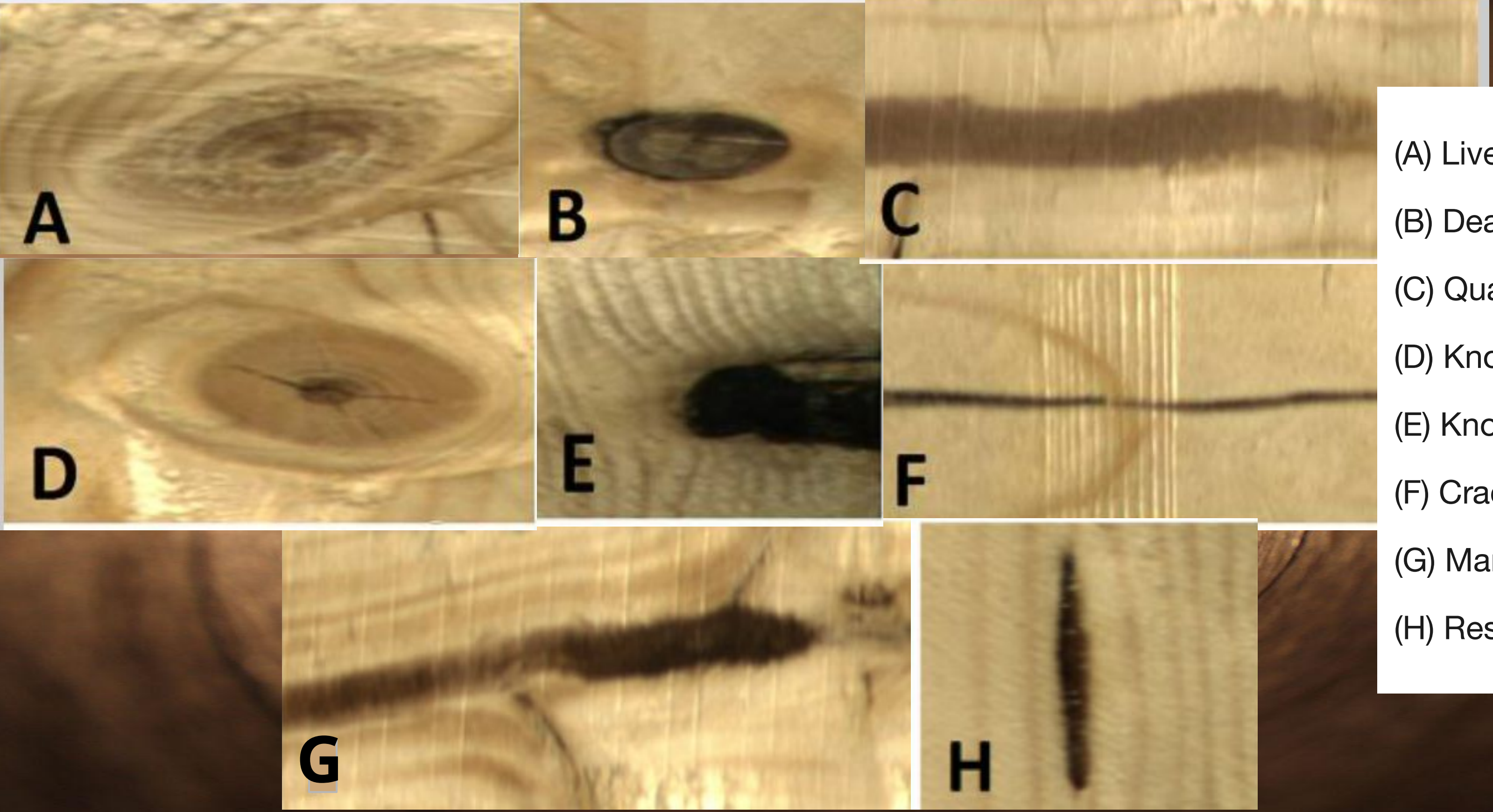
Segmentation and Detection of Wooden Surface Defects

By:
Harshita Jangir
Gaury Jithesh
Ayushi Shukla

What is the importance of defect detection in wood manufacturing?

