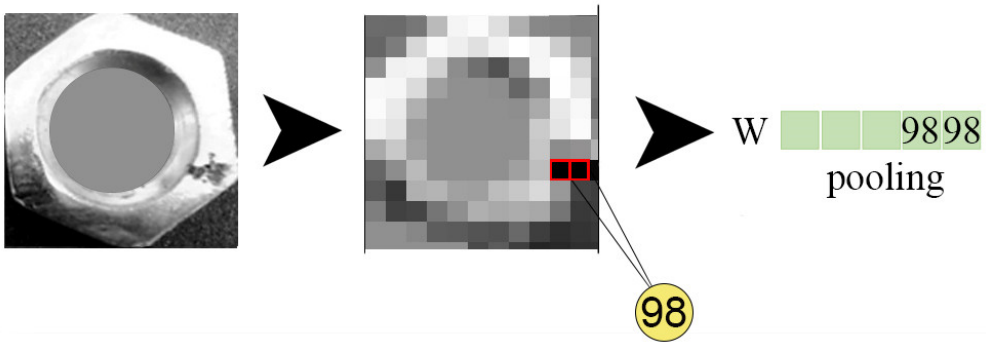

Figure 12. Detected k percentage values of the metal nut piece whose surface image is captured (belongs to the defective product).

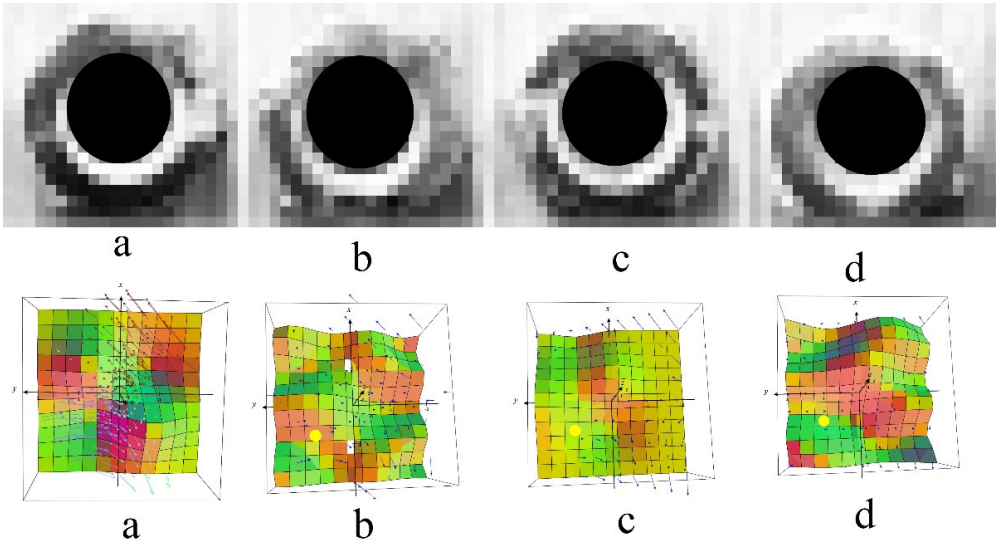

Figure 13. Analysis results of four different defective parts, with 3D graphic displays.

The performance values obtained from using the seven models designed during the study and the 2D-CNN [37], YOLOv3 method, which is currently applied, are shown in Table 2. Performance results between the models created in 10 and the 2D-CNN, YOLOv3 method currently used are given. You Only Look Once, Version 3 (YOLOv3) is a real-time object identification method that recognizes items in films, live feeds, or photos. It is sometimes known by its full name, You Only Look Once. When trying to identify an item, the YOLO machine learning algorithm consults the characteristics that have been learned by a convolution neural network. As a result, Model 1 gives the best performance in this study. In addition, when Table 2 is examined, it is seen that there are only four models that exceed 90%, and the weakest models are Models 2, 3, and 4, respectively. The average value of the performance results obtained from seven models is equal to 83.62%. At this time, the performance result obtained by Model 1 (90.88%) is better than that of models 2d-CNN and YOLOv3 (90.61%), which are currently used in autonomous robots [38] at a similar level.

