ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 9, September 2023

An Artificial Intelligence-Based Approach to Detect the Quality of Wooden Panels using Convolutional Neural Networks

[1] Tom Tuunainen, [2] Olli Isohanni, [3] Mitha Rachel Jose

[1] [2] [3] Centria University of Applied Sciences, Kokkola, Finland

Corresponding Author Email: [1] tom.tuunainen@centria.fi, [2]olli.isohanni@centria.fi, [3]mitha.jose@centria.fi

Abstract— Convolutional Neural Network (CNN) is a network architecture for deep learning that learns directly from data [10]. CNN are used to obtain patterns from images to recognize objects, classes, and categories. These are used to classify audio, time series, and signal data. MobileNet is a CNN architecture that is developed to build light-weight deep neural networks. Image classification categorizes the input images into pre-defined labels or categories. This study is performed to detect the quality of Medium Density Fiberboard (MDF) wooden panels. This model learns from the image dataset of wooden panels and produces accurate results. The panels are suitable for many kinds of interior styles. The panels can be made of MDF or woods such as high-quality spruce or pine. This experiment focuses on the quality of the MDF panels and the wooden panels of various sizes, which might differ in length, depth and thickness. The analysis aims to detect the quality of wooden panels using a CNN architecture called MobileNet. The CNN is trained with three categories; PASS, FAIL, and empty, with each category containing 1000 samples. The samples containing high-quality panels were categorized as PASS, and broken or defective panels were categorized as FAIL. If there are no panels, then the CNN defaults to the empty category.

Index Terms — Artificial Intelligence, Convolutional Neural Network, Deep Learning, Image Classification, MobileNet.

I. BACKGROUND

Manufacturers of wood-based materials attach great importance to the quality of their products in order to meet the high market demand, which is a real challenge for a human or a camera [6]. This issue can be solved using much research in artificial intelligence and computer vision to characterize an MDF or wood panel based on its quality, dimensions, labeling, or other factors. Process automation in the woodworking industry has been carried out to increase quality and productivity in the company. Therefore, this study proposes an approach to find defects in wood panels. Wood defects such as color discoloration, scratches, cracks, broken or not broken panels, or delamination were detected using a computer vision approach. The idea for this wood defect detection came from a blog published in 2019 by PyImageSearch [12]. The blog shows a video of a person doing weight lifting, and while this video is passed to a video classification using Keras and a deep learning algorithm, the activities in the video are detected as ”activity: weight lifting”. The same approach is applied to wooden boards to detect defects.

II. INTRODUCTION

Traditionally manufacturers of wood-based products used manpower to identify the surface defects of wood panels and it had many disadvantages such as low work efficiency, high labor, and high cost. Therefore, to replace humans for identifying and detecting the quality of wood, it was introduced with an automation process, thereby improving efficiency, increasing profit and reducing cost. The quality control of wooden panels needs to be refined by identifying the defective wood products in the production line. New technologies like artificial intelligence can be used to improve quality assurance to support human activity. The image recognition neural network method can recognize the image from a trained dataset [9]. A dataset helps to make predictions with some accuracy. It takes all the bad predictions and adjust the weight inside the model to create a model with fewer mistakes. A deep neural network is beneficial when replacing human labor with autonomous work without compromising efficiency. The usage of a deep neural network has various applications in real life. A deep neural network can recognize voice, sound, and graphics, do an expert review, and perform many actions that require prediction, creative thinking, and analytics. Deep neural networks represent a type of mission learning and the system uses many layers of nodes to derive high-level functions from input information. This means transforming the data into an abstract component. It learns how to respond to new situations when it gets new information from the system. The various deep neural network methods applied in this study are shown in figure 1.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 9, September 2023

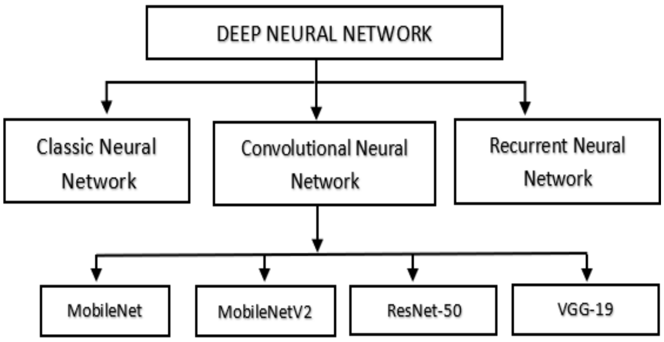

Figure 1. Deep Neural network.