Our Project

The dataset we will use comprises 4,000 images of wooden surfaces, annotated for eight different defect types: Quartzity, Live Knot, Marrow, Resin, Dead Knot, Knot with Crack, Knot Missing, and Crack.



(A) Live Knot

(B) Dead Knot

(C) Quartzity

(D) Knot with crack

(E) Knot missing

(F) Crack

(G) Marrow

(H) Resin

8 types of Defects

Workflow

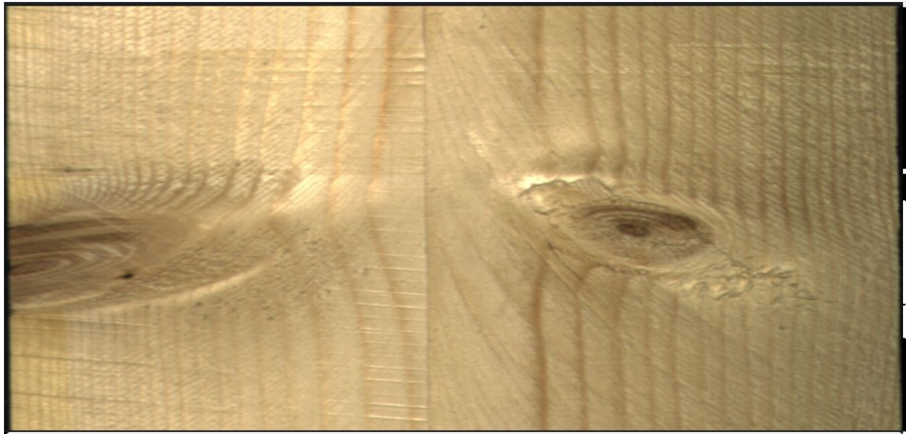
CV + ML

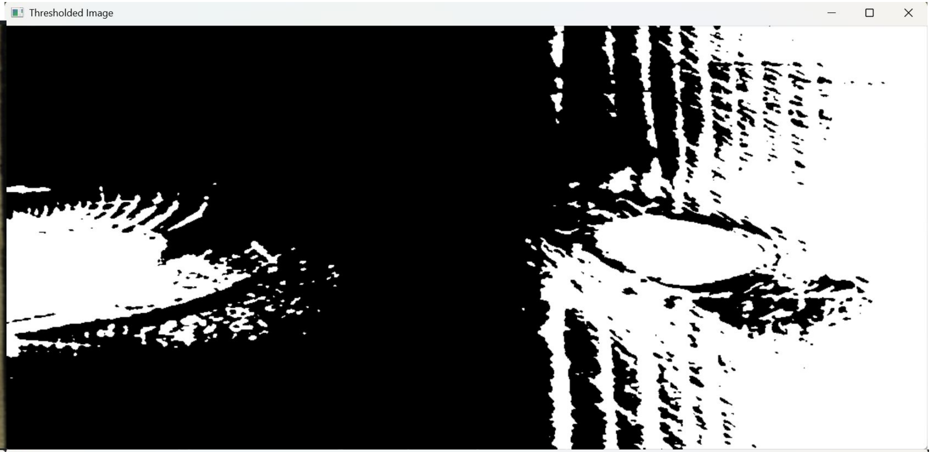
1. Image Loading and Preprocessing
2. Defect Segmentation
3. Bounding Box Extraction
4. Defect Feature Vector Preparation
5. Defect Classification
6. Visualization

