Advanced Image Segmentation and Classification Techniques for Enhanced Voids Detection in Composite Oriented Strand Boards

Yi Sun, *Student Member, IEEE*

***Abstract*—Composite Oriented Strand Board (COSB) is an aerospace-grade material made from laminated, unidirectional carbon fiber-epoxy prepreg strands. This manufacturing method allows precise control over the material's thickness, flatness, and microstructure, ensuring high-quality production. Accurately assessing and quantifying void content in COSB using X-ray computed tomography (XCT) scan data is essential for understanding its microstructure and ensuring quality assurance in post-analysis. Traditional methods, such as manual inspection and basic image processing, are often time-consuming and imprecise. Previous studies leveraging deep learning algorithms, including Fully Convolutional Networks (FCN), U-net, and SegNet, have shown potential for autonomous void detection. However, these methods struggle with significant class imbalances, leading to low accuracy and precision in void classification.**

This research presents a novel approach that combines image enhancements, k-means clustering, convolutional kernels, and local XGBoost classifiers within a semi-supervised learning framework. Initially, image enhancement techniques are applied to improve the quality of CT-scanned images of COSBs. Convolutional kernels are then employed to extract meaningful features from the images. Next, K-means clustering is used to segment the enhanced images with extracted features, reducing the impact of class imbalance and complexity. Finally, local XGBoost classifiers within each cluster are trained on these features to obtain classification results.

Our study demonstrates that this integrated approach significantly enhances the detection and classification of voids in COSBs compared to previous deep learning methods. Specifically, the unsupervised k-means clustering helped local classifiers to better recognize voids within their clusters. The results show significant improvement in accuracy, Intersection over Union (IoU), and Boundary F1 (BF) scores. These advancements contribute to more reliable material quality assessment and characterization in COSB manufacturing, potentially leading to better quality control and reduced defects.

***Index Terms*—Composite Oriented Strand Boards (COSBs), image enhancement, image segmentation, semi-supervised learning, k-means clustering, convolutional kernels, XGBoost.**

I. Introduction

DR. CARINA LI [Introducing COSB]

Voids in Composite Oriented Strand Boards (COSBs) are typically unwanted defects originating from the manufacturing process. These voids can negatively impact various mechanical properties, including interlaminar shear stress, but are often hidden and undetected. To restrict void formation in the manufacturing process, it is essential to accurately detect these voids to gain a deeper understanding of their measurements, distribution, and evolution. By investigating porosity evolution, we can optimize cure parameters (pressure, temperature, duration, etc.) and protocols to improve the design process, potentially limiting void formation. However, most void detection and measurement currently require extensive human intervention, which is often inaccurate and time-consuming. To improve accuracy and efficiency in response to the growing sector, autonomous detection methods are essential [1].

Previous studies comparing three deep learning algorithms—Fully Convolutional Neural Network (FCN), U-net, and SegNet—have demonstrated the potential of autonomous void detection, with FCN outperforming the others. Despite this, the challenge of significant class imbalance, where the majority of labels belong to the background class, remains an issue for accurate void detection [2], [3]. Due to this highly imbalanced class distribution, the models were not able to effectively learn the characteristics of voids, causing them to fail in recognizing voids precisely and accurately.

This research aims to provide a solution to the limitations of traditional methods while addressing the issue of class imbalance found in previous deep learning studies. We aim to improve the accuracy, Intersection over Union (IoU), and Boundary F1 (BF) scores for both the background class and void class. To achieve this, we propose a novel approach that integrates several advanced techniques within a semi-supervised learning framework:

Image Enhancement: Initially, image enhancement techniques are applied to improve the quality and clarity of the CT-scanned images of COSBs. These techniques help reduce noise and highlight relevant features, making the subsequent steps more effective.

Convolutional Kernels: Convolutional kernels are employed to extract meaningful features from the enhanced images. These kernels help capture edges, textures, and other important characteristics crucial for accurate void detection.

K-means Clustering: The extracted features are then segmented using k-means clustering, which helps address the issue of class imbalance by isolating voids from the overwhelming background data. This unsupervised learning step reduces the complexity of the classification problem.

Local XGBoost Classifiers: Finally, local XGBoost classifiers are trained on the segmented features within each cluster. These classifiers focus on specific regions of the feature space, improving the precision and accuracy of void detection.