A. Deep Neural Network

The term deep learning refers to artificial neural networks with multiple layers. One of the most common deep neural networks is the CNN. The primary purpose of a deep neural network is to receive a set of inputs, perform complex calculations on them, and produce output to solve real world problems like classification and predictions [1]. Deep Learning Model is shown in figure 2.

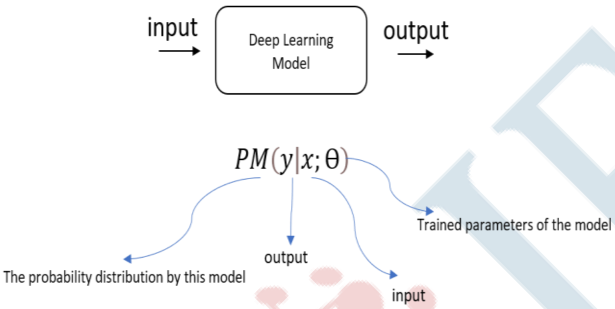

Figure 2. Deep Learning Model.

The objective of a deep learning model is to transform some input X to an output Y. This will be a probabilistic definition of deep learning models in which the output Y can conform to a probability distribution (conditionally input X and model parameters Θ), concerned with the architecture of the deep learning model, that takes from the input to the output.

B. Convolutional Neural Network

Convolutional Neural Networks (CNN) are applied in image processing problems as it is a subset of neural network. It takes its name from linear mathematical operations between matrices called Convolution. The convolutional network includes a convolutional layer, nonlinearity layer, pooling layer, and fully connected layer. CNN yields high preferences for object recognition and classification. The depth of CNN's originates from the point that they can extract information based on pixel intensities and learn features automatically. The feature extraction and classification algorithms are based on Deep Learning algorithms. The figure 3 represents the three-dimensional representation of CNN.

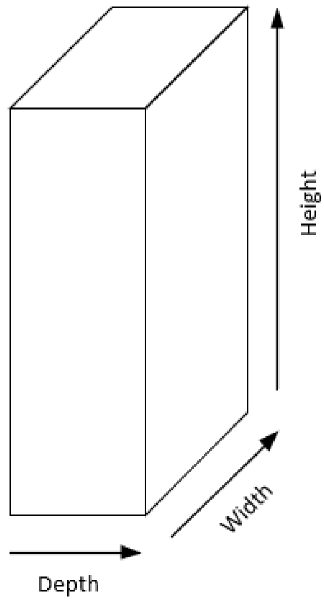

Figure 3. A three-dimensional Input representation of a Convolutional Neural Network.

Multiple layers, each corresponding to a different filter, but looking at the same region is shown in figure 4.

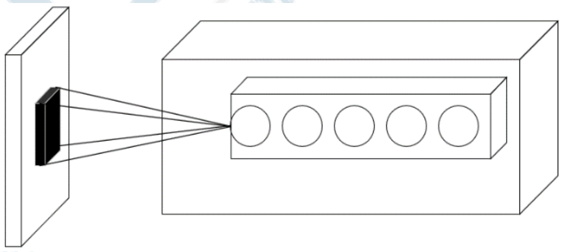

Figure 4. Filtering of multiple layers of CNN.

The figure 5 shows the basic architecture of CNN that has four layers, which consists of a convolutional layer, nonlinearity layer, pooling layer, and fully connected layer.

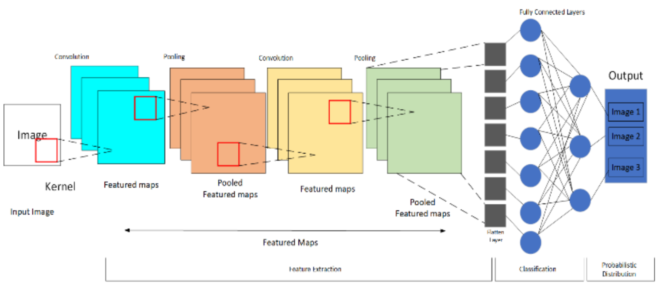

Figure 5. Basic architecture of CNN.

