Lappeenranta University of Technology
School of Engineering Science
Master's Programme in Computational Engineering and Technical Physics
Intelligent Computing Major

Master's Thesis

Nikolay Rudakov

DETECTION OF MECHANICAL DAMAGES IN SAWN TIMBER USING CONVOLUTIONAL NEURAL NETWORKS

Examiners: Professor Heikki Kälviäinen
Adjunct Professor, Dr. Tuomas Eerola

Supervisors: Adjunct Professor, Dr. Tuomas Eerola
Professor Lasse Lensu
Professor Heikki Haario
Professor Heikki Kälviäinen

ABSTRACT

Lappeenranta University of Technology
School of Engineering Science
Master's Programme in Computational Engineering and Technical Physics
Intelligent Computing Major

Nikolay Rudakov

Detection of mechanical damages in sawn timber using convolutional neural networks

Master's Thesis

2018

55 pages, 41 figures, 3 tables.

Examiners: Professor Heikki Kälviäinen
Adjunct Professor, Dr. Tuomas Eerola

Keywords: computer vision, image processing, pattern recognition, convolutional neural networks, damage detection, surface inspection, sawn timber

Quality control of timber products is a vital issue for the sawmill industry. One of the laborious parts of this control is to search mechanical damages in boards. Automation of this process decreases the time needed for board inspection and, therefore, improves the overall quality of timber products. Distinguishing barely noticeable mechanical damage from high contrast natural wood textures is a challenging problem. This study aimed to develop a robust and efficient method for recognition and localization of various mechanical damages on the images of sawn timber. Convolutional neural networks (CNN) was selected as a method to determine the existence of the mechanical damages on parts of the board images. In the proposed method the patches are extracted from the image of the board and classified with the CNN. The defects are localized based on the CNN prediction and coordinates of the patches. In this study four existing CNN architectures, namely AlexNet, GoogLeNet, VGG-16, and ResNet-50 were tested for mechanical damages detection. The VGG-16 architecture achieved the best results with over 90% classification accuracy for the image patches and 58% average defect localization accuracy.

PREFACE

I would like to thank my supervisors, LUT and ITMO University staff and my family for supporting me in my studies.

Lappeenranta, May 25, 2018

Nikolay Rudakov

CONTENTS

1 INTRODUCTION **7**
1.1 Background . . . 7
1.2 Objectives and delimitations . . . 8
1.3 Structure of the thesis . . . 9

2 QUALITY INSPECTION OF SAWN TIMBER **10**
2.1 Defect types . . . 10
2.2 Sensors for quality inspection . . . 10
2.3 Defect detection using computer vision . . . 11
2.3.1 Local binary patterns . . . 11
2.3.2 Gray level co-occurrence matrix . . . 12
2.3.3 Scale-invariant feature transform . . . 12
2.3.4 Speeded up robust features . . . 13
2.3.5 Tamura texture features . . . 14
2.3.6 Convolutional neural networks . . . 15

3 OBJECT DETECTION WITH CONVOLUTIONAL NEURAL NETWORKS **18**
3.1 Object detection by classification . . . 19
3.2 Regions with CNN features . . . 21
3.3 Fast R-CNN . . . 21
3.4 Faster R-CNN . . . 22
3.5 Mask R-CNN . . . 23
3.6 You Only Look Once . . . 23
3.7 Region-based Fully Convolutional Network . . . 25

4 PROPOSED METHODS **27**
4.1 Patch extraction . . . 28
4.2 Classification with convolutional neural networks . . . 32
4.2.1 AlexNet . . . 33
4.2.2 GoogLeNet . . . 33
4.2.3 VGG . . . 34
4.2.4 ResNet . . . 35
4.3 Damage localization . . . 36

5 EXPERIMENTS **38**
5.1 Data . . . 38
5.2 Evaluation criteria . . . 40
5.3 Description of experiments . . . 41

5.4 Results 42
5.4.1 Experiment 1: classification of the combined mechanical damages 42
5.4.2 Experiment 2: classification of the feed roller periodic traces . . . 42
5.4.3 Inference time 43

6 DISCUSSION **48**
6.1 Current study 48
6.2 Future work 49

7 CONCLUSION **50**

REFERENCES **51**

LIST OF ABBREVIATIONS

ASI	Automated Surface Inspection
CNN	Convolutional Neural Network
DoG	Difference of Gaussians
FCN	Fully-Convolutional Network
GLCM	Gray Level Co-occurrence matrix
GPU	Graphical Processing Unit
IoU	Intersection over Union
LBP	Local Binary Patterns
MLR	Multinomial Logical Regression
R-CNN	Regions with Convolutional Neural Network features
R-FCN	Region-based Fully-Convolutional Network
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
RPN	Region Proposal Network
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
YOLO	You Only Look Once

1 INTRODUCTION

1.1 Background

Nowadays the sawmill industry plays an important role in our life. Since sawn timber is used as construction material, it is especially vital to control the quality of the output production. The purpose of this research is to develop a computer vision system that can detect and recognize mechanical damages on sawn timber. Sawn timber can be mechanically damaged during the production process. Usually those defects appear as a series of marks or traces along the board. Typically, marks appear because the feed rollers (see Figure 1) may excessively press the board their spikes while moving it through the sawing machine [1].

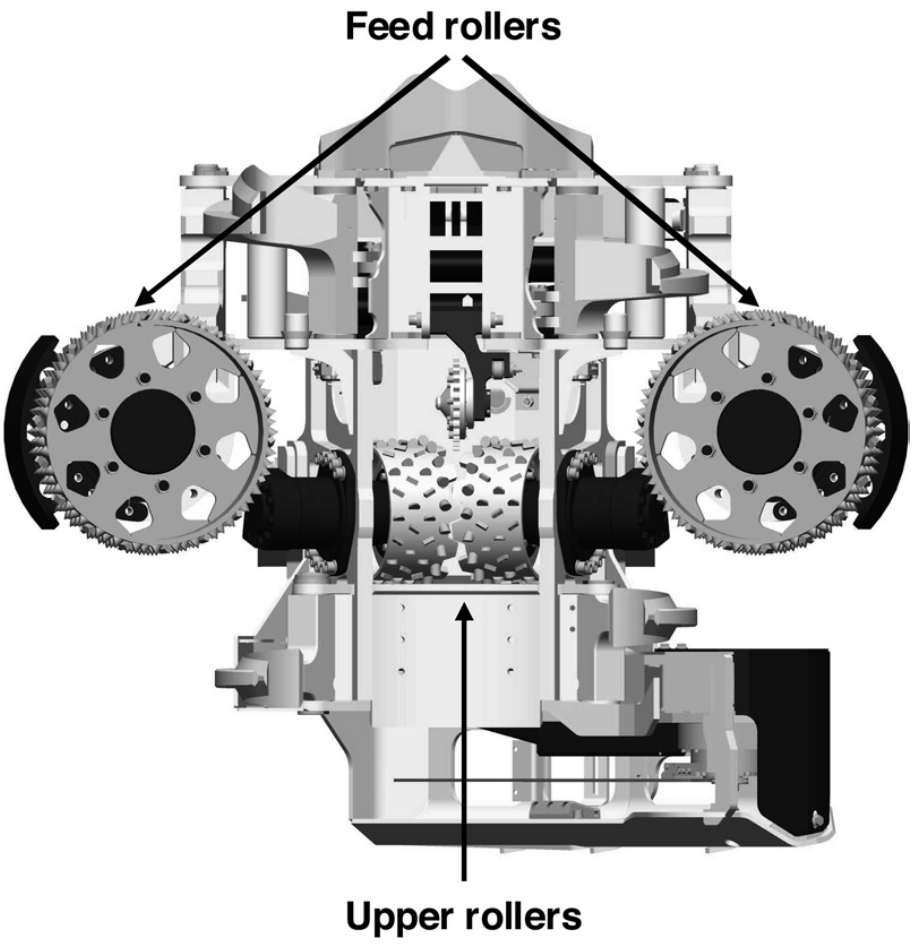


Figure 1. Feed rollers on harvester head. [1]

An example of a mechanical damage on sawn timber is shown in Figure 2. It should be noted that not every board surface unevenness is a mechanical damage. For instance, wanes have a natural origin, and thus they should not be defined as mechanical. Although, it may be useful to detect different kinds of defects for the flexibility of the inspection system. Convolutional neural network (CNN) is a leading technique for image classification and object detection [2]. Being a universal function approximator, CNN can be potentially trained to extract almost any high-level features from the input image including features of mechanical damages of sawn timber. Therefore, they were selected as a feature extractor and classifier for the images of sawn timber. This Master's thesis is a part of the DigiSaw research project [3]. The DigiSaw research project targets on enhancement the productivity of the sawmill industry using modern digital technologies. It is believed that computer vision is the most time- and cost-efficient technique to automate verification of the product quality.

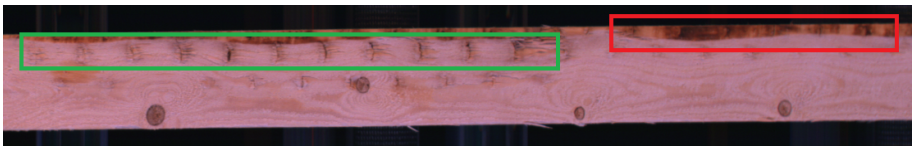


Figure 2. An example of feed roller traces as mechanical damage is surrounded with green bounding box. Wane, which is not a mechanical damage, is highlighted with red bounding box.

1.2 Objectives and delimitations

The main objectives of this master's thesis are as follows:

1. to prepare an annotated dataset to train and to evaluate computer vision methods for the mechanical damages detection in sawn timber task,
2. to construct, train, and evaluate a convolutional neural network based method to detect mechanical damages in sawn timber and
3. to compare the accuracy of different convolutional neural networks architectures.

The prospective damage detection system is supposed to work the following way. The color image of the timber taken with a digital camera is used as input data. The system should return locations and types of mechanical damages if they are present on that timber. Optionally, the system could also detect other kind of defects, but in this study only

mechanical damages are considered. The principal idea of the proposed system is shown in Figure 3.

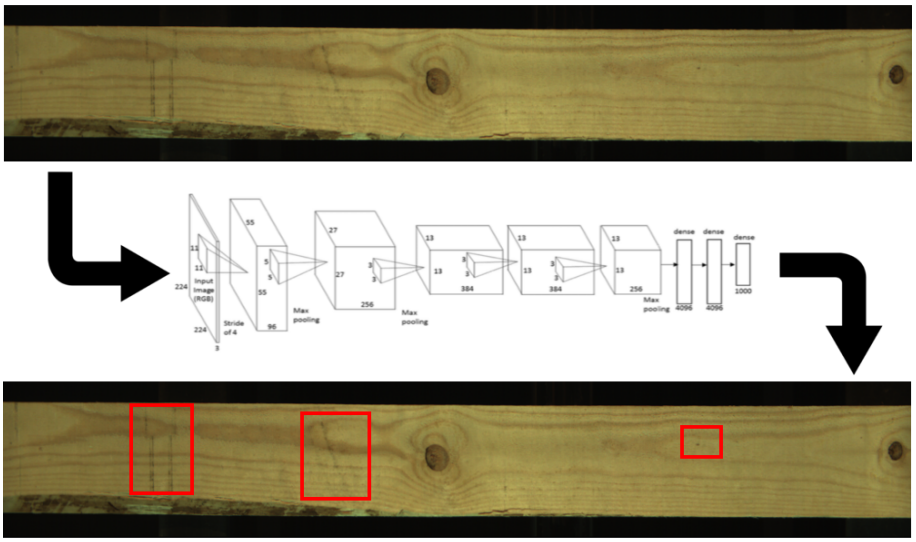


Figure 3. The basic idea of the mechanical damages detection system. The proposed algorithm should find the locations of mechanical damages on the image of sawn timber.

1.3 Structure of the thesis

This thesis is organized as follows. Chapter 2 gives review on existing cases of surface inspection. Different approaches of convolutional neural network based methods for object detection task are listed in Chapter 3. Chapter 4 contains detailed description of the proposed method for detection of sawn timber mechanical damages. Result analysis of the method can be found in Chapter 5. In Chapter 6, the discussion of achieved results and possible future issues are mentioned. Finally, the conclusions are drawn in Chapter 7.

2 QUALITY INSPECTION OF SAWN TIMBER

2.1 Defect types

A lot of research has been made previously in the field of surface inspection in the sawmill industry including defects detection. Most of the existing researches consider detection of natural timber defects such as different kinds of knots, bark pockets, wane, fungus, worm holes, cracks, and resin [4]. Examples of the defect types are shown in Figure 4.

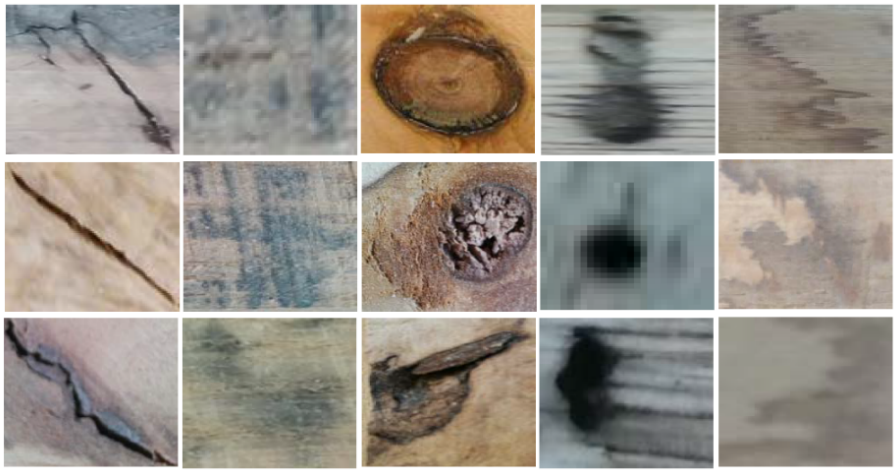


Figure 4. Examples of several defect types. From the left to the right: crack, fungus, knot, worm hole, watermark. [5]

2.2 Sensors for quality inspection

Different sensors have been used on timber surface and inwards inspection. Using of ultrasonic measurements allows to detect internal defects. For example, in [6] the quality of wood specimen was inspected by using 50 kHz sound source and sound receiver with a contact to the wood surface. Depending on the sound travel time, the locations of holes were detected. Similar technique was used in [7] to detect cavity defects inside wooden logs. Computed tomography X-ray scanners were used in [8] and [9] to detect the placement of knots inside a log. Using of infrared thermography for internal defects detection is introduced in [10] and [11]. The analysis of emitted radiation from the surface can get the information of the internal structure. In [12] the surface defects were detected with

imaging spectrograph. All mentioned above approaches requires either high-cost equipment or precise calibration and time to install the sensors and obtain the measurements. Using of a digital camera and computer vision approach for timber quality inspection task tends to be the most common because of price-efficiency and high performance [4].