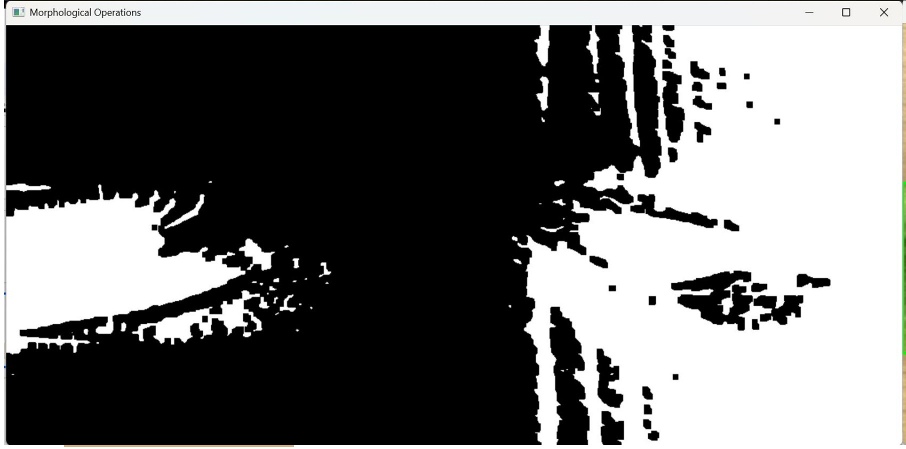
Version 0.0

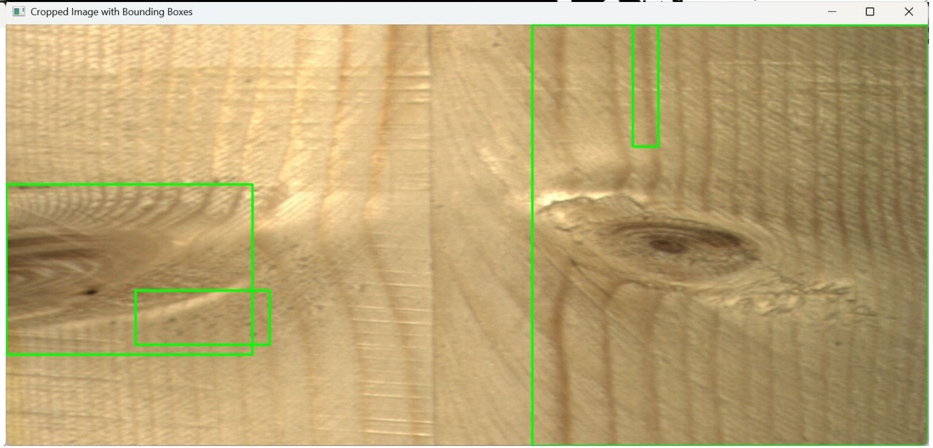
Computer Vision

1. **Image Loading and Preprocessing**
    - Remove unnecessary black regions from the left and right edges.
    - Convert the cropped image to grayscale using cv2.cvtColor.
2. **Gaussian Blur**
    - Apply cv2.GaussianBlur to reduce noise and smooth the image.
3. **Global Thresholding**
    - Apply cv2.threshold to create a binary image for further processing.
4. **Morphological Operations**
    - Use cv2.morphologyEx with a kernel to close small gaps and enhance defect shapes.
5. **Contour Detection**
    - Detect contours using cv2.findContours.
    - Filter contours based on size to retain only significant defects.