1) *MobileNet*

MobileNet is a simple CNN architecture that uses depth-wise separable convolutions. The depth-wise separable convolution is a separate convolution that determines a standard convolution and a point-wise convolution. For MobileNets, the depth-wise convolution applies a single filter to each input channel. The point-wise convolution then applies a 1×1 convolution to combine the outputs of the

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 10, Issue 9, September 2023

depth-wise convolution. The process of combining the inputs and applying a standard convolution to both filters helps to produce a new set of outputs in one step. The depth-wise separable convolution splits the output layers into two, a separate layer for filtering and a separate layer for combining. These factorization effects drastically reduce computation and the model size [4]. The figure 6 represents the point-wise convolution in the case of depth-wise separable convolution.

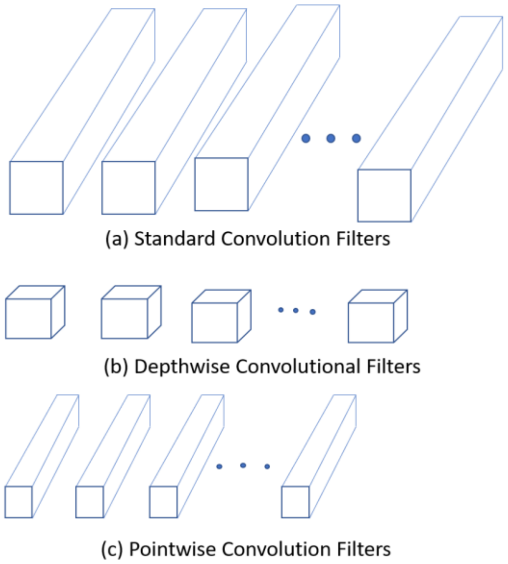

Figure 6. Point-wise convolution in the context of depth-wise separable convolution.

The combination of depth-wise convolution and pointwise convolution is called depth-wise separable convolution. A small reduction in accuracy occurs when MobileNet uses a 3×3 depth-wise separable convolution. The figure 7 shows the standard convolution layer and the depth-wise separable convolution.

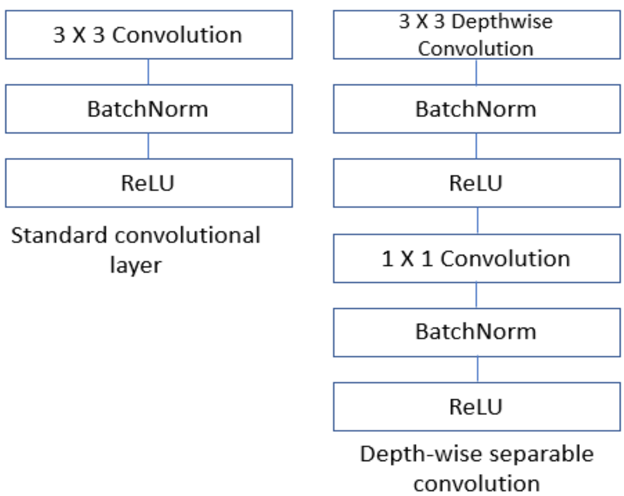

Figure 7. Left: Standard convolutional layer with the batch norm and Rectified Linear Unit (ReLU). Right: Depth-wise separable convolutions with depth-wise and point-wise layers followed by batch norm and ReLU.

2) *MobileNetV2*

MobileNetV2 is a Neural Network architecture that attempts to perform effectively on mobile devices. This is based on an inverted residual structure where the residual connections are between the restricted layers.

MobileNetV2 is the modified version of MobileNet. It is modified by introducing an inverted residual block, linear bottlenecks, and the ReLU6 activation function in place of ReLU. [14]

