Embedded Selforganizing Systems

Special Issue Topic: “International Symposium on Computer Science and Educational Technologies “

Automated Identification of Wood Surface Defects Based on Deep Learning

Besfort Syla
Computer Engineering Dept.
Technische Universität Chemnitz
Munich, Germany
besfort.syla@s2020.tu-chemnitz.de

Shadi Saleh
Computer Engineering Dept.
Technische Universität Chemnitz
Chemnitz, Germany
shadi.saleh@informatik.tu-chemnitz.de

Wolfram Hardt
Computer Engineering Dept.
Technische Universität Chemnitz
Chemnitz, Germany
wolfram.hardt@cs.tu-chemnitz.de

***Abstract*—Wood plates are widely used in the interior design of houses primarily for their aesthetic value. However, considering its esthetical values, surface defect detection is necessary. The development of computer vision and CNN-based object detection methods has opened the way for wood surface defect detection process automation. This paper investigates deep-learning applications for automatic wood surface defect detection. It includes the evaluation of deep learning algorithms, including data generation and labeling, preprocessing, model training, and evaluation. Many adjustments regarding the dataset size, the model, and the modification of the neural network were made to evaluate the model's performance in the specified challenge. The results indicate that modifications can increase the YOLOv5s performance in detection. The model with GCNet added and trained in 4800 images has achieved 88.1% of mAP. The paper also evaluates the time performance of models based on different GPU units. The results show that in A100 40GB GPU, the maximum time to process a wood plate is 2.2 seconds. Finally, an Active learning approach for the continual increase in performance while detecting with the smaller size of manual labeling has been implemented. After detecting 500 images in 5 cycles, the model achieved 98.8% of mAP. This scientific paper concludes that YOLOv5s modified model is suitable for wood surface defect detection. It can perform with high accuracy in real time. Moreover, applying the active learning approach can facilitate the labeling process by increasing the performance during detection.**

Keywords: Wood surface, defect detection, YOLOv5, Active Learning

I. Introduction

Wood is a ubiquitous natural material with diverse applications across various industries, including furniture, construction, and paper production. The quality of wood is a critical factor in ensuring the safety and durability of structures and products made from it. Traditionally, the identification of wood defects has been performed manually by experts in the wood industry, which is time-consuming and labor-intensive. With the development of technology, new hardware methods were invented to make the process easy. Those methods are known as Non-destructive wood testing methods [1]. With the advent of computer vision, the automatic identification of wood surface defects has become a promising area of research. The Line-camera scanners have been developed for capturing high-resolution images for wood industry applications. In the beginning, those scanners were combined with the classical (hand-crafted) machine learning methods for detecting defects. Those methods, despite usability, leaked performance regarding precision.

With the development of deep learning methods, such as convolutional neural networks (CNNs), there has been a great interest in applications in the detection and classification of wood defects. Also, several approaches have been applied in small-size damage detection.

Convolutional Neural Network methods for object detection can be categorized based on the organization into two groups: The two-stage approach and the one-stage approach. The two-stage approach contains the generation of the region of proposals in the first stage, then applies CNN, classification, and detection using boundary boxes in the second stage, known as the Classification and Regression stage. Differently from the two-stage approach, the one-stage approach makes classification and regression directly from the input image. In subsections are listed algorithms designed based on both methods. One of the representatives of two-stage methods is You Only Look Once – YOLO [17] which is still under development.

This research work aims to develop an automated identification system of wood surface defects based on current state-of-the-art detection methods such as YOLOv5[22] with several modifications for better accuracy in small-size detection and with active learning for a better-trained model with a rich, confident dataset.

However, there exist problems with the application and precision of such methods:

- Higher detection precision needed (above 90%)
- Real-time detection
- Previous solution more oriented on general defect detection than on class of error detection
- Continual learning for new defects

The objectives of this work are:

- Development of the wood surface defect detection system based on active learning
- Modification of the current state-of-the-art model for small-size detection

- Achievement of higher mean average precision (above 90%)
- Real-time detection

The wood surface is characterized by décor which can lead to misdetection of the real defect. It is challenging for the YOLOv5 network to detect small targets in the meantime ensuring the detection of larger targets.

To overpass this challenge we have proposed the introduction of attention layers in the backbone for feature extraction such as SENet[56] and GCNet[57].

Another challenge is to detect defects in high-resolution images without losing the information of the small defects. As the images are captured by a line scanning camera with a high resolution, we have proposed several steps of image preprocessing for splitting the image without down-sampling or reducing the resolution. While the splitting of the image and detection in several images sequentially can lead to longer time and non-real-time detection, we have monitored the iteration time during detection and compared it on several GPUs with different qualities.

