

## Article

# Automated Identification of Wood Veneer Surface Defects Using Faster Region-Based Convolutional Neural Network with Data Augmentation and Transfer Learning

Augustas Urbonas <sup>1</sup>, Vidas Raudonis <sup>2</sup>, Rytis Maskeliūnas <sup>3</sup>  and Robertas Damaševičius <sup>3,\*</sup> 

<sup>1</sup> UAB “Vakaru Medienos Grupe”, LT-92100 Klaipėda, Lithuania; augustas.urbonas@gmail.com

<sup>2</sup> UAB “Power of Eye”, LT-51392 Kaunas, Lithuania; vidas.raudonis@gmail.com

<sup>3</sup> Institute of Mathematics, Silesian University of Technology, 44-100 Gliwice, Poland; rytis.maskeliunas@polsl.pl

\* Correspondence: robertas.damasevicius@polsl.pl

Received: 9 October 2019; Accepted: 12 November 2019; Published: 15 November 2019



**Abstract:** In the lumber and wood processing industry, most visual quality inspections are still done by trained human operators. Visual inspection is a tedious and repetitive task that involves a high likelihood of human error. Currently, new automated solutions with high-resolution cameras and visual inspection algorithms are being tested, but they are not always fast and accurate enough for real-time industrial applications. This paper proposes an automatic visual inspection system for the location and classification of defects on the wood surface. We adopted a faster region-based convolutional neural network (faster R-CNN) for the identification of defects on wood veneer surfaces. Faster R-CNN has been successfully used in medical image processing and object tracking before, but it has not yet been applied for wood panel surface quality assurance. To improve the results, we used pre-trained AlexNet, VGG16, BNInception, and ResNet152 neural network models for transfer learning. The results of the experiments using a synthetically augmented dataset are presented. The best average accuracy of 80.6% was obtained using the pretrained ResNet152 neural network model. By combining all the defect classes, a 96.1% accuracy of finding wood panel surface defects was achieved.

**Keywords:** defect detection; quality control; wood veneer; data augmentation; transfer learning; faster R-CNN; deep learning

## 1. Introduction

According to the United Nations, there is a growth in the wood industry worldwide. In 2017, the world production of wood panels reached 402 million m<sup>3</sup> per year [1]. In the Asian market alone, timber production grew by 40% over the period 2011–2015. One of the largest wood processing areas is the production of wood veneer, which, together with the production of plywood panels, has become the dominant market, accounting for 39% of the total wood processing market. Wood veneer is used for coating and decorating surfaces of furniture, doors, or interior design elements. Due to the heterogeneity of the raw material and the complexity of the manufacturing process, the panels produced may have various defects, such as scratches, stains, or wood cores.

One of the most popular wood panel quality control methods is random testing in the production process, random selection of products, and quality checking. If the quality of the checked products is maintained, the production continues, but if defects or discrepancies are found, the production line is stopped, a solution is sought, and the manufacturer suffers from downtime and productivity losses.

In the wood processing industry, visual quality analysis is often carried out by trained experts, but it is tedious and repetitive work that retains the human error factor. Although this method is widely used, it does not ensure that the entire production volume is checked. With large production volumes, manufacturers are simply not able to check all the products, and there is always the possibility of human error factors such as distraction or fatigue. Humans rarely have the ability to achieve more than 70% reliability in visual analysis [2], so wood panel manufacturing companies are looking for ways to automate this process, thus increasing reliability and quality production.

Modern automated inspection systems are able to operate without interruption, at high speed, and following strictly defined rules using a variety of sensors, so that they can check all the products they produce and ensure reliable results [3]. Computer vision systems using high-resolution and high-speed video cameras scan products and report on product quality through various algorithms and techniques. Several methods can be used for surface analysis, such as Gabor [4] or wavelet [5] transforms, and deep learning techniques [6–10], but they are not always sufficiently accurate or fast. For example, the double threshold method is used to check the quality of materials in the textile industry [11] to reduce the amount of data in the images used for the training of neural networks. Although this method is one of the fastest, it is only suitable for smooth lighting systems, and cannot be used with deep texture objects such as smooth ceramic tiles [12] or fine-textured textiles.

Wavelet transform is a multiscale analysis and decomposition method. First, the image is sharpened using high-pass filters, thereby isolating local brightness changes; then the image is filtered using low-pass filters and reduced twice to obtain a compressed approximate image. The operation is repeated to get the second level of wavelet transform. This method was applied for the textile defect detection system [13]. The algorithm was tested in the industry and achieved 98% accuracy of recognition of textile defects. The method was also used to classify wood branches and cores in the wood industry [14].

Gabor filter is a linear filter used to describe an image pattern or a pattern with a given frequency spectrum around the point of inspection. This method was applied for the analysis of the quality of wood veneer [4], when using a multilayer perceptron with Gabor attributes and RGB histograms. The system was designed for the classification of tree branches and surface roughness. The trained multilayer perceptron achieved 85–90% accuracy. This method is also widely used in the textile industry [15]. Chacon and Alonso [16] used features obtained from Gabor filters and a fuzzy self-organizing neural network as a classifier to identify four different types of knots found on wood surfaces, achieving 91.17% accuracy.

Traditionally, classical machine learning methods such as support vector machines (SVM) and K-nearest neighbor (KNN) have been used for the recognition of wood surface defects. Gu et al. [17] used the intensity and size characteristics of these defects and applied SVM. Mahram et al. [18] identified knots using KNN, SVM, and texture descriptors. YongHua and Jin-Cong [19] used a gray-level co-occurrence matrix (GLCM) and Tamura texture parameters for the classification of knots on wood surfaces. Hittawe et al. [20] proposed using local binary patterns (LBP) and speeded-up robust features (SURF) for feature extraction, and SVM for the detection of cracks and knots in wood surface images. Zhao and Wang [21] suggested using principal component analysis (PCA) or multidimensional scaling (MDS) with Mahalanobis distance for identifying wood vessels in wood hyperspectral microscopic images.

Convolutional neural networks (CNN) are well suited for solving image classification problems, but they need a high number of samples for training. These networks are made up of several filters, whose outputs represent convolution and pooling operations. When images are analyzed using a sliding window, CNNs are very slow, because the object search time is dependent on the size of the image [22]. One of the solutions to this problem is to use regional matching methods, assuming that the searched visual objects have common visual features and stand out from the background of an image. Once a region with a high probability of finding an object is found, it is checked by using a

more accurate classification method, thus determining its exact position and class. Using this method, the number of search regions is reduced considerably [22].

One of the major detectors of this type is region-based CNN (R-CNN) [23]. Faster R-CNN is an improved version of R-CNN with a faster training time [24]. In faster R-CNN, the image is processed using a series of sequentially connected convolution and maximum pooling layers, thereby producing a single map of the convolved features. This feature map selects the regions of interest (RoI) that are being processed in the pooling layers of the RoI. In this layer, each feature region is transformed into a vector of fixed size attributes, which are processed in fully connected network layers. After processing the region with a fully convoluted layer, the layer is branched into two separate output layers: the classification layer with softmax function is determined by the object class, and the positioning layer defines the positions of the rectangular points limiting the objects. Faster R-CNN has been used with success in medical image processing [25] and object tracking [26], but is still little used for the qualitative analysis of wood panel surfaces.

Other types of neural network architectures have also been used for surface defect recognition. Tao et al. [27] used a cascaded autoencoder (CASAE) to transform the surface image into a pixel-wise prediction mask. Then the surface regions are categorized into their respective classes using a compact CNN, achieving an accuracy of 89.60% for the metallic surface defect detection task. Yuce et al. [28] used PCA for the determination of the critical features before using ANN to identify wood veneer defects. Ren et al. [29] used a pretrained decaf CNN [30] to generate a surface defect heatmap. Then the heatmap was binarized and segmented by adopting the graph-based Felzenszwalb's method. In [31], the VGG-16 network architecture achieved over 92% accuracy while allowing localization of the timber surface defects. Some recently proposed neural network architectures such as densely connected convolutional network [32], attentional convolutional network [33], and adversarial auto-encoder (with convolutional structure) [34] have proven very successful in various image recognition tasks. For example, the YOLO model [35] demonstrated very good speed for real-time object detection. However, the methods have not yet been tested on wood surface defects.

For image pre-processing, fuzzy binarization [36–38] and fuzzy partitioning [39] have been proposed to extract candidate regions. Also, foreground segmentation could be used to improve the model results. For example, the segmentation map of the image can be fed into the network along with the raw image [40–42].