2.3 Defect detection using computer vision

There are many existing solutions for defect detection on sawn timber surface. The used methods, detected defect types and image requirements are individual for each particular issue, but three common stages can be highlighted. Firstly, the defects should be localized. Secondly, the features of each defect are extracted. Finally, the feature-based classification of the defects is performed [4]. There are several reviews on existing solutions which are comparing the performances and categorizing them based on the methods used for defect localization, feature extraction and defect classification [4, 5, 13].

The feature localization stage can be implemented with the combination of various image filtering, segmentation, thresholding, connected components labeling, region merging and many other image processing techniques. Most of the previous studies have used texture features for timber defect classification [5]. The most common feature extraction methods are local binary patterns (LBP) [14], gray level co-occurrence matrix (GLCM) [15], scale-invariant feature transform (SIFT) [16], speeded up robust features (SURF) [17] and Tamura texture features [18].

2.3.1 Local binary patterns

The LBP algorithm [14] splits the analyzed image into square blocks. Each pixel within a block is compared with pixels of its 8-neighborhood to create the binary mask. The elements of the mask have zero value if the central pixel has higher brightness than the corresponding pixel, otherwise the elements have value of one. The binary mask excluding its central element represents an integer value. The next step is to compute histogram of the obtained integer values within a block. Finally, the corresponding bins of histograms from all blocks are summed. The result of LBP is a vector of n texture features, where n equals to the number of bins in the histogram.

2.3.2 Gray level co-occurrence matrix

The GLCM operator [15] has two arguments: direction and distance and it describes the spatial-brightness frequencies of the pixels on the image. The element of GLCM matrix C_{ij} represents how often the pixel with j value occurs in the given direction and at a given distance from the pixel with value i. To analyze the GLCM matrix statistical descriptors can be applied to the matrix.

2.3.3 Scale-invariant feature transform

In the SIFT algorithm [16] the analyzed image is convolved with several Gaussian functions of different deviations σ as a scale parameter as

$$L(x, y, \sigma) = I(x, y) * G(x, y, \sigma), \tag{1}$$

where $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$, x and y are the coordinates of a pixel.

Next, the difference of Gaussians (DoG) is computed as

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma), \tag{2}$$

where k is a scale factor. and the DoG function D is interpolated by the Taylor expansion as

$$D(\boldsymbol{t}) = D + \frac{\partial D^T}{\partial \boldsymbol{t}}\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}^T\frac{\partial^2 D}{\partial \boldsymbol{t}^2}\boldsymbol{t}, \tag{3}$$

where $\boldsymbol{t} = (x, y, \sigma)$. Thresholded local extrema of interpolated DoG functions are considered as a keypoints. The magnitude m and the orientation Θ are computed based on convolved images as

$$m(x, y, \sigma) = \sqrt{\begin{matrix}(L(x+1, y, \sigma) - L(x-1, y, \sigma))^2 + \\ (L(x, y+1, \sigma) - L(x, y-1, \sigma))^2\end{matrix}} \tag{4}$$

$$\Theta(x, y, \sigma) = \arctan\left(\frac{\partial L(x, y, \sigma)}{\partial y} \Big/ \frac{\partial L(x, y, \sigma)}{\partial x}\right) \tag{5}$$

The area around the keypoint is split into 4x4 pixels regions where the 8-bin histogram of average gradient absolute values along every orientation within 16 regions around the keypoint are computed. As a result, a feature vector of 128 elements is used to represent every keypoint.

2.3.4 Speeded up robust features

The SURF detector [17] is based on an integral image and the rectangular approximation of Gaussian second-order partial derivatives. Each pixel of the integral image is equal to the sum of intensities of all pixels above and left from the current pixel and defines as

$$I_{\text{integral}}(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \tag{6}$$

The second order partial derivatives of Gaussian function that is used in the SURF detectors are shown in Figure 5. The detector response is computed as a weighted determinant of the Hessian matrix

$$H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix}, \tag{7}$$

where $L(x, y, \sigma)$ is a convolution of rectangular approximation of second order Gaussian function with an integral image. The weighted discriminant of this matrix is determined the following way:

$$\det H(x, y, \sigma) = L_{xx} L_{yy} - (0.9 L_{xy})^2 \tag{8}$$

After non-maximum suppression is applied to Hessian discriminant responses, the local maximums of the responses represent the keypoints. The descriptor of keypoint is computed with Haar wavelet responses around the keypoint.

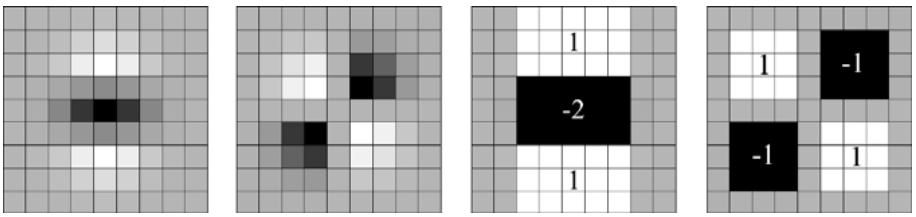


Figure 5. The second order derivatives of Gaussian functions in xx and xy directions (two left images) and their rectangular approximations (two right images). [17]

2.3.5 Tamura texture features

Tamura features [18] consist of six computational measures: coarseness, contrast, directionality, line-likeness, regularity, and roughness.

Coarseness feature is computing as

$$F_{\text{crs}} = \frac{1}{mn} \sum_{i}^{m} \sum_{j}^{n} S_{\text{best}}(i, j), \tag{9}$$

where m and n are the image dimensions, $S_{\text{best}} = 2^k$. The exponent k is defined by

$$k = \arg\max_{k} \{|A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y)|\}, \tag{10}$$

where $A_k(x, y)$ is an average value of the pixels within the square neighborhood with the side of 2^k pixels and computed as

$$A_k(x, y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} \frac{f(i, j)}{2^{2k}} \tag{11}$$

The function for the contrast feature is defined as

$$F_{\text{con}} = \frac{\sigma}{\sqrt[4]{\alpha_4}}, \tag{12}$$

where σ is a standard deviation of pixels values and α_4 is a kurtosis.

The directionality feature is obtained by applying Prewitt edge detector to find the magnitudes and directions of the edges. Next, the histogram of edge directions is computed, the peaks and valleys on the histogram are found. Finally, the directionality feature is computed as

$$F_{\text{dir}} = 1 - rn \sum_{p=1}^{n} \sum_{\phi \subseteq w_p} (\phi - \phi_p)^2 H_D(\phi), \tag{13}$$

where r is a normalization factor, n_p is a number of peaks in the histogram, ϕ is a direction, w_p is a range of directions between two nested valleys where p-th peak is located, ϕ_p is a location of the p-th peak of the histogram H_D.

Line-likeness feature requires to compute direction co-occurrence matrix. Direction co-occurrence matrix is similar to GLCM, but the elements represent statistical information

about the edge directions on specified distance instead of pixel values. The formula of line-likeness feature is

$$F_{\text{lin}} = \sum_{i}^{m}\sum_{j}^{n} P_d \cos[(i-j)\frac{2\pi}{n}] / \sum_{i}^{n}\sum_{j}^{n} P_d(i,j), \tag{14}$$

where P_d is a direction co-occurrence matrix for the distance d.

The regularity feature is a combination of standard deviations of the four features and it is calculated as

$$F_{\text{reg}} = 1 - r(\sigma_{\text{crs}} + \sigma_{\text{con}} + \sigma_{\text{dir}} + \sigma_{\text{lin}}), \tag{15}$$

where r is a normalization coefficient.

The last of Tamura features is roughness and it is a sum of coarseness and contrast features:

$$F_{\text{rgh}} = F_{\text{crs}} + F_{\text{con}} \tag{16}$$

2.3.6 Convolutional neural networks

With the increase of computer performance, deep learning approach has become more popular in image processing [19]. The most recent articles considering defect detection rely on deep learning approach using convolutional neural networks (CNN). Ruoxu Ren et al. [13] introduced the Generic Deep-Learning-Based Approach for Automated Surface Inspection (ASI) consists of the following steps:

1. The pretrained Decaf CNN [20] as a feature extractor and a classifier.
2. Heatmap generation.
3. The binarization using the Otsu method [21].
4. The graph-based Felzenszwalb's segmentation method [22] for defect localization.

The overview of the method is presented in Figure 6.

In [23] a method for crack detection on concrete surface also using a single CNN was introduced. The image is processed using a sliding window, and the corresponding part of the image is fed to CNN for each position of the window. The CNN is trained to check

the existence of a crack on a part of the image. Figure 7 contains the overview of the method. The CNN architecture is shown in Figure 8.

The principals of CNN and approaches of using CNN in object detection problems are described in more detail in Chapter 3.

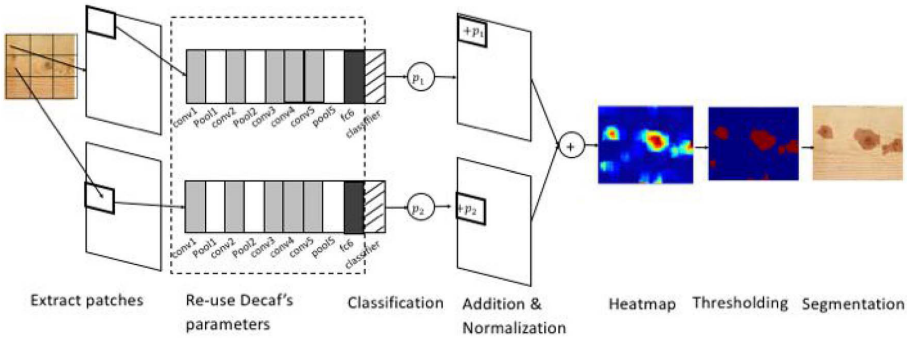


Figure 6. Deep learning based approach for automated surface inspection. An input image is divided by a grid into sub-images. Each sub-image is processed independently. [13]

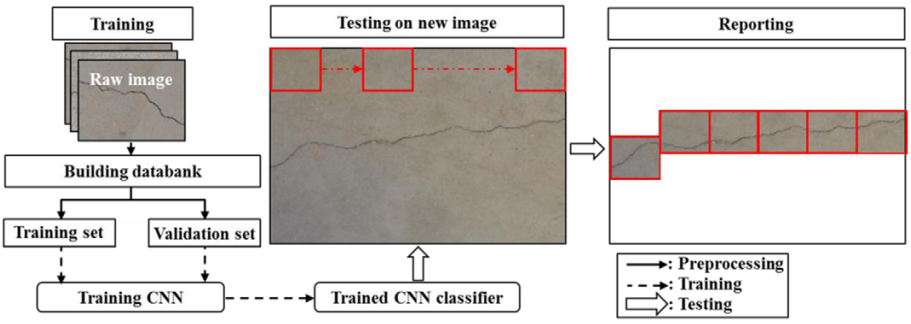


Figure 7. The use of CNN as a defect detector. A sliding window moves through an input image. CNN performs classification for each window position. [23]

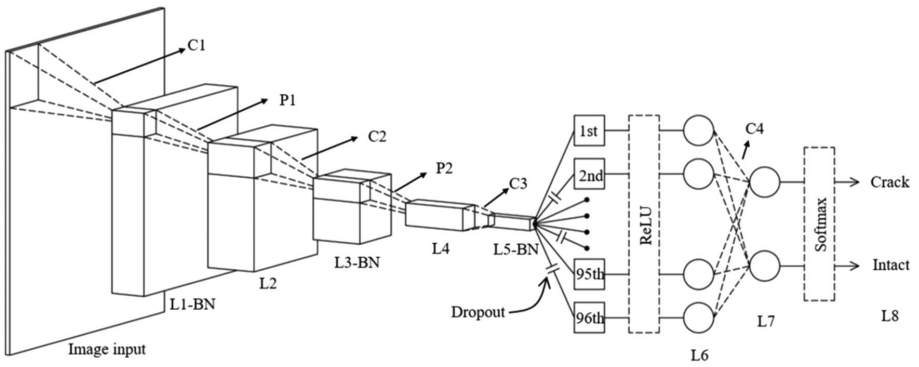


Figure 8. The CNN architecture for crack detection in concrete surface. The CNN consists of three convolutional and three pooling layers. The two fully connected layers complete the network. The outputs of the network are two numbers: the probability of a crack existence and the probability of an intact region. [23]

3 OBJECT DETECTION WITH CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural network (CNN) is a type of neural networks which was originally introduced by Yann LeCun et al. [24] to adopt a neural network approach [25] for handwritten digits recognition. A classical fully-connected neural network is not suitable for working with images as an input because of enormous number of weights needed to be trained.

The typical CNN architecture is presented in Figure 9. The idea of CNN is to use the repeated combination of convolutional and pooling layers [24]. The output of each convolutional layer is a convolution operation with a set of moving windows of weights called kernels or filters. Multilayer networks can be treated as a universal function approximator because of applying nonlinear activation function to the neuron output [26]. The most common activation functions are as follows:

- sigmoid

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{17}$$

- hyperbolic tangent

$$f(x) = \tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \tag{18}$$

- rectified linear unit (ReLU)

$$f(x) = \max(0, x) \tag{19}$$

During the training stage each kernel is learning for specific local features on the images. To reduce the output dimensions of convolutional layer, pooling layers are used. Typically, the output image from convolutional layer is split into non-overlapping regions and maximum function is applied to each region. After the series of convolutional and pooling layers, several classical fully-connected layers are added. In other words, CNN is learning to extract high-level feature maps from the image and perform final classification based on those maps but not the entire image. The benefit of CNN is that the features are taken from pixels located close to each other, while in classical neural network the features based on distant pixels have the same importance as local features because of full-connectivity of such networks [24].

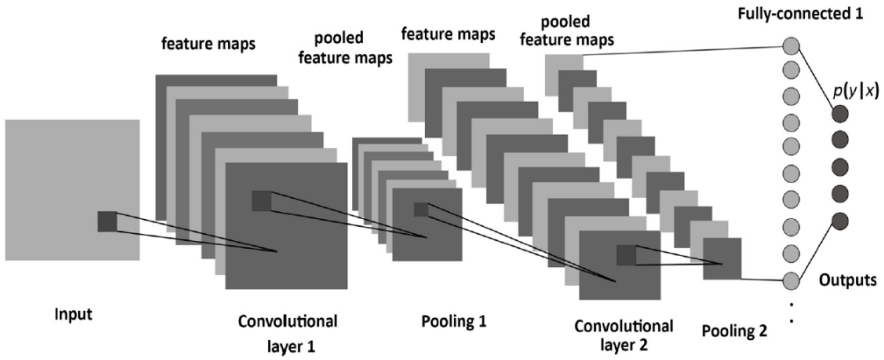


Figure 9. The typical CNN architecture. [27]

Since the number of weights that are needed to be trained is usually relatively high comparing to the amount of training data samples, the CNNs are prone to overfitting. It means that the model learns training data excellent but demonstrating poor performance on new data that was not used in the training process. Even more, CNNs can fail to be trained at all, if the database is small. To improve learning ability in case of insufficient size of training database, transfer learning [28] can be used. Transfer learning involves using the pretrained weights from the model that were trained for a similar problem on another larger image dataset rather than training from scratch.

