AI-assisted wood knots detection from historic timber structure imaging

Master Thesis

Master of Science in Digital Technologies in Heritage Conservation

Pan, Junquan

December 2, 2024

Supervisor:

1st: Prof. Dr. Christian Ledig
2nd: Dr.-Ing. Maria Chizhova

Chair of Explainable Machine Learning
Faculty of Information Systems and Applied Computer Sciences
Otto-Friedrich-University Bamberg

Abstract

Over time, various environmental and mechanical factors affect the stability and integrity of historic timber structures. Despite their cultural and architectural significance, these structures are often undervalued due to the lack of accurate assessments of their mechanical strength and structural parameters. Traditional methods, which rely on manual inspection and measurement, are time-consuming, prone to human error and inadequate for challenging conditions such as poor lighting or inaccessible areas.

Digitally documenting and monitoring the defects on historical wood surface, such as knots, cracks, and other surface irregularities, is therefore critical to preserving the structural condition of wood and ensuring the longevity and sustainability of historic timber structures. By integrating modern computer tools and methods, it is possible to overcome these limitations and provide a systematic and accurate approach to the assessment and maintenance of these invaluable heritage assets.

The main objective of this master thesis is to develop an automated process for the detection of wood knots. With the wide application of AI tools in different fields, this approach could help to provide an accessible, efficient and accurate solution for conservators, enabling them to perform detailed analysis and documentation of historic wooden surfaces automatically. This system may not only make a significant impact on the field of heritage protection but also offers the potential to reduce the excessive consumption of historical resources.

The methods based on machine learning and deep learning will be discussed, developed and experimented with in this thesis. The main detection process on wood knots is divided into two stages. In the first stage, a segmentation model such as Detectron2 from Meta will be used to determine the target timber surface, while in the second stage, Yolov8 will be tested to detect the wood knots. For further research, the results from the preview stages will be used to build an abstract geometric model that will combine the realistic measurements obtained by the mobile phone sensors to estimate the dimension of detected wood knots.

The resulting fully automated system combines the results of the segmentation model and the detection model. Various data from labelled datasets and unseen collection were also tested to confirm the performance of the final models. There are cases of inaccurate and unstable results during the testing process, which will continue to be improved in future research. And the final system will be used as a mobile phone application by heritage conservators.

Abstract

Im Laufe der Zeit wirken sich verschiedene Umwelt- und mechanische Faktoren auf die Stabilität und Integrität historischer Holzkonstruktionen aus. Trotz ihrer kulturellen und architektonischen Bedeutung werden diese Holzkonstruktionen oft unterschätzt, da es an genauen Bewertungen ihrer mechanischen Festigkeit und ihrer strukturellen Parameter mangelt. Herkömmliche Methoden, die auf manuellen Inspektionen und Messungen basieren, sind zeitaufwändig, anfällig für menschliche Fehler und unzureichend für schwierige Bedingungen wie schlechte Beleuchtung oder unzugängliche Bereiche.

Die digitale Dokumentation und Überwachung von Defekten an historischen Holzoberflächen wie Ästen, Rissen und anderen Fehler ist daher von entscheidender Bedeutung, um den strukturellen Zustand des Holzes zu erhalten und die Langlebigkeit und Nachhaltigkeit historischer Holzkonstruktionen zu gewährleisten. Durch die Integration moderner Computerwerkzeuge und -methoden ist es möglich, diese Einschränkungen zu überwinden und einen systematischen und präzisen Ansatz für die Bewertung und Erhaltung dieser unschätzbaren Kulturgüter zu bieten.

Das Hauptziel dieser Masterarbeit ist die Entwicklung eines automatisierten Verfahrens zur Erkennung von Holzästen. Angesichts der weit verbreiteten Anwendung von KI-Tools in verschiedenen Bereichen könnte dieser Ansatz dazu beitragen, eine zugängliche, effiziente und genaue Lösung für Restauratoren bereitzustellen, die es ihnen ermöglicht, detaillierte Analysen und Dokumentationen von historischen Holzoberflächen automatisch durchzuführen. Dieses System könnte nicht nur einen bedeutenden Einfluss auf den Bereich der Denkmalpflege haben, sondern bietet auch das Potenzial, den übermäßigen Verbrauch historischer Ressourcen zu reduzieren.

Methoden basiert auf maschinellen Lernens und Deep Learning werden in dieser Arbeit diskutiert, entwickelt und getestet. Der Hauptprozess der Asterkennung ist in zwei Schritte unterteilt. In der ersten Stufe wird ein Segmentierungsmodell wie Detectron2 von Meta verwendet, um die Zielholzoberfläche zu bestimmen, während in der zweiten Stufe YOLOv8 getestet wird, um die Holzäste zu erkennen. In der weiteren Forschung werden die Ergebnisse der Voruntersuchungen verwendet, um ein abstraktes geometrisches Modell zu erstellen, das die realistischen Messungen der Handysensoren kombiniert, um die Größe der erkannten Holzäste zu schätzen.

Das resultierende vollautomatische System kombiniert die Ergebnisse des Segmentierungsmodells und des Erkennungsmodells. Verschiedene Daten aus markierten Datensätzen und nicht markierten Sammlungen wurden ebenfalls getestet, um die Leistung der endgültigen Modelle zu bestätigen. Es gab Fälle von ungenauen und instabilen Ergebnissen während des Testprozesses, die in zukünftigen Forschungsarbeiten weiter verbessert werden sollen. Das endgültige System wird von Restauratoren als mobile Anwendung genutzt werden.

Acknowledgements

I would like to begin by expressing my sincere gratitude to Prof. Christian Ledig, my thesis supervisor, for his unwavering support and insightful guidance throughout my master's thesis. His academic openness and professionalism have been invaluable to this work and have significantly influenced my entire academic journey.

As next, I extend heartfelt thanks to Dr. Maria Chizhova for her collaboration and partnership during the development of this thesis, as well as for future cooperation on the project.

I am also deeply grateful to Prof. Mona Hess, my degree programme director, for her continuous support and thoughtful guidance throughout my master's studies.

Additionally, I would like to thank Dr. Thomas Eißing and Frank Ebner for their friendly cooperation throughout the whole project and the members of the DDT examination committee for their support in granting the special permission for my thesis.

Lastly, my profound thanks go to my parents and friends, whose encouragement and moral support were essential in helping me complete this work. Their belief in me was a constant source of strength.

Contents

List of Figures

List of Tables

List of Acronyms

AI	Artificial Intelligence
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSP	Cross Stage Partial (used in CSPDarknet53)
C2f	Cross Stage Partial with Focused Fusion
DL	Deep Learning
DFL	Distributional Focal Loss
DIN	Deutsches Institut für Normung
DIN EN	Deutsches Institut für Normung Europäische Norm
DSLR	Digital Single-Lens Reflex (camera)
FCN	Fully Convolutional Network
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
GLCM	Gray Level Co-Occurrence Matrix
GPU	Graphics Processing Unit
IoU	Intersection over Union
kNN	k-Nearest Neighbors
LBP	Local Binary Patterns
LiDAR	Light Detection and Ranging
mAP50	Mean Average Precision at 50% Intersection over Union (IoU)
mAP50-95	Mean Average Precision at IoU thresholds from 50% to 95% in steps of 5%
ML	Machine Learning
MOE	Modulus of Elasticity
MOR	Modulus of Rupture
NMS	Non-Maximum Suppression
ODCA	Omni-Dynamic Convolution Coordinate Attention
PR curve	Precision-Recall Curve
PSO	Particle Swarm Optimization
RAM	Random Access Memory
ResNet	Residual Network
R-CNN	Region-based Convolutional Neural Network
ROI	Region of Interest
RPN	Region Proposal Network
RGB	Red, Green, Blue (color space)
SNR	Signal to Noise Ratio
SPP	Spatial Pyramid Pooling
SSD	Single Shot MultiBox Detector
SVM	Support Vector Machine
TL-ResNet34	Transfer Learning-Residual Network 34
TP	True Positive
YOLO	You Only Look Once

Notation

Input and Features in ResNet

$I \in \mathbb{R}^{H \times W \times 3}$	Input image of height H, width W, and 3 color channels (RGB)
$F_2 \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C_2}$	Feature map at Stage P_2
$F_3 \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times C_3}$	Feature map at Stage P_3
$F_4 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_4}$	Feature map at Stage P_4
$F_5 \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C_5}$	Feature map at Stage P_5

Convolution Operation

$F_{i,j,k}$	Value of the feature map at position (i, j) in the k-th output channel
$I_{i+m,j+n,c}$	Pixel values of input image at position $(i + m, j + n)$
$W_{m,n,c,k}$	Convolutional kernel value at position (m, n) for the c-th input channel and k-th output channel

Residual Block in ResNet

y	Output from residual block
x	Input features from the previous layer
$F(x, W_i)$	Output from convolutional layers parameterized by weight W_i

Feature Pyramid Network (FPN)

P_l	Feature map from the l FPN layer
F_l	Feature map from ResNet at Stage l
P_{l+1}	Feature map from the previous FPN layer

Region Proposal Network (RPN)

$\hat{p}_i$	Classification prediction in RPN
σ	Sigmoid function
W_{cls}	Weight matrix for the classification layer
P_i	Feature vector for the corresponding anchor
$\hat{t}_i$	Refined bounding box coordinates predicted by RPN
W_{bbox}	Weight matrix for the regression layer

ROI Head

$V(x, y)$	Bilinear interpolation for feature alignment
x, y	Coordinates in the feature map
w_{ij}	Interpolation weights
x_i, y_j	Nearest pixel coordinates
$\hat{p}_i$	Class prediction in ROI head
$Softmax$	Softmax function
W_{cls}	Weight matrix for classification in ROI head
$F_{align,i}$	i-th feature from the whole aligned feature maps F_{align}
$\hat{t}_i$	Bounding box regression in ROI head
W_{bbox}	Weight matrix for bounding box regression in ROI head
$Mask_{i,j}$	Predicted Mask
$F_{align,i+m,j+n}$	Value of the aligned feature map at position $(i+m, j+n)$
W_{mask}	Weight matrix for mask prediction in ROI head

YOLO Architecture

F_{CSP}	Cross Stage Partial (CSP) feature fusion output
t_x, t_y	Offset for the bounding box center prediction relative to the grid cell
x_i, y_i	Coordinates of bounding box center
w_i, h_i	Width and height of bounding box
p_w, p_h	Predefined width and height of grid cell
$\hat{c}_i$	Predicted confidence score
P_i	Feature vector corresponding to the i-th prediction
W_{conf}	Weight matrix for confidence layer

Loss Functions and Metrics in YOLO

$total\_loss$	Total loss
$loss\_cls$	Classification loss
$loss\_box\_reg$	Bounding box regression loss
$loss\_mask$	Mask prediction loss

$loss\_rpn\_cls$	RPN classification loss
$loss\_rpn\_loc$	RPN localization loss
TP	True positives
FP	False positives
FN	False negatives
mAP50	Mean Average Precision at IoU=0.5
mAP50-95	Mean Average Precision averaged over IoU thresholds from 0.5 to 0.95 in steps of 0.05
$P(Object)$	Objectness score
$P(Class\|Object)$	Class confidence score

Otsu's Thresholding

$\sigma_b^2(t)$	Inter-class variance
$w_0(t), w_1(t)$	Weights (proportions) of foreground and background pixels
$\mu_0(t), \mu_1(t)$	Average grey value of foreground and background pixels

1 Introduction

1.1 Motivation

The stability of timbers changes over time and needs to be monitored accordingly. In timber production, new timbers are inspected and graded according to their structural quality parameters (e.g. strength values). The existing criteria for wood grading in Germany (DIN 4074[1], DIN EN 1995) cover only new timber and do not contain any direct specifications for historic timber in existing buildings. The exact strength values are usually not known for old wood. For safety reasons, very low values are generally assumed for existing timber structures (which also means low quality and grade), although the actual strength values may be higher. As a result, more (waste) wood is sorted out and replaced than is technically necessary, i.e. valuable natural resources are wasted unnecessarily.

However, it is still possible to transfer the used wood to a higher grade. Additional methods can improve the quality of existing historic timber, enabling it to be used for more cost-effective and value-added purposes. Such visually recognizable characteristics, such as knots (knotholes and their diameter), serve as individual evidence to estimate the wood strength and to adjust the sorting class. The waste timbers recorded with optical measuring sensors are automatically analysed for knots. With the help of this analysis, objective and comprehensible conclusions can be drawn about strength and sorting class.

By automatic surface characteristic recognition, digital twins of timber constructions (roof structures, individual timbers) can be significantly enriched with this information. Such models are meaningful for objective monitoring and estimating the need for renovation, energy efficiency and conservation measures, without the need for additional costly expertise. It is therefore essential to be able to automatically identify, classify and thus objectively analyse corresponding wood surface characteristics in order to observe and understand the condition and change of this material.

1.2 Proposal of the Thesis

The primary aim of this thesis is to develop a comprehensive workflow that focuses on the automated knots detection in historic timber structures. This workflow will assess both the dimensions of the timber and the detected knots from digital 2D images using a variety of computer graphics and computer vision algorithms, including state-of-the-art deep learning models.

To achieve this goal, several key aspects must be addressed. Before diving into the practical research process, a thorough examination of current research on the conservation of historic timber structures is essential, along with an exploration of

[1]DIN 4074 is a specification of the German Industrial Standard (Deutsches Institut für Normung, or DIN for short), which is dedicated to the grading of solid wood for structural purposes.

how modern machine learning and deep learning methods can enhance traditional workflows and accelerate conservation efforts. Besides, due to the specialized task of this research, it is anticipated that there will be a scarcity of useful resources, such as operational datasets for machine learning tasks. Therefore, the creation of well-defined datasets will be critical to provide a solid foundation for subsequent experiments.

During the whole research process, the development of a robust and accurate system for the automated detection and estimation of knots in historic timber structures is crucial. This task involves several sub-tasks, including image acquisition, preprocessing, and the application of advanced computer vision techniques. The goal is to create a pipeline that can handle various conditions of the timber, such as different lighting or texture variations, while still providing reliable results. The algorithm should be adaptable enough to work with different types of historic wooden structures, which often vary in size, shape, and various historical traces.

The thesis will also explore the optimisation of deep learning models specifically tailored to this task, evaluating different architectures such as Detectron2 and YOLO. The challenge is not only to select the most appropriate model, but also to fine-tune it to account for the unique characteristics of timber structures, such as irregular surfaces, complex grain patterns and varying knot formations. The performance of these models is rigorously tested and validated against manually annotated data to ensure that the automated process achieves high levels of accuracy and reliability.

To conclude, this thesis aims to bridge the gap between traditional manual conservation methods and modern computational techniques. By developing a comprehensive, automated workflow, the research not only contributes to the academic field, but also has practical value in helping to preserve cultural heritage for future generations.

2 Theoretical Foundation

As the most widely used building material, wood has demonstrated great value throughout the architectural history of mankind. In Germany, the percentage of approved buildings constructed predominantly with wood will exceed 20 percent in 2023 for both new residential (22 percent) and non-residential (23.4 percent) buildings[2]. This trend underlines the continuing importance of wood as a primary material in contemporary architectural projects and highlights its long-standing use as a fundamental building material. In the field of heritage conservation, historic timber structures represent a significant challenge due to their susceptibility to environmental degradation and ageing. The conservation and analysis of these historic materials require both traditional manual methods and modern technological advances. This chapter aims to provide a comprehensive theoretical foundation for related studies, focusing on two key aspects: the conservation and analysis of historic timber using traditional techniques, and the application of machine learning in related fields to enhance current practices.

2.1 Wood knots analysis in timber mechanical properties

2.1.1 The impact of wood knots on timber

In the research by Ramage et al. (2017), factors such as tree species, growing conditions, wood processing and handling, natural defects (e.g., knots or spiral grain), and processing-induced defects (e.g., cracks) significantly affect the mechanical properties of wood. Knots form as a result of knot growth; if the knot is alive during knot growth, tightly bound "live knots" are formed. Conversely, if the branch dies before it grows, "dead knots" are formed and these knots may be dislodged from the wood during processing. Knots can change the grain direction of the surrounding wood, resulting in an interruption of fibre continuity and thus creating areas of stress concentration. In the research by Saad and Lengyel (2022), the position and dimensions of wood knots were found to significantly influence the ultimate load capacity of timber structures.

There are several research methods for studying the effect of knots on the mechanical properties of wood, the most prominent being mechanical testing of actual knotty wood in laboratories and simulation experiments to model the impact of knots on wood properties. Zhang et al. (2024) investigates the effect of knots on the mechanical properties of chinese fir using a three-point bending test, X-ray computed tomography, and digital image correlation, revealing that knot size and position significantly influence strain distribution and mechanical behaviour, particularly modulus of elasticity (MOE) and modulus of rupture (MOR). Hu et al. (2018) presents a novel laboratory method combining optical and laser scanning to accurately map growth layer geometry and 3D fiber orientation around knots in Norway spruce,

[2]Holzbau Deutschland, Lagebericht Zimmerer/Holzbau 2024

revealing the significant impact of knots on timber's mechanical properties and providing essential data for modeling and strength prediction. Fan et al. (2023) models the strength-reducing effects of knots on Douglas Fir lumber using tensile testing and Bayesian analysis, offering a more precise method for evaluating lumber quality.

Further simulation studies have explored the impact of knots on the mechanical properties of wood through advanced modeling techniques. Baño et al. (2011) developed a finite element model (FEM) to simulate the effects of knots and grain deviation on the flexural strength of timber beams. The model was found to accurately predict failure loads with an error of less than 9.7%, which highlights the critical role of knots in stress distribution and structural failure. Similarly, Burawska et al. (2013) demonstrated that knots and equivalent openings of the same shape, size, and position exhibit comparable effects on bending strength parameters of structural timber. Expanding on this, Lukacevic et al. (2019) introduced a 3D model to simulate knots and related fiber deviations in sawn timber, accurately predicting mechanical properties like bending stiffness and strain distribution, while highlighting the importance of pith reconstruction and fiber deviation patterns for timber grading and structural analysis.

Laboratory analysis and simulation offer valuable tools for further estimating the impact of wood knots on mechanical properties, as demonstrated by Fink and Kohler (2014), who developed a predictive model for tensile strength and stiffness of knot clusters within structural timber using destructive and non-destructive tensile tests to uncover key relationships between knot morphology and mechanical performance.

2.1.2 Evaluation of wood knots on historical timber

However, for historical timber, accurately analysing the effects of knots on the mechanical properties of extant wood structures is challenging due to environmental factors by historical wood structures such as temperature and humidity variations, as well as potential insect damage. Consequently, conservators and engineers can indirectly assess the impact of knots on the properties of historical timber by studying surface features and wear patterns. By examining traces such as knots and their proportions in relation to the overall timber structure, they can estimate how knots may have affected the mechanical behaviour and stability of the wood over time, aiding both conservation strategies and structural assessments.

The Figure 1 shows the determination of the largest single branch by the dimension of knot and corresponding timber in DIN 4074. The master thesis by Frank Ebner (Ebner, 2018), supervised by Prof. Dr. Thomas Eißing, involved detailed investigations and experiments to evaluate the strength of structural timbers. The research included an in-depth study of the effect of knots on the strength of structural timber, focusing on how the size, frequency, and placement of knots within the timber affect its load-bearing capacity. By examining different types of knots and their influence on stress distribution, it provides a more accurate assessment of timber integrity in historic structures, where knots often contribute to variability in timber strength. Traditional methods for measuring wood knots and other types of wood

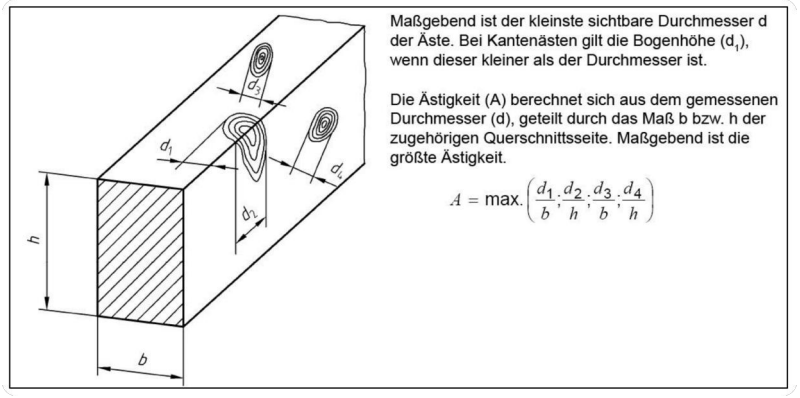


Figure 1: The determination of the largest single branch by the dimension of knot and corresponding timber. A describes the ratio between the smallest diameter of the largest individual knot and the corresponding height and width of the timber, used for grading wooden components. d_1, d_2, d_3, d_4 denote the measured diameters of the visible knots, while b stands for the width and h for the height of the timber.(DIN 4074:2012-06, S. 4, Bild 1)

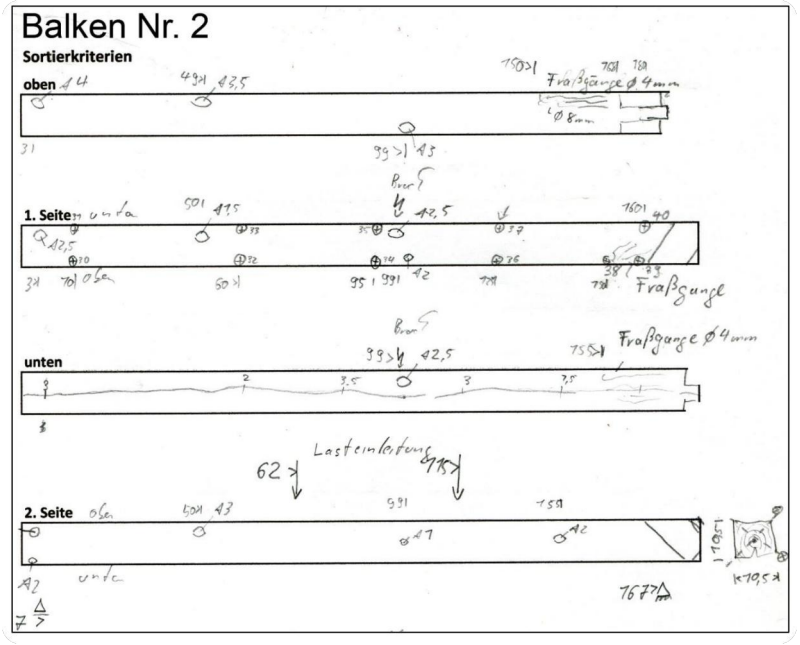


Figure 2: The on-site sketch by Frank Ebner illustrates visual grading in accordance with DIN 4074-1:2012-06. The grading class is designated as S10 for bar 2, primarily due to the evaluation on observed wood knots according to the calculation Figure 1. The sketch was originally created during the manual measurement of wood knots and will be further refined for more detailed analysis.(Ebner, 2018)

damage, such as cracks, are often time-consuming and susceptible to human error. These approaches typically involve direct measurement with calipers or similar tools, followed by manual documentation and further improvements.(Figure 2) This

documentation process, along with subsequent physical calculations, heavily relies on mechanics theory to estimate the impact of knots on the structural properties of wood.

Figure 3: Roof structure of the Dominican Church

Accurate measurement is particularly critical in heritage preservation, where the variability in timber strength caused by knots, as described earlier, poses significant challenges. Assessing the stability of historical wooden structures, such as the roof structure shown in Figure 3, is further complicated by safety concerns and the inherent limitations of traditional methods. These methods, which rely on manual measurement and documentation of knots and other wood damages, are often time-consuming and prone to human error. Therefore, digital technologies for detecting, measuring, documenting, and systematically analysing wood knots in historical wooden structures are essential. These methods can significantly enhance the efficiency and accuracy of conservation strategies, enabling better assessment and preservation of structural integrity.

2.1.3 The Expectation of AI-assisted system

To improve traditional manual processes for measuring wood knots, advanced computational tools can replace or refine outdated practices. By using digital data, the mechanical performance of wood can be analysed with greater accuracy, enabling more reliable assessments of historical wood conditions. This improved approach facilitates the development of data-driven conservation strategies tailored to ongoing analysis.

Building on the background discussed in Chapter 1.1, the introduction to the impact of wood knots on timber in 2.1.1 and actual analysis on historical timber in 2.1.2, the Figure 4 summarizes the key problems in evaluating historical timber, the current

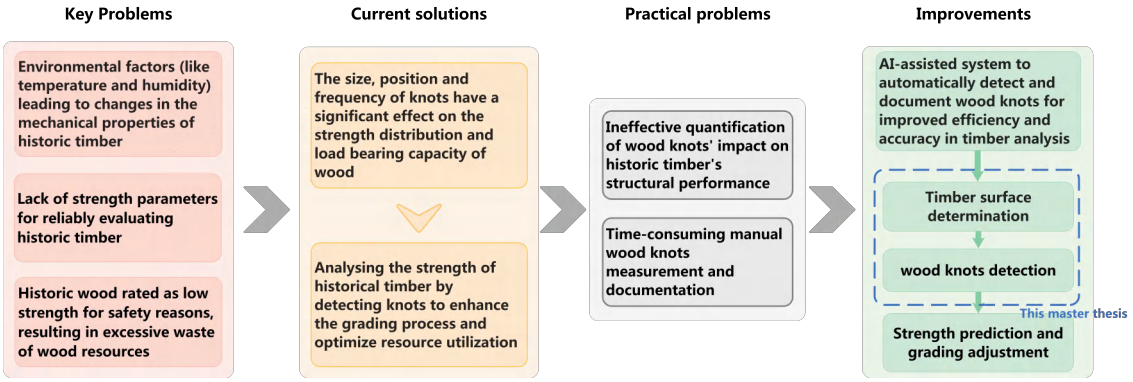


Figure 4: Pathway from Key Problems to AI-Assisted Solutions for historical timber analysis and grading

solutions for analysing wood knots to estimate the strength of historical timber, the practical challenges in implementation, and the proposed improvements through an AI-assisted system aimed at enhancing efficiency, accuracy, and resource utilization.

