NUST

NATIONAL UNIVERSITY
OF SCIENCES & TECHNOLOGY

Project Report

Efficient Timber Surface Defect Detection Using a Compact YOLO Architecture for Resource-Constrained Devices

Submitted To:
Dr. Hashir Moheed Kiyani

Submitted By:

Sara Adnan Ghori (411228)
Umar Farooq (406481)

Contents

Table of Tables

Table of Figures

Efficient Timber Surface Defect Detection Using a Compact YOLO Architecture for Resource-Constrained Devices

Abstract

Timber fault detection is an important step towards structural safety, material usage and automation of industries. Although recent deep-learning detectors especially YOLO-based models have gained a significant accuracy improvement, the detectors tend to be based on heavy backbones, attention, and multiscale fusion modules which restrict their application in real-time or embedded edge computing. Lightweight versions such as efficient ones would often have a hard time with small, irregular, or texture-occluded defects typical of timber.

The current study will overcome these shortcomings by creating a small YOLO architecture that is optimized to detect timber defects and deploy edges. With an augmented and balanced dataset that we use to train our model, we strike a balance between producing an efficient computationally and competitive model and accurate predictions. This is supported by a carefully chosen to set of architectural elements, optimized backbone depth, a reduced-parameter neck sampling and performance maintaining detection heads- to deliver 2.23 GMACs (4.46 GFLOPs) with good performance on challenging defects. The findings indicate that a well considered lightweight design can deliver viable performance without having overly high-computational overheads that are in current models of state-of-the-art.

1. Introduction

Structural integrity and industrial wood-processing pipelines are based on timber grading and fault detection. Standard inspection is manual which may contain errors and mostly this mode of inspection relies on operator capacity and tiredness. Deep learning-based visual inspection, especially the YOLO-like object detectors, has demonstrated outstanding potential, as it allows large-scale inferences in real time, as well as a high degree of accuracy in a variety of visual settings.

Timber flaws detection, nevertheless, does not offer easy tasks:

1. Wood textures, which are under high variation, depending on the grain pattern and the light.

2. Little and random flaws, which are also sometimes half-closed or blended with natural forms.
3. Generalization gaps Model trained on mixed-species datasets fail on defect-specific timber structures.
4. Bulky computation Since most high-performing models employ heavy attention modules, large backbones, or multi-scale processing and are thus unsuitable in real-time or edge deployment.

In a bid to address these shortcomings, the given project is coming up with a lightweight, timber-based YOLO architecture that is more edge-oriented and can still achieve high detection performance.

2. Literature Review

2.1 Overview of Related Research

In the recent literature, one can see the fast progress of the models of wood and structural defects detection. Nevertheless, the majority of high-performance frameworks (i.e. SGN-YOLO, WDNET-YOLO, SiM-YOLO, CWB-YOLO) introduce advanced elements-attention blocks, multi-branch fusions, large backbones to enhance precision but with much higher inference costs. Lightweight methods usually trade accuracy at the cost of fine or coarse defects.

Key findings:

- **SGN-YOLO** integrates a Semi-Global Network backbone achieving 86.4% mAP, but remains heavy for edge devices. [1]
- **CWB-YOLO** introduces CondConv and BiFormer attention, improving generalization at the cost of high inference latency. [2]
- **WDNET-YOLO** uses RepVGG, ECA attention, and adaptive upsampling, achieving strong performance but still computationally intensive. [3]
- **FDD-YOLO**, **SiM-YOLO**, and **BPN-YOLO** enhance small-defect detection with specialized fusion and attention modules but remain unsuitable for embedded deployment. [4]
- Broader surveys emphasize persistent gaps: small-defect localization, high computational cost, and limited timber-specific datasets. [5]

Identified Gaps

1. Excessive computational power requirement of state-of-the-art models.
2. Weak ultra-light models in detecting defects.

3. Scarcity of timber-specific detection models. [5]
4. Limited work addressing real-time resource-constrained **edge devices**.
5. General wood defects model not trained on timber. [5]

Research Opportunity

A lightweight YOLO architecture **specialized for timber defect detection** optimized for **resource-constrained and edge devices**.

3. Dataset Preparation

3.1 Base Dataset

The basic dataset used was **Roboflow "defects-in-timber"** [6] **dataset**, with four classes of cracks:

1. Crack
2. Dead_Knot
3. Live_Knot
4. Knot_with_Crack

In the base dataset, classes were imbalanced, especially for Cracks and Knot_with_Cracks defects. TO deal with this issue we did Data Augmentation.

3.2 Data Augmentation

To create a balanced dataset, we applied augmentation to 2 classes:

- **Classes augmented:** 0 (Crack), 3 (knot_with_crack)
- **Augmentation library:** Albumentations
- **Transformations applied:**
  - Horizontal Flip
  - Vertical Flip
  - Rotate
  - Shift Scale Rotate
  - Bounding-box aware augmentation

3.3 Final Dataset Statistics (After Augmentation)

Table 1: Dataset Class Distribution

Class	Train		Validation		Test	
Crack (0)	2628	3246	365	454	254	307
Dead Knot (1)	2772	3897	376	510	263	357
Live Knot (2)	3431	5431	488	778	305	506
Knot with Crack (3)	2866	3445	406	494	269	332