The wood surface detection by a supervised learning technique requires a large dataset for model training. To overcome this challenge, we tried to evaluate the transfer of the weights from a model trained on a larger dataset (COCO 2017) with no similarity to ours and to evaluate the size of the dataset by generating images based on images of raw wood plates and segmented defects using a Defector-API based on GAN network with Autoencoder using U-Net as a generator. In the end, we have performed an Active Learning strategy for continual improvement of the model on newly detected data with the least confidence.

II. Related Works

During the process of researching our topic, we have classified several scientific papers and works, which are listed below with their ideas, use cases, implementation, results, and conclusions.

Zhao et al. [44] proposed a real-time defect detection system for particleboard surfaces using an improved version of the You Only Look Once version 5 (YOLOV5) deep learning model. The system achieved a mean average precision (mAP) of 0.906, outperforming other state-of-the-art object detection models for particleboard surface defect detection.

Ding et al. [45] proposed a defect detection system for solid wood panels using a modified version of the Single Shot Detector (SSD) deep learning algorithm. The system achieved a mean average precision (mAP) of 0.927, outperforming other state-of-the-art object detection models for solid wood panel defect detection. The system also demonstrates robustness to various lighting conditions and camera angles, suggesting its practicality for industrial use.

Fang et al. [46] proposed an automated system for detecting surface knots on sawn timbers using the YOLOv5 deep learning model. The proposed system achieves a mean average precision (mAP) of 0.925, outperforming other state-of-the-art object detection models for surface knot detection.

Bunjemea et al. [47] proposed a modification to the YOLOv5 deep learning model to improve small object detection in the context of autonomous vehicles and named it YOLO-Z. They used DenseNet instead of CSPDarknet-53 in the backbone and FPN instead PAN in the neck. YOLO-Z achieved mAP a 95.5 % on small objects, outperforming the state-of-the-art model.

Yao et al. [19] proposed a real-time detection algorithm for detecting defects in kiwifruits using the YOLOv5 deep learning model. The proposed algorithm achieved an average precision (AP) of 94.7% on the test set, outperforming other state-of-the-art methods for kiwifruit defect detection. The proposed algorithm can potentially improve quality control in the food industry.

III. Methodology and Implementation

In this chapter will be discussed the concept of the paper with the proposed models and their implementation. The concept includes image preprocessing, dataset evaluation, and modification of the YOLOv5 object detector to create a new model with higher performance. The different models of YOLOv5 have been trained, and their results have been discussed. In our concept, we propose using the Active learning process for continually training the model in newly detected images. Also, the impact of transfer learning during the training process is evaluated. The schematical representation of the concept is represented in Figure 1.

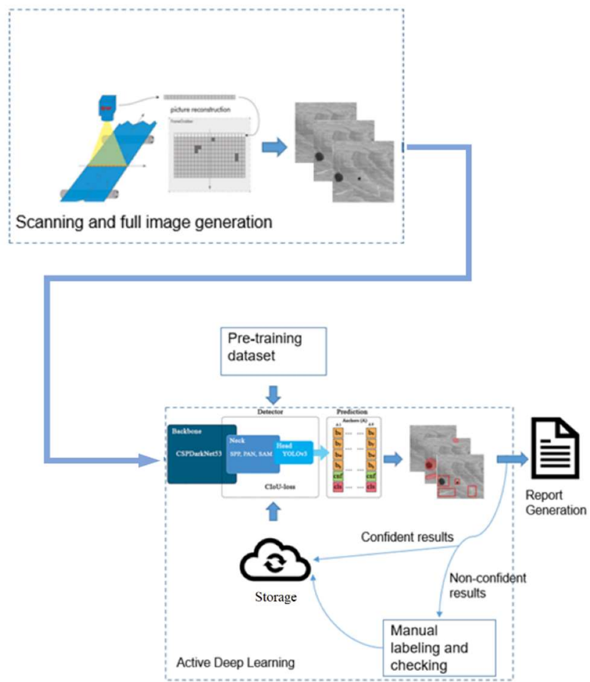

Fig. 1. Defined classes of defects

A. Technical Architecture

This research uses the YOLOv5 object detector provided by Ultralytics and the PyTorch library, which is an open-source machine learning framework for deep learning developed by Google Brain. The hardware site used is the Google Colab platform, which provides a runtime environment in different graphic cards and a high amount of RAM suitable for training large amounts of data. Two shared GPUs have been used from Google Colab, the Tesla T4 with 16GB and the A100 with 40GB. The proposed models have been trained on A100 and tested on all the above graphic cards.