AI-based image recognition techniques can further streamline the process by detecting and mapping wood knots systematically. This allows for a more detailed analysis of their impact on the wood's load-bearing capacity, providing valuable insights into potential structural weaknesses. The integration of measurement, calculation, and analysis supports a data-driven approach to conservation, ensuring that the structural health of heritage architecture is assessed and preserved with greater accuracy and efficiency.

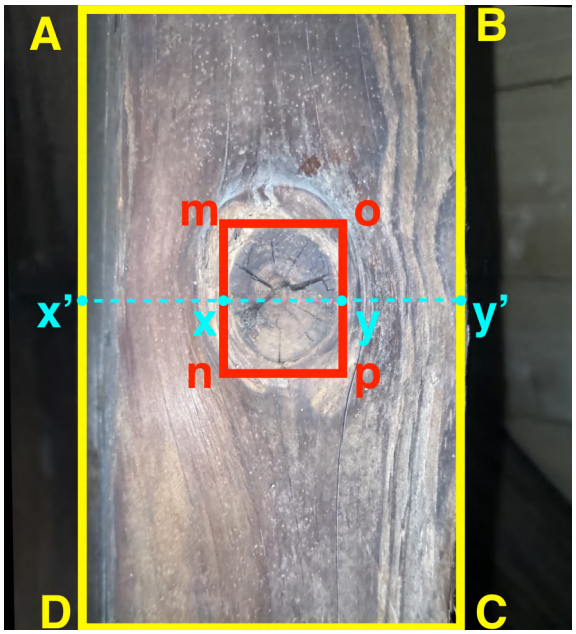


Figure 5: Ideal Output: Geometric representation of wood knot and its relative position within timber boundaries

The expected final output from the complete AI-assisted system shown in Figure 5 illustrates the geometric representation of a wood knot and its relative position within the timber boundaries. The outer yellow dashed quadrilateral $ABCD$ defines the segmented timber boundary, with $AB \parallel DC$, $AD \parallel BC$, and potentially $AD \perp AB$. The red dashed rectangle $mnop$ represents the bounding box of the detected wood knot, with $mn \parallel AD$ and $op \parallel BC$. The midpoint x of mn defines the line xx', which satisfies $xx' \perp mn \parallel AD$, while the midpoint y of op defines the line yy', which satisfies $yy' \perp op \parallel BC$.

To evaluate the effect of knots on the surface of the wood, the ratios $Ratio_A = \frac{x'x}{x'y'}$ and $Ratio_B = \frac{yy'}{x'y'}$ are calculated to represent the relative geometric relationship of the knot to the boundaries of the wood. These ratios provide normalised metrics that help to quantify the position and orientation of the knot within the timber, enabling further structural analysis. By standardising the position and size of the knot relative to the dimensions of the wood, these ratios support comparative studies between different samples and facilitate predictions of the mechanical properties of the wood.

2.2 Artificial Intelligence

2.2.1 Machine learning and deep learning

As a subfield of artificial intelligence, machine learning focuses on using statistical methods (algorithms) to enable computer systems to learn patterns from data and generalize these patterns to unseen data, allowing them to solve specific tasks. This process relies on manual feature extraction. Humans define and extract key patterns or textures in an image (e.g., shape, size of wood knots, etc.) and then feed these features into the classifier to complete the classification task. This approach can be achieved through various learning paradigms, including supervised learning, unsupervised learning, and reinforcement learning (Goodfellow et al., 2016). In supervised learning, the system is trained on labelled data, where each input is associated with a known output. In this way, the model establishes a relationship between inputs and outputs and uses this to predict the outcome of new inputs. In contrast, unsupervised learning works with unlabelled data, allowing the system to identify hidden patterns or groupings in the data. Reinforcement learning works differently, as the system interacts with the environment, receives feedback in the form of rewards or punishments, and adjusts its actions over time to improve performance.

In the early work of Gu et al. (2010), the researchers addresses wood defect classification using Support Vector Machines (SVMs, Cortes and Vapnik (1995)), focusing on a tree-structured classifier to differentiate between wood knots types. SVM is a widely used supervised machine learning that classifies data by finding an optimal line or hyperplane that maximises the distance between each class. The further study Muhammad Redzuan and Yusoff (2019) expands the pre-processing pipeline

and adds Local Binary Patterns(LBP, Ojala et al. (1994)) descriptor to captures local contrasts and patterns on wood knots based on enhanced SVM method.

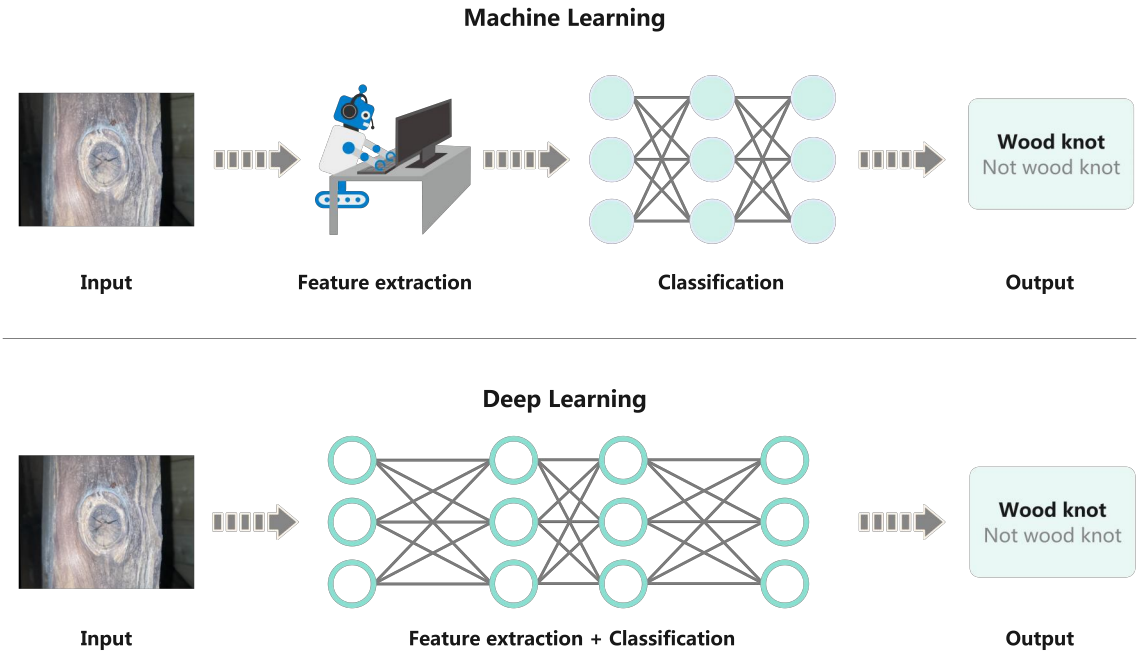


Figure 6: Workflow differences between machine learning and deep learning

Deep learning is a specialized subfield of machine learning that focuses on using neural networks[3] with multiple layers to automatically learn feature representations from raw data. While traditional machine learning requires manual definition and extraction of features, deep learning models learn these features directly during the training process without human intervention. The Figure 6 shows the differences between machine learning and deep learning in the context of wood knot detection and classification tasks.

2.2.2 Convolutional Neural Network

The two primary deep learning frameworks used in this master thesis, YOLOv8 and Detectron2, are fundamentally based on Convolutional Neural Network (CNN), which was initially introduced by LeCun et al. (1989) and further optimized in LeCun et al. (1998). The basic CNN framework consists of an input layer, which receives and preprocesses the raw data, followed by convolutional and pooling layers, which work together to extract important features. A fully connected architecture is then typically used to combine the learned features and produce the final classification or prediction results, as shown in Figure 7. In the convolutional layers, the network

[3]A neural network (S. and Walter, 1943) refers to a computational model inspired by the structure and function of the human brain, which consists of interconnected nodes (or "neurons") organized in layers and process information by simulating how biological neurons communicate.

applies filters (convolution kernels) to the input data by sliding them over the image, capturing important features locally across the input. The subsequent pooling layers help reduce the spatial dimensions of these feature maps, keeping the most critical information and minimizing computational load. Pooling also reduces the risk of overfitting by preventing the model from learning overly specific details that may not generalize well to new data.

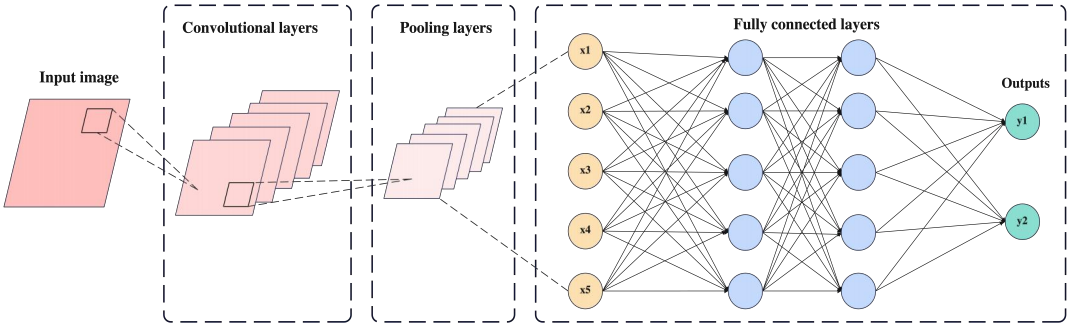


Figure 7: Example of a typical CNN structure

2.3 Further AI-based methods in related fields

In the preliminary research, several further valuable studies have applied machine learning methods to detect wood knots and other defects, offering notable insights for this master thesis.

In Qayyum et al. (2016), the researchers propose to use texture features extracted from the Gray Level Co-Occurrence Matrix (GLCM, Haralick et al. (1973)) as input parameters for a feed-forward neural network trained using Particle Swarm Optimisation (PSO, Kennedy and Eberhart (1995)) to classify different types of wood defects, focusing on wood knots. The GLCM transforms images into a matrix representation that captures spatial relationships between pixel intensities, providing features like contrast, correlation, energy, and homogeneity. The PSO algorithm optimises the weights and biases of the neural network by mimicking the behaviour of a swarm, such as birds or fish, allowing the system to efficiently converge to an optimal solution for accurate defect classification. He et al. (2019) introduces a Mixed Fully Convolutional Neural Network for locating and classifying wood defects based on self-made dataset with defects like live knots, dead knots and cracks. As shown in Figure 8 and Figure 9 the researchers collected 1200 original images with defects and established two datasets with data augmentation that dataset 1 contains 117,091 images within 6 classes to classify the defects and dataset 2 contains 3227 defect images and corresponding masks under more accurate labelling to segment the defects.

In 2020, Ding et al. (2020) integrates the DenseNet (Huang et al., 2017) into the single shot MultiBox Detector(SSD, Liu et al. (2016)) to enhance the feature extraction process and accelerate the efficiency of detection on wood knots. Thanks to

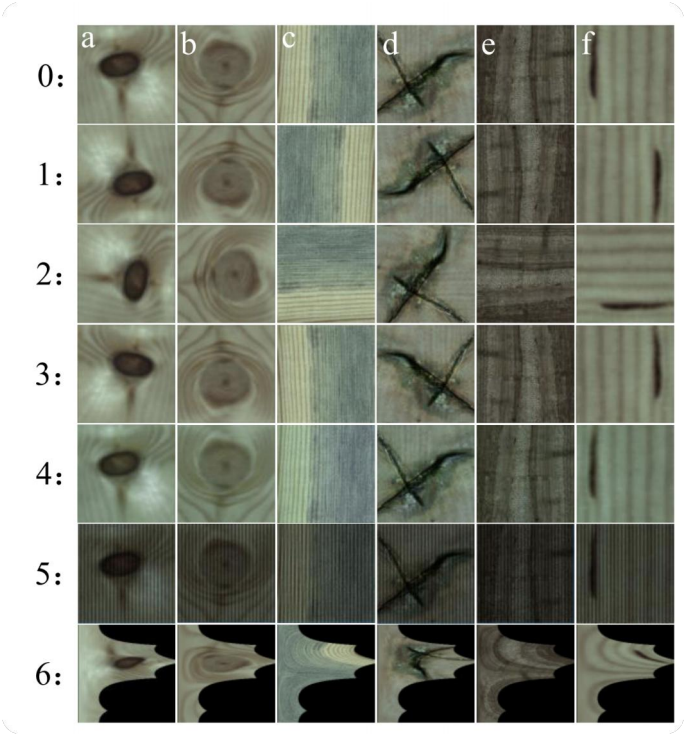


Figure 8: Six common wood defects and data augmentation of dataset 1, six common wood defects: a. a dead knot, b. a live knot, c. blue stain, d. crack, e. brown stain, f. pitch streak. Original images and same data augmentation method: 0. original images, 1. rotated by 180∘, 2. diagonal flip with 45∘ diagonal, 3. vertical mirror, 4. increased the hue by 0.09, 5. added Gaussian noise to image, and 6. polar coordinates are transformed with coordinates (200, 200) as the centre of polar coordinates.(He et al., 2019)

the innovative Dense Blocks, where each slice is directly connected to all previous slices (Figure 10), DenseNet effectively reuses features and ensures efficient gradient flow, resulting in superior performance in various applications such as image classification, object detection and medical imaging, which is a potential deep learning model to be used in the future researches.

Further studies, such as Gao et al. (2021), proposed a migration-based residual neural network (TL-ResNet34) built on ResNet34 to improve the accuracy of wood knots detection. Their study used a timber knot dataset from the University of Oulu, consisting of 448 images of spruce knots. After data augmentation, the dataset was expanded to include 1,885 training images, 636 validation images, and 615 test images. As illustrated in Figure 11, the dataset contains seven classes, with Row A representing the original images for each class and Rows B-G showing augmented samples. Although their method achieved relatively high accuracy, this dataset does not align with the specific characteristics of the target scenario in this master thesis.

In the meanwhile, the YOLO model family has also been widely applied in research related to wood defect detection. In 2020, Liu et al. (2020) used an early version,

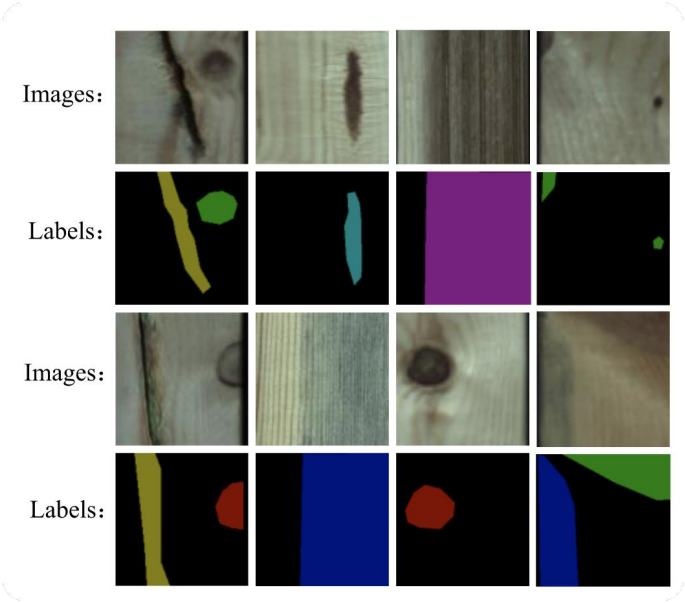


Figure 9: Defect images and corresponding label images in dataset 2. In label images after visualization, black means background, yellow means crack, green means live knot, red means dead knot, blue means blue stain, light blue means pitch streak, and purple means brown stain.(He et al., 2019)

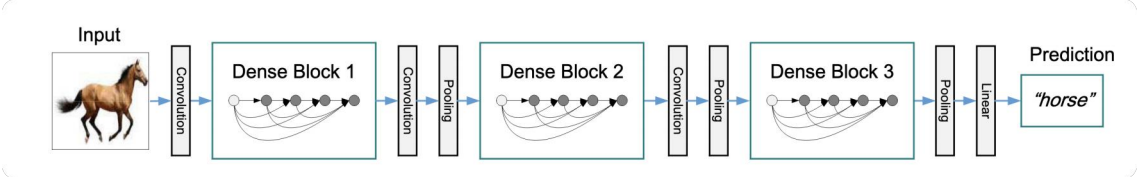


Figure 10: A deep DenseNet with three dense blocks.(Huang et al., 2017)

YOLOv3, to automatically detect timber cracks. Later, Fang et al. (2021) employed the enhanced YOLOv5 model to detect surface knots on sawn timbers (Figure 12). While this research task is similar to the focus of this master's thesis, it emphasizes fresh wood material, and the characterization of wood knots does not translate well to historical timber structures. Cui et al. (2023) implements Spatial Pyramid Pooling (SPP, He et al. (2015)) also in YOLO V3 to enhance the model performance on real-time wood defects detection.

Wang et al. (2023b) adds a novel Omni-Dynamic Convolution Coordinate Attention(ODCA, Li et al. (2022)) mechanism in YOLOv7 to enhance feature extraction and small-target detection. Similarly, the paper by Wang et al. (2023a) introduces a series of enhancements to the YOLOv8n model also aimed at improving the detection of small defects in sawn timber surfaces using a tiny target detection head. Further studies, such as the use of the improved YOLOv8 for automatic wood surface defect detection Xi et al. (2024) based on the dataset provided by Kodytek et al. (2021), have also demonstrated valuable impact.

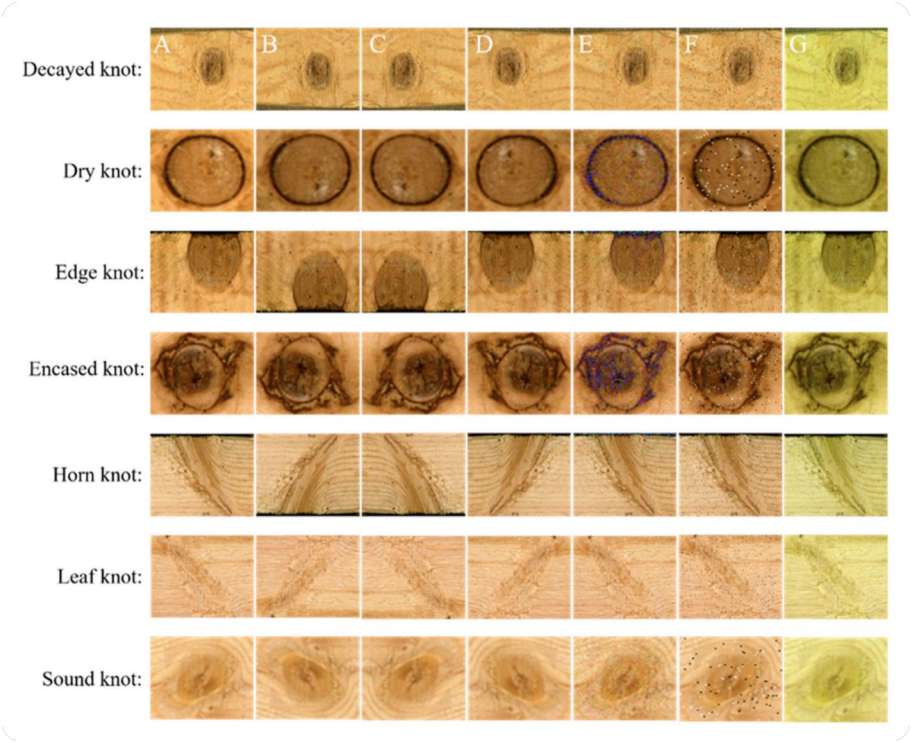


Figure 11: Samples collected by Oulu University with data augmentations

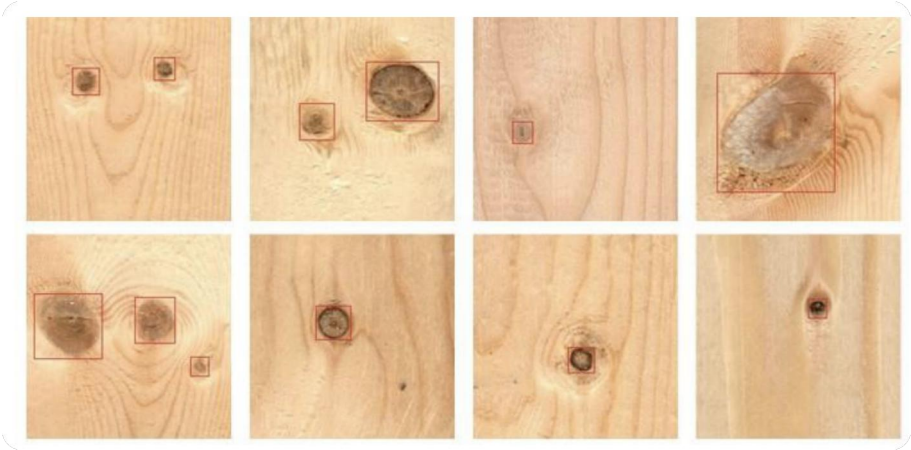


Figure 12: Sawn timbers samples used in Fang et al. (2021)

Research in related fields clearly shows that applying machine learning and deep learning for the automated identification of wood knots on historic timber surfaces presents significant challenges. The first challenge lies in the quality of datasets, which must be tailored to the unique characteristics of historic wooden structures. Another challenge involves optimizing various models and mechanisms to achieve accurate performance in practical scenarios. In the next section, the dataset-building process will be further elaborated, followed by a discussion of the methodology in subsequent sections.

3 Dataset

As shown in Figure 5, the detection of timber boundaries mainly depends on accurate segmentation results, while the precise location of wood knots relies heavily on the performance of detection models. Therefore, reliable and robust model performance depends heavily on high-quality, standardized labelled datasets, which are essential during the planning stages to effectively support the supervised learning process of these deep learning models. Supervised learning uses these labelled examples to train the model, guiding it to accurately recognise patterns and make informed predictions based on previously seen data. This structured labelling ensures that the model learns effectively, improving its generalisation and accuracy on new, unseen data.

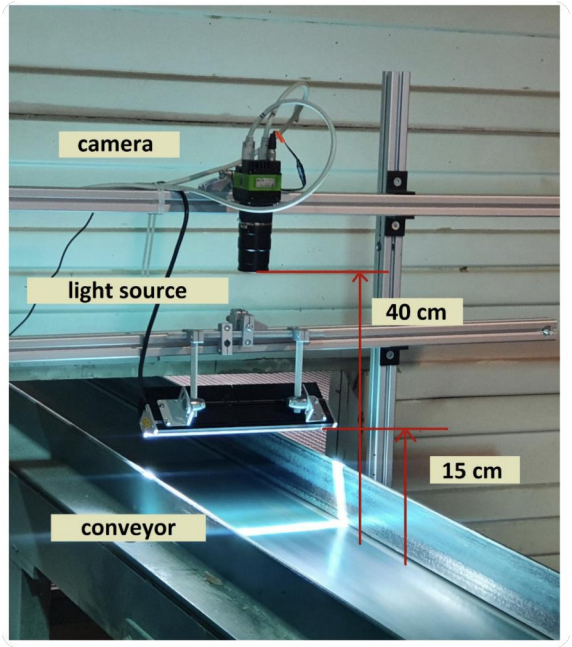


Figure 13: The mechanical system to capture wood surfaces by Kodytek et al. (2021)

In the early research phase, it is crucial to focus on identifying publicly available open-source datasets for general detection on wood defects. There are several available datasets like "A large-scale image dataset of wood surface defects for automated vision-based quality control processes" (Kodytek et al., 2021). As showed in Figure 13 the images within the above dataset are captured through a specially designed mechanical construction combining conveyor belts and vertical cameras. The whole dataset contains 992 images of sawn timber with no defects and 18,283 images of timber with one or more surface defects, the whole collection is classified into 10 types of wood surface defects. The exported images (Figure 14) shows some labelled images with live knot, dead knot and knot with crack. An initial experiment with this dataset was conducted, but the model's performance was suboptimal when applied to detect knots in historic timber structures. The likely reason is that the

dataset comprises only new timber, and the labelled masks encompass not just knots but also other surface defects like cracks.

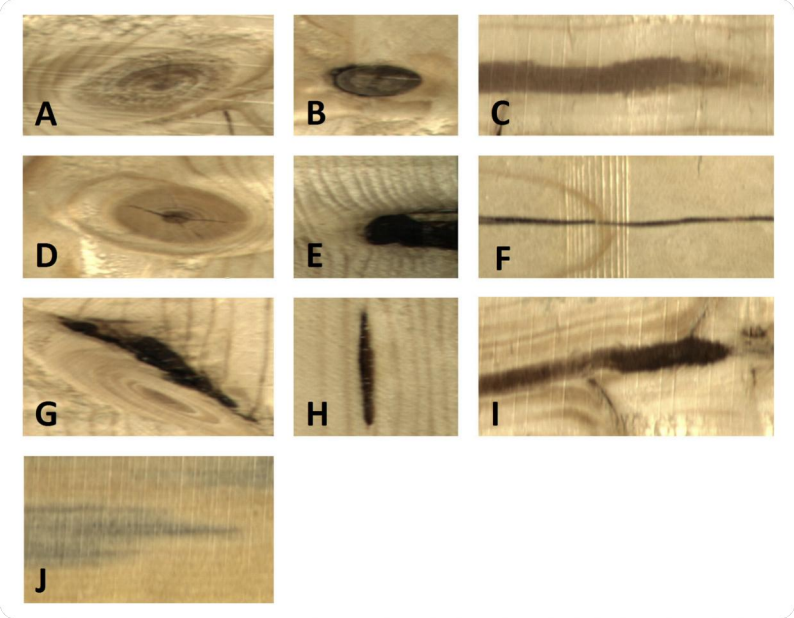


Figure 14: Some samples with live knot, dead knot and knot with crack from the datasets presented in Kodytek et al. (2021)

Therefore, due to the specific challenges of identifying wood knots on historical timber structures, it is necessary to construct one or several reasonably applicable datasets. The target dataset should include several notable features:

1. **Sufficient Data Volume:** To comprehensively evaluate and test the potential models of varying complexity, it is imperative that the dataset be of substantial size with annotated wood knots. The annotated data should also be extended through data augmentation, which helps to increase the variability of the dataset, simulate different real-world scenarios and improve the robustness of the model. The size of the dataset is critical to ensure that both simple and complex models can be adequately trained without suffering from overfitting or underfitting.