This integrated approach aims to overcome the limitations

Yi Sun is with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089 USA (e-mail: ysun9487@usc.edu).

of existing deep learning methods by leveraging the strengths of both unsupervised and supervised learning techniques. By improving the detection and classification of voids in COSBs, this research contributes to more reliable material quality assessment and characterization, potentially leading to better quality control and reduced defects in COSB manufacturing.

II. Related Work

A. Voids Detection in COSBs

Voids in Composite Oriented Strand Boards (COSBs) are critical defects that negatively impact mechanical properties. Traditionally, void detection has relied on manual inspection and basic image processing techniques. It is typically processed in two steps: First, tomography is used to detect porosity through C-scan visualization of regions with lower density. Next, trained experts manually identify and inspect void morphology, location, volume fraction, and distribution. However, the traditional method is often time-consuming, labor-intensive, and prone to human error. As a result, there is a growing need for automated void detection solutions that can enhance accuracy and efficiency.

B. Deep Learning Approaches

Recent advances in deep learning have shown great potential in automating the detection and analysis of voids in composite materials. Various neural network architectures have been explored for this purpose.

1) Fully Convolutional Networks (FCNs): Fully Convolutional Networks (FCNs) are designed for pixel-level classification tasks and have been successfully applied to semantic segmentation problems. FCNs transform image pixels into pixel classes using only convolution (subsampling and upsampling) without the need for fully connected layers, making them suitable for varying input image sizes, which allows more diverse training data. Studies have demonstrated that FCNs can achieve high accuracy in void detection by providing detailed boundary-aware segmentation [2].

2) SegNet: SegNet is a deep convolutional encoder-decoder architecture specifically designed for semantic segmentation. It includes an encoder network for feature extraction and a corresponding decoder network for pixel-wise classification. SegNet is known for its ability to handle large-scale and high-resolution images, making it suitable for applications requiring detailed scene understanding. However, SegNet may over-segment voids and fail to contour their shape, resulting in larger detected areas than the actual voids [2].

3) U-net: U-net is another popular architecture variant to FCN and was originally designed for biomedical image segmentation. It features an encoder-decoder structure that enables precise localization of objects within an image. U-net has been shown to perform well in identifying small-scale voids but may encounter difficulties with continuous boundary depiction, particularly when dealing with irregular shapes and perturbations [2].

C. Challenges and Limitations in Existing Methods

1) Data Annotation and Human Intervention: Traditional void detection methods require extensive human intervention for data inspection, which is time-consuming and subject to human error [3].

2) Class Imbalance: One of the primary challenges for machine learning and deep learning models in void detection is the significant class imbalance in the dataset. The majority of pixels in the images belong to the background class, making it difficult for models to learn the characteristics of the void class accurately and effectively. This imbalance may still result in decent global accuracy, but this is misleading because it could just classify all pixels as the background and still have a high global accuracy. The imbalance issue leads to low accuracy and precision in the void class, as the models are biased towards the background class [2].

D. Semi-Supervised Solution

While deep learning approaches have significantly advanced the field of void detection in COSBs, challenges such as class imbalance and the need for extensive human intervention remain. This research aims to address these challenges by integrating advanced image enhancement techniques, convolutional kernels, k-means clustering, and local XGBoost classifiers within a semi-supervised learning framework. By leveraging the strengths of both unsupervised and supervised learning methods, this study seeks to reduce the impact of the imbalance in the dataset and improve the accuracy and efficiency of void detection, contributing to more reliable material quality assessment and characterization in COSB manufacturing.

III. Methodology

A. Image Enhancement and Preprocessing

In order to make models better at recognizing the COSB scanned images, preprocessing is often necessary to enhance the input data. The primary goals of these preprocessing techniques include noise reduction, detail smoothing, edge preservation, and contrast enhancement.

The Bilateral Filter was chosen for its ability to smooth the image while preserving edges. This filter operates through a nonlinear combination of nearby pixel values, combining gray levels or colors based on both geometric closeness and color similarity. It combines both spatial and range Gaussian filters. The spatial Gaussian acts like a traditional Gaussian filter, assigning larger weights to nearby pixels and smaller weights to distant pixels, while the range Gaussian assigns more weight to pixels that are similar in color to the central pixel. The combination of two Gaussian filters allows the filter to preserve edges and significant details while smoothing areas of gradual color variation [4].