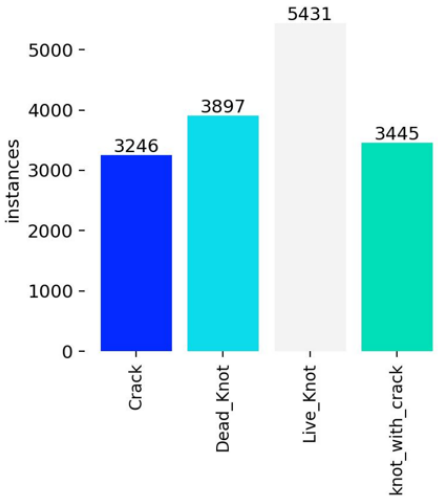


Figure 1: Class Distribution

4. Proposed Method / Model Architecture

The proposed model is a **lightweight, high-performance object detector** tailored for timber defect detection. It combines **efficiency**, **multi-scale feature extraction**, and **context-aware fusion**, making it suitable for detecting cracks and knots of varying sizes while being deployable on edge devices.

At a **high-level**, the architecture consists of three main components (see **Figure 2**):

1. **Backbone**: Enhanced MobileNetV3 Small with multi-scale refinement and attention for feature extraction.
2. **Neck**: Light FPN-PAN style network with attention, SPPF, and a lightweight transformer for multi-scale feature fusion.
3. **Detection Head**: YOLOv8-style anchor-free predictor producing multi-scale predictions.

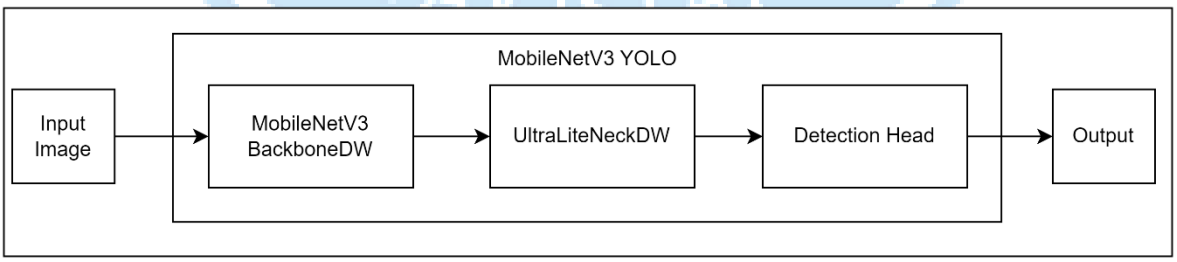


Figure 2: High level Model Architecture

4.1 Backbone: Enhanced MobileNetV3 with Multi-Scale Refinement

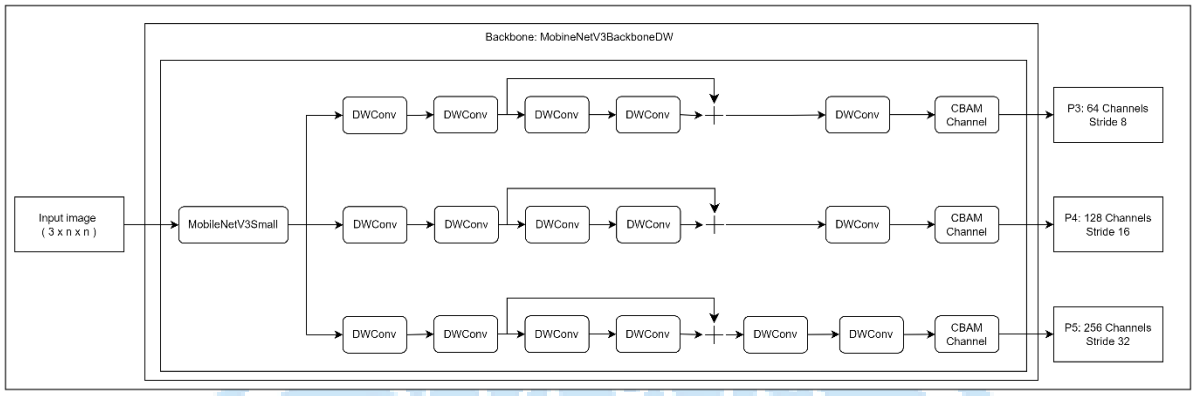


Figure 3: MobileNetV3Backbone Architecture

The backbone is responsible for extraction hierarchical feature maps at 3 stride levels that are **8, 16, and 32**, corresponding to:

- **P3 (64 channels)**: fine details for small cracks
- **P4 (128 channels)**: mid-level defect texture and shape
- **P5 (256 channels)**: high-level semantic context for large knots

The backbone start with a **MobileNetV3 Small** variant, because of its efficiency and easy deployment on edge devices. However, further layers were added to improve the performance and better feature extraction.

4.1.1 Depthwise Separable Convolution Blocks (DWConvCustom)

Depthwise separable convolutions are applied extensively in all three stages to keep the model lightweight while expanding representational capacity. They reduce computation by factorizing convolution into:

- **Depthwise 3×3 convolution** (per-channel filtering)
- **Pointwise 1×1 convolution** (channel mixing)

This allows deeper stacks of feature refinement without increasing FLOPs.

4.1.2 CBAM-Based Attention (Channel + Spatial)

Each scale output (P3, P4, P5) is passed through an enhanced **CBAM attention module** to strengthen important structural cues present in wood defects.

- **Channel attention** highlights discriminative channels useful for detecting cracks and knots.
- **Spatial attention** emphasizes defect-prone regions while suppressing background texture.

This improves robustness in cluttered wooden surfaces with grain variations.