2. **High-Quality Image:** All data should be captured in high resolution to enable precise extraction of visual features from the annotated areas. Image quality is crucial for preserving fine details like texture, grain orientation, and surface irregularities in wood knots. This clarity ensures that machine learning models can accurately detect intricate patterns that may affect the structural integrity of historical timber.

3. **Variety in Scale:** The data set needs to include wood knots of different sizes, ranging from small, barely noticeable knots to large, visually prominent knots. Ensuring this diversity in scale is crucial for developing a model that can generalise across different knot types and wood structures, which will better under-

stand the scale invariance of features, allowing for more accurate detection in different historical contexts.

4. **Variety in Environmental Conditions:** Given the suboptimal lighting conditions of historical timber structures, the dataset must include images captured in various environmental settings. These structures may be poorly lit, with surfaces showing variations in texture, colour, and aging. Capturing images under different lighting conditions ensures the model's adaptability to real-world challenges, allowing it to accurately detect wood knots despite shadows, reflections, and uneven illumination common in historical preservation work.

3.1 Data Sources and Acquisition

3.1.1 Wood Workshop in Schweinfurt

Figure 15: Image capture setup with softbox and camera in Schweinfurt

A first step in the early stages of data collection was the collection of wood knots from a timber workshop in Schweinfurt for the detection task, in collaboration with the conservator and engineer. These were mainly dismantled wooden posts[4], averaging 3-5 metres in length and 10-15 centimetres in width, which had been removed

[4]A wooden post is a vertical structural element used to support the weight of a building or structure, transferring loads from above to the foundation below.

from the old historic wooden house. The old wooden house is now being rebuilt into a wood workshop. Those timber posts were carefully selected based on their historical significance and the presence of visible wood knots, which are crucial for understanding structural weaknesses.

Figure 16: Perform color calibration prior to capture to standardize color accuracy

To ensure high-quality and accurate images of the wood knots, a constant artificial lighting setup was implemented using multi-angle soft boxes and stabilized light sources. These soft boxes provided diffused, even lighting to eliminate harsh shadows and glare, which are common obstacles when capturing fine details on wood surfaces. A Nikon D850 DSLR camera equipped with a 100 mm macro lens and a ring flash was used throughout the capture process in Schweinfurt, resulting in an image resolution of 8256 $\times$ 5504 pixels in RGB colour space. This setup allowed for high-resolution images to be captured, accurately capturing the intricate details of the wood knots under standard capture conditions.

Since the wooden posts to be captured were removed from the old historic wooden house, they are all lying on the side of the intern room so that they can be captured separately with the help of two support stands. With the lighting angle and camera setup remaining the same, all the wooden posts were photographed individually from an angle that was perpendicular to the ground(Figure 15). The process of capturing these posts focuses on the specific characteristics of the knots, and therefore on capturing at different scales.

Figure 17: Image captured inside roof structure in Schweinfurt

In addition to the workshop area, where stacks of wooden posts have been carefully arranged, there is an adjoining wooden house that is also being rebuilt. The roof of the house, a particularly valuable wooden contruction, consists of intricate wooden beams[5] and joints. Given the importance of preserving these architectural features, the wooden joints in the roof and surrounding areas were meticulously photographed. Using the same camera setup and ring flash as in the workshop, high-resolution images were captured without softbox and the artificial light.

Due to the limited number of wood posts, combined with the need for high detail and tightly controlled acquisition conditions, a total of 209 high-quality, fine-grained images were captured from the available wooden posts. Approximately half of these images focused on close-up details of the wood knots, while the other half captured the wood posts on a larger scale, providing a comprehensive visual record of both intricate textures and the overall structural features. Meanwhile, due to the fast freehand photography used, about 589 images were taken in the roof space. Despite the challenging lighting conditions within the roof structure, the images captured are still extremely valuable(Figure 17). They contribute also significantly to the standard dataset for timber knot detection and meet the targeted requirements for quality and detail.

3.1.2 Dominican Church in Bamberg

Following the data collection in Schweinfurt, the Dominican Church in Bamberg was selected as the next research site to expand the data set. This provided an opportunity to gather results under real-life conditions and to test different models and strategies. The Dominican Church, originally built by the Dominican Order before 1400, is located in the heart of Bamberg's historic city centre, a UNESCO World Heritage Site. With its medieval architectural features, the church reflects

[5]A timber beam is a horizontal structural element designed to support and distribute loads across openings by transferring weight to vertical supports such as posts or walls.

the craftsmanship and religious significance of the late Middle Ages. Its vaulted ceilings, pointed arches and timber roof structure are characteristic of the period. Since 1803, following secularisation, it has ceased to function as a church and is now used as the Aula of the University of Bamberg. The building, which has undergone several restorations, continues to be an important historical landmark and provides an insight into the conservation and adaptive reuse of heritage structures.

In particular, the roof structure of Dominican Church(Figure 18) is of significant historic and structural value due to the preservation of its original finishes. Thorough conservation has ensured that most of the original timber framing remains intact, with carefully applied structural reinforcements to maintain the stability of the roof. These interventions have been carried out with minimal invasiveness, preserving the authenticity of the original materials.The roof represents a well-preserved example of medieval timber construction, making it an ideal subject for advanced analytical studies.

Figure 18: Roof structure of the Dominican Church

The roof structure of the Dominican Church consists of two primary sections: a small tower loft, containing around 20 large wooden timber structures available for data capture, and a vast, cavernous main roof space. The tower loft also serves as an exhibition space for visitors, with various displays on the history of the church and the conservation methods used. Therefore, this space will be the main space for data collection as it is more artificially lighted and suitable for collecting high quality images with wood knots. In the main roof space, the timber framework serves as the primary load-bearing structure for the historic roof, supporting its overall integrity.

In addition to the data collected from the roof of the Dominican Church, the initial survey of the church and surrounding outbuildings included a preliminary collection

of timber knot data from the roof structure of a nearby outbuilding. Although the roof structure of this outbuilding lacks adequate lighting and no artificial light sources were available during the initial survey, the data collected is still valuable. Despite the rough quality, the wood knots are still discernible to the human eye and have the potential to increase the variety and robustness of the final dataset. For the initial survey, a half-frame camera NIKON D3400 with and a full-frame camera Sony A7R IV were used to capture the entire wooden roof of the outbuilding at an appropriate scale, with a focus on the surrounding context of the wooden structure. A total of 552 images with a resolution of 4000 × 6000 pixels were captured using the Nikon camera(Figure 19), while 615 images with a resolution of 4160 × 6240 pixels were captured using the Sony camera(Figure 20). All of these images will be added to the main dataset of the church's roof structure and have undergone the same subsequent processing as the primary data collection.

(a) (b)

Figure 19: Original images from the outbuilding using NIKON D3400

(a) (b)

Figure 20: Original images from the outbuilding using Sony A7R IV

Returning to the main roof structure of the Dominican Church, the primary data were captured through multiple stages due to the design of the complete workflow for detecting wood knots on the historic timber surface, each aligned with practical application requirements. Further details on the practical application will be presented in Section 4, which covers the applied methods. In the room of the tower loft,

there are about 296 original images for the detection task, which were taken with NIKON D3400 following the similar guideline of the capture process in Schweinfurt, that the wood knots were taken in close range to obtain precise details of the knots and its surrounding wooden surface(Figure 21). Despite the fact that the room was illuminated and the camera was equipped with a ring flash, the reflection of light on the surface of the wood continues to vary from one area to another. The resulting images have a more pronounced light and dark character on wood knots, which also simultaneously meet the previous definition of an ideal dataset so that the model can learn features under more diverse conditions.

(a) (b)

Figure 21: Close-range capture on wood surface in Dominican Church using NIKON D3400

Meanwhile, due to the need for wood segmentation in the early stage of the workflow (Section 4.2), additional images were acquired specifically for timber segmentation. For this acquisition, the 3D Scanner App, installed on an iPhone 13 Pro Max, was used. The goal is to eventually run the entire workflow on low-cost devices, and the segmentation task focuses on isolating the main timber surfaces from unnecessary surrounding areas. The use of this application and the iPhone ensures more consistent results, as the capture conditions remain uniform across various data sources. The app leverages the iPhone's built-in sensors to capture data in multiple formats, such as images and point clouds, using LiDAR or photogrammetry method. As a result, 820 images at a resolution of 3024×4032 were taken in the tower loft on a larger scale (Figure 22), with each image capturing both the main target timber surface and adjacent connected timbers.

The process described above outlines the collection of raw data used for training the machine learning and deep learning models in this thesis. In addition to the previously mentioned data, a substantial amount of supplementary data, including images of timber knots, was collected from the main roof structure of the Dominican Church. This additional dataset will serve as test data to evaluate the performance of the trained models and their combinations. Moreover, this data will provide a foundation for future research. Approximately 1,500 images (Figure 23), captured using the same 3D Scanner App installed on an iPhone 14 Pro, will be processed and utilized in the continuation of this study. Unlike the earlier raw images, this

(a) (b)

Figure 22: Timber structures captured through the 3D Scanner App for segmentation task

dataset was captured at an unsystematic scale, containing both close-up images of wood knots and wider shots of the timber structures, enabling a more comprehensive analysis. This also implies further classification and processing steps will be necessary to organize the data effectively for subsequent stages of the workflow.

(a) (b)

Figure 23: Close-range images captured through the 3D Scanner App

3.1.3 Summary of whole acquisated raw data

As outlined in the acquisition process, the entire set of captured raw data is summarized in Table2. For the detection task, a total of 3,725 raw images were collected through a variety of equipment, at different scales of knots and under different lighting conditions, out of which 2,225 images will be processed, and the wood knots in these images will be manually annotated. As previously mentioned, the remaining 1,500 images will be reserved for the final testing of the model's performance and will be handled in subsequent steps. For the segmentation task, 820 raw images

are available, which will also be standardised and manually annotated for model training.

Table 2: Summary of Collected Data

Location	Device	Images	Resolution	Task	Description
Schweinfurt Workshop	Nikon D850	209	8256×5504	Detection	Close-ups and large-scale images of wooden posts and knots
Roof Space (Freehand)	Nikon D850	589	8256×5504	Detection	Freehand capture of roof space for detection tasks
Outbuilding Roof	Nikon D3400/Sony A7R IV	552/615	4000×6000/ 4160×6240	Detection	Images of the outbuilding's timber roof
Dominican Church – Tower Loft	Nikon D3400	296	4000×6000	Detection	Close-ups of wood knots in the tower loft
Dominican Church – Roof Space	iPhone 13 Pro Max	820	3024×4032	Segmentation	Timber surface segmentation using 3D Scanner
Dominican Church – Main Roof	iPhone 14 Pro	1,500	3024×4032	Detection	Mixed close-ups of knots and large-scale timber structure images

Note: The resolution is represented as width × height in pixels.

3.2 Data Processing

After data acquisition, the images from each sub-collection, captured across different phases and for different tasks, must be standardized to ensure consistent conditions. For the detection task, this standardization process facilitates the integration of data from multiple sources, thus expanding the overall dataset. Additionally, it enables a comparative evaluation of model performance when using the combined dataset versus data from individual sources. Similarly, the smaller dataset for the segmentation task will be standardized, though with different criteria and baselines tailored specifically to the requirements of the segmentation task.

3.2.1 Data standardisation

In general, all raw images collected for both detection and segmentation tasks were converted to JPG format and underwent a manual filtering process to select those that clearly depicted distinctive wood knots or similar timber structures with the target surface and its surroundings. Images were assessed based on the visibility and clarity of the wood features, ensuring that only those with well-defined knots

or comparable structural elements were retained for further analysis. Images with minor focus imperfections were retained if the wood knots remained sufficiently recognizable to the human eye.

The next step involves adjusting the filtered samples for detection task to an appropriate size that contains the ideal wood knots. Due to differences in capture scales, each sub-collection will be processed separately. Collections containing close-range images of wooden knots will be cropped into a square format, while collections with larger-scale images will first be divided into several smaller sections and then further filtered to obtain optimal samples. For instance, in the case of the extensive dataset collected from the roofs of the outbuilding of Dominican Church, the images were segmented from their original size into smaller tiles at a resolution of 640 × 640, resulting in a total of 3,767 tiles. This process was followed by an automatic selection step using a brightness threshold filter to ensure that only samples meeting the minimum brightness requirement were retained, meaning images with lower threshold values, containing excessive dark areas or lacking visible wood knots, were discarded. After this automated process, all selected tiles were manually re-examined to ensure that the final set of smaller clips were suitable for further use in model training (Figure 24). For the unique collection for the segmentation task, the image size will remain unchanged as the annotation for segmentation requires highly accurate labelling and the image should preserve the original characteristics of the image.

(a) (b) (c) (d) (e)

Figure 24: Segmented tiles from the images captured in outbuilding

Although all the images were processed into a square format, the sizes of the images in each sub-collection still vary. However, at this stage, the sizes will not be further standardized. Instead, the data will be annotated based on the segmented images from the original samples. This approach helps preserve the original characteristics of the images to some extent and allows for flexibility in adapting input sizes for different models if needed.

3.2.2 Data annotation and augmentation

For data annotation and subsequent augmentation, the online platform Roboflow was used to annotate each sub-collection individually, with different tasks for segmentation and detection assigned to the corresponding sub-collections. Roboflow is a dataset management and annotation platform dedicated to computer vision tasks, providing researchers with a set of tools to process, annotate, augment and manage

image datasets. It also provides access to various public datasets uploaded by other users through its Roboflow Universe platform. All sub-collections were individually imported by Roboflow into a workspace according to the different task. For the detection task, once the sub-collections were imported into the workspace in Roboflow, the primary class for the detection task was established as "wood_knot". For each sample, manual annotations were carried out, with bounding boxes used to label the wood knots. In cases where wood knots overlapped or were close to complex textures, extra care was taken to distinguish the knots from the surrounding features in order to maintain the accuracy of the bounding box annotations. This thorough approach was crucial to maximise the quality of the dataset and ensure that the subsequent detection models could be trained on reliable, high quality labels. For the segmentation task using the individual sub-collections, two classes were created with 'main_beam' and 'side_beam'[6] to represent the target to detect the main surface and its surrounding wood surface.

After the annotation was completed for each sample, the annotated data was saved in a separate space for further processing. Once all samples in an individual sub-collection were fully annotated, Roboflow's "Generate" function was used to create datasets tailored to specific requirements. This feature allows users not only to export the dataset with basic transformations, such as scaling from image dimensions, but also to apply various data augmentation techniques to expand the dataset and improve model performance (Figure 25).

For the wood knot detection task, the data augmentation strategy generates two to three variants of each training sample. This is achieved by applying a $\pm 15^{\circ}$ shear transform in both horizontal and vertical directions, adjusting luminance between -50% and +50%, and adding noise to 5% of the image pixels. These augmentations increase the dataset's diversity, enabling the model to learn the characteristics of wood knots under varying angles, lighting, and noise conditions, thereby enhancing the model's generalization and robustness.

For the timber surface segmentation task, only luminance adjustments between -59% and +59% and noise to 2% of the image pixels are applied. This is because segmentation tasks require precise edge and shape information, where excessive transformations could distort the object boundaries and degrade model performance.

As well as offering various data augmentation methods, Roboflow also provides the option of automatically splitting the dataset into training, validation and test sets based on specified percentages. In this work, all datasets were split into training, validation and test sets using the following percentages: 80%-10%-10%.

[6]The segmented results include both wooden beams and wooden posts. For the sake of simplicity and consistency, all such structural elements are referred to as "wooden beams" in the following texts, encompassing both vertical (posts) and horizontal (beams) wooden components.

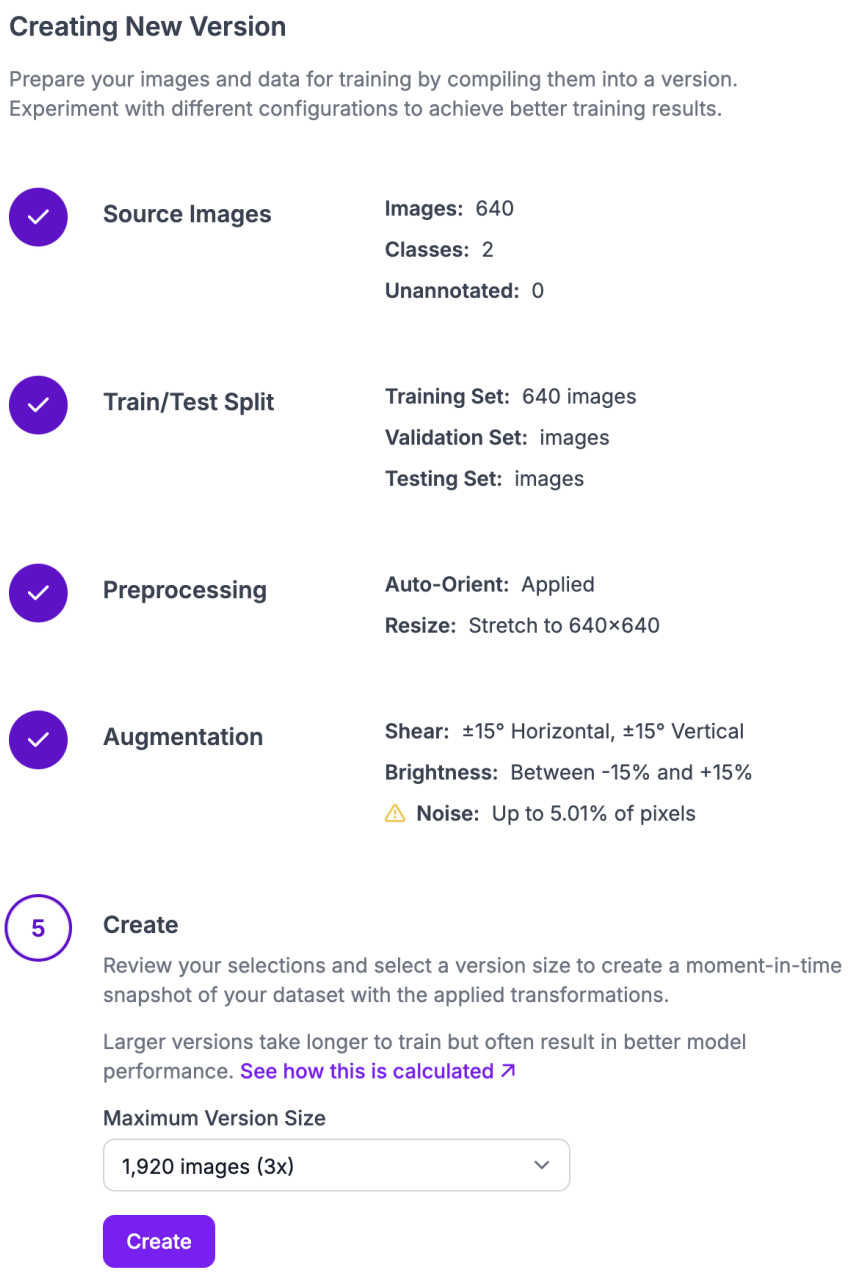


Figure 25: Interface of roboflow for data augmentation and export

3.2.3 Resulted Datasets

After the data formatting, normalisation and export steps, five datasets were finally generated, namely Det-sf-v1, Det-dominik-v1, Det-dominik-v2, Det-dominik-v3, and Seg-dominik-v1. These datasets support both detection and segmentation tasks, as summarized in Table 3 below:

The Det-sf-v1 dataset, constructed from 640 annotated images with bounding boxes captured in a standardized lighting environment at a wood workshop in Schweinfurt,

Table 3: Summary of Established Datasets

Dataset	Original annotations	Amounts with data augmentation	Resolution	Task
Det-sf-v1	640	1536	640×640	Detection
Det-dominik-v1	632	2202	640×640	Detection
Det-dominik-v2	290	870	640×640	Detection
Det-dominik-v3	-	-	640×640	Detection
Seg-dominik-v1	584	1330	1024×1024	Segmentation

Note: The resolution is represented as width × height in pixels.

was augmented to 1536 samples. This dataset, depicted in Figure 26, is tested independently due to its standardized lighting conditions.

(a) (b) (c) (d) (e)

Figure 26: Data samples from dataset Det-sf-v1

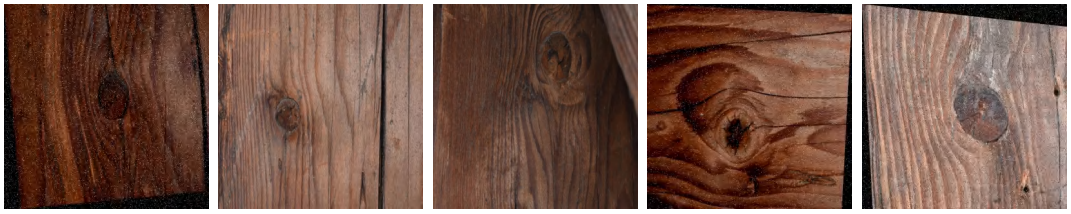
The Det-dominik-v1 dataset, consisting of 631 annotated samples captured from the outbuilding roof, was expanded to 2202 images through data augmentation, as shown in Figure 27. Similarly, Det-dominik-v2 was created from 290 annotated images captured in the tower loft and augmented to 870 samples, as illustrated in Figure 28.

(a) (b) (c) (d) (e)

Figure 27: Data samples from dataset Det-dominik-v1

The further-captured data from the main roof, although standardized, has not yet been labelled. This dataset, named Det-dominik-v3, will be used in future inspection tasks and was included in the final testing stage to evaluate the performance of the overall workflow which shown in Figure 29.

The only dataset dedicated to segmentation tasks is Seg-dominik-v1. It contains 584 annotated samples and is extended to 1330 samples. It focuses on wood surface segmentation, with two classes: main_beam and side_beam(Figure 30), designed to



(a) (b) (c) (d) (e)

Figure 28: Data samples from dataset Det-dominik-v2

(a) (b) (c) (d) (e)

Figure 29: Data samples from dataset Det-dominik-v3

distinguish the target wood surface from the surrounding areas. The higher resolution of 1024×1024 ensures accurate boundary segmentation, allowing the model to accurately detect and locate knots and defects in the wood.

(a) (b) (c) (d) (e)

Figure 30: Data samples from dataset Seg-dominik-v1

In summary, the datasets Det-sf-v1, Det-dominik-v1 and Det-dominik-v2, all containing annotated samples, were used to test the primary detection pipeline. While Det-sf-v1 was tested individually, all three datasets were later combined to further evaluate model performance on mixed datasets. The Det-dominik-v3 dataset is used to test the performance of detection models in unfamiliar samples. The Seg-dominik-v1 dataset is used to train and validate the segmentation model.

4 Methodology

This chapter outlines the methodology used to detect wood knots in historic timber structures. As already shortly mentioned in chapter 3, the whole process integrates several key stages, including image pre-processing(segmentation), perspective correction and detection.(Figure 31) The primary objective is to ensure accurate detection and classification of wood knots using advanced machine learning and image processing techniques. By addressing various challenges such as image inconsistencies, distortions and complex backgrounds, the proposed workflow aims to deliver reliable results that contribute to the broader task of preserving and analysing historic timber structures.

The following sections provide a detailed explanation of each stage in the workflow, along with in-depth introductions to YOLOv8 and Detectron2, including the corresponding mathematical descriptions throughout the model processes.

4.1 General workflow

Ideally, the system will prioritise intact wood surfaces collected from either vertical or horizontal orientations, deliberately excluding areas close to wood joints. The inherent complexity of wood joints poses a challenge to the current detection system, making these regions difficult to process accurately. Despite this limitation, addressing the intricacies of wood joints remains a critical area for future research and development to improve the robustness of the system and extend its applicability.

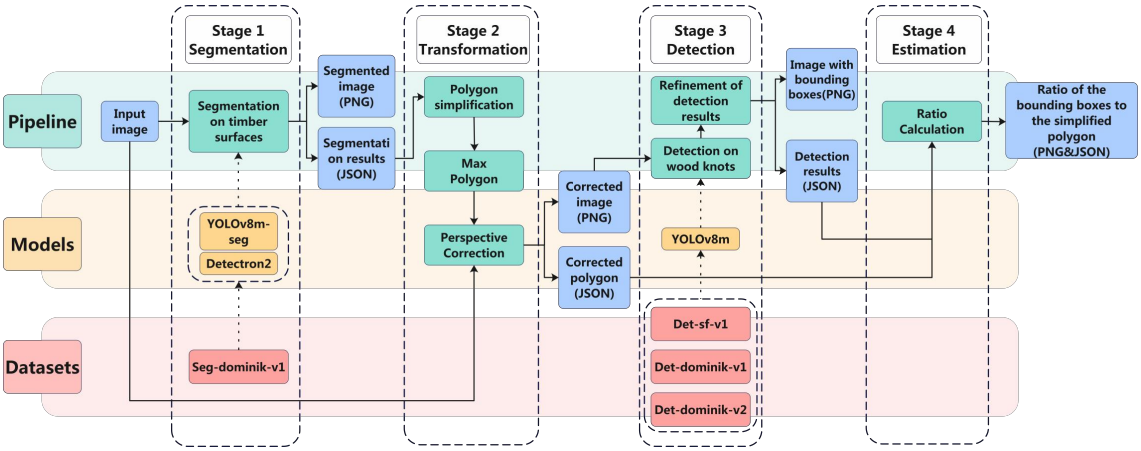


Figure 31: General workflow for AI-assisted detection on wood knots (Higher resolution in Appendix C)

To further focus on non-standard characteristics in the input images, such as variations in shooting angles and inconsistencies in the original input, preprocessing is essential to ensure uniformity. This step is critical for achieving more consistent and reliable results in the subsequent detection stage, ultimately improving the overall robustness of the process. The first step involves applying a segmentation model

to segment the entire input image, effectively dividing the wooden timber structure into multiple regions. For the segmentation task, the state-of-the-art deep learning framework Detectron2 and YOLO (you only look once) are used in experiments within the unique dataset established for wood segmentation. Although the YOLO model is widely known for its powerful detection capabilities, this study also tested its performance in a segmentation task due to the convenient method it provides. The results include both the primary area of interest, where the wood knots are located, and the surrounding surfaces.

