Hybrid Detection and Segmentation of Wood Defects

Brandon Ramos-Lopez[1], Jesus Hernandez-Barragan[2], Javier Gomez-Avila[2], Michel Lopez-Franco[1], and Carlos Lopez-Franco[1]

[1] Departamento de Ciencias Computacionales, Centro Universitario de Ciencias Exactas e Ingenierías, Universidad de Guadalajara, Blvd. Marcelino García Barragán 1421, Guadalajara C.P. 44430, Jalisco, Mexico
[2] Departamento de Innovación Basada en la Información y el Conocimiento, Centro Universitario de Ciencias Exactas e Ingenierías, Universidad de Guadalajara, Blvd. Marcelino García Barragán 1421, Guadalajara C.P. 44430, Jalisco, Mexico

Abstract. This paper presents a hybrid approach for detecting and segmenting wood defects, designed for deployment on low-resource embedded devices. The approach combines the efficiency of YOLO (You Only Look Once) for defect detection with traditional image processing techniques for precise segmentation. The use of traditional methods helps reduce the computational burden, making the approach more suitable for resource-constrained environments. This methodology leverages the strengths of both approaches while mitigating their individual limitations.

Keywords: Computer Vision · Image Processing · YOLO (You Only Look Once) · Wood Defect Detection · Image Segmentation · Industrial Quality Control

1 Introduction

The wood industry constantly faces challenges in quality control due to the complex and variable nature of wood defects. Natural defects such as knots, cracks, and fiber deviations during tree growth. Additionally, processing defects occur during wood transformation, including cutting problems, inadequate drying, and incorrect classification, as shown in Fig 1.

https://doi.org/10.61728/AE20255183

Fig. 1. Examples of possible defects in a tangential wood cut [2].

All these issues can significantly compromise the mechanical strength of the material. These defects, along with environmental factors such as humidity and temperature, can cause structural deformations, reduced mechanical resistance, and increased susceptibility to biological deterioration [1].

Early and accurate detection of these defects is crucial to maintaining high-quality production standards. However, traditional manual visual inspection methods are prone to errors, time-consuming, and inconsistent. This has driven the need to develop automated inspection systems that can quickly and reliably identify and characterize defects.

Currently, multiple solutions for defect detection in the industry have emerged, many of these solutions with custom designs of convolutional neural networks [4]. Architectures such as YOLO (You Only Look Once) have emerged as efficient solutions for identifying objects or, in this case, wood defects. YOLO stands out for its ability to process images in real-time and its high detection accuracy, requiring only bounding box annotations. However, precise defect segmentation requires more complex manual labeling, which can be costly and time-consuming. Neural networks may also be computationally expensive and complex, which translates into a difficult industrial application.

In Table Fig 1 we can see the size and speed of a YOLO11N detection network compared to three segmentation networks. It is clearly evident that just the process of training a network will be more costly, not to mention the human work involved in the segmentation training.

Table 1. Comparative of segmentation and detection models [5].

Model	Parameters (M)	FLOPs (B)	CPU speed (ms)
YOLO11n	2.6	6.5	56.1
YOLO11n-seg	2.9	10.4	65.9
MobileSAM	10.1	-	98543
FastSAM	11.8	-	140

This is where classical computer vision methods, such as Otsu thresholding method, offer a practical and efficient alternative. These techniques, although simpler, can provide robust results when applied to specific regions identified by YOLO. Otsu method, in particular, is effective in automatically determining the optimal threshold to separate the defect from the background based on the image's intensity distribution, without requiring prior training or exhaustive annotations.

This hybrid approach, combines YOLO capabilities for initial detection with classical segmentation methods, represents a practical solution that balances precision with computational efficiency and ease of implementation. Additionally, it significantly reduces the need for annotated data and simplifies the system's development and maintenance process.

2 Defect detection

The process begins by extracting sub-images from the bounding boxes generated by YOLO. Each sub-image represents a specific region where a potential defect has been detected, reducing background noise and enabling local thresholding. The complete process is described in Fig 2.

In the first phase, an image is passed to the YOLO method which detects and assigns a class to all the defects on the image. Then in the second phase each detected defect is passed to the Otsu method. Finally, all the results of the Otsu method are combined into a single image that represents the segmented image.