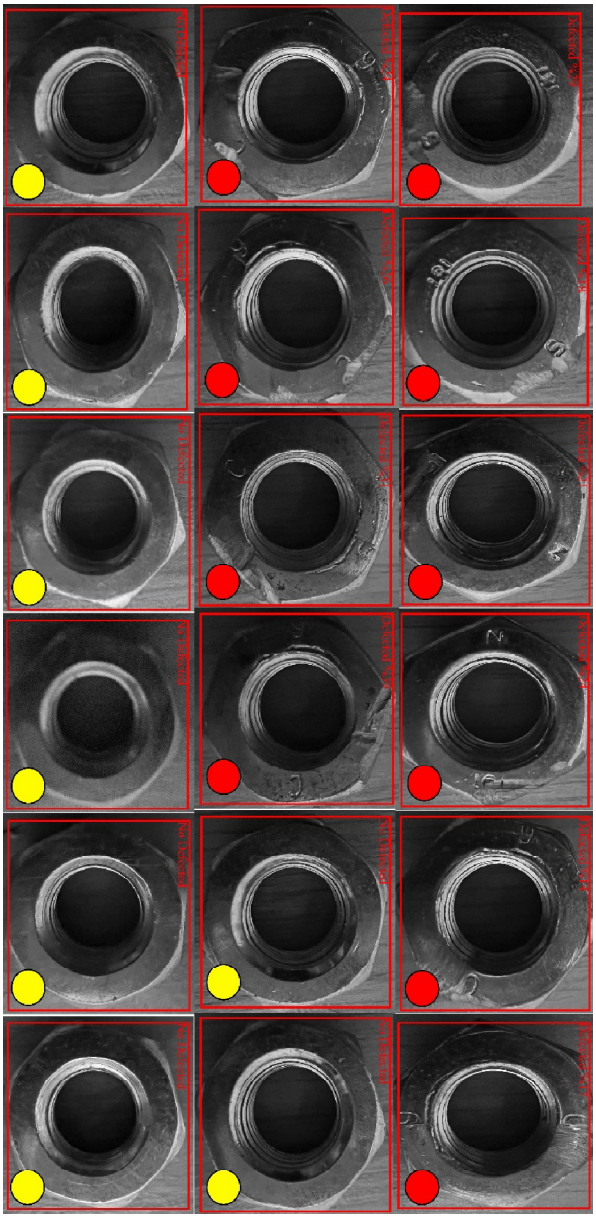

Figure 14. Detection results of faulty and defect-free parts were compared.

7. Conclusions and Discussions

This article consists of two parts. In the first part, a prototype system was set up; in the second part, the product's surface images were captured using this system. Analysis results were obtained using the GCN method, which has gained a new place recently in the literature. Based on the analysis results, it is seen that the process of learning whether the visuals of the product are faulty or error-free is more successful than other methods, due to the fixed light and fixed camera angle of one-dimensional images over one color. Table 2 gives 2.9554% more successful results than 2D-CNN and YOLOv3, which is another method added to the analysis for comparison. Apart from these, the defective surface is detected 18% faster than by other methods. The point recommended here is that the images of the product should be captured as grayscale.

The surface defect detection implied the creation of seven different algorithms that were tested, and the one with the best performance was indicated by the lowest error rate; however, the processing speed was also a qualification in the production line. The production speed and the type of chosen algorithms themselves were a challenge because of the low-cost hardware (lenses and computers) used.

Choosing a lens was another challenge because of the perspective angle; additionally, for image processing applications, one must examine the sensor that will be used. Sensor size and pixel size are of utmost priority in the selection process. Standard lenses produce images that perfectly match the object captured, including all details and brightness.

Apart from these, one of the most critical points is the subject of light. There should not be any shadow areas on the product's surface whose image is captured in the prototype work. When the shadow parts in the analysis are close to black tones, the system will label them as errors. It is necessary to use two light sources with 5000 lm light power, adjusted at an angle of 45 degrees on both sides, to prevent this problem. By doing this, no shadow areas will be created on the product whose image is captured. The main finding of this work is that the proposed method and system show excellent potential for deep learning for industrial applications, which generally have big-sized image datasets. In future studies, the same analysis method can be used in different color tones, and the probability of success can be compared with other methods. It is also believed that the proposed model can be extended for other manufacturing products.

We consider, therefore, that our research contributes to the mainstream of scientific research by guiding later studies on computer science, as well as being enlightening about computer vision.