Thresholded Image

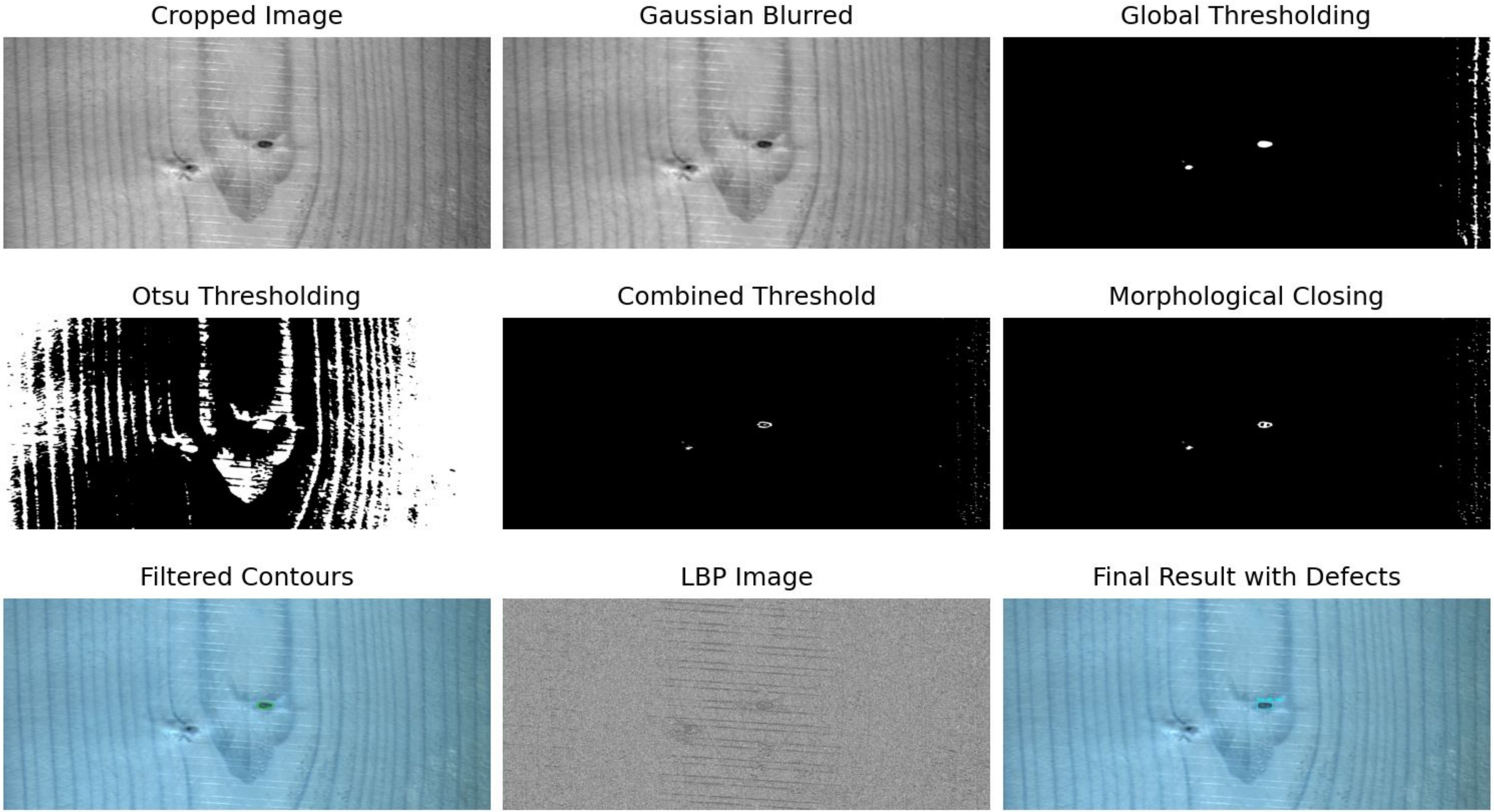

Morphological Operations


Cropped Image with Bounding Boxes

Version 1.0

Computer Vision

1. **Image Preprocessing**
   - Convert image to grayscale
   - Apply Gaussian blur for noise reduction
2. **Thresholding Techniques**
   - Global thresholding
   - Otsu's thresholding
   - Adaptive thresholding
3. **Combination and Refinement**
   - Combine thresholding results using bitwise operations
   - Perform morphological operations (e.g., closing) (using an elliptical kernel) to connect broken parts and fill holes within the detected defects.
4. **Contour Detection**
   - Detect and filter contours based on size to remove noise
5. **Feature Extraction**
   - Apply Local Binary Patterns (LBP) for texture analysis
6. **Defect Classification**
   - Use LBP histograms to classify types of wood defects