4.1.3 Multi-Stage Refinement Blocks

Unlike standard MobileNetV3, each feature scale includes **4–6 sequential refinement blocks**, forming deeper processing pipelines:

- P3 includes 5 DWConv stacks (critical for small cracks)
- P4 includes 5 DWConv stacks (mid-level context + mid scale cracks)
- P5 includes 6 DWConv stacks (largest receptive field)

These refinements significantly enhance feature quality before fusion.

4.1.4 Output Feature Maps

The backbone outputs:

- P3: 64 channels, stride 8
- P4: 128 channels, stride 16
- P5: 256 channels, stride 32

These are passed into the neck for multi-scale fusion.

4.2 Neck: UltraLite Feature Fusion Network

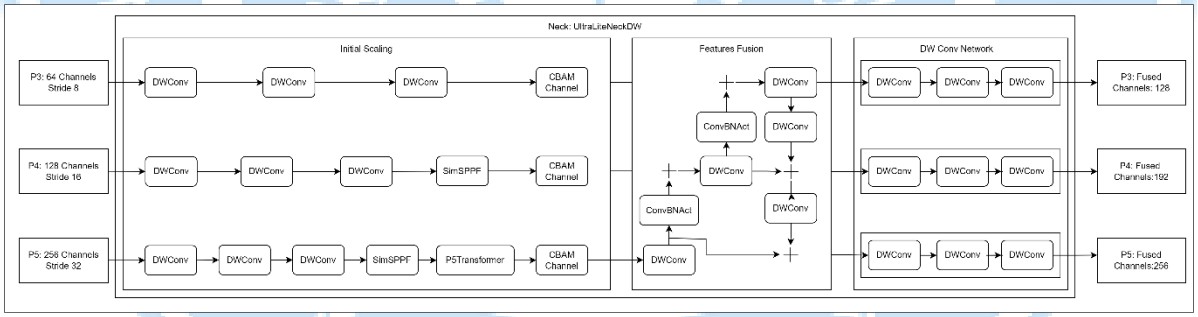


Figure 4: UltraLiteNeck Architecture

The neck is responsible for the aggregation of multi-scale features and production of three refined outputs with the following number of channels:

- P3: 128
- P4: 192
- P5: 256

The neck follows a lightweight **FPN + PAN**-like design but optimized specifically for speed and low memory.

4.2.1 Top-Down Path (FPN)

The high-level P5 features are up-sampled and fused with P4, followed by further fusion with P3 strengthening the fine-resolution maps with contextual information, improving detection of tiny crack structures.

4.2.2 Bottom-Up Path (PAN)

After the top-down fusion, selected maps are again down sampled and merged upward enhancing spatial flow and preserving multi-scale feature richness.

4.2.3 Transformer Block for P5

A lightweight **P5 Transformer** is added for improvement in long-range context modelling with minimal computational cost. It includes:

- 1×1 projection
- LayerNorm
- 2 transformer encoder layers
- 1×1 reconstruction layer

This module benefits detection of large, irregular knots.

4.2.4 SimSPPF Module

A simplified **Spatial Pyramid Pooling – Fast (SPPF)** block is also present in the neck to provide effective multi-scale receptive fields for better global context.

4.3 Detection Head: YOLO-Style Anchor-Free Predictor

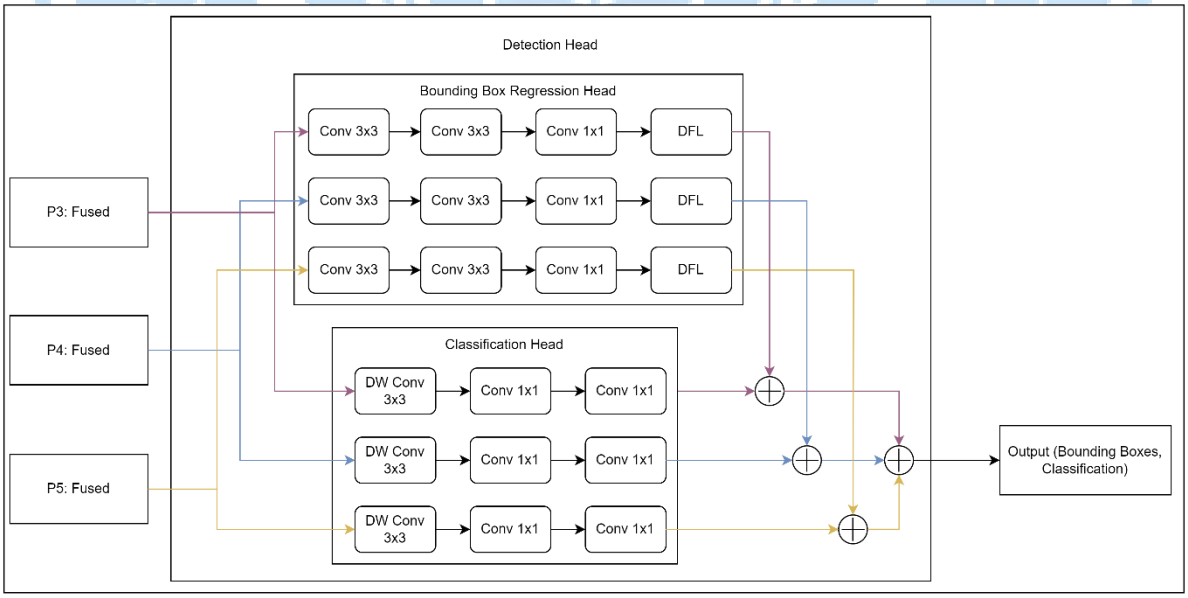


Figure 5: Detection Head Architecture

The head is a **YOLOv8 Detect module**, which receives the multi-scale fused features as input as shown below:

- P3' → 128 channels
- P4' → 192 channels
- P5' → 256 channels

4.3.1 Anchor-Free Decoding

The head outputs for each scale as:

- bounding box offsets,
- objectness score,
- per-class probabilities.

