Wood Defects Computer Vision Project

*Note: Sub-titles are not captured in Xplore and should not be used

Kai Xin Ng
Lee Kong Chian Faculty of Engineering & Science (LKC FES)
Universiti Tunku Abdul Rahman (UTAR)

Emily Kit Lum Tan
Lee Kong Chian Faculty of Engineering & Science (LKC FES)
Universiti Tunku Abdul Rahman (UTAR)

Shi Yao Lim
Lee Kong Chian Faculty of Engineering & Science (LKC FES)
Universiti Tunku Abdul Rahman (UTAR)

Lee Chuan Ng
Lee Kong Chian Faculty of Engineering & Science (LKC FES)
Universiti Tunku Abdul Rahman (UTAR)

***Abstract*—**To ensure and promise the quality of wood in manufacturing industries, wood surface defect detection is a vital and critical step. However, the traditional method (Human inspection) is time-consuming and it's not precise enough to detect all the defects. In this assignment, we will utilize YOLO to accurately detect and classify the wood defects (crack, dead knot, live knot and marrow) based on the dataset we get from Roboflow. This model will also demonstrate robust real-time detection capabilities and strong generalization to unseen data. We will use YOLO with 3 different versions, YOLOv11, YOLOv10 and YOLOv8. Multiple metrics will be used to evaluate and compare the performance of wood defect detection of these 3 versions. For example, accuracy, precision, recall, F1-score, mAP@0.5, and mAP@0.5:0.95. From the results we get, YOLOv8 had achieved the best overall performance which results in better localization accuracy and class prediction confidence of wood defect detection. Although YOLOv11 demonstrated a higher recall F1-Score than YOLOv8, but still YOLOv8 have better overall performance in all other matrices which proves its robustness indirectly. These findings have proved to us that although there is steady improvement of the YOLO architecture, it might not result in better wood defect detection capabilities. In short, we analyze that YOLOv8 is the current best model for wood defect detection, which reinforces the importance and advantages of utilizing object detection technologies in automated wood defect detection rather than human inspection.

I. Introduction

Traditionally, wood surface defect detection is manually inspected by humans in many wood manufacturing industries. However, this method is significantly a time-consuming method, yet the precision of detection is not promised (Inconsistency). Besides, it also strongly relies on the inspector experience and how sharp his/her eyes to detect the wood defects. This will worsen if the industry is in a high-volume production environment. These reasons had significantly motivated humans to explore real-time, reliable and accurate deep learning models for automated wood defect detection.

Thus, from the manual human inspection to now the deep learning models like YOLO (You Only Look Once) for automated wood defect detection, the driving of technology nowadays has significantly improved the technology of wood defect detection. By using YOLO, it provides us a fast and real-time detection, increases the efficiency and accuracy of detection the quality is promised. Thus, in this assignment, we will explore and try 3 different models of YOLO to determine which is the best and the most suitable model for wood defects detection.

YOLO is based on Convolutional Neural Network (CNN). CNN is a class of deep learning and it's one of the main categories to do tasks like image recognition, image classification, object detection and etc. CNN are excel and manage to detect relevant pattern through layers of convolution and pooling from the digital image or videos. This makes CNN superior in image classification problems like wood defects detection. The ability of networks learns from labeled examples and makes CNN a powerful tool for image recognition and classification (IBM, 2021).

The objective of this report is to construct an object detection model utilizing the latest YOLO architecture and image processing methodologies to accurately detect and classify objects within images. The model should demonstrate robust real-time detection capabilities and strong generalization to unseen data. Thus, we implement and compare 3 different version of YOLO (YOLOv8, YOLOV10 and YOLOv11) based on a same custom wood defect dataset but with different metrics and try to determine the best suitable model who provide the best efficiency and accuracy for our case in order for us to utilize it.

II. RELATED WORK

A. CFIS-YOLO: A Lightweight Multi-Scale Fusion Network for Edge-Deployable Wood Defect Detection

This study proposes CFIS-YOLO, a lightweight object detection model optimized for edge devices. It introduces an enhanced C2f structure, a dynamic feature recombination module, and a novel loss function that incorporates auxiliary bounding boxes and angular constraints. These developments greatly lower computational overhead while improving small object localization and multi-scale feature fusion [1].

The model introduces an improved C2f module that enhances feature extraction capabilities while maintaining a lightweight architecture. Additionally, a FasterBlock module integrated into C2f structure to achieve higher FPS inference speed and facilitate edge device deployment. In order to address the restricted precision of small target identification, the model also incorporates an Inner-SIoU loss that combines internal intersection-over-union and angle awareness to reduce positioning error for micro-defects.