Limitations of the study included the different graph convolutional neural network method settings, systems, and selection bias, which include different light source setups.

There are plans to include the cloud feature in the system for future studies.

Author Contributions: Conceptualization, Y.S.B. and B.S.; methodology, C.C.Ç.; software, Y.S.B.; writing—original draft preparation, Y.S.B.; writing—review and editing, C.C.Ç. and S.H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, H.; Pang, Y.; Hu, Q.; Liu, K. Solar cell surface defect inspection based on multispectral convolutional neural network. *J. Intell. Manuf.* **2020**, *31*, 453–468. [CrossRef]
2. Du, W.; Shen, H.; Fu, J.; Zhang, G.; He, Q. Approaches for improvement of the X-ray image defect detection of automobile casting aluminum parts based on deep learning. *NDT Int.* **2019**, *107*, 102–144. [CrossRef]
3. Ferguson, M.; Ak, R.; Lee, Y.T.T.; Law, K.H. Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning. *Smart Sustain. Manuf. Syst.* **2018**, *2*, 1007121126. [CrossRef]
4. Lv, X.; Duan, F.; Jiang, J.J.; Fu, X.; Gan, L. Deep active learning for surface defect detection. *Sensors* **2020**, *20*, 1650. [CrossRef]
5. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, San Diego, CA, USA, 20–26 June 2005. [CrossRef]
6. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
7. Wang, J.; Perez, L. The effectiveness of data augmentation in image classification using deep learning. *arXiv* **2017**, arXiv:1712.04621v1.
8. Perez, H.; Tah, J.H.M.; Mosavi, A. Deep learning for detecting building defects using convolutional neural networks. *Sensors* **2019**, *19*, 3556. [CrossRef]
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
10. Awan, M.J.; Rahim, M.S.; Salim, N.; Mohammed, M.A.; Garcia-Zapirain, B.; Abdulkareem, K.H. Efficient Detection of Knee Anterior Cruciate Ligament from Magnetic Resonance Imaging Using Deep Learning Approach. *Diagnostics* **2021**, *11*, 105. [CrossRef]
11. Hechtlinger, Y.; Chakravarti, P.; Qin, J. A generalization of convolutional neural networks to graph-structured data. *arXiv* **2017**, arXiv:1704.08165v1.