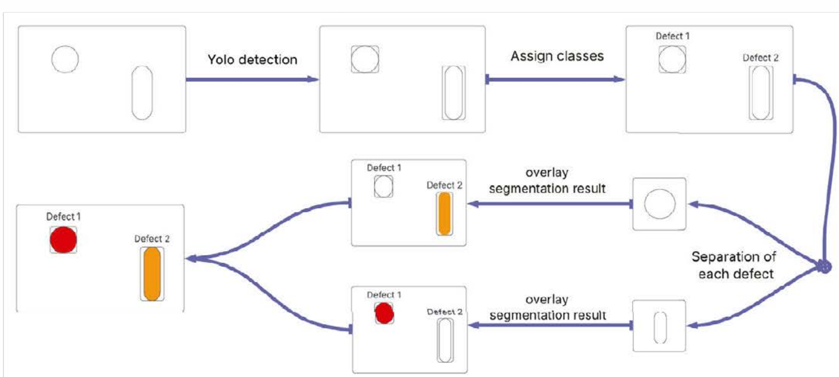


Fig. 2. Defect detection Diagram

To observe the results of the process, the Fig 2. shows an image of dataset with their detection make by YOLO. Finally Fig 3 shows an image after the segmentation process, as described in the previous diagram.

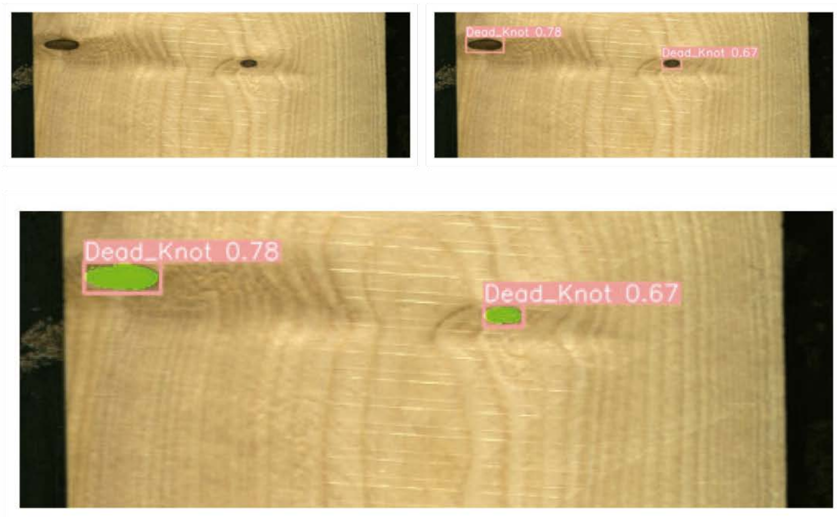


Fig. 3. Example of Defect detection including segmentation by Otsu

3 Experimental Results

The dataset [6] used in this work contains 20,276 images with annotations in COCO format. Due to computational limitations and the need for optimized training, a stratified sampling was performed, selecting 2,000 images for training and 100 for validation. This reduction maintained a balance between data representation and training efficiency.

The process for reducing the data set as shown in Fig 4, involved converting the COCO annotations to YOLO format, ensuring correct bounding box coordinate translation and label integrity. The images were converted from BMP to JPG format to reduce storage space, and their dimensions were resized from 2800x1027 pixels to 640x256 pixels, maintaining the original aspect ratio while optimizing computational efficiency.

In our case we have chosen YOLO11n, wich in the experiments has shown a balanced efficiency and accuracy. A default learning rate of 0.01 was used with the Adam optimizer, a batch size of 4 images, and training over 50 epochs.

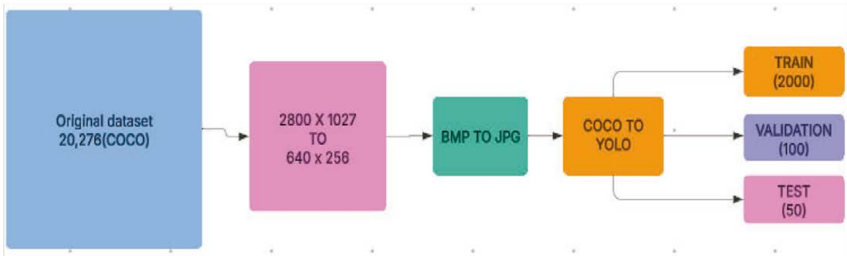


Fig. 4. Weight reduction for dataset

The implementation has been carried out on the Ultralytics framework, which provides an efficient interface for YOLO [7]. The data was organized following the structure required by the framework, with separate folders for training and validation, and labels in YOLO format. The model configuration was done using a YAML file specifying the data path and defect classes.

