Architecture Analysis of Rice Panicle Using Deep Learning

Lam Thai Nguyen   Trung Kien Pham

March 14, 2024

I Introduction

Plant phenotyping entails the measurements and analysis of observable, predefined traits of plants. In the field of rice inflorescence, this process encompasses the examination of morphological features, including panicle length, the number and order of branches, as well as the count and sizes of grains [1]. These insights provide researchers with valuable information to explore the diversity of rice panicle resources and their morphological traits, which play a crucial role in tasks such as rice breeding and yield optimization. The branching pattern of rice panicles is intricately shaped by a complex interplay of environmental factors, including climatic conditions, soil composition, and light availability, alongside the influential roles played by genes and hormonal regulation. The architectural composition of a rice panicle comprises essential components such as the main axis (rachis), primary branches (Pb), secondary branches (Sb), tertiary branches (Tb), and even higher order branches. These distinct elements are delineated as a blue line, black lines, green lines, orange lines, respectively (Figure 1). A typical rice panicle is characterized by a central main axis, to which lateral branches (*i.e.*, primary branches) are attached. Subsequently, these primary branches bear secondary branches, further giving rise to higher order branches. Nodes (ARM, Aborted Rachis Meristem), represented by red dots (Figure 1), along with contact points of branches of different order, are hereinafter referred to as junctions.

In the tasks of measuring the aforementioned traits, as well as detecting and classifying junctions, traditional methods often require manual labor, a process that can be time-consuming and potentially detrimental to the plants. Considering the importance of delving into the diversity of rice panicle resources and the pressing need to acquire their ecological features, there remains a critical gap for the development of an accurate and efficient vision-based approach for extracting rice panicle junctions without relying on intensive equipment.

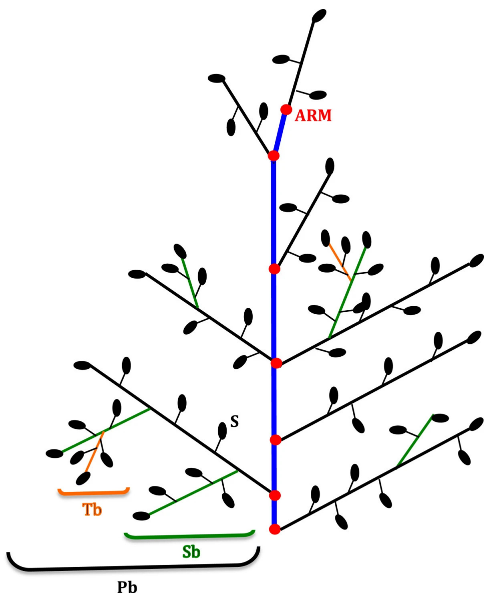


Figure 1: A rice panicle structure.

Source: Adapted from [1].

Recent research in the field of crack detection has showcased remarkable image-based methods for automatically identifying target cracks on concrete surfaces. Some of the state-of-the-art crack detection models include HED [2], U-net [3], Deepcrack [4], RUC-Net [5], BSNBM [6], U2cracknet [7], among others [8–17]. Cracks in infrastructure, including roads, bridges, and buildings, require regular detection and maintenance to minimize potential hazards and ensure the safety of traffic. However, the risky, time-consuming tasks are still carried out through manual visual inspection. This challenge emphasizes the necessity for an automatic and reliable method to precisely detect cracks on surfaces of infrastructure, ensuring safety and efficiency in maintenance practices. As a result, novel crack detection algorithms have been introduced and obtained satisfactory results signify a superior approach compared to the traditionally laborious methods that have been employed. By leveraging the innovative capabilities of crack detection algorithms, we can facilitate the task of identifying rice panicle junctions.

Concrete cracks and rice panicles share several similarities in their visual characteristics. Both exhibit characteristics of thin continuous lines, with cracks forming distinct linear patterns on concrete surfaces, while rice panicles feature slender stalks and branching structures. In terms of pixel intensity, both concrete cracks and rice panicles tend to appear darker than their respective backgrounds, creating a distinct contrast. This shared set of features provides a basis for leveraging crack detection models developed for concrete surfaces, such as RUC-Net [5], to effectively identify and extract the branching structures of rice panicles, demonstrating the transferability of image-based methodologies across diverse domains. In Figure 2, the identified features are visually highlighted by comparing the crack detected by Deepcrack [4] (Subfigure 2b) in Subfigure 2a with the skeleton (Subfigure 2d) extracted from the rice panicle in Subfigure

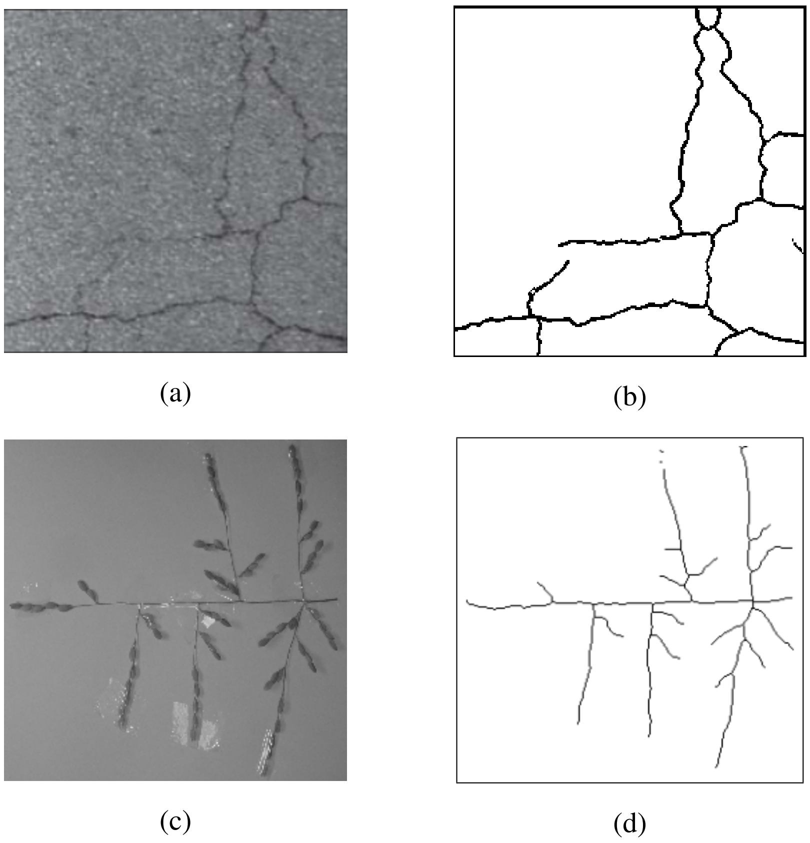


Figure 2: Similarities between concrete cracks and rice panicles.
(a) Original crack image. (b) Detected crack.
(c) Original rice panicle image. (d) Detected rice panicle structure.