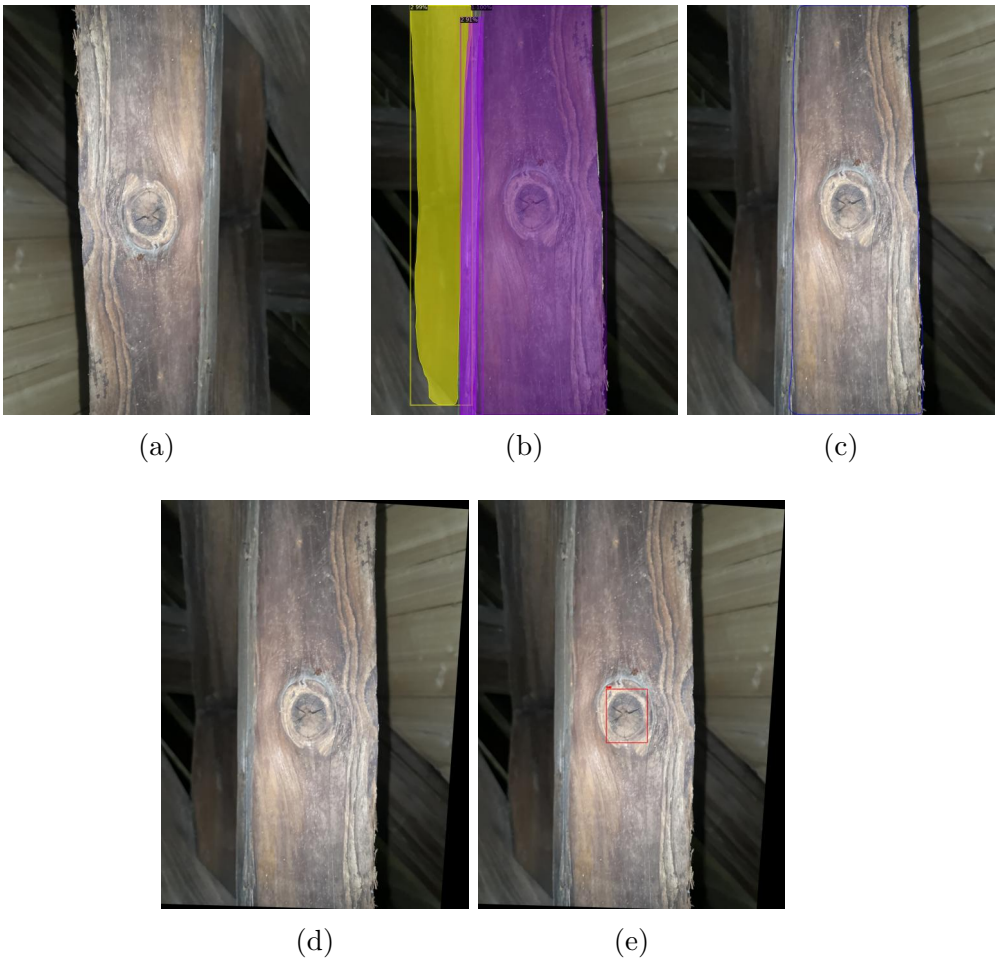


Figure 32: Example test and outputs according to the pipeline in Figure 31 from stage 1 to stage 3.
a:Original input image; b:Image with segmented timber surface (Stage 1); c:Image with segmentation polygon (Stage 2); d:Perspectively corrected image (Stage 2); e:Image with detected knot (Stage 3)

After segmentation, the boundary of the target area is simplified from a complex polygon to a quadrangular polygon, so that the image can be further applied by perspective transformation according to the quadrants. This transformation corrects distortions caused by the original image's viewpoint, ensuring that the target region is properly aligned. Perspective correction is crucial for tasks requiring geometrically

consistent input data, as it standardizes the representation of the target object, enhancing the accuracy and reliability of detection in subsequent stages.

Next, the image is processed through the primary detection pipeline using the pre-selected YOLO model. This model was chosen as a comparatively better option based on prior experimentation, where different versions of the YOLO model, varying in size and capacity, were trained and tested. These versions were evaluated on multiple datasets, comparing their accuracy, recall, and detection speed. After a comprehensive comparison, the model with relatively superior performance across these metrics was selected for the detection stage.

In this stage, the detection process specifically focuses on identifying wood knots in historical timber structures. The selected model was fine-tuned based on extensive experimentation with the collected data, as described in the previous section. Achieving accurate detection of wood knots is crucial for the subsequent assessment of the timber's condition, providing key data that supports further analysis and future conservation efforts.

Once the wood knots are detected, the final bounding boxes are combined with the perspective-transformed polygon representing the target region. This step calculates the ratio between the detected knots and the corresponding surface area of the wood, adjusted for perspective distortion. The resulting ratio provides essential information for assessing the condition of the timber, classifying the wood, and recommending appropriate preservation methods.

This multi-step process, from image segmentation to post-recognition refinement, is carefully designed to enhance the precision and efficiency of wood knots detection within the target regions of historical timber structures. By addressing challenges such as geometric distortions and complex backgrounds, the approach ensures that the detection system remains robust in a variety of real-world scenarios. The incorporation of perspective correction, along with the accurate selection of target areas, guarantees that the recognition model operates with consistent, high-quality input data. This refined workflow significantly improves the accuracy of wood knots identification and provides critical data for subsequent analyses, such as assessing timber condition and informing conservation strategies.

As outlined in the previous section on the general workflow, two deep learning framework/model were employed to accomplish the segmentation and detection tasks. The choice of model depended on the complexity of the model structure and the specific requirements of each task. Detectron2 was primarily used for training and testing in the segmentation task during the preprocessing of input images, while YOLO was also explored in a limited number of experiments for comparison purposes. For the primary detection pipeline, YOLO will be primarily experimented with due to its lightweight model architecture and fast one-shot detection. It also offers the potential to realise real-time detection in the future or to be integrated on low-cost devices such as the iPhone.

4.2 Detectron2

4.2.1 Introduction

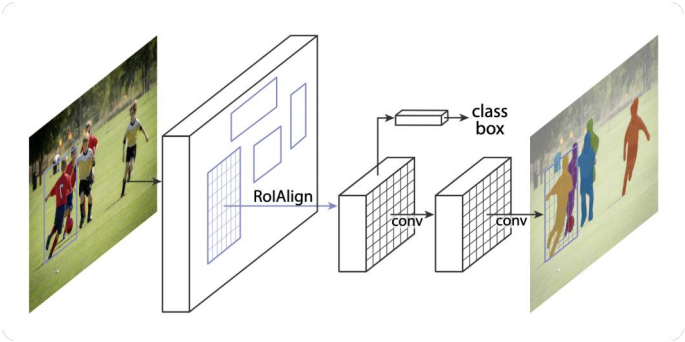


Figure 33: Mask R-CNN, He et al. (2017)

Detectron2(Wu et al., 2019), which is based on a modular and flexible framework, can be implemented with several state-of-the-art models such as Faster R-CNN(Ren et al., 2015) model and Mask R-CNN(He et al., 2017) model. The employed models based on the initial R-CNN(Region-based Convolutional Neural Networks, Girshick et al. (2016)), which can be implemented with a two-stage process to classify the objects. First, it extracts regions of interest (ROIs) from the input image using selective search(Uijlings et al., 2013). These ROIs are then further classified using convolutional neural networks to compute relevant features(Figure 34). Thanks to its region-based classifier, the early R-CNN model performances competitively in object detection task.

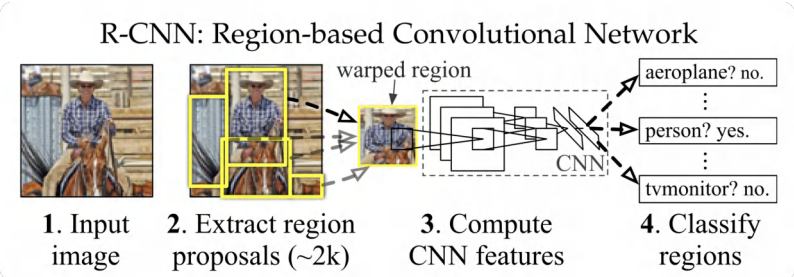


Figure 34: R-CNN, Girshick et al. (2016)

The Mask R-CNN models, primarily used in the Detectron2 model zoo for instance segmentation, were tested and applied to segment target timber surfaces during the preprocessing stage of input images. Compared to the earlier Faster R-CNN model, which focuses solely on object detection, Mask R-CNN enhances instance segmentation by incorporating an additional fully convolutional network (FCN) applied to each region of interest (ROI).(Figure 33) This additional network enables Mask R-CNN to predict precise pixel-level segmentation masks for each detected object, offering significantly more granular control and accuracy over object boundaries.

This improvement is particularly beneficial for segmenting complex structures, such as wood knots and surface details, where fine pixel-level accuracy is crucial for subsequent analysis. The ability to accurately delineate object contours at the pixel level greatly enhances the quality of segmentation results, making it an ideal choice for tasks requiring high precision.

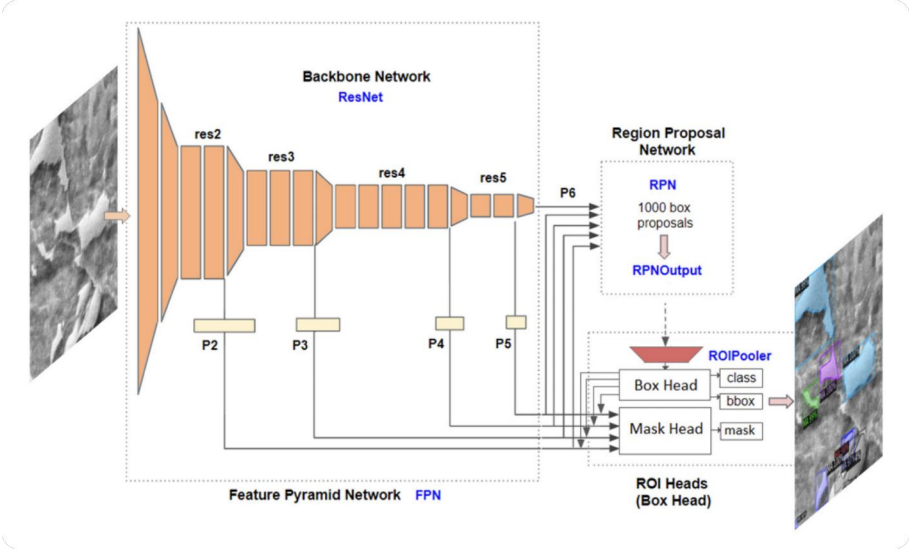


Figure 35: Schematic architecture of Detectron2, Ackermann et al. (2022)

4.2.2 Model structures

In the general framework of Detectron2 like the example in Figure 35, the process starts with a backbone network, typically using models such as ResNet (He et al., 2016) or ResNeXt (Xie et al., 2017), to extract features from the input image. ResNet primarily uses several residual blocks (Figure 36) to propagate both the features extracted through convolutional kernels and the original features to deeper layers in the network.

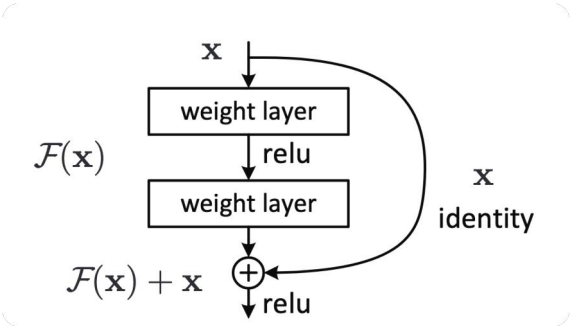


Figure 36: Residual block, He et al. (2016)

Assuming the input image is $I \in R^{H \times W \times 3}$, the Stage 1 contains only a convolutional layer followed by a pooling layer without residual block to extract the initial feature

maps. Normally, a large filter (e.g. 7×7) will be applied to quickly reduce the spatial dimensions while capturing initial feature information. The following features can be obtained in successive steps from the residual blocks in ResNet:

$$F_2 \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C_2}, \text{ for Stage } P_2$$
$$F_3 \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times C_3}, \text{ for Stage } P_3$$
$$F_4 \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times C_4}, \text{ for Stage } P_4$$
$$F_5 \in \mathbb{R}^{\frac{H}{32}\times\frac{W}{32}\times C_5}, \text{ for Stage } P_5$$

through the following convolutional calculation:

$$F_{i,j,k} = \sum I_{i+m,j+n,c} \cdot W_{m,n,c,k}$$

where $I_{i+m,j+n,c}$ is the pixel values of the input image at position $(i+m, j+n)$, $W_{m,n,c,k}$ is the value of the convolutional kernel at position (m, n), for the $c-th$ input channel, applied to produce the $k-th$ output channel. Roughly speaking, the residual block of the ResNet will process the input features according to the following equation:

$$y = F(x, W_i) + x$$

where x is the input features(I) from the previous layer, $F(x, W_i)$ represents the output of the residual block, which consists of convolutional layers parameterized by weights W_i. This enables the network to propagate both the extracted and original features to deeper layers, thereby facilitating the training of deeper networks by mitigating issues such as vanishing gradients.

The features extracted from each block of the backbone network are fed into a Feature Pyramid Network (FPN), which combines them to generate multi-scale feature maps. These multi-scale feature maps enable the model to detect objects of varying sizes. The FPN utilizes an up-sampling mechanism to compute feature maps for higher-resolution inputs, which means the FPN process operates in the reverse direction compared to the backbone architecture. The following equation is used to compute the feature maps:

$$P_l = F_l + UpSample(P_{l+1})$$

where F_l represents the features extracted from the corresponding layer of the ResNet, while P_{l+1} is the feature map from the previous (higher-level) FPN layer. Accordingly, the feature maps for different stages of the residual network are computed as follows:

$$P_5 = F_5, \text{ for Stage } P_5$$

$$P_4 = F_4 + UpSample(P_5), \text{ for Stage } P_4$$
$$P_3 = F_3 + UpSample(P_4), \text{ for Stage } P_3$$
$$P_2 = F_2 + UpSample(P_3), \text{ for Stage } P_2$$

Typically, the P_1 stage, which corresponds to the original resolution of the input image, is not computed. Since the purpose of the FPN is to address the challenge of multi-scale feature fusion, the P_1 tends to contain redundant low-level information and lacks strong semantic features. Therefore, starting the FPN from P_2 ensures a better balance between spatial resolution and semantic richness in the feature maps.

Next, a Region Proposal Network (RPN) utilizes features from both the last layer of the backbone and the multi-scale feature maps generated by the FPN at each stage to propose candidate regions that may contain objects. The RPN use k anchors (typically 9 anchors per location, with various sizes and aspect ratios) to generate a set of bounding box $B_i \in \mathbb{R}^{k\times 4}$, where k represents the number of anchors (or candidate bounding boxes) defined by the coordinates $(x_{min}, y_{min}, x_{max}, y_{max})$.

For each anchor, the RPN predicts whether the anchor contains an object using the following equation:

$$\hat{p}_i = \sigma(W_{cls} \cdot P_i),\ \hat{p}_i \in \mathbb{R}^1, W_{cls} \in \mathbb{R}^{1\times d}$$

where W_{cls} is the weight matrix for the classification layer in RPN and P_i represents the feature vector for the corresponding anchor. The anchor is likely to contain an object when the predicted value $\hat{p}_i$ approaches 1. A sigmoid function σ is applied to output a binary classification result.

In parallel, the RPN also performs bounding box regression to refine each anchor's coordinates. The bounding box adjustments are computed using the following equation:

$$\hat{t}_i = \sigma(W_{bbox} \cdot P_i),\ \hat{t}_i \in \mathbb{R}^4, W_{bbox} \in \mathbb{R}^{4\times d}$$

where W_{bbox} is the weight matrix for the regression layer in RPN and $\hat{t}_i$ represents the refined bounding box coordinates. The goal of this step is to accurately adjust the candidate anchor's bounding box to better fit the object. Finally, the RPN applies non-maximum suppression (NMS) to filter the bounding boxes and retain the top 1000 proposals with the highest likelihood of containing objects.

Following the region proposals, a ROI head processes each candidate region using the candidate bounding boxes $B_1, B_2, \ldots, B_N$ and the multi-scale feature maps P_2, P_3, P_4, P_5. Since the feature maps are extracted at various scale, so it is necessary to align the size of those feature maps so that each candidate region can have the corresponding feature map. This alignment step can be applied using the following equation with bilinear interpolation:

$$V(x,y) = \sum_{i=0}^{1}\sum_{j=0}^{1} w_{ij} \cdot P(x_i, y_j)$$

where the coordinate (x, y) represent the interpolated result within the feature map of the candidate region, x_i, y_j are the nearest pixel coordinates in the feature map and w_{ij} are the interpolation weights.

Within the ROI head, a box prediction branch (Box Head) predicts the class and bounding box of the detected objects. The class prediction layer uses $Softmax$ to predict the class of each aligned feature $F_{align,i}$ using the following equation:

$$\hat{p}_i = Softmax(W_{cls} \cdot F_{align,i}),\ \hat{p}_i \in \mathbb{R}^{N\times K}, W_{cls} \in \mathbb{R}^{K\times d}$$

where W_{cls} is the weight of the classification layer in ROI head and $F_{align,i}$ represents the i-th feature from the whole aligned feature maps F_{align}. The predicted bounding boxes $\hat{t}_i$ are obtained through following bounding box regression:

$$\hat{t}_i = W_{bbox} \cdot F_{align,i},\ \hat{t}_i \in \mathbb{R}^{N\times 4}, W_{bbox} \in \mathbb{R}^{4\times d}$$

where W_{bbox} is the weight matrix of the bounding box regression layer in the ROI head, which provides a more precise adjustment of the predicted bounding boxes compared to the regression performed in the RPN.

In the meanwhile, the aligned feature maps F_{align} will be applied not only through the box head but also through a mask head which can generate segmentation masks for objects based on these feature maps in pixel-level. For the target feature $F_{align,i,j}$ in position (i, j), a convolutional kern will be applied through the whole feature maps to calculate the weighted sum according to the size of kern size. The following equation shows the resulted mask:

$$Mask_{i,j} = \sigma(\sum_{m,n} F_{align,i+m,j+n} \cdot W_{mask})$$

where W_{mask} is the weight of mask prediction layer in ROI head. The m, n are the row and column offsets used in the convolution operation to access nearby features relative to the centre position (i, j).

4.2.3 Short summary

From the above introduction, it is clear that Detectron2's architecture is highly flexible, allowing for modular customization of different components, such as the backbone or ROI heads. The Detectron2 model zoo offers various pre-trained models tailored to different tasks, such as object detection using Faster R-CNN or RetinaNet (Lin et al., 2018) as detection heads. This flexibility also provides models of varying complexity and size, catering to diverse requirements and use cases.

On the other hand, due to its two-stage structure, the training and inference processes of models like Faster R-CNN are typically slower compared to single-stage models such as YOLO. As a result, Detectron2 is often used for professional tasks, such as medical image segmentation, where accuracy is a higher priority. However, the modular structure of Detectron2 can be challenging for beginners to configure, requiring a solid understanding of the overall model architecture, even though Meta AI provides detailed tutorials for training on custom datasets.

Therefore, Detectron2 is mainly used in the pre-instance segmentation part of the overall workflow, which helps to identify the boundaries of the target wood surface. This step requires high accuracy to ensure accurate segmentation of the wood surface, which is critical for subsequent perspective transformation and further wood knots detection. Meanwhile, YOLO was also used to perform some tests for the pre-segmentation, which will be further explained in the following sections on the YOLO model and the corresponding experiments.

4.3 YOLOv8

4.3.1 Introduction

Compared to the two-stage region-based segmentation framework of Detectron2, the YOLO framework featured a noticeably lighter structure, which employs a single convolutional neural network to detect objects and predict their locations and classes simultaneously. The original YOLO model (Redmon et al., 2016) divides the input image into a $S \times S$ grid, where each grid cell is passed through 24 convolutional layers followed by 2 fully connected layers to detect objects within a single grid cell(Figure 37). This structure transfers the detection problem to a single neural network regression problem. This approach reframed the detection task as a single neural network regression problem. While this model achieved faster recognition speeds than alternatives like R-CNN, its accuracy and multi-object detection performance were suboptimal.

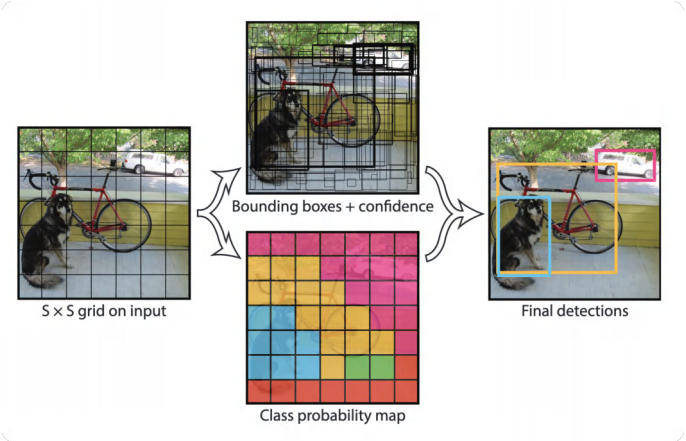


Figure 37: Process for gridding the input by YOLO, Redmon et al. (2016)

In this thesis, the focus will be on YOLOv8, which was released in early 2023 and follows a framework similar to Detectron2. YOLOv8 maintains a balance between speed and accuracy with a streamlined architecture. However, in 2024, two notable improvements to the YOLO framework emerged: YOLOv9(Wang et al., 2024b), which integrates Vision Transformers (Dosovitskiy et al., 2021) to enhance feature representation across the network, and YOLOv10(Wang et al., 2024a), which introduces a dual-pathway approach ("one2one" and "one2many") to improve both individual object and multi-object detection. Despite these advancements, YOLOv8 continues to offer a fast and simplified approach, delivering stable performance for many applications.

4.3.2 Model Structures

The architecture of YOLOv8 consists of three key modules (Figure 38): the Backbone, which extracts features from the input image at multiple scales; the Neck, which aligns the feature maps across the different stages; and the Prediction Head, which classifies the target and regresses the bounding boxes. While maintaining a streamlined structure like its predecessors, YOLOv8 introduces enhancements that significantly improve both speed and accuracy.

Assuming the input image is $I \in \mathbb{R}^{H\times W\times 3}$, which is normaly in 640×640 resolution, the feature extraction process involves progressively down-sampling and feature propagation through multiple convolutional layers, which can be expressed mathematically as follows:

$$F_{i,j,k} = \sum I_{i+m,j+n,c} \cdot W_{m,n,c,k}$$

where $I_{i+m,j+n,c}$ represents the input pixel at position $(i+m, j+n)$ and $W_{m,n,c,k}$ represents the convolutional kernel applied to the corresponding input feature channel. Unlike the ResNet backbone typically used in Detectron2, YOLO adopts a custom CSPDarknet53 backbone, which was initially introduced in YOLOv3 for feature extraction as Darknet-53(Redmon and Farhadi, 2018)). The cross-stage feature fusion in CSPDarknet53 allows the network to retain richer and more reliable information across layers while maintaining computational efficiency. A key innovation in CSPDarknet53 is the C2f (Cross Stage Partial with Focused Fusion) module, which serves as an optimized component of the architecture. The C2f module enhances the network's ability to balance feature retention and computational efficiency by focusing on selective feature fusion.