ModifiedOpenLabeling[51], a tool provided in GitHub, has been used to label the dataset.

B. Initial Dataset

The dataset is collected especially for the use case of this project. The initial dataset has 20 images of different sizes from 2874×4174 to 5477 × 5414 pixels. It was collected using a 4i Scanner, which measures the plates that arrive individually via a conveyor belt and are scanned from below and above with a line camera as they pass through. The maximum plate size can be up to 1.5 meters wide and 3 meters long, depending on the Scanner. Approximately ten plates per minute are checked in transit. Special features of this system are the scaling invariance and the illumination invariance. The illumination and scaling invariance comes from using a line scan camera. The image dataset represented 6 different wood designs: Sonoma Eiche, Mammut Eiche, Bergeiche, Marmor Dunkel, Beton treuffelgrau, and Wildeiche. The size of the images was not appropriate for YOLO v5 input, leading to the loss of pixels and not a clear representation of defects. To address this, cropping of images was performed and 277 images of size 1280×1280 pixels were generated. 11 classes of defects were labeled manually by defining the bounding box around them. These classes included decay, deep hole, shallow hole, crack, scratch, deep scratch, saw scratch, live knots, dead knots, sticker, and carving. The defined classes in the dataset are represented in Figure 2.

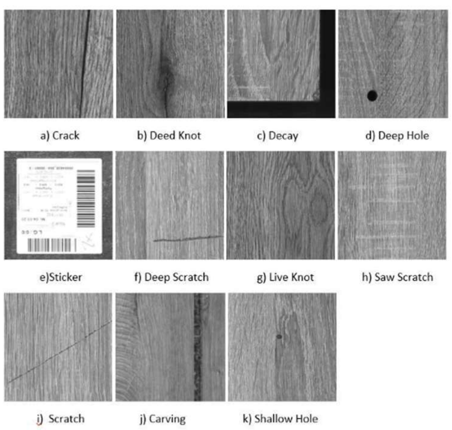

Fig. 2. Defined classes of defects

For the training of a model, the dataset should be divided into training and validation data in report 80% to 20%. By this ratio, the training set contains 218 images, and the validation set 59 images.

C. Extended Dataset

The dataset has been extended with provided ground images and segmented defects. In general, 258 images without defects and 139 segmented defects were provided by the company. The size of base images variated from 7.5 MB to 187MB or 1704×4600 pixels to 8320×23626 pixels. For generating the dataset, company Hecht AG provided Defektor-API and data augmentation tool implemented in [52]. To create variances and not focus on specific features for creating overfitting, the images without defects and segmented defects were transformed during defection. The transformation was mainly based on brightness, rotation, size, and deformation. The deformation has been implemented using a resolution change of the image and Gaussian low pass filter. First, the coordinates of the images were extracted from a lower-resolution image and then remapped on the original image. Then the Gaussian filter has been applied to filter the high band noises and reshape the image with the result. The parameters for the filter have been chosen: sigma in the range of 15 to 20 and alfa 5 to 20. The sigma parameter represents the frequency of the filter, and the alfa represents the gain. The higher the gain, the higher the shift of XY, and the lower the sigma, the higher the representation of the high band noises [52].

The extended dataset contains 4800 images of size 1280 × 1280 pixels consisting of 800 images for each wood plate décor. The background images or True negative images represent 21% of the dataset. In the extended dataset, the number of defect classes is reduced to six after no need for several class detection which represents the decors of the wood design. The defect classes include decay, deep hole, shallow hole, scratch, a deep scratch, and sticker.

D. Image Preprocessing

The output of the 4i Scanner, as we explained in the Dataset section, is constructed in high resolution, which corresponds to the size of the wood plate with a maximum of 1.5m wide and 3m long. The images captured by the Scanner also contain a sliding track in the background which moves the wood plate inside the Scanner. To get rid of the track and get the wood surface as a region of interest, preprocessing steps have been implemented to extract the contour of the wood plate. After extracting the wood plate image and applying the perspective transform, the image needed to resize to the nearest multiplied value of 1280 in width and height. This resizing has a low effect on quality loss. In the end, the image is splinted by the nearest multiplier value of 1280 found before in N × M images of 1280 × 1280 pixels. The 1280 × 1280 images are now fit to be used as input in YOLOv5 convolutional neural network. The flow chart of the preprocessing steps can be found in Figure 3.