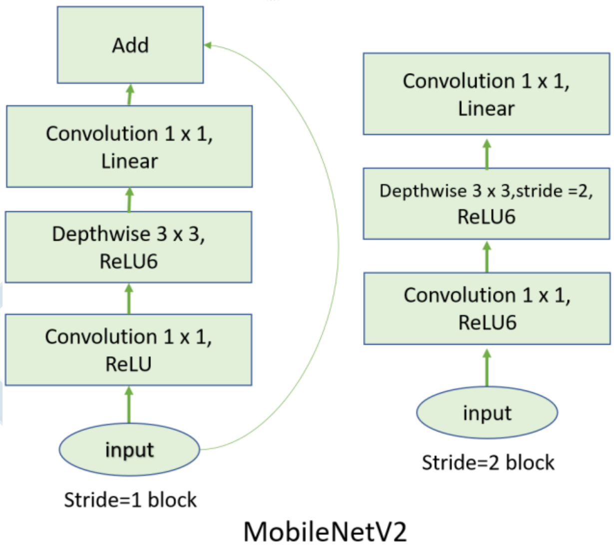

Figure 8. MobileNetV2.

The figure 8 shows the architecture of MobileNetV2. The architecture contains the initial convolution layer with 32 filters, followed by 19 residual bottleneck layers. This uses ReLU6 as the nonlinearity when used with low precision computation to increase the robustness. Also, it measures performance based on ImageNet [13] classification, COCO object detection [8], and VOC image segmentation [2]. It gives the understanding of accuracy, the number of operations measured, actual latency, and the number of parameters. [14]

On the ImageNet dataset, MobileNetV2 architecture improved the performance points. For object detection tasks, MobileNetV2 detects real-time detectors on the COCO dataset, both in terms of accuracy and model complexity. In addition, when MobileNetV2 was combined with the SSDLite detection module, it had 20 times less computation and ten times fewer parameters than YOLOv2. Moreover, the convolutional block in MobileNetV2 has a unique property that separates the bottleneck inputs' network expressiveness. [14]

3) *ResNet-50*

ResNet-50 is a convolutional neural network that is 50 layers deep. An existing pre-trained version of the network can classify images into many object categories, which means, the network can learn a substantial number of feature representations for a wide range of images [7, 3].

The disadvantage of Convolutional Neural Networks is Vanishing Gradient Problem. During backpropagation, the value of the gradient decreases significantly, and thus hardly any change comes to weights. This is resolved by using ResNet-50 architecture. ResNet-50 architecture makes use of 'SKIP CONNECTION'. Skip connection means adding the original input to the output of the convolutional block [3]. The architecture of ResNet-50 is shown in figure 9.

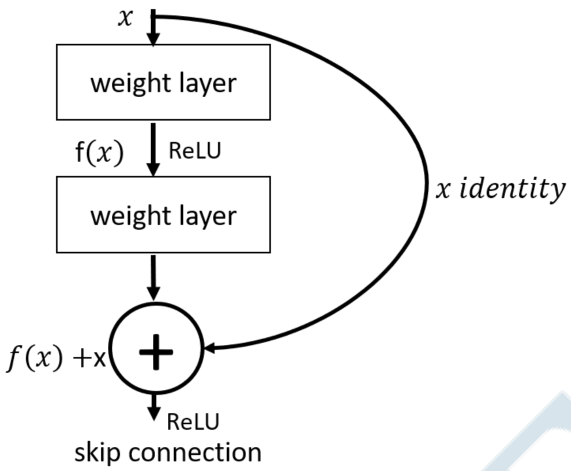

Figure 9. ResNet-50.

All algorithms train on the output '*y*' but, ResNet-50 trains on *f(x)*. ResNet-50 tries to make *f(x)=0* so that *y=x*. 'SKIP CONNECTION' is a direct connection that skips over some model layers. The output is different due to this skip connection. Without the skip connection, input x is multiplied by the layer weights, and then a bias term is added [11].

4) VGG-19

VGG-19 is a convolutional neural network with 19 layers of depth. An existing pre-trained version of the network can classify images into many object categories, which means the network can learn a substantial number of feature representations for a wide range of images [7, 3]. VGG-19 network has an image input size of 224x224. The number 19 in VGG stands for the number of layers with trainable weights, which means there are 16 convolutional layers and three fully connected layers. The network takes a (224, 224, 3) RBG image as the input [5, 15].

VGG-19 CNN is used as a pre-processing model. It is improved in network depth compared to traditional convolutional neural networks. Instead of using a single convolution, it is structured with multiple convolutional layers and non-linear activation layers. The layered structure extracts image features, using Maxpooling for down-sampling, and modifies the Rectifier Linear Unit (ReLU) as the activation function. The down-sampling layer is used to enhance the anti-distortion ability of the network to the image, and it retains the original features of the sample and reduces the number of parameters [15]. The architecture of VGG-19 is shown in figure 10.