While both CSPDarknet53 and ResNet employ skip connections, they handle feature fusion differently. ResNet uses residual connections, where the output of the convolutional layer is summed with the original input features. This additive operation simplifies the learning of residual mappings and helps to solve the gradient vanishing problem, but it does not preserve the original features as explicitly as CSPDarknet53 does. CSPDarknet53 splits the input features into two parts: one passes through the convolutional layers, while the other bypasses them. The two

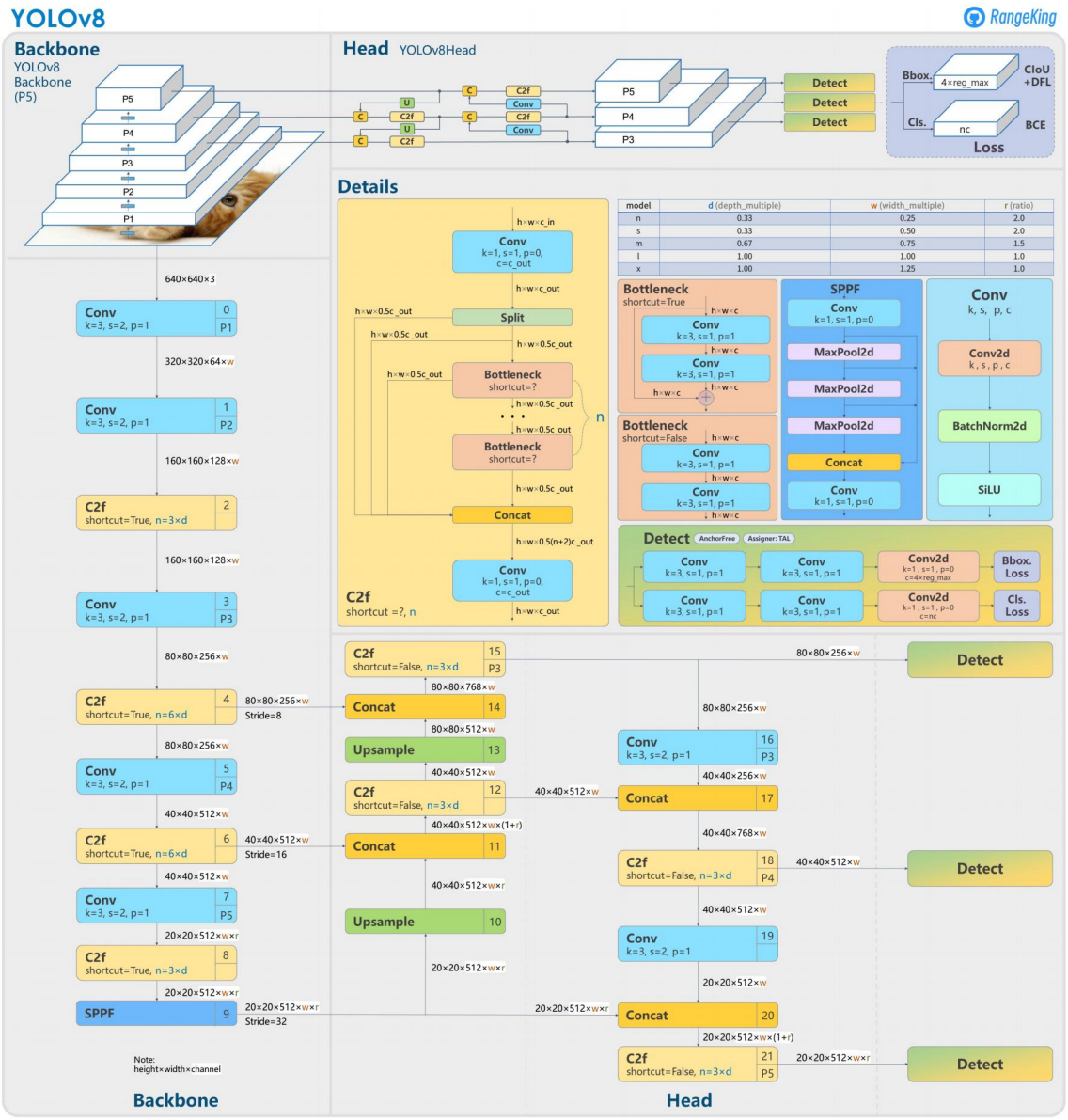


Figure 38: Detailed structure of Yolov8, King (2023)

output features are then merged using concatenation as the final export features, which can be expressed as:

$$F_{CSP} = Concat(F_{conv}, F_{skip})$$

Then in the Backbone stage, the output $F_1, F_2, \ldots, F_5$ at various scales can be represented as:

$$F_1 \in \mathbb{R}^{320\times320\times C_1}$$

$$F_2 \in \mathbb{R}^{160\times160\times C_2}$$

$$F_3 \in \mathbb{R}^{80\times80\times C_3}$$
$$F_4 \in \mathbb{R}^{40\times40\times C_4}$$
$$F_5 \in \mathbb{R}^{20\times20\times C_5}$$

These multi-scale outputs from the Backbone are then passed through the Neck structure. In this stage, lower-resolution feature maps are upsampled to align with the higher-resolution maps to ensure that multi-scale features are integrated effectively. The upsampling process can be expressed as:

$$P_{up} = UpSample(F_{low})$$

After upsampling, the feature maps are concatenated with the corresponding higher-resolution feature maps to align features from both low and high resolutions, which enhances detection performance:

$$P_{con} = Concat(P_{up}, F_{high})$$

In YOLOv8, the outputs from the stages P_3, P_4, P_5 after feature alignment in the Neck will be further passed to the Detection Head. These multi-scale feature maps are crucial as they provide plenty of information across different resolutions, enhancing object detection for both large and small objects. Unlike the anchor-based structure used in Detectron2, YOLOv8 adopts an anchor-free approach for predicting the bounding box, class, and confidence level of the detected object.

For each target, YOLOv8 predicts the bounding box using the following parameters:

$$\hat{b}_i = (x_i, y_i, w_i, h_i)$$

where (x_i, y_i) represents the coordinates of the target's centre point in the feature map. These coordinates are calculated by predicting the offset(t_x, t_y) relative to the upper-left corner of the grid cell(x_{grid}, y_{grid}):

$$x_i = \sigma(t_x) + x_{grid},\ y_i = \sigma(t_y) + y_{grid}$$

and (w_i, h_i) are the width and height of the target, expressed as a ratio(t_w, t_h) relative to the predefined image dimensions (p_w, p_h):

$$w_i = p_w \cdot exp(t_w),\ h_i = p_h \cdot exp(t_h)$$

For each predicted bounding box, YOLOv8 uses a classification layer to predict the class of the detected object:

$$\hat{p}_i = \text{Softmax}(W_{cls} \cdot P_i), \quad \hat{p}_i \in \mathbb{R}^K$$

where W_{cls} is the weight matrix for the classification layer, and K is the total number of classes. The softmax function ensures that the model outputs a probability distribution over the possible classes for each detected object.

Thus, the confidence score of the predicted object, which indicates the likelihood that an object is present in the predicted bounding box, can be calculated using the wight matrix W_{conf} as:

$$\hat{c}_i = \sigma(W_{conf} \cdot P_i)$$

Where P_i represents the feature vector for the corresponding bounding box. This confidence score helps determine the final detection by filtering out low-confidence predictions.

Based on the detection structure in YOLO, the segmentation model YoloSeg shares a similar architecture in both the backbone and neck. However, instead of solely relying on the detection head, Yoloseg introduces an additional Mask Branch over the detection head, specifically designed to generate segmentation masks for the detected objects directly. This Mask Branch outputs pixel-level masks for each object in addition to the bounding boxes and class predictions from the detection head.

4.3.3 Short summary

Thanks to its one-stage feature extraction and object detection pipeline, YOLOv8 is ideal for real-time applications, such as processing video frames captured by a camera. It can also be integrated into low-cost devices with limited processing power, like smartphones, making fast and convenient detection processes feasible for future applications. In general, it is also computationally efficient due to its global prediction of all bounding boxes simultaneously within the entire image and the single forward pass through the entire network.

However, YOLO does have some limitations. It may struggle with accurately localizing small objects, as the grid system used for predictions assigns a fixed number of bounding boxes. Small objects that span across multiple grid cells can be ignored or poorly detected. Additionally, compared to models like Faster R-CNN, YOLO tends to produce less precise bounding boxes, particularly for objects with irregular shapes. Another challenge arises when YOLO encounters objects with aspect ratios that differ significantly from those it has been trained on, leading to suboptimal detection results even for objects that appear similar to human vision. This aspect ratio sensitivity remains a valuable issue to be addressed and will be further explored in the analysis of the experimental results.

In summary, the single-pass detection mechanism of YOLO makes it one of the fastest object detection models available. Official documentation highlights its balanced performance in terms of accuracy and speed. Further optimizations of the

YOLO architecture, specifically tailored for detecting wood knots on historical timber surfaces, are possible. For instance, adding additional detection heads to track biological features along the growth of wood knots could enhance its effectiveness in such specialized tasks.

5 Experiments

The experiments using the deep learning frameworks Detectron2 and YOLOv8 will focus on two main stages of the workflow: segmentation of the target timber surface and the primary detection pipeline for identifying wood knots. As introduced in Section 3, one dataset for segmentation task and three basic datasets for detection task will be employed in the experiments. Additionally, the performance of both the segmentation and detection models will be tested on an additional unannotated dataset, which contains images captured from the main roof of the Dominican church.

The evaluation standards for these experiments will include various loss functions and performance metrics, such as accuracy, precision, and recall, along with visual inspection of the results. The goal of these experiments is to comprehensively assess the strengths and weaknesses of the models through both quantitative and qualitative analyses on the collected dataset. Additionally, these evaluations will serve as a reference point for subsequent in-depth optimizations, providing a benchmark for improving model performance in future research.

For all experiments, a virtual machine with a CPU (62.5 GB RAM, 24 cores) and GPU (24 GB RAM) will be utilized. The datasets will be stored on an 8 TB HDD, while the training and evaluation tests will be executed on a 512 GB SSD for faster processing.

5.1 Testing of Segmentation Models

5.1.1 Introductions of Model Setups

The Detectron2 model zoo offers a wide range of models for instance segmentation. Based on the performance test data provided by Detectron2, the models mask_rcnn_R_50_FPN_3x and mask_rcnn_R_101_FPN_3x were selected for further training and validation on wood timber surface segmentation. Both models are based on the Mask R-CNN architecture, which extends Faster R-CNN by adding a branch for predicting segmentation masks, as introduced earlier. The mask_rcnn_R_50_FPN_3x model uses ResNet-50 (denoted as R_50) with 50 layers for hierarchical feature extraction from the input image. In contrast, the mask_rcnn _R_101_FPN_3x model uses the deeper ResNet-101 backbone (denoted as R_101), which has 101 layers, providing more depth and leading to better feature extraction and overall accuracy, particularly for more complex datasets.

As mentioned in the previous section, both models utilize the Feature Pyramid Network (FPN) to detect objects at multiple scales by combining high-level semantic features with low-level spatial information. These pre-trained models were trained with a 3x schedule, meaning they were trained for three times the default number of iterations, which corresponds to approximately 37 epochs on the COCO-seg dataset. The COCO-seg dataset is derived from the original COCO dataset(Lin

et al., 2014), which contains 330K images. For pre-training in Detectron2, two subsets of the COCO-seg dataset were used: Train2017, consisting of 118K images for instance segmentation, and Val2017, which contains 5K images with corresponding annotations for evaluation.

For the YOLOv8 experiments, there are fewer pre-trained models available from the official model repository. YOLOv8 offers five different models, each varying in size and performance. The model YOLOv8m-seg was selected because it strikes a good balance between mean Average Precision (mAP), inference speed, and the number of parameters. While it is being used to compare results against Detectron2, only the training results will be reported, as the focus here is on evaluating model performance rather than direct comparison in testing.

The Figure 39 outlines the dataset preparation, augmentation process, and experimental setup for training and evaluating segmentation models on the Seg-dominik-v1 dataset using different models based on Detectron2 and YOLO.

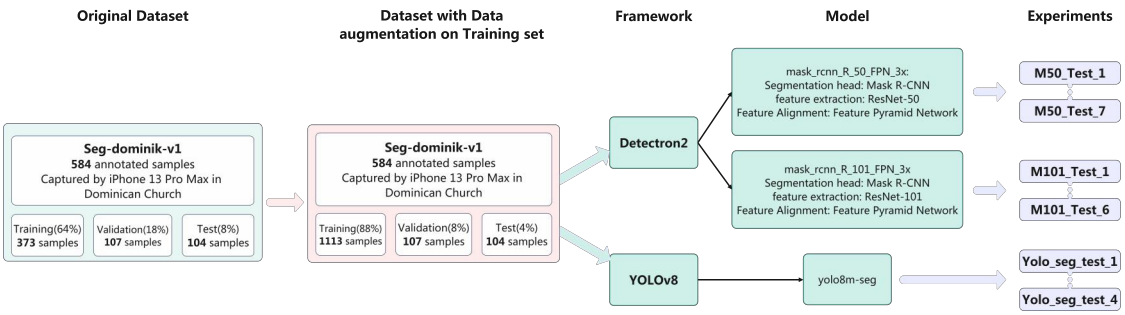


Figure 39: Pipeline for dataset augmentation of Seg-dominik-v1 and experimental evaluations of Detectron2 and YOLO models for timber surface segmentation

5.1.2 Experiments Results

In the training strategy using both mask_rcnn_R_50_FPN_3x and mask_rcnn_R_101_FPN_3x, the initial learning rate was set to 0.00025. The WarmupMultiStepLr mechanism, which combines warm-up and multi-step learning rate decay, is employed to optimize the learning rate schedule. This approach can significantly improve both model performance and convergence speed.

During the experiments in Detectron2, the warm-up phase linearly increases the learning rate from 0.00000025 to 0.00025 over the first 1000 iterations. Since no specific learning rate decay steps are defined, the learning rate will remain at 0.00025 after the warm-up phase. This decision was made because the model's performance on the dataset was unknown prior to the experiments. By setting a smaller learning rate initially, the performance of Detectron2 on the dataset can be evaluated as the number of training epochs increases. The batch size was set to 16, and with 1113 images in the training set, it takes approximately 70 iterations to complete one epoch, where the entire dataset is processed once.

Table 4: Segmentation tests using mask_rcnn_R_50_FPN_3x on augmented Segdominik-v1 dataset

Tests	Iterations	Epoch	Training Time	Total loss	Cls reg loss	Box reg loss	Mask loss	Accuracy
M50_Test_1	700	10	0:12:38	0.7875	0.1577	0.1949	0.2993	0.865
M50_Test_2	1400	20	0:28:06	0.6332	0.1436	0.1608	0.2306	0.897
M50_Test_3	2800	40	0:55:14	0.5028	0.1102	0.1329	0.1955	0.911
M50_Test_4	4174	60	1:21:42	0.4179	0.09532	0.1169	0.1655	0.923
M50_Test_5	5565	80	1:54:54	0.3803	0.07595	0.1007	0.1583	0.929
M50_Test_6	6957	100	2:22:53	0.3669	0.06863	0.09949	0.1487	0.935
M50_Test_7	8348	120	2:51:26	0.3416	0.06461	0.1034	0.1435	0.938

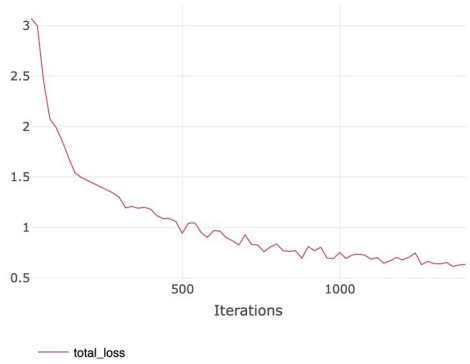
The above Table 4 presents the training results using mask_rcnn_R_50_FPN_3x model, demonstrating how different loss components evolve with increasing iterations. The total loss will be calculated through the $loss\_cls$, which evaluates the model's performance on the classification task, the $loss\_box\_reg$, which measures the accuracy of the predicted bounding boxes relative to the ground truth, the $loss\_mask$, which assesses how well the model predicts object masks, specific to the Mask R-CNN framework, and the $loss\_rpn\_cls$ and the $loss\_rpn\_loc$, which are outputs from the Region Proposal Network (RPN) that measure how accurately it classifies anchors and predicts the bounding box locations, can be calculated as following:

$$\begin{aligned} total\_loss = {} & w_1 \times loss\_cls + w_2 \times loss\_box\_reg \\ & + w_3 \times loss\_mask + w_4 \times loss\_rpn\_cls \\ & + w_5 \times loss\_rpn\_loc \end{aligned}$$

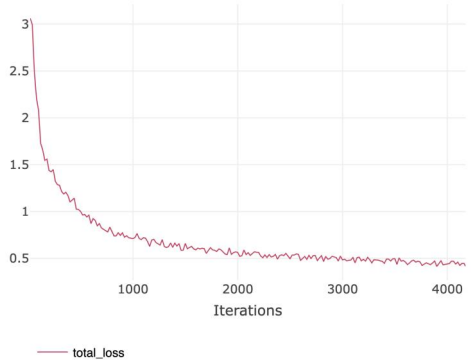
where the $w_1, w_2, \ldots, w_5$ are weights to balance the contribution of each loss component to the total loss.

The test results show a clear decrease in total loss, from an initial value of 0.7875 to 0.3416, indicating a steady improvement in the model's performance as training iterations and time increase. From Figure 40a, which represents the total loss at epoch 20 (1400 iterations), to Figure 40b at epoch 60 (4174 iterations), the total loss decreases rapidly, demonstrating significant learning progress in the early stages. As seen in Figure 40c at epoch 100 (6957 iterations), the rate of loss reduction slows down, although the model continues to improve. By epoch 120 (8348 iterations), shown in Figure 40d, the learning curve begins to flatten, indicating that the model is approaching convergence. This gradual flattening suggests that while further training leads to slight improvements in accuracy, the returns are diminishing.

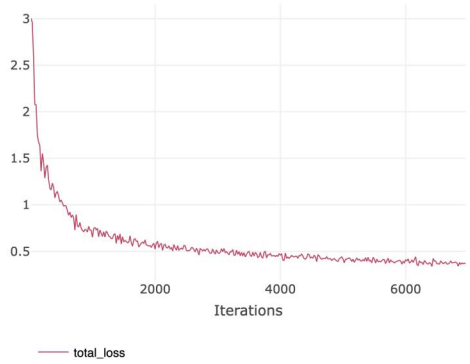
In Figure 41 (classification loss), Figure 42 (box regression loss), and Figure 43 (mask loss), the gradual flattening of the loss curves at 8348 iterations shows that the model is progressively optimizing its parameters. However, each type of loss



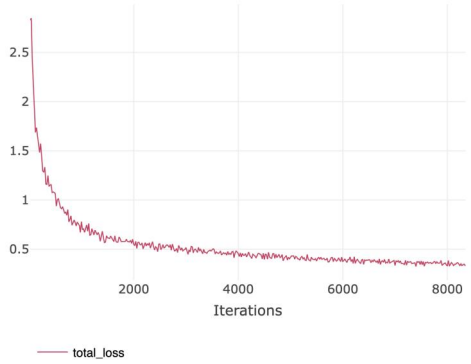
(a) Total Loss from M50_Test_2 with 1400 training iterations



(b) Total Loss from M50_Test_4 with 4174 training iterations



(c) Total Loss from M50_Test_6 with 6957 training iterations



(d) Total Loss from M50_Test_7 with 8348 training iterations

Figure 40: Total Loss from tests with different training Iterations using mask_rcnn_R_50_FPN_3x model from Detectron2

exhibits unique behaviour that highlights the different complexities of the tasks involved in training.

The classification loss decreases sharply in the early stages of training, forming a convex curve. This indicates that the model learns to discriminate between the two wood surface classes relatively quickly. As the classification task focuses on identifying categorical differences, which are easier to separate than in regression tasks, the model can make significant improvements in a short time. After the initial rapid drop, the classification loss curve flattens, indicating that the model has learned most of the features needed for classification.

Unlike the classification loss, the box regression loss shows a unique trend, starting with a temporary increase during the warm-up period (first 1000 iterations). This initial increase can be attributed to the model being cautious with its updates due

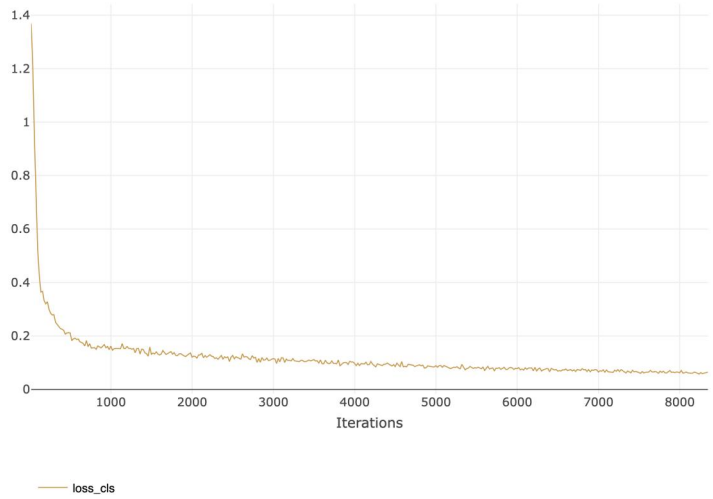


Figure 41: Classification loss from M50_Test_7 with 8348 training iterations

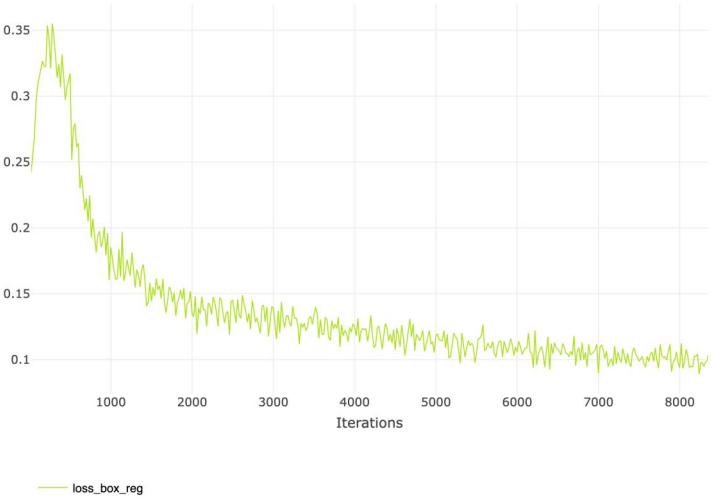


Figure 42: Box regression Loss from M50_Test_7 with 8348 training iterations

to the low learning rate in the warm-up period. During these early iterations, the model makes large adjustments to improve the bounding box predictions, which can temporarily increase the loss. As the learning rate increases, the model begins to learn more effectively, leading to a gradual reduction in the regression loss. The consistent oscillations in box regression loss throughout training highlight the complexity of accurately predicting continuous values such as bounding box positions. Small deviations in object positioning can cause significant loss spikes, and the different sizes and shapes of the objects in different mini-batches introduce additional instability. Nevertheless, the overall downward trend indicates that the model is

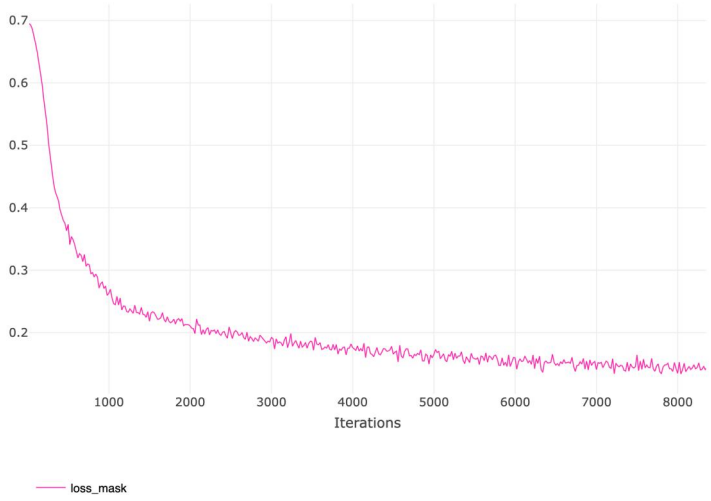


Figure 43: Mask Loss from M50_Test_7 with 8348 training iterations

improving, although the oscillations suggest that bounding box regression is a challenging task that takes longer to stabilise.

The mask loss decreases steadily, but at a slower rate than the classification loss. This is expected because the segmentation task involves accurately predicting a mask for each pixel of an object, which is inherently more complex than classifying an entire image. The segmentation task requires precise accuracy at the pixel level, which requires more iterations for fine-tuning. While the mask loss decreases over time, the curve shows that it takes longer to achieve high segmentation accuracy. As training progresses, the rate of decrease slows down, indicating that the model is entering a more refined phase of optimisation, where the focus shifts from broad improvements to fine-tuning the exact pixel-level segmentation.

The accuracy of the segmentation task, where positive samples are defined by an IoU (Intersection over Union) threshold greater than 0.5, shows a consistent and gradual improvement. After the warm-up stage, the accuracy increases steadily from 0.883 at 1000 iterations to 0.938 at the end of the training process. As the number of iterations increases, both false negatives and false positives progressively decrease. This indicates that the model is getting better at dealing with classification errors early in the training, which in turn improves the instance segmentation performance. Additionally, both false positives and false negatives stabilise after about 1000 iterations, indicating that the model has reached a more refined learning state.

To further evaluate the performance of the trained models, images from the test dataset were evaluated using models at different training iterations. Figure 45 presents results from the model after 4174 iterations, Figure 46 shows the results after 6957 iterations, and Figure 47 displays the results from the model after 8348 iterations. While the model trained for a longer duration demonstrates improved segmentation quality and accuracy, it still struggles with accurately segmenting com-

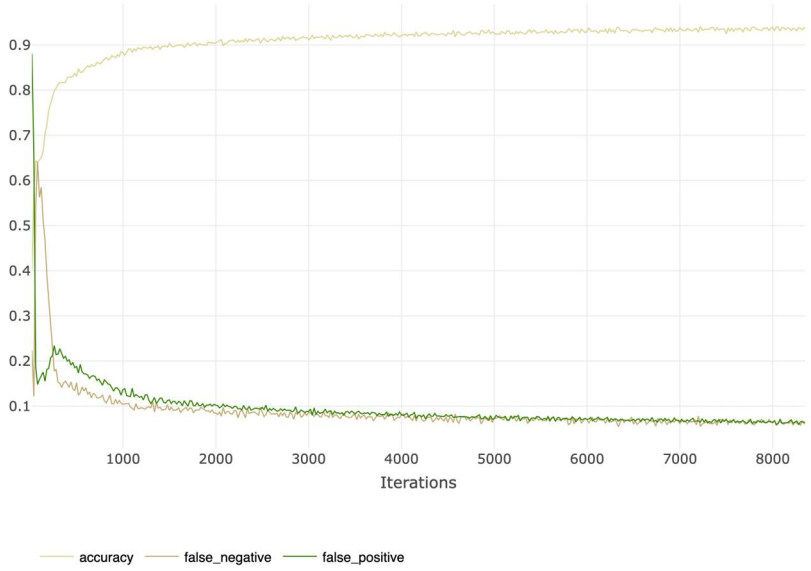
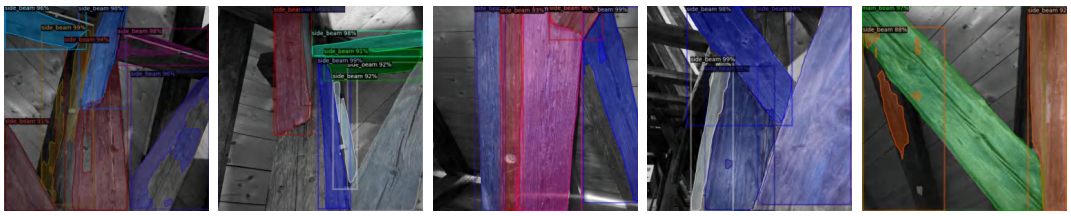


Figure 44: Mask R-CNN accuracy from M50_Test_7 with 8348 training iterations

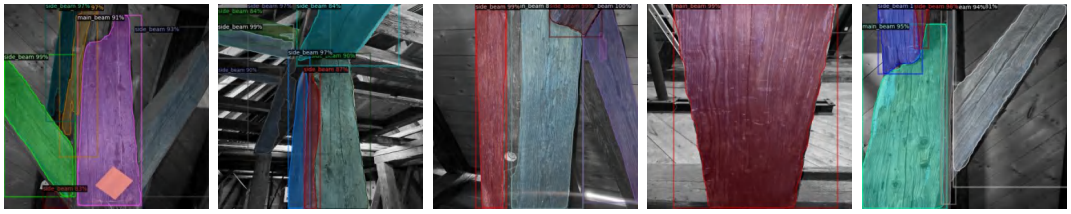
plex structures. However, as previously discussed, the primary application of this segmentation task is focused on close-range, perpendicular segmentation of timber surfaces, as illustrated in Figure 46d. Therefore, further evaluation of performance using different models under these specific conditions is required for a more comprehensive analysis.