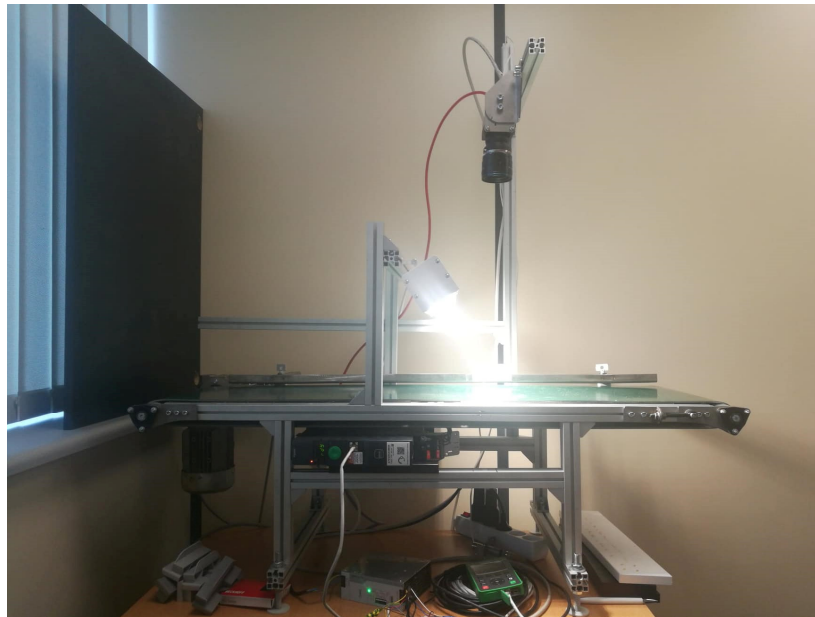
In this paper, we present an automatic visual inspection system for the location and classification of defects on the surface of wood veneer. Our main contribution is speed optimization of the defect identification task as the defect recognition system has to run on an actual conveyor belt and is programmed on a wood veneer sorting conveyor line; therefore, our task is to find the optimum solution for the existing equipment. Here we have adopted the faster R-CNN method to evaluate the quality of wood veneer. We address the problem as a small data problem [43] and apply data augmentation [44,45] and transfer learning [46] to improve the classification results. We present our methodology as well as the results of the experiments.

## 2. Materials and Methods

### 2.1. Hardware and Software

A special conveyor belt was produced for data collection. This conveyor is depicted in Figure 1. The conveyor was capable of running up to 1 m/s speed. The length of the conveyor belt was 1 m, with a width of 30 cm. The distance between the camera lens and the subject was 30 cm. The luminaire was mounted at a height of 10 cm and at a 20° angle to the object. A line scan camera was used to acquire the image. We used a monochromatic Basler raL4096-24gm camera (Basler AG, Ahrensburg, Germany) with an Awaiba DR-4k-7 sensor (Awaiba Holding SA, Yverdon-les-Bains, Switzerland). This camera has a recording speed of 26 kHz and a horizontal resolution of 4096 pixels. The resolutions and speeds of this camera are sufficient to capture a wood veneer moving at a speed of 4 m/s while maintaining a

resolution of  $0.25 \text{ mm}^2/\text{px}$ . Camera shooting is synchronized with the moving conveyor belt using the pulse encoder LIKA CK59-Y-500ZCZ214R (Lika Electronic Srl, Carre', Italy).



**Figure 1.** Conveyor belt developed for wood surface quality evaluation.

Python and C# programming language (Microsoft Corporation, WA, USA) were used to implement this project. Microsoft CNTK Pack 2.6 (Microsoft Corporation, WA, USA) was used to implement neural networks. All calculations were done using a computer with Intel I7-7700 3.6 Ghz 4-core processor, 16 GB GDDR5 operating memory, and two GTX 1060 6GB video cards. A solid-state drive (SSD) disk was used to store the data on the computer. Wood veneer images have been classified and labeled using a VoTT (Visual object Tagging Tool) (Microsoft Corporation, WA, USA) program for tagging and sorting images for using with deep training networks.

## 2.2. Materials

Wood veneer is a thin (0.5–3 mm), usually hardwood sheet, intended for coating various surfaces of furniture, doors, or interior elements. The wood veneer is illustrated in Figure 2. The visual quality of the wood veneer is also very important, and depends on where it will be used. For example, in the manufacture of furniture for surface coating, the visual quality of the veneer is very important, with as few branches, stains, or cracks as possible. In the production of veneer, standard-quality veneer is used in the surface layers, and the veneer with the worst visual quality is placed in the inner layers.



**Figure 2.** Example of wood veneer under study.

When defining the types of defects and their positions, the RoIs in the images are classified into five defect types according to the parameters given in Table 1.

**Table 1.** Explanation of defect types.

| Sort | Branch Size, px <sup>2</sup> | No. of Branches | Allowed Area of Cores, px <sup>2</sup> | Allowed Area of Scratches, px <sup>2</sup> | Allowed Number of Scratches | Allowed Area of Blemish, px <sup>2</sup> |
|------|------------------------------|-----------------|----------------------------------------|--------------------------------------------|-----------------------------|------------------------------------------|
| A/B  | 6300                         | 6               | 0                                      | 2100                                       | 3                           | 0                                        |
| E    | 4200                         | 6               | 0                                      | 12,000                                     | 3                           | 0                                        |
| C    | 16,000                       | unlimited       | 10% of area                            | 12,000                                     | 8                           | 5% of area                               |
| D    | 90,000                       | unlimited       | unlimited                              | 12,000                                     | 8                           | 20% of area                              |
| G    | unlimited                    | unlimited       | unlimited                              | unlimited                                  | unlimited                   | unlimited                                |

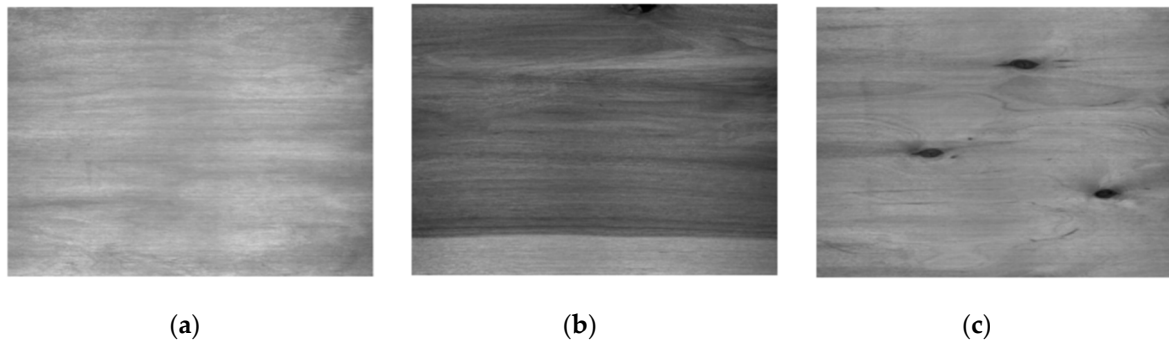
### 2.3. Dataset

The 250 veneers of size 1525 × 1525 mm were scanned in 300 × 300 mm batches for training and testing. Each veneer was scanned at a resolution of 4000 × 3000 pixels in monochrome single-channel images using the equipment described in Section 2.1. The overall number of usable images was 4729. Out of this number, we obtained 353 veneer images (300 × 300 mm) with defects of wood, which contained 982 branch, 288 core, 398 split, and 253 stain defects. In total, 285 images had at least one defect and six images had no defects. The remainder (defect-free images) were categorized as background. For training, we used 291 images, and for testing, we used 62 images representing the most defective single sheets of veneer. The examples of the resulting veneers are depicted in Figure 3. To improve neural network training, the original dataset was augmented. Each image from the dataset contained from eight to 12 defective regions of interest (ROI); therefore, the actual number of examples used for training and testing is 10 times larger. In total, there were 353 × 10 = 3530 of defective ROIs available.

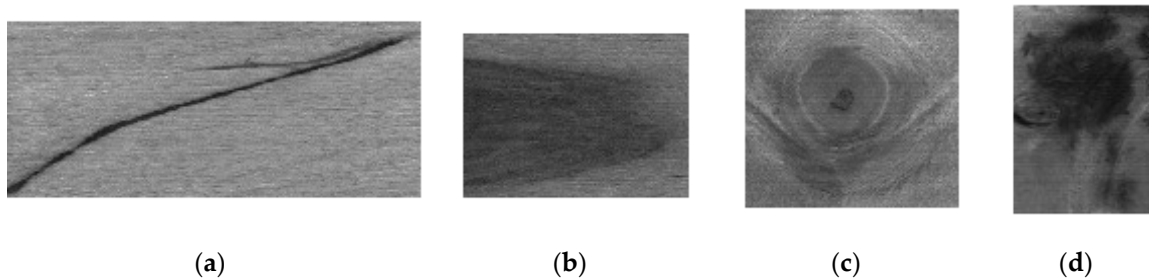
Dataset images have been reduced to 800 × 600 pixel resolution to speed up the training of a neural network. This size was selected heuristically to achieve the best quality and speed. Reducing the size of an image reduces the visibility of small objects such as splits or scratches.

In this work, four major wood veneer defects were identified: branches are places where tree branches are visible; scratches/splits are places where cracks, splits, or scratches are visible; a core is

where the core or bark of a tree is visible; and stains are water or pigmentation-induced wood spots. Examples of these defects are shown in Figure 4.