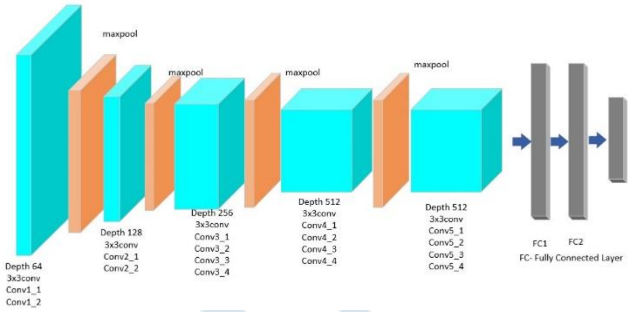

Figure 10. VGG-19 network architecture.

III. DATASET

The images used in the dataset are panels of Medium Density Fiberboard (MDF) or woods like high-quality spruce or pine. The images are classified into train and test datasets. The training dataset contains 2000, and the test dataset is 700 images. This dataset has three exclusive categories; PASS, FAIL, and empty. The dataset in figures 11 and 12 present the defective and non-defective categories, respectively. Due to logistical issues or miscarriage, the panels can be defective or non-defective.

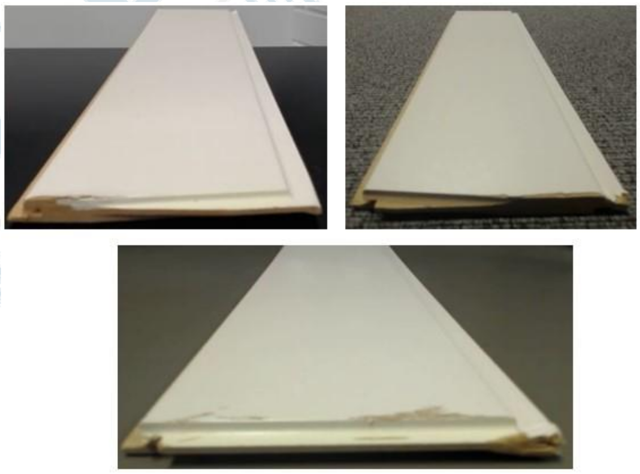

Figure 11. Defective panels.

Figure 12. Non-defective panels.

IFERP
connecting engineers...developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 9, September 2023

IV. RESULTS AND DISCUSSION

The MobileNet model has been trained using the wooden panel dataset. The wooden panels are manufactured massively by different manufacturers. All the wooden panels manufactured by the different manufacturers are of the exact specification, but with different surface structures, which means that they differ in surface color and polish. The method called mean image subtraction normalization is used for normalizing the dataset. To normalize the data, it should calculate the RGB values of the images and subtract them from all the images in the dataset. This helps to make the network less sensitive to varying backgrounds and lighting conditions. The parameters of the model are saved and loaded from a checkpoint file. Figure 13 depicts the evaluation of the network, and figure 14 shows the plot of accuracy when the MobileNet model was trained. The accuracy of the trained network was 99%. The model was trained with a batch size of 32 and 65 steps per epoch for approximately 50 epochs. Figure 13 also shows the precision, recall, f1-score, accuracy, macro average, and weighted average.

```
In [11]:
# evaluate the network
print("[INFO] evaluating network...")
predictions = model.predict(testX, batch_size=32)
print(classification_report(testY.argmax(axis=1),
    predictions.argmax(axis=1), target_names=lb.classes_))

[INFO] evaluating network...
              precision    recall  f1-score

           #       1.00      1.00      1.00
        Fail       0.99      0.99      0.99
        Pass       0.99      0.99      0.99

    accuracy                           0.99
   macro avg       0.99      0.99      0.99
weighted avg       0.99      0.99      0.99
```

Figure 13. Evaluation of the network.