Source: Subfigure 2a is adapted from [4].

2c. This visual representation effectively illustrates the shared characteristics, such as the thin continuous lines and the branching pattern, further emphasizing the potential applicability of crack detection algorithms in capturing and analyzing similar features in rice panicle structures. Since crack detection models only provide us with the binary image of the rice panicle (Subfigure 3b), additional image processing techniques are required to extract the junction positions, including clustering algorithms and skeletonization algorithms.

In this paper, we propose a novel vision-based approach for detecting rice panicle junctions. This approach capitalizes on a state-of-the-art crack detection algorithm, complemented by skeletonization and clustering methods. The rest of this paper is organized as follows. Section II presents the methodology, related image-based algorithms as well as the dataset and used evaluation metrics. Specifically, in Subsection 2, we introduce a novel crack detection model named RUC-Net [5], employed to extract the binary image from the initial rice panicle. Subsection 3 elaborates on the clustering algorithms utilized for junction detection. Subsection 4 describes the two skeletonization algorithms employed to thin the binary image of rice panicles generated by crack detection models. Section III discusses the experimental results of our work.

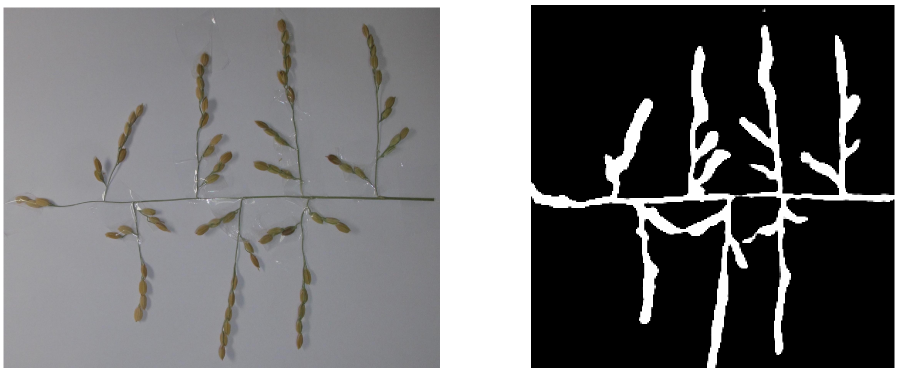

(a) (b)

Figure 3: Rice panicle binary image produced by crack detection model.
(a) Original rice panicle image. (b) Extracted binary image.

II Methodology

1 Our proposed approach

The analysis of rice panicle architecture has proven to be a pivotal aspect in delving into the diversity of its resources and morphological traits. This exploration provides researchers with valuable insights and new methodologies to enhance rice breeding practices and optimize yield. Nevertheless, the current analysis practice relies on manual inspection, a process that can be time-consuming, prone to inaccuracies, and potentially detrimental to the plants. Recently, an image-based approach, *P-TRAP* [1], has demonstrated satisfactory results in extracting the ecological features of rice panicles through analyses of panicle structure, spikelet/grain counting, and seed shape examination. While this method has achieved notable success, it lacks automation in tasks related to identifying rice panicle junctions and may, to some extent, depend on the user's prior knowledge of the examined plants.

In this Section, we propose a novel vision-based approach that can automatically extract the junction position on a given image of rice panicle. Our proposed method capitalizes on a state-of-the-art crack detection model named RUC-Net [5], together with clustering algorithms and skeletonization algorithms to effectively process and extract the desired junctions. The image processing procedure consists of the following steps:

- **Step 1 (Binarizing)**: Considering the similarities between concrete cracks and rice panicle structures, we employ a crack detection model RUC-Net [5] to extract binary images from provided rice panicles. Obtained results from RUC-Net are presented at a fixed dimension of 256 x 256, preserving the structure of rice panicle. An illustration of this process is shown in Figure 3.
- **Step 2 (Thinning/Skeletonizing)**: Subsequent thinning step is necessary to obtain unitary

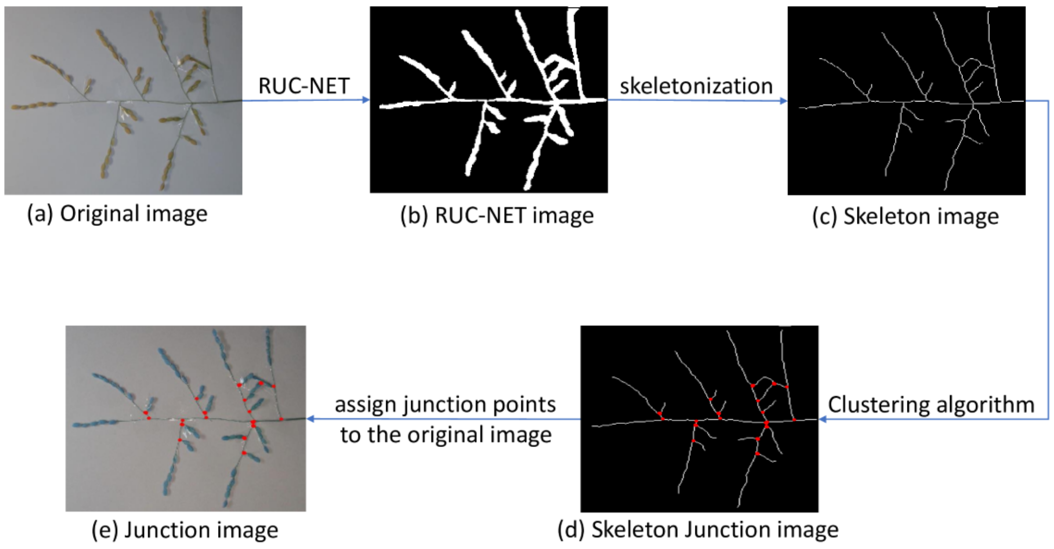


Figure 4: Rice panicle junction detection process.

thickness skeletons, which are later used for clustering algorithms. In this step, we employ 2 skeletonization algorithms in [18] and [19] to construct a crack-like, thin continuous line that preserves the shape and connectivity of rice panicle while removing a large portion of original white pixels.

- **Step 3 (Junction detection)**: Assuming he densest areas of a skeleton image lie around where different branches intersect, several clustering algorithms and image-processing methods are employed to detect the junction position. Identified junctions are then compared to pre-labelled junctions to evaluate the accuracy of the process.