This model achieves 77.5% mAP@0.5 on public wood defect datasets which is a 4-percentage-point gain over the baseline YOLOv10s. On SOPHON BM1684X edge devices, it delivered 135 FPS and reduced power consumption to 17.3% of the original implementation.

B. An Efficient and Accurate Surface Defect Detection Method for Wood Based on Improved YOLOv8

This study builds on improvements to the YOLOv8 model which shows notable performance improvements in managing multi-scale and small-target faults that frequently observed in wood. It also presents an effective and accurate method for detecting surface flaws in wood.

In order to create a multi-scale feature extraction mechanism that can precisely identify and segment objects in images, the DWR module makes it easier for the network to adjust to features at various scales. The convolution kernel's size and shape are dynamically altered by DLKA attention to match different visual attributes。 This could improve the model's capacity to adapt to irregularly shaped and multi-scale targets. The Dynamic-Head Detection Head was developed to improve the object detection head's representational capabilities while preserving computing efficiency. MPDIoU Loss Function Based on Inner Ideas enables the model to learn to predict the precise location of the bounding box, closely aligning it with the actual bounding box of the detected target. It enables the model to perform faster and more effective regression [2].

This model incorporates an attention mechanism and optimization of the loss function to improve the detection of wood defects. The proposed algorithm achieves a high recognition rate, with a mAP of 91.10% and an average detection time of 6 ms. This performance represents a 5.1% improvement in mAP and a reduction of 1 ms in average detection time compared to the original model. However, the detection accuracy of the model for Quartzity defects, one of the common defects in wood, stands at 68.5%, indicating the necessity for further improvements. In short, the experimental results demonstrate that the improved model achieves a 5.5% increase in mAP accuracy compared to the original model and outperforms current mainstream algorithms [2].

C. WD Detector: deep learning-based hybrid sensor design for wood defect detection

This study implements the model using the Xception convolutional neural network (CNN) model which is efficient in capturing intricate patterns in image data. It also classifies the wood defects with twelve different classical machine learning algorithms. This hybrid approach shows it's strengths on deep learning for feature extraction and classical algorithms for classification to improve detection accuracy and computational efficiency.

This proposed model achieved a remarkable accuracy of 99.32% in detecting wood surface defects and highlights the effectiveness of combining deep learning-based feature extraction with classical machine learning classifiers. This demonstrates a significant potential for real-world applications in the woodworking industry where accurate and efficient defect detection is critical for quality control and resource optimization.

The integration of deep learning and classical machine learning in this model offers a promising solution for automated wood defect detection. By achieving high accuracy, the system can potentially reduce reliance on manual inspection and increase the efficiency and consistency in quality assessments. Furthermore, the approach can be adapted to different types of wood defects and different industrial settings which leads it to a versatile tool for the woodworking sector.

III. METHODOLOGY

The dataset used in this project is "Wood Defects Computer Vision Project" dataset which is uploaded in Roboflow one year ago. This dataset has total 14100 images, including 12625 images in training set, 1101 images in valid set and 374 images in test set, providing numerous images of wood surfaces with various type of defects such as cracks, knots, and holes.

All the images in the dataset have been stretched to 640x640 pixels and automatically oriented to consistent direction. Resizing the images can ensure the application of YOLO in different version due to the requirement of consistent input size of images, matching the default image size of 640x640 pixels. Besides, augmentation was applied at the dataset level to produce 3 outputs per training, including flip, 90° rotate and bounding box. Data augmentation is the process of generating new data from existing data to enrich the dataset for training a new model. It is vital to improve the accuracy of predictions, enhance overall performance and provide a better result[4].

In this project, YOLOv8, YOLOv10, and YOLOv11 were selected as the core CNN architectures for training in this project. Each model is built on a standard convolutional block comprising a Conv2d layer, followed by BatchNorm2d and the SiLU (Sigmoid Linear Unit) activation function. SiLU, also known as Swish, helps maintain smooth gradient flow, which is particularly useful in deeper networks [5].

YOLOv8 is an open-source, community-driven model structured into three main components: the backbone, neck, and head. The backbone uses CSPDarknet53, which integrates 53 convolutional layers and Cross-Stage Partial (CSP) connections to improve gradient flow and reduce redundancy. The neck replaces the traditional Feature Pyramid Network (FPN) with the C2f module to merge features from different scales. The head generates predictions for bounding boxes, objectness scores, and class probabilities, which are combined to produce the final detection results [6].