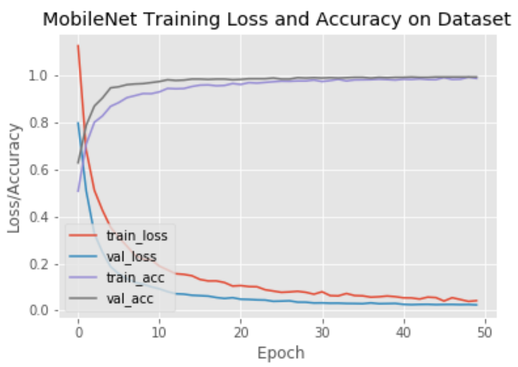

Figure 14. A plot of accuracy when MobileNet model is trained.

Figure 14 gives the number of epochs that progress in the X-axis and a process of Loss/Accuracy in the Y-axis. The red line represents the performance of the training loss, and the purple line shows the training accuracy. The blue color represents the performance of validation loss, and the purple line represents the performance of validation accuracy. According to these figures, there is stability after about the 20th epoch, and the MobileNet model has given an accuracy of 99%.

The wooden panels were tested using the MobileNet model and were analyzed with the results shown in tables 1, 2, and 3 with their corresponding charts.

Table 1. Recorded results of the wooden panels with the black background.

Test with black background			
Wood Product Nrb.	The Truth	MobileNet Prediction	
		Label	Percentage (%)
A	FAIL	PASS	51,2
B	FAIL	FAIL	95,3
C	FAIL	FAIL	95,5
D	FAIL	FAIL	93,3
E	PASS	PASS	64,3
F	PASS	PASS	85,2
G	FAIL	FAIL	94,5
H	PASS	PASS	85,6
I	FAIL	FAIL	90,9
J	PASS	PASS	85,7

Chart 1 corresponds to table 1, which demonstrates the test results of the black background.

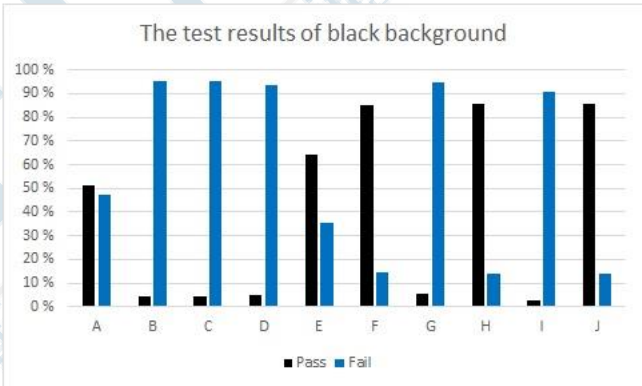

Chart 1. The test results of the black background.

Table 1 represents the test results from the test with a black background. The wooden panels were numbered A to J, and the results show *The Truth*. The MobileNet predictions result with the label and its percentage. The results from the black background represent five PASS labels and five FAIL labels. Out of the five PASS labels, the Wood Product Number [A] shows the label as PASS with 51,2%, but *The Truth* is FAIL. The reason for this variation is the property called LabelBinarizer. Binarizing a data means that it sets feature values to zero or one according to a threshold. The value is greater than the threshold will map to one and values less than or equal to the threshold will map to zero. By default, the threshold is zero and the positive values map to one. Binarization is a common operation on text count data where the analyst can decide to only consider the presence or absence of a feature rather than a quantified number of occurrences for instance. This is a one-vs-all fashion. At

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 9, September 2023

prediction time, the model gave the greatest confidence, so LabelBinarizer makes this easy with the inverse transform method. *The Truth* column in table 1 shows the pass percentage with the PASS result. Likewise, the percentage of failure with the FAIL result.

The wooden panels tested over the green background is given in table 2.

Table 2. Recorded results of the wooden panels with the green background.

Test with green background			
Wood Product Nrb.	The Truth	MobileNet Prediction	
		Label	Percentage (%)
A	FAIL	PASS	95,3
B	FAIL	FAIL	88,5
C	FAIL	FAIL	75,5
D	FAIL	FAIL	70,1
E	PASS	PASS	95,9
F	PASS	PASS	99,1
G	FAIL	FAIL	95,1
H	PASS	PASS	99,1
I	FAIL	PASS	93,6
J	PASS	PASS	95,5

Chart 2 corresponds to table 2, demonstrating the green background test results.