The dropout technique [29] was introduced by Hinton et al. to improve generalization ability of the neural networks. Neuron weights can be set to zero or dropped out with the predefined probability. If the weight is zero it means that the corresponding neuron does not contribute to the network output and therefore is not participating in current training iteration. So, for each new training sample the architecture is modified by randomly dropping out new set of neurons.

The optimal number of layers, number of convolution kernels for each layer, dimensionality and strides of kernels, activation functions and other parameters are specific for each task. They can be determined experimentally [19].

3.1 Object detection by classification

CNN is the state-of-the-art approach for image classification task according to the results of ImageNet large scale visual recognition challenges starting from 2012 [2]. Not only CNN is a universal approach to classify images, but also it can be used to solve the de-

tection task. The image that is fed to the CNN can be a local part of some larger image. In this case, if the input image is classified to particular class, the knowledge of image location on the larger image can also be interpreted as a location of the object. A sliding window technique can be used to classify part of the entire image for every position of the window and therefore to detect objects. This approach has been implemented in several solutions [13, 23, 30, 31]. The steps of the sliding window approach are shown in Figure 10.

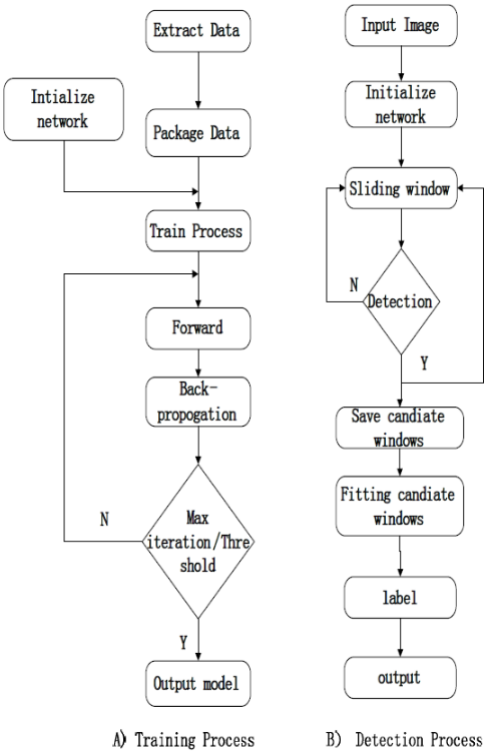


Figure 10. Steps of the sliding window approach for object detection. [31]

There are many successful CNN architectures. AlexNet [26] was the first CNN successfully dealing with relatively large square images with 227 pixel sides. GoogLeNet [32] is a much deeper CNN requiring 10 times less parameters than AlexNet. VGG [33] uses only small convolutional kernels. ResNet [34] uses additional connections between layers to make training of very deep CNNs possible.

The main disadvantage of the sliding window technique is that it requires a lot of images to be fed into network, therefore the time needed for detection depends on the size of the entire image. To reduce the processing time, the window stride can be increased, but it would negatively affect the detection resolution [35].

3.2 Regions with CNN features

To efficiently detect objects on images using CNN a Regions with CNN features (R-CNN) method was developed in 2014 by Ross Girshick et al. [36]. The overview of the method is shown in Figure 11.

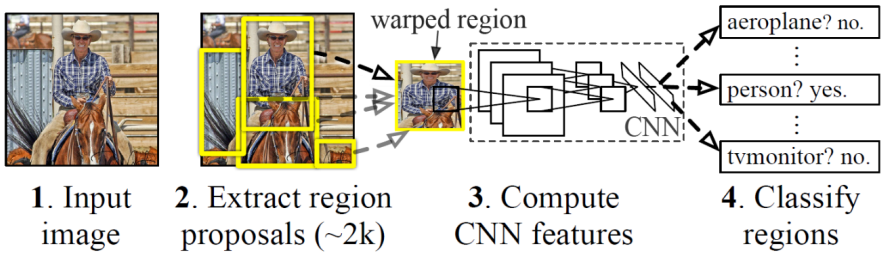


Figure 11. R-CNN method overview. [36]

The first step of this method is to propose regions containing desired objects. A selective search algorithm was selected for this purpose [37]. Selective search performs initial image segmentation with graph-based method developed by Pedro Felzenszwalb and Daniel Huttenlocher [22]. Then the regions are hierarchically grouped based on their similarity. The object locations hypotheses as a bounding boxes are generated using the grouping information. After the potential locations of the objects are found, they are cropped out from the original image. Every cropped sub-image is fed to CNN input for feature extraction. Finally, the extracted features are classified with linear support vector machines (SVMs) distinct for each class.

3.3 Fast R-CNN

In [38] a new method to detect objects called Fast R-CNN was introduced. The architecture of the Fast R-CNN approach is presented in Figure 12. Nonlinearity Object proposals are generated the same way with the selective search algorithm. The performance of an object detection is improved by feeding the whole image with potential object locations to a single CNN. After the feature map for the image are computed, the features for every proposal are extracted with a pooling layer into a feature vector. The vectors are processed with a fully-connected network to get the classes probabilities and the bounding box approximation of an object. The Fast R-CNN method has $9\times$ improvement in train-

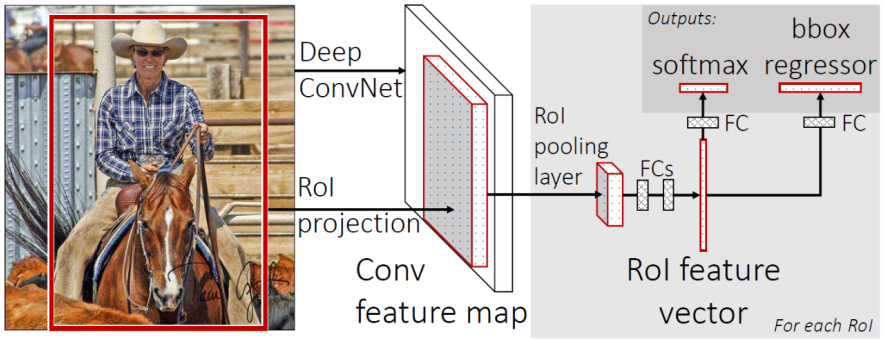


Figure 12. Fast R-CNN method architecture. [36]

ing time, 213× faster in testing and it also shows better detection accuracy than R-CNN method [38].

3.4 Faster R-CNN

The main disadvantage of the Fast R-CNN approach is that the input image should be preprocessed to estimate the objects locations. Faster R-CNN [39] is an object detection approach where the region proposal network (RPN) and the Fast R-CNN use shared convolutional layers. The principals of RPN are shown in Figure 13.

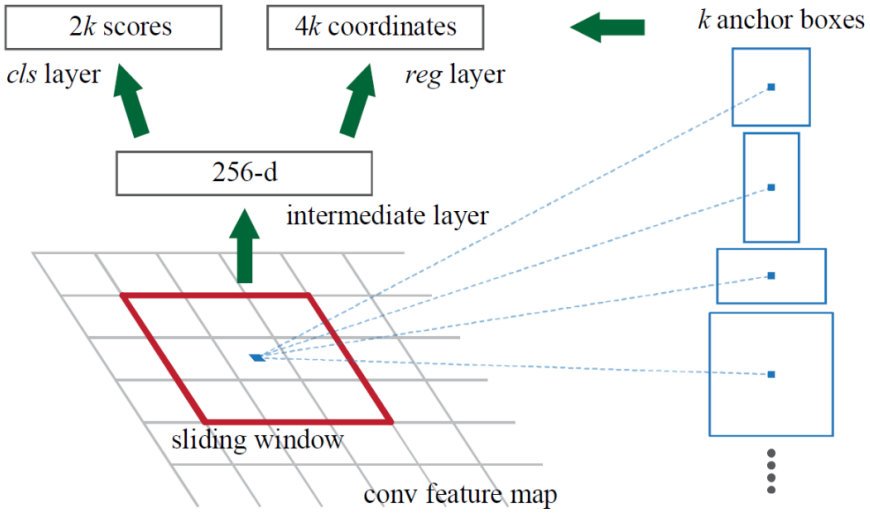


Figure 13. Region proposal network (RPN) for object candidates localization. [39]

Instead of selective search segmentation, a window of co-centered anchors with different aspect ratios slides over the feature maps of the input image from the last convolutional layer. For each position of the window, the feature vector is extracted and forwarded to the pair of fully connected layers. The outputs of the first one is the confidence level whether an object and not-object is located in current window position. The outputs of the second layer are the coordinates and dimensions of the object proposal bounding box. Since the proposals are located, the rest of object detection process are the same as in Fast R-CNN method [39].

3.5 Mask R-CNN

Mask R-CNN [40] is a succeeding enhancement of the Faster R-CNN method. In this modification, the third branch is added to the object candidates processing stage of Faster R-CNN. The new branch is an additional fully-convolutional network (FCN) [41] that produces a binary mask of an object. The Mask R-CNN approach is presented in Figure 14.

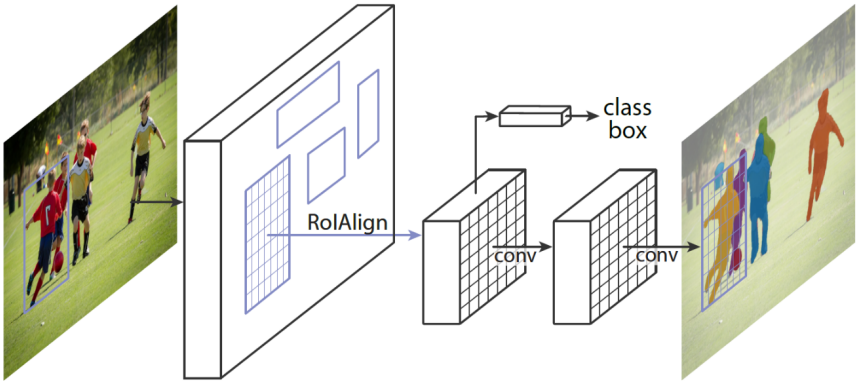


Figure 14. The basics of Mask R-CNN approach. [40]

3.6 You Only Look Once

You Only Look Once (YOLO) is a fast method to object detection which was introduced in [42]. The researchers were inspired by the fact that humans are able to recognize an objects presented at the image and their relationship or interaction immediately. The main idea of the YOLO approach is to combine the steps of the R-CNN method such as

proposal detection, classification and post-processing into a single network. The steps of the YOLO method are presented in Figure 15.

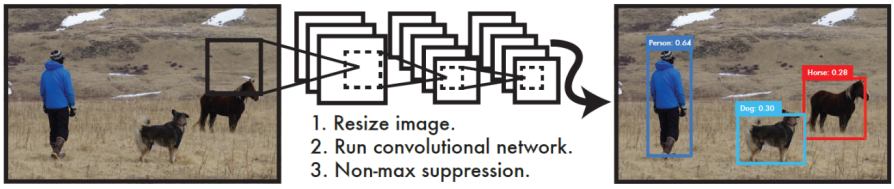


Figure 15. YOLO approach. [42]

Since YOLO works only with images of single size. Thus, all input images should be reduced or expand to the size equal to the dimensions of the first layer in CNN. Then the image is divided into square grid. The CNN is trained to predict the bounding boxes dimensions and coordinates of the center, confidence level and class probabilities for each cell of the grid. Every cell of the grid is responsible for constant number of bounding boxes. However, there are only one set of class probabilities per cell. The architecture of the CNN in YOLO approach is shown in Figure 16 and the method overview is shown in Figure 17.

The main disadvantage of the YOLO method is that it can detect only a limited number of closely located objects. However, YOLO can process images at relatively high rate of 40 images per second. Thus, YOLO can be used in real-time [42].

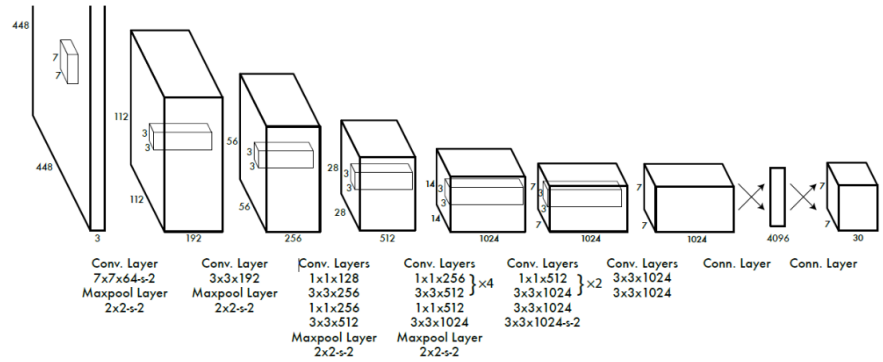


Figure 16. Architecture of CNN used in YOLO. The numbers represent dimensions of the layers, dimensions and strides of the convolutional and pooling kernels. [42]

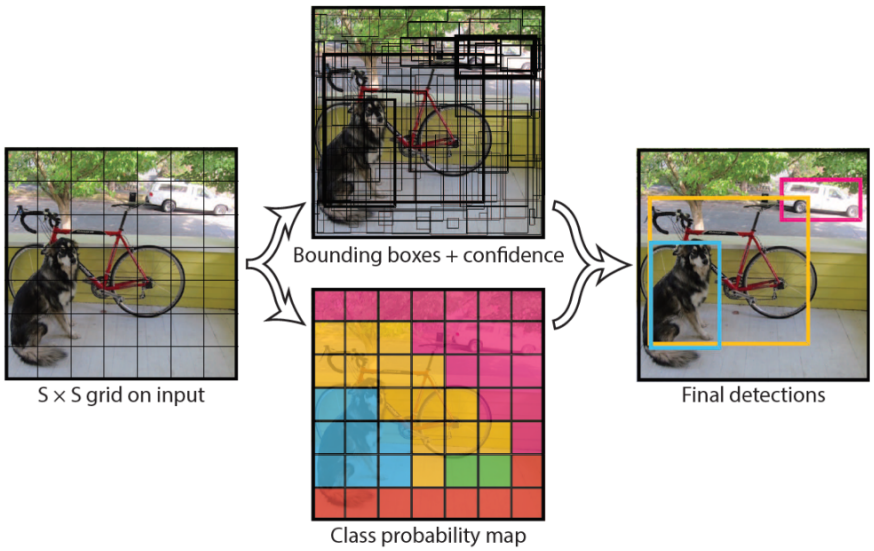


Figure 17. The overview of YOLO approach. [42]