Anchor-free design simplifies computation and dramatically improves performance on irregular-shaped wood defects.

4.3.2 Multi-Scale Predictions

The head produces three prediction maps corresponding to each three-stride level:

- 8 (small defects)
- 16 (medium knots)
- 32 (large knots / crack clusters)

This ensures balanced detection capability across defect types and sizes.

4.4 Summary of Architectural Significance

Component	**Purpose / Significance**
MobileNetV3 Backbone	Efficient multi-scale feature extraction for small to large defects
DWConv stacks	Lightweight feature refinement, essential for subtle defect detection
CBAM Attention	Highlights discriminative features, suppresses background noise
UltraLite Neck	Multi-scale fusion improves feature representation and localization
Transformer on P5	Adds long-range context awareness for large knots
YOLO Head	Fast, anchor-free detection optimized for edge deployment

Table 2: Architectural Components Significance

5. Model Training

The proposed MobileNetV3-YOLO model was trained on the **timber defect dataset** for **robust defect detection**, including cracks, live knots, dead knots, and knots with cracks. The training pipeline is designed to optimize convergence speed, model generalization, and detection accuracy on edge devices.

5.1 Training Setup

- **Dataset:** defects-in-timber (4 classes: Crack, Dead_Knot, Live_Knot, Knot_with_Crack)
- **Input Image Size:** 640 × 640
- **Batch Size:** 16
- **Epochs:** 200
- **Optimizer:** SGD with **momentum 0.937** and **weight decay 5e-4**
- **Learning Rate:** initial lr = 0.01, cosine decay to lr_final = 0.01 × 0.01
- **Warmup:** 5 epochs
- **Early Stopping:** Patience = 25 epochs
- **Workers:** 4
- **Device:** NVIDIA GPU 0

Equation for learning rate schedule:

$$lr_t = lr_0 \times 0.5\left(1 + \cos\left(\pi \frac{t}{T}\right)\right)$$

Where t = current epoch and T = total epochs.

5.2 Loss Functions

The model optimizes three components simultaneously:

Loss Name	Full Form	Description
Box Loss	**Complete Intersection over Union Loss (CIoU Loss)**	Bounding box regression accuracy by jointly optimizing overlap area, centre distance, and aspect ratio consistency between predicted and ground-truth boxes.
Classification Loss	**Binary Cross-Entropy Loss (BCE Loss)**	Classification error for multi-class object detection by independent sigmoid activations of each class.
Distribution Focal Loss	**Distribution Focal Loss (DFL)**	Improves fine-grained bounding box localization by modelling bounding box

		offsets as discrete probability distributions rather than single scalar values.

Table 3: Loss Functions

Total loss: $\mathcal{L}_{total} = \mathcal{L}_{box} + \mathcal{L}_{cls} + \mathcal{L}_{dfl}$

5.3 Training Metrics

Following metrics were monitored during training:

- **Precision (P)** – fraction of correctly predicted defects
- **Recall (R)** – fraction of ground truth defects detected
- **mAP@50 (B)** – mean Average Precision at 0.5 IoU
- **mAP@50-95 (B)** – mean Average Precision over 0.5–0.95 IoU

The subscript (B) indicates batch-wise metrics.

5.4 Training Curves

The model exhibited **steady convergence**, with all three losses decreasing and metrics improving progressively.

5.4.1 Loss Curves

Figure 6-9: Training and validation losses over the first 114 epochs

- **Box Loss** decreased from 4.05 → 1.41 (train), 3.27 → 1.47 (val)
- **Classification Loss** decreased from 4.68 → 0.97 (train), 2.99 → 0.94 (val)
- **DFL Loss** decreased from 3.62 → 1.36 (train), 3.18 → 1.49 (val)

Observation: The **loss curves** show smooth convergence without overfitting. Early epochs exhibit rapid reduction due to **warmup learning rate**.