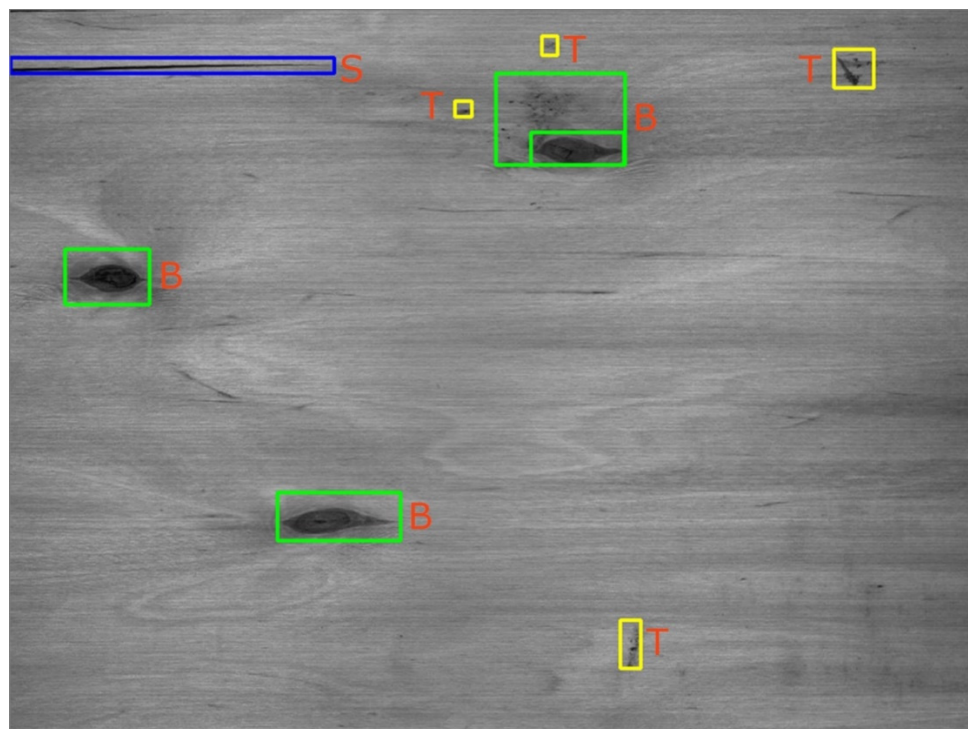


**Figure 3.** Samples of veneers: (a) nondefective; (b) with visible core; (c) with visible branches.



**Figure 4.** Examples of wood veneer defects: (a) split; (b) core; (c) branch; (d) stain.

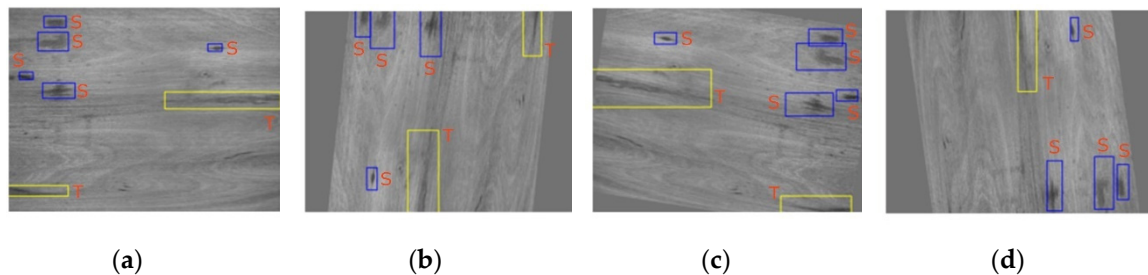
The defects were selected and labeled using the VoTT program to determine the size, position, and class to which it is assigned. The VoTT program marked image is shown in Figure 5.



**Figure 5.** An image of wood veneer labeled using VoTT. Green is used to label the branch (B) defects. Blue is used to label the split (S) defects. Yellow is used to label the stain (T) defects.

## 2.4. Data Augmentation

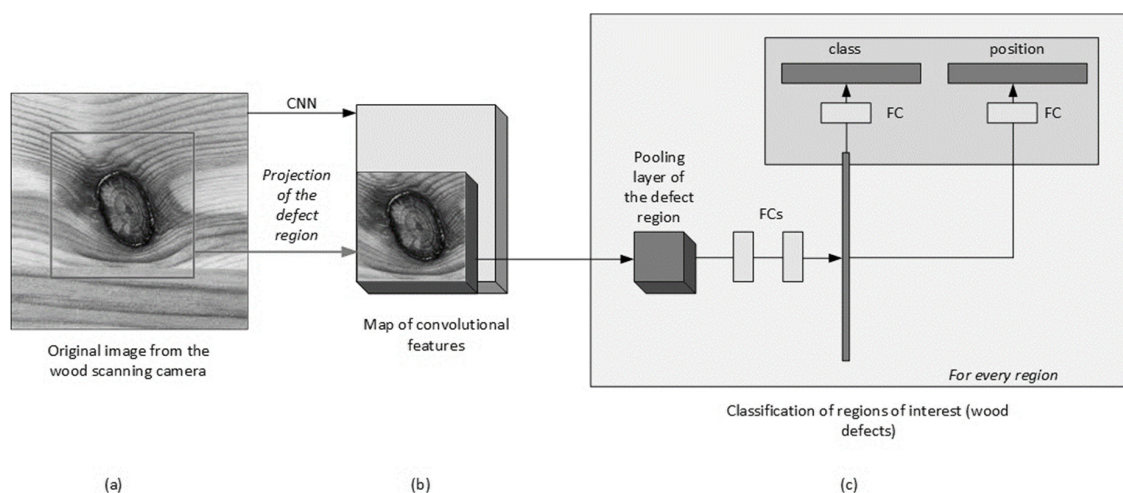
Augmentation of a dataset is often used to increase the accuracy and adaptability of neural networks for classification. In this work, three different geometric transformations were applied: flip, rotation transformation, and resize transformation. For each of the 291 training pictures, four random synthetic pictures were created. Examples of the synthesized images are shown in Figure 6.



**Figure 6.** Example of data augmentation for increasing the number of images for training: (a) original images; (b–d) images obtained by rotation transformation. Blue is used to label the split (S) defects. Yellow is used to label the stain (T) defects.

## 2.5. Architecture of Neural Network

In the faster R-CNN method, the image is processed using a series of sequentially connected convolution and maximum pooling layers, thereby producing a single map of the convolutional features. This feature map selects the regions of interest that are being processed in the maximum pooling layers. In this layer, each feature region is transformed into a vector of fixed size features that are processed in the fully connected network layers. After processing the region with a fully connected layer, the layer is branched into two separate output layers: the classification layer with the softmax function determines the object class, and the positioning layer is for defining the positions of the four rectangular points limiting the objects. The simplified operation of the method is depicted in Figure 7. This method is more accurate than R-CNN. During the training, all network layers are trained and there is not much room for character storage during training [47].



**Figure 7.** Operation of faster R-CNN: (a) original image; (b) convolution features; (c) classification of regions of interest.

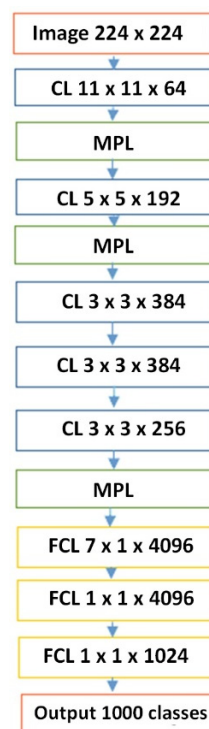
In the faster R-CNN method, the search method of options is replaced by the region proposal network (RPN). This method also introduced a new method of regional exclusion using anchor boxes. Window size multipliers are selected during training, and the aspect ratio varies between 1:1, 1:2, and 2:1, thus creating nine regions in each location. For example, the basic sliding window size is  $16 \times 16$  px

resolution, so there are three different aspect ratio windows:  $8 \times 24$  px,  $16 \times 16$  px, and  $24 \times 8$  px. These window sizes are multiplied by the three selected multipliers, thus obtaining nine different windows. The output of the RPN network is a feature map that indicates the position, width, and length of each of the regions in question and the probability of being an object or background. Applying the softmax selection function leaves the specified number of regions that are processed in the pool of regions of interest. In this layer, regions of different sizes are converted into vector vectors of fixed size. These vectors are ultimately used in the R-CNN network, which defines the object class and position, and the background images are rejected. In the R-CNN network, the fully connected layers are used, the first layer contains  $N + 1$  neurons, where  $N$  is the number of classes with an additional class background, and the second is the layer of  $4N$  neurons, which indicates the position and size of the object.

## 2.6. Transfer Learning Using Pre-Trained Neural Networks

Transfer learning is a neural network learning method that utilizes a similar problem solved by the network model, but then retrains it with the training data using only a specific part of the trained model [48]. There are currently two main methods to implement transfer learning. The first method retrains only the last layer, while the original model remains as a tool-defined feature. In the second method, other layers are also trained. This method is called fine-tuning. These are currently some of the most popular transfer models, such as: AlexNet, VGG, ResNet, and GoogleLeNet models.