3.7 Region-based Fully Convolutional Network

Jifeng Dai et al. proposed Region-based Fully-Convolutional Network (R-FCN) approach for object detection in [43]. The R-FCN approach is illustrated in Figure 18. The feature maps of the entire image are produced with FCN. Those features are shared between RPN and R-FCN. The RPN proposes regions potentially containing an object. R-FCN produces position-sensitive score maps for every class and background. The more the pixel of position-sensitive map is activated, the closer the object is to that pixel. Then, position-sensitive pooling is applied: for each proposal region the position-sensitive score map is averaging within every particular position. One pooled map is generated for every class and proposal. The more pooled map is activated in general, the higher vote it receive. This position-sensitive approach is shown in Figures 19 and 20. [43]

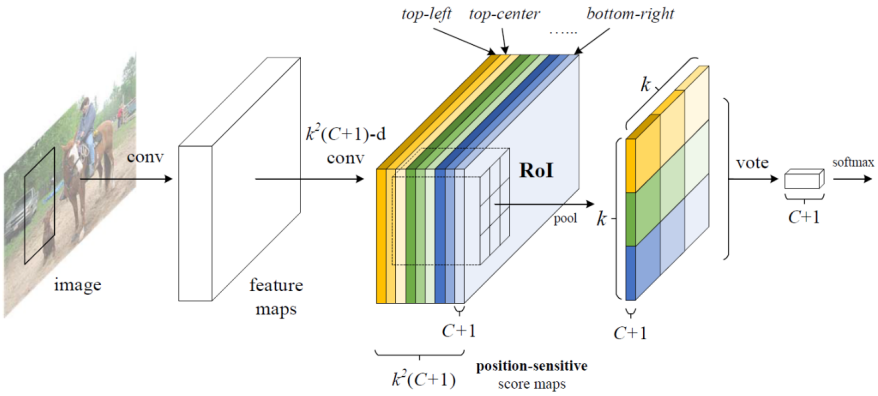


Figure 18. Region-based fully convolutional network. [43]

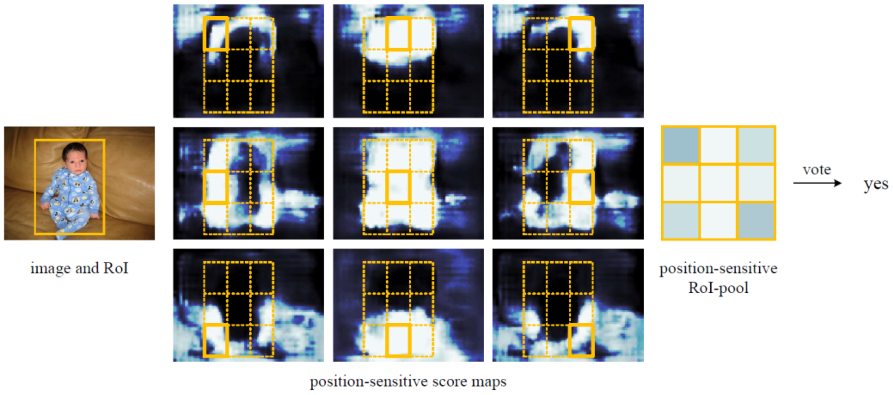


Figure 19. Position-sensitive map and pooled map in case of correct region proposal. [43]

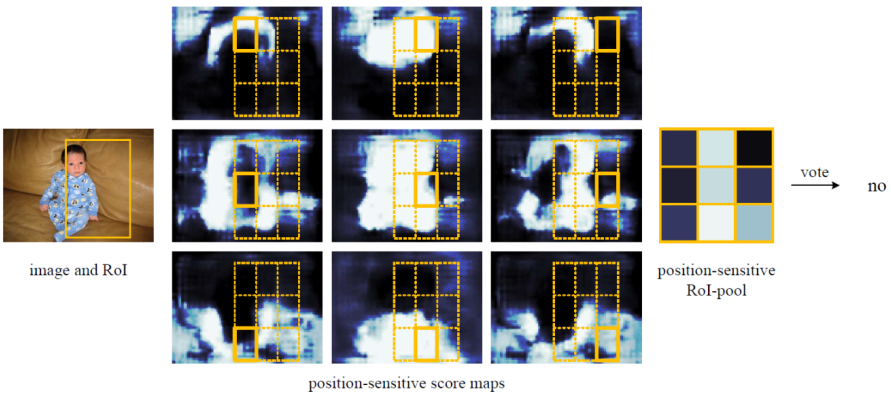


Figure 20. Position-sensitive map and pooled map in case of incorrect region proposal. [43]

4 PROPOSED METHODS

Unfortunately, it is challenging to apply existing robust CNN-based object detection methods for mechanical damages localization on sawn timber images. This is because of the following characteristics in the data:

1. High width to height ratio of board images.
2. Low number of images with defects in the training dataset.
3. High variation in the shape and sizes of bounding boxes.

However, it is worth noting that only one-dimensional localization along the longitudinal direction of the board is relevant since the defected parts are going to be sawn off. In this study the proposed method for mechanical damages detection on sawn timber images consists of the following steps:

1. Patches extraction from the board image.
2. Patches classification with CNN according to the defect type located within the patch.
3. Defect localization using the coordinates of the labeled patches.

The steps of the proposed method are shown in Figure 21.

Sawn timber boards are typically long and narrow and that is why their surface images have high width to height ratio. The example of a sawn timber board image is shown in Figure 22.a. At the same time existing trained CNN architectures require input images to be typically scaled to a fixed size with aspect ratio of 1. Since the feed roller mechanical defects consist of quite narrow notches scaling a board image to required dimensions (approximately 8 times in the horizontal dimension) leads to losing the significant information about defect appearance and make difficulties in learning process as a consequence.

To resolve this problem, the patch extraction method has been used. This method is a combination of ideas of sliding window technique and region proposal from R-CNN. The general idea is to segment the board from the image and split the segmented region into

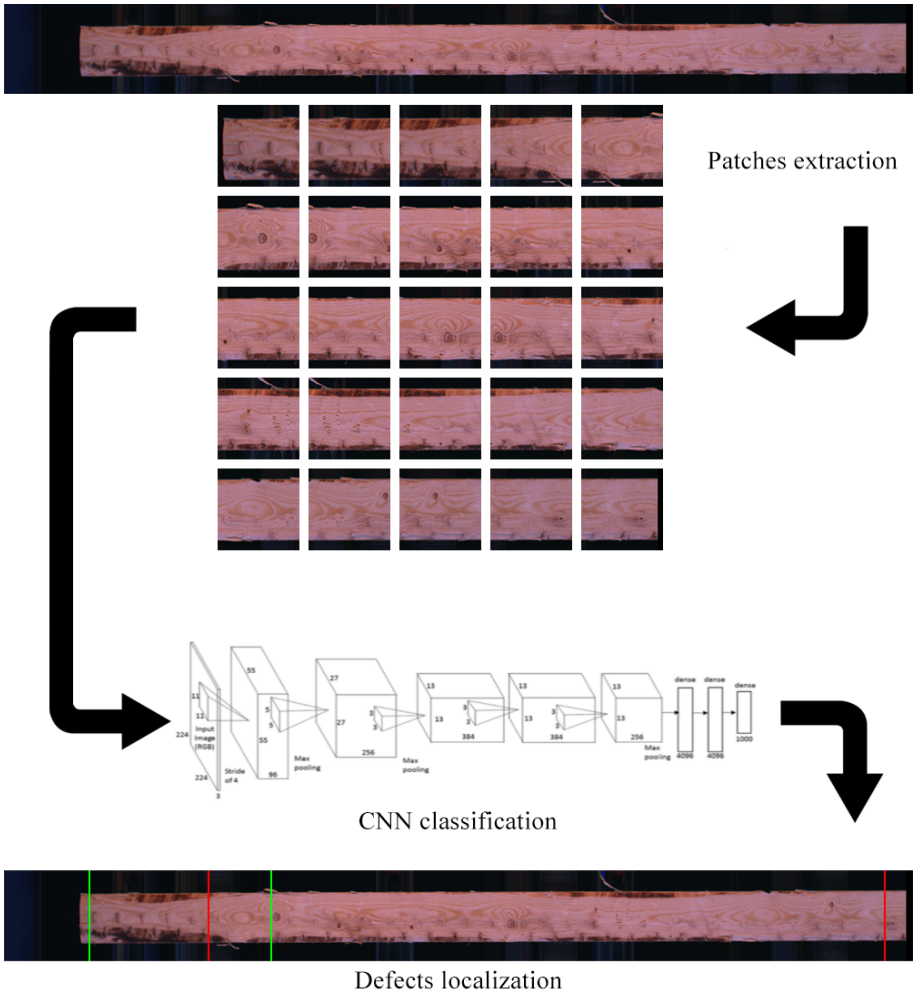


Figure 21. The scheme of the proposed method for mechanical damages detection on sawn timber images: the board image is split into overlapping patches, patches are fed to CNN, and then classified patches and their original locations determine the beginning (green line) and the end (red line) positions of mechanical defects in longitudinal directions.

a series of small overlapping images with required size called patches. The proposed patch extraction technique is the first step of the proposed mechanical damages detection method and it turns the problem of detection into the patch classification task.

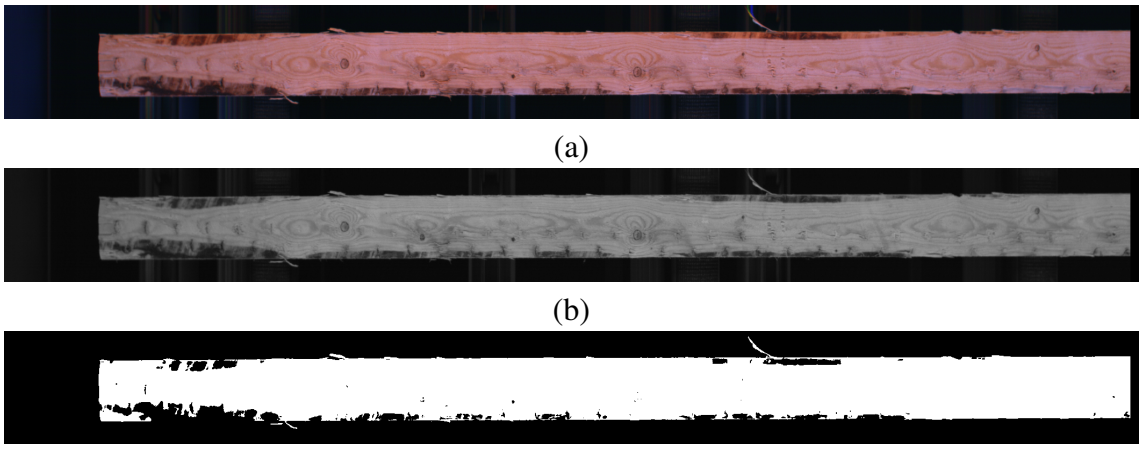
4.1 Patch extraction

As it can be seen in Figure 22.a, the most of the board pixels are quite bright. The board does not cover the whole image and there is also almost black background area. The percentage of that area varies from image to image because of the different board sizes. The position of the board on the image should be estimated in order to reduce the

background influence on CNN learning process. It is especially vital when only a part of the board is on the image and most of the pixels are background. Still some background pixels around the board should be included in patches because of the fact that mechanical damages are often located at the very edges of the board or close enough to them. The proposed algorithm for patch extraction consists of the following stages:

1. The board and the background segmentation.
2. Search of the biggest connected component.
3. Approximation of board angle to the horizontal.
4. Approximation of board edges.
5. Patches extraction from the board.

For the board segmentation the image is converted from RGB to grayscale as shown in Figure 22.b and the Otsu binarization method [21] is applied. The binarized board image as a result of Otsu algorithm is shown in Figure 22.c.



(a)

(b)

(c)

Figure 22. Board and background segmentation: (a) a typical image of a sawn timber board; (b) a grayscaled board image; (c) segmented board (white) and background (black) with the Otsu method.

The Otsu method starts with histogram computation. The histogram of the image represents the number of pixels for every possible intensity value. The Otsu method varies threshold to find at which position the intra-class variance reaches its minimum. It is equal

to find the maximum of inter-class variance. Since the Otsu is a binarization method the are only two classes: the ”black” class which contain pixels below the threshold and the ”white” class with pixels which are equal or above the threshold. The inter-class variance is computed as

$$\sigma^2(t) = N_0(t)N_1(t)[\mu_0(t) - \mu_1(t)]^2, \tag{20}$$

where N_0 and N_1 are the numbers of the pixels in ”black” and ”white” classes, t is the class separating threshold, μ_0 and μ_1 are the mean values of pixels in those classes [21]. The visualized inter-class variance of the board image from Figure 22.b is shown in Figure 23

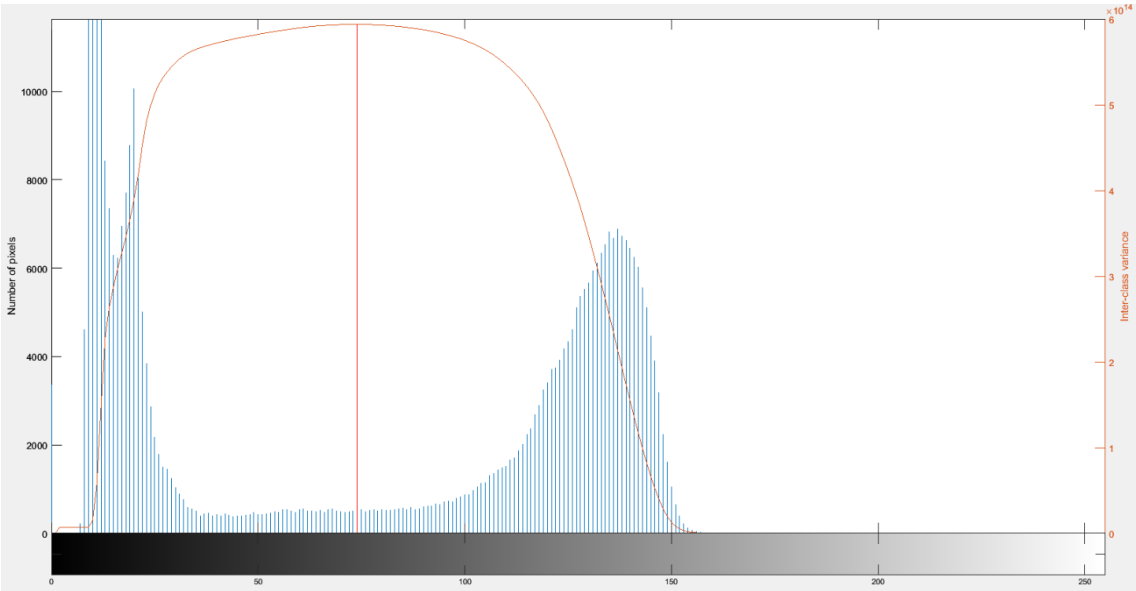


Figure 23. The visualization of the Otsu binarization method: image histogram (blue bins), inter-class variance (brown) and its maximum (red vertical line).