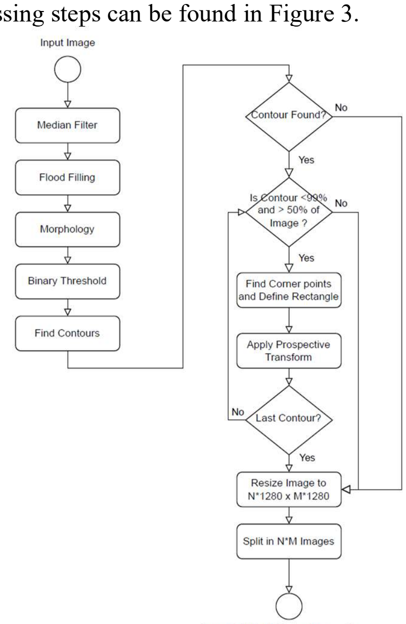

Fig. 3. Image Preprocessing Flow Chart

E. Modifications in Convolutional Neural Network

During the performance of the first trained models, it has been seen as essential to add attendance layers as proposed in different papers [53][54][55]. Therefore, we have selected to train two modified models, the first by adding Squeeze and Excitation Network in the Backbone of the YOLOv5. Feature fusion between channels of the convolution operation in the backbone network is the main focus of the SENet[56]. The primary innovation of this network is the model's ability to automatically determine the significance of various channel features by concentrating on the connections between channels. The SE module mainly performs operations using excitation and compression (Squeeze). In order to encode the full spatial feature on a channel as a local feature, the Squeeze operation uses global average pooling. The Squeeze operation first collects the channel information, then builds a gate mechanism out of two fully linked layers and activates it with the Sigmoid function. In the end, the output weight of SENet is mapped to the original feature. The SENet has been incorporated in the last feature extraction layer (C3) of the backbone of the YOLOv5 small model.

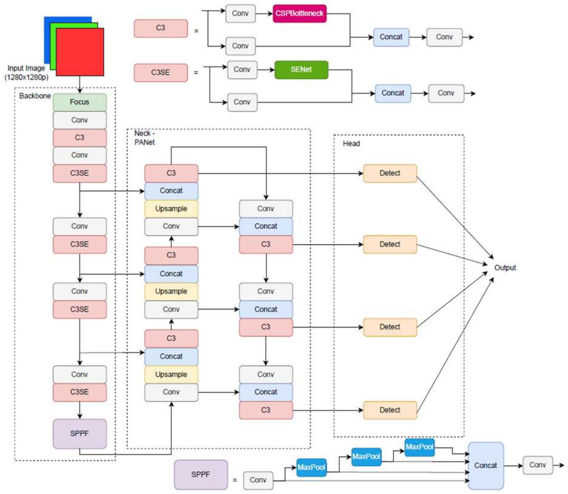

Fig. 4. The schema of modified YOLOv5s with SENet added

The second model has been modified by adding Global Context Network (GCNet)[57]. The suggested GCNet can effectively model the global context through additional fusion as the Non-Local network (NLNet). As the NLNet is heavy and difficult to integrate into several layers, GCNet acquires SENet, which represents the lightweight property.
As a result, on critical benchmarks for multiple recognition tasks, GCNet can outperform both NLNet and SENet thanks to more efficient global context modeling.
By using GCNet as the backbone network in YOLOv5, the architecture can benefit from its ability to capture both global and local context information, which can improve the accuracy of object detection. During the application of GCNet in the YOLOv5 small model, we have replaced the current feature extractor module, which contains CSPBottlenecks with three convolutional layers (C3 module), with a modified feature extractor module, C3_GC, which has three convolutions and CSPBottleneck, the Global Context block. Except for the first feature extractor module, which remained the same, all other C3 modules have been replaced with C3_GC.

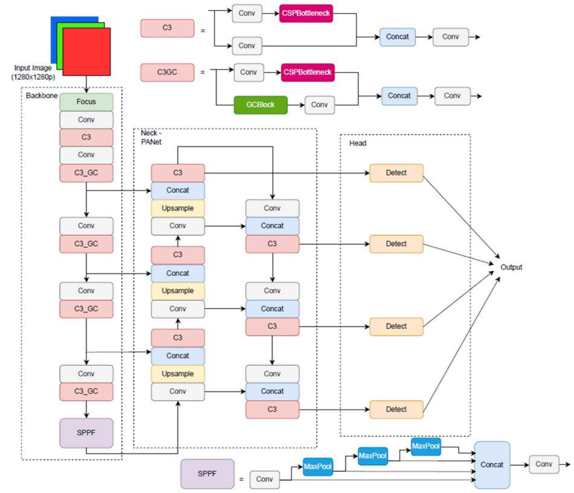

Fig. 5. The schema of modified YOLOv5s with GCNet added.