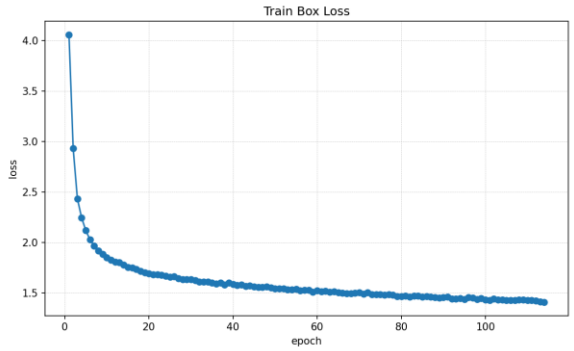


Figure 6: Training Box Loss

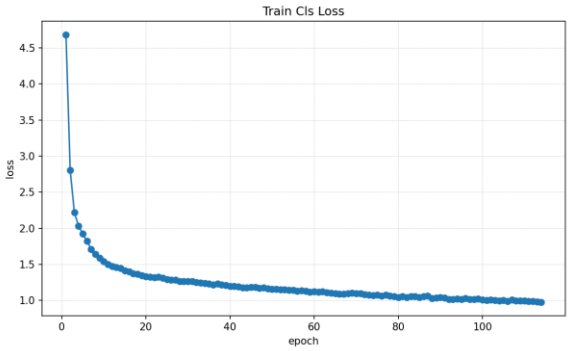


Figure 7: Training Classification Loss

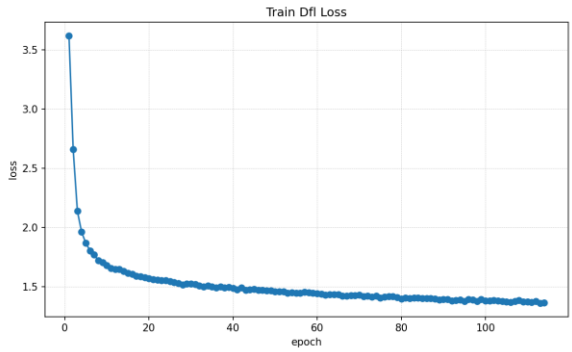


Figure 8: Training DFL Loss

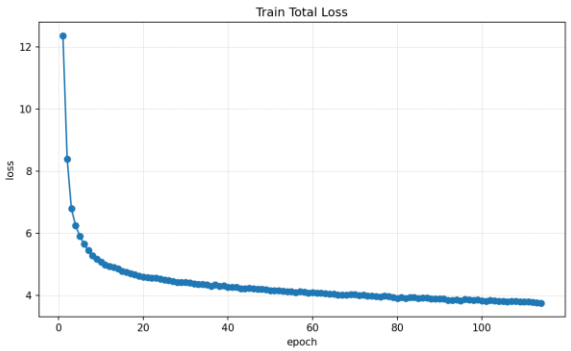


Figure 9: Total Training Loss

5.4.2 Precision, Recall, and mAP Curves

Performance metrics over 114 epochs

- **Precision (P)**: 0.38 → 0.76
- **Recall (R)**: 0.16 → 0.77
- **mAP@50 (B)**: 0.07 → 0.78
- **mAP@50-95 (B)**: 0.02 → 0.46

Observation: Both **precision and recall increase steadily**, indicating that the model detects defects accurately while minimizing false positives. The **mAP curves** confirm strong generalization on validation data.

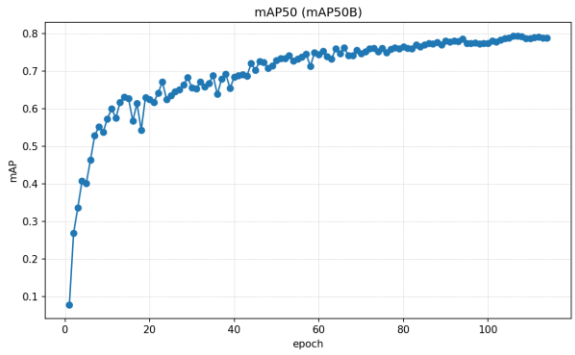


Figure 10: mAP50 Training Curve over Epochs

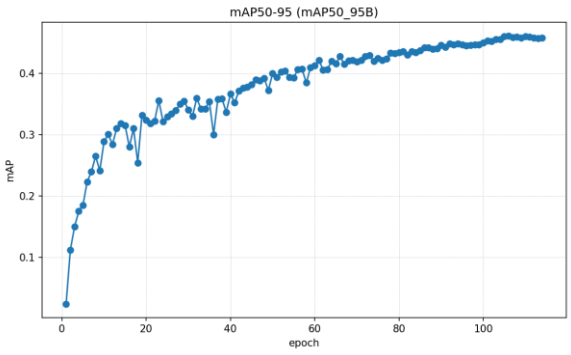


Figure 11: mAP50-95 Training Curve over Epochs

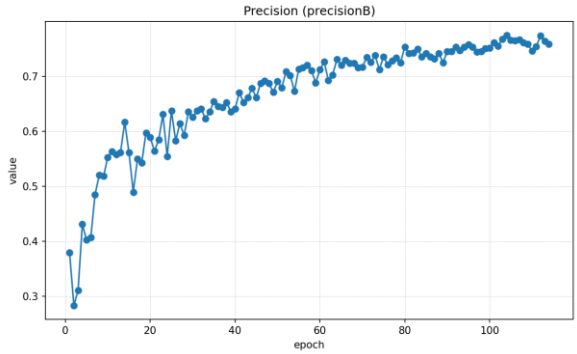


Figure 12: Precision Curve over Epochs

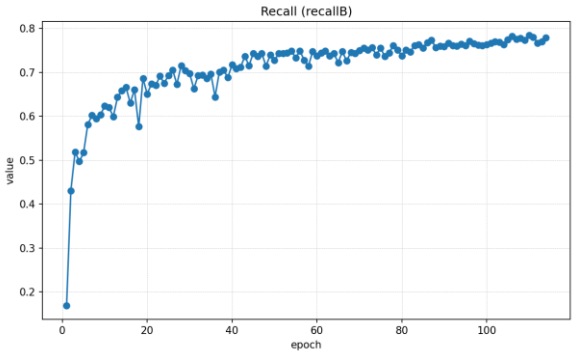


Figure 13: Recall Curve over Epochs

5.4.3 Learning Rate Schedule

Figure 14,15,16: Cosine decay of learning rate with warmup

- Initial **warmup** (first 5 epochs) allows stable gradient updates.
- Cosine decay ensures **fine-tuning** in later epochs.