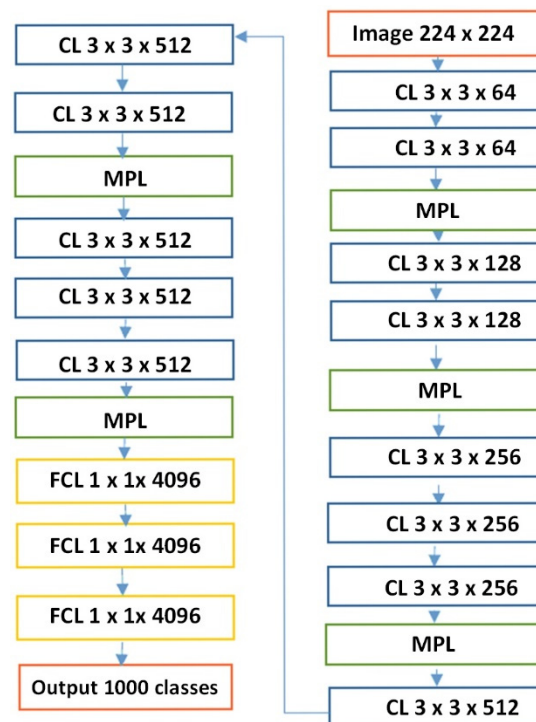
The AlexNet neural network consists of 11 layers, of which there are five convolution layers and three layers of maximum pooling [49]. The architecture of this network is depicted in Figure 8. The network input is a  $224 \times 224 \times 3$  pixel-sized vector that matches the size of the image, and the network output receives a 1000-value vector that indicates the class to which the object was assigned.



**Figure 8.** Architecture of the AlexNet neural network. CL is the convolution layer, MPL is the maximum pooling layer, and FCL is the fully connected layer.

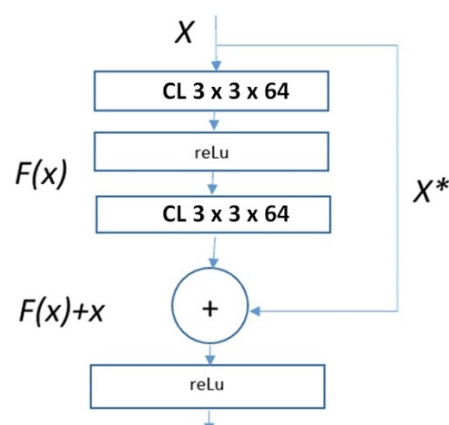
The VGG-16 architecture consists of 21 layers, of which 13 layers are convolution layers, five are maximum pooling layers, and three are fully connected layers [50]. A feature of this network is that all layers of the fold are  $3 \times 3$  pixels in size. The VGG16 architecture is illustrated in Figure 9. The input to

this architecture, like AlexNet, is a  $224 \times 224 \times 3$ -sized vector, and the output is a 1000-value vector that indicates the class to which the image belongs.



**Figure 9.** Architecture of the VGG16 neural network. CL is the convolution layer, and MPL is the maximum pooling layer.

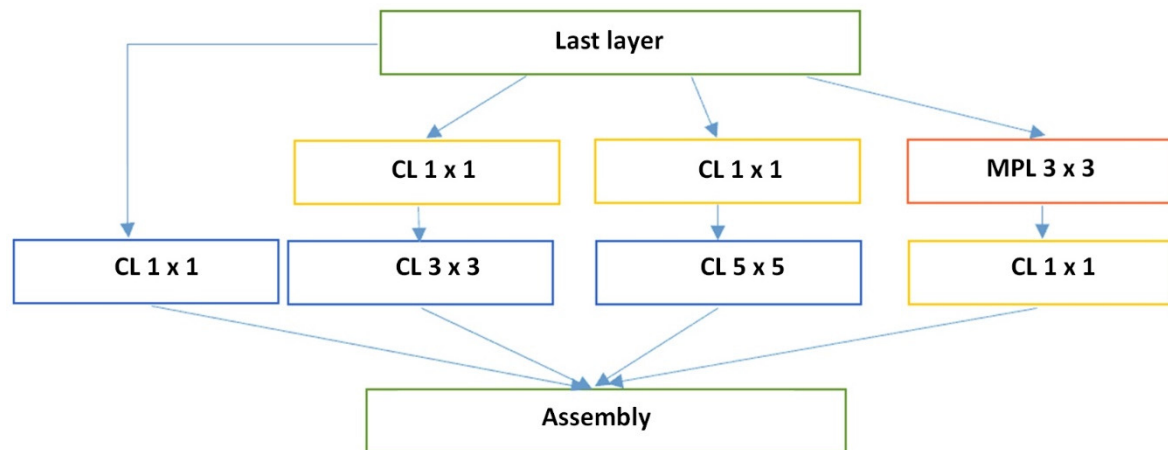
A residual network (ResNet) [51] addresses the vanishing gradient problem. This problem occurs when the network is too deep and the gradient value after the loss function is reduced, and later it becomes too small to influence the layer multipliers, so the network no longer learns. The ResNet architecture solves this problem by creating additional connections between different layers, where gradient information is transmitted directly by skipping the intermediate layers. Figure 10 shows the ResNet simplified layer link.



**Figure 10.** ResNet unit. CL is the convolution layer; reLu is a rectified linear unit.

The GoogleNet or inception network architecture has 22 neural layers [52]. The main features of this network were  $1 \times 1$  size convolution operations and the inception module. The  $1 \times 1$  convolution operations were first introduced on the NIN (network in network) [53]. NIN was used to distinguish attributes instead of linear filters before classification, which increases the accuracy. Also, NINs were

used instead of maximum pooling layers, and activation layers were created, the number of which is proportional to the number of classes. Applying the average operation and the soft peak layer results in class prediction. This application of NIN increased the speed and accuracy of the method. The second feature of the GoogleNet network is the inception module. In this module, the convolution filters of different size are applied in parallel to the input layer and are assembled to obtain a new vector of the output attributes. The diagram of the inception module is shown in Figure 11.



**Figure 11.** Inception module in GoogleNet. CL is the convolution layer and MPL is the maximum pooling layer. Dimensionality reduction layers are depicted in yellow, the maximum pooling layer is depicted in red, and the layers in the convolution operations are shown in blue.

In the connection layer, all results are combined into one feature map. The BNInception method uses batch normalization in the input layer, but also in the hidden layers, where each batch of data in the new iteration is adapted to the mean and distribution values learned [54].

Batch normalization makes it possible to train models with higher training speeds and reduced distribution, and it also reduces learning when the model adapts too much to the training data.

### 2.7. Evaluation

The accuracy of defect detection is determined by the intersection over union (IoU) metric, which defines the relationship between the ground truth defect and the found defect. This parameter is the Jaccard index. The object is usually validated as found when IoU is equal to or greater than 0.5. In this study, the accuracy was calculated using the ratio of true positive TP to the sum of false positives FP and true positives:

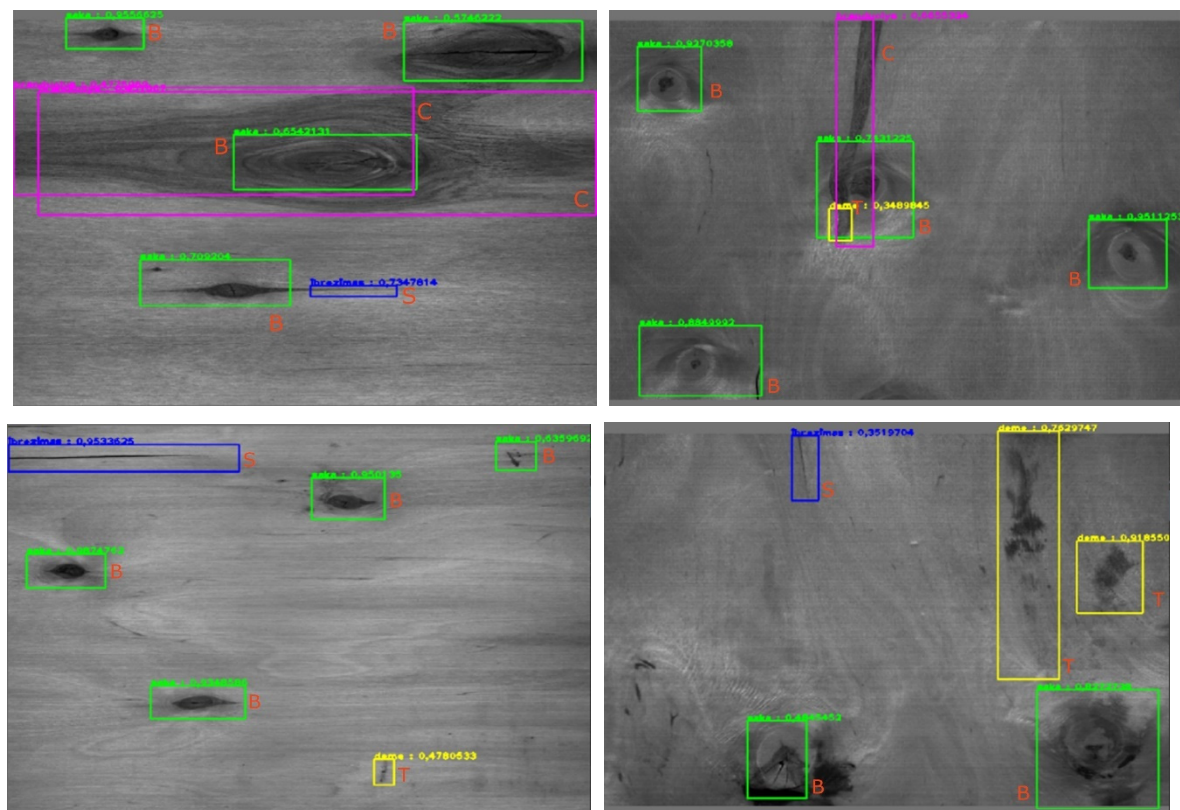
$$acc = \frac{TP}{FP + TP}, \quad (1)$$

where TP is found for defects with the value of IoU greater than the set threshold, and FP is for defects with the value of IoU less than the threshold value.