A further detailed performance comparison of different image enhancement filters will be presented in an ablation study in the experimental section.

B. Convolutional Kernels

Covolutional kernels are designed to capture different aspects of the image by extracting features from images. It is mathematically represented as follows:

$$O(x,y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} I(x-i, y-j) \cdot K(i,j) \tag{1}$$

where:

- $O(x,y)$ is the output pixel value at position (x,y),
- $I(x-i, y-j)$ denotes the pixel value from the input image at position $(x-i, y-j)$,
- $K(i,j)$ is the convolution kernel applied over the image, affecting pixels within the range defined by a and b, the kernel's half-width and half-height respectively.

This operation slides the kernel K over the entire image I, applying it to every pixel location (x,y). The effect of applying this kernel is to extract features like edges, textures, or other relevant characteristics based on the nature of K. In our context, this step is crucial for highlighting the boundaries of voids in COSBs or enhancing other features that help differentiate between voids and the background in the pixel-wise classification process [7].

In our study, the feature extraction process went through 2 pass of edge detection kernel, which includes simple Edge Detection Filters, Bottom Sobel Filters, Left Sobel Filters, Right Sobel Filters, Sharpen Filters, and Emboss Filters, 1 pass of general kernel, which includes Box Blur, Gaussian blur (3x3 and 5x5), and Unsharp masking, and 1 pass of 10 random 5x5 random kernels, each targeting specific features:

1) Edge Detection: Edge detection kernels are applied to highlight the boundaries within the images. This is critical for identifying the edges of voids against the composite background.

- **Edge Detection Kernel 1:**

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

  This kernel enhances vertical and horizontal edges by emphasizing differences in pixel intensities.

- **Edge Detection Kernel 2:**

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

  This variant enhances edges from all directions, providing a more uniform detection of boundary features.

2) Sobel Filters: Sobel filters are used to detect edge directionality, which helps in understanding the orientation of structures within COSBs.

- **Bottom Sobel Filter:**

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

  This filter detects edges that are horizontally oriented, which are crucial for horizontal void boundaries.

- **Right Sobel Filter:**

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

  This filter enhances vertical edges on the right side, aiding in vertical boundary detection of voids.

3) General Purpose Kernels: General kernels like blurring and sharpening are applied to enhance or suppress features across the image uniformly:

- **Gaussian Blur 3x3:**

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

  Gaussian blurring helps in reducing image noise and detail, which can be essential for improving the performance of subsequent classification algorithms by reducing the effect of noise on the feature extraction process.

- **Sharpen:**

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

  Sharpening increases the contrast at edges, which is beneficial for emphasizing the boundaries of voids in the processed images.

4) Random Kernels: Introduced to explore additional textural patterns that might be indicative of voids.

These convolutional kernels are systematically applied to extract and enhance various features from the CT-scanned COSB images, providing a more in-depth dataset for clustering and classification.

C. Semi-supervised Learning Framework

To address the limitations from previous studies, this study explores a semi-supervised learning framework that integrates k-means clustering with local XGBoost classifiers.

1) K-Means: The K-Means algorithm partitions the feature space into K distinct clusters by minimizing the within-cluster sum of squares (WCSS). Mathematically, the objective function for K-Means is given by:

$$J = \sum_{k=1}^{K} \sum_{\mathbf{x} \in S_k} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2 \tag{2}$$

where $\mathbf{x}$ represents a feature vector in the dataset, S_k is the set of all feature vectors assigned to cluster k, and $\boldsymbol{\mu}_k$ is the centroid of cluster k [8].

This method aims to effectively reduce the data's dimensionality and variance, isolating the voids from the overwhelming background based on the similarity of the features.

The choice of clustering was determined by simplicity and efficiency. K-means can efficiently cluster the data because of its simple structure and unsupervised nature, which allows it to process large datasets and reduce the impact of class imbalance in a small amount of time.

To initialize the centroids for the K-means algorithm, we used the 'greedy k-means++' method. The greedy k-means++ initialization algorithm improves the convergence speed and the final clustering results by selecting initial cluster centers that are distant from each other.