F. Transfer Learning

This work also conducts the impact of transfer learning on the performance results of YOLOv5 in object detection. The use of transfer learning was tested by using pre-trained weights with and without freezing the layers.
Freezing layers means that layers selected to be frozen are prevented from training on the target dataset by transferring the initial weights trained on the pre-trained dataset. Only the weights remaining unfrozen layers are tuned on the target dataset while training the network. The training of the dataset without frozen layers means all weights are typically continuously tuned depending on the features of the target dataset. The creators of YOLOv5 have provided the pre-trained model and its weights trained in the COCO dataset. The COCO dataset contains over 330 000 images displaying 1.5 million objects and is trained on 90 classes. Training with frozen layers requires fewer resources while the process of training and also performs faster training, but it may result in a negative transfer and reduction of final trained accuracy.
In our research, we have used both frozen and unfrozen transfer learning in different phases. The standard YOLOv5s was trained with a completely frozen backbone. As the modifications have happened the in the backbone of feature extraction layers, it was not possible to use transfer learning with frozen layers in before mentioned modified models. The model used to be trained on the extended dataset has not been pre-trained as it was interesting for the topic to see the performance without transferring weights.

G. Active Learning

This paper introduces Active Learning, a semi-supervised learning method for object detection using deep learning. The dataset used is unlabeled and the labeled dataset pool is empty in the first stage. The dataset consists of 500 images divided into five parts of 100 images. 30 images with the least confident result are selected for manual labeling, and the images without objects are entitled to 0 as a confidence score. The model was trained on 50 epochs and the batch size was set to 4. After each cycle, the trained weights were used on the next training cycle. This process was possible by using transfer learning for weight transfer from pre cycled model.

IV. Results and Discussion

The first training includes modified models based on the YOLOv5s model. The model with added GCNet and the pre-trained YOLOv5s model with a frozen backbone have shown better results in precision than the model with added SENet. In the recall metric, the models have almost similar results after 300 epochs, but the model with added GCNet and the standard pre-trained model with a frozen backbone have faster saturation (after 70 epochs).

Regarding mAP, the YOLOv5+GCNet model has slightly better results in mAP0.5 than the standard model with a frozen backbone. Meanwhile, the model with SENet added, after 300 epochs, but with slower saturation, is not far behind. The situation is also reflected in mAP0.5-0.95, where SENet is still lower in the result and with slower saturation than the other two models.

On the other hand, the validation results of mAP make a difference. The model with GCNet added has shown better results than the two other models compared in this phase. This model has reached 72% of mAP 0.5 and 44.3% of mAP 0.5-0.95. Meanwhile, the standard YOLOv5s model with a frozen backbone has reached 69.9% of mAP 50% and 44.1 of mAP with an IoU threshold of 50-95%. The model with added SELayer showed the lowest performance with 68.6% (mAP0.5) and 36.9 (mAP0.95). The standard YOLOv5s with frozen backbone and SENet added model leaks on detection of defect with IoU threshold 50%. Otherwise, the GCNet added model had overpassed the standard pre-trained model for 1.6% in mAP0.5 and 8% in mAP0.5-0.95.

The model trained in an extended dataset during the training process has achieved nearly 90% of mAP with IoU 50%. It seems that after 120 epochs, the results became stable. Compared with the second phase, it has a 20% improvement. As per mAP0.5-0.95, the model has achieved nearly 60% during training, and the saturation of the result came after 150 epochs. The results compared to the second phase model are nearly 15% higher during the training. During the validation, we have the same situation. The mAP0.5 result is 88.1% which is 16.1% higher than the second phase model. The result of mAP with IoU threshold variable 50-95% in validation is 58.4%. This result is 12.1% higher than the second phase model.

TABLE I. Representation of validation results for mean average precision with IoU threshold 50% and IoU threshold 50-95% for respective trained models.

Model	Parameters			
	Batch Size	*Epochs*	*mAP50%*	*mAP50-95%*
Pretrained Yolov5s6 (freeze backbone)	16	300	0.699	0.441
Pretrained Yolov5s+SENet	16	300	0.686	0.369
Pretrained Yolov5s + GCNet	16	300	0.72	0.443
Non-pretrained Yolov5s+ GCNet model (Extended Dataset)	16	300	0.881	0.584

TABLE II. Representation of iteration time of detection for trained models in various GPUs

Model	Graphical Processing Units		
	GTX1650Ti 4GB	*Tesla T4 16GB*	*A100 40GB*
Pretrained Yolov5s6 (freeze backbone)	139ms	21.6ms	14.3ms
Pretrained Yolov5s+SENet	110ms	20.2ms	12.8ms
Pretrained Yolov5s + GCNet	143ms	23.1ms	17.8ms
Non-pretrained Yolov5s+GCNet model (Extended Dataset)	143ms	23.7ms	17.4ms