YOLOv10 builds on YOLOv8 with several architectural upgrades for both accuracy and efficiency. The backbone is an enhanced CSPNet with better feature reuse. The core convolutional units remain the same—Conv2d, BatchNorm2d, and SiLU—but with a deeper structure. The neck uses Path Aggregation Network (PAN) layers, using lateral connections to better integrate features across scales. A notable change in YOLOv10 is its dual-head structure: the one-to-many head generates multiple predictions per object during training, while the one-to-one head selects the most accurate prediction for final output[7].

YOLOv11 further evolves the architecture with the integration of C3k2, SPPF, and C2PSA modules[9]. The backbone replaces earlier units with C3k2 blocks—lightweight convolutional units that use a 2×2 kernel for capturing better details. The neck includes the SPPF (Spatial Pyramid Pooling – Fast) module, which combines spatial context using a sequence of pooling layers. In addition, the C2PSA (Convolutional block with Parallel Spatial Attention) module enhances spatial attention mechanisms. The detection head continues to use C3k2 blocks to enhance multi-scale feature processing and deliver more precise object predictions [9].

YOLOv8n, YOLOv10n, and YOLOv11n models were trained using the Ultralytics YOLO framework under the same settings to ensure a fair comparison. The dataset was split into training, validation, and test sets, defined in a data.yaml file. Each model was trained for 8 epochs with an image size (imgsz) of 640, while other parameters like batch size and learning rate used Ultralytics defaults. The training used default batch size and learning rate, with SGD with momentum (0.937), weight decay (0.0005), and an initial learning rate of 0.01. The loss function combined bounding box regression, objectness, and classification losses, automatically balanced during training. Validation mode evaluated performance with metrics like mAP, precision, and F1-score, while predict mode generated visual outputs for test samples [11]. The command is given below:

- yolo task=detect mode=train data=Wood-Defects-7/data.yaml model=yolov8n.pt epochs=8 imgsz=640
- yolo task=detect mode=val model="runs\detect\train6\weights\best.pt" data=Wood-Defects-7/data.yaml
- yolo task=detect mode=predict model="runs\detect\train6\weights\best.pt" conf=0.25 source=Wood-Defects-7\test\images save=True

IV. EXPERIMENTAL RESULTS

After running the YOLO versions 8, 10, and 11 on detecting the wood defect, we get the performance of metrics such as accuracy, precision, recall, F1-score, mAP at IoU 0.5 (mAP@0.5), mAP averaged from IoU 0.5 to 0.95 (mAP@0.5:0.95).

TABLE I. RESULTS OF RUNNING THREE VERSION OF YOLO

YOLO Version	***Precision***	***Recall***	***F1-score***	***mAP@0.5***	***mAP@0.5: 0.95***
YOLOv8	0.6187	0.5857	0.6187	0.5997	0.6187
YOLOv10	0.5657	0.5358	0.5503	0.5551	0.2746
YOLOv11	0.6017	0.6041	0.6034	0.5987	0.2976

Fig. 1. Model's Performance of YOLOv8, YOLOv10 and YOLOv11

From this table, it shows the 5 key performance metrics for each YOLO version after training each version for 8 epochs. For YOLOv8, it achieves the highest precision result (0.6187) and a better result for mAP@0.5 (0.5997) and mAP@0.5:0.95 (0.3008) compared to the other two YOLO versions, which shows that it can correctly predict more wood defects. However, YOLOv11 gets the highest result for its F1-score (0.6034) and recall (0.6041), showing that it can find more real wood defects in this project. In addition, based on this table, we can observe that YOLOv10 does not have any significant results, and even all its results are lower than YOLOv8 and YOLOv11.

Since the object detection will not output the accuracy result, we will analyse the accuracy based on the result of precision, recall and mAp@0.5 together. By observing the table, the YOLOv8 achieves high precision, high recall and high mAP@0.5 result; therefore, it gets the highest effective accuracy of about 60 to 62 percent in detecting the wood defects among these three different YOLO versions.

Besides, it indicates that the only improvement from YOLOv8 to YOLOv11 is recall performance, while the precision and mAP@0.5 performance are not improved but decreased.

V. DISCUSSION

A. Interpretion on Experiment Result and it's implications

The experiment results show that among the three YOLO versions tested, YOLOv8 achieved the highest precision (0.6187), mAP@0.5 (0.5997), and mAP@0.5:0.95 (0.3008). This proves its effectiveness in accurately identifying wood defects with fewer false positives. It also shows generalizability and precision even for small or ambiguous defects. In short, YOLOv8 is the most reliable for scenarios requiring precise defect detection.