(a) (b) (c) (d) (e)

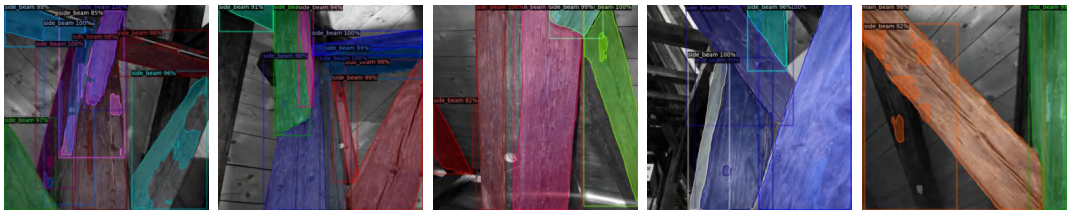
Figure 45: Images tested on test set of augmented Seg-dominik-v1 with model from M50_Test_4 with 4174 training iterations

In the next stage of experiments, the model mask_rcnn_R_101_FPN_3x was tested using a similar training strategy as in the previous experiments. However, the training iterations for mask_rcnn_R_101_FPN_3x begins from 2800 iterations to 20,869 iterations spanned from 2800 to 20,869 iterations, covering approximately 300 epochs. Since ResNet-101 is a deeper architecture compared to ResNet-50, it typically requires more training time to fully capture complex features and improve overall accuracy. This is because the increased depth of ResNet-101 allows it to learn more detailed hierarchical representations, but it also means the model needs longer train-



(a) (b) (c) (d) (e)

Figure 46: Images tested on test set of augmented Seg-dominik-v1 with model from M50_Test_6 with 6957 training iterations



(a) (b) (c) (d) (e)

Figure 47: Images tested on test set of augmented Seg-dominik-v1 with model from M50_Test_7 with 8348 training iterations

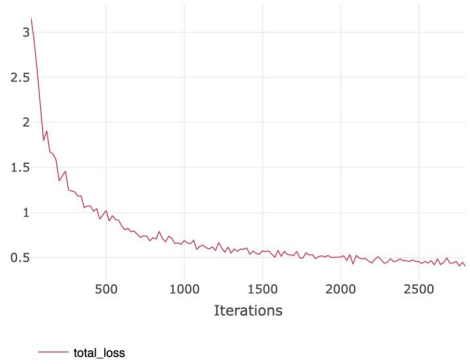
ing periods to converge effectively and to generalize well across a variety of input data. The training results were shown at Table 5.

Table 5: Segmentation tests using mask_rcnn_R_101_FPN_3x on augmented Seg-dominik-v1 dataset

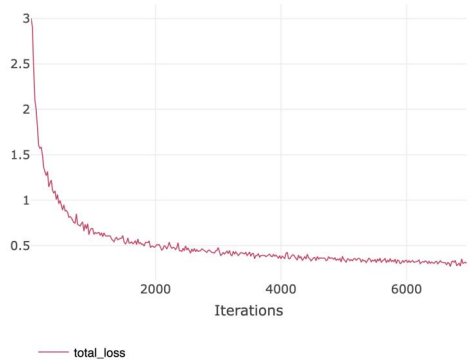
Tests	Iterations	Epoch	Training Time	Total loss	Cls reg loss	Box reg loss	Mask loss	Accuracy
M101_Test_1	700	10	0:16:34	0.6827	0.1565	0.1863	0.2544	0.891
M101_Test_2	2800	40	1:07:51	0.4008	0.09174	0.1089	0.1639	0.927
M101_Test_3	5565	80	2:15:30	0.3474	0.06191	0.08828	0.1442	0.939
M101_Test_4	6957	100	2:49:29	0.3036	0.05163	0.08733	0.1337	0.940
M101_Test_5	13913	200	5:41:13	0.2082	0.0284	0.05899	0.1012	0.954
M101_Test_6	20869	300	8:32:52	0.2101	0.0247	0.05628	0.1022	0.956

Compared to the training results from mask_rcnn_R_50_FPN_3x, the mask_rcnn_R_101_FPN_3x consistently demonstrates better overall performance at the same iteration count, particularly in terms of box regression loss and mask loss. As expected, the training took significantly longer due to the deeper architecture of the feature extraction network, which requires more time to capture finer details from the input images. Additionally, the loss trends observed with ResNet-101 exhibit similar oscillations to those seen in the previous experiments using mask_rcnn_R_50_FPN_3x, including classification loss, box loss, and mask loss. For instance, the behavior of the different losses at 13,913 iterations (Figure 49) reflects these patterns.

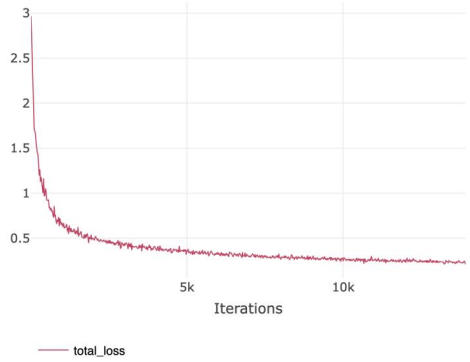
When examining the results of the training, it is clear that the model achieves close to optimal performance after 200 epochs, with only a minimal improvement ob-



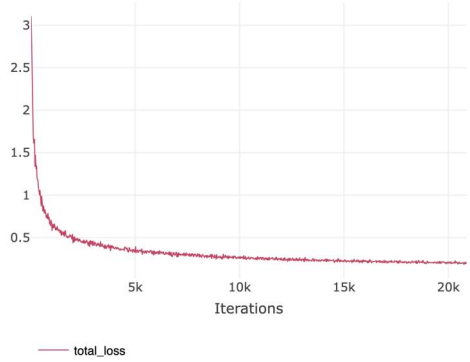
(a) Total Loss from M101_Test_2 with 2800 training iterations



(b) Total Loss from M101_Test_4 with 6957 training iterations



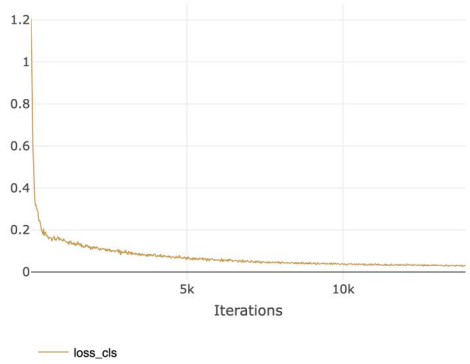
(c) Total Loss from M101_Test_5 with 13913 training iterations



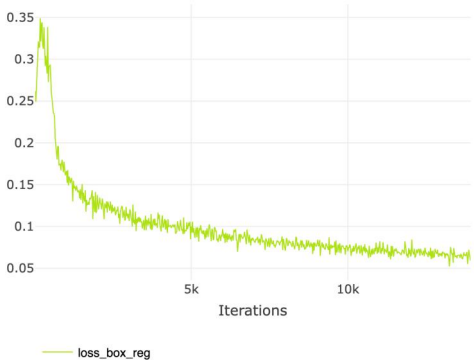
(d) Total Loss from M101_Test_6 with 20869 training iterations

Figure 48: Total Loss from tests with different training Iterations using mask_rcnn_R_101_FPN_3x model from Detectron2

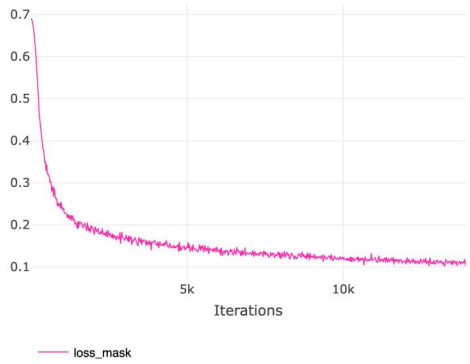
served when comparing the results between 200 and 300 epochs. This suggests that the model is likely to have reached convergence at 200 epochs, and that extending training for an additional 100 epochs resulted in only marginal performance gains. This plateau in improvement suggests that additional training did not significantly improve the model's ability to generalise. At this stage, to avoid unnecessary consumption of computational resources the early stopping mechanism is needed, as further training beyond 200 epochs did not yield proportional benefits. Early stopping could have been used to stop training when performance reached a plateau, preventing overfitting and optimising resource efficiency. Implementing this mechanism in future experiments would help to avoid resource waste and increase training efficiency.



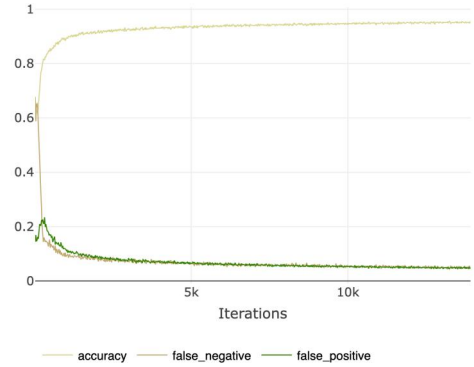
(a) classification loss at 13913 iterations



(b) Box regression Loss at 13913 iterations



(c) Mask Loss at 13913 iterations



(d) Mask R-CNN ACC at 13913 iterations

Figure 49: Total Loss across Different Iterations using mask_rcnn_R_101_FPN_3x model from Detectron2

The performance of all models trained with mask_rcnn_R_101_FPN_3x was tested on several images from the test set. Figure 50 displays the segmentation results for the timber surface. Compared to the model trained with mask_rcnn_R_50_FPN_3x in Figure 47 the segmentation edges produced by mask_rcnn_R_101_FPN_3x are noticeably more precise. However, in some instances, unwanted background areas or similar regions are still mistakenly included in the masks. Despite these occasional inaccuracies, the overall model performance improved with mask_rcnn_R_101_FPN_3x, particularly after sufficient training. Although some challenging areas still require further refinement, the deeper ResNet-101 backbone allows for better feature extraction, resulting in more accurate segmentation.

Following after the trainings experiments, the YOLOv8m-seg model is also used to further evaluate the performance of a one-stage model, which has a balanced model performance among all offered segmentation models. The Table 6 and Table 7 present the evaluation results for the predicted bounding boxes and masks over

(a) (b) (c) (d) (e)

Figure 50: Images tested on test set of augmented Segdominik-v1 using model from M101_Test_5 with 13913 training iterations

different training epochs within all classes. In addition to training time, the evaluation focuses on three key metrics: precision, recall, and mAP50, which are critical for assessing model performance both by Segmentation tests and Detection tests. Precision measures the proportion of positive predictions that are correct, which emphasizes the correctness of the positive predictions and can be formulated as:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

Recall indicates the proportion of actual positive instances that are correctly identified by the model. It focuses on the model's ability to detect all relevant instances and is defined as:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

And mAP50 (Mean Average Precision at IoU = 0.5) evaluates the model's performance by considering the precision and recall across multiple thresholds or categories. The value "50" refers to the IoU (Intersection over Union) threshold, meaning that a prediction is considered correct if the overlap between the predicted and ground truth bounding boxes is greater than 50%.

$$\text{mAP}_{50} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i$$

These metrics provide a comprehensive understanding of the model's effectiveness in detecting objects and segmenting regions accurately. They will be evaluated intensively by the experiments in YOLO, both for the Segmentation task and for the further Detection experiments. Especially the test results from the individual class *main_beam* and *side_beam* are also listed in Table 8 and Table 9, which can also track the different training results for each class since the used dataset are not balanced in class amounts.

Table 6: Segmentation tests using YOLOv8m-seg on augmented Seg-dominik-v1 dataset (Box metrics)

Tests	Planned epochs(trained)	Training Time	Box precision	Box recall	Box mAP50
Yolo_seg_test_1	50(50)	0:20:09	0.733	0.771	0.787
Yolo_seg_test_2	100(100)	0:38:16	0.746	0.755	0.800
Yolo_seg_test_3	150(146)	0:55:44	0.744	0.745	0.798
Yolo_seg_test_4	200(154)	0:59:20	0.726	0.784	0.780

Table 7: Segmentation tests using YOLOv8m-seg on augmented Seg-dominik-v1 dataset (Mask metrics)

Tests	Planned epochs(trained)	Training Time	Mask precision	Mask recall	Mask mAP50
Yolo_seg_test_1	50(50)	0:20:09	0.731	0.769	0.779
Yolo_seg_test_2	100(100)	0:38:16	0.744	0.749	0.787
Yolo_seg_test_3	150(146)	0:55:44	0.737	0.739	0.790
Yolo_seg_test_4	200(154)	0:59:20	0.720	0.778	0.769

Table 8: Segmentation tests using YOLOv8m-seg on augmented Seg-dominik-v1 dataset (main_beam)

Tests	Class	Box precision	Box recall	Box mAP50	Mask precision	Mask recall	Mask mAP50
Yolo_seg_test_1	main_beam	0.803	0.893	0.886	0.803	0.893	0.886
Yolo_seg_test_2	main_beam	0.798	0.853	0.894	0.800	0.853	0.887
Yolo_seg_test_3	main_beam	0.777	0.880	0.902	0.777	0.880	0.900
Yolo_seg_test_4	main_beam	0.786	0.880	0.890	0.786	0.880	0.885

Table 9: Segmentation tests using YOLOv8m-seg on augmented Seg-dominik-v1 dataset (side_beam)

Tests	Class	Box precision	Box recall	Box mAP50	Mask precision	Mask recall	Mask mAP50
Yolo_seg_test_1	side_beam	0.663	0.648	0.688	0.660	0.645	0.672
Yolo_seg_test_2	side_beam	0.694	0.657	0.706	0.688	0.645	0.687
Yolo_seg_test_3	side_beam	0.711	0.610	0.695	0.697	0.597	0.681
Yolo_seg_test_4	side_beam	0.667	0.689	0.670	0.654	0.677	0.653

From the general tests using YOLOv8m-seg, it is evident that the performance trends are more complex compared to the steady improvements observed in the Detectron2 tests. With increasing training epochs, the best general box precision (0.746) and box mAP50 (0.800) are achieved in Yolo_seg_test_2 (100 epochs). Meanwhile, the highest box recall (0.784) is observed in Yolo_seg_test_4 after 154 training epochs. This suggests that the model reaches optimal performance in bounding box prediction around 100 epochs, and further training may lead to overfitting in the timber surface detection task.

For the class main_beam, both box precision and box recall slightly decrease as training time increases, but there is a notable improvement in the mAP50 value. This implies that although the overall detection accuracy of the bounding boxes slightly drops, the model becomes better at identifying objects with higher confidence in this class.

The results for the class side_beam are more challenging to interpret. The best box precision is 0.711 in Yolo_seg_test_3, while the highest box recall is 0.689 in Yolo_seg_test_4, and the best mAP50 is 0.706 in Yolo_seg_test_2. These variations suggest that despite potential overfitting with longer training times, the model is still learning to distinguish features of this class, as indicated by the increase in box precision. However, the unbalanced sample size of the side_beam class may also have significantly impacted the results, complicating the evaluation of model performance for this specific class.

The evaluation metrics for mask generation in YOLO are influenced by the performance of bounding box prediction because YOLO's mask generation is dependent on its one-shot detection process. In YOLO, the mask is generated based on the detected bounding boxes, so any errors in box prediction can directly affect mask accuracy. In contrast, Detectron2 generates masks from aligned feature maps after region proposals, which makes its mask prediction process somewhat independent of bounding box performance. This multi-stage approach in Detectron2 also allows for separate optimization of detection and segmentation tasks.

(a) (b) (c) (d) (e)

Figure 51: Images tested on test set of augmented Seg-dominik-v1 using model from Yolo_seg_test_4 with 154 training iterations

Figure 51 presents the test results using the Yolo_seg_test_4 model. Compared to the results shown in Figure 50, which are based on the mask_rcnn_R_101_FPN_3x model, the YOLO-based results are comparable with well-segmented timber surfaces and clear mask boundaries. Although some areas, such as the central area in Figure 51b, have unclearly defined boundaries, considering the shorter training time and lower resource consumption, the performance of the YOLOv8m-seg model shows significant potential for real-time segmentation under limited computational resources. Since YOLO's segmentation results heavily rely on accurate detection, future improvements could focus on enhancing the model's detection capabilities to further improve segmentation performance.

5.1.3 Summary of Segmentation tests

For the segmentation tasks, both Detectron2 and YOLOv8m-seg provide valuable insights into training performance and testing on unfamiliar samples. Detectron2, with its two-stage detection and segmentation process, offers greater flexibility in modular architecture and can be easily integrated with additional mechanisms. Since its classification, detection, and segmentation processes are decoupled, improvements focus primarily on the acquisition of appropriate aligned feature maps. For more complex segmentation scenarios, the Detectron2 architecture can deliver more precise and detailed segmentation results.

On the other hand, when real-time segmentation and detection are required, the YOLOv8m-seg model offers a balanced solution, especially under limited local computing resources. For instance, in applications deployed on low-cost devices like smartphones, the lightweight architecture of YOLO provides significant benefits. Its efficiency and speed make it ideal for real-time processing, delivering solid performance without the need for extensive computational power, which is crucial for future mobile-based applications.

5.2 Testing of Detection Models

5.2.1 Introductions of model setup

As discussed in Section 3, a total of three basic datasets were collected, standardized, and annotated for the primary detection pipeline. Both the Det-sf-v1 dataset with only original annotated samples and with data augmentation were individually trained using the YOLO framework due to its systematically controlled acquisition conditions and high sample quality. Additionally, augmented mixed datasets created from these three basic datasets were also trained and evaluated to assess model performance.

Similar to the YOLOv8m-seg model for segmentation, the YOLOv8m model for detection was chosen to run further tests on various datasets because of its balanced performance (Figure 52). With an initial learning rate of 0.001 and a final learning rate of 0.00001, the model uses the Adam optimiser and a weight decay of 0.0005 to avoid overfitting. Training time was used as the primary parameter to run tests across different datasets.

In addition to the recall, precision and mAP50 values, there are two other metrics that can be used to further assess model performance. The mAP50-95 calculates the average precision over several IoU thresholds (ranging from 0.5 to 0.95 in steps of 0.05). This metric is more challenging for the evaluation because it requires the model not only to have high precision at IoU 0.5, but also to perform well over a range of IoU thresholds. In other words, the mAP50-95 metric measures the ability of the model to accurately locate objects, even in cases where the prediction box needs to overlap more closely with ground truth. The F1 score is a balanced metric that considers both precision and recall. It is calculated as the harmonic mean of

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Figure 52: Performance metrics using YOLO models with various sizes on the COCO dataset for the detection task.(Jocher et al., 2023)

precision and recall, providing a single score that balances the trade-off between these two metrics. This is particularly useful to assess how well the model balances the need to avoid false positives (precision) and false negatives (recall). A high F1 score means that the model has both high precision and high recall, reflecting good overall detection performance. The F1 score can be calculated as follows

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Alongside the evaluation metrics, the Confidence value in a detection model such as YOLO is a key parameter for assessing the model's confidence in its predictions. It is mainly derived from the Objectness Score, which is the probability that the model determines whether a target exists in a given bounding box, and the Class Confidence, which predicts the probability of the specific class of the detected target in the bounding box. In YOLO, each grid cell predicts multiple candidate bounding boxes. For each candidate bounding box, the model outputs an Objectness Score indicating the probability that a target is present in that box, which can be written as $ObjectnessScore = P(Object)$. Then the class possibility of that object is calculated as $P(Class|Object)$. The final confidence score is acquired as the product of these two values: $Confidence = P(Object) \times P(Class|Object)$. In the experiment analysis, this confidence score serves as a threshold. By incrementally adjusting this threshold, the model filters out detections below a certain confidence level and recalculates the corresponding evaluation metrics (e.g. precision, recall) to assess the model's performance at different confidence levels.

5.2.2 Experiments results

The Figure 53 shows the Pipeline for dataset augmentation on Det-sf-v1 dataset and experimental evaluation of YOLOv8m model in wood knots detection. The Table 10 and Table 11 presents the general training evaluation metrics include the recall, precision, mAP50 value, mAP50-95 and different loss values from both training and

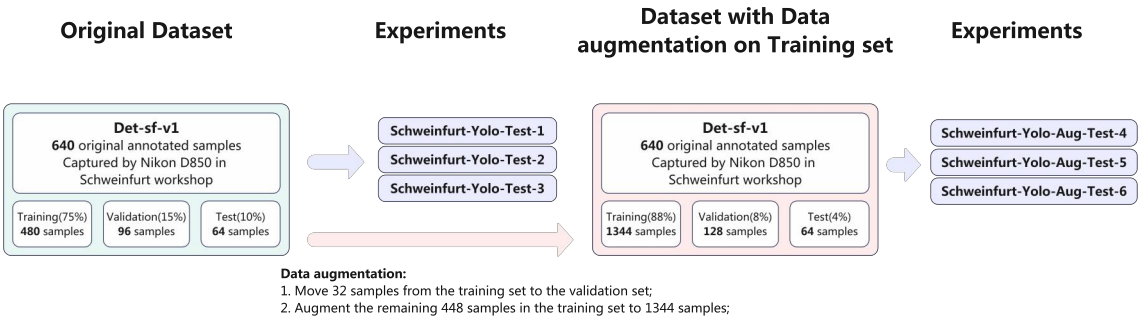


Figure 53: Pipeline for dataset augmentation on Det-sf-v1 dataset and experimental evaluation of YOLOv8m model in wood knots detection

validation stage during the whole process. Schweinfurt-Yolo-Test-1 to Schweinfurt-Yolo-Test-3 show the tests on original annotated images, while Schweinfurt-Yolo-Aug-Test-4 to Schweinfurt-Yolo-Aug-Test-6 show the results using the extended Det-sf-v1 dataset. The classification loss (cls loss) represents the model's ability to correctly classify detected features. The box loss measures how accurately the model predicts the position and size of the bounding boxes, by comparing the predicted boxes to the ground truth. Lastly, the DFL (distributional focal loss) evaluates the model's ability to distinguish between similar features among the detected objects.

From the training results, although both Schweinfurt-Yolo-Test-3 and Schweinfurt-Yolo-Aug-Test-6 were trained for the longest number of epochs, they were stopped at 174 and 155 epochs respectively due to the early stopping mechanism. This resulted in a decrease in box precision of 0.8298 for Schweinfurt-Yolo-Test-3, while recall increased by 0.8483. This suggests that while additional training improves recall to some extent, this may be at the expense of precision. This reflects the trade-off between precision and recall, indicating that longer training does not always improve precision and may create an imbalance between the two metrics. This finding can also be summarized from the Figure 54c.

Table 10: General metrics for testing with YOLOv8m on original Det-sf-v1 dataset and augmented Det-sf-v1 dataset

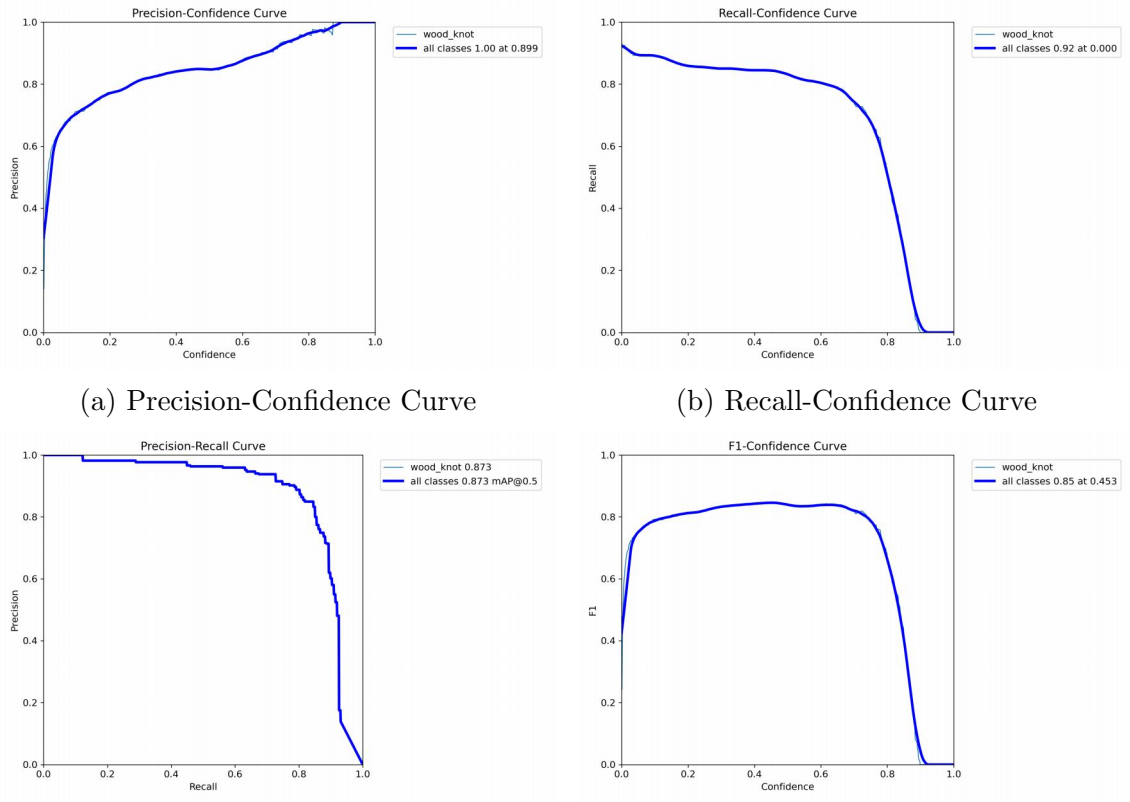
Tests	Planned Epoch(trained)	Training time	Box precision	Box recall	Box mAP50	Box mAP50-95
Schweinfurt-Yolo-Test-1	50(50)	0:09:33	0.8557	0.8181	0.8924	0.4997
Schweinfurt-Yolo-Test-2	100(100)	0:15:29	0.8818	0.8235	0.8864	0.4951
Schweinfurt-Yolo-Test-3	200(174)	0:23:59	0.8298	0.8483	0.8981	0.5015
Schweinfurt-Yolo-Aug-Test-4	50(50)	0:19:27	0.8448	0.8396	0.8762	0.4481
Schweinfurt-Yolo-Aug-Test-5	100(100)	0:37:08	0.8608	0.8396	0.8743	0.4611
Schweinfurt-Yolo-Aug-Test-6	200(155)	0:52:14	0.8457	0.8449	0.8727	0.4683

Despite the fact that Schweinfurt-Yolo-Test-3 had lower loss values on the training set compared to other tests without data augmentation, its validation loss was higher than Schweinfurt-Yolo-Test-2, which had fewer training epochs. This suggests that increasing training time does not always lead to improved model performance and

may even lead to overfitting, as indicated by the increasing gap between training and validation loss.