The greedy k-means++ algorithm works as follows. In every step, we sample ℓ candidate centers $c_{i+1}^1, \ldots, c_{i+1}^\ell$ from the constructed distribution, not just one. Next, for each candidate center c_{i+1}^j, we compute the new cost of the solution $\varphi(X, C \cup \{c_{i+1}^j\})$ if we add this candidate to our set of centers. Then we pick the candidate center that minimizes this expression.

The steps for the greedy 'k-means++' initialization are as follows:

Greedy K-Means++ Initialization:

Algorithm 1 Greedy k-means++ seeding

Input: X, k, ℓ
Output: C
Uniformly independently sample $c_1^1, \ldots, c_1^\ell \in X$
Let $c_1 = \arg\min_{c \in \{c_1^1, \ldots, c_1^\ell\}} \varphi(X, c)$ and set $C_1 = \{c_1\}$
for $i \leftarrow 1, 2, \ldots, k-1$ **do**
    Sample $c_{i+1}^1, \ldots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\varphi(x, C_i)}{\varphi(X, C_i)}$
    Let $c_{i+1} = \arg\min_{c \in \{c_{i+1}^1, \ldots, c_{i+1}^\ell\}} \varphi(X, C_i \cup \{c\})$ and set $C_{i+1} = C_i \cup \{c_{i+1}\}$
end for
return $C = C_k$

This method offers significant improvements in convergence speed and final clustering performance compared to random initialization [9].

By segmenting images into clusters using greedy K-Means before classification, this approach aims to reduce the impact of class imbalance by isolating the voids from the overwhelming background data based on the similarity of the features, thereby enhancing the local classifier's classification precision and accuracy. In addition, by clustering the data first, we also reduce the complexity of the problem. Each cluster represents a subset of the data that presumably shares certain characteristics. This allows classifiers to focus on specific regions of the feature space, which can simplify the classification task because the variability within each cluster is likely reduced compared to the variability in the entire dataset.

2) Local XGBoost Classifiers: Following clustering, local XGBoost classifiers are trained on each cluster's data. XGBoost (Extreme Gradient Boosting) operates by constructing a sequence of models that attempt to correct the errors of the previous models in the ensemble. The objective function optimized by XGBoost is:

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k} \Omega(f_k), \tag{3}$$

where l is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i, and Ω represents the regularization term that penalizes the complexity of the model, with f_k denoting the k-th tree [10].

The choice of local classifier was based on XGBoost's speed and efficiency, and accuracy. XGBoost is highly optimized for performance and speed. It can process large datasets significantly faster than other ensemble methods. In addition, unlike other bagging tree models that build models in isolation, XGBoost corrects previous models' errors in a sequential manner. This approach ensures that each successive model focuses on the hardest examples, which refines the classifiers' ability to detect nuanced features in void detection with higher precision [10].

Local XGBoost classifiers focus on data within their respective clusters, addressing the class imbalance by learning cluster-specific features that might be overlooked by a global classifier. This approach enhances the model's capability to detect voids with higher precision and accuracy, as it tailors the learning process to the unique characteristics of each cluster.

D. Semi-supervised

The semi-supervised framework leverages the unsupervised learning capability of K-Means to structure the data into manageable subsets, which significantly enhances the supervised learning process performed by XGBoost. By segregating the data into homogeneous groups, the model complexity is reduced, leading to faster convergence and improved generalization of the classifiers across unseen data. The reduction in inter-cluster variability allows XGBoost to efficiently model the intra-cluster distributions, thereby boosting the overall performance in detecting and classifying voids in COSBs.

IV. Experimentation

A. Dataset

The dataset comprises 8 high-resolution computed tomography (CT) scanned images of Composite Oriented Strand Boards (COSBs), each varying in size. Each image is associated with a ground truth segmentation label matrix indicates the presence of voids, annotated by domain experts. This relationship between images and labels is mathematically formulated as follows:

We denote each image as a tuple $(\mathbf{x}, \mathbf{y})$, representing an element of the dataset $\mathcal{X}$. Here, $\mathbf{x}$ symbolizes an individual image, and $\mathbf{y}$ represents its corresponding segmentation label. Both $\mathbf{x}$ and $\mathbf{y}$ are matrices of dimensions $H \times W$, where H and W are the height and width of the image in pixels, respectively.

$$\mathbf{x} \in \mathbb{R}^{H \times W}$$

Each element in $\mathbf{x}$ corresponds to a pixel value indicating the grayscale intensity at that particular location within the image.

$$\mathbf{y} \in \{0, 1\}^{H \times W}$$

Each element of $\mathbf{y}$, denoted as y_{ij}, is a binary label corresponding to the pixel (i, j) in the image $\mathbf{x}$. The label $y_{ij} = 1$ signifies that the pixel is part of a void, and $y_{ij} = 0$ indicates background.