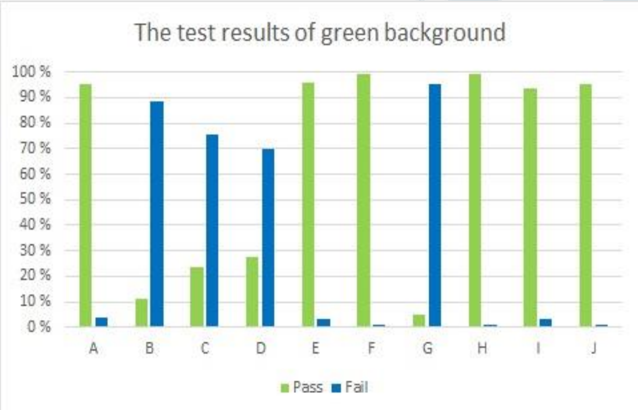

Chart 2. The test results of the green background.

Table 2 represents the test results from the test with a green background. The wooden panels were numbered A to J, and the results show *The Truth*. The MobileNet predictions result with the label and its percentage. The results from the green background represent six PASS labels and four FAIL labels. The Wood Product Numbers [A] and [I] show the wrong label, but *The Truth* is the opposite. The reason for this variation is the property called LabelBinarizer. This is a one-vs-all fashion. At prediction time, the model gave the greatest confidence, so LabelBinarizer makes this easy with the inverse transform method. *The Truth* column in table 2 shows the pass percentage with the PASS result. Likewise, the percentage of failure with the FAIL result.

The wooden panels tested over the grey background is given in table 3.

Table 3. Recorded results of the wooden panels with the grey background.

Test with grey background			
Wood Product Nrb.	The Truth	MobileNet Prediction	
		Label	Percentage (%)
A	FAIL	PASS	82,1
B	FAIL	FAIL	88,6
C	FAIL	FAIL	84,7
D	FAIL	FAIL	86,7
E	PASS	PASS	80,8
F	PASS	PASS	72,7
G	FAIL	FAIL	96,6
H	PASS	PASS	81,9
I	FAIL	PASS	91,6
J	PASS	PASS	94,8

Chart 3 corresponds to table 3, demonstrating the grey background test results.

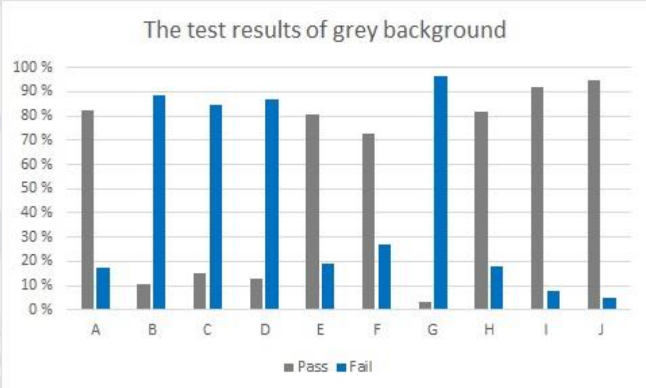

Chart 3. The test results of the grey background.

Table 3 represents the test results from the test with the grey background. The wooden panels were numbered A to J, and the results show *The Truth*. The MobileNet predictions result with the label and its percentage. The results from the grey background represent six PASS labels and four FAIL labels. Out of the six PASS labels, the Wood Product Number [A] shows the label as PASS with 82,1%, but *The Truth* is FAIL, and the Wood Product Number [I] shows the label as PASS with 91,6%, but *The Truth* is FAIL. The reason for these variations is the property called LabelBinarizer. This is a one-vs-all fashion. At prediction time, the model gave the greatest confidence, so LabelBinarizer makes this easy with the inverse transform method. *The Truth* column in table 3 shows the pass percentage with the PASS result. Likewise, the percentage of failure with the FAIL result.

The chart 4 demonstrates the highest PASS and FAIL percentages of the wooden panels. The highest PASS percentage from the black background is 85,7%, and the highest FAIL percentage is 95,5%. Also, the highest PASS percentage from the green background is 99,1%, and the highest FAIL percentage is 95,1%. Likewise, the highest PASS percentage from the grey background is 94,8%, and the highest FAIL percentage is 96,6%. The above results are promising, since both the PASS and FAIL percentages are above 80%.