This work includes a comparison of detection time per images on various Graphic Processing Units. In the case of the YOLOv5s + GCNet model, it has achieved 23.1ms for the detection of one image with the size of 1280 × 1280 pixels on a Tesla T4 GPU with 16GB. We can see that this time is lower in A100 GPU taking just 17.8ms. Compared with the other two models (standard and with SENet), the GCNet added model has a higher detection time due to more GFLOPs. The SENet added model takes less time than two other models for detection. Based on the results, for a scanned wood plate (the dataset image with the largest size of 8320 × 23626 pixels), in T4 GPU it needs less than 3 seconds to detect defects in a full image. Of course, this time is lower if A100 GPU with 40GB is used. In A100, the model takes 17ms to detect an image of size 1280 × 1280. For the 8230 × 23626 pixels image, it takes 2 seconds to detect defects. Considering that the scanner scans ten wood plates in one minute, the maximum time for proceeding with one wood plate would be 6 seconds. In low-quality 1650Ti GPU, it will need 143ms per image or approx. Eighteen seconds per whole wood plate.

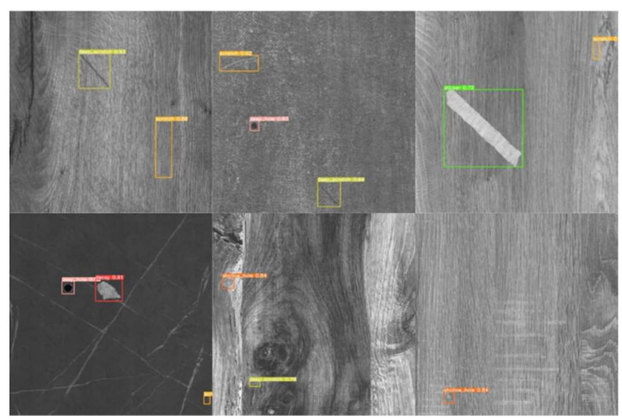

Fig. 6. Detection results in all décors for all classes of defects of Yolov5s + GCNet model trained on the extended dataset.

A. Active Learning Process Results

After the first phase, the mAP0.5 got raised to 95.8%. The mAP0.5-0.95 also got raised to 78.6%. In the second phase, there is a dropdown of mAP results. The mAP0.5 goes under 60%, and the mAP 0.5-0.95 drops to 50%. In the third phase, mAP0.5 rises to 90%, and mAP0.5-0.95 rises to nearly 80%. The rise also continues in the fourth phase, where mAP0.5 raised to 99.5%, which is a very high performance. And the mAP0.5-0.95 raises to above 90% to 93.1%. This means that the location of bounding boxes of detected inferences is so accurate, nearly in the same place as with labels of the validation set. In the final phase, the mAP0.5 results become more stable, and the drop is just 0.7%, meaning 98.8%. On the other hand, mAP0.5-0.95 drops to 74.7%.

Comparing the results before and after active learning, we find out that the improvement in mAP 0.5 is 10.7%, and in mAP 0.5-0.95 is an improvement of 16.3%.

V. Conclusion and Further Works

The results demonstrate the excellent performance of YOLOv5 models in detecting wood surface defects. The modifications in feature extraction by adding attendance layers such as GCNet has shown improvement in increasing performance. Otherwise, adding SELayer showed a decrease in performance. At the same time, transfer learning with freezing all layers has shown that it is not an excellent choice to undertake. Adding GCNet has performed an increase of 1.6% in mAP0.5 while the mAP0.5-0.95 got increased by 8%. The modified YOLOv5s+GCNet proved that it could be real-time operable while detecting the maximum wood plate image, while the model extra-large failed in this direction. For an image of 1280 × 1280 pixels, it needs 143ms in low-performance GPU, 23.7ms in medium-performance GPU and 17.4ms in high-performance GPU. Moreover, while performing dataset evaluation, the modified model YOLOv5s+GCNet can perform much better in larger datasets. Considering that transfer learning was not performed, we have achieved 88.1% of mAP0.5 and 58.4% of mAP0.5-0.95. These results show a significant improvement in mAP by more than 10%. Those improvements come from a variety of data used with balanced representation for all types of wood plates. But the results could have been higher if labeling had been performed in a more professional way by an expert in the wood industry. Finally, the Active learning approach proved the necessary need for application in the wood industry for the improvement of performance. It has been shown that the mAP performance can change drastically while continually training the model in new detections. Achieving 98.8% of mAP0.5 and 74.7% of mAP0.5-0.95 after 500 detected images greatly impacts this process to create a highly performable model.