This structured representation allows for precise pixel-wise classification, essential for analyzing and processing the voids within the COSBs through computational methods.

B. Ablation Study on Image Enhancement Filters

An ablation study was conducted to compare the performance of different image enhancement filters. We applied a set of image enhancement filters, including Contrast Limited Adaptive Histogram Equalization (CLAHE), Gaussian Filter, Mean Filter, Unsharp Masking, and Bilateral Filter. Since we will be using a semi-supervised approach that involves using KMeans in the later classification section, here we will use KMeans to evaluate the effectiveness of different image enhancements.

It has been shown that CLAHE allows for the detection of clinical anomalies, as effectively as interactive intensity windowing, by providing enhanced visual differentiation of lesions in CT chest studies [4]. Mean while, Unsharp Masking enhances high-frequency components by adding a filtered image to the original image. On the other hand, smoothing effect, which serves as noise removal, is also crucial in image preprocessing. Gaussian Filter removes noise by smoothing and following the curve of the image. Similarly, Mean Filter apply smoothing by averages the values of pixels within a defined window size to calculate the central point. The Bilateral Filter stands out for its edge-preserving capability while smoothing the image, combining pixel values based on geometric distance and photometric similarity. It operates on all image bands simultaneously, preventing phantom effects on edges [5].

To evaluate how well these filters performed compareed to each other and the original raw images, we uses the training data and label to evaluate the results of kmeans clustering with 2 clusters, background and non-background. The comparative results are summarized in Table I and the visualizations are shown in Fig 1.

Among different methods, the Bilateral Filter has the highest accuracy, F1-Micro, F1-Macro, and Precision. It also has the second highest Recall. Based on the evaluation result, we would pursue with Bilateral Filter as our image enhancement preprocessing method.

C. Implementations

1) Feature Extraction: After image preprocessing, we applied feature extraction to the enhanced filtered images. We utilized convolutional kernels to perform blurring, sharpening, and edge detection. The general kernel is applied once and the edge detection kernels are applied twice to the entire images. To capture a diverse range of features, we also generate a set of random 5x5 kernels. In total, we applied F-1 different kernels to generate F features. During convolution, the kernel matrix moves through the entire image matrix from left to right and top to bottom. At each step, it returns a single pixel value as a result of the convolution operation, a feature corresponding to the area of the image under the kernel [3].

a) Convolution Process: The convolution process is applied to each image in the dataset, represented by the tuple $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} \in \mathbb{R}^{H \times W}$ is the image matrix and $\mathbf{y} \in \{0,1\}^{H \times W}$ is the corresponding label matrix indicating voids and background. Specifically, using Equation 1:

- $\mathbf{x}(x-i, y-j)$ refers to the intensity of the pixel at position $(x-i, y-j)$.
- $K(i,j)$ alters the pixel values within its reach, impacting the resultant feature map.
- $O(x,y)$ is computed as the sum of the products, producing a feature-enhanced output at each pixel.

The kernel matrix K moves across the entire image matrix $\mathbf{x}$, from left to right and top to bottom. At each position (x, y), the kernel is applied by centering it on (x, y) and multiplying the overlapping image pixels by the corresponding kernel weights. The results are summed to produce a single pixel value in the output feature map O, capturing the enhanced characteristics of the underlying area.

The enhanced features include both the high-frequency components emphasized by the edge detection and random kernels, as well as the original pixel values, which are preserved to retain the inherent information of the image. Finally, the features extracted for each image will be combined to form a feature vector that represents the enhanced image in a transformed feature space.

2) Semi-supervised Learning: Since we already extracted the feature, we can shuffled the data and split for training and testing. After splitting, the data were standardized to normalize the feature space, ensuring that the magnitude of the features does not bias the k-means clustering algorithm. Next, K-Means clustering is applied to the scaled training data to partition it into two clusters. The same clustering model is then used to assign cluster labels to the scaled test data. This step helps in reducing the complexity of the classification problem by grouping similar data points together. Now, we have a cluster of background and a cluster of voids for both the training and testing data.