Table 11: Loss values for tests with YOLOv8m on original Det-sf-v1 dataset and augmented Det-sf-v1 dataset

Tests	Train cls loss	Train box loss	Train dfl loss	Val cls loss	Val box loss	Val dfl loss
Schweinfurt-Yolo-Test-1	0.5329	1.0125	1.2945	0.8222	1.6581	1.7956
Schweinfurt-Yolo-Test-2	0.4007	0.7899	1.1525	0.7820	1.7003	2.0299
Schweinfurt-Yolo-Test-3	0.3835	0.6460	1.0193	0.8062	1.7668	2.2130
Schweinfurt-Yolo-Aug-Test-4	0.3884	0.8790	1.1599	0.8133	1.8452	1.9719
Schweinfurt-Yolo-Aug-Test-5	0.2933	0.6826	1.0185	0.8139	1.8455	2.0276
Schweinfurt-Yolo-Aug-Test-6	0.3578	0.6710	1.0224	0.8950	1.8113	2.0475



(a) Precision-Confidence Curve

(b) Recall-Confidence Curve

(c) Precision-Recall Curve

(d) F1-Confidence Curve

Figure 54: Evaluation metrics for Schweinfurt-Yolo-Aug-Test-6 on augmented Det-sf-v1 dataset

Similarly, tests with data augmentation show a similar trend. The training loss values suggest that data augmentation improved generalisation performance on the training set. However, higher validation loss values indicate that there may be an imbalance in the validation set compared to the test set. As the validation samples were randomly selected from both the original and augmented data, this is likely to have resulted in a distributional discrepancy that affected the validation loss.

Another potential reason, as Hernández-García and König (2020) points out, could be that data augmentation acts as implicit regularisation by increasing data diversity, indirectly improving generalisation without reducing the expressive power of a model. However, it requires careful tuning, as improper use of data augmentation together with explicit regularisation techniques (e.g. weight decay, dropout) can degrade performance.

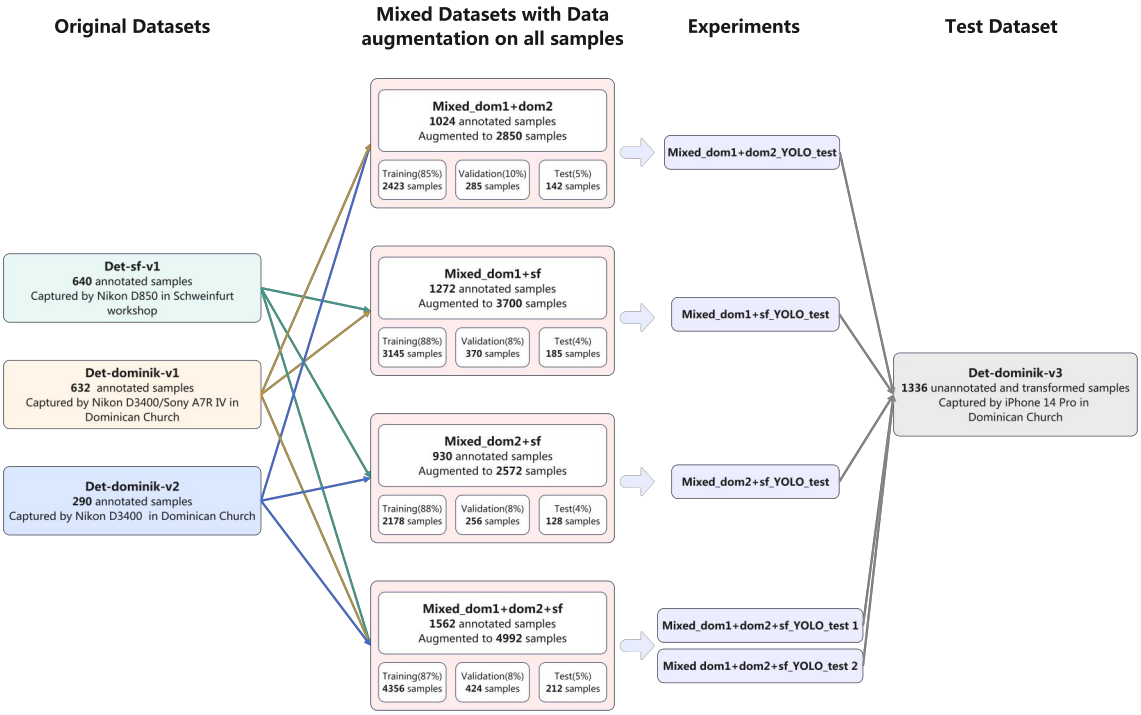


Figure 55: Pipeline for dataset augmentation on mixed datasets and experimental evaluation of YOLOv8m model in wood knots detection

In the next phase of testing, the following experiments focus on testing augmented mixed datasets derived from three primary datasets as shown in Figure 55, each with data augmentation: Det-sf-v1, Det-dominik-v1, and Det-dominik-v2. Each dataset is combined with the other two individually, with both the training and validation sets mixed accordingly. The first three experiments are conducted with 200 epochs. While previous tests on Det-sf-v1 indicated potential overfitting at 200 epochs, this might differ with the augmented mixed datasets due to the larger sample size. The model requires more time to learn and converge effectively when dealing with an increased variety of data. Therefore, two additional experiments with 200 epochs and 300 epochs are conducted using the combination of all three datasets to further compare model performance. The Table 12 and Table 13 present the overall training results for the various augmented mixed datasets, along with the corresponding loss values for both the training and validation sets.

When comparing the tests trained for 200 epochs, the box precision remains consistently high across all tests, ranging from 0.9799 to 0.9863, indicating the model's

Table 12: General metrics for testing with YOLOv8m on augmented mixed datasets

Tests	Dataset amount	Planned Epoch(trained)	Training time	Box precision	Box recall	Box mAP50	Box mAP50-95
Mixed_dom1+dom2_YOLO_test	3072	200(200)	1:50:01	0.9862	0.9706	0.9857	0.8656
Mixed_dom1+sf_YOLO_test	3738	200(200)	2:33:29	0.9819	0.9855	0.9886	0.8141
Mixed_dom2+sf_YOLO_test	3406	200(200)	1:43:59	0.9799	0.9570	0.9725	0.8441
Mixed_dom1+dom2+sf_YOLO_test_1	4608	200(200)	3:21:27	0.9863	0.9809	0.9915	0.8523
Mixed_dom1+dom2+sf_YOLO_test_2	4608	300(300)	4:37:08	0.9907	0.9729	0.9921	0.8780

Table 13: Loss values for tests with YOLOv8m on augmented mixed datasets

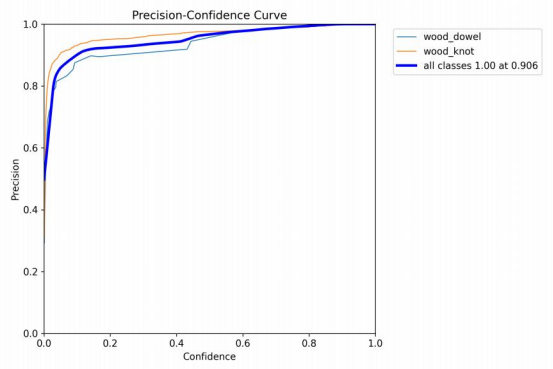
Tests	Train cls loss	Train box loss	Train dfl loss	Val cls loss	Val box loss	Val dfl loss
Mixed_dom1+dom2_YOLO_test	0.2802	0.4435	0.9194	0.2751	0.5316	0.9602
Mixed_dom1+sf_YOLO_test	0.3158	0.5530	0.9852	0.3554	0.7311	1.1618
Mixed_dom2+sf_YOLO_test	0.2604	0.4675	0.9402	0.3167	0.5990	1.0261
Mixed_dom1+dom2+sf_YOLO_test_1	0.2900	0.5202	0.9558	0.3003	0.6592	1.0706
Mixed_dom1+dom2+sf_YOLO_test_2	0.2428	0.4221	0.8954	0.2914	0.5211	0.9624

ability to effectively recognize objects and minimize false alarms. Notably, the Mixed_dom2+s_YOLO_test has the lowest box precision at 0.9799 and the lowest box recall at 0.9570, with the recall significantly lower than in other tests. This suggests potential overfitting during training, and the dataset may contain samples with highly similar features. Further evidence of this can be seen in the low overall training losses for Mixed_dom2+sf_YOLO_test, while the classification loss on the validation set is much higher than in the training results. This indicates that the mixed dataset from Det-dominik-v2 and Det-sf-v1 contributes less to improving the model's generalization performance in YOLOv8m.

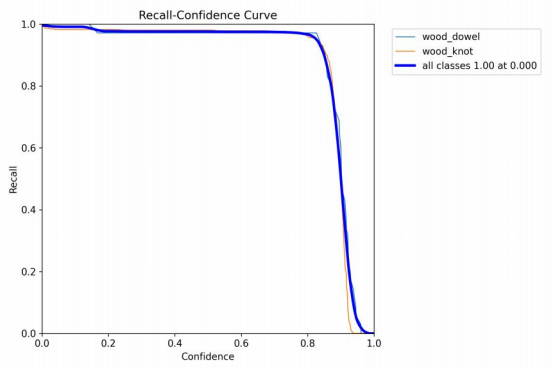
The Mixed_dom1+dom2+sf_YOLO_test_1, trained with all three basic datasets for 200 epochs, demonstrated a generally balanced performance. It achieved the highest box precision at 0.9863, the best mAP50 value at 0.9915, and a strong box recall of 0.9809. Although its mAP50-95 value was slightly lower compared to the Mixed_dom1+dom2_YOLO_test, the test results from Mixed_dom1+dom2+sf _YOLO_test_1 displayed a more consistent and balanced performance across all metrics at 200 epochs. The loss values suggest that the mixed dataset might have underfitted the model, indicating the need for additional training.

In contrast, the Mixed_dom1+dom2+sf_YOLO_test_2 with 300 epochs training time achieved superior results with a box precision of 0.9907, a box mAP50 of 0.9921, and an mAP50-95 of 0.8780. The slightly lower box recall suggests that the model may have sacrificed some detection capability to improve overall accuracy. The highest mAP50-95 value indicates that this model is more robust across different IoU thresholds. The training loss is significantly lower than the validation loss, particularly in box loss and dfl loss, which may indicate some overfitting. However, despite these losses, both recall and precision remain high, suggesting that the overfitting is not severe. Further increasing data diversity or adjusting regularization parameters could help reduce validation losses and improve model generalization. In the meanwhile, the Figure 73 shows that during the training process, the loss function gradually decreases with the increase of training rounds, indicating that the model gradually

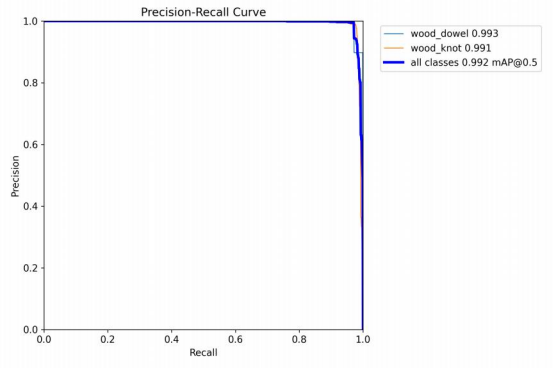
converges and the performance continues to improve. Although precision, recall and mAP50 stabilise after about 200 rounds, the continued increase in mAP50-95 and the continued decrease in losses on the validation set suggest that the model's ability to generalise continues to improve. This implies an improved performance of the model under tighter IoU thresholds as well as a better adaptation to the different samples in the validation set.



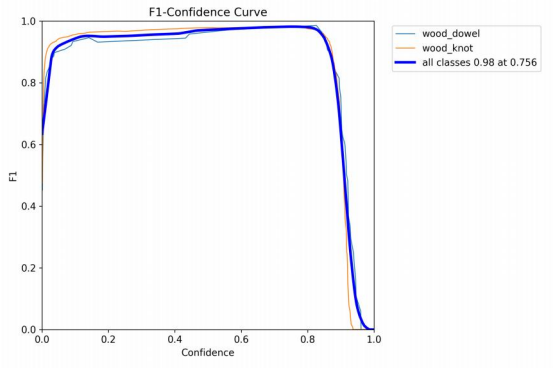
(a) Precision-Confidence Curve



(b) Recall-Confidence Curve



(c) Precision-Recall Curve



(d) F1-Confidence Curve

Figure 56: Evaluation curves for Mixed_dom1+dom2+sf_YOLO_test_2 on augmented Mixed_dom1+dom2+sf dataset

The figure 56 shows more concretely the differential changes through the variable evaluation metrics for the Mixed_dom1+dom2+sf_YOLO_test_2. The precision-confidence curve shows that as the confidence increases, so does the precision, which is close to 1.0 at a confidence of 0.85. This curve is generally steeper, indicating that the model has a lower probability of misclassification at medium to high confidence levels. Unlike the precision-confidence curve, the recall-confidence curve shows that the recall remains high at low confidence levels up to a confidence value of around 0.8. As the curve drops rapidly at higher confidence levels, this means that the model has a leakage problem in some cases, especially when the task requires a higher confidence level. The Precision-Recall curve is essentially one that rises quickly and flattens out at the top, suggesting that the model is maintaining better performance at different thresholds, which is already evidenced by the high

mAP50 and mAP50-95 values. The final F1 confidence curve shows that at around confidence 0.82, F1 peaks at around 0.98. This value means that at confidence 0.82 the model has the best balanced performance between precision and recall.

(a) (b) (c) (d) (e)

Figure 57: Images tested on Det-dominik-v3 dataset using model from Mixed_dom1+dom2_YOLO_test

(a) (b) (c) (d) (e)

Figure 58: Images tested on Det-dominik-v3 dataset using model from Mixed_dom1+sf_YOLO_test

(a) (b) (c) (d) (e)

Figure 59: Images tested on Det-dominik-v3 dataset using model from Mixed_dom2+sf_YOLO_test

To further evaluate the models' performance on unfamiliar samples, the models were tested on standardized images from the Det-dominik-v3 dataset after using Segmentation and Transformation processes. Figures 57 to 60 present the detection results of various test models on the same samples. Unexpectedly, the results on

(a) (b) (c) (d) (e)

Figure 60: Images tested on Det-dominik-v3 dataset using model from Mixed_dom1+dom2+sf_YOLO_test_2

Det-dominik-v3 showed that the model from Mixed_dom1+sf_YOLO_test performed the best, successfully detecting wood knots in 507 out of 1336 standardized images, whereas Mixed_dom1+dom2+sf_YOLO_test_2 detected only 373 samples with wood knots. This suggests that the model trained with Mixed_dom1+sf_YOLO_test has better generalization performance on Det-dominik-v3.

The differences between training and testing results could stem from several factors. First, the unbalanced mixture of the three base datasets may have introduced biases in the learned features, which means that some features critical to Det-dominik-v3 may have been underrepresented in training. Additionally, the higher complexity in the mixed dataset might have led the model to learn irrelevant or less useful features, negatively affecting its performance on the Det-dominik-v3 dataset. Lastly, localized overfitting could explain why the model performs well on some sub-datasets but poorly on others with different characteristics. This could also potentially lead to the bad performance on specific dataset like Det-dominik-v3.

5.2.3 Summary of detection tests

The evaluation of YOLOv8m model for wood knot detection across various datasets has provided several key insights, particularly regarding the challenges of generalization, dataset mixing, and overfitting. Theoretically, datasets containing mixed samples from various subsets should lead to more robust model performance and improved generalization to unfamiliar scenarios. The results were generally positive, with mixed data sets yielding high precision and recall. However, when the models were tested on real samples, it showed peculiar detection results across different samples. This required that the complexity and variety should be intensively considered for the future data collection process and dataset acquisition. The higher complexity in the augmented mixed datasets may have caused the model to learn irrelevant or less useful features, making it harder to generalise to new data sources such as Det-dominik-v3.

In conclusion, the use of augmented mixed datasets and longer training epochs did enhance precision and mAP50-95 for the detection task, as seen in tests such as Mixed_dom1+dom2+sf_YOLO_test_2. However, this improvement came at the

expense of generalization, as evidenced by the model's poorer performance on the Det-dominik-v3 dataset. Despite these challenges, augmented mixed datasets still have strong potential for achieving better generalization. Going forward, it will be important to ensure that the critical features are balanced and well-represented during dataset construction. Additionally, enhanced data augmentation strategies, along with regularization adjustments, could help reduce overfitting and further improve the model's generalization capabilities.

6 Analysis

This section provides a detailed analysis of specific subprocesses within the overall workflow, followed by potential improvement methods. The second part of the analysis focuses on challenges related to datasets, models, experiments, and further production. Additionally, various iterative approaches are discussed and tested throughout the whole section to explore continuous improvements.

6.1 Potential improvements of general workflow

6.1.1 Image preprocessing

As mentioned in Section 3, there are several issues with the captured raw data that make some images unsuitable for further processing. Common problems include unclear image quality, such as blurriness or poorly defined boundaries of the wooden timber surface. These issues can also arise during practical implementation of the workflow, making it essential to first examine the images against specific standards.

The initial examination should assess key factors such as image clarity, global and local contrast levels, lighting conditions, and the sharpness of the wood's edges. This pre-check ensures that the image meets the minimum quality required for further segmentation. Figure 61 provides examples of images captured with insufficient overall quality during the raw data acquisition process and an appropriate captured image which is ideal for further processing. Potential pre-processing discriminators could be calculating the Laplace operator of the image to ensure that the captured image has a high Laplace value, or using edge detection algorithms such as Canny edge detection to detect the sharpness of wooden edges in images. Methods such as Signal to Noise Ratio (SNR) and other metrics can also be helpful to quickly determine the level of noise in an image, which also helps to filter images captured under non-ideal conditions.

(a) Capture without sufficient contrast (b) Blurry captyre (c) Capture with poor lighting (d) Appropriate capture

Figure 61: Samples under insufficient collection conditions

6.1.2 Segmentation and transformation

Once image quality is ensured, the area captured within the image becomes crucial for achieving more accurate results. As demonstrated in the segmentation experiments, the captured area should include the two longest boundaries of the timber to accurately calculate the ratio between the timber's width and the detected bounding box. Figure 62 illustrates an improper capture of the timber surface. Ensuring proper capture of the entire timber surface, including its full boundaries, will be critical in future applications.

(a) Obscured timber boundary (b) Distinct timber boundary

Figure 62: Comparison between images with obscured and distinct boundaries of timber structure

The examples in the experiment section for segmentation focus on the vertical or horizontal timber, which means that the sloping timber as shown in Figure 63a could not be further processed at the moment, as the simplified polygon cannot be transformed appropriately perspectively.

Future improvements should focus on segmenting various timber structures more effectively and simplifying them to accommodate different wood orientations. This would enable detection of features such as wood knots and allow for a comprehensive evaluation of the entire timber structure. Several potential methods have already been investigated during workflow development. For example, extending the two longest edges of the segmented area to form a potential outer quadrilateral (Figure 63b) can help correct perspective distortions and standardize the image.

6.1.3 Detection

In the detection experiments, there are several general issues that need to be addressed. By reviewing the testing results on Det-dominik-v3, some objects are incorrectly detected as wood knots, as shown in Figure 64. Although future studies

(a) (b)

Figure 63: Sloping timber structure

will aim to limit the detection area to within the main timber structure, these incorrect detections can still significantly affect the robustness of the results. To reduce false positives and improve focus on wood knot features, a database with feature space will be extracted from the annotated images. This feature space will be used to distinguish the detected features more accurately with kNN method. The kNN method (Cover and Hart, 1967) classifies a sample by calculating the distance between the sample and its K nearest neighbours in the training set, then determining the sample's class based on a majority vote of these neighbours' classes.

Figure 64: False detection of wood knots within the dataset Det-dominik-v3

This method first extracted the annotated features from the mixed dataset from all three base datasets for recognition using ResNet18 (Figure 65). The original

ResNet18 contains 18 layers with four residual blocks (Figure 36). However, for feature extraction, the last fully connected layer is removed, so that the output with the shape $(512, 1, 1)$ from the last convolution layer with high dimensional features is stored. Once the database of extracted features is created, the high dimensional feature from the detected object (bounding box) is also extracted using the same modified ResNet18. The feature from the detected object will be discriminated using the database with kNN method to further confirm whether the detected object represents the feature of wood knots. Using this method, the bounding box in Figure 64 has been removed. The general results demonstrate that it significantly improves the filtering of positive samples and effectively reducing false detections.

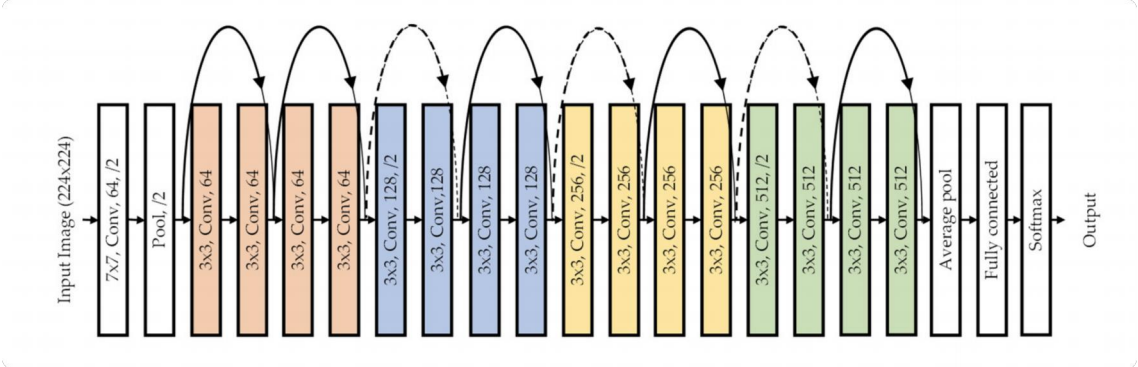


Figure 65: Structure of the Resnet-18 Model, Brown et al. (2022)

Apart from the problem of incorrect detections, the results of the detected bounding boxes also highlight two additional issues regarding the accuracy of the bounding boxes. The first issue is that a single wood knot may have overlapping bounding boxes, as shown in Figure 66a. The possible reason for this could be that the YOLO model makes multiple predictions at different scales, feature map hierarchies, or anchor boxes, resulting in multiple overlapping boxes. To address this problem, Non-Maximum Suppression (NMS) is applied to reduce the bounding boxes with lower confidence scores.

With NMS the bounding box with the highest confidence score in each image is selected as the reference box. Then, the IoU (Intersection over Union) is calculated between the reference box and the remaining bounding boxes, which is the ratio of the intersection area to the union area. Assuming the image has n detected bounding boxes, the IoU is calculated as follows:

$$\text{IoU}(Ref, B_{n-1}) = \frac{|Ref \cap B_{n-1}|}{|Ref \cup B_{n-1}|}$$

Based on the IoU, bounding boxes with an IoU greater than a certain threshold are removed, while those with an IoU less than the threshold are retained. Once no more bounding boxes are removed, the process repeats: the bounding box with the highest confidence among the remaining boxes is selected as the new reference box, and the same steps are followed until no more bounding boxes can be removed. The

(a) Detection with bounding box overlapping

(b) Detection after NMS refined

Figure 66: Bounding box reduction using NMS

figure 66b shows the results after the NMS method, which means that the NMS can effectively reduce the bounding box overlap.

The second issue observed in the detection results is that the size of the bounding boxes is larger than the actual wood knot dimensions. Even though the detection is correct, since the task involves estimating the ratio between the timber boundary and the bounding box, achieving more precise bounding boxes that closely fit the wood knot will lead to more accurate results. To address this, Otsu's Thresholding is applied to the detected bounding boxes, converting the feature into a binary feature space and segmenting the foreground from the background using an optimal threshold.

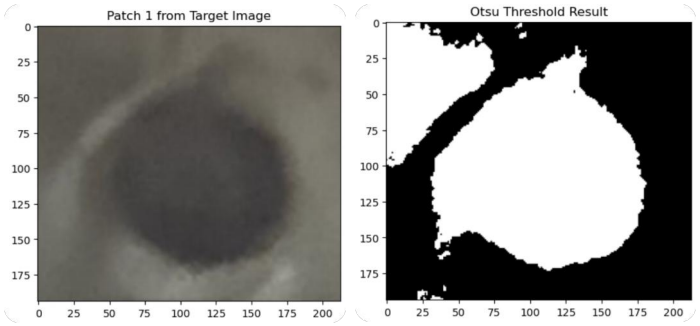
Otsu's Thresholding first calculates the histogram of grey values within the bounding box. For each threshold t, the grey-level histogram is divided into two classes: pixels with values less than t are classified as background, while pixels with values greater than t are classified as foreground. For each possible threshold t, the inter-class variance between the foreground and background is calculated using the following equation:

$$\sigma_b^2(t) = w_0(t) \cdot w_1(t) \cdot [\mu_0(t) - \mu_1(t)]^2$$

while $w_0(t)$ and $w_1(t)$ are the weights (proportions) of the foreground and background pixels at threshold t, and $\mu_0(t)$ and $\mu_1(t)$ are the average grey value of foreground and background pixels. are the average grey values of the foreground and background pixels, respectively. The algorithm iterates over all possible thresholds t to find the one that maximizes $\sigma_b^2(t)$ value.