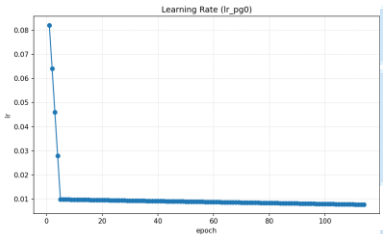


Figure 14: Learning rate (lrpg_0) over Epochs

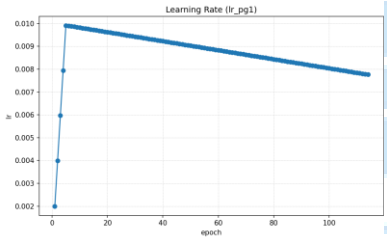


Figure 15: Learning rate (lrpg_1) over Epochs

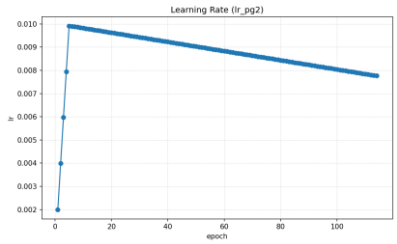


Figure 16: Learning rate (lrpg_2) over Epochs

5.5 Observations

1. **Rapid initial convergence** due to warmup and SGD momentum.
2. **Gradual improvement of mAP@50-95**, showing better localization across defect scales.
3. **No overfitting observed**, validation loss tracks training loss closely.
4. **Stable late-stage training** (epoch 90–114) indicates sufficient model capacity without overfitting.

6. Results and Findings

The suggested MobileNetsV3-YOLO was brought to the timber defect dataset to detect defects in the structure, which may be cracks, live knots, dead knots, and cracks with knots. The training pipeline is to be optimized with respect to convergence speed, model generalization and detection accuracy on edge devices.

6.1 Model Complexity and Efficiency

The model is small but powerful and includes 4.24 million parameters and 2.23 GMac or 4.46 GFLOPs showing that it is much less computationally complex than traditional YOLO-based detectors, suitable to be deployed to edge devices with limited resources.

The model requires an average inference time of 14.9 ms/image, preprocessing and postprocessing times of 1.0 ms and 0.8 ms, respectively. Such timings affirm applicability at the real-time in industrial inspection situations.

6.2 Validation Performance

The first test performed was on the validation dataset. Per-class metrics are displayed in Table 6.1, and they indicate good performance in all timber defect types.

Class	Images	Instances	Precision	Recall	mAP@50	mAP@50-95
Crack	365	454	0.680	0.756	0.725	0.368
Dead_Knot	376	510	0.690	0.735	0.714	0.388
Live_Knot	488	778	0.794	0.775	0.808	0.421
knot_with_crack	406	494	0.861	0.891	0.927	0.674
All Classes	930	2236	0.756	0.789	0.794	0.463

Table 4: Validation Metrics while training

Observations:

- The model achieves the best performance on **knot_with_crack**.
- Small defects such as **Crack** exhibit slightly lower recall, reflecting the inherent difficulty of detecting fine-grained features within textured timber surfaces.
- Overall, validation metrics indicate a balanced detection capability across defect types.

6.3 Test Performance

Evaluation on test set shows strong generalization. Table 6.2 summarizes per-class performance metrics.

Class	Images	Instances	Precision	Recall	mAP@50	mAP@50-95
Crack	254	307	0.709	0.707	0.711	0.337
Dead_Knot	263	357	0.738	0.767	0.758	0.426
Live_Knot	305	506	0.800	0.737	0.806	0.416
knot_with_crack	269	332	0.849	0.898	0.923	0.672
All Classes	627	1502	0.774	0.777	0.800	0.463

Table 5: Test Metrics

Observations:

- Test metrics were nearly similar to validation performance, showing model's strong generalization.
- High mAP@50 for **knot_with_crack** and **Live_Knot** reflects effective multi-scale feature extraction and anchor-free detection design.
- Slightly lower mAP@50-95 for **Crack** emphasizes challenges in small-defect detection, consistent with prior observations.

6.4 Confusion Matrix Analysis

To narrow down on the prediction behaviour in terms of classes, the confusion matrix was calculated.