For each of the four defect classes and the background (i.e., no defect class), the average accuracy is calculated, and the overall performance is evaluated using grand average of accuracy, precision, recall, and f-score metric values.

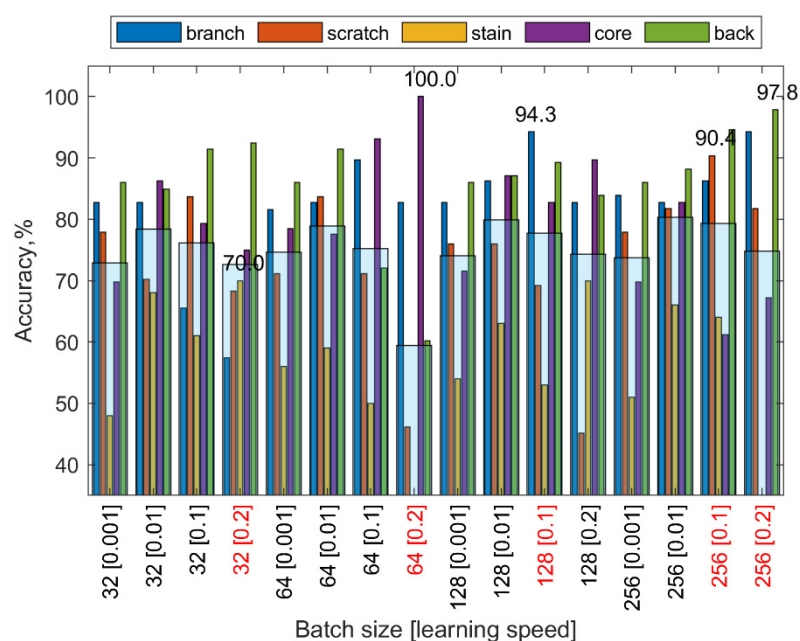
## 3. Results

Some of the typical defect classification results are given in Figure 12.

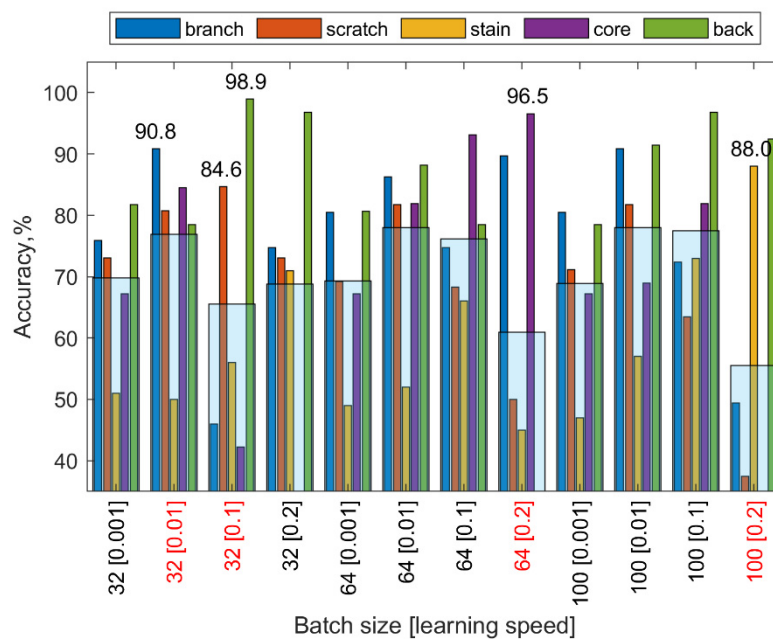


**Figure 12.** Examples of wood veneer defect classification using faster R-CNN. Branches (B) are labeled in green. Scratches (S) are labeled in blue. Cores (C) are labeled in violet. Stains (T) are labeled in yellow.

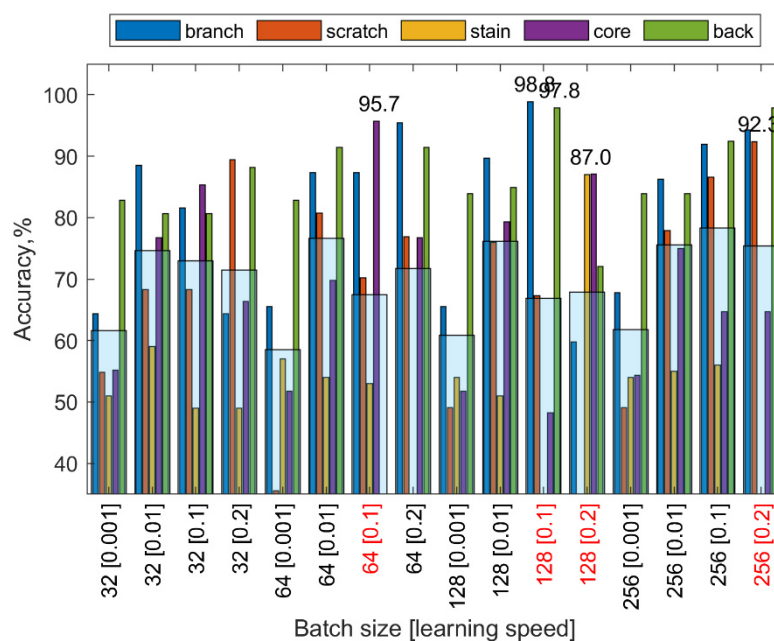
The classification accuracy using the pre-trained AlexNet, VGG16, BNInception, and ResNet152 neural network models trained with different network parameters (batch size and learning speed) is summarized in Figures 13–16.



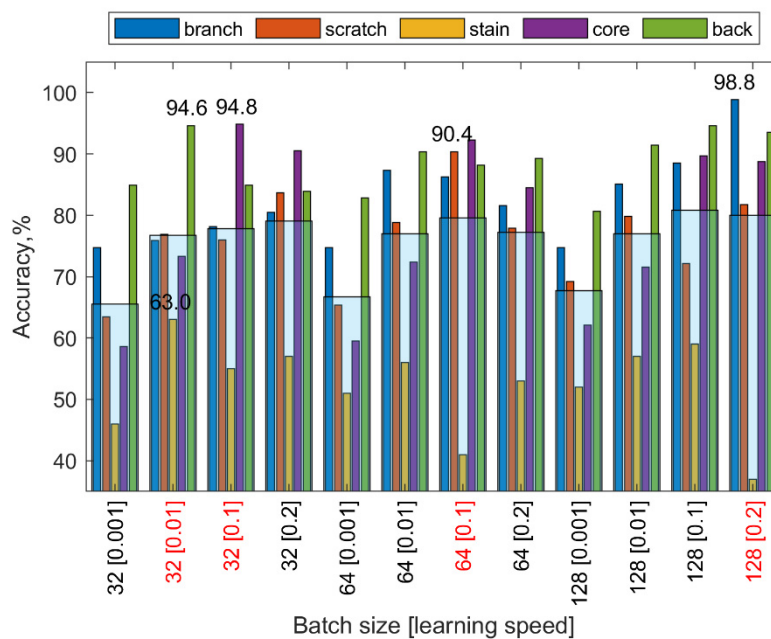
**Figure 13.** Classification accuracy using AlexNet neural network trained with different network parameters (batch size and learning speed). The best combinations of parameter values are indicated in red.



**Figure 14.** Classification accuracy using VGG16 neural network trained with different network parameters (batch size and learning speed). The best combinations of parameter values are indicated in red.



**Figure 15.** Classification accuracy using the BNInception neural network trained with different network parameters (batch size and learning speed). The best combinations of parameter values are indicated in red.



**Figure 16.** Classification accuracy using ResNet152 neural network trained with different network parameters (batch size and learning speed). The best combinations of parameter values are indicated in red.

The best results while using the AlexNet neural network were achieved using 32 batch size and 0.2 learning speed for stain class, 64 batch size and 0.2 learning speed for core class, 128 batch size and 0.1 learning speed for branch class, 256 batch size and 0.1 learning speed for scratch class, and 256 batch size and 0.1 learning speed for background class. The results with respect to different sliding window multipliers are presented in Table 2. The best accuracy, 76.6%, was achieved using the sliding window sizes of [4, 8, 12].

**Table 2.** Classification results with different sliding window size with AlexNet neural network. The best results are shown in bold.

| Sliding Window Size | Accuracy     |              |              |              | Average Accuracy |
|---------------------|--------------|--------------|--------------|--------------|------------------|
|                     | Branch       | Scratch      | Stain        | Core         |                  |
| [1, 2, 4]           | 88.2%        | 38.7%        | 25.0%        | 69.5%        | 70.5%            |
| [2, 4, 8]           | 84.8%        | <b>40.0%</b> | 20.0%        | 56.5%        | 67.8%            |
| [2, 8, 16]          | 85.3%        | 33.3%        | 25.0%        | 63.6%        | 71.9%            |
| [2, 8, 32]          | 86.2%        | 27.7%        | 27.2%        | 72.0%        | 65.4%            |
| [2, 16, 32]         | 84.3%        | 35.4%        | 17.2%        | 72.0%        | 61.5%            |
| [4, 8, 12]          | <b>90.5%</b> | 39.2%        | <b>30.7%</b> | 76.1%        | <b>76.6%</b>     |
| [4, 8, 16]          | 86.2%        | 35.4%        | 25.0%        | 65.2%        | 71.4%            |
| [8, 16, 32]         | 90.1%        | 33.3%        | 21.4%        | <b>78.2%</b> | 71.8%            |

The best results while using the VGG16 neural network were achieved using 32 batch size and 0.01 learning speed for branch class, 32 batch size and 0.1 learning speed for scratch and background classes, 64 batch size and 0.2 learning speed for core class, and 100 batch size and 0.2 learning speed for background class.

