Journal of Computer Science

Review Article

Comparative Analysis of Classical Feature Detection Methods for UAV-Based Tomato Detection

[1]Muhammad Sarwar Jahan Morshed, [2]Md. Nasim Adnan and [2]Md. Rafiqul Islam

[1]Department of Computer Science and Engineering, Jashore University of Science and Technology, Jashore, Bangladesh
[2]Department of Computer Science, American International University-Bangladesh, Dhaka, Bangladesh

Article history
Received: 15-03-2025
Revised: 09-05-2025
Accepted: 29-05-2025

Corresponding Author:
Muhammad Sarwar Jahan Morshed
Department of Computer Science and Engineering, Jashore University of Science and Technology, Jashore, Bangladesh
Email: p220102.cse@student.just.edu.bd

Abstract: Remote sensing technologies, especially Unmanned Aerial Vehicles (UAVs), become crucial for Precision Agriculture (PA) to perform different tasks such as crop detection, yield prediction, leaf disease diagnosis, weed detection, and harvest forecasting to ensure higher productivity