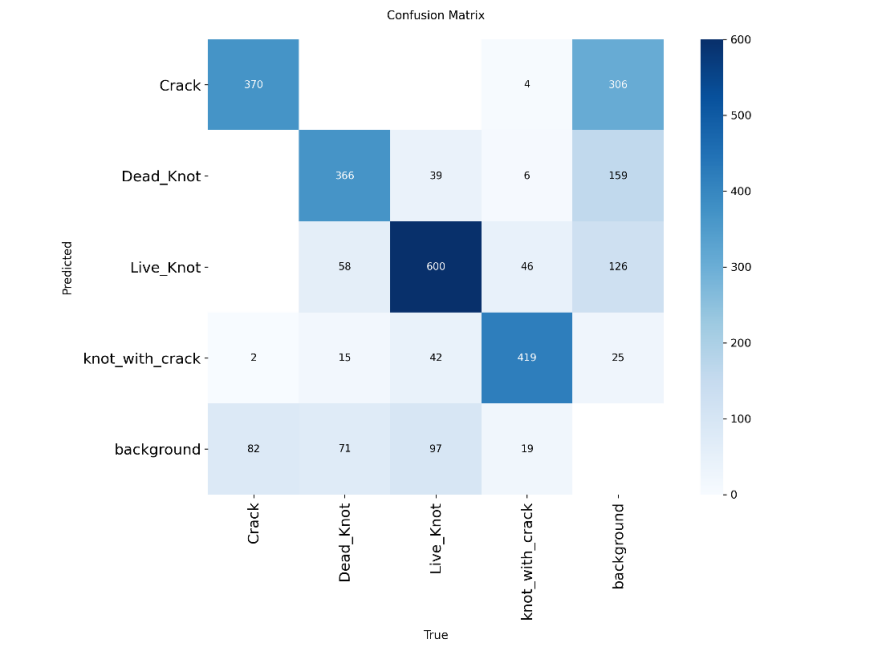


Figure 17: Confusion Matrix: Validation Set

Observations:

- Misclassification is primarily observed between **Dead_Knot** and **Live_Knot**, highlighting areas for potential improvement.
- Detection of **Crack** and **knot_with_crack** is highly reliable, consistent with mAP and precision metrics.
- Background misclassification remains minimal, confirming effective defect-background discrimination.

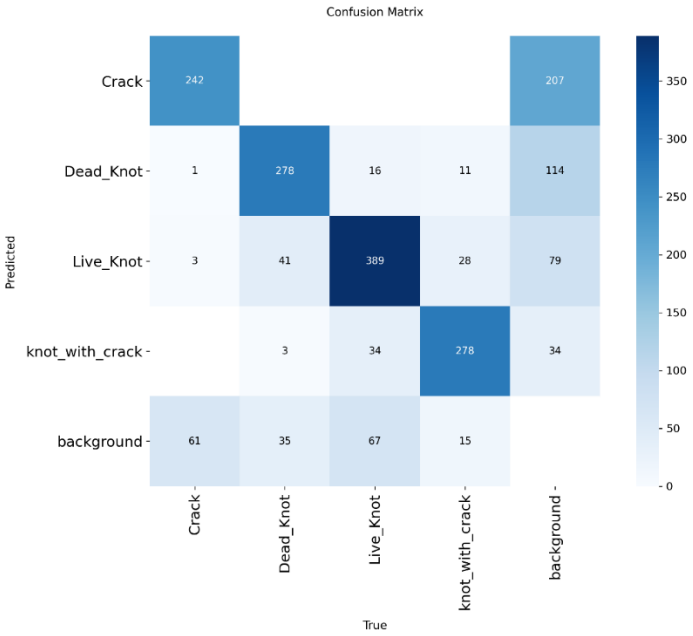
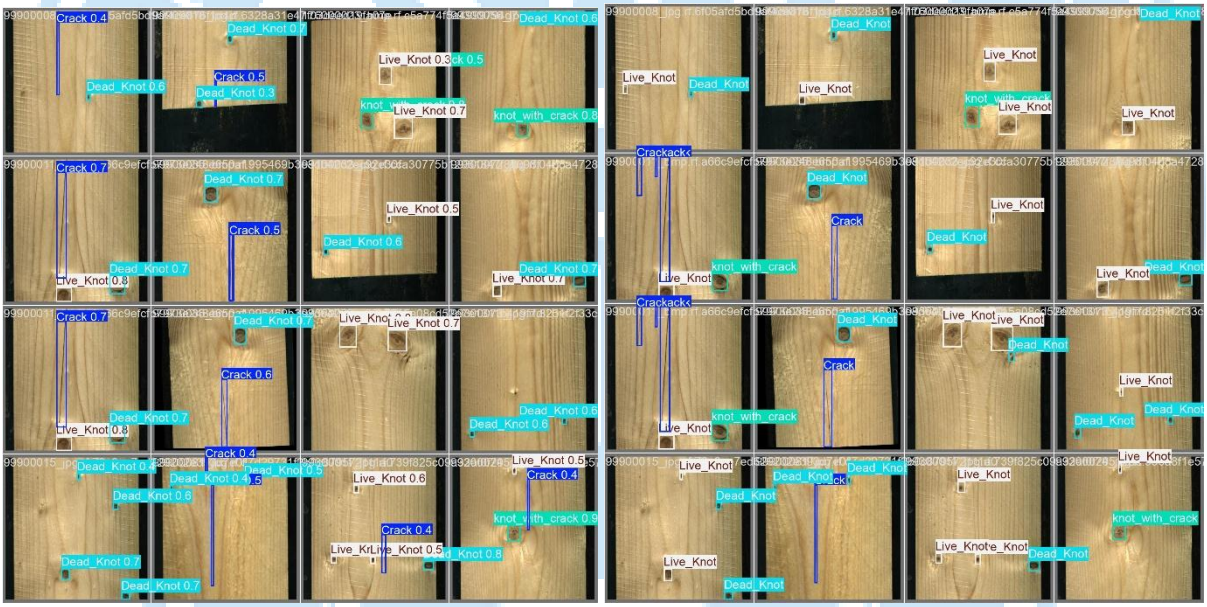


Figure 18: Confusion Matrix: Test Set



6.5 Summary of Findings

1. **Lightweight and Efficient:** The model reaches 4.46 GFLOPs, which costs a lot less in computation compared to the cost of the detection of the model.
2. **Robust Multi-Scale Detection:** CBAM attention, depth separable convolutions and the transformer layer improve the capacity to detect the presence of small cracks, as well as large knots.
3. **Strong Generalization:** Results of validation and test measurements show that the evaluation results are consistent with the operation of all defect classes.