With this optimal threshold t, the bounding box is binarized and effectively segmented into foreground and background, as illustrated in Figure 67. The resulting refined bounding boxes, shown in Figure 68, demonstrate improved precision in

comparison to the original detection results. Although some bounding boxes may still be larger than the wood knot, the overall performance has improved and the resulted bounding box is closer to the target wood knot dimension.



(a) Extracted annotated feature (b) Binary image using Otsu threshold

Figure 67: Binarization of annotated feature using Otsu threshold

(a) Original detection results (b) Refined bounding box sizes using Otsu threshold

Figure 68: Comparison of the reduction of the size of detected bounding boxes based on the Otsu threshold

6.1.4 Estimation

Although the primary focus of this master's thesis is to describe the methods and experiments using machine learning and deep learning techniques, the estimation of the ratio between the wooden timber boundary and the bounding box can also be

further improved. Currently, the ratio is easily computed using the perspectively corrected polygon and the bounding box of the detected wood knots, as shown in Figure 69. However, this approach lacks comparison and adjustment between image pixels and absolute values in real-world scenarios. There may be discrepancies between the computer vision-based method and the actual physical measurements.

Figure 69: Schematic explanation of the radio export based on the size and position of the detected bounding box and the perspectively corrected polygon of the timber boundary

To improve the accuracy of these estimations, calibration between the digital process and manual measurements should be considered. There could be several differential values to assess how well the system performs compared to manual testing. For example, the system could be designed to automatically adjust errors after several manual calibrations. Although this problem has not been addressed in this thesis, it is important to note for future optimization.

6.2 Challenges and further Optimisations

6.2.1 Datasets

The currently available datasets for detection are primarily captured from the Dominican Church. As shown by the experiments with various mixed datasets, the Mixed_dom1+sf_YOLO_test, which consists of samples from both the Dominican Church and the wood workshop in Schweinfurt, delivers more robust detection results when tested on the Det-dominik-v3 dataset. This suggests that datasets from multiple sources may help the model learn a richer and more diverse set of features, thereby improving the model's ability to generalize to different scenarios. In

contrast, using samples predominantly from a single source may cause the resulting model to become overly dependent on specific scenarios or data characteristics, making it more likely to underperform in unfamiliar environments.

Although data augmentation can simulate more varied samples, building datasets from multiple sources provides a more reliable approach to ensure diversity. While standardizing samples from different sources might pose challenges, the extraction of high-dimensional features can help make these samples comparable. The reduction in false detections through the use of kNN further supports this theory.

Additionally, annotation quality plays a critical role in detection performance. Incorrect or inappropriate annotations can lead to the exclusion of important features or the inclusion of unnecessary ones, both of which can mislead the training process. To address this, future experiments will involve expert review of the annotated datasets, particularly by specialists in wood studies, to ensure a more reliable training process.

6.2.2 Models

At the moment, only the standard models based on Detectron2 and YOLO have been tested. To further enhance model performance in both segmentation and detection tasks, task-specific mechanisms should be considered, which means that both processes need to be more attuned to the features of wood timber and wood knots. As illustrated in Figure refffig:analyse-model-edge, biological growth patterns could also aid in more precisely locating wood knots. The combination of deep learning models with feature-specific techniques has the potential to significantly improve detection results.

It is also important to note that other wood damage, such as wood cracks or mould on the surface of the wood, may also need to be identified as they are also parameters that affect the mechanical performance of the timber structure. A deeper analysis of the biological patterns on timber surfaces could make the detection process more explainable. If wood knots can be tracked based on their growth patterns, the model's detection capabilities will likely become more robust and reliable.

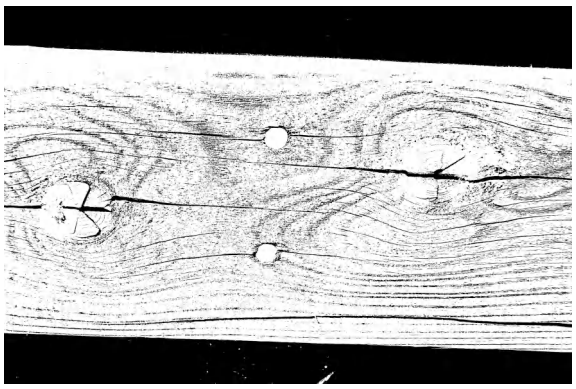


Figure 70: Clear wood grain through the application of a threshold filter

6.2.3 Experiments

In Section 5, the experimental hyperparameters primarily focus on training time, differences in feature extraction using ResNet50 or ResNet101, and the use of mixed datasets. Other important hyperparameters, such as learning rate, optimization algorithms, and regularization methods, have not been thoroughly explored. Particularly for flexible frameworks like Detectron2, there is potential for more fine-tuning of individual modules by adjusting diverse hyperparameters independently.

Validation methods such as cross-validation or k-fold validation could also be introduced to yield more reliable experimental results. Running multiple experiments on different data splits helps avoid model dependence on specific data divisions and produces more generalizable evaluation outcomes.

In general, these improvements in the experimental process aim to enhance the overall performance and stability of the model, making it more robust and effective in real-world applications. Additionally, by iterating on the experimental design, the model could achieve better accuracy and less overfitting, and improve its generalization capabilities across different tasks and datasets.

7 Conclusion

In conclusion, the primary objective of this master's thesis is to develop an automated detection system for assessing wood knots and their proximity to timber boundaries, thereby supporting the digital and automatic evaluation of the stability of historic wooden structures. Throughout the project, multiple datasets were created for detection and segmentation tasks, which were tested and evaluated using models such as Detectron2 and YOLOv8m-seg for segmentation, and YOLOv8m for primary detection.

The process of data collection and dataset creation posed several challenges, including image clarity, contrast, brightness, and the visibility of wood knots and timber structures. These factors are critical during dataset creation. Additionally, image normalization, labelling, and export are essential steps in ensuring a high quality dataset. These considerations will remain important as the datasets continue to evolve.

Despite using pre-trained standard models for testing, the comparison of training times, model architectures, and dataset usage yielded insightful results. The robustness of the models showed that their performance on unfamiliar data can differ significantly from the training process. This highlights the need for more precise evaluation metrics and further informs the optimization of models and testing strategies. The variation in model performance between training and real-world testing scenarios requires ongoing attention and fine-tuning. Furthermore, aspects like execution time, resource consumption, and cost-efficiency were not covered in this thesis but will be addressed in future experiments to ensure reliable implementation on low-cost devices.

As the entire workflow is designed to be implemented on low-cost devices such as smartphones, integrating data from built-in sensors such as accelerometers, gyroscopes and level sensors can help improve stability and accuracy throughout the whole process. These sensors can provide vital information about the orientation and movement of the device, enabling more accurate data capture and reducing errors caused by inconsistent positioning or movement during operation.

Although this thesis establishes a standard process for automated wood knot detection, as discussed in the previous section, the entire process, from dataset creation to model performance, needs to be iteratively validated in real-life scenarios. Collaboration with experts will be crucial in optimizing both the system and the datasets. The datasets will be continuously expanded and tested for quality, while additional mechanisms will be introduced to enhance the tracking of the unique characteristics of historic timber. These improvements will ensure that the system becomes more accurate and reliable over time.

A Declaration

Some of the research findings from this thesis have been published or presented in the following formats:

Published Article:

Article Title:

Towards automatic defects analyses for 3D structural monitoring of historic timber

Journal/Workshop:

The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-2/W4-2024 10th Intl. Workshop 3D-ARCH "3D Virtual Reconstruction and Visualization of Complex Architectures"

Publication Date:

21–23 February 2024, Siena, Italy

Publication DOI:

10.5194/isprs-archives-xlviii-2-w4-2024-103-2024

Article Title:

Machbarkeitsstudie zur automatisierten Zustandsanalyse verbauter historischer Holzbalken

Tagung:

44. Wissenschaftlich-Technische Jahrestagung der DGPF e.V.

Publication Date:

14 August 2024

Publication DOI:

10.24407/KXP:1885713169

Workshop Presentation:

Workshop Title:

KI und Denkmalpflege.Potenziale nutzen, Risiken erkennen

Organisation:

ICOMOS workshop

Presentation Date and Location:

8-9 October 2024, Berlin

B Code Availability

The code used for this study is made freely available under the MIT licence. It can be accessed at the following GitHub repository:
https://github.com/happy-panda-ops/xAI_Masterthesis_Pan.git

This repository contains all the code used for model training, along with some test results. Thanks to the tutorial of Bhattiprolu (2023) the training code of Detectron2 is based on it. Test results based on YOLOv8, as well as other related models, can be found in the corresponding subfolders. However, due to GitHub's file size limitations for model exports, tests conducted with Detectron2 have not been uploaded. If you require access to these models, please submit a request in the Issues section of the repository.

Please note that the datasets mentioned in this master thesis are not publicly available due to confidentiality. If access is needed, please contact the authors directly.

Additionally, the final production code for general test will also be provided. Detailed information regarding usage and setup will be included in the README file in GitHub.

C General workflow

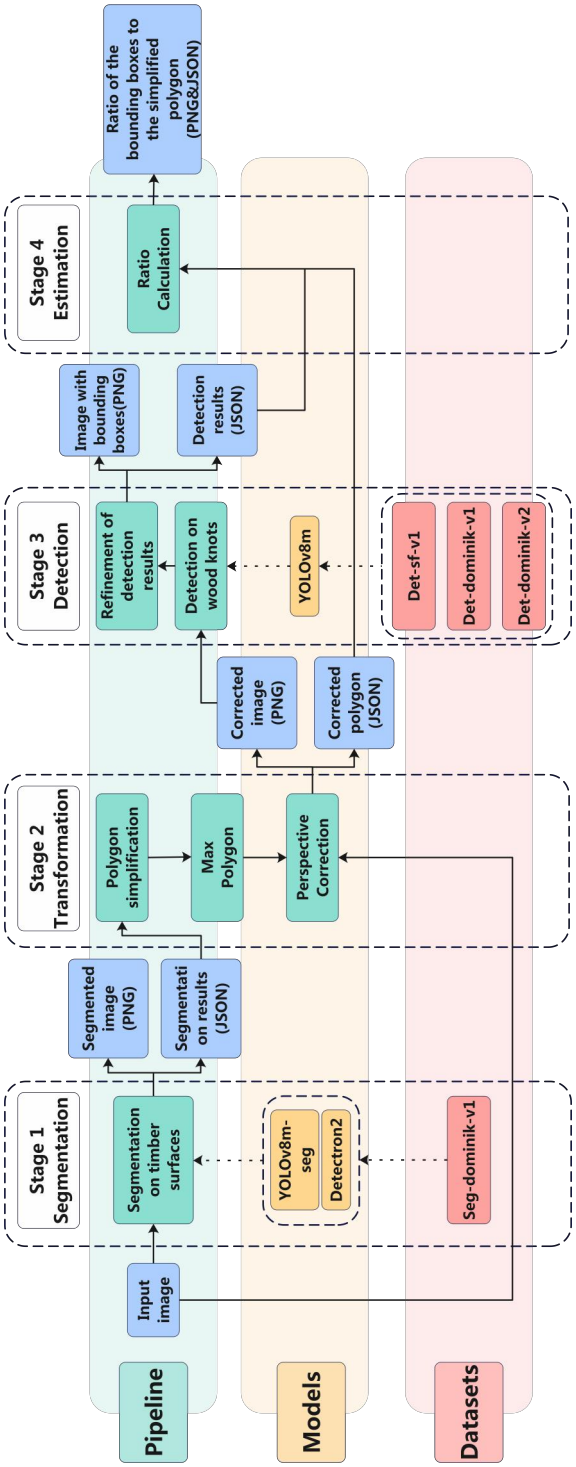


Figure 71: General workflow of AI-assisted detection on wood knots

D Figure

The following figures show the training and validation metrics during the training processes for the Schweinfurt-Yolo-Aug-Test-6 test and the Mixed_dom1+dom2+sf_YOLO_test_2 test on the wood knot detection task based on YOLOv8m.

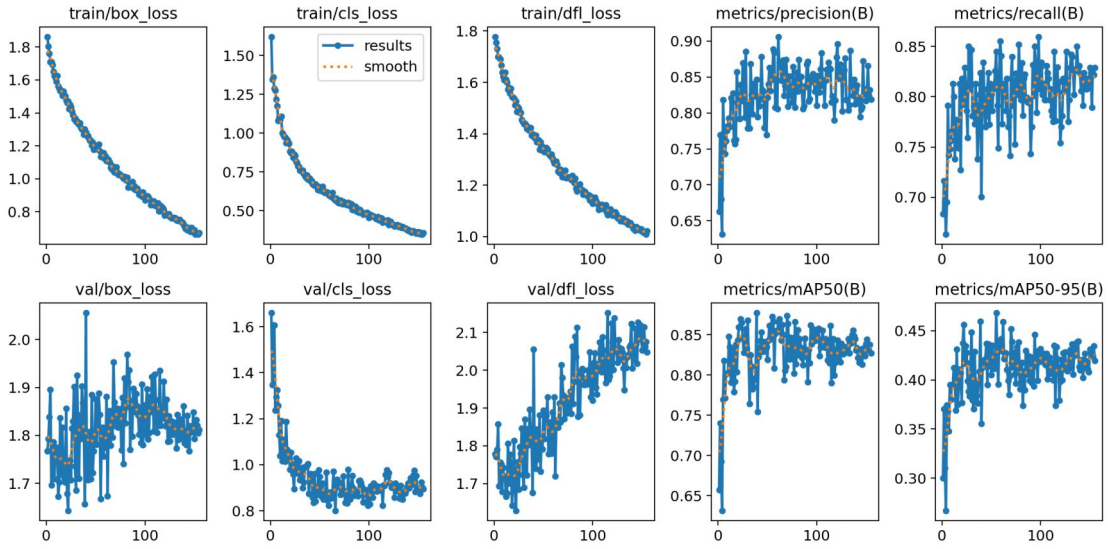


Figure 72: Training and Validation Metrics for Schweinfurt-Yolo-Aug-Test-6 on augmented Det-sf-v1 dataset

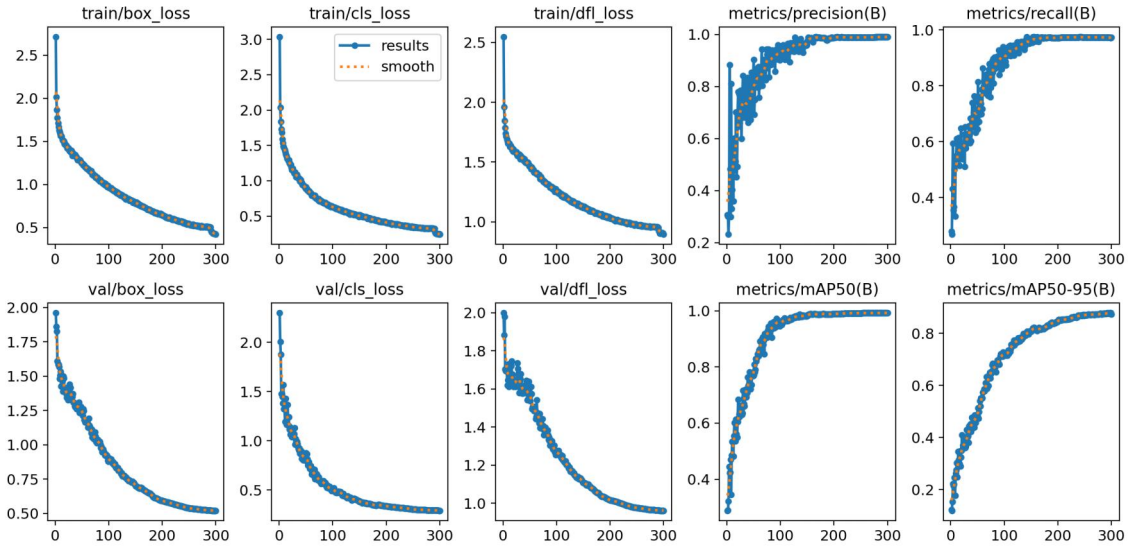


Figure 73: Training and Validation Metrics for Mixed_dom1+dom2+sf_YOLO test_2 on augmented Mixed_dom1+dom2+sf dataset

Bibliography

Marc Ackermann, Deniz Iren, Sebastian Wesselmecking, Deekshith Shetty, and Ulrich Krupp. Automated segmentation of martensite-austenite islands in bainitic steel. *Materials Characterization*, 191:112091, September 2022. ISSN 10445803. doi: 10.1016/j.matchar.2022.112091.

V. Baño, F. Arriaga, A. Soilán, and M. Guaita. Prediction of bending load capacity of timber beams using a finite element method simulation of knots and grain deviation. *Biosystems Engineering*, 109(4):241–249, August 2011. ISSN 15375110. doi: 10.1016/j.biosystemseng.2011.05.008.

S. Bhattiprolu. 330 - fine tuning detectron2 for instance segmentation using custom data, 2023. URL `https://youtu.be/cEgF0YknpZw`. Video, YouTube, August 23.

Jason Brown, Zahra Gharineiat, and Nawin Raj. CNN Based Image Classification of Malicious UAVs. *Applied Sciences*, 13(1):240, December 2022. ISSN 2076-3417. doi: 10.3390/app13010240.

Izabela Burawska, Marcin Zbie, Jacek Kalici, and Piotr Beer. Technical simulation of knots in structural wood. *Annals of Warsaw University of Life Sciences - SGGW, Forestry and Wood Technology*, 82:105–112, 2013.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00994018.

T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, January 1967. ISSN 0018-9448, 1557-9654. doi: 10.1109/TIT.1967.1053964.

Yuming Cui, Shuochen Lu, and Songyong Liu. Real-time detection of wood defects based on SPP-improved YOLO algorithm. *Multimedia Tools and Applications*, 82(14):21031–21044, June 2023. ISSN 1380-7501, 1573-7721. doi: 10.1007/s11042-023-14588-7.

Fenglong Ding, Zilong Zhuang, Ying Liu, Dong Jiang, Xiaoan Yan, and Zhengguang Wang. Detecting Defects on Solid Wood Panels Based on an Improved SSD Algorithm. *Sensors*, 20(18):5315, September 2020. ISSN 1424-8220. doi: 10.3390/s20185315.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021.

Frank Ebner. Festigkeitsuntersuchung an verbauten konstruktionshölzern. Master's thesis, Otto-Friedrich-Universität Bamberg and Hochschule für angewandte Wissenschaften Coburg, Bamberg, Germany, 2018. In German.

S. Fan, S. W. K. Wong, and J. V. Zidek. Knots and their effect on the tensile strength of lumber: A case study. *Journal of Quality Technology*, 55(4):510–522, 2023. doi: 10.1080/00224065.2023.2180457.

Yiming Fang, Xianxin Guo, Kun Chen, Zhu Zhou, and Qing Ye. Accurate and Automated Detection of Surface Knots on Sawn Timbers Using YOLO-V5 Model. 2021.

Gerhard Fink and Jochen Kohler. Model for the prediction of the tensile strength and tensile stiffness of knot clusters within structural timber. *European Journal of Wood and Wood Products*, 72(3):331–341, May 2014. ISSN 0018-3768, 1436-736X. doi: 10.1007/s00107-014-0781-0.

Mingyu Gao, Dawei Qi, Hongbo Mu, and Jianfeng Chen. A Transfer Residual Neural Network Based on ResNet-34 for Detection of Wood Knot Defects. *Forests*, 12 (2):212, February 2021. ISSN 1999-4907. doi: 10.3390/f12020212.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, January 2016. ISSN 1939-3539. doi: 10.1109/TPAMI.2015.2437384.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Irene Yu-Hua Gu, Henrik Andersson, and Raul Vicen. Wood defect classification based on image analysis and support vector machines. *Wood Science and Technology*, 44(4):693–704, November 2010. ISSN 0043-7719, 1432-5225. doi: 10.1007/s00226-009-0287-9.

Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3 (6):610–621, November 1973. ISSN 2168-2909. doi: 10.1109/TSMC.1973.4309314.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, September 2015. ISSN 1939-3539. doi: 10.1109/TPAMI.2015.2389824.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, October 2017. doi: 10.1109/ICCV.2017.322.

Ting He, Ying Liu, Chengyi Xu, Xiaolin Zhou, Zhongkang Hu, and Jianan Fan. A Fully Convolutional Neural Network for Wood Defect Location and Identification. *IEEE Access*, 7:123453–123462, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2937461.

Alex Hernández-García and Peter König. Data augmentation instead of explicit regularization, November 2020.

Min Hu, Andreas Briggert, Anders Olsson, Marie Johansson, Jan Oscarsson, and Harald Säll. Growth layer and fibre orientation around knots in Norway spruce: A laboratory investigation. *Wood Science and Technology*, 52(1):7–27, January 2018. ISSN 0043-7719, 1432-5225. doi: 10.1007/s00226-017-0952-3.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, Honolulu, HI, July 2017. IEEE. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.243.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. URL `https://github.com/ultralytics/ultralytics`.

J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, November 1995. doi: 10.1109/ICNN.1995.488968.

Range King. Brief summary of yolov8 model structure 189. `https://github.com/RangeKing`, 2023. Accessed: 2024-10-13.

P. Kodytek, A. Bodzas, and P. Bilik. A large-scale image dataset of wood surface defects for automated vision-based quality control processes. *F1000Research*, 10:581, 2021. doi: 10.12688/f1000research.52903.2.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Chao Li, Aojun Zhou, and Anbang Yao. Omni-Dimensional Dynamic Convolution, September 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8693, pages 740–755. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10601-4 978-3-319-10602-1. doi: 10.1007/978-3-319-10602-1_48.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection, February 2018.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46448-0.

Y. Liu, M. Hou, A. Li, Y. Dong, L. Xie, and Y. Ji. AUTOMATIC DETECTION OF TIMBER-CRACKS IN WOODEN ARCHITECTURAL HERITAGE USING YOLOv3 ALGORITHM. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2020:1471–1476, August 2020. ISSN 1682-1750. doi: 10.5194/isprs-archives-XLIII-B2-2020-1471-2020.

Markus Lukacevic, Georg Kandler, Min Hu, Anders Olsson, and Josef Füssl. A 3D model for knots and related fiber deviations in sawn timber for prediction of mechanical properties of boards. *Materials & Design*, 166:107617, March 2019. ISSN 02641275. doi: 10.1016/j.matdes.2019.107617.

Fakhira Iwani Muhammad Redzuan and Marina Yusoff. Knots timber detection and classification with C-Support Vector Machine. *Bulletin of Electrical Engineering and Informatics*, 8(1):246–252, March 2019. ISSN 2302-9285, 2089-3191. doi: 10.11591/eei.v8i1.1444.

T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 582–585 vol.1, October 1994. doi: 10.1109/ICPR.1994.576366.

R. Qayyum, K. Kamal, T. Zafar, and S. Mathavan. Wood defects classification using GLCM based features and PSO trained neural network. In *2016 22nd International Conference on Automation and Computing (ICAC)*, pages 273–277, Colchester, United Kingdom, September 2016. IEEE. ISBN 978-1-86218-132-8. doi: 10.1109/IConAC.2016.7604931.

Michael H. Ramage, Henry Burridge, Marta Busse-Wicher, George Fereday, Thomas Reynolds, Darshil U. Shah, Guanglu Wu, Li Yu, Patrick Fleming, Danielle Densley-Tingley, Julian Allwood, Paul Dupree, P.F. Linden, and Oren Scherman. The wood from the trees: The use of timber in construction. *Renewable and Sustainable Energy Reviews*, 68:333–359, February 2017. ISSN 13640321. doi: 10.1016/j.rser.2016.09.107.

Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement, April 2018.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.91.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf`.

McCulloch S., Warren and Pitts Walter. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943. ISSN 0092-8240. doi: 10.1007/BF02478259.

Khaled Saad and András Lengyel. A parametric investigation of the influence of knots on the flexural behaviour of wood beams, August 2022.

J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision*, 104(2):154–171, September 2013. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-013-0620-5.

Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. YOLOv10: Real-Time End-to-End Object Detection, May 2024a.

Chien-Yao Wang, I.-Hau Yeh, and Hong-Yuan Mark Liao. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information, February 2024b.

Mingtao Wang, Mingxi Li, Wenyan Cui, Xiaoyang Xiang, and Huaqiong Duo. TSW-YOLO-v8n: Optimization of detection algorithms for surface defects on sawn timber. *BioResources*, 18(4):8444–8457, October 2023a. ISSN 19302126, 19302126. doi: 10.15376/biores.18.4.8444-8457.

Rijun Wang, Fulong Liang, Bo Wang, and Xiangwei Mou. ODCA-YOLO: An Omni-Dynamic Convolution Coordinate Attention-Based YOLO for Wood Defect Detection. *Forests*, 14(9):1885, September 2023b. ISSN 1999-4907. doi: 10.3390/f14091885.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

Honglei Xi, Rijun Wang, Fulong Liang, Yesheng Chen, Guanghao Zhang, and Bo Wang. SiM-YOLO: A Wood Surface Defect Detection Method Based on the Improved YOLOv8. *Coatings*, 14(8):1001, August 2024. ISSN 2079-6412. doi: 10.3390/coatings14081001.

Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, Honolulu, HI, July 2017. IEEE. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.634.

Xie Zhang, Huibo Sun, Gangbiao Xu, Yanjun Duan, Jan Jan, Joris Joris, and Jiangtao Shi. Understanding the Effect of Knots on Mechanical Properties of Chinese Fir under Bending Test by Using X-ray Computed Tomography and Digital Image Correlation. *Forests*, 15(1):174, January 2024. ISSN 1999-4907. doi: 10.3390/f15010174.

Declaration of Authorship

Ich erkläre hiermit gemäß §9 Abs. 12 APO, dass ich die vorstehende Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Des Weiteren erkläre ich, dass die digitale Fassung der gedruckten Ausfertigung der Abschlussarbeit ausnahmslos in Inhalt und Wortlaut entspricht und zur Kenntnis genommen wurde, dass diese digitale Fassung einer durch Software unterstützten, anonymisierten Prüfung auf Plagiate unterzogen werden kann.

Place, Date	Signature