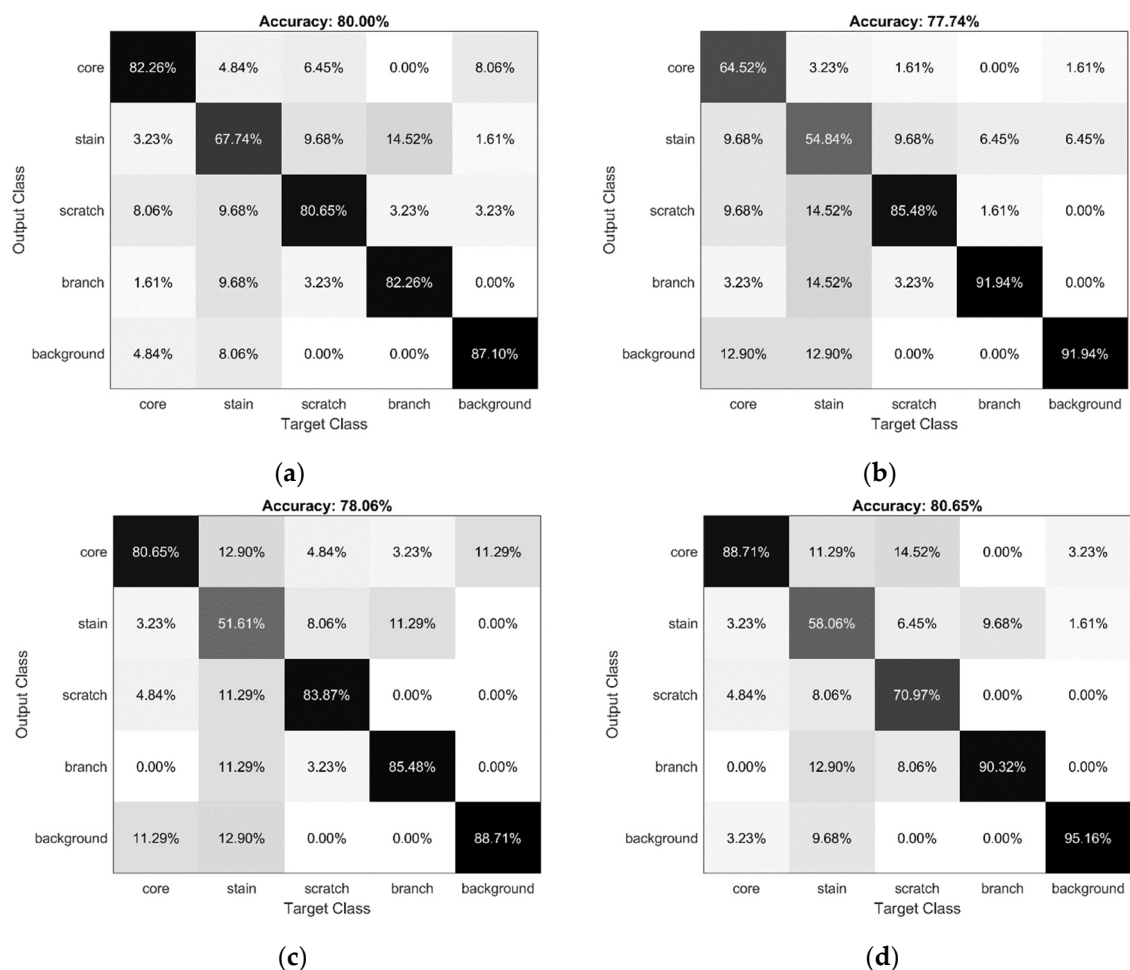
The best results while using the BNInception neural network were achieved using 64 batch size and 0.1 learning speed for core class, 128 batch size and 0.1 learning speed for branch and background

classes, 128 batch size and 0.2 learning speed for stain class, and 256 batch size and 0.2 learning speed for scratch class.

The best results while using the ResNet152 neural network were achieved using 32 batch size and 0.01 learning speed for stain and background class, 32 batch size and 0.1 learning speed for core class, 64 batch size and 0.1 learning speed for scratch class, and 128 batch size and 0.2 learning speed for branch class.

Using an augmented dataset, the total accuracy of the method for finding defects in all classes increased from 60.5% to 68.3%. The accuracy of the quality grade increased from 67.7% to 69.3%. The augmentation of the dataset with synthetic data had the greatest impact on the accuracy of the core defect identification, which increased from 45.8% to 68.0%. The lowest impact was on branch type defects, the accuracy of which decreased from 82.8% to 79.7%.

For transfer learning, each of the four models was pretrained 16 times using different combinations of batch size and learning speed parameters. The confusion matrices for all four transfer learning models are summarized in Figure 17.



**Figure 17.** Confusion matrices of classification results while using (a) AlexNet; (b) VGG16; (c) BNInception; (d) ResNet152.

For AlexNet, most misclassifications occur between the branch and stain classes (14.5%). For VGG16, most misclassifications occur between the stain and scratch classes (14.5%). For BNInception, most misclassifications occur between the stain and core classes (12.9%). For ResNet152, most misclassifications occur between the stain and branch classes (12.9%).

By training an additional classifier using transfer learning, the highest accuracy was achieved by the ResNet152 model-based neural network, which reached 80.6% accuracy for classification, but

was the slowest, as one image was processed in 48.01 ms. The fastest was the AlexNet model-trained method, which reached 80.0% accuracy for classification and processed one image in 6.76 ms, which is 7.1 times faster than the ResNet152 method.

By combining all the defect classes into one type and training the faster R-CNN method, 96.10% accuracy in finding the surface defects was achieved.

A comparison of neural network training speed and classification accuracy for the best neural network models is given in Table 3. The average accuracy value and its 95% confidence intervals were calculated by applying a standard 10-fold cross-validation procedure. The highest overall accuracy of 80.6% was achieved using the ResNet152 pre-trained model. The AlexNet model achieved an average accuracy of 80.2%. The class of branches was best classified using the BNInception pretrained model, which achieved 91.9% accuracy in branch classification. The scratches class was also best identified using the BNInception pretrained model, reaching an accuracy of 86.5%. The best classification accuracy was achieved with the background class that the ResNet152-model-trained model classified with an accuracy of 94.6%. The core defect class was best classified by the ResNet152 model, which reached 89.6% accuracy.

**Table 3.** Accuracy classification when using AlexNet, VGG16, BNInception, and ResNet152 neural network models for transfer learning. The best results are shown in bold.

| Neural Network Model | Average Accuracy of Classes, % |                   |                   |                   |                   | Grand Average Accuracy, % | Performance, ms |
|----------------------|--------------------------------|-------------------|-------------------|-------------------|-------------------|---------------------------|-----------------|
|                      | Branch                         | Scratch           | Stain             | Core              | Background        |                           |                 |
| AlexNet              | <b>82.7 ± 1.8</b>              | 81.7 ± 1.7        | <b>66.0 ± 1.2</b> | 82.7 ± 1.7        | 88.1 ± 1.9        | 80.0 ± 1.7                | <b>6.76</b>     |
| VGG16                | 86.2 ± 2.1                     | 81.7 ± 1.9        | 52.0 ± 1.4        | 81.9 ± 1.9        | 88.1 ± 2.1        | 78.0 ± 1.9                | 23.04           |
| BNInception          | <b>91.9 ± 2.3</b>              | <b>86.5 ± 2.1</b> | 56.0 ± 1.5        | 64.6 ± 2.1        | 92.4 ± 2.3        | 77.7 ± 2.1                | 13.12           |
| ResNet152            | 88.5 ± 1.7                     | 72.1 ± 1.6        | 59.0 ± 1.1        | <b>89.6 ± 1.6</b> | <b>94.6 ± 1.7</b> | <b>80.6 ± 1.6</b>         | 48.01           |

Finally, in Table 4 we summarize the performance of neural networks using precision, recall, and f-score metrics. Although all the analyzed models perform similarly, the best results are obtained by the ResNet152 neural network model.