The next stage is to find the connected component with the largest area or the largest number of pixels on the binarized image. The connected component search is performed with the flood-fill algorithm that combines neighboring pixels of the same color [44]. The connected component with the largest area is considered to be a board.

The coordinates of the bounding box for the board is computed using the coordinates of the top-, bottom-, left- and right-most pixels of the connected component. The board on the image could be located imprecisely in horizontal direction and could have a non-zero angle. This angle and possible presence of slivers makes the direct usage of connected component extreme points not reliable for accurate patches extraction as shown

in Figure 24.a. Nevertheless, the left and right sides of the bounding box can be used as reference points.

To overcome the accurate board location issue, more properties of the connected component should be evaluated. The board centroid is calculated as a mean values of the connected component pixels coordinates. The last two properties are the orientation and the minor axis length of the ellipse with the same normalized central moment as the connected component has [45]. To find the ellipse, the origin of the pixels coordinates are shifted to the position of the centroid. It should be noted that to obtain the correct orientation the sign of the vertical coordinates should be inversed. The normalized second central moments are computed with the following formulas:

$$\begin{aligned} M_{xx} &= \sum \frac{x^2}{N} + \frac{1}{12} \\ M_{yy} &= \sum \frac{y^2}{N} + \frac{1}{12} \\ M_{xy} &= \sum \frac{xy}{N} \end{aligned} \tag{21}$$

where x and y are the pixel coordinates with the origin at the centroid, N is a number of pixels in the connected component.

The formula to compute minor axis length of the ellipse is

$$A_{\text{minor}} = 2\sqrt{2}\sqrt{M_{xx} + M_{yy} - \sqrt{(M_{xx} - M_{yy})^2 + 4M_{xy}^2}} \tag{22}$$

Finally, the ellipse orientation in degrees is computed as follows:

$$\theta = \begin{cases} \frac{180}{\pi} \arctan \frac{M_{yy} - M_{xx} + \sqrt{(M_{yy} - M_{xx})^2 + 4M_{xy}^2}}{2M_{xy}} & \text{if } M_{yy} < M_{xx} \\ \frac{180}{\pi} \arctan \frac{2M_{xy}}{M_{xx} - M_{yy} + \sqrt{(M_{xx} - M_{yy})^2 + 4M_{xy}^2}} & \text{otherwise} \end{cases} \tag{23}$$

The height of the board is considered to be the minimum value of the bounding box height and the minor ellipse axis multiplied by $\cos\theta$. If the height of bounding box is taken, then the upper board edge is approximated using the coordinates of one of the top bounding box corners as a reference. The lower board edge is a parallel line that goes under the upper edge at the approximated board height distance. Otherwise the reference is a central line passing through a centroid with the angle to the horizontal θ. The upper and lower edges are at the distance of half board height from the central line. The upper and lower edges for the example board are shown in Figure 24.b.

The patches positions are computed using the approximated top and bottom board edges and left and right bounding box sides. Moreover, the top and bottom parts of all the patches include the background margins with the size of 0.1 of the board height. Also, the background margin of 10 pixels is added to the left and rights ends of the board. The next patch overlaps the previous patch at the half of its size. Finally, patches are scaled to the sizes required by the used CNN architecture, for example the squares with the sides of 224 or 227 pixels. An example of localized patches is shown in Figure 24.c.

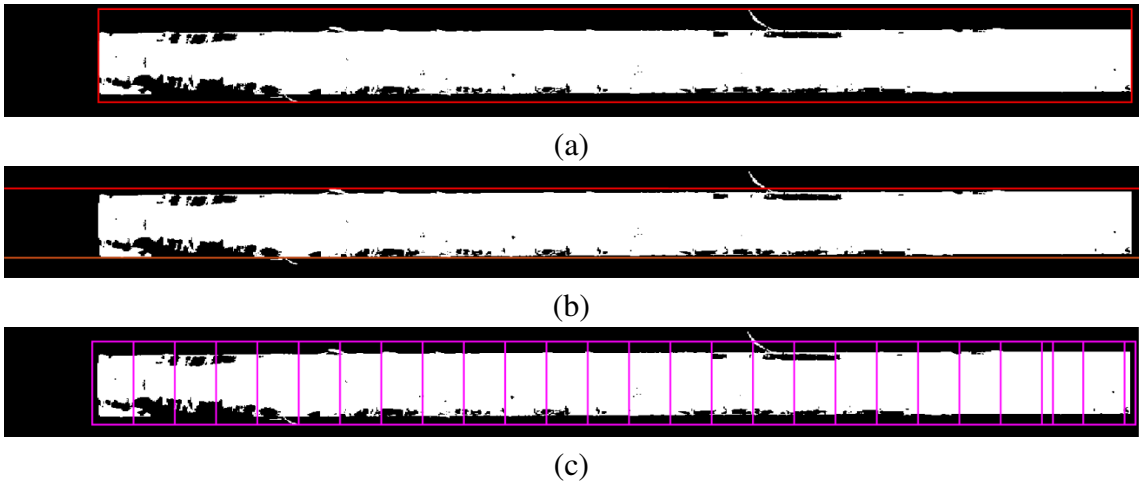


Figure 24. Determine position of patches on the segmented board image:
(a) bounding box of the largest connected component. The sliver on the top of the board influences on the bounding box since too many background pixels are within the bounding box. Although the left and right borders are correct;
(b) top (red) and bottom (brown) board edges approximation;
(c) positions of the overlapping patches.

4.2 Classification with convolutional neural networks

After the patches have been extracted, they are fed to the CNN network. In this study four CNN architectures were selected to compare their ability to recognize mechanical damages on sawn timber patches. These architectures are AlexNet [26], GoogLeNet [32], VGG-16 [33], and ResNet-50 [34].

4.2.1 AlexNet

AlexNet [26] is a CNN which consists of five convolutional layers for features extraction and three fully-connected layers for feature classification. Maximum pooling is applied after the first, second and fifth convolutional layers. Saturating nonlinearity is introduced with the ReLU function (19). The dropout probability is set to 0.5.

AlexNet is designed to be trained on a system with two graphical processing units (GPU). The second, fourth and fifth convolutional layers have independent regions of responsibility for both GPUs. The first GPU performs feature maps processing on the top half of the feature maps, whilst the second GPU on the bottom half. The weights are cross-connected between GPUs at first and third convolutional layers and at all fully-connected layers. The architecture of AlexNet CNN is illustrated in Figure 25.

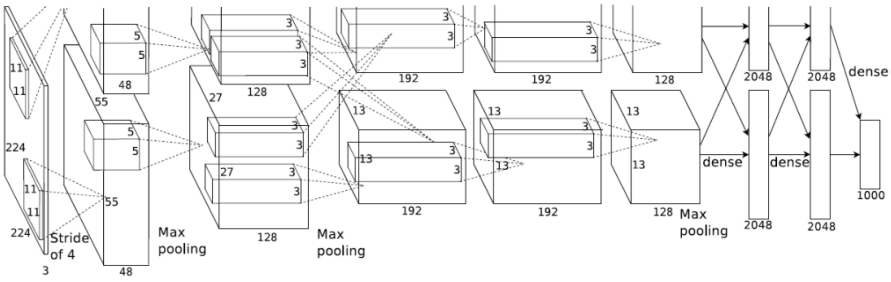


Figure 25. AlexNet CNN architecture [26].

4.2.2 GoogLeNet

GoogLeNet [32] architecture introduces an inception module. When the CNN architecture is being designed, one of the unclear issue is the optimal dimensions of the convolutional kernels. The idea of the inception module is that, instead of selecting one particular set of kernel dimensions for a given convolutional layer, the network should select the most appropriate kernel dimensions or the combination of those kernels by itself. The inception model consists of parallel convolutional layers with 1×1, 3×3 and 5×5 kernels. Additional convolutional layers with 1×1 kernels are used before layers with 3×3 and 5×5 kernels as a cross-channel pooling. The architecture of GoogLeNet CNN is illustrated in Figure 26.

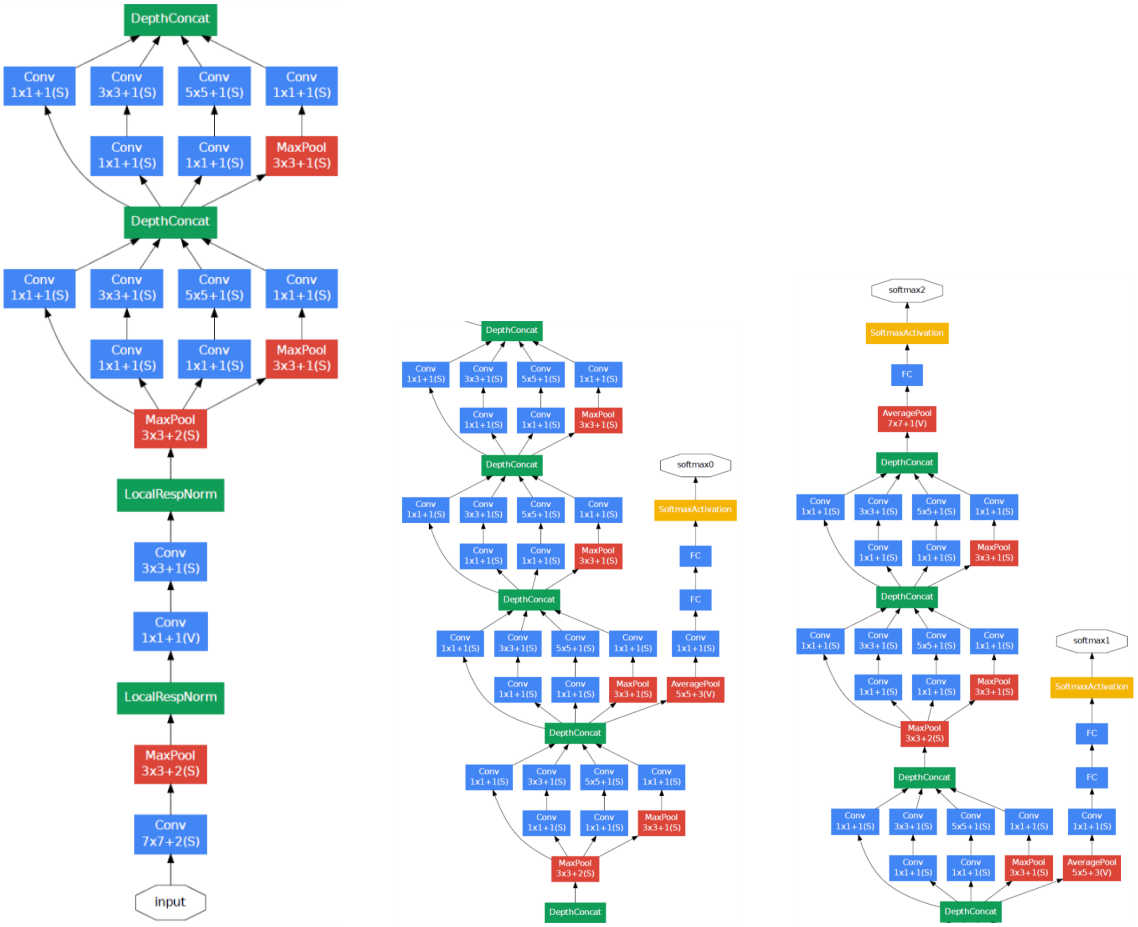


Figure 26. GoogLeNet CNN architecture: convolution and fully-connected layers (blue), maximum pooling layers (red), concatenation layers (green), softmax activation functions before output layers (yellow), input and output layers (white). One layer before and one layer after the breakpoints are the same layers. [32]

4.2.3 VGG

The Visual Geometry Group (VGG) [33] architecture uses convolutional kernels with 3×3 dimensions and maximum pooling kernel with 2×2 dimensions. The developers of the VGG network claim that the stacked combination of three 3×3 kernels have the same effective receptive field as a single 7×7 convolutional kernel. There are two major benefits of using the stacked kernels of smaller sizes. Firstly, advantage is that the number of weights to be trained in stacked version is significantly smaller. Secondly, stacked architecture allows to apply an activation function after every kernel in a stack. Therefore, more complex features can be extracted.

There are several variants of VGG networks which differ by the number of convolutional layers but all of them have three fully-connected layers as a classifier. In this study the VGG-16 variant was chosen containing 13 convolutional layers. The VGG-16 architec-

ture is illustrated in Figure 27.

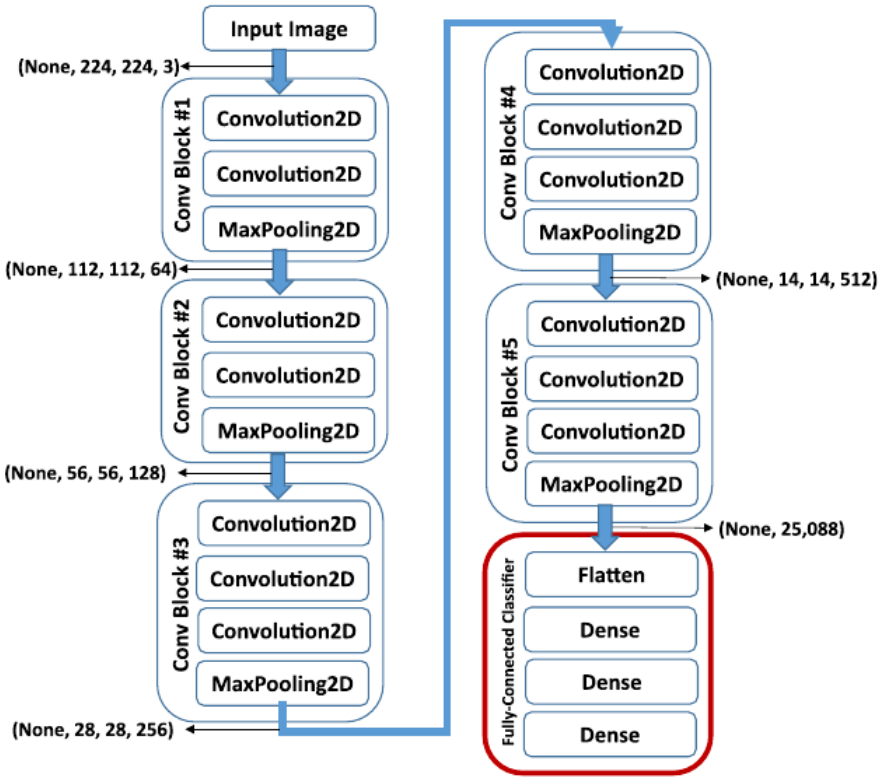


Figure 27. VGG-16 CNN architecture. Only 3×3 convolutional and 2×2 max-pooling kernels are used. [46]

4.2.4 ResNet

With the growth of the number of layers, the vanishing gradient problem becomes more substantial. It means that the backpropagation of the output error does not reach the bottom (closer to input) layers, hence they remain unchanged [34]. ResNet [34] uses direct connections between not neighboring convolutional layers. When the number of channels being increased the direct connections are extended either with zero padding or by applying 1×1 convolution which as a universal method to handle the number of channels. The architecture of ResNet-34 CNN is illustrated in Figure 28.