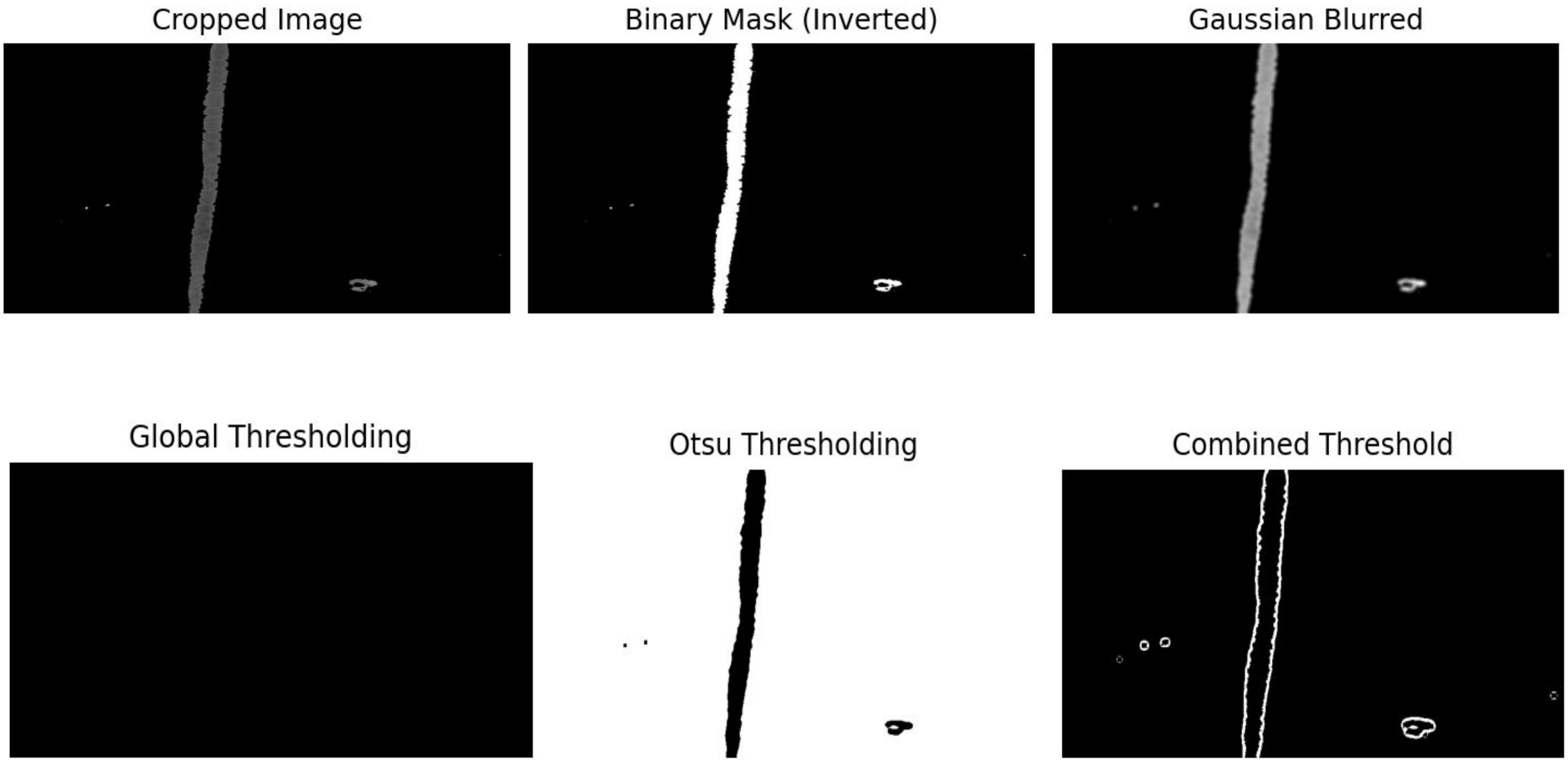


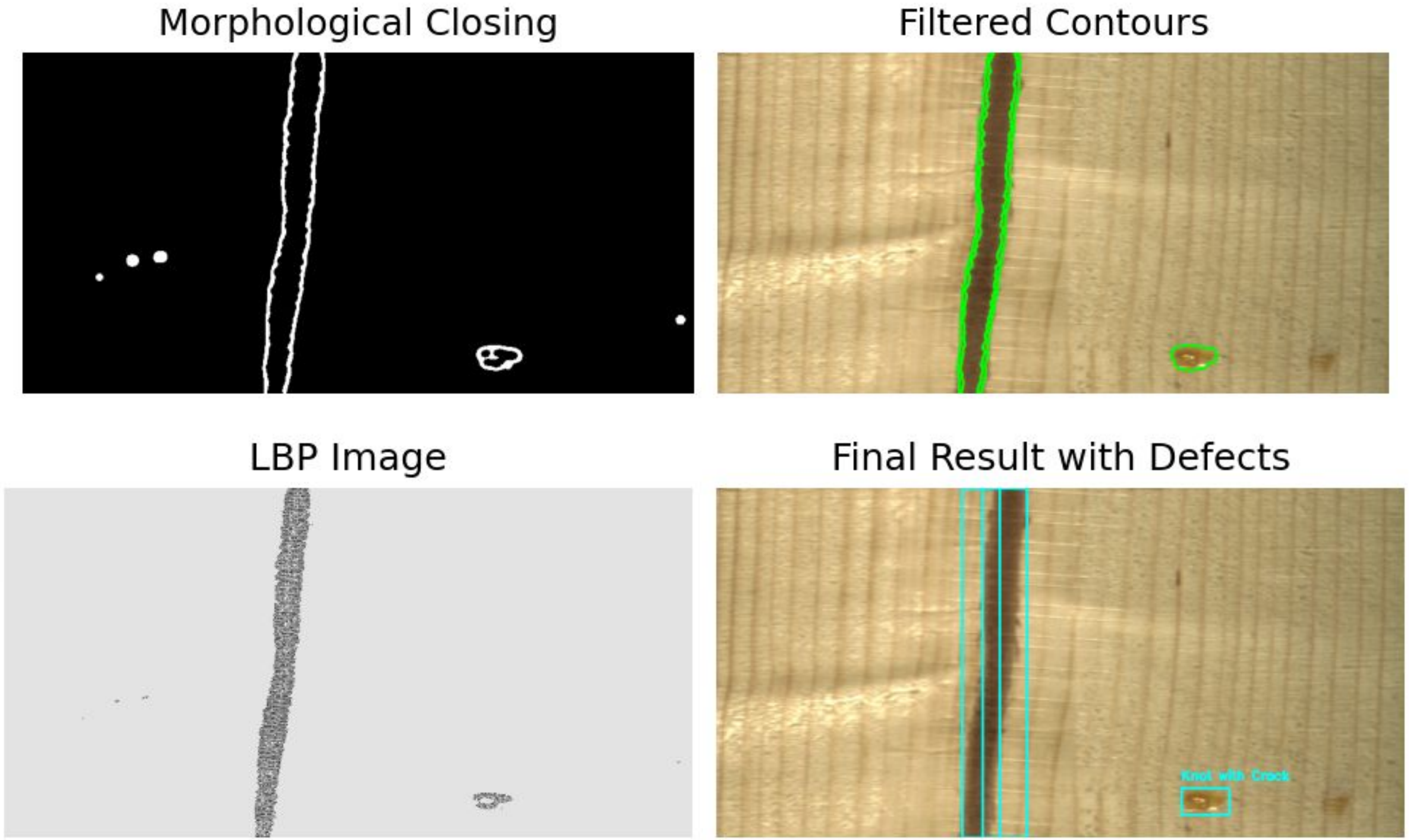
Cropped Image
Gaussian Blurred
Global Thresholding
Otsu Thresholding
Combined Threshold
Morphological Closing
Filtered Contours
LBP Image
Final Result with Defects