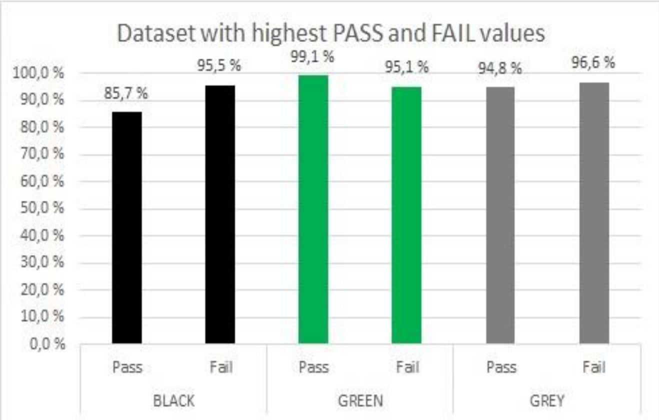

Chart 4. Dataset with the highest PASS and FAIL values.

V. CONCLUSION

The defect detection of MDF fiberboard or wooden panels has high demand in the furniture industry, products for homes, offices, shops, organizations, or institutions. Therefore, a defect detection process called MobileNet is used for checking the quality of a wooden panel by applying a convolutional neural network algorithm to achieve better accuracy. MobileNet algorithm was used to get the best performance in fault detection. The performance of MobileNet is with the highest recall of 0,99 compared with other existing models. The outcome of the study is to achieve high accuracy quality detection of wooden panels using computer vision approaches. In this study, this model was tested over ten different samples of wooden panel for defect detection using one of the neural network approaches. It is possible to apply the same technique over the other neural network models with more samples in the future. A newly intelligent artificial intelligence-based model will apply in the future to demonstrate the different defects in the wooden panel.

REFERENCES

[1] Saad Albawi, Tareq Abed Mohammed, and Saad AlZawi. Understanding of a convolutional neural network https://doi.org/10.1109/ICEngTechnol.2017. 8308186, August 2017. pp. 1-6.

[2] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. https://doi.org/10.1007/s11263-014-0733-5, June 2015. pp. 98–136.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. https://arxiv.org/pdf/1512.03385.pdf, June 2016. pp. 770778.

[4] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. https://doi.org/10.48550/arXiv.1704.04861 April 2017.

[5] The MathWorks Incorporation. Vgg-19 convolutional neural network. https://www.mathworks.com/help/deeplearning/ref/vgg19.html, January 2023. Accessed: 2023-01-24.

[6] Mark Irle, Marius C. Barbu, Peter Niemz, Luisa Hora Carvalho, Jorge Manuel Martins, Carlos A. V. Costa, Lech Muszyski, Maximilien E Launey, Milan Sernek, Manfred Dunky, Jochen Aderhold, Burkhard Plinke, Olaf Treusch, Martin Ohlmeyer, and Wulf Paul. Wood-based panels: An introduction for specialists, August 2010. ISBN: 978-1902316-82-6.

[7] Stanford Vision Lab. Imagenet. http://www.image-net.org, 2020. Accessed: 2023-01-24.

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C. Lawrence´ Zitnick. Microsoft coco: Common objects in context. https://doi.org/10.48550/arXiv.1405.0312, Feb-ruary 2015. pp. 740–755.

[9] Huajie Liu, Dadong Wang, Ke Xu, Peng Zhou, and Dongdong Zhou. Lightweight convolutional neural network for counting densely piled steel bars. https://doi.org/10.1016/j.autcon.2022.104692, February 2023. ISSN: 0926-5805.

[10] The MathWorks. What is a convolutional neural network? https://se.mathworks.com/discovery/convolutionalneural-net work-matlab.html, January 2023. Access-ed: 202301-24.

[11] Aditi Rastogi. Resnet50. https://blog.devgenius.io/resnet 50642934db431, March 2022. Accessed: 2023-01-24.

[12] Adrian Rosebrock. Video classification with keras and deep learning. https://pyimagesearch.com/2019/07/15/videoclassification-with-keras-and-deep-learning/, 2019. Accessed: 2023-01-24.

[13] 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. https://doi.org/10.1007/s11263-015-0816-y, January 2015. pp. 211–252.

[14] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. https://doi.org/10.48550/arXiv.1801.04381, March 2019. pp. 4510-4520.

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. https://doi.org/10.48550/arXiv.1409.1556, April 2015.