Despite the result, more work needs to be done in this field.

A more detailed definition of the defects is required for labeling which leads to a better classification of defects. This process requires the involvement of an expert in the wood industry which none of the authors can replace.

Also, the depth of the defect represents a piece of important information for the reliability of the system. As sensor fusion techniques have shown remarkable results in automotive applications, a similar approach needs to be researched for combining line-scanning cameras and laser-based depth sensors with CNN object detection techniques for defect detection. Recently, YOLOv8 has been released and has shown an increase in performance compared with YOLOv5. In the future, newer modifications should be researched to improve defect detection techniques.

Acknowledgment

We are thankful to Hecht AG for providing the dataset and other material for the realization of this work.

References

[1] M. Zielińska and M. Rucka, "Non-destructive Testing of Wooden Elements," in Proceedings of the 2020 International Conference on Non-Destructive Evaluation (ICNDE), 2020, pp. 1-6.

[2] P. Niemza and D. Mannes, "Non-destructive testing of wood and wood-based materials," in Proceedings of the 2012 IEEE International Conference on Imaging Systems and Techniques, Manchester, United Kingdom, July 16-18, 2012, pp. 464-468.

[3] P. Palma and R. Steiger, "Structural health monitoring of timber structures – Review of available methods and case studies," Construction and Building Materials, 2020, vol. 248, p. 118528.

[4] A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, California, USA: O'Reilly Media, 2019.

[5] I.H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," SN COMPUT. SCI. 2, 160 (2021), https://doi.org/10.1007/s42979-021-00592-x.

[6] A. Vahab, M. Naik, P. G. Raikar and S. R. Prasad, "Applications of Object Detection System," International Research Journal of Engineering and Technology (IRJET), vol. 6, no. 4, pp. 4457-4461, April 2019.

[7] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 511-518, 2001.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi:10.1109/CVPR.2005.177.

[9] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004, https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[10] I. Zafar, G. Tzanidou, R. Burton, N. Patel and L. Araujo, "Deep Learning And Convolutional Neural Networks," in Hands-On Convolutional Neural Networks with TensorFlow, Birmingham, Mumbai, Packt Publishing Limited, 2018.

[11] A. Lindholm, N. Wahlström, F. Lindsten and T. B. Schön, "Machine Learning – A First Course for Engineers and Scientists," Cambridge University Press, pp. 124-129, 2022.

[12] H. Hakim and A. Fadhil, "Survey: Convolution Neural networks in Object Detection," Journal of Physics: Conference Series, vol. 1945, 012020, 2021.

[13] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587, 2014.

[14] R. Girshick, "Fast R-CNN," in 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440-1448, 2015.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu and A. Berg, "SSD: Single shot multibox detector," in Proceedings of the European Conference on Computer Vision (ECCV), pp. 21–37, 2016.

[16] F. Sultana, A. Sufian and P. Dutta, "A Review of Object Detection Models based on Convolutional Neural Network," ArXiv abs/1905.01614, 2019.

[17] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.779-788, 2016.

[18] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.6517-6525, 2017.

[19] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," ArXiv:1804.02767v1, 2018.

[20] A. Bochkovskiy, C. Wang and H. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," ArXiv abs/2004.10934, 2020.

[21] M. Maity, S. Banerjee, and S. Chaudhuri, "Faster R-CNN and YOLO based Vehicle detection: A Survey," in 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 1442-1447.

[22] G. Jocher, "YOLOv5," [Online]. Available: https://github.com/ultralytics/yolov5. [Accessed: June 20, 2022].

[23] J. Yao, J. Qi, and J. Zhang, "A real-time detection algorithm for kiwifruit defects based on YOLOv5," Electronics, vol. 10, no. 14, p. 1711, 2021.

[24] R. Padilla, W. L. Passo, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit," IEEE Access, vol. 10, pp. 42290-42306, 2021.

[25] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," IEEE Transactions on Image Processing, vol. 30, pp. 1686-1696, 2021.

[26] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 10, pp. 3624-3638, Oct. 2021.

[27] I. Zafar, G. Tzanidou, R. Burton, N. Patel, and L. Araujo, "Hands-On Convolutional Neural Networks with TensorFlow," Packt Publishing Limited, Birmingham, Mumbai, 2018.

[28] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.

[29] Y. Bengio, "Deep learning of representations: Looking forward," in Proceedings of the International Conference on Statistical Language and Speech Processing (SLSP), 2012, pp. 137.

[30] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, 2016.