For each cluster identified by K-Means in the training set, a separate individual XGBoost classifier is trained. To optimize the performance of each local XGBoost classifier, we employed grid search cross-validation to iterate over a predefined set of hyperparameters. This included variations in maximum depth, subsample ratios, and the learning rate. The grid search was performed within each cluster's training subset, ensuring that the best parameters were cluster-specific and tailored to the unique distribution of data within each cluster. The best combination of hyperparameters that maximizes the area under the ROC curve were stored and used for the test clusters.

Once the best XGBoost model is found for each cluster, the function uses these models to predict the probabilities of the test data belonging to the voids class. A specific precision threshold is chosen to determine the optimal decision threshold for each cluster. The optimal decision threshold was determined by identifying the elbow point where both precision and recall are high. The target threshold of 0.89 for precision is determined, ensuring that the recall was maximized without compromising the precision unduly.

3) Semi-supervised Learning: Since we already extracted the features, we can shuffle the data and split it for training and testing. After splitting, the data were standardized to normalize the feature space, ensuring that the magnitude of the features does not bias the k-means clustering algorithm. Next, K-Means clustering is applied to the scaled training data to partition it

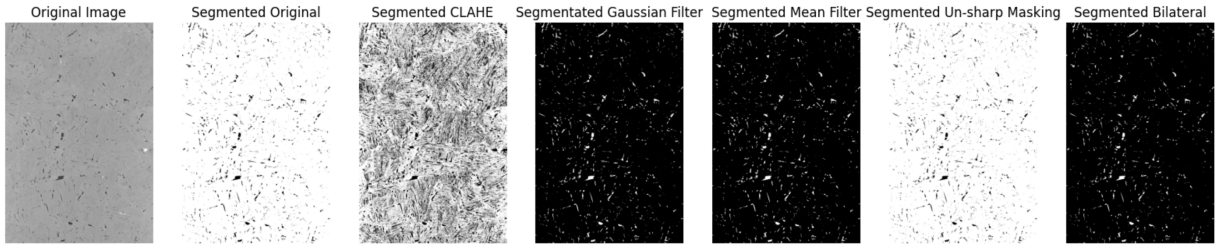

Fig. 1. Comparison of image enhancement techniques: original image and enhanced results using CLAHE, Gaussian filter, Mean filter, Unsharp Masking, and Bilateral filter.

TABLE I
COMPARISON OF K-MEANS CLUSTERING PERFORMANCE WITH DIFFERENT IMAGE ENHANCEMENT TECHNIQUES

Enhancement Method	Accuracy	F1-Micro	F1-Macro	Precision	Recall
Original K-Mean	0.87842	0.87842	0.636547	0.208042	0.930439
Gaussian Filter + K-Mean	0.98128	0.98128	0.883497	0.648877	**0.967436**
Unsharp Masking + K-Mean	0.790289	0.790289	0.549185	0.12547	0.875935
Mean Filter + K-Mean	0.979645	0.979645	0.873655	0.63197	0.946611
Bilateral Filter + Kmean	**0.984201**	**0.984201**	**0.896585**	**0.694667**	0.946877
CLAHE + K-Mean	0.541304	0.541304	0.382013	0.036636	0.499118

into two clusters. The same clustering model is then used to assign cluster labels to the scaled test data. This step helps in reducing the complexity of the classification problem by grouping similar data points together. Now, we have a cluster of background and a cluster of voids for both the training and testing data.

For each cluster identified by K-Means in the training set, a separate individual XGBoost classifier is trained. XGBoost, or Extreme Gradient Boosting, is a powerful machine learning algorithm that uses gradient boosting framework to build a series of weak learners, typically decision trees, to improve model performance. To optimize the performance of each local XGBoost classifier, we employed grid search cross-validation to iterate over a predefined set of hyperparameters. This included variations in maximum depth, subsample ratios, and the learning rate:

Maximum Depth (`max_depth`): This parameter controls the maximum depth of each tree in the ensemble. Increasing the depth allows the model to capture more complex patterns in the data but can also lead to overfitting. In our grid search, we varied `max_depth` between 3 and 10.

Learning Rate (`learning_rate`): This parameter controls the step size at each iteration while moving towards a minimum of the loss function. Lower values make the model more robust to overfitting but require more boosting rounds. We tested learning rates ranging from 0.01 to 0.3.