In the task of detecting rice panicle junction, we utilize machine learning, deep learning and several image-processing techniques. The aforementioned procedure is simplified in Figure 4.

2 Crack detection algorithm

RUC-Net [5] is an U-shaped encoder–decoder network that combines elements from both the U-net [3] and Resnet architectures. In order to reduce the model parameters and computational complexity, the final channel number of RUC-Net was 512 after four downsamplings. The residual block from ResNet replaces the two 3×3 convolution layers in the encoder of the original Unet. This modification allows RUC-Net [5] to extract more precise crack feature information. RUC-Net incorporates the scSE attention module to enhance detection performance. The scSE module improves the network's ability to focus on relevant spatial and channel information. To address class imbalance in pavement crack segmentation, RUC-Net [5] utilizes the focal loss function. This helps improve the network's ability to handle challenging crack patterns. The rice panicle binary image produced by RUC-Net are shown in Figure 3b.

3 Clustering algorithms

Clustering, a fundamental data analysis technique that partitions objects into subsets (*i.e.*, clusters) based on their similarities, holds significant importance across various fields. This algorithm plays a crucial role in diverse tasks including market segmentation, medical image analysis, email grouping, and fraud detection. Extensive research has been dedicated to the development of various clustering methods, each employing distinct approach to dataset analysis [20–25]. These include partitioning-based clustering algorithms, proximity-based clustering algorithms, hierarchical clustering algorithms, as well as more advanced methods aimed at dataset improvement [23] or capitalizing on semantic relations between objects [24].

Under the assumption that the densest areas of a skeleton image lie around where different branches intersect (*i.e.*, junctions), we can utilize clustering methods to automatically detect these subsets, as well as capturing the their centers. A sufficiently small subset can strongly indicate the position of the junction (*i.e.*, the cluster center) as represented by red dots in Figure 6. In this section, we will discuss 6 used clustering algorithms, along with 2 other image-processing techniques to detect rice panicle junctions. A brief description and the number of parameters required by each algorithm are reported in Table 1.

3.1 DBSCAN

This is a popular algorithm in machine learning. Clusters are formed from *core points* and *non-core points*. A point is considered a core point if the number of surrounding neighbors is greater than a pre-defined threshold. On the other hand, if it has fewer than *minPts* within epsilon distance, it is classified as a non-core point. A non-core point becomes a border point if it is in the neighborhood proximity of a core point, and it will be added to the cluster. If a non-core point is not a border point, it is considered *noise* and is ignored. Two parameters need to be defined in this algorithm: *minPts*, which represents the minimum threshold number of data points required to define a high-density epsilon neighborhood, and *epsilon*, which is a distance value used to determine the epsilon neighborhood of any data point. In other words, if the distance between two points is lower than or equal to epsilon, they are considered neighbors.

3.2 DP

The Density Peak algorithm uses high local density values and appropriate distances to find clusters. The local density, denoted as ρ_i, is equal to the number of points that are closer than the cutoff distance, *dc*, to point *i*. The distance from points of higher density, δ_i, is measured

Algorithms	Brief description	Parameter number
DBSCAN	Clusters are formed from core points when the number of surrounding neighbors exceeds a pre-defined threshold, and non-core points when they are within the pre-defined radius of a core point.	2
DP	Cluster centers are assigned to data points with high local density values and appropriate distances between each other.	1
DenMune	Seed points are used to construct the backbone of the clusters while non-seed points either succeed in merging with existing clusters or are marked as noise and removed from the clustering process.	1
HIAC	Ameliorating the dataset through gravitation which forces similar objects to move towards each other.	4
FINCH	Clusters are initially estimated based on the first neighbors' integer vector, which is then recursively chain-linked using the first neighbor of each data point.	0
DP-Dsets	Iteratively extracting a dense subset of a real cluster and expanding it until the final cluster is formed.	0
BWMORPH	Built-in MATLAB function for image processing.	0
Crossing number	Treating each object as the central cell in a 3-by-3 matrix and assign it to different subsets based on surrounding objects.	0

Table 1: Used algorithms and their attributes.

by computing the minimum distance between point *i* and any other point with a higher density, given by $\delta_i = \min_{j:\rho_j>\rho_i}(d_{ij})$. For the point with the highest density, we conventionally take $\delta_i = \max_i(d_{ij})$. Cluster centers are recognized as points for which the value of δ_i is abnormally large. DP (Density Peak) is cited as [21] and allows the finding of nonspherical clusters but works only for data defined by a set of coordinates and is computationally costly.

3.3 DenMune

Under the assumption that any data point should belong to the same cluster as its k-Mutual-Nearest-Neighbors (k-MNN), *DenMune* [22] introduces a novel method to meet the challenges associated with overlapping clusters or those with arbitrary shapes, varying densities. By following the principle of mutual nearest neighbor consistency and requiring a single parameter k from the user, *DenMune* possesses the capability to detect both target clusters and noise points. *DenMune* adopts the mechanism that classifies each data point into a subset of either strong points (seed points), weak points (non-seed points) or noise points. Subsequently, seed points are used to define the structure of the target clusters, while non-seed points either succeed in joining existing clusters or are marked as noise and excluded from the clustering process along with other noise points. The implementation of this mechanism ensures that *DenMune* produces robust clustering results with a flexible user-defined parameter k that has demonstrated stability across a wide value range.

3.4 HIAC

With an approach to ameliorate the dataset and make it more friendly to clustering algorithms, a clustering optimization method, *HIAC* (**H**ighly **I**mproving the **A**ccuracy of **C**lustering Algorithms) [23], was proposed to enhance the accuracy of clustering algorithms. Assuming that improving the principles of clustering algorithms alone is sub-optimal, *HIAC* presents a method to shrink the dataset by applying gravitational forces across the data points. This process forces similar data points to move towards each other, making the dataset more friendly to clustering algorithms, irrespective of their principles. Unlike existing dataset-ameliorating methods, *HIAC* distinguishes itself as the first to employ the mechanism of applying gravitational forces exclusively among valid-neighbors, preventing dissimilar data points from approaching one another. While a vast number of clustering algorithms performs poorly on certain datasets where the distances between data points within a cluster are similar those of data points from different clusters, causing clusters to overlap, initially applying *HIAC* to ameliorate the dataset has been proven to improve the accuracy of clustering algorithms up to 253.6% [23]. *HIAC* involves 1) Identifying valid-neighbors, where each data point's weight is computed, and those with weights exceeding the threshold (inferred from the decision graph of data points' weight probability distribution curve) are termed valid-neighbors. Valid-neighbors are later applied to the gravitational forces. 2) Objects motion, where similar points (*i.e.*, valid-neighbors) move towards their initial valid-neighbors, while invalid-neighbors remain in place. With its capability to detect noise points and isolate clusters, *HIAC* stands as a state of the art method to ameliorate datasets, hence achieving accurate clustering results.