On the other hand, YOLOv11 performs well in recall (0.6041) and F1-score (0.6034) which proves its strength in identifying a larger number of real wood defects with a slightly higher false positive rate. This works better in situations when it's essential in quality control or safety inspections where failing to notice a flaw could have dire consequences. Unfortunately, YOLOv10 underperformed across all metrics which shows that it is less effective in both detecting and correctly classifying wood defects.

In summary, the experiment results show that YOLOv8 is the best choice for high-precision applications while YOLOv11 could be optimized further to enhance precision without sacrificing its superior recall. Therefore, YOLOv8 is recommended for industrial use because of its critical accuracy while YOLOv11 may serve as a base for further enhancement in defect detection sensitivity.

B. Strengths and Weaknesses of the Methodology and Model

1) Strengths

We first purpose the preprocessing process before the experiment by standardizing the image dimensions to 640×640 pixels which aligns with YOLO's input requirements to ensure consistency across the dataset. We also apply data augmentation techniques such as flipping, 90° rotation, and bounding box adjustments to enhance the model's ability to generalize by exposing it to varied representations of wood defects.

Furthermore, the exploration of each YOLO version's architecture provides valuable context as the YOLOv8 model implements CSPDarknet53 as its backbone and integrating 53 convolutional layers with Cross-Stage Partial connections to enhance gradient flow and reduce redundancy. Additionally, we also introduce an enhanced CSPNet backbone and a dual-head structure by comprising one-to-many and one-to-one heads to improve prediction accuracy in the YOLOv10 model. Finally, we incorporate advanced modules like C3k2, SPPF, and C2PSA to bolster feature extraction and spatial attention mechanisms in YOLOv11 model.

On top of that, we also provide the performance metrics analysis by evaluating it with various fields such as precision, recall, F1-score, mAP@0.5, and mAP@0.5:0.95. These data fields offer a multifaceted view of model performance. We can tell that the YOLOv8 model has a superior accuracy in defect detection as it achieved the highest data in precision (0.6187), mAP@0.5 (0.5997), and mAP@0.5:0.95 (0.3008). In addition, the YOLOv11 also shows that it has a greater ability to identify actual defects but with a slight trade-off in precision since it achieves the highest score in recall (0.6041) and F1-score (0.6034) by observing the experiment result.

2) Weaknesses

One of the weaknesses we can find during the experiment is the limited training duration. Training each model for only 8 epochs may not suffice for optimal convergence especially for complex architectures like YOLOv10 and YOLOv11. An extended training may potentially improve performance metrics.

Besides that, the methodology does not mention the use of cross-validation techniques which are essential for assessing the models' generalizability and mitigating overfitting. Additionally, we did not analyze the types of errors made by each model such as false positives, false negatives or analyze performance across different defect categories. We should conduct a detailed error analysis that could offer insights into specific areas where models struggle.

C. Challenges Encountered During the Experimentation Process and it's potential improvements.

One of the challenges we faced in the experiment process is the imbalanced detection trade-offs. This is said so when the recall of YOLOv11 improved but it sacrificed the precision. Optimizing for one often negatively impacts the other and it is difficult to achieve a perfect balance. We may try to apply Weighted IoU Loss to help balance the trade-off between precision and recall by penalizing hard-to-detect defects more heavily.

Additionally, we also encounter some difficulties with generalization issues across IoU thresholds. Based on the result, we can see the YOLOv10 is significantly lagged in mAP@0.5:0.95 which reflects the poor adaptability to varying IoU thresholds. This affects its reliability in detecting defects of different shapes and sizes. Thus, we may try to enhance the training dataset with additional augmented samples such as applying different lighting conditions, rotated wood surfaces or variable defect sizes to improve generalization across IoU thresholds.

VI. CONCLUSION

In conclusion, we successfully implemented and compared 3 versions of YOLO object detection models (YOLOV8, YOLOv10, YOLOv11) in our wood defects detection dataset. We analyze that YOLOv8 is the current best model for wood defect detection as it delivered the most consistent and accurate result. Besides, it has a strong performance across all the evaluation metrics. YOLOv11 does have a competitive performance since it slightly surpassed YOLOv8 in recall and F1-Score, but YOLOv8 still has the better overall performance as YOLOv8 has a good balance in other evaluation matrices like precision and mAP@0.5. These findings have proved to us that although there is steady improvement of the YOLO architecture, it might not result in better localization accuracy and class prediction confidence of wood defect detection.