**Table 4.** Mean precision, recall, and f-score values of AlexNet, VGG16, BNInception, and ResNet152 neural network models used for transfer learning. The best results are shown in bold.

| Neural Network Model | Precision     | Recall        | F-Score       |
|----------------------|---------------|---------------|---------------|
| AlexNet              | 0.7999        | 0.8001        | 0.7998        |
| VGG16                | 0.7804        | 0.7774        | 0.7716        |
| BNInception          | 0.7778        | 0.7806        | 0.7754        |
| ResNet152            | <b>0.8053</b> | <b>0.8065</b> | <b>0.8010</b> |

#### 4. Evaluation and Discussion

In this paper, we have described the development of an automatic visual inspection system for the location and classification of defects on wood veneer surfaces. Each layer of the veneer has to be individually inspected before gluing them all together into plywood. The defective regions of each layer have to be replaced with non-defective pieces of wood. Therefore, the visual inspection system has to output the location of the defective region and the type of defect. In comparison with the results of other authors working on the topic of wood surface defect recognition (see Table 5), our approach works well, achieving an average accuracy of 80.6% with a high performance in terms of defect recognition. The best performance, achieved by the AlexNet architecture (6.76 ms), allows the developed automatic visual inspection system to be used on a wood veneer sorting conveyor line in a real-world industrial wood veneer processing facility. The results can be explained by the combination of the fully connected and convolutional layers, which ensures high recognition performance.

**Table 5.** Comparison of results.

| Article                  | Method                                                                                                                                                                                                                                                                                                                                                                                                       | Results                                                                              |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| Qayyum et al. [55]       | The authors focused on automatic inspection of wood knot defects. They presented a method that extracts image features from GLCM (gray level co-occurrence matrix) and classifies them using a feed-forward neural network (FFNN). ANN was optimized using a particle swarm optimization (PSO) algorithm. The authors used a relatively small database consisting of only 90 images of defective wood knots. | MSE—0.3483<br>Accuracy—78.26%                                                        |
| Packianather et al. [56] | The authors focused in the visual inspection of wood veneer defects. They proposed classifying statistical image features using multilayer ANN. The proposed ANN takes 17 features as inputs and classifies 13 different defect classes. The authors used a dataset of 232 examples of different defects.                                                                                                    | Accuracy—88%                                                                         |
| Zhao et al. [57]         | Damaged wood was chosen as the research object. The authors applied ANN and acoustic emission (AE) for damaged wood detection. Thirty different wood pieces were used in the research, from which 400 AE signals were generated.                                                                                                                                                                             | No objective measurement of accuracy                                                 |
| Qi et al. [58]           | The authors applied the Hopfield neural network dynamic model for the detection of log boundaries. The experimental results showed that the proposed algorithm outputs accurate boundaries for the log. The method was applied on the digital log images acquired using the X-ray imaging system. The number of training and testing images is not specified.                                                | Accuracy is not specified                                                            |
| Chen et al. [59]         | The authors presented a structure-aware crack defect detection scheme. It is based on two mathematical models, where the first one models a cracked surface, and the second one models a structured surface of a solar panel. The proposed method can detect a crack defect in a homogeneously textured background. The dataset consisted of more than 10,000 images.                                        | Accuracy—94.9%<br>Average time—53 ms                                                 |
| Jia et al. [60]          | The authors focused on the detection of unpredictable fabric defects. They proposed an automatic fabric inspection method based on lattice segmentation and template statistics. The defect is detected using the lattice similarity value. The authors used a dataset consisting of 247 image samples of 256 x 256 pixels.                                                                                  | Accuracy 97.7%                                                                       |
| Wang et al. [61]         | The authors proposed a method based on the kernel correlation filter (KCF) target tracking algorithm for the detection of surface defects in moving particleboards on the production line.                                                                                                                                                                                                                   | Surface quality indexes (number, total area and maximum depth of defects) calculated |
| Azizah et al. [62]       | The authors applied CNN for defect detection on fruit surfaces. The proposed algorithm is sensitive to lighting conditions. The authors used a relatively small testing database consisting of only 120 test images.                                                                                                                                                                                         | Precision—97.5%                                                                      |
| Fröhlich et al. [63]     | The authors applied computer vision method and CNN for defect detection in shearography images. Shearography is used to visualize areas of the pipes, which were repaired using glass fiber patches. The training and testing database consisted of 256 shearography images.                                                                                                                                 | Precision—79%                                                                        |
| Konrad et al. [64]       | The authors applied CNN for defect detection on different surfaces in the images, taken by unmanned aerial vehicles. Experiments were performed with 2500 images. The dataset consisted of four different background surfaces and exhibited differences in image lighting and motion blur.                                                                                                                   | Precision—99%                                                                        |
| Tyagi et al. [65]        | The authors applied fine-tuned CNNs for phenotype classification of zebrafish embryo. The proposed algorithm automatically classifies phenotype changes due to toxic substances.                                                                                                                                                                                                                             | Precision—100%                                                                       |
| Raghavendra et al. [66]  | The authors applied deep CNNs for the diagnosis of glaucoma from eye fundus images. They had a dataset consisting of 1426 images (589 images of normal eye fundus and 837 images with glaucoma).                                                                                                                                                                                                             | Precision—98.3%                                                                      |

## 5. Conclusions

In this paper, we have adopted the faster R-CNN neural model for the automated analysis of wood veneer surface quality. We used data augmentation and transfer learning (using pretrained AlexNet, VGG16, BNInception, and ResNet152 models).

The results demonstrated the applicability of data augmentation and transfer learning techniques for the identification of four classes of wood veneer surface defects. The best average accuracy was obtained using the pretrained ResNet152 neural network model (80.6%), while by combining all the defect classes into one type, 96.1% accuracy in finding surface defects was achieved. Our results show that our surface detection method algorithm can be used for industrial wood processing applications. Furthermore, the method can also be adopted for detecting surface defects in other industrially processed materials. More complex data augmentation and transfer learning schemes could be explored to further improve the results.

The limitation of the method is the need to have manually labeled images for the training of neural networks, which may not be error-free.

Future works will include the application of deep learning methods for analyzing the surface defects of other types of wood panels, such as laminate and decorated wood.

**Author Contributions:** Conceptualization, V.R.; Data curation, V.R.; Investigation, A.U. and V.R.; Methodology, V.R.; Resources, A.U. and V.R.; Software, A.U.; Supervision, V.R.; Validation, A.U., V.R. and R.M.; Visualization, A.U., R.M., and R.D.; Writing—original draft, A.U., V.R. and R.M.; Writing—review & editing, R.D.

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

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

## References

1. Forest Products Statistics. Available online: <http://www.fao.org/forestry/statistics/80938/en/> (accessed on 15 January 2019).
2. Gu, I.Y.H.; Andersson, H. Automatic Classification of Wood Defects Using Support Vector Machines. In *Computer Vision and Graphics, ICCVG 2008*; Springer: Berlin/Heidelberg, Germany, 2008.
3. Hashim, U.R.; Hashim, S.Z.; Muda, A.K. Automated vision inspection of timber surface defect: A review. *J. Teknol.* **2015**, *77*, 127–135. [[CrossRef](#)]
4. Lampinen, J.; Smolander, S.; Korhonen, M. Wood Surface Inspection System Based on Generic Visual Features. In *Industrial Applications of Neural Networks*; Soulié, F.F., Gallinari, P., Eds.; World Scientific: Singapore, 1998; pp. 35–42. [[CrossRef](#)]
5. Cetiner, I.; Ali Var, A.; Cetiner, H. Classification of Knot Defect Types Using Wavelets and KNN. *Electron. Electr. Eng.* **2016**, *22*. [[CrossRef](#)]
6. Wang, Y.; Wang, H.; Mo, L. Research on recognition of wood texture based on integrated neural network classifier. In *Proceedings of the International Conference on Intelligent Control and Information Processing, ICICIP 2010, Part 2, Dalian, China, 13–15 August 2010*; pp. 512–515.
7. Wenshu, L.; Lijun, S.; Jinzhuo, W. Study on wood board defect detection based on artificial neural network. *Open Autom. Control. Syst. J.* **2015**, *7*, 290–295. [[CrossRef](#)]
8. Thomas, E. An artificial neural network for real-time hardwood lumber grading. *Comput. Electron. Agric.* **2017**, *132*, 71–75. [[CrossRef](#)]
9. Hu, J.; Song, W.; Zhang, W.; Zhao, Y.; Yilmaz, A. Deep learning for use in lumber classification tasks. *Wood Sci. Technol.* **2019**, *53*, 505–517. [[CrossRef](#)]
10. Loke, K.S. Texture recognition using a novel input layer for deep convolutional neural network. In *Proceedings of the IEEE 3rd International Conference on Communication and Information Systems, ICCIS, Singapore, 28–30 December 2018*; pp. 14–17. [[CrossRef](#)]
11. Karayiannis, Y.A.; Stojanovic, R.; Mitropoulos, P.; Koulamas, C.; Stouraitis, T.; Koubias, S.; Papadopoulos, G. Defect Detection and Classification on Web Textile Fabric Using Multiresolution Decomposition and Neural Networks. In *Proceedings of the 6th IEEE International Conference Electronics, Circuits Systems, Pafos, Cyprus, 5–8 September 1999*; pp. 765–768.