3.5 FINCH

The *FINCH* [24] algorithm is a form of a single clustering equation that can directly discover groupings in the data. The main idea is that the first neighbor of each sample is all one needs to discover large chains and find the groups in the data. To achieve this, the algorithm computes an adjacency link matrix as described in [24]. *FINCH* provides a hierarchical structure as a small set of partitions where each successive partition is a superset of the preceding partitions. This method does not require any hyper-parameters, distance thresholds, or the need to specify the number of clusters.

3.6 DP-Dsets

The majority of existing clustering algorithms have some prominent defects when applied to real-world data. Specifically, they often rely on one or more parameters, and they may not be optimal choices for datasets characterized by the Gaussian distribution, a common feature in real-world data [25]. To address this challenge, a novel clustering algorithm named *DP-Dsets* [25] was introduced, combining two existing clustering algorithms - the dominant set (DSet) algorithm and an improved version of the density peaks algorithm (DP). By iteratively detecting a dense subset of a real cluster with the dominant set algorithm and subsequently expanding it using the density peaks algorithm until the final cluster is formed, *DP-Dsets* is able to capture clusters successively. With a parameter-free approach of Dset and the suitability of DP for data of Gaussian distribution, *DP-Dsets* designs a novel method tailored for clustering tasks in real-world data.

3.7 BWMORPH

This function is available in MATLAB and is used for finding branch points of objects with a branching structure, indicated by the input argument 'branchpoints'. It removes any pixels around the intersections. Furthermore, *BWMORPH* has various arguments that find applications in the field of image processing. In this paper, another argument called 'skel' will also be utilized. The 'skel' argument is used to remove pixels from the boundaries of objects without causing them to become separated. In simpler terms, it narrows the objects down to a line width of 1 pixel, which is also the terminology used in this paper.

3.8 Crossing number

As a specialized method designed for single-line-width binary images, *Crossing number* analyzes each pixel in an image and assigns it to distinct subsets based on its surrounding pixels. The process includes considering each pixel as the central cell of a 3-by-3 matrix, with each of the 8 surrounding pixels labelled as $P_{i:1\to 8}$ (Figure 5). Given the intensity value of each surrounding pixel (0 or 1), for each pixel P with high intensity, the crossing number value CN is then calculated by Equation 1, where $P_9 = P_1$.

$$CN = \frac{1}{2}\sum_{i=1}^{8} |P_{i+1} - P_i| \tag{1}$$

The value of CN and its corresponding characteristic are described in Table 2. As *Crossing number* is specifically tailored for single-line-width binary images, any examined pixel can only be surrounded by 0 to 4 other pixels. A pixel is considered isolated when it has no adjacent high intensity pixels (*i.e.*, $CN = 0$). End-point pixels ($CN = 1$) represent the last pixels of a

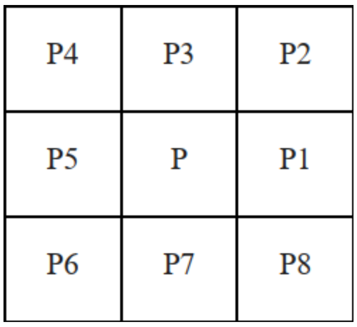


Figure 5: A 3-by-3 matrix whose central cell (P) is the examined object.

CN	Characteristic
0	Isolated pixel
1	End-point pixel
2	Pixel surrounded by 2 other pixels
3	Pixel surrounded by 3 other pixels
4	Pixel surrounded by 4 other pixels

Table 2: Value of CN and its corresponding characteristic.

continuous line. $CN = 2$ denotes the examined pixel is surrounded by 2 other pixels, signifying its role as a middle point contributing to the formation of a continuous line, a scenario that occurs most frequently. The pixels that are surrounded by 3 or 4 other pixels (*i.e.*, $CN = 3$ or $CN = 4$) are identified as junction points, where multiple lines intersect.

4 Skeletonization algorithms

Skeletonization stands out as a widely employed image processing technique, characterized by its ability to reduce the foreground regions in a binary image to a skeletal remnant. This remnant effectively retains the connectivity of the original region while discarding a significant portion of the original foreground pixels. The resulting skeleton represents the thinned representation of the object. Skeletonization algorithms achieve this by progressively eliminating border pixels in binary images, ensuring the preservation of connectivity among neighboring pixels and maintaining the shape of the original object. Skeletonization algorithms offer several advantages, including reduced file size and enhanced rendering speed. They also provide the flexibility to conceal specific elements, such as furniture, without entirely eliminating them from the final output. However, it is essential to note potential challenges, such as confusion or errors, that may arise if certain parts of the image are hidden while others remain visible, impacting the accuracy of the results obtained through this method. In the analysis of rice panicle architecture, skeletonization algorithms play a crucial role by thinning the binary image generated by a crack detection model. This thinned representation is subsequently employed for junction detection through clustering algorithms. In this Subsection, we will discuss two outstanding skeletonization algorithms from [18] and [19].

4.1 Hi-LASSIE

From only 20-30 online images in the wild without any user-defined shape or skeleton templates, *Hi-LASSIE* [18], a novel technique specialized for reconstructing 3D articulated shape from sparse image ensemble without using any 2D/3D annotations or templates, can automatically estimate the 3D skeleton, shape, camera viewpoints of a wild animal, which is an extremely challenging task. In discovering 3D skeleton, *Hi-LASSIE* initiates the process by extracting the 2D skeleton through a skeletonize/morphological thinning algorithm, employing the well-known Zhang-Suen thinning method [26]. This method involves eliminating all contour points from the image except those contributing to the skeleton of unitary thickness. Widely adopted across various applications, this thinning method has proven to be efficient not only in the domain of wild animal reconstruction but also in exploring the architecture of rice panicles.

4.2 Gradient-based optimization

The *gradient-based optimization* skeletonization algorithm [19], unveiled in 2023, takes a grayscale image as its starting point. In its process, the algorithm engages in iterative boundary-peeling, where these boundaries consist of straightforward points. Each of these points is distinguished using either the Euler characteristic or Boolean characteristics, offering flexibility based on the user's preferences. Two key parameters, namely β (Boltzmann temperature) and τ, are crucial in defining the behavior of this algorithm.