Subsample Ratio (`subsample`): This parameter specifies the fraction of training instances used to grow each tree. Setting it to a value less than 1.0 can prevent overfitting by introducing randomness into the training process. We varied the subsample ratio between 0.5 and 1.0.

The grid search was performed within each cluster's training subset, ensuring that the best parameters were cluster-specific and tailored to the unique distribution of data within each cluster. The best combination of hyperparameters that maximizes the area under the ROC curve (AUC) was stored and used for the test clusters.

Once the best XGBoost model is found for each cluster, the function uses these models to predict the probabilities of the test data belonging to the voids class. A specific precision threshold is chosen to determine the optimal decision threshold for each cluster. The optimal decision threshold was determined by identifying the elbow point where both precision and recall are high. The target threshold of 0.89 for precision is determined, ensuring that the recall was maximized without compromising the precision unduly.

V. RESULTS AND DISCUSSION

A. Evaluation Metrics

To evaluate the result of our semi-supervised model on the COSB test dataset, we compared the accuracy, Intersection over Union (IoU), and Boundary F1 (BF) score with the results of the previous study (Baseline Results).

Accuracy provides a general measure of the model's ability to correctly predict pixels as either voids or background. Intersection over Union (IoU) score measures the accuracy of prediction based on corresponding ground truth [2]. It is calculated as:

$$\text{IoU} = \frac{\text{Ground Truth} \cap \text{Prediction}}{\text{Ground Truth} \cup \text{Prediction}} = \frac{2}{13} = 0.15$$

Boundary F1 (BF) score is a contour matching measurement between a predicted boundary of an object and the associated ground truth boundary [2]. The following formula is used:

$$\text{BF score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Recall} + \text{Precision}}$$

We also measured Precision, Recall, and F1-Macro score. Precision indicates the percentage of the voids or background the model predicted were actually voids or background. Recall indicates the percentage of the actual voids or background were correctly identified by the model. F1-Micro measures the overall performance of the model and F1-Macro score

calculates the mean of the F1-scores per class, giving equal weight to each class, which highlights the model's ability to perform well across all classes, irrespective of their frequency in the dataset.

B. Results

The best-performed model among the three neural networks in the baseline result from the previous study was FCN, which yielded the most precise boundary and successfully classified most voids [2]. However, the results of the global measurements, specifically mean accuracy (0.4929), mean IoU (0.4701), and mean BF Score (0.4399) are not satisfactory. Additionally, local measurements of the void class, such as accuracy (0.0133), IoU (0.0082), and Mean BF Score (0.1112), are all very low.

In this study, we observe a significant improvement across all metrics using the semi-supervised framework combining KMeans and local XGBoost classifiers. The global accuracy improves from 0.9320 (FCN) to 0.9930, mean accuracy from 0.4929 to 0.9492, mean IoU from 0.4701 to 0.9023, and mean BF Score from 0.4399 to 0.9462.

Specifically, for the void class, the accuracy improves from 0.0133 to 0.9023, IoU from 0.0082 to 0.8118, and Mean BF Score from 0.1112 to 0.8961. These improvements highlight the effectiveness of the proposed semi-supervised framework in addressing the class imbalance and improving the overall performance of void detection.

C. Discussion

The semi-supervised approach combining KMeans clustering and local XGBoost classifiers has shown significant improvements over the previously studied deep learning methods. The results demonstrate that our method effectively addresses the class imbalance issue and enhances the accuracy, IoU, and BF scores for void detection in COSBs. This indicates a promising direction using the semi-supervised framework in pixel-wise void detection. By improving the detection and classification of voids in both speed and accuracy, our study contributes to effective and reliable material quality assessment and characterization. This can help scientists gain deeper understandings of void and its evolution, potentially reducing defects and improving the mechanical properties of the final product.

While our method has shown promising results, there are several limitations to consider. First, the effectiveness of KMeans clustering may vary with different types of image data. Second, the local XGBoost classifiers' performance might relied heavily on the quality of the features extracted, which might not capture all relevant and detailed information of the image.

Future research can explore several avenues based on our findings. One potential direction is to investigate different types of convolutional kernels, capturing different aspects of the image. Additionally, exploring the applicability of our method to other types of defects and composite materials would also be valuable.