12. Carew, T.; Ghita, O.; Whelan, P.F. A Vision System for Detecting Paint Faults on Painted Slates. In Proceedings of the ICASE International Conference on Control, Automation and Systems, Jeju Island, Korea, 17–21 October 2001.
13. Li, Y.; Ai, J.; Sun, C. Online Fabric Defect Inspection Using Smart Visual Sensors. *Sensors* **2013**, *13*, 4659–4673. [\[CrossRef\]](#)
14. Kumar, A.; Pang, G. Defect detection in textured materials using Gabor filters. *IEEE Trans. Ind. Appl.* **2000**, *38*, 425–440. [\[CrossRef\]](#)
15. Liu, X.; Wen, Z.; Su, Z.; Choi, K.-F. Slub Extraction in Woven Fabric Images Using Gabor Filters. *Text. Res. J.* **2008**, *78*, 320–325. [\[CrossRef\]](#)
16. Chacon, M.I.; Alonso, G.R. Wood Defects Classification Using a SOM/FFP Approach with Minimum Dimension Feature Vector. In *Advances in Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1105–1110.
17. Gu, I.Y.H.; Andersson, H.; Vican, R. Wood defect classification based on image analysis and support vector machines. *Wood Sci. Technol.* **2010**, *44*, 693–704. [\[CrossRef\]](#)
18. Mahram, A.; Shayesteh, M.G.; Jafarpour, S. Classification of wood surface defects with hybrid usage of statistical and textural features. In Proceedings of the 35th International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, 3–4 July 2012; pp. 749–752.
19. YongHua, X.; Jin-Cong, W. Study on the identification of the wood surface defects based on texture features. *Opt. Int. J. Light Electron. Opt.* **2015**, *126*, 2231–2235. [\[CrossRef\]](#)
20. Hittawe, M.M.; Muddamsetty, S.M.; Sidibé, D.; Mériaudeau, F. Multiple features extraction for timber defects detection and classification using SVM. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Quebec, QC, Canada, 27–30 September 2015; pp. 427–431.
21. Zhao, P.; Wang, C.-K. Hardwood Species Classification with Hyperspectral Microscopic Images. *J. Spectrosc.* **2019**, 2019. [\[CrossRef\]](#)
22. Hosang, J.; Benenson, R.; Dollar, P.; Schiele, B. What Makes for Effective Detection Proposals? *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 814–830. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 580–587.
24. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Nasrullah, N.; Sang, J.; Alam, M.S.; Mateen, M.; Cai, B.; Hu, H. Automated lung nodule detection and classification using deep learning combined with multiple strategies. *Sensors* **2019**, *19*, 3722. [\[CrossRef\]](#)
26. Dominguez-Sanchez, A.; Cazorla, M.; Orts-Escolano, S. A new dataset and performance evaluation of a region-based CNN for urban object detection. *Electronics* **2018**, *7*, 301. [\[CrossRef\]](#)
27. 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. [\[CrossRef\]](#)
28. Yuce, B.; Mastrocinque, E.; Packianather, M.S.; Pham, D.; Lambiase, A.; Fruggiero, F. Neural network design and feature selection using principal component analysis and taguchi method for identifying wood veneer defects. *Prod. Manuf. Res.* **2014**, *2*, 291–308. [\[CrossRef\]](#)
29. Ren, R.; Hung, T.; Tan, K.C. A generic deep-learning-based approach for automated surface inspection. *IEEE Trans. Cybern.* **2018**, *48*, 929–940. [\[CrossRef\]](#)
30. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. DeCAF: A deep convolutional activation feature for generic visual recognition. In Proceedings of the 31st International Conference on International Conference on Machine Learning, ICML'14, Beijing, China, 21–26 June 2014.
31. Rudakov, N.; Eerola, T.; Lensu, L.; Kälviäinen, H.; Haario, H. Detection of mechanical damages in sawn timber using convolutional neural networks. *Ger. Conf. Pattern Recognit.* **2019**, 115–126. [\[CrossRef\]](#)
32. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
33. Minaee, S.; Abdolrashidi, A. Deep-emotion: Facial expression recognition using attentional convolutional network. *arXiv* **2019**, arXiv:1902.01019.

34. Minaee, S.; Wang, Y.; Aygar, A.; Chung, S.; Wang, X.; Lui, Y.W.; Fieremans, E.; Flanagan, S.; Rath, J. MTBI Identification From Diffusion MR Images Using Bag of Adversarial Visual Features. *IEEE Trans. Med. Imaging* **2019**, *38*, 2545–2555. [[CrossRef](#)]
35. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
36. Santosh, K.C.; Wendling, L.; Antani, S.; Thoma, G.R. Overlaid Arrow Detection for Labeling Regions of Interest in Biomedical Images. *IEEE Intell. Syst.* **2016**, *31*, 66–75. [[CrossRef](#)]
37. Santosh, K.C.; Wendling, L.; Antani, S.K.; Thoma, G.R. Scalable Arrow Detection in Biomedical Images. In Proceedings of the 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 3257–3262.
38. Santosh, K.C.; Alam, N.; Roy, P.P.; Wendling, L.; Antani, S.; Thoma, G.R. A Simple and Efficient Arrowhead Detection Technique in Biomedical Images. *Int. J. Pattern Recognit. Artif. Intell.* **2016**, *30*, 1657002. [[CrossRef](#)]
39. Cheng, H.D.; Chen, Y.-H. Fuzzy partition of two-dimensional histogram and its application to thresholding. *Pattern Recognit.* **1999**, *32*, 825–843. [[CrossRef](#)]
40. Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
41. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [[CrossRef](#)] [[PubMed](#)]
42. Minaee, S.; Wang, Y. Screen content image segmentation using sparse decomposition and total variation minimization. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3882–3886.
43. Qi, G.-J.; Luo, J. Small Data Challenges in Big Data Era: A Survey of Recent Progress on Unsupervised and Semi-Supervised Method. *arXiv* **2019**, arXiv:1903.11260.
44. Wang, J.; Perez, L. Convolutional Neural Networks Visual Recognition. *arXiv* **2017**, arXiv:1712.04621.
45. Taylor, L.; Nitschke, G. Improving Deep Learning using Generic Data Augmentation. *arXiv* **2017**, arXiv:1708.06020.
46. Boukli Hacene, G.; Gripon, V.; Farrugia, N.; Arzel, M.; Jezequel, M. Transfer Incremental Learning Using Data Augmentation. *Appl. Sci.* **2018**, *8*, 2512. [[CrossRef](#)]
47. Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083.
48. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [[CrossRef](#)]
49. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *6*, 84–90. [[CrossRef](#)]
50. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
51. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
52. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
53. Lin, M.; Chen, Q. Network in Network. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
54. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML'15), Lille, France, 6–11 July 2015; pp. 448–456.
55. Qayyum, R.; Kamal, K.; Zafar, T.; Mathavan, S. Wood defects classification using GLCM based features and PSO trained neural network. In Proceedings of the 22nd International Conference on Automation and Computing (ICAC), Colchester, UK, 7–8 September 2016; pp. 273–277.

56. Packianather, M.S.; Drake, P.R.; Pham, D.T. Feature selection method for neural network for the classification of wood veneer defects. In Proceedings of the World Automation Congress, Hawaii, HI, USA, 28 September–2 October 2008; pp. 1–6.
57. Zhao, D. Automated Recognition of Wood Damages Using Artificial Neural Network. In Proceedings of the International Conference on Measuring Technology and Mechatronics Automation, Zhangjiajie, China, 11–12 April 2009; pp. 195–197.
58. Qi, D.; Zhang, P.; Jin, X.; Zhang, X. Applying Hopfield neural network to defect edge detection of wood image. In Proceedings of the 6th International Conference on Natural Computation, Yantai, China, 10–12 August 2010; pp. 1459–1463.
59. Chen, H.; Zhao, H.; Han, D.; Liu, W.; Chen, P.; Liu, K. Structure-Aware-based Crack Defect Detection for Multicrystalline Solar Cells. *Measurement* **2019**. [\[CrossRef\]](#)
60. Jia, L.; Chen, C.; Xu, S.; Shen, J. Fabric defect inspection based on lattice segmentation and template statistics. *Inf. Sci.* **2019**. [\[CrossRef\]](#)
61. Wang, C.; Liu, Y.; Wang, P. Extraction and Detection of Surface Defects in Particleboards by Tracking Moving Targets. *Algorithms* **2019**, *12*, 6. [\[CrossRef\]](#)
62. Azizah, L.M.; Umayah, S.F.; Riyadi, S.; Damarjati, C.; Utama, N.A. Deep learning implementation using convolutional neural network in mangosteen surface defect detection. In Proceedings of the 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 24–26 November 2017; pp. 242–246.
63. Frohlich, H.B.; Fantin, A.V.; de Oliveira, B.C.F.; Willemann, D.P.; Iervolino, L.A.; Benedet, M.E.; Goncalves, A.A., Jr. Defect classification in shearography images using convolutional neural networks. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–7.
64. Konrad, T.; Lohmann, L.; Abel, D. Surface Defect Detection for Automated Inspection Systems using Convolutional Neural Networks. In Proceedings of the 27th Mediterranean Conference on Control and Automation (MED), Akko, Israel, 1–4 July 2019; pp. 75–80.
65. Tyagi, G.; Patel, N.; Sethi, I. A Fine-Tuned Convolution Neural Network Based Approach For Phenotype Classification Of Zebrafish Embryo. *Procedia Comput. Sci.* **2018**, *126*, 1138–1144. [\[CrossRef\]](#)
66. Raghavendra, U.; Fujita, H.; Bhandary, S.V.; Gudigar, A.; Tan, J.H.; Acharya, U.R. Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images. *Inf. Sci.* **2018**, *441*, 41–49. [\[CrossRef\]](#)



© 2019 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/>).