5 Dataset and evaluation metrics

In this paper, we employ images featuring 21 rice panicles sourced from diverse accessions of *O. Sativa*, *O. Glaberrima*, and *O. Barthii*, originating from Asia and Africa [1]. Each rice panicle is meticulously captured in an extended position, secured in place using tapes or metal pins against a white background. To assess the outcomes of junction detection, rice panicle junctions have been pre-labelled based on their specific roles (Figure 6). Illustrated in Table 3, the generating junctions, marking the initiation point where the main axis commences development and the termination point where it ceases growth, are delineated by yellow dots. Primary junctions and secondary junctions are denoted by white dots and blue dots, respectively.

Level of junction	Corresponding color
Generating	Yellow
Primary	White
Secondary	Blue

Table 3: Rice panicle junction classification.

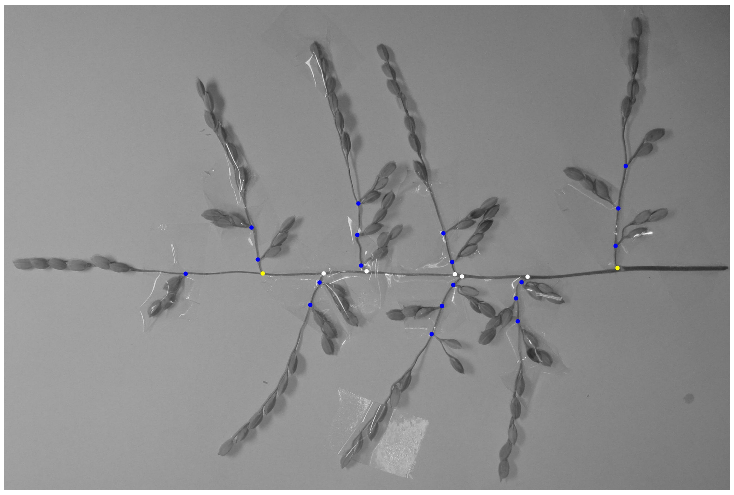


Figure 6: Labelled image of rice panicle.

We evaluate the accuracy of our proposed junction detection method using the F_1-$score$, which is the harmonic mean of precision and recall (Equation 2). Owing to the indirect and complex nature of employing various vision-based methods, evaluating the accuracy of the proposed method within the margin of 1 pixel error is impractical. As a result, we proceed to calculate the precision and recall as follows. Each predicted junction, represented by a single pixel, is considered as a true positive (tp) if it belongs to the labelled/true junction's proximity of 24 surrounding pixels. Specifically, we view each true junction as the central of a 5-by-5 matrix and the examined junction is correctly predicted if it falls within this matrix. False positives (fp) are the number of junctions that remain outside the target matrices, while false negatives (fn) represent the number of true junctions that are not encompassed by any predicted one.

$$F_1 = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{2}$$

where

$$\text{precision} = \frac{\text{tp}}{\text{tp + fp}}$$
$$\text{recall} = \frac{\text{tp}}{\text{tp + fn}}$$

III Experimental results

A recent Java-based application, *P-TRAP* [1], has shown promising results in analyzing the structure of rice panicles, including panicle structure analysis, spikelet/grain counting, and seed shape examination. However, this method lacks automation in identifying rice panicle junctions and may require prior user knowledge of the plants under examination. To address this limitation, we conducted extensive experiments to evaluate the performance of our approach in automatically detecting rice panicle junctions.

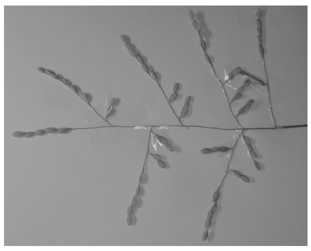
We conducted experiments on 21 rice panicles sourced from Asia and Africa. Initially, we utilize a novel crack detection model named RUC-Net [5] to extract binary images from provided rice panicles. Subsequently, we construct the skeleton of each rice panicle using two techniques from [18] and [19]. These skeletons are then subjected to implementation with six clustering algorithms [20–25] and two image-processing techniques to extract junctions. In this Section, we will discuss some issues related to *P-TRAP* application [1], and present the experimental results while selecting the most appropriate algorithms for exploring rice panicle structure.

1 P-TRAP in junction detection

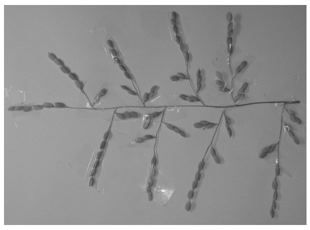
Proposed in [1], *P-TRAP* is a Java-based application designed for plant panicle architecture analysis, offering three primary functionalities: panicle structure analysis, spikelet/grain counting, and seed shape analysis. However, the process of identifying junctions in a rice panicle using *P-TRAP* involves several manual steps. Firstly, users must accurately crop the rice flower image from the input image to ensure that all junctions are captured for subsequent processing. Secondly, users need to manually identify and assign the level for several junction, including determining starting and ending generating points. Finally, users must connect the identified intersection points and ending points in the correct order to calculate the distance and coordinates of these points. Overall, these manual steps are time-consuming and prone to errors during implementation. Therefore, in this paper, we propose utilizing clustering algorithms as an alternative approach to automate the process of junction detection.

2 Crack detection outcomes

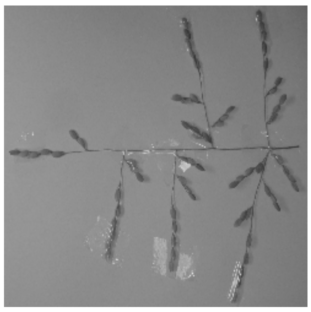
In the first step, we employ the crack detection model RUC-Net [5] to generate binary images of rice panicles. The process is illustrated in Figure 7, displaying several notable outcomes. The original rice panicle images, varying in dimensions, are initially resized to a standardized dimension of 256 x 256, ensuring consistency across the dataset for subsequent processing. While extracted binary images generally preserve the original shapes of rice panicles, there



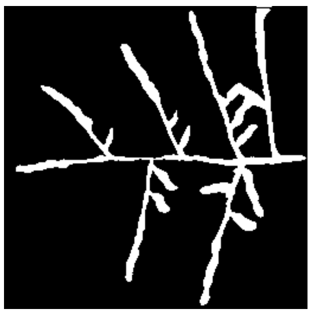
(a)



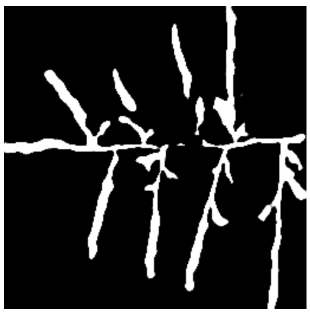
(b)



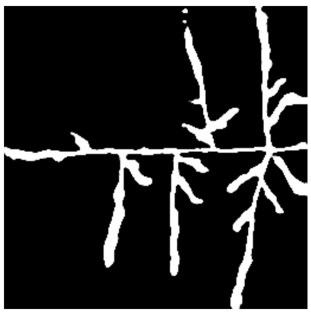
(c)



(d)



(e)



(f)

Figure 7: RUC-Net outcomes.