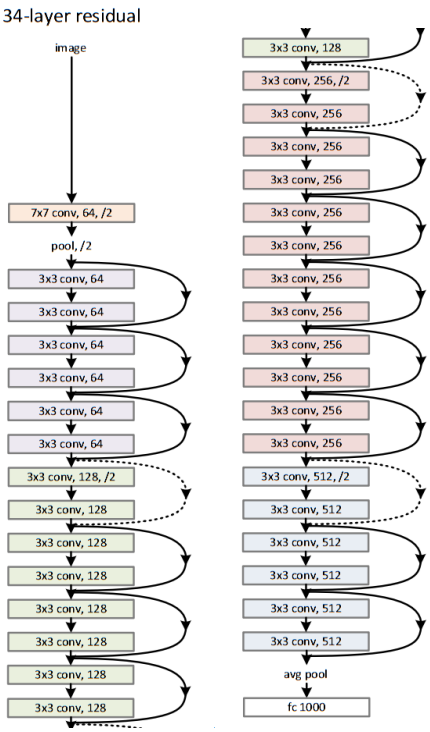


Figure 28. ResNet-34 CNN architecture. One layer before and one layer after the breakpoint are the same layer. Dotted arrows are modified direct connections either with zero padding or by 1×1 convolution. [34]

4.3 Damage localization

During extraction step, patches coordinates on entire image are preserved. The patch is considered to contain the mechanical damage if the damage covers more than the half of the patch in the longitudinal direction of the board. Also, the patches are overlapping at the half of their width. To reduce the localization, the defect left boarder is considered as $x + 0.25w$ of the first patch in a sequence of defective patches while the right boarder is considered as $x + 0.75w$ of the last patch in a sequence of defective patches, where x is the longitudinal coordinate of the top left corner of the patch and w is the width of the patch. The damage localization is visualized in Figures 29 and 30. An example of the output of the proposed method is shown in Figure 31.

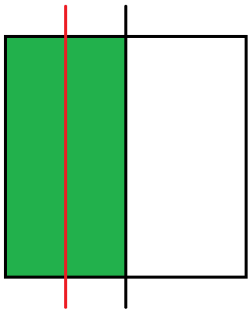


Figure 29. If the patch (black square) is the first in the sequence of defective patches, the ground truth left border of the damage is expected to be within the left half of the patch (green area). Since the exact defect border is unknown the predicted left border of the defect (red line) is assumed to be at the middle of the expected area.

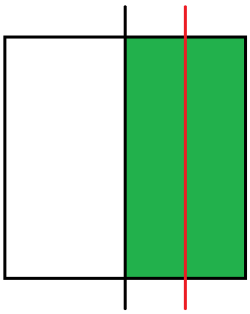


Figure 30. If the patch (black square) is the last in the sequence of defective patches, the ground truth right border of the damage is expected to be within the right half of the patch (green area). Since the exact defect border is unknown the predicted right border of the defect (red line) is assumed to be at the middle of the expected area.

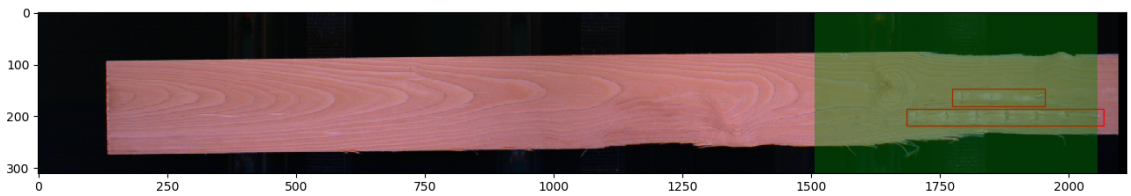


Figure 31. An example of the proposed method output. Predicted defective area is highlighted with green color. The ground truth labels are marked with red bounding boxes.

5 EXPERIMENTS

5.1 Data

The dataset consists of the images of 127 sawn timber boards. Six overlapping images (three from the top and three from the bottom sides) were taken for every board to cover the whole length of the board. The dataset contains locations of the defects of the following types:

- Feed roller periodic mechanical damages.
- Other non-periodic mechanical damages.
- Knots.
- Wanes.
- Dirt.

The defects examples are shown in Figure 32.

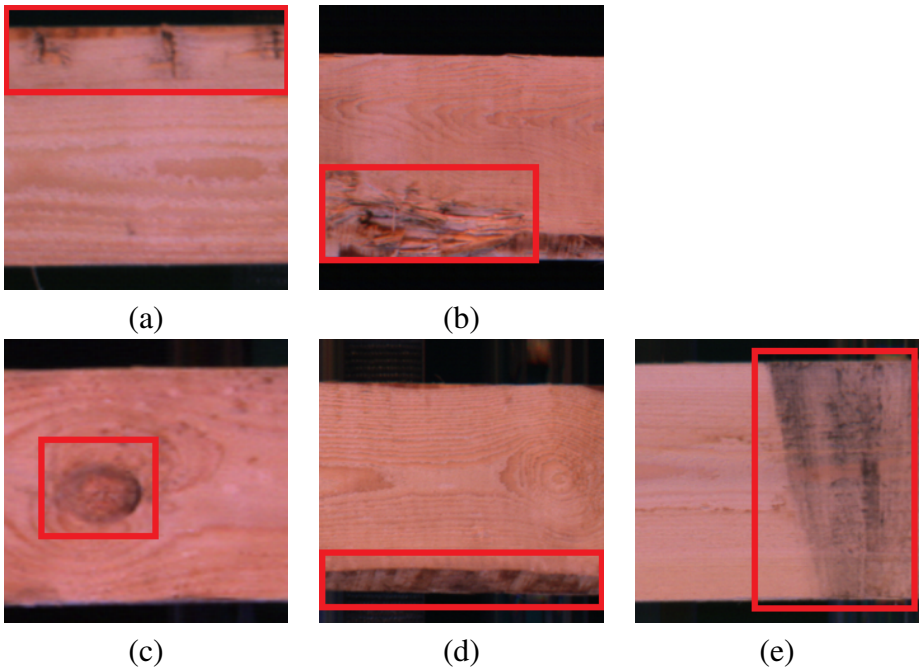


Figure 32. Examples of the annotated defect types: (a) feed roller periodic mechanical damage; (b) other non-periodic mechanical damage; (c) knot; (d) wane; (e) dirt.

The defect locations on the board images where manually labeled with bounding boxes

and defect type corresponding class numbers by a non-expert. The annotation tool for defect labeling with bounding boxes was specially implemented in MATLAB [47]. The interface of the tool is shown in Figure 33.

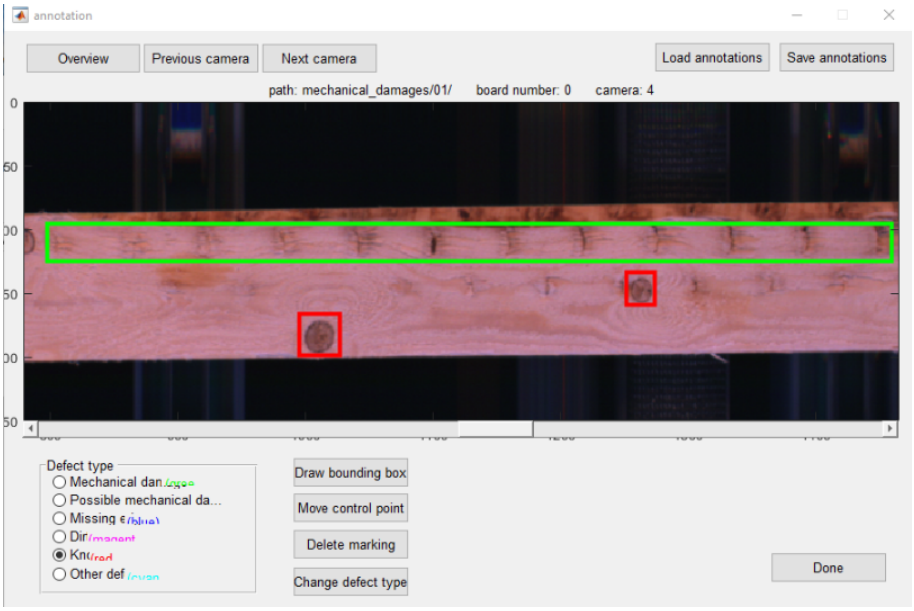


Figure 33. Defects labeling with bounding boxes using the annotation tool for MATLAB.

The patch extraction algorithm was applied to the labeled dataset. As a result, 10808 patches were extracted from the database. All patches crossed by the bounding box of the mechanical damage at least on the half of its width were labeled as containing a defect. As the most of the patches represent defectless parts of the board, there is a large class imbalance in the dataset. For example, in case of feed roller damages, there are 8683 clear patches and only 2125 defective. The number of defective patches was equalized by generating additional patches with horizontal and vertical mirroring and 180° rotation. As a result, the number of defective patches increased in 4 times and reached 8500. The patches were board-wise randomly split on train and test sets in proportion 70 and 30%. That means that all the patches from the same board were used either in training or in test phase. Images of 89 boards were used for training of the models, while the performance was evaluated on the rest 38 boards. Additional patches were used only in training stage. Thus, the training set for feed roller mechanical damages classification consists of 12209 patches (6045 clear and 6164 defective) and the testing set consists of 3222 patches (2638 clear and 584 defective).

5.2 Evaluation criteria

To evaluate the CNN architecture performance the following metrics were used:

- Accuracy

  Accuracy is the basic metric to evaluate the correctness of the classification. It defined as

  $$ACC = \frac{\sum_i^C T_i}{N}, \tag{24}$$

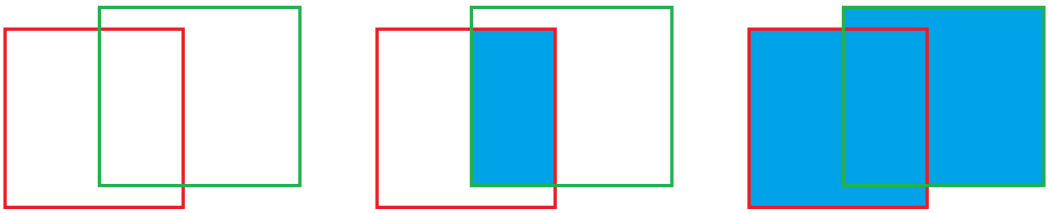
  where C is the number of classes, T_i is the number of correctly recognized samples for i-th class, N is the total number of samples.

- Jaccard metric

  Also called intersection over union (IoU), Jaccard metric is used to measure how close the predicted object location is to the ground-truth location. It is defined as

  $$S_{\text{Jaccard}} = \frac{|A_P \cap A_{GT}|}{|A_P \cup A_{GT}|}, \tag{25}$$

  where A_P is the area of the predicted object bounding box and A_{GT} is the area of the ground truth bounding box. An example of the IoU for a random ground truth and predicted bounding boxes is shown in Figure 34. In case of one-dimensional localization, the length of the object is used instead of the area. An example of the intersection and the union of the ground truth and predicted mechanically damaged one-dimensional regions of the board is shown in Figure 35.



(a) (b) (c)

Figure 34. Jaccard metric: (a) ground truth (red) and predicted (green) bounding boxes; (b) their intersection (blue area); (c) their union (blue area).

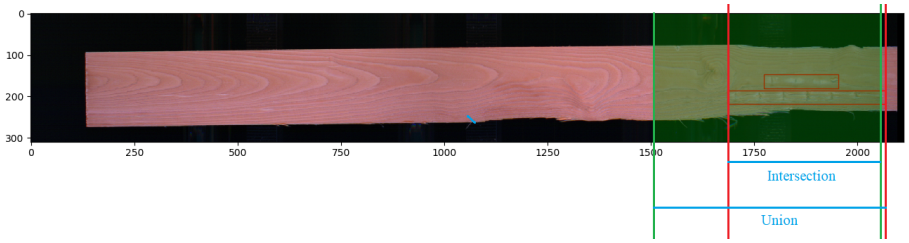


Figure 35. Jaccard metric in case of one-dimensional mechanical damage localization on the board image. The ground truth defect borders are red. The predicted defect borders are green.

- Confusion matrix

  Element C_{ij} of a confusion matrix represents how many samples of i-th class being classified as j-th class. In case of ideal classifier, the confusion matrix equals to the identity matrix of the same size.

- Inference time

  Inference time is a time needed to classify a new patch. It characterizes the computational costs of using the particular CNN architecture.

5.3 Description of experiments

While annotating the dataset, it was noticed that except the feed roller periodic damages there are also non-periodic mechanical damages. In order to reveal more details of the CNN architectures defect detection abilities, the following experiments were carried out:

1. Classification of all mechanical damages treated as a single class.
2. Classification of feed roller periodic traces only.

The purpose of first experiment is to estimate the ability of the selected CNN architecture to distinguish the mechanically damaged parts of the board regardless of the damage type. Non-periodic damages may significantly affect the accuracy because they are more variable in appearance and less common in the existing dataset. That is why the second experiment was carried out to estimate the CNN performance to classify periodic damages since they are more regular. Finally, the last experiment measures the ability of the network not only to detect the damage, but also to distinguish its type.

All datasets including annotations were generated with MATLAB [47]. The CNN architectures were trained with the Caffe [48] deep learning framework. The performance was evaluated with the scripts for Python 3.6 programming language with Caffe module. To reduce training time and increase classification accuracy, the transfer learning approach was applied. The CNNs were initialized with the pretrained models from another Master's thesis [49] within the DigiSaw project, because of the similarity of the used images.

5.4 Results

5.4.1 Experiment 1: classification of the combined mechanical damages

The performance of localization and binary classification of combined periodic and non-periodic mechanical damages to a single class is provided in Table 1. The confusion matrices for each of the trained CNN architectures are shown in Figure 36.

Table 1. Performance comparison of different CNN architectures in case of combined binary classification.