Final Steps Version 2.0

Computer Vision

1. **Image Preprocessing**
   - Crop 8% from each side to eliminate edge artifacts.
2. **HSV Masking for Color Filtering**
   - Convert the RGB image to HSV.
   - Calculate the median HSV color value of the image.
   - Define lower and upper bounds based on median HSV value.
   - Apply a mask to remove areas similar to the background color.
   - Use bitwise operations to retain defect regions in grayscale.
3. **Noise Reduction with Gaussian Blur**
4. **Thresholding Techniques**
   - Global Thresholding
   - Otsu's Thresholding
   - Adaptive Thresholding
5. **Combination and Refinement**
   - Combine thresholding results.
   - Perform **morphological closing**
6. **Contour Detection**
   - Detect and filter contours based on size to remove noise
7. **Bounding Box Calculation**
   - For each significant contour, calculate a bounding box.
8. **Feature Extraction using Local Binary Patterns (LBP) (Optional)**


Cropped Image
Binary Mask (Inverted)
Gaussian Blurred
Global Thresholding
Otsu Thresholding
Combined Threshold


Morphological Closing
Filtered Contours
LBP Image
Final Result with Defects
Knot with Crack

Steps

Machine Learning

- **Data Loading and Preprocessing**
  - Load bounding box data (x, y, width, height) and defect class labels from CSV and drop unnecessary columns like filename.
- **Feature and Target Separation**
  - Define X as bounding box feature columns: x_center, y_center, width, height.
  - Define y as the defect class (target) column.
- **Train-Test Split**
  - Split data into training and testing sets (80-20 split).
- **Model Training**
  - Train models using GridSearchCV for hyperparameter tuning on:
    - Random Forest
    - SVM
    - KNN
    - Decision Tree
- **Prediction and Evaluation**
  - Predict defect classes on the test set for all models.
  - Evaluate each model's performance using a classification report (precision, recall, F1-score for each defect class).
- **Defect Classification in Images**
  - For each bounding box detected in an image, prepare a feature vector (bounding box coordinates).
  - Use all models to classify defect type.
  - Map the predicted class to its corresponding defect label for display.

Random Forest

64%

Random Forest Classification Report:

	precision	recall	f1-score	support
Quartzity	0.81	0.40	0.54	42
Live_Knot	0.64	0.84	0.73	819
Marrow	0.42	0.54	0.47	41
Resin	0.71	0.66	0.68	125
Dead_Knot	0.67	0.49	0.57	584
Knot_with_Crack	0.37	0.14	0.21	98
Knot_Missing	0.60	0.25	0.35	24
Crack	0.66	0.57	0.61	110
accuracy			0.64	1843
macro avg	0.61	0.49	0.52	1843
weighted avg	0.64	0.64	0.62	1843

Random Forest Accuracy: 0.64

SVM

61%

```
SVM Classification Report:
                 precision    recall  f1-score   support

       Quartzity       0.91      0.24      0.38        42
       Live_Knot       0.62      0.87      0.72       819
          Marrow       0.22      0.05      0.08        41
           Resin       0.67      0.61      0.64       125
       Dead_Knot       0.60      0.42      0.50       584
 Knot_with_Crack       0.00      0.00      0.00        98
    Knot_Missing       0.00      0.00      0.00        24
           Crack       0.53      0.71      0.61       110

        accuracy                           0.61      1843
       macro avg       0.44      0.36      0.37      1843
    weighted avg       0.57      0.61      0.57      1843

SVM Accuracy: 0.61
```

KNN

59%

KNN Classification Report:

	precision	recall	f1-score	support
Quartzity	0.56	0.33	0.42	42
Live_Knot	0.63	0.76	0.69	819
Marrow	0.33	0.27	0.30	41
Resin	0.68	0.43	0.53	125
Dead_Knot	0.56	0.51	0.53	584
Knot_with_Crack	0.29	0.15	0.20	98
Knot_Missing	0.14	0.04	0.06	24
Crack	0.54	0.56	0.55	110
accuracy			0.59	1843
macro avg	0.47	0.38	0.41	1843
weighted avg	0.57	0.59	0.57	1843