12. Lim, D.U.; Kim, Y.G.; Park, T.H. SMD Classification for Automated Optical Inspection Machine Using Convolution Neural Network. In Proceedings of the 3rd IEEE International Conference on Robotic Computing, I.R.C, Naples, Italy, 25–27 February 2019. [CrossRef]
13. Yang, Y.; Pan, L.; Ma, J.; Yang, R.; Zhu, Y.; Yang, Y.; Zhang, L. A high-performance deep learning algorithm for the automated optical inspection of laser welding. *Appl. Sci.* **2020**, *10*, 933. [CrossRef]
14. Lopac, N.; Hržić, F.; Vuksanović, I.P.; Lerga, J. Detection of Non-Stationary GW Signals in High Noise from Cohen's Class of Time-Frequency Representations Using Deep Learning. *IEEE Access* **2022**, *10*, 2408–2428. [CrossRef]
15. Lin, H.; Li, B.; Wang, X.; Shu, Y.; Niu, S. Automated defect inspection of L.E.D. chip using deep convolutional neural network. *J. Intell. Manuf.* **2019**, *30*, 2525–2534. [CrossRef]
16. Jamshidi, A.; Roohi, S.F.; Núñez, A.; Babuska, R.; De Schutter, B.; Dollevoet, R.; Li, Z. Probabilistic Defect-Based Risk Assessment Approach for Rail Failures in Railway Infrastructure. *IFAC-Pap. Online* **2016**, *49*, 73–77. [CrossRef]
17. Faghih-Roohi, S.; Hajizadeh, S.; Nunez, A.; Babuska, R.; De Schutter, B. Deep convolutional neural networks for detection of rail surface defects. In Proceedings of the International Joint Conference on Neural Networks, Vancouver, BC, Canada, 24–29 July 2016; pp. 2161–4407. [CrossRef]
18. Bruna, J.; Sprechmann, P.; LeCun, Y. Super-resolution with deep convolutional sufficient statistics. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings, San Juan, Puerto Rico, 2–4 May 2016.
19. Atwood, J.; Towsley, D. Diffusion-convolutional neural networks. In Advances in Neural Information Processing Systems. 2016. Available online: https://proceedings.neurips.cc/paper/2016/hash/390e982518a50e280d8e2b535462ec1f-Abstract.html (accessed on 24 September 2022).
20. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings, Toulon, France, 24–26 April 2017.
21. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A comprehensive survey on graph neural networks. *arXiv* **2019**, arXiv:1901.00596v4. [CrossRef]
22. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and deep locally connected networks on graphs. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings, Banff, AB, Canada, 14–16 April 2014.
23. Li, R.; Wang, S.; Zhu, F.; Huang, J. Adaptive graph convolutional neural networks. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, New Orleans, LO, USA, 2–7 February 2018.
24. Zhuang, C.; Ma, Q. Dual graph convolutional networks for graph-based semi-supervised classification. In Proceedings of the Web Conference 2018 World Wide Web Conference, W.W.W., Lyon, France, 23–27 April 2018. [CrossRef]
25. Rathi, P.C.; Ludlow, R.F.; Verdonk, M.L. Practical High-Quality Electrostatic Potential Surfaces for Drug Discovery Using a Graph-Convolutional Deep Neural Network. *J. Med. Chem.* **2020**, *63*, 8778–8790. [CrossRef]
26. Wang, Y.; Gao, L.; Li, X.; Gao, Y.; Xie, X. A New Graph-Based Method for Class Imbalance in Surface Defect Recognition. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–16. [CrossRef]
27. Wang, Y.; Gao, L.; Gao, Y. A new graph-based semi-supervised method for surface defect classification. *Robot. Comput. Integr. Manuf.* **2021**, *68*, 102083. [CrossRef]
28. Villalba-Diez, J.; Schmidt, D.; Gevers, R.; Ordieres-Meré, J.; Buchwitz, M.; Wellbrock, W. Deep learning for industrial computer vision quality control in the printing industry 4.0. *Sensors* **2019**, *19*, 3987. [CrossRef]
29. Wang, M.; Li, Y.; Zhang, Y.; Jia, L. Spatio-temporal graph convolutional neural network for remaining useful life estimation of aircraft engines. *Aerosp. Syst.* **2020**, *4*, 29–36. [CrossRef]
30. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G. Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; Volume 70.
31. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling Relational Data with Graph Convolutional Networks. In *Lecture Notes in Computer Science*; Including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics; Springer: Berlin/Heidelberg, Germany, 2018. [CrossRef]
32. Mou, L.; Lu, X.; Li, X.; Zhu, X.X. Nonlocal Graph Convolutional Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8246–8257. [CrossRef]
33. Jain, V.; Seung, H.S. Natural image denoising with convolutional networks. Advances in Neural Information Processing Systems. In Proceedings of the 2008 Conference NIPS, Vancouver, BC, Canada, 8–11 December 2008.
34. Shilpashree, K.S.; Lokesha, H.; Shivkumar, H. Implementation of Image Processing on Raspberry Pi. *Int. J. Adv. Res. Comput. Commun. Eng* **2015**, *4*, 199–202. [CrossRef]
35. Gonzalez, R.C.; Woods, R.E.; Masters, B.R. Digital Image Processing, Third Edition. *J. Biomed. Opt.* **2009**, *14*, 029901. [CrossRef]
36. Zhang, J.; Kang, X.; Ni, H.; Ren, F. Surface defect detection of steel strips based on classification priority YOLOv3-dense network. *Ironmak. Steelmak.* **2020**, *48*, 547–558. [CrossRef]

37. Zhou, Y.; Yu, L.; Zhi, C.; Huang, C.; Wang, S.; Zhu, M.; Ke, Z.; Gao, Z.; Zhang, Y.; Fu, S. A Survey of Multi-Focus Image Fusion Methods. *Appl. Sci.* **2022**, *12*, 6281. [CrossRef]
38. Xu, Z.F.; Jia, R.S.; Liu, Y.B.; Zhao, C.Y.; Sun, H.M. Fast method of detecting tomatoes in a complex scene for picking robots. *IEEE Access* **2020**, *8*, 55289–55299. [CrossRef]