Architecture	Accuracy	S_{Jaccard}
AlexNet	0.800	0.361
GooLeNet	0.860	0.474
VGG-16	0.861	0.500
ResNet-50	0.799	0.433

5.4.2 Experiment 2: classification of the feed roller periodic traces

The performance of localization and binary classification of feed roller periodic mechanical damages only is provided in Table 2. The confusion matrices for each of the trained CNN architectures are shown in Figure 37. The percentage of images containing feed roller mechanical defects and the defects were detected correctly with respect to the Jaccard metric threshold for every CNN architecture is shown in Figure 38. Examples of feed rollers periodic damages detection results with GoogLeNet CNN architecture are given in Figure 39. Comparison of the results obtained for the same image with every trained CNN is shown in Figures 40 and41.

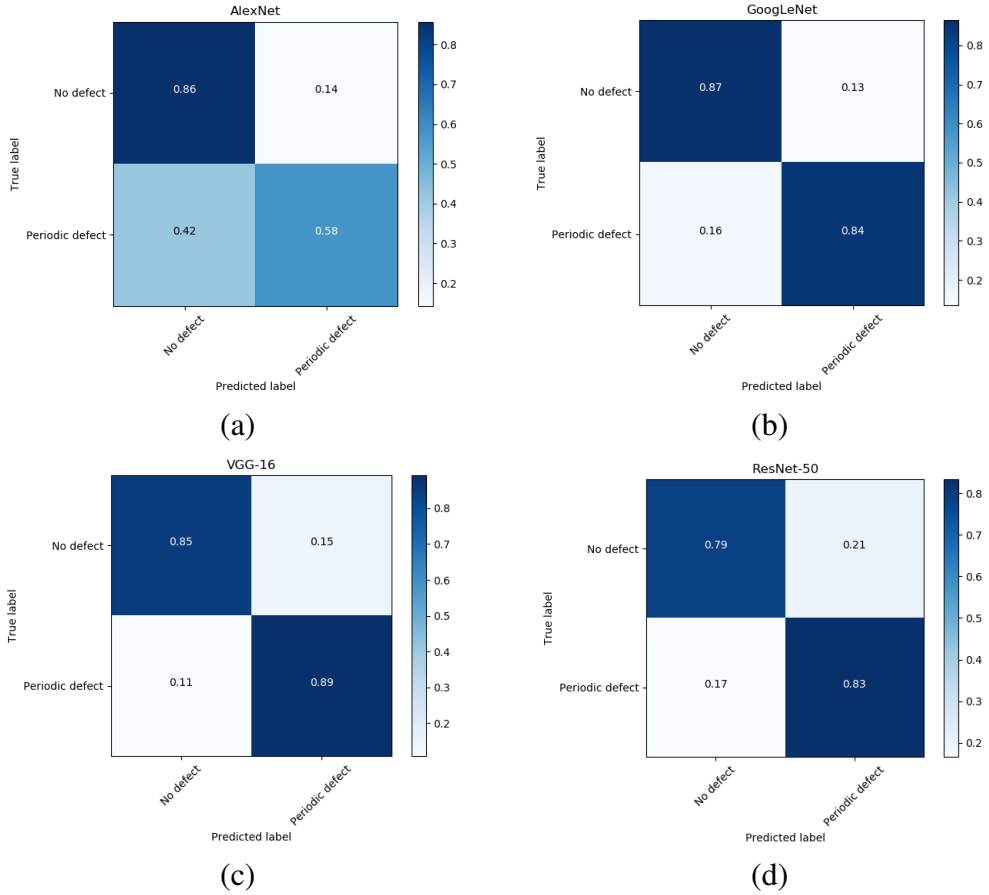


Figure 36. Confusion matrices of CNN architectures trained for combined binary classification: (a) AlexNet, (b) GoogLeNet, (c) VGG-16, (d) ResNet-50.

5.4.3 Inference time

The most time consuming stage of the proposed method is the sequential patches classification with the CNN. The lower inference time consumed by a single patch classification, the more effective the given CNN architecture. Inference time for all CNNs was measured on MSI GE70 laptop with Intel Core i7-4700MQ @ 2.4 GHz processor, NVIDIA GeForce GTX 760M single GPU, and Ubuntu 17.10 operating system.

Table 3 contains average time required to classify one patch of the board image for selected CNNs.

Table 2. Performance comparison of different CNN architectures in case of feed rollers periodic mechanical damages binary classification.

Architecture	Accuracy	S_{Jaccard}
AlexNet	0.857	0.384
GooLeNet	0.907	0.580
VGG-16	0.927	0.588
ResNet-50	0.913	0.546

Table 3. Single patch average inference time for each of the trained CNN architectures.

Architecture	Inference time, seconds
AlexNet	0.022
GoogLeNet	0.023
VGG-16	0.101
ResNet-50	0.051

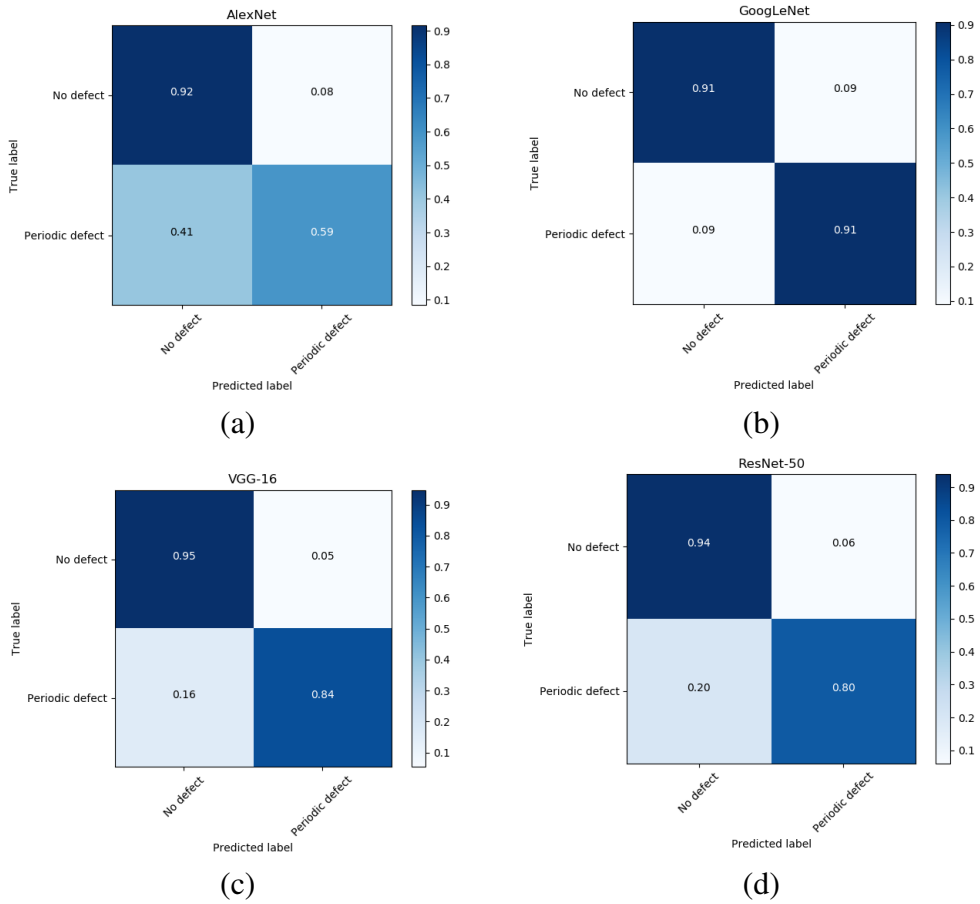


Figure 37. Confusion matrices of CNN architectures trained for feed rollers periodic mechanical damages binary classification: (a) AlexNet, (b) GoogLeNet, (c) VGG-16, (d) ResNet-50.

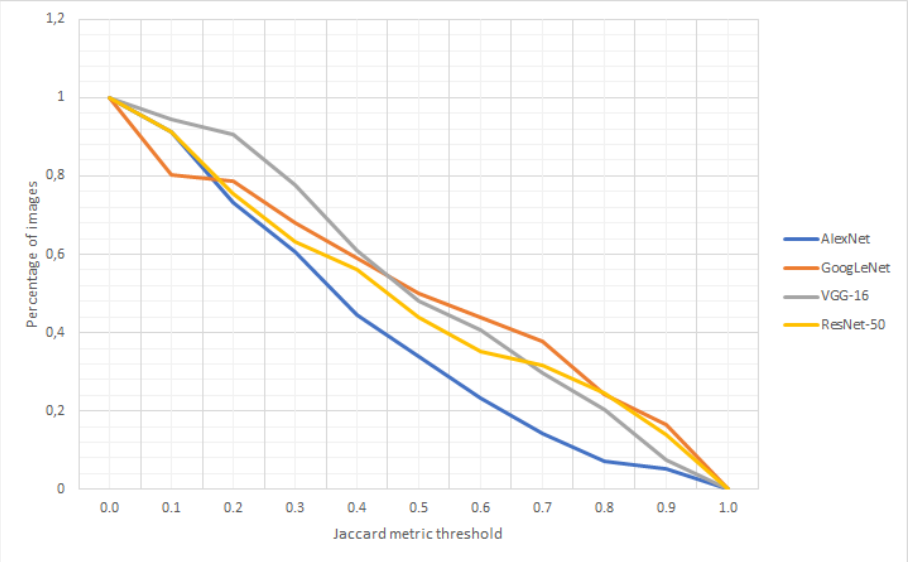


Figure 38. Percentage of images with correctly detected feed roller mechanical defects with respect to the Jaccard metric threshold.

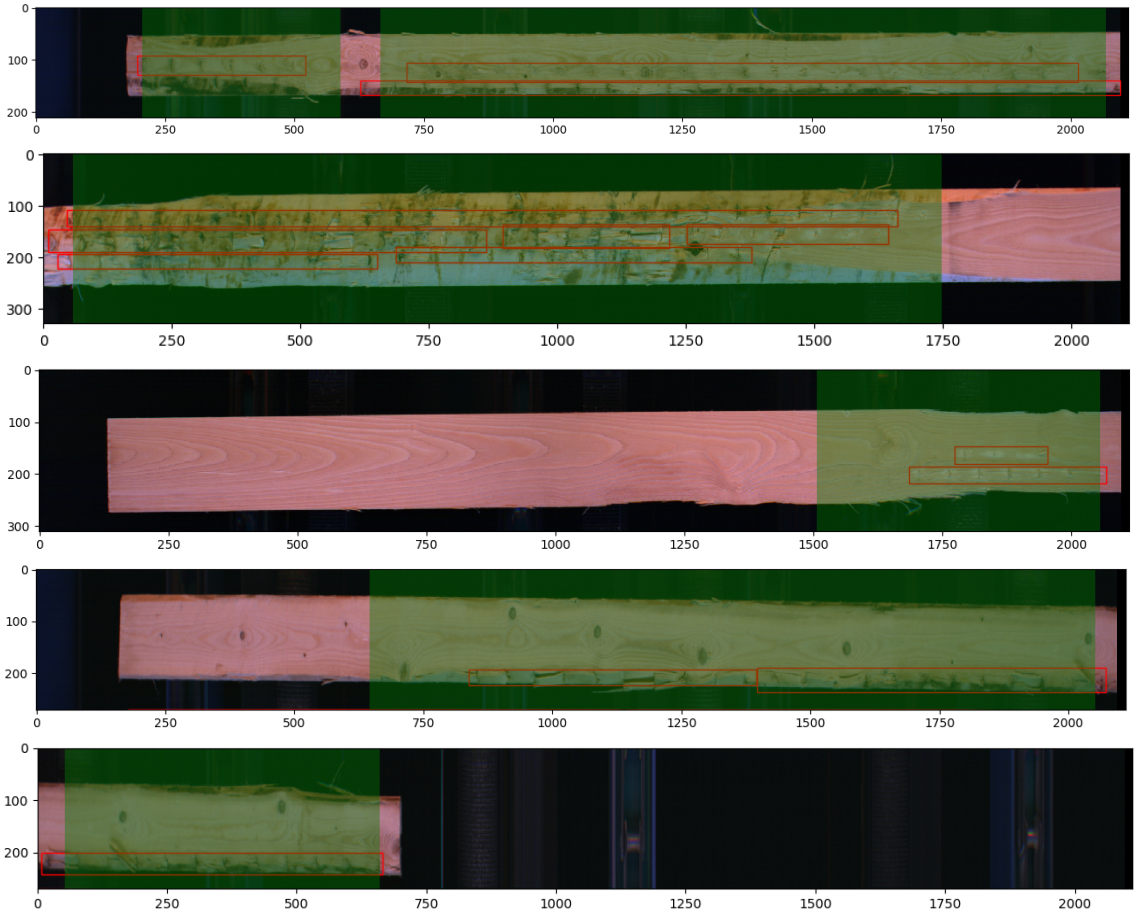


Figure 39. Several examples of feed roller periodic mechanical damages detection with GoogLeNet CNN architecture. Ground truth damages are marked with red bounding boxes, while the predicted defective parts of boards are highlighted with green color.

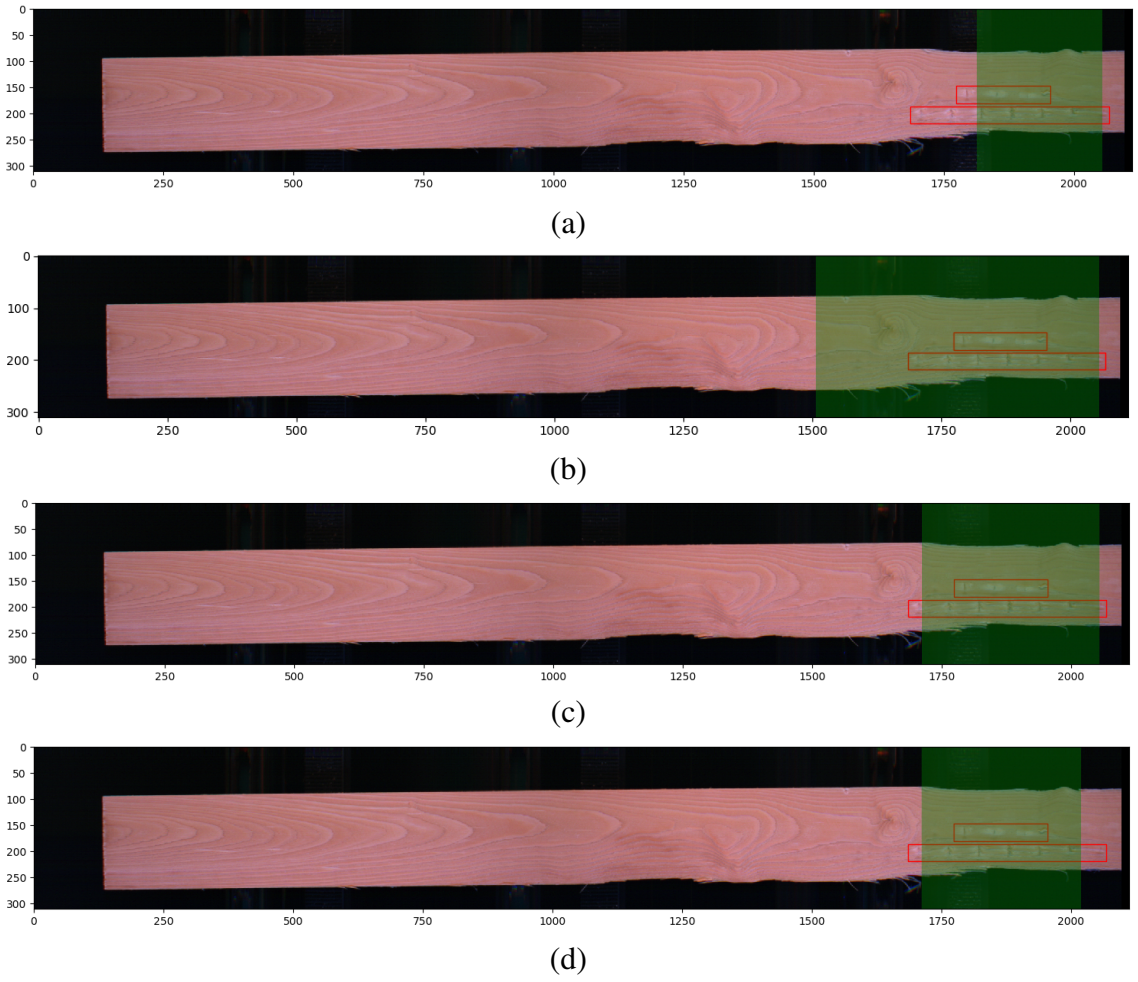


Figure 40. Example of detection of the same slight feed roller periodic mechanical damages with: (a) AlexNet; (b) GoogLeNet; (c) VGG-16; (d) ResNet-50. Ground truth damages are marked with red bounding boxes, while the predicted defective parts of boards are highlighted with green color.