4. **Edge Deployment Ready:** The inference of about 15 ms per image in the real-time proves that it is suitable to run in resource-constrained industrial use.
5. **Areas for Improvement:** The lack of distinction between DeadKnot and LiveKnot implies that there could be more optimization of features or that features should be further augmented in order to better fine-grain defect discrimination.

Conclusion: The experimental findings confirm the hypothesis that the proposed MobileNetV3-YOLO architecture will be an effective solution in terms of lightweight and real-time timber defect detection with high accuracy and proving that the architecture can be deployed on the edges of industry.

7. Discussion

The experimental findings confirm the fact that the suggested MobileNetV3-YOLO model can adequately balance between computational and detection performance in detecting timber defects. The compact nature of the model with 4.24 million parameters and 4.46 GFLOP allows it to be deployed on edge devices and has high performance in large and small defects.

7.1 Interpretation of Results

- The model also had the best precision and mAP on knotwithcrack defects, and this specializes in detecting large defects that are irregularly shaped and where contextual information is high.
- Lower recall and mAP for Crack defects indicate that detecting subtle, fine-grained features remains challenging due to wood texture variability and defect size.
- The confusion matrix has shown that DeadKnot sometimes was misclassified with LiveKnot, and so these defect types have similar visual appearances, which cannot be separated using the existing feature representations.

7.2 Comparison with Prior Work

The proposed architecture has competitively accurate models that use significantly lower computational cost than the models previously suggested e.g. SGN-YOLO, WDNET-YOLO and CWB-YOLO. For example:

- SGN-YOLO achieves mAP of 0.864 but with a heavier backbone unsuitable for edge deployment.
- WDNET-YOLO and CWB-YOLO improve small-defect detection but incur high latency and memory usage.

By comparison, the model suggested ends up with mAP 50 = 0.794 (val) and 0.800 (test) with approximately 15 ms of inference time per image which forms a viable model of real-time equivalence.

7.3 Error Analysis

- Misclassifications mostly result between similar defect types that are seen visually (DeadKnot vs LiveKnot) and fine cracks in textured areas.
- False alarms on minor defects indicate that more augmentations or refining on the attention would have to be done to further increase the sensitivity to the subtle patterns.

7.4 Implications for Industry:

The suggested light architecture will allow a real-time inspection of defects associated with timber on the edge machine in the wood-processing facility reducing latency and easy integration into automated quality control pipelines. The anchor-free and multi-scale design is both flexible toward any defect size and orientation.

7.5 Limitations and Future Work

- Dataset size and diversity remain limited; rare or occluded defect types may still be underrepresented.
- The model currently relies solely on RGB images; incorporating multi-modal inputs (e.g., depth, hyperspectral imaging) could improve small-defect detection.
- Future work could explore self-supervised learning or active learning strategies to further enhance generalization on unseen timber species and defect morphologies.

8. Conclusion

This study will introduce an edge-deployable, lightweight, high-performance YOLO-based architecture that can be used to detect timber defects. Key contributions include:

- A single compact MobileNetV3-YOLO model (4.46 GFLOPs) achieved the capability to run inference in real-time.
- The ability to show strong results in both small crack and large knot defect detection with depthwise separable convolutions, CBAM attention, and a small transformer module.
- Prediction on an augmented timber defect dataset, with mAP 50 of 0.794 (validation), 0.800 (test), and inference times of about 15 ms/image.
- Confusion matrix and error pattern analysis will be made which will help to make better defect differentiation in future.

The results assert the fact that optimally designed lightweight frameworks may provide detection performance that is industrially relevant requiring very few computations and enables implementation of real-time edge cards in scales in timber quality inspection systems.

9. Bibliography

[1] Y. L. H. Z. P. Wang, "SGN-YOLO: Detecting wood defects with improved YOLOv5 based on Semi-Global Network," vol. 13, no. Algorithms, p. 4, 2023.

[2] Y. Z. e. al, "Wood defect detection based on the CWB-YOLOv8 algorithm," 2024.

[3] .. C. e. al, "WDNET-YOLO: Enhanced deep learning for structural timber defect detection to improve building safety and reliability," no. Sensors, 2025.

[4] L. Y. e. al, "FDD-YOLO: A novel detection model for detecting surface defects in wood," no. Machines, 2025.

[5] X. P. e. al., "A review of recent advances in data-driven computer vision methods for structural damage evaluation: algorithms, applications, challenges, and future opportunities," *Springer,* no. Struct. Control Health Monitoring, 2025.

[6] "Roboflow," [Online]. Available: https://universe.roboflow.com/mushroom-species/defects-in-timber.

[7] R. W. e. al, "SiM-YOLO: A wood surface defect detection method based on improved YOLOv8," no. Sensors, 2024.

[8] J. H. e. al, "BPN-YOLO: A novel method for wood defect detection based on YOLOv7," no. Computers & Electronics in Agriculture, 202.

[9] Z. L. a. K. Sun, "Research progress of deep learning-based wood surface defect detection," *WSR Journal,* 2025.

[10] N. Z. e. al, "DAM-Faster R-CNN: Few-shot defect detection method for wood based on dual attention mechanism,," *Sci. Rep,* 2025.

[11] A. K. a. P. Singh, "Automated timber defect detection: A survey of traditional and deep learning approaches," 2023.