(a, b, c) Original rice panicles and (d, e, f) their corresponding binary images.

are some notable cases. Overlapping branches, as observed in Subfigures 7a and 7d, have the potential to lead to undesired outcomes, specifically the creation of false junctions. This can adversely impact the accuracy of detecting high-order junctions. However, this challenged can be addressed by further spreading the panicles. Another challenge arises when thin branches are secured in place by reflective tapes, making RUC-Net susceptible to misdetecting the target structure. This can result in a broken, discontinuous pattern in several branches, as illustrated in Subfigures 7b and 7e. Finally, an ideal result is depicted in Subfigures 7c and 7f, where the original shape is preserved as a continuous structure without any occurrence of overlap.

3 Optimal Clustering algorithms selection

We implement 6 clustering algorithms [20–25] along with two image-processing techniques to evaluate their efficiency in detecting rice panicle junctions. *Crossing number* method stands out as an effective and direct approach for identifying intersections in structures with unitary thickness. Additionally, *DBSCAN* is well-suited for our application, as it can detect points with a defined number of neighbors within a pre-defined proximity. The MATLAB function *BWMORPH* also proves to be reliable, as it can easily identify branching structures in a given skeleton. Superb performance of these techniques are depicted in Figure 8. Given the same skeletons, *Crossing number*, *DBSCAN* and *BWMORPH* can efficiently detect the target inter-

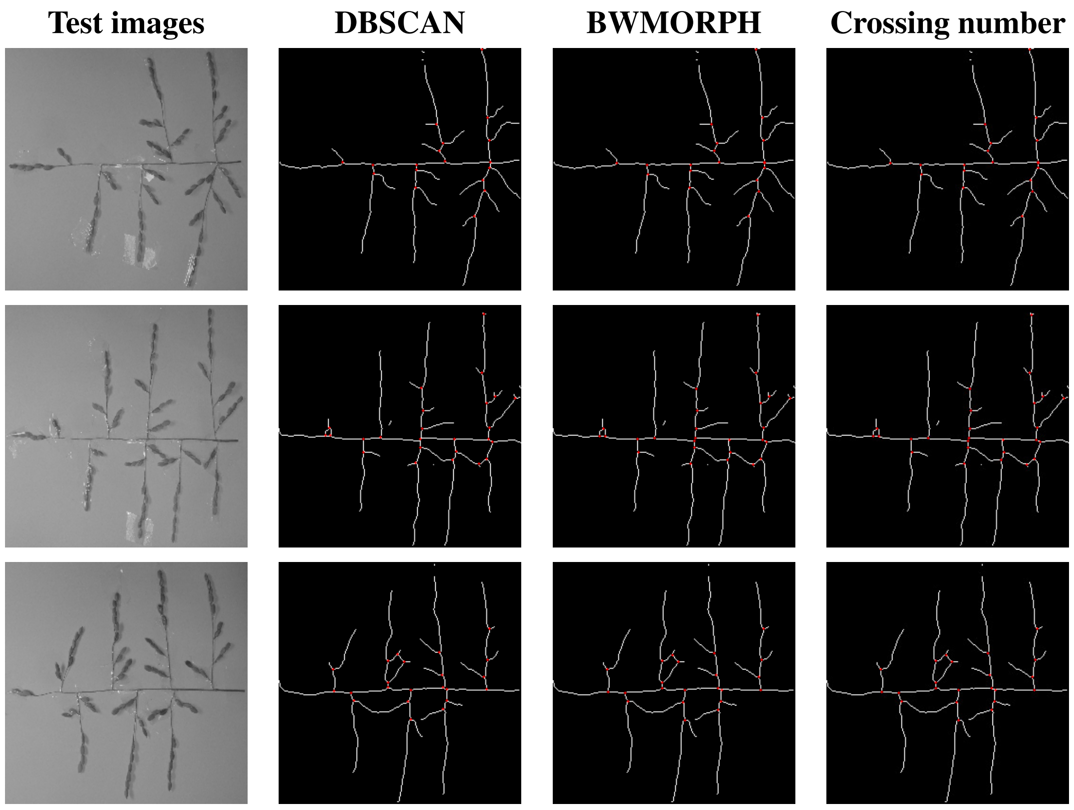


Figure 8: Qualitative results of clustering algorithms.

Method	F_1	Precision	Recall
Crossing number	0.6946	0.7633	0.6491
BWMORPH	0.6818	0.7022	0.6862
DBSCAN	0.5930	0.6447	0.5085

Table 4: Performance comparison of clustering algorithms.

section points (*i.e.*, junctions). The predicted junctions are marked as red dots. While satisfactory results are achieved using these three techniques, the average F_1-$score$ of each method in finding junctions in rice panicle main axis across the dataset is provided in Table 4. *Crossing number* and *BWMORPH* exhibit similar accuracy, reaching nearly 70%, while *DBSCAN* achieves an average F_1-$score$ of 60%. The results highlight the advantage of *Crossing number* and *BWMORPH* in examining each object individually, as opposed to averaging the junction positions from objects in a subset identified by *DBSCAN*.

Other participating clustering algorithms have different approaches in identifying clusters that are tested unsuitable for our application, leading to undesired sets of junctions. The junction detection performances of *DP* [21], *FINCH* [24] and *HIAC* [23] are illustrated in Figure 9, where red dots represent the predicted junctions. Obtained results lead us to employ the three aforementioned techniques, including *Crossing number*, *BWMORPH*, and *DBSCAN* for the task

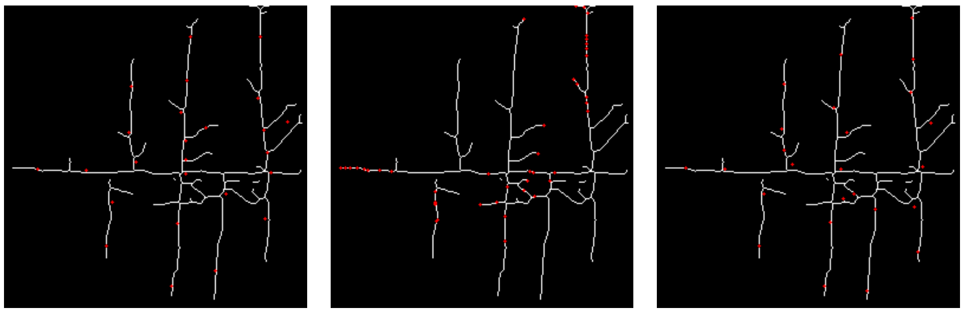


Figure 9: Sub-optimal junction detection performance.

of identifying rice panicle junctions.