The training was initialized using the pre-trained YOLOn weights of COCO (Common Objects in Context) dataset to improve convergence. During inference, the framework facilitates image processing and prediction retrieval through the predict method, which provides the coordinates of the bounding boxes, confidence scores, and class labels needed for subsequent segmentation.

The second phase of the proposed approach focus on precise segmentation of the defects detected using YOLO. This process avoids complex neural networks and instead relies on sequential classical processing steps.

For defect segmentation, Otsu thresholding method [3] has been choosen. This technique determines an optimal threshold by minimizing intra-class variance. However, during experiments, the threshold calculated by Otsu method was often lower than required for effective class separation. To address this, an adjustment factor of 1.2 was introduced to the original threshold, enhancing segmentation precision to have a result like Fig 5 .

The final segmentation is applied to each sub-image, generating a binary mask that separates the defect from the background.

To visualize the results we can observe Fig 6 and Fig 7, in these examples a color channel composition technique was implemented. The original image is decomposed into its three RGB channels. The binary mask from Otsu thresholding modifies only the green channel, keeping the red and blue channels intact. This visualization technique clearly marks defects in green over the original image, achieving results comparable to more complex segmentation methods.

Experimental results shown in Fig 8, evaluated using the Intersection over Union (IoU) index against manual segmentation maps, demonstrate the feasibility and effectiveness of this combined approach.

To visualize efficiency, 4 tests were performed for each defect found; they are shown in Table 2.

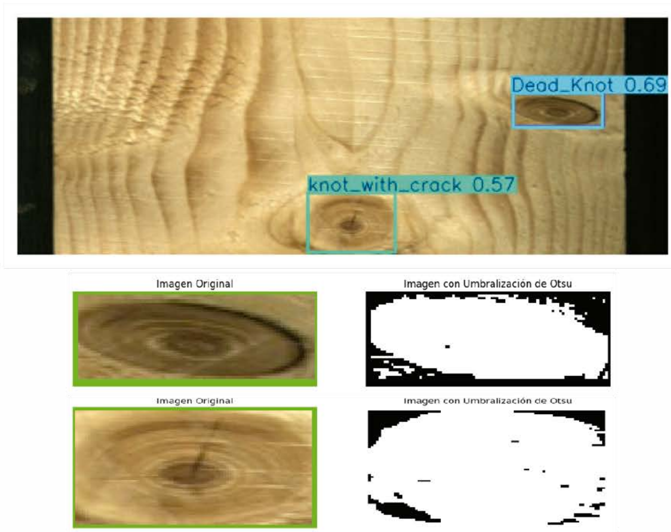


Fig. 5. Detection by YOLO found 2 defects, then the segmentation of both defects is obtained using the Otsu method.

Fig. 6. Defect segmentation example.

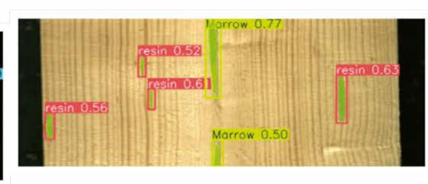


Fig. 7. Multiple defect segmentation example.

Table 2. This table shows the mean of the metrics

IoU	Accuracy	Precision	Recall
0.74084	0.78845	0.88817	0.82385

These results suggest that your segmentation algorithm performs reasonably well, considering the low resources required by the Otsu method. Additionally, it is important to note that the segmentation maps provided for comparison are somewhat poor in pixel-by-pixel labeling. It is easy to see through visual inspection that, for the most part, the results of the proposed method are better than the existing manual segmentation, which clearly impacts the metric results, lowering their score, as seen in the example shown in Fig 3 where the image below is the original semantic map, it is clear that the method had a better result, compromising the result of the metrics.