VI. Conclusion

In this study, we utilized bilateral filter to enhance our image data, then we employ convolutional kernels to extract features that captures the important information of the image, finally we applied kmeans clustering to the image pixels and build a independent local XGBoost classifier for each cluster.

The result of the model not only surpass the global measurements of previous best model, FCN, but also shows a significant improvement particularly on the void class measurements, which was very low due to the imbalance of classes.

The effectiveness of semi-supervised learning shows a promising direction for pixel-wise image classification, even if the data are highly imbalance. It could be further extended to multi-class defects detection and classification with the increment of numbers of clusters.

Moreover, the methodology can be adapted to other types of composite materials and defect detection tasks, broadening its applicability.

References

[1] R. A. Smith, "Composite defects and their detection," *Materials Science and Engineering*, vol. III, Encyclopedia of Life Support Systems (EOLSS), 2009.

[2] W. Hu, X. Wang, C. Bowland, P. Nguyen, C. Li, S. Nutt, and B. Jin, "Deep learning for void detection in composite oriented strand board," in *Conference proceeding*, CAMX Anaheim, United States, October 17–20, 2022.

[3] W. Hu, X. Wang, Y. Eftekhari, S. Callander, C. Bowland, F. Nguyen, J. McCaslin, C. Schaal, G. X. Gu, C. Li, and B. Jin, "Application of k-nearest neighbors algorithms for void classification in composite oriented strand board," *Journal-article*.

[4] I. Ilić-Krstić, I. Shopovska, and Z. A. Ivanovski, "Image segmentation using bilateral filter," 2014.

[5] S. M. Pizer, R. E. Johnston, J. P. Ericksen, B. C. Yankaskas, and K. E. Muller, "Contrast-limited adaptive histogram equalization: speed and effectiveness," in *Proceedings of the First Conference on Visualization in Biomedical Computing*, Atlanta, GA, USA, 1990, pp. 337–345, doi: 10.1109/VBC.1990.109340.

[6] B. Tahir, S. Iqbal, M. U. Khan, T. Saba, Z. Mehmood, A. Anjum, and T. Mahmood, "Feature enhancement framework for brain tumor segmentation and classification," *Microscopy Research and Technique*, vol. 82, no. 5, pp. 803–811, 2019.

[7] I. Namatēvs, "Deep Convolutional Neural Networks: Structure, Feature Extraction and Training," *Information Technology and Management Science*, vol. 20, pp. 40–47, 2017, doi: https://doi.org/10.1515/itms-2017-0007.

[8] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, vol. 622, pp. 178–210, 2023, doi: https://doi.org/10.1016/j.ins.2022.11.139.

[9] C. Grunau, A. A. Özüdoğru, V. Rozhoň, and J. Tětek, "A Nearly Tight Analysis of Greedy k-means++," *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1012–1070, 2023, doi: https://doi.org/10.1137/1.9781611977554.ch39.

[10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

TABLE II
GLOBAL SEGMENTATION METRICS

Method	Global Accuracy	Mean Accuracy	Mean IoU	Weighted IoU	Mean BF Score
KMeans + XGBoost	**0.9930**	**0.9492**	**0.9023**	**0.9867**	**0.9462**
FCN	0.9320	0.4929	0.4701	0.8930	0.4399
SegNet	0.8258	0.4685	0.4219	0.7912	0.4316
U-net	0.9058	0.4908	0.4611	0.8682	0.4288

TABLE III
SEGMENTATION METRICS BY CLASS

Method	Label Area	Accuracy	IoU	Mean BF Score
KMeans + XGBoost	Void	**0.9023**	**0.8118**	**0.8961**
KMeans + XGBoost	Background	**0.9961**	**0.9927**	**0.9964**
FCN	Void	0.0133	0.0082	0.1112
FCN	Background	0.9724	0.9319	0.7686
SegNet	Void	0.0782	0.0186	0.1527
SegNet	Background	0.8587	0.8252	0.7104
U-net	Void	0.0376	0.0165	0.1709
U-net	Background	0.9440	0.9056	0.6867

TABLE IV
SEMI-SUPERVISED BINARY CLASSIFICATION RESULT

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Background	0.9966	0.9961	0.9964	9983808
Voids	0.8900	0.9023	0.9000	347607
Overall	**Score**			
F1-Macro	0.9460430390157714			