4 Optimal Skeletonization algorithm selection

We proceed to conduct a performance comparison between two discussed skeletonization algorithms, the *Gradient-based Optimization* skeletonization algorithm [19] and the *Zhang-Suen* thinning method [26], to see which one suits our approach better. The *Crossing number* technique is employed for junction detection in both skeleton datasets produced by the two skeletonization algorithms. Experimental results, conducted on our dataset of 21 rice panicle images, are recorded in Table 5. We successively estimate F_1-$score$, precision, and recall for each rice panicle along its main axis, arranging them in descending order of F_1-$score$. As depicted in the final row of Table 5, the *Zhang-Suen*-based approach demonstrated a higher average accuracy, reaching nearly 70%, compared to the *Gradient-based Optimization* method. Higher precision and recall in junction detection are also achieved when we employ the *Zhang-Suen* thinning method to extract rice panicle skeletons. The subtle superiority is illustrated in Figure 10, where slightly more reasonable junction positions are obtained through *Zhang-Suen* rice panicle skeleton.

Further accuracy estimation is also conducted for higher-order junctions (*i.e.*, secondary junctions and beyond). However, considerably lower results are obtained in detecting higher-order junctions (Table 6). Specifically, only an average of 32.59% true junctions are correctly identified on *Zhang-Suen* skeletons. The undesired results can be attributed to various factors, including overlapping branches, the discontinuous pattern produced by the crack detection model, and the nature of the skeletonizing method. The iterative boundary-peeling process of the skeletonizing method may lead to poor junction detection, especially when the original structure has an even-pixel width. As a result, given its overall higher accuracy, we opt to utilize the *Zhang-Suen* thinning method for exploring the architecture of the rice panicle.

Gradient-based Optimization			Zhang-Suen		
F_1	Precision	Recall	F_1	Precision	Recall
0.9231	1.0000	0.8571	1.0000	1.0000	1.0000
0.9091	1.0000	0.8333	1.0000	1.0000	1.0000
0.8889	1.0000	0.8000	0.8889	1.0000	0.8000
0.8333	0.8333	0.8333	0.8333	0.8333	0.8333
0.8000	0.8000	0.8000	0.8000	1.0000	0.6667
0.7500	1.0000	0.6000	0.8000	0.8000	0.8000
0.7273	0.8000	0.6667	0.8000	1.0000	0.6667
0.7143	0.7143	0.7143	0.8000	0.8000	0.8000
0.6667	0.6667	0.6667	0.7692	0.8333	0.7143
0.6667	0.8000	0.5714	0.7273	1.0000	0.5714
0.6000	0.6000	0.6000	0.7273	0.8000	0.6667
0.6000	0.7500	0.5000	0.6667	0.7143	0.6250
0.6000	0.6000	0.6000	0.6667	1.0000	0.5000
0.5714	0.5714	0.5714	0.6667	0.7143	0.6250
0.5714	0.6667	0.5000	0.6000	0.6000	0.6000
0.5000	0.6667	0.4000	0.6000	0.6000	0.6000
0.4444	0.6667	0.3333	0.5455	0.6000	0.5000
0.4000	0.4000	0.4000	0.5000	0.5000	0.5000
0.4000	0.4000	0.4000	0.4615	0.5000	0.4286
0.3636	0.4000	0.3333	0.4000	0.4000	0.4000
0.3077	0.4000	0.2500	0.3333	0.3333	0.3333
0.6304	0.7017	0.5824	0.6946	0.7633	0.6491

Table 5: Performance Comparison between Gradient-Based Optimization and Zhang-Suen in descending order of F_1-$score$.

Table 6. Performance comparison in higher order junctions.

Method	F_1	Precision	Recall
Gradient	0.3142	0.2884	0.3511
Zhang-Suen	0.3259	0.3149	0.3432