KNN Accuracy: 0.59

Decision Tree

62%

Decision Tree Classification Report:

	precision	recall	f1-score	support
Quartzity	0.68	0.40	0.51	42
Live_Knot	0.63	0.84	0.72	819
Marrow	0.37	0.51	0.43	41
Resin	0.64	0.58	0.61	125
Dead_Knot	0.64	0.44	0.52	584
Knot_with_Crack	0.32	0.11	0.17	98
Knot_Missing	0.40	0.25	0.31	24
Crack	0.67	0.60	0.63	110
accuracy			0.62	1843
macro avg	0.54	0.47	0.49	1843
weighted avg	0.61	0.62	0.60	1843

Decision Tree Accuracy: 0.62

Some Outputs


Decision Tree: Dead_Knot
KNN: Dead_Knot
SVM: Dead_Knot
Random Forest: Dead_Knot
Decision Tree: Live_Knot
KNN: Live_Knot
SVM: Live_Knot
Random Forest: Live_Knot
Decision Tree: Dead_Knot
KNN: Live_Knot
SVM: Live_Knot
Random Forest: Dead_Knot
Decision Tree: Resin
KNN: Crack
SVM: Crack
Random Forest: Marrow
Decision Tree: Marrow
KNN: Marrow
SVM: Crack
Random Forest: Crack
Decision Tree: Resin
KNN: Resin
SVM: Resin
Random Forest: Resin
Decision Tree: Crack
KNN: Resin
SVM: Resin
Random Forest: Resin
Decision Tree: Dead_Knot
KNN: Dead_Knot
SVM: Live_Knot
Random Forest: Dead_Knot
Decision Tree: Resin
KNN: Quartzity
SVM: Resin
Random Forest: Resin

Some Outputs with Errors

Decision Tree: Dead_Knot
KNN: Dead_Knot
SVM: Dead_Knot
Random Forest: Resin
Decision Tree: Resin
KNN: Dead_Knot
SVM: Dead_Knot
Random Forest: Dead_Knot
Decision Tree: Live_Knot
KNN: Crack
SVM: Marrow
Random Forest: Marrow
Decision Tree: Dead_Knot
KNN: Live_Knot
SVM: Live_Knot
Decision Tree: Live_Knot
Random Forest: Live_Knot
Decision Tree: Live_Knot
Random Forest: Live_Knot
Decision Tree: Dead_Knot
KNN: Live_Knot
SVM: Live_Knot
Random Forest: Dead_Knot
Decision Tree: Live_Knot

Comparison with Deep Learning Models

Challenges and Limitations

Challenges

- **Feature Consistency**
  - LBP addition changes feature set; requires retraining for consistency.
- **Lighting & Image Quality**
  - Segmentation affected by varying lighting conditions; needs robust pre-processing.
- **Thresholding & Segmentation**
- **Contour Filtering**
  - Fixed contour size thresholds may miss defects or capture noise.
- **LBP Texture Extraction**
  - Limited performance on low-contrast textures; may miss subtle details.

Limitations

- **Class Imbalance**
  - Rare defect types lead to skewed classification results.
- **Computational Load**
  - High resource demands; challenging for real-time or large-scale data.
- **Bounding Box Accuracy**
  - Inaccurate bounding boxes impact defect classification reliability.
- **Generalization**
  - Limited adaptability to new wood types and defect patterns.

Insights, Future Work and Improvements

Conclusion

This project developed a method to detect and classify wood surface defects by combining computer vision techniques with machine learning models. Key steps included using HSV color masking, thresholding, and morphological operations to segment defects, while Local Binary Patterns (LBP) helped extract meaningful features. Both Random Forest and SVM classifiers performed well in distinguishing defect types, demonstrating the effectiveness of classical methods in industrial applications. This solution can streamline quality control in woodworking, offering an efficient and interpretable alternative to manual inspection.