[31] A. Sufian, A. Ghosh, A. Sadiq, and F. Smarandache, "A Survey on Deep Transfer Learning and Edge Computing for Mitigating the COVID-19 Pandemic," Journal of Systems Architecture, vol. 108, 2020, Article ID 101830.

[32] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1717-1724.

[33] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks," in Advances in Neural Information Processing Systems (NIPS), 2014, pp. 3320-3328.

[34] H. Shin and J.K. Lee, "How to fine-tune deep neural networks for object recognition," in Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), 2016, pp. 1-8.

[35] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 6848-6856.

[36] [36] M. Tan and Q.V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proceedings of the International Conference on Machine Learning (ICML), 2019, pp. 6105-6114.

[37] B. Settles, "Active learning," Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 6, no. 1, pp. 1-114, 2012.

[38] A. Dasgupta and D. Hsu, "Hierarchical sampling for active learning," in Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 449-456.

[39] K. Lang and E. Baum, "Query learning can work poorly when a human oracle is used," in Proceedings of the IEEE International Joint Conference on Neural Networks, 1992, pp. 335-340.

[40] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1994, pp. 3-12.

[41] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, 1992, pp. 287-294.

[42] S. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in Proceedings of the 18th International Conference on Machine Learning (ICML-01), 2001, pp. 441-448.

[43] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in Proceedings of the 33rd International Conference on Machine Learning (ICML-15), 2015, pp. 1050-1059.

[44] Z. Zhao, X. Yang, Y. Zhou, Q. Sun, Z. Ge and D. Liu, “Real-time detection of particleboard surface defects based on improved YOLOV5 target detection”, In Scientific Reports 11, 2021

[45] F. Ding, Z. Zhuang, Y. Liu, D. Jiang, X. Yan and Z. Wang, “Detecting Defects on Solid Wood Panels Based on an Improved SSD Algorithm”, In Sensors 20(18), 2020.

[46] F. Fang, X. Guo, K. Chen, Z. Zhou and Q. Ye, “Accurate and Automated Detection of Surface Knots on Sawn Timbers Using YOLO-V5 Model”, In BioResources 16(3), 2021.

[47] A. Benjumea, I.Teeti, F. Cuzzolin and A. Bradley, “YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles”, In arXiv:2112.11798, 2021.

[48] B. Battseren, M. S. Harras, D. Orjuela Aguirre, S. Saleh, and W. Hardt, "Tree Detection and Localization Approach for UAV-based Forest Inspection," European Scientific Journal, vol. 9, no. 3, pp. 73-81, 2022. doi: 10.14464/ess.v9i3.557.

[49] C. Wang, A. Bochkovskiy, and H. M. Liao, “Scaled-yolov4: Scaling cross stage partial network,” CoRR, vol. abs/2011.08036, 2020. [Online]. Available:https://arxiv.org/abs/2011.08036.

[50] S. Saleh, S. Khwandah, A. Heller, W. Hardt, and A. Mumtaz, "Traffic signs recognition and distance estimation using a monocular camera," in Actual Problems of System and Software Engineering (APSSE 2019), 2019.

[51] I. Goncharov, "ModifiedOpenLabeling," [Online]. Available: https://github.com/ivangrov/ModifiedOpenLabelling.

[52] J. Bruckner, "Deep learning for image data-based surface inspection of wood décor,"Master's thesis, Hochschule Heilbronn, 2018.

[53] J. Ye, Z. Yuan, C. Qian, and X. Li, "CAA-YOLO: Combined-Attention-Augmented YOLO for Infrared Ocean Ships Detection," Sensors, vol. 22, no. 10, p. 3782, 2022.doi: 10.3390/s22103782.

[54] J. Zhang, J. Lei, X. Qin, B. Li, Z. Li, H. Li, Y. Zeng, and J. Song, "A fitting recognition approach combining depth-attention YOLOv5 and prior synthetic dataset," Applied Sciences, vol. 12, no. 21, p. 11122, 2022. doi: 10.3390/app122111122.

[55] J. Li, C. Liu, X. Lu, and B. Wu, "CME-YOLOv5: An efficient object detection network for densely spaced fish and small targets," Water, vol. 14, no. 15, p. 2412, 2022. doi:10.3390/w14152412.

[56] Y. Hu, J. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132-7141.

[57] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local Networks Meet Squeeze- Excitation Networks and Beyond," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 1429-1438.

[58] A. Efraimsson and E. Lemón, "Railway Fastener Fault Detection using YOLOv5," Master's thesis, Uppsala University, Sweden, 2021. [Online]. Available: https://www.utn.uu.se/sts/student/wpcontent/uploads/2022/02/2207_ Alva_Efraimsson_Elin_Lemon.pdf. Accessed on: December 21, 2022