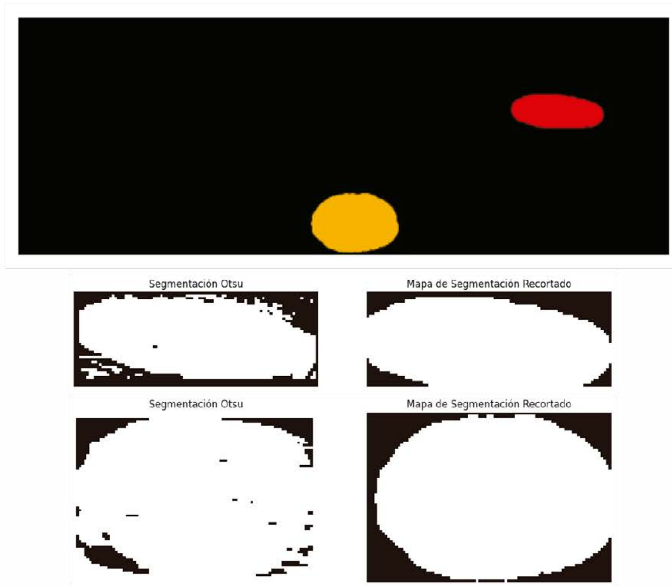
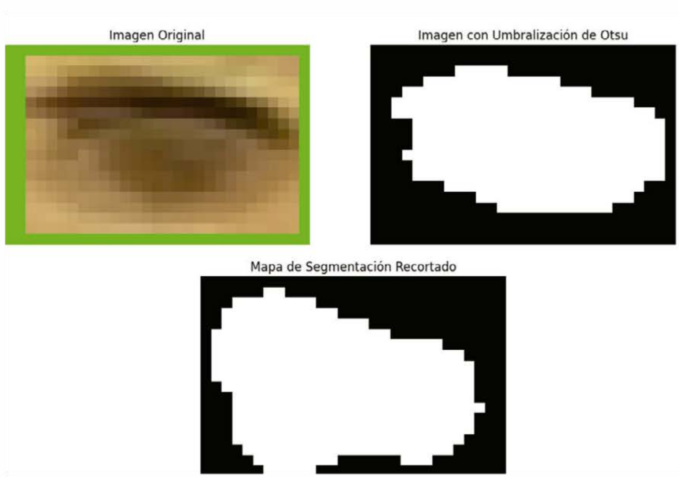


Fig. 8. original segmentation map and both defects compared by IoU



4 Conclusion

The proposed methodology balances the advantages of both approaches: the speed and robustness of neural networks for detection and the pixel-level precision of traditional segmentation techniques. This balance is particularly valuable in industrial environments where speed, accuracy, and ease of implementation in compact and simple devices are key factors. Identified limitations include the dependency on accurate initial detection by YOLO for successful subsequent segmentation. In addition, classical segmentation techniques may require initial parameter tuning based on real-world experiments.

For future work, we propose to implement auto-tuning techniques for segmentation parameters to enhance adaptability. Developing specific metrics to evaluate each type of wood defect, considering both geometric precision and industrial relevance. Optimizing the processing pipeline for embedded systems and real-time applications. The results obtained suggest that this hybrid approach offers a viable alternative for automated quality inspection in the wood industry, providing an effective balance between accuracy, speed, and computational efficiency.

References

1. Daza, N.R.L., Abadía, J.A.M.: Técnicas de Control de Calidad en la Madera Tomo 1. Ph.D. thesis, Corporación Universitaria Autónoma de Occidente, Ingeniería Industrial (1985), https://red.uao.edu.co/server/api/core/bitstreams/58fb9880-cd0b-4d42-9626-c5a7d52bd441/content
2. Ninja, D.: Wood defect detection dataset (2024), https://datasetninja.com/wood-defect-detection
3. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics **9**(1), 62–66 (1979). https://doi.org/10.1109/TSMC.1979.4310076
4. Tabernik, D., Šela, S., Skvarč, J., Skočaj, D.: Segmentation-based deep-learning approach for surface-defect detection. arXiv preprint arXiv:1903.08536v3 (2019), https://arxiv.org/pdf/1903.08536v3
5. Ultralytics: Models - ultralytics documentation (2024), https://docs.ultralytics.com/models/, accessed: 2025-02-21
6. Unknown: Wood defect detection dataset (2021). https://doi.org/10.5281/zenodo.4694695, https://zenodo.org/records/4694695, download Dataset
7. Varghese, R., M., S.: Yolov8: A novel object detection algorithm with enhanced performance and robustness. In: 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS). pp. 1–6 (2024). https://doi.org/10.1109/ADICS58448.2024.10533619