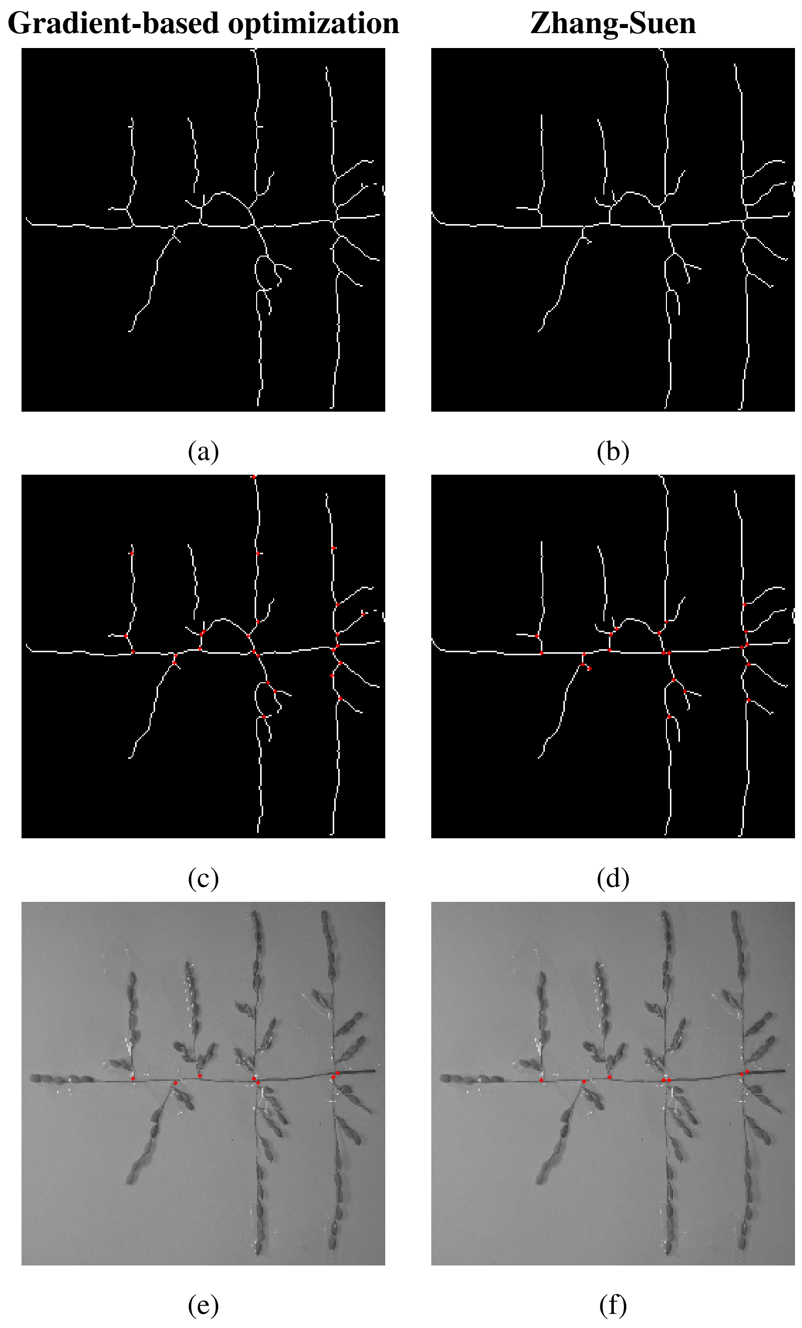


Figure 10: Qualitative results of participating Skeletonization algorithms.
(a, b) Original rice panicle skeletons. (c, d) Corresponding junctions detection.
(e, f) Junctions detected on rice panicle main axis.

IV Conclusion

In this paper, we present a novel approach for the architecture analysis of rice panicles using Deep Learning. Our proposed method offers advantages in automatically detecting rice panicle junctions, addressing a crucial need in plant phenotyping. The examination of rice inflorescence architecture is essential for understanding the diversity of rice panicle resources and optimizing traits relevant to rice breeding and yield improvement. Traditionally, such analyses have relied on manual labor, which can be time-consuming and prone to inaccuracies. To overcome these limitations, we leverage recent advancements in crack detection algorithms, which have shown promising results in automatically identifying features on concrete surfaces.

Experimental results demonstrate the effectiveness of our approach in automatically detecting rice panicle main axis junctions across a dataset of diverse rice accessions from Asia and Africa. The undesired results obtained when detecting high order junctions also call for further work. By automating the process of junction detection, our method offers significant improvements in efficiency and accuracy compared to traditional manual methods.

In conclusion, our proposed vision-based approach provides a valuable tool for researchers and breeders to analyze the architecture of rice panicles efficiently and accurately. Future work may involve further refinement of the algorithm for the task of high order junction detection and its application to other plant species, advancing our understanding of plant morphology and supporting efforts in crop improvement and agricultural sustainability.

References

[1] F. AL-Tam, H. Adam, A. d. Anjos, M. Lorieux, P. Larmande, A. Ghesquière, S. Jouannic, and H. R. Shahbazkia, “P-trap: a panicle trait phenotyping tool,” *BMC Plant Biology*, vol. 13, no. 1, p. 122, Aug 2013.

[2] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[3] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *CoRR*, vol. abs/1505.04597, 2015.

[4] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, “Deepcrack: Learning hierarchical convolutional features for crack detection,” *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1498–1512, 2019.

[5] G. Yu, J. Dong, Y. Wang, and X. Zhou, “Ruc-net: A residual-unet-based convolutional neural network for pixel-level pavement crack segmentation,” *Sensors*, vol. 23, no. 1, 2023.

[6] Q. Zhu and Q. P. Ha, “A bidirectional self-rectifying network with bayesian modeling for vision-based crack detection,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 3, pp. 3017–3028, 2023.

[7] P. Shi, F. Zhu, Y. Xin, and S. Shao, “U2cracknet: a deeper architecture with two-level nested u-structure for pavement crack detection,” *Structural Health Monitoring*, vol. 22, no. 4, pp. 2910–2921, 2023.

[8] T. H. Dinh, V. T. T. Anh, T. Nguyen, C. Hieu Le, N. L. Trung, N. D. Duc, and C.-T. Lin, “Toward vision-based concrete crack detection: Automatic simulation of real-world cracks,” *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–15, 2023.

[9] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bian, L. Zhang, X. Bai, and J. Tang, “Richer convolutional features for edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, p. 1939–1946, Aug. 2019.

[10] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[11] C. V. Dung and L. D. Anh, “Autonomous concrete crack detection using deep fully convolutional neural network,” *Automation in Construction*, vol. 99, pp. 52–58, 2019.

[12] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, “Feature pyramid and hierarchical boosting network for pavement crack detection,” 2019.

[13] K. Li, B. Wang, Y. Tian, and Z. Qi, “Fast and accurate road crack detection based on adaptive cost-sensitive loss function,” *IEEE Transactions on Cybernetics*, vol. 53, no. 2, pp. 1051–1062, Feb 2023.

[14] Y. Fei, K. C. P. Wang, A. Zhang, C. Chen, J. Q. Li, Y. Liu, G. Yang, and B. Li, “Pixel-level cracking detection on 3d asphalt pavement images through deep-learning- based cracknet-v,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 273–284, 2020.

[15] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, “Basnet: Boundary-aware salient object detection,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7471–7481.

[16] F. Guo, Y. Qian, J. Liu, and H. Yu, “Pavement crack detection based on transformer network,” *Automation in Construction*, vol. 145, p. 104646, 2023.

[17] H. Chen and H. Lin, “An effective hybrid atrous convolutional network for pixel-level crack detection,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[18] C.-H. Yao, W.-C. Hung, Y. Li, M. Rubinstein, M.-H. Yang, and V. Jampani, “Hi-lassie: High-fidelity articulated shape and skeleton discovery from sparse image ensemble,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4853–4862.

[19] M. J. Menten, J. C. Paetzold, V. A. Zimmer, S. Shit, I. Ezhov, R. Holland, M. Probst, J. A. Schnabel, and D. Rueckert, “A skeletonization algorithm for gradient-based optimization,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 21 394–21 403.

[20] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD’96. AAAI Press, 1996, p. 226–231.

[21] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[22] M. Abbas, A. El-Zoghabi, and A. Shoukry, “Denmune: Density peak based clustering using mutual nearest neighbors,” *Pattern Recognition*, vol. 109, p. 107589, 2021.

[23] Q. Li, S. Wang, X. Zeng, B. Zhao, and Y. Dang, “How to improve the accuracy of clustering algorithms,” *Information Sciences*, vol. 627, pp. 52–70, 2023.

[24] S. Sarfraz, V. Sharma, and R. Stiefelhagen, “Efficient parameter-free clustering using first neighbor relations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[25] J. Hou, H. Yuan, and M. Pelillo, “Towards parameter-free clustering for real-world data,” *Pattern Recognition*, vol. 134, p. 109062, 2023.

[26] T. Y. Zhang and C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” *Commun. ACM*, vol. 27, no. 3, p. 236–239, mar 1984.