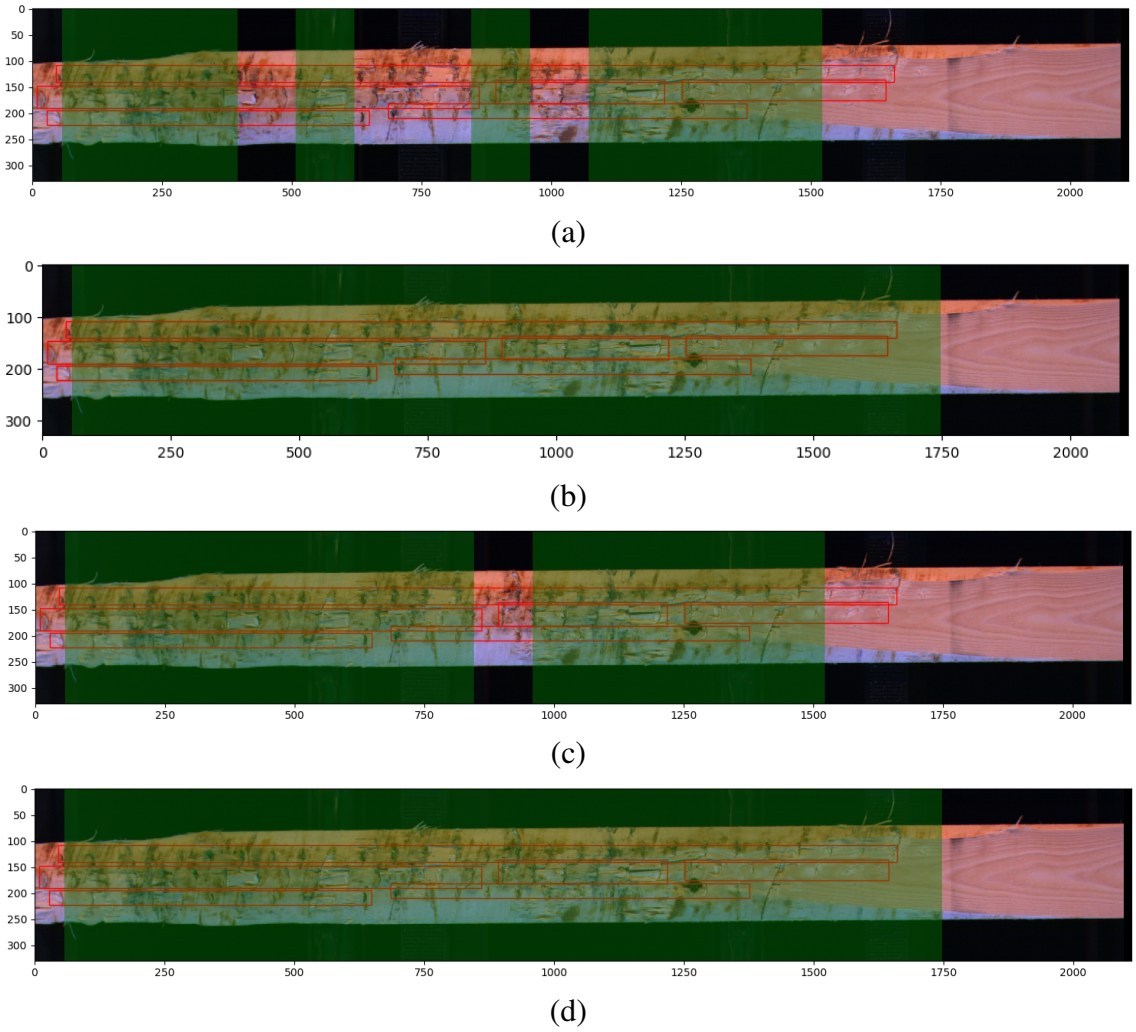


Figure 41. Examples of detection of the same hard feed roller periodic mechanical damages with: (a) AlexNet; (b) GoogLeNet; (c) VGG-16; (d) ResNet-50. Ground truth damages are marked with red bounding boxes, while the predicted defective parts of boards are highlighted with green color.

6 DISCUSSION

6.1 Current study

Mechanical damages on sawn timber can appear during the sawmilling process. Mechanical damages can be caused by the feed rollers of the sawing machine or by other excessive interactions with the board surfaces. Those damages significantly affect the quality and the price of the particular board, therefore it is crucial to detect them. Based on the literature review, it can be said that the automatic detection of mechanical damages has not been studied. The majority of the existing solutions in sawn timber surface inspection solves the problem of classification and detection natural damages such as knots, worm holes, watermarks and fungus. However, the mechanical damages are no less important.

For the purpose of the mechanical damages detection, the dataset of images of 127 sawn timber boards was annotated with bounding boxes and labels specific for different defect types including natural. The proposed method segments the board from the background, splits the segmented board into the overlapping patches, classify them and, finally, localize the mechanical defects according to the classification results. This database of the board patches was used to train four CNN architectures: AlexNet, GoogLeNet, VGG-16 and ResNet-50 to distinguish the defective and normal regions of the boards. Because of the limited number of the board images, and the significant variability of the defects width to height ratio, the 1-dimension sliding window detection technique was used instead of straightforward detection on a single image with a state-of-the-art CNN-architectures such as Faster R-CNN, YOLO or R-FCN.

The proposed solution achieved very promising individual patch classification accuracy of more than 92% for VGG-16 architecture. At the same time the GoogLeNet architecture achieved the lowest false negative and false positive rates both of 9% and the GoogLeNet architecture is four time faster in single patch processing than the VGG-16. Both GoogLeNet and VGG-16 architectures have shown the best detection accuracy with the Jaccard coefficient greater than 0.58. The worst classification and detection accuracy (85.7% and 0.384) was shown by AlexNet architecture. Also, AlexNet showed the worst false positive rate of 41%. It could be explained by the low number of convolutional layers in AlexNet. The possible reason for the similar results of the VGG-16 and GoogLeNet architectures is the fact that the VGG-16 uses only 3×3 kernels, while the GoogLeNet uses the combination of 1×1, 3×3 and 5×5 kernels in every inception layer. It means that the GoogLeNet can degrade to the architecture similar to the VGG-16 during training

stage. The main drawback of the proposed method is a large computational time needed to process all the images of a single board, since those images are split into patches and are fed sequentially to the CNN input.

6.2 Future work

The main possible objectives for the future work include accuracy improvement and required time reduction. Accuracy can be improved by using a larger image dataset labeled by the experts of the sawmill industry. To reduce the computational costs, the single image end-to-end CNN-based architecture could be used. However, the large variability of the defect sizes and large width to height ratio of board images should be taken into account. Finally, the combined automated surface inspection system that would be able to detect and distinguish any kind of defects both mechanical and natural can be implemented.

7 CONCLUSION

This study introduced the problem of the mechanical damage detection on the surface of sawn timber boards. The existing methods applied for timber inspection were reviewed. The principals of convolutional neural networks and their adaptations for object localization were surveyed since the CNN-based approaches are the most promising in the image classification tasks.

In this thesis the method for mechanical damages detection on sawn timber images was introduced. The proposed method segments the board on the image, splits the part of the image containing the board into overlapping patches, classifies the patches with the CNN, and, finally, determines the defect location based on classification results and the coordinates of patches. The experimental part of the work contained the performance comparison of four CNN architectures: AlexNet, GoogLeNet, VGG-16, and ResNet-50. The VGG-16 architecture produced the best results with a very promising classification accuracy of more than 92% for individual patches.

REFERENCES

[1] Yrjö Nuutinen, Kari Väätäinen, Antti Asikainen, Robert Prinz, and Jaakko Heinonen. Operational efficiency and damage to sawlogs by feed rollers of the harvester head. *Silva Fennica*, 44(1):121–139, 2010.

[2] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[3] Digisaw — leap of digitalisation for sawmill industry. `http://www2.it.lut.fi/project/digisaw/index.shtml`. [Online; accessed May 25, 2018].

[4] UR Hashim, SZ Hashim, and AK Muda. Automated vision inspection of timber surface defect: a review. *Jurnal Teknologi*, 77(20):127–135, 2015.

[5] Hau Lee Tong, Hu Ng, Tzen Vun Timothy Yap, Wan Siti Halimatul Munirah Wan Ahmad, and Mohammad Faizal Ahmad Fauzi. Evaluation of feature extraction and selection techniques for the classification of wood defect images. *Journal of Engineering and Applied Sciences*, 12(3):602–608, 2017.

[6] SK Sharma and SR Shukla. Properties evaluation and defects detection in timbers by ultrasonic non-destructive technique. *Journal of the Indian Academy of Wood Science*, 9(1):66–71, 2012.

[7] Shan Gao, Na Wang, Lihai Wang, and Jingquan Han. Application of an ultrasonic wave propagation field in the quantitative identification of cavity defect of log disc. *Computers and Electronics in Agriculture*, 108:123–129, 2014.

[8] Fleur Longuetaud, Frédéric Mothe, Bertrand Kerautret, Adrien Krähenbühl, Laurent Hory, Jean Michel Leban, and Isabelle Debled-Rennesson. Automatic knot detection and measurements from X-ray CT images of wood: a review and validation of an improved algorithm on softwood samples. *Computers and Electronics in Agriculture*, 85:77–89, 2012.

[9] Erik Johansson, Dennis Johansson, Johan Skog, and Magnus Fredriksson. Automated knot detection for high speed computed tomography on pinus sylvestris l. and picea abies (l.) karst. using ellipse fitting in concentric surfaces. *Computers and Electronics in Agriculture*, 96:238–245, 2013.

[10] A Wyckhuyse and Xavier Maldague. A study of wood inspection by infrared thermography, part i: Wood pole inspection by infrared thermography. *Journal of Research in Nondestructive Evaluation*, 13(1):1–12, 2001.

[11] Gamaliel López, L-Alfonso Basterra, Gemma Ramón-Cueto, and Agustín de Diego. Detection of singularities and subsurface defects in wood by infrared thermography. *International Journal of Architectural Heritage*, 8(4):517–536, 2014.

[12] Olle Hagman. Multivariate prediction of wood surface features using an imaging spectrograph. *Holz als Roh-und Werkstoff*, 55(6):377–382, 1997.

[13] Ruoxu Ren, Terence Hung, and Kay Chen Tan. A generic deep-learning-based approach for automated surface inspection. *IEEE Transactions on Cybernetics*, 48(3):929–940, 2018.

[14] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[15] Robert M Haralick, Karthikeyan Shanmugam, and Its'hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973.

[16] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157. IEEE, 1999.

[17] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features SURF. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[18] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6):460–473, 1978.

[19] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826. IEEE, 2016.

[20] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: a deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, volume 32, pages 647–655. PMLR, 2014.

[21] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[22] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[23] Young-Jin Cha, Wooram Choi, and Oral Büyüköztürk. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5):361–378, 2017.

[24] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, pages 319–345. Springer, 1999.

[25] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th Conference on Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[27] Saleh Albelwi and Ausif Mahmood. A framework for designing the architectures of deep convolutional neural networks. *Entropy*, 19(6):242, 2017.

[28] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[29] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[30] Moojoon Shin and Jee-Hyong Lee. Accurate lithography hotspot detection using deep convolutional neural networks. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, 15(4):043507, 2016.

[31] Yue Liu, Jun Yong, Liang Liu, Jinlong Zhao, and Zongyu Li. The method of insulator recognition based on deep learning. In *4th International Conference on Applied Robotics for the Power Industry (CARPI)*, pages 1–5. IEEE, 2016.

[32] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9. IEEE, 2015.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

[35] Bin Wang, Sheng Tang, Ruizhen Zhao, Wu Liu, and Yigang Cen. Pedestrian detection based on region proposal fusion. In *17th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2015.

[36] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587. IEEE, 2014.

[37] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision (IJCV)*, 104(2):154–171, 2013.

[38] Ross Girshick. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1440–1448. IEEE, 2015.

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th Conference on Neural Information Processing Systems (NIPS)*, pages 91–99, 2015.

[40] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2980–2988. IEEE, 2017.

[41] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440. IEEE, 2015.

[42] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788. IEEE, 2016.

[43] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: Object detection via region-based fully convolutional networks. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS)*, pages 379–387, 2016.

[44] Shane Torbert. *Applied Computer Science*. Springer, 2016.

[45] Robert M Haralick and Linda G Shapiro. *Computer and Robot Vision*. Addison-Wesley, 1992.

[46] Kasthurirangan Gopalakrishnan, Siddhartha K Khaitan, Alok Choudhary, and Ankit Agrawal. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, 157:322–330, 2017.

[47] MATLAB. *version 9.4.0 (R2018a)*. The MathWorks Inc., Natick, Massachusetts, 2018.

[48] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd International Conference on Multimedia*, pages 675–678. ACM, 2014.

[49] Dmitrii Shustrov. Species identification of wooden material using convolutional neural networks. Master's thesis, Lappeenranta University of Technology, Finland, 2018.