This analysis shows the importance and advantages of utilizing object detection technologies in automated wood defect detection. By utilizing the CNN- based YOLO model, the efficiency and accuracy of our defect detection and classification is highly promising as compared with traditional human visualization.

One of the key lessons we learned is that it's important to evaluate models not just rely on one single metric. This is said so because by considering various matrices, we can only know which models have better overall performance and only determine which models are the best for us. Otherwise, if we only rely on single matrices like recall or F1-score, we might think that YOLOv11 is the best model for our case. In future, we think that we may explore further optimization of training parameters like augmentation strategies and regularization techniques which could enhance the model's performance.

Last but not least, we would like to express our gratitude to all those who keep support us throughout this project. Special thanks to all our group members for their efforts and collaborations. We would also like to acknowledge Roboflow for providing us the dataset and Ultralytics YOLO framework for enabling our training and evaluation. We would also Finally, we would like to thank our lecturer for his guidance, support and the computing resources provided throughout the project.

VII. CONTRIBUTION

No	Member	Roles(s)	Responsibilities	Unique Additions
1	Emily Tan Kit Lum	-Dataset preparation -Preprocessing	-Collect the dataset of wood defects from the website of Roboflow -Split the data of wood defects for training, validation, and testing.	-Execute the data cleaning process
2	Lim Shi Yao	-Model training -Algorithm development	-Implements the YOLOv8 and YOLOv10 for wood detection -Customize the model setting	-Deeply introduce various data processing
3	Ng Kai Xin	-Model training -Analysis performance	-Implements the YOLOv11 for wood detection -Analyse the key performance metrics for each model such as precision, recall, F1-score, mAP@0.5 and mAP@0.5:0.95.	-Provide a table and graph that include each model's performance metric for comparison.
4	Ng Lee Chuan	-Deployment -Model optimization	-Deploy each trained model in local testing -Optimized the models available by adjusting the learning rates	-Make a clear and accurate introduction and conclusion to summarize our project

REFERENCES

[1] Arxiv.org. (2019). "CFIS-YOLO: A Lightweight Multi-Scale Fusion Network for Edge-Deployable Wood Defect Detection." [online] Available: https://arxiv.org/html/2504.11305v1., in press

[2] Wang, R., Liang, F., Wang, B., Zhang, G., Chen, Y. and Mou, X. (2024). "An Efficient and Accurate Surface Defect Detection Method for Wood Based on Improved YOLOv8." Forests, [online]. Available: doi:https://doi.org/10.3390/f15071176., in press.

[3] Kılıç, K., Kılıç, K., Doğru, Özcan,U. "WD Detector: deep learning-based hybrid sensor design for wood defect detection." Eur. J. Wood Prod. 83, 50 (2025). Available:https://doi.org/10.1007/s00107-025-02211-5

[4] Amazon Web Services, Inc. (n.d.). "*What is Data Augmentation? - Data Augmentation Techniques Explained - AWS*." [online] Available: https://aws.amazon.com/what-is/data-augmentation/.

[5] Timilsina, A. (2024). "*YOLOv8 Architecture Explained!*" [online] Medium. Available: https://abintimilsina.medium.com/yolov8-architecture-explained-a5e90a560ce5.

[6] Torres, J. (2024). "*YOLOv8 Architecture; Deep Dive into its Architecture -Yolov8*." [online] YOLOv8. Available: https://yolov8.org/yolov8-architecture/#Key_Innovations_in_YOLOv8

[7] Consiglio, A. (2024). "*YOLOv10 — Breaking Speed Barriers with NMS-Free Detection (with code)*." [online] Medium. Available: https://medium.com/@antonioconsiglio/yolov10-breaking-speed-barriers-with-nms-free-detection-with-code-d1e39a503823

[8] Ultralytics.com. (2024). "*YOLOv10*." [online]. Available: https://docs.ultralytics.com/models/yolov10/#overview

[9] Khanam.R, Muhammad.H. (2015). "YOLOv11: An Overview of the Key Architectural Enhancements." [online] Arxiv.org. Available: https://arxiv.org/html/2410.17725v1#S4., in press

[10] Ultralytics.com. (2025). "Activation Function - Discover the role of activation functions in neural networks, their types, and real-world applications in AI and machine learning." [online]. Available: https://www.ultralytics.com/glossary/activation-function

Ultralytics.cp, (n.d.). *"Ultralytics YOLO Hyperparameter Tuning Guide"*. [online]. Available: https://docs.ultralytics.com/guides/hyperparameter-tuning/.