

Article

# Dual-Kernel-Based Aggregated Residual Network for Surface Defect Inspection in Injection Molding Processes

Hwaseop Lee  and Kwangyeol Ryu \* 

Manufacturing Data Analytics Laboratory, Department of Industrial Engineering, Pusan National University, Busan 46241, Korea; hslee@pusan.ac.kr

\* Correspondence: kyryu@pusan.ac.kr; Tel.: +82-51-510-2473

Received: 30 August 2020; Accepted: 16 November 2020; Published: 18 November 2020



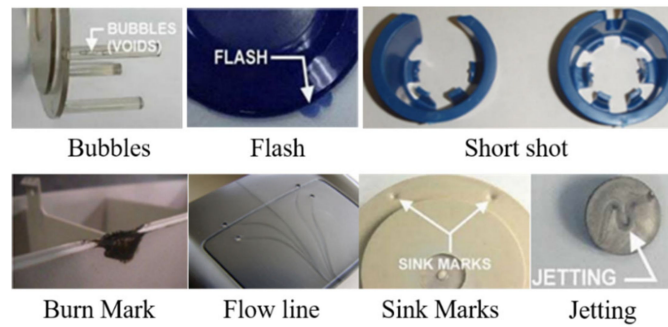
**Abstract:** Automated quality inspection has been receiving increasing attention in manufacturing processes. Since the introduction of convolutional neural networks (CNNs), many researchers have attempted to apply CNNs to classification and detection of defect images. However, injection molding processes have not received much attention in this field of research because of product diversity, difficulty in obtaining uniform-quality product images, and short cycle times. In this study, two types of dual-kernel-based aggregated residual networks are proposed by utilizing a fixed kernel and a deformable kernel to detect surface and shape defects of molded products. The aggregated residual network is selected as a backbone, and a fixed-size, deformable kernel is applied for extracting surface and geometric features simultaneously. Comparative studies are conducted by including the existing research using the Weakly Supervised Learning for Industrial Optical Inspection dataset, which is a DAGM dataset. A case study reveals that the proposed method is applicable for inspecting the quality of injection molding products with excellent performance.

**Keywords:** aggregated residual learning network; deformable convolutional network; defect classification; injection molding process; surface inspection

## 1. Introduction

Quality inspection in manufacturing industries is one of the essential processes that can reduce the risk and cost of failure of delivery to customers. Manufacturing companies conduct a total inspection when the product is a part directly used by customers and related to human safety. Most small- and medium-sized enterprises (SMEs) that produce parts with injection molding operations continue to inspect such parts manually to classify defective products. Manual inspection is subject to human error caused by fatigue in continuous repetitive work [1]. In such an environment, injection molding companies consume enormous time, effort, and cost for manual inspection. To solve this problem, automated quality inspection for detecting defects has been developed by many researchers to reduce human effort during inspection processes [2,3].

Injection molding is a crucial underlying technology that can produce a wide variety of products that differ significantly in size, complexity, and application [4]. Injection molding companies must conduct a total inspection of products directly related to customer safety to ensure the proper functionality and appearance of the products. The primary defect types in injection molding products are unformed shapes (e.g., bubbles, flash, and short shots) and poor surface appearance (e.g., burn marks, flow lines, sink marks, and jetting). As depicted in Figure 1, most defects in injection molding products can be found on the surface of the parts. Therefore, automated visual inspection using image processing techniques such as feature extraction, classification, and object detection is applicable [5–7].



**Figure 1.** Primary defects of injection molded products.

However, the application of image processing techniques for automated visual inspection is difficult because of the characteristics of the injection molding process. First, multiple products are usually produced in one cycle in one mold. Because the mold ejects several products during one cycle time, obtaining uniform-quality image data of multiple parts in the manufacturing process is challenging. Second, the injection process has a short cycle time. According to the aforementioned characteristics, the injection molding process requires more effective and faster feature-extraction methodologies for a visual inspection system. In this study, a dual-kernel-based aggregated residual network (ResNeXt) is proposed to solve the defect detection problem for the injection molding process. ResNeXt has been modified accordingly to achieve better detection performance while decreasing network complexity; this forms the backbone of the proposed model [8]. In addition, a deformable kernel is applied for robustness in processing scaled and rotated defect images [9,10].

The remainder of this paper is organized as follows. Section 2 presents a literature review on surface defect detection, ResNeXt, and the Deformable convolution network (Deformable ConvNet), which are core concepts behind the defect-detection model proposed in this study. Section 3 describes the details of the proposed methodologies and verification experiments using the DAGM 2007 competition dataset (Weakly Supervised Learning for Industrial Optical Inspection). Here, DAGM stands for Deutsche Arbeitsgemeinschaft für Mustereerkennung which means that the German chapter of the International Association for Pattern Recognition. A case study of an experiment on defect inspection using real data of an injection molding product is presented in Section 4, and Section 5 presents concluding remarks.

## 2. Literature Review

### 2.1. Surface Defect Detection

Several models of defect detection for industrial products have been developed since the convolutional neural network (CNN) was developed by Lecun et al. [11]. The CNN model has led to significant breakthroughs in computer vision and is widely used in a variety of applications such as image classification [12], image segmentation [13], and object tracking [14]. Surface defect detection is a technology that detects failures in the appearance of fabric, metal, wooden, and plastic products via image processing technologies [15–17]. Although the targets can be different, surface defect detection is a feature-extraction problem to identify anomalies that can be distinguished from textures. Algorithms to extract features from textures for detecting surface defects can be defined into four categories: statistical, structural, filter-based, and model-based approaches [18]. Representatively, statistical and filter-based approaches have been prevalent. For example, histogram properties categorized in the statistical approach have been applied to various studies and proved to have high performance even at low cost and effort [19,20]. Among joint spatial/spatial-frequency methods categorized in filter-based approaches, Gabor transforms (using modulated Gaussian filters) are widely used because they are similar to the human visual system [21]. After the CNN was developed, a filter-kernel-based neural network was proposed, and CNN-based feature-extraction techniques have been rapidly developed in image processing and machine learning research fields. Ren et al. [22] applied a generic deep-learning

approach based on the CNN model for automated surface inspection. Staar et al. [23] performed anomaly detection for industrial surface inspection by learning the deep matrix using a triplet network, which is a modified CNN model. Wang et al. [24] proposed a twofold joint detection model inspired by a CNN for classifying industrial surface inspection. Tao et al. [25] proposed a cascaded autoencoder architecture based on a CNN for segmenting and localizing multiple defects from industrial product data. In addition, many researchers have proposed various powerful CNN-based models to solve image classification problems or defect localization problems for diverse industrial surface defects [26,27]. Representatively, research on concrete crack detection using CNN-based models has been actively conducted in recent years. Deng et al. [28] applied an ad-hoc faster region-based CNN (faster R-CNN) to distinguish between handwriting scripts and crack on a concrete surface. Chun et al. [29] detected cracks on the concrete surface using a light gradient boosting machine (LightGBM) considered pixel values and geometric shapes. You only look once (YOLO), VGG Net, Inception Net, Mask R-CNN were frequently applied to detect concrete cracks in research on civil and infrastructure engineering [30,31]. However, research considering the characteristics of injection molding processes and products is yet to receive much attention. In this study, a novel defect classification model is proposed for the injection molding process by redesigning ResNeXt and Deformable ConvNet.

## 2.2. ResNeXt

ResNeXt is inherited from residual networks (ResNets) proposed for image recognition [32]. The ResNet using “shortcut connections” prevents gradient vanishing, exploding, or degradation, which can occur with increase in number of layers of a deep network. This methodology has proved that optimizing residual mapping is easier than optimizing the original unreferenced mapping. The ResNet has the advantage that even deeper networks can be easily optimized, and the accuracy can be significantly improved because of the increased depth in the networks without notorious problems such as gradient vanishing, exploding, and accuracy degradation.

The ResNeXt proposed by Xie et al. [8] utilizes cardinality, which is a modulated residual bottleneck block to maintain simplicity and increase the accuracy of classification performance simultaneously. Cardinality is formulated in the network-in-neuron form and performs splitting, transforming, and aggregating in the module. This is a network model that proves that it is more efficient to increase cardinality, which is the bottleneck building block, than the depth and width of a network. In addition, ResNeXt derives better performance from simple structural changes, including shortcut connections and cardinality, on the convolutional network. The numbers of hyperparameters of ResNet-50 and ResNeXt-50 with the same stacked layers are  $25.5 \times 10^6$  and  $25.0 \times 10^6$ , respectively. ResNeXt is proven to improve accuracy while maintaining the simplicity of the model complexity and number of parameters. The structural simplicity offers the possibility of applying a deformable kernel to ResNeXt.

## 2.3. Deformable ConvNet

CNNs are limited in defining geometric transformation because of their fixed kernel structures. There are two ways to solve geometric variation or transformation in object scale, pose, viewpoint, and part deformation. The first one is to augment datasets, but this entails a high cost of training and complex model parameters. The second one is to change fixed geometric feature maps into feature maps that allow for the adaptive determination of scales or receptive field sizes. To overcome these limitations, Deformable ConvNet, proposed by Dai et al. [9], applies a deformable convolutional layer and a deformable region-of-interest (ROI) pooling layer using a position-changeable kernel to enhance the capability of modeling geometric transformation on the CNN structure. Both deformable convolution and ROI pooling modules are ready to be applied to the structure of plain CNNs and augment the spatial sample location with additional offset. These modules learn the offset from target images by extracting features without space restrictions. Deformable ConvNet has proven to be an effective structure for feature extraction, object segmentation, and detection.

Deformable ConvNet V2, proposed by Zhu et al. [10], expanded the utilization of deformable convolution layers with offset learning capacity to control sampling over a broader range of feature levels. The modulation mechanism in deformable convolution modules is expanded not only to offset learning but also to a mechanism for learning the amplitude of features. Deformable ConvNet V2 utilizes R-CNN, ResNet, and ResNeXt as the backbone to verify better image classification ability compared to regular CNN and Deformable ConvNet. The improved model adds more convolutional layers than the previous model and improves ability of image classification by introducing the concept of R-CNN feature mimicking. In addition, this model modulates the amplitude of the feature to set a particular location to an uninterested domain, giving the freedom to adjust the spatial location. This study proved that Deformable ConvNet V2 with the backbone of ResNeXt exhibited better accuracy than regular CNN and Deformable ConvNet with ResNet in image classification and object detection [8].

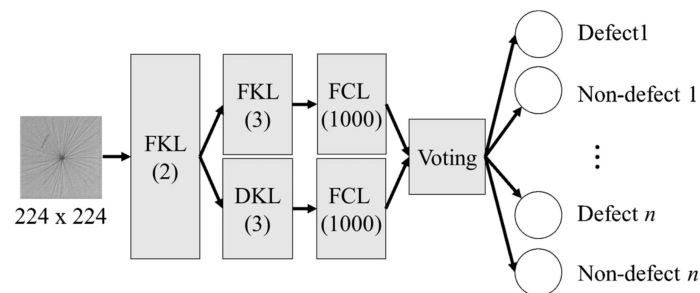
### 3. Dual-Kernel-Based Aggregated Residual Networks

Two types of dual-kernel-based aggregated residual networks (DK-ResNeXt) are proposed herein. The first type is an ensemble model that improves defect detection accuracy through weighted voting. This model consists of identical layers using different kernel types (i.e., the fixed kernel and deformable kernel). The second type is a multimodal network composed of a mainframe and subframe. The fixed kernel network (mainframe) and deformable kernel (subframe) networks have independent output classes. This model combines the classification results of the two networks to improve defect detection accuracy.

#### 3.1. Design of the DK-ResNeXt

##### 3.1.1. Parallel Ensemble ResNeXt

A parallel ensemble ResNeXt (PE-ResNeXt) model, as shown in Figure 2, utilizes ResNeXt as the backbone of the network. The fixed and deformable kernel layers are structurally contrasted.



**Figure 2.** Schematic of the PE-ResNeXt architecture.

PE-ResNeXt has an input image size of  $224 \times 224$  pixels. In Figure 2, FKL (2) indicates two fixed kernel layers, and DKL (3) indicates three deformable kernel layers. FCL (1000) is a fully connected layer composed of 1000 nodes. This network applies weighted voting for ensemble classification. Each independently trained model proposes a classification result for each class. As an ensemble model of networks, it has structural simplicity with relatively better classification accuracy compared to deep and wide CNNs. We intervene to mediate the voting weight when a specific model performs better classification results in a specific class. After two models derive the classification results, the final classification results are derived by combining the two results by weighted voting. If the results of the two models are equal, the model submits the result as it is. Otherwise, the model submits the results of the model with high prediction performance.

Detailed model information is presented in Table 1. We modified the structure based on ResNeXt-50 ( $32 \times 4d$ ) and 101 ( $32 \times 8d$ ), which are proven to have the best detection capability among ResNeXt

structures. FKL1 ( $7 \times 7$ , 64, stride 2) indicates that the  $7 \times 7$  kernel performs convolution operations with stride 2. This layer creates 64 channels and downsizes the input image to the  $112 \times 112$  feature maps. FKL2 ( $3 \times 3$  max pool, stride 2) performs max-pooling using a  $3 \times 3$  fixed kernel with a stride of 2.  $[C = 32]$  in FKL2 is 32 bottleneck blocks defined as the cardinality of ResNeXt.  $[\ ] \times 3$  indicates three ResNeXt modules of cardinality 32. FKL3 and DKL3 in the third layer utilize the same  $56 \times 56$  feature maps extracted from the second FKL. This model trains the same feature maps on two independent models with different kernels from the third layer. FKL is not sufficient for extracting the geometrical features of an image compared to a deformable kernel [9,10]. However, we expect that FKL can effectively extract its distinctive features, such as different background textures. DKL has proven to be able to extract geometrical features more effectively than fixed kernels. We assume that DKL can effectively classify sketches appearing on the surface and defects with unformed shapes. FCL has a global average pooling of 1000 nodes, and the activation function of FCL is the softmax function.

Table 1. Detailed structure of PE-ResNeXt.

Stage	Output	PE-ResNeXt-50 ( $32 \times 4d$ )	PE-ResNeXt-101 ( $32 \times 8d$ )
FKL1	$112 \times 112$	$7 \times 7$ , 64, stride 2	
		$3 \times 3$ max pool, stride 2	
FKL2	$56 \times 56$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 256 \end{bmatrix}, C = 32 \times 3$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 256 \end{bmatrix}, C = 32 \times 3$
FKL3	$28 \times 28$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 256 \end{bmatrix}, C = 32 \times 4$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 256 \end{bmatrix}, C = 32 \times 4$
DKL3			
FKL4	$14 \times 14$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 1024 \end{bmatrix}, C = 32 \times 6$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 1024 \end{bmatrix}, C = 32 \times 23$
DKL4			
FKL5	$7 \times 7$	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024 \\ 1 \times 1, 1024 \end{bmatrix}, C = 32 \times 3$	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024 \\ 1 \times 1, 1024 \end{bmatrix}, C = 32 \times 3$
DKL5			
FCL	$1 \times 1$	Global average pooling, 1000-d, softmax	Global average pooling, 1000-d, softmax

### 3.1.2. Double-Frame ResNeXt

The double-frame ResNeXt (DF-ResNeXt) is inspired from a fast and robust CNN using the twofold approach proposed by Wang et al. [24]. DF-ResNeXt, illustrated in Figure 3, has a mainframe composed of FKL (5) and FCL (1000) and a subframe composed of FKL (2), DKL (2), and FCL (1000).

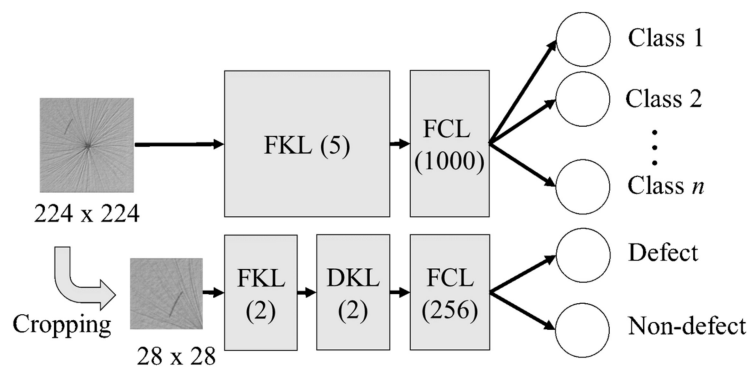


Figure 3. Schematic of the DF-ResNeXt architecture.

The mainframe inputs a  $224 \times 224$  image. The network structure of the mainframe is the same as that of ResNeXt presented in Table 1. The input size of the images for the subframe was  $28 \times 28$ . The mainframe classifies classes, and the subframe determines the defects. The decision of two frames is combined to determine which class has defects. Detailed model information is listed in Table 2.

**Table 2.** Detailed structure of the subframe of DF-ResNeXt.

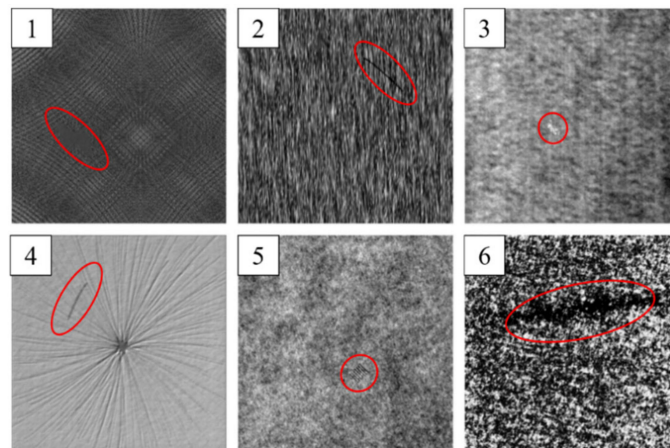
Stage	Output	DF-ResNeXt-50 (32×4d)	DF-ResNeXt-101 (32×8d)
FKL1	$28 \times 28$	$3 \times 3, 32, \text{stride } 1, \text{padding } 1$	$3 \times 3, 32, \text{stride } 1, \text{padding } 1$
FKL2	$14 \times 14$	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32, C = 32 \\ 1 \times 1, 64 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32, C = 32 \\ 1 \times 1, 64 \end{bmatrix} \times 4$
		$2 \times 2 \text{ max pool, stride } 2$	$2 \times 2 \text{ max pool, stride } 2$
DKL1	$14 \times 14$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 128, C = 32 \\ 1 \times 1, 128 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 128, C = 32 \\ 1 \times 1, 128 \end{bmatrix} \times 23$
DKL2	$7 \times 7$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128, C = 32 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128, C = 32 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
FCL	$1 \times 1$	Global average pooling, 256-d, softmax	Global average pooling, 256-d, softmax

FKL1 ( $3 \times 3, 32, \text{stride } 1, \text{padding } 1$ ) indicates that the  $3 \times 3$  kernel performs convolution operations with one stride and one padding in the first fixed kernel layer. The subframe is composed of only FKL and DKL. FCL has 256 nodes, and the activation function is the softmax function. We cropped the image to emphasize the defect features presented in various forms, and we assumed that the cropped image can further improve the performance of DKL. We assumed that the cropped input images emphasized the defect features. Therefore, we expected to be able to distinguish various defects using a simple network structure.

### 3.2. The Dataset

A DAGM dataset, Weakly Supervised Learning for Industrial Optical Inspection, provided by the German Chapter of the European Neural Network Society, is an essential industrial image processing dataset that represents characteristics similar to those of real-world problems. It consists of 10 classes of datasets, each consisting of 1000 images of background textures without defects and 150 images with one labeled defect. Figure 4 shows the image samples of six classes with a pixel size of  $512 \times 512$ . The DAGM dataset has two critical issues, which make defect detection challenging. The first issue is local textural irregularities, which is the primary concern for most visual surface inspection applications. The other issue is the global deviation of color and texture, where regional patterns or textures do not exhibit abnormalities. Various researchers have conducted studies on this dataset for feature extraction by applying statistical and filter-based approaches, including CNNs. Existing studies have proven that feature-extraction techniques using Weibull features [33], scale-invariant feature transforms (SIFT), artificial neural networks (ANNs) [34], statistical features [35], and CNNs [24,36] are sufficient for detecting defects. We conducted comparative studies based on the existing research.

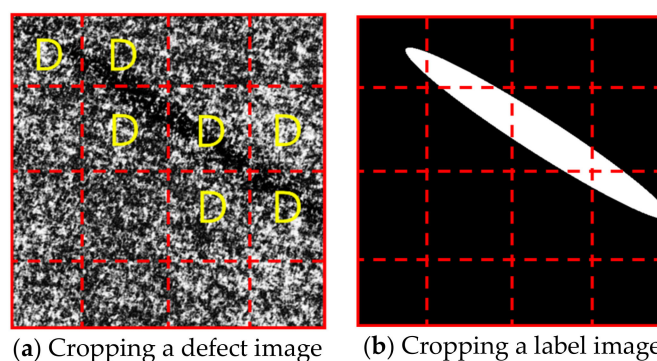




**Figure 4.** Image samples of six classes from the DAGM dataset.

### 3.3. Training Details

The DAGM dataset has six classes, and each class has defect and nondefect images. Therefore, PE-ResNeXt has 12 output classes (defect and nondefect for each of the 6 classes). DF-ResNeXt has eight output classes. The mainframe classifies six classes, and the subframe classifies only the defect or nondefect classes. We selected 70% and 30% of the dataset for training and validation datasets, respectively. Therefore, the numbers of images for training and validation were 700/106 (nondefect/defect images) and 300/44, respectively. To reduce model computation and cost while increasing the number of training images, the training image was cropped to a pixel size of  $128 \times 128$ . Therefore, we augmented the number of images from 700/106 to 11,200/1696 (nondefect/defect images). The defect image was automatically cropped using the label image (see Figure 5). The pixel value of the label image was only 0 or 255. A pixel value of 255 indicates the defect boundary. The training image and the label image were cropped simultaneously, and the defective part was automatically labeled with a value of 255. To prevent overfitting, the cropped images were augmented by flipping, rotating, and scaling with a 70% probability. The number of images was increased for the training purpose to 34,300/18,656 (nondefect/defect images) for each class. Training images were resized to  $224 \times 224$  for the input of PE-ResNeXt and the mainframe of DF-ResNeXt. For the subframe of DF-ResNeXt, we downsized the training images to  $28 \times 28$ .



**Figure 5.** Example of defect image cropping.

The cross-entropy function was used as a loss function. The activation functions of FKL and DKL were rectified linear units. The optimizer was a stochastic gradient descent with a learning rate of 0.01, momentum of 0.5, batch size for learning of 384, and 10 epochs. We conducted transfer learning for fine tuning the pretrained ResNeXt 50 and 101 models implemented by PyTorch. Therefore, our model is quick and easy to train and test. Deformable ConvNet V2 was implemented by modifying the

MMdetection [37] and pytorch-deform-conv-v2 [38]. The training environment employed two GPUs (NVIDIA TITAN RTX) for parallel computing.

### 3.4. Experimental Results

The PE-ResNeXt model was validated using the DAGM dataset. Table 3 lists the experimental results of comparing the basic ResNeXt-50 ( $32 \times 4d$ ), ResNeXt-101 ( $32 \times 8d$ ), Deformable ConvNet V2, and our proposed PE-ResNeXt models. As listed in Table 3, the highest accuracy is 99.97% for the PE-ResNeXt-101 ( $32 \times 8d$ ) model.

**Table 3.** Experimental results for PE-ResNeXt.

Model	Accuracy (%)
ResNeXt-50 ( $32 \times 4d$ )	98.79
ResNeXt-101 ( $32 \times 8d$ )	99.42
Deformable ConvNet V2	97.94
PE-ResNeXt-50 ( $32 \times 4d$ )	98.83
PE-ResNeXt-101 ( $32 \times 8d$ )	99.97

Table 4 presents the experimental results of the DF-ResNeXt model. Output class 6 is the classification-test result for the background texture of each class, and class 2 is the experimental result of classifying defects and nondefects. Output class 6/2 is the experimental result of a test that simultaneously detects six classes and two defects. The highest accuracy achieved is 100% with DF-ResNeXt-101 ( $32 \times 8d$ ). As listed in Table 3, ResNeXt has better classification accuracy than Deformable ConvNet V2. According to Table 4, Deformable ConvNet V2 exhibits better performance than ResNeXt when experiments are conducted for each class and defect. In particular, in defect classification, Deformable ConvNet V2 performs much better than ResNeXt. Therefore, we assume that the deformable kernel performs better than the fixed kernel for the distinction between texture and pattern.

**Table 4.** Experimental results for the DF-ResNeXt.

Output Classes	Model	Accuracy (%)
6	Mainframe [ResNeXt-50 ( $32 \times 4d$ )]	99.42
	Mainframe [ResNeXt-101 ( $32 \times 8d$ )]	100
	Subframe [Deformable ConvNet V2]	100
2	Mainframe [ResNeXt-50 ( $32 \times 4d$ )]	93.62
	Mainframe [ResNeXt-101 ( $32 \times 8d$ )]	93.62
	Subframe [Deformable ConvNet V2]	100
6/2	DF-ResNeXt-50 ( $32 \times 4d$ )	96.14
	DF-ResNeXt-101 ( $32 \times 8d$ )	100

Figure 6 presents the training, validation loss, and accuracy of the proposed model. The results show a decrease in training, validation loss, and an increase in accuracy of the DF-ResNeXt-101 ( $32 \times 8d$ ) and PE-ResNeXt-101 ( $32 \times 8d$ ). The DF-ResNeXt-101 ( $32 \times 8d$ ) performed low training and validation loss from the first epoch and obtained the first highest accuracy at five epochs. A bit of overfitting occurred at six epochs, but validation loss was decreased from seven epochs. The computation time was 9718 s (2 h, 41 min, and 58 s). The PE-ResNeXt-101 ( $32 \times 8d$ ) exhibited higher training and validation loss values than the DF-ResNeXt-101 ( $32 \times 8d$ ). No overfitting or underfitting occurred. The accuracy gradually increased, but the maximum accuracy did not exceed 99.97% even more epochs were added. The computation time was 9519 s (2 h, 38 min, and 39 s). The computation time of the DF-ResNeXt 101 ( $32 \times 8d$ ) was longer than the PE-ResNeXt-101 ( $32 \times 8d$ ), about 2%, but the accuracy of the DF-ResNeXt 101 ( $32 \times 8d$ ) was 0.03% higher.



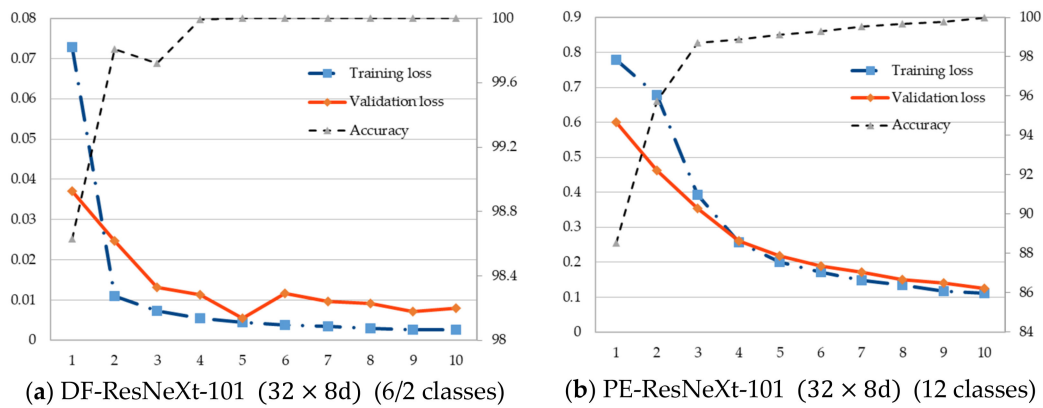


Figure 6. Training, validation loss, and accuracy (%).

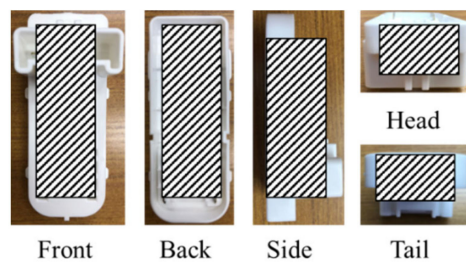
Table 5 provides a comparative summary of the true positive rate (TPR), true negative rate (TNR), and accuracy between the proposed models and models from previous studies. The PE-ResNeXt and DF-ResNeXt models proposed in this study have 12 and 8 final output nodes, respectively, and their structural characteristics are different. PE-ResNeXt, which has 12 output classes, has a 99.9% accuracy. This is 0.7% higher than the 99.2% value of DCNN [36] and 0.1% higher than that of FR-CNN [24]. DF-ResNeXt performed 0.2% better than FR-CNN. The PE-ResNeXt model, proposed by Weimer et al. [36], is an advanced DCNN model with 12 output nodes. DF-ResNeXt is an advanced twofold CNN model proposed by Wang et al. [24]. A conclusion can be drawn that both proposed models are able to effectively classify images by applying a basic CNN. In this study, we build a more effective model and maintain better classification performance while maintaining the structural characteristics of DCNN and FR-CNN.

Table 5. Comparison of accuracy (%) of the proposed and previous models.

Classes	ResNeXt-101 (32 × 8d) (6/2 Classes)	PE-ResNeXt-101 (32 × 8d) (12 Classes)	FR-CNN [24] (6/2 Classes)	DCNN [36] (12 Classes)	SIFT/ANN [22] (12 Classes)
<b>TPR</b>					
1	100	100	100	100	98.9
2	100	100	100	100	95.7
3	100	100	100	95.5	98.5
4	100	99.3	100	100	-
5	100	100	99.7	98.8	98.2
6	100	100	100	100	99.8
<b>TNR</b>					
1	100	100	100	100	100
2	100	100	100	97.3	91.3
3	100	100	100	100	100
4	100	100	93.2	98.7	-
5	100	99.9	100	100	100
6	100	100	100	99.5	100
<b>Average accuracy</b>	100	99.9	99.8	99.2	98.2

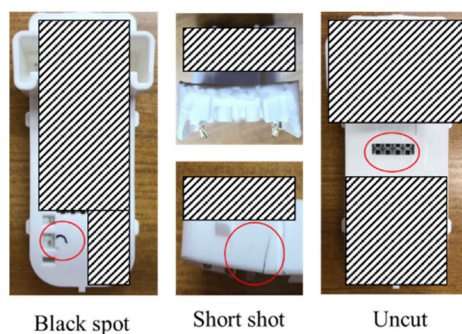
#### 4. Case Study

In this section, we present a case study that performs defect inspection with the proposed model using real injection-molded products. The target product presented in Figure 7 is a signal switch used for the electronic equipment operation inside a vehicle. Because the target product is a product that is directly related to the driver’s safety, it has to be thoroughly tested. The target product is partially blinded due to the security issue and request from the supplier.



**Figure 7.** Five classes depending on the product surface.

The manufacturing process consists of magnetic core insertion, injection molding, and core cutting. Surface defects such as flash, sink marks, flow lines, and black spots and shape defects such as short shots and uncut cores occur in this process. The target defects presented in Figure 8 are the black spots, short shots, and uncut defects that occur most frequently. Several types of defects may occur in one product. Another problem is the difficulty in arranging products to acquire product images for automated inspection. Two products are produced in one mold, and the cycle time is within 3 s. A worker must immediately insert the magnetic core inside the mold after the product is ejected. It is challenging to acquire images of equal scale and quality because multiple products are simultaneously ejected from a mold. We assume that, when the product is ejected, the alignment of the product is not constant. Therefore, it is necessary to focus on detecting defects that may occur in all parts and sides of the product.

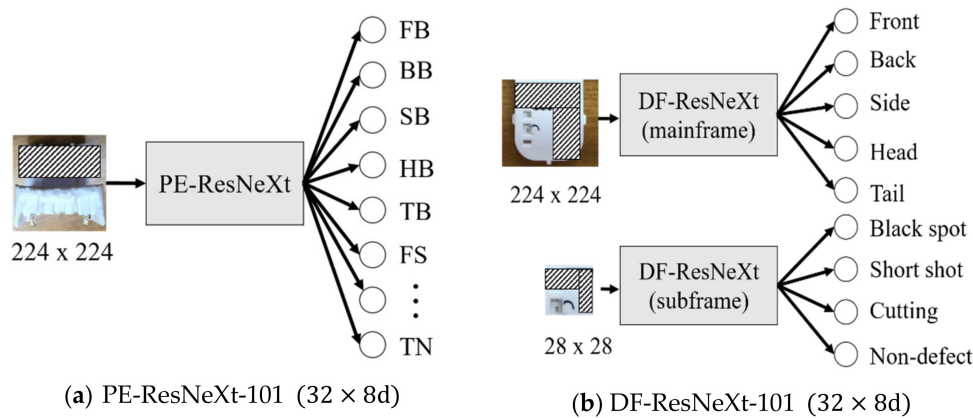


**Figure 8.** Target defects.

Five classes are defined for the target product according to the appearance of the product: the front, back, side, head, and tail (Figure 7). The three most common defects are selected (black spots, short shots, and uncut defects) in the target product, as shown in Figure 8. Images included 656 black spots, 224 short shots, 64 uncut defects, and 560 nondefect images. We assigned 1304 images for training and 200 images for validation with 50 defect images and 50 nondefect images. The number of training images was increased to 20,864 via image cropping and augmentation. The models applied for the experiment were PE-ResNeXt-101 ( $32 \times 8d$ ) and DF-ResNeXt-101 ( $32 \times 8d$ ) models, each with 20 and 9 output classes, respectively. Table 6 lists 20 output classes of PE-ResNeXt. FB in Table 6, for example, indicates a black spot on the front of the part, BB indicates a black spot on the back of the part, and so on. DK-ResNeXt has nine output classes, as shown in Figure 9b, including five outputs of the mainframe (e.g., front, back, side, head, and tail) and four outputs of the subframe (e.g., black spot, short shot, cutting, and nondefect). Figure 9 shows schematic network models of PE-ResNeXt and DF-ResNeXt.

**Table 6.** Output classes of PE-ResNeXt.

Criteria	Front	Back	Side	Head	Tail
Black spot	FB	BB	SB	HB	TB
Short shot	FS	BS	SS	HS	TS
Cutting	FC	BC	SC	HC	TC
Nondefect	FN	BN	SN	HN	TN



**Figure 9.** DK-ResNeXt used for the case study.

The experiment was performed 10 times, and the average TPR, TNR, and accuracy are presented in Table 7. We compared the proposed models (PE-ResNeXt-101(32 × 8d) and DF-ResNeXt-101 (32 × 8d)) with ResNeXt-101 (32 × 8d) and Deformable ConvNet V2. DF-ResNeXt-101 (32 × 8d) exhibited the highest classification accuracy of 98.5%, while PE-ResNeXt-101 (32 × 8d) had an accuracy of 97.8%. All four models tend to have a slightly higher TNR than TPR, which means that the sensitivity of the model is slightly lower than the specificity.

**Table 7.** Experimental results for the case study.

Criteria (%)	DF-ResNeXt-101 (32 × 8d)	PE-ResNeXt-101 (32 × 8d)	ResNeXt-101 (32 × 8d)	Deformable ConvNet V2
TPR	97.4	96.6	92.3	85.3
TNR	100	99.0	95.0	93.6
Accuracy	98.5	97.8	94.6	91.2

Figure 10 shows the average accuracy and standard deviation of each model. The model with the highest variability was PE-ResNeXt-101 (32 × 8d). The standard deviation was 0.7898, which was significantly higher than other models. The model with the smallest variability was DF-ResNet-101 (32 × 8d). The standard deviation was the lowest at 0.3444, but there was no significant difference with ResNeXt-101 (32 × 8d) and Deformable ConvNet V2. The standard deviations of ResNeXt-101 (32 × 8d) and Deformable ConvNet V2 are 0.4115 and 0.5332, respectively.

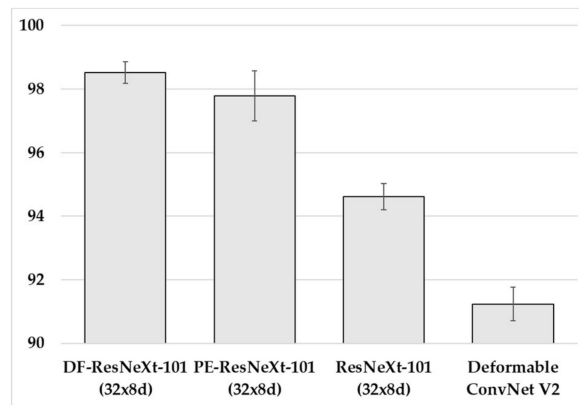


Figure 10. Mean accuracy with standard deviation error bar.

The most frequent error in this case study was to classify black spots as short shots. The black spots used in the experiment can be defined as a normal spot, smeared spots, and crack spot, as shown in Figure 11. Normal spots were classified with excellent accuracy in most experiment cases, but smeared spots and crack spots were occasionally classified as short shots. In particular, a case in which crack spots were classified as short shots occurred in the DK-ResNeXt-101 (32 × 8d) experiment. We assumed that cropping and feeding of the DK-ResNeXt-101 (32 × 8d) could be a drawback that causes misclassification of the black spot and short shot (see Figure 12). Further research should identify and analyze this drawback that may arise from the interactions of the multimodal.

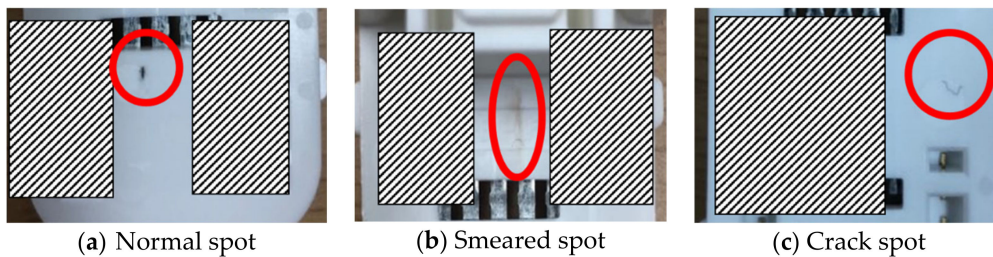


Figure 11. Three types of black spots.

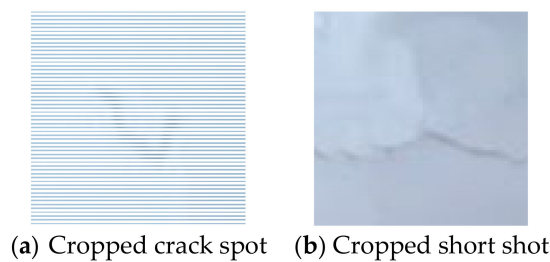


Figure 12. Examples of cropped images.

Even though one of our models proposed in this study (i.e., DF-ResNeXt-101 (32 × 8d)) gave the highest classification accuracy compared to other models, it is not enough to apply the model to the real production process immediately. The DAGM dataset has clearly independent background textures and types of defects for each class. Defects in real injection-molded products, however, may occur on all sides of the product. PE-ResNeXt occasionally does not classify accurate locations (front, back, side, head, or tail) of defects and only identifies the type of defects. Therefore, the classification performance of the model using real product data is slightly lower than that using DAGM data. The methodology proposed in this study, however, has the potential to be improved via a detailed analysis of defect types and locations or strategies of data acquisition. For example, the DF-ResNeXt-101 (32 × 8d) model is very robust for the product location classification, but an error occurred in distinguishing

crack spots and short shots. The proposed methodology consists of two models that train only image data. In order to deal with actual defects, however, a novel network model can be proposed, which train multimodal data (e.g., mixing data of images, discrete and continuous data, etc.). The proposed model has the flexibility to integrate various characteristics of networks such as autoencoder, recurrent learning network, resign proposal network, and generative adversarial network. We assume that the product weight is a significant parameter for understanding process stability and product defects. Besides, we expect to extract not only product defects but also process anomalies through a different multimodal approach by simultaneously acquiring defect images, product weight, and machining noise generated during the injection molding process.

## 5. Conclusions

In this study, two classification models are proposed using fixed and deformable kernels simultaneously. A product that simultaneously has defects on its surface and a defective shape can be detected effectively by referring to appearance defects, such as foreign matter, color variations on the surface of perfect products, defects in geometrical shape, and defects in which both characteristics occur. The practical designs of the deformable kernel—which has been proven to extract geometrical features effectively—and the fixed kernel are expected to effectively extract both surface and shape defects. The proposed model is robust to rotational or positional changes of the product. As demonstrated in the scaled MINIST dataset [10], a model using a serially designed fixed kernel and deformable kernels is robust to the positional change and distortion of the target characteristics. PE-ResNeXt, the model proposed in this study, applies ResNeXt as a backbone, which is known as a simple structure, and includes three deformable kernel layers at the end of the network to combine classification results. The majority voting method is applied to improve classification accuracy. DF-ResNeXt uses a multimodal network method to classify classes and defects using subframes composed of deformable kernel layers with cropped input images. The proposed model has a deformable kernel layer modified from the fixed kernel layer to compensate for the weakness of the existing CNN, and it combines the classification results. The image cropping and feeding method achieve high detection accuracy with a small network and low complexity. In addition, the proposed model exhibits better performance than the deep CNN model by changing the detailed design of the network. The proposed model is demonstrated and proven to be comparable to the models from existing studies using the DAGM dataset. In the test with DAGM data, the classification accuracies of the PE-ResNeXt and DF-ResNeXt models were 99.9% and 100%, respectively. The proposed models with the same output numbers are superior to FR-CNN with output class 6/2 and DCNN with output class 12. The case study demonstrates that the proposed model is valid for products with defects on the surface or in shape. This research conducts comparative research with previous excellent studies. This study contributes to applying and analyzing models verified with existing open-source datasets to actual products. In addition, this paper suggests one of the solutions to the problem that occurs in real cases. The proposed model also has the advantage of easy modification by adapting pretrained networks. However, the classification accuracy needs to be improved for the model to be applied in actual industrial processes. As opposed to actual product data which have variations, there is a lack of variation in the experimental data, and model verification and accuracy improvement with more experiments need to be performed in future studies. Furthermore, we will conduct further research on process anomaly detection by integrating the proposed model, noise recognition of a machining process using LSTM, and operator and part recognition based on R-CNN.

**Author Contributions:** Conceptualization, H.L. and K.R.; methodology, H.L.; software, H.L.; validation, H.L. and K.R.; formal analysis, H.L.; investigation, K.R.; resources, H.L. and K.R.; data curation, H.S. and K.R.; writing—Original draft preparation, H.L.; writing—Review and editing, K.R.; visualization, H.L.; supervision, K.R.; project administration, K.R.; funding acquisition, K.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.



**Acknowledgments:** This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2016R1A2B4014898), and this work was supported by the IoT-big data-based value chain innovation support project for mold production of the Korea Institute for Advancement of Technology (KIAT) granted financial resource from the Ministry of Trade, Industry and Energy, Republic of Korea (No. P0001955).

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

## References

1. Latorella, K.A.; Prabhu, P.V. A review of human error in aviation maintenance and inspection. *Int. J. Ind. Ergon.* **2000**, *26*, 133–161. [\[CrossRef\]](#)
2. Bruning, J.H.; Feldman, M.; Kinsel, T.S.; Sittig, E.K.; Townsend, R.L. An automated mask inspection system—AMIS. *IEEE Trans. Electron Devices* **1975**, *22*, 487–495. [\[CrossRef\]](#)
3. Cohen, F.S.; Fan, Z.; Attali, S. Automated inspection of textile fabrics using textural models. *IEEE PAMI* **1991**, *8*, 803–808. [\[CrossRef\]](#)
4. Malloy, R.A. *Plastic Part Design for Injection Molding*, 2nd ed.; Hanser Publishers: New York, NY, USA, 1994; pp. 285–294.
5. Qiu, Z.; Chen, J.; Zhao, Y.; Zhu, S.; He, Y.; Zhang, C. Variety identification of single rice seed using hyperspectral imaging combined with convolutional neural network. *Appl. Sci.* **2018**, *8*, 212. [\[CrossRef\]](#)
6. Polat, H.; Danaei, M.H. Classification of pulmonary CT images by using hybrid 3D-deep convolutional neural network architecture. *Appl. Sci.* **2019**, *9*, 940. [\[CrossRef\]](#)
7. Oh, S.L.; Vicnesh, J.; Ciaccio, E.J.; Yuvaraj, R.; Acharya, U.R. Deep convolutional neural network model for automated diagnosis of schizophrenia using EEG signals. *Appl. Sci.* **2019**, *9*, 2870. [\[CrossRef\]](#)
8. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
9. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Venice, Italy, 22 October 2017; pp. 764–773.
10. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9308–9316.
11. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [\[CrossRef\]](#)
12. Krizhevsky, A.; Sutskever, I.; Hinton, G. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards realtime object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
14. Fan, J.; Xu, W.; Wu, Y.; Gong, Y. Human tracking using convolutional neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2010**, *21*, 1610–1623.
15. Wang, H.; Li, S.; Song, L.; Cui, L. A novel convolutional neural network based fault recognition method via image fusion of multi-vibration-signals. *Comput. Ind.* **2019**, *105*, 182–190. [\[CrossRef\]](#)
16. Wu, C.; Jiang, P.; Ding, C.; Feng, F.; Chen, T. Intelligent fault diagnosis of rotating machinery based on one-dimensional convolutional neural network. *Comput. Ind.* **2019**, *108*, 53–61. [\[CrossRef\]](#)
17. Xie, X. A review of recent advances in surface defect detection using texture analysis techniques. *ELCVIA* **2008**, *7*, 1–22. [\[CrossRef\]](#)
18. Gao, Z.; Cecati, C.; Ding, S.X. A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches. *IEEE Trans. Ind. Electron* **2015**, *62*, 3757–3767. [\[CrossRef\]](#)
19. Boukouvalas, C.; Kittler, J.; Marik, R.; Petrou, M. Color grading of randomly textured ceramic tiles using color histograms. *IEEE Trans. Ind. Electron* **1999**, *46*, 219–226. [\[CrossRef\]](#)

20. Pietikainen, M.; Maenpaa, T.; Viertola, J. Color texture classification with color histograms and local binary patterns. In *Workshop on Texture Analysis in Machine Vision*; Machine Vision Group, University of Oulu: Oulu, Finland, 2002; pp. 109–112.
21. Escofet, J.; Navarro, R.; Pladellourens, M.M.J. Detection of local defects in textile webs using Gabor filters. *Opt. Eng.* **1998**, *37*, 2297–2307.
22. Ren, R.; Hung, T.; Tan, K.C. A generic deep-learning-based approach for automated surface inspection. *IEEE Trans. Cybern.* **2017**, *48*, 929–940. [[CrossRef](#)] [[PubMed](#)]
23. Staar, B.; Lütjen, M.; Freitag, M. Anomaly detection with convolutional neural networks for industrial surface inspection. *Proc. CIRP* **2019**, *79*, 484–489. [[CrossRef](#)]
24. Wang, T.; Chen, Y.; Qiao, M.; Snoussi, H. A fast and robust convolutional neural network-based defect detection model in product quality control. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 3465–3471. [[CrossRef](#)]
25. Tao, X.; Zhang, D.; Ma, W.; Liu, X.; Xu, D. Automatic metallic surface defect detection and recognition with convolutional neural networks. *Appl. Sci.* **2018**, *8*, 1575. [[CrossRef](#)]
26. Chen, J.; Liu, Z.; Wang, H.; Núñez, A.; Han, Z. Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network. *IEEE Trans. Instrum. Meas.* **2017**, *67*, 257–269. [[CrossRef](#)]
27. Zhou, S.; Chen, Y.; Zhang, D.; Xie, J.; Zhou, Y. Classification of surface defects on steel sheet using convolutional neural networks. *Mater. Technol.* **2017**, *51*, 123–131.
28. Deng, J.; Lu, Y.; Lee, V.C.S. Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network. *Comput. Aided Civ. Infrastruct. Eng.* **2020**, *35*, 373–388. [[CrossRef](#)]
29. Chun, P.J.; Izumi, S.; Yamane, T. Automatic detection method of cracks from concrete surface imagery using two-step light gradient boosting machine. *Comput. Aided Civ. Infrastruct. Eng.* **2020**. [[CrossRef](#)]
30. Yamane, T.; Chun, P.J. Crack Detection from a Concrete Surface Image Based on Semantic Segmentation Using Deep Learning. *J. Adv. Concr. Technol.* **2020**, *18*, 493–504. [[CrossRef](#)]
31. Dung, C.V. Autonomous concrete crack detection using deep fully convolutional neural network. *Automat. Constr.* **2019**, *99*, 52–58. [[CrossRef](#)]
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
33. Timm, F.; Barth, E. Non-parametric texture defect detection using Weibull features. In Proceedings of the Image Processing: Machine Vision Applications IV, San Francisco, CA, USA, 7 February 2011; p. 78770.
34. Siebel, N.T.; Sommer, G. Learning defect classifiers for visual inspection images by neuro-evolution using weakly labelled training data. In Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, China, 1–6 June 2008; pp. 3925–3931.
35. Jiang, X.; Scott, P.; Whitehouse, D. Wavelets and their applications for surface metrology. *CIRP Ann. Manuf. Technol.* **2008**, *57*, 555–558. [[CrossRef](#)]
36. Weimer, D.; Scholz-Reiter, B.; Shpitalni, M. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Ann. Manuf. Technol.* **2016**, *65*, 417–420. [[CrossRef](#)]
37. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Zhang, Z. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint* **2019**, arXiv:1906.07155.
38. PyTorch Implementation of Deformable ConvNets v2 (Modulated Deformable Convolution). Available online: <https://Github.com/4uiiurz1/pytorch-deform-conv-v2> (accessed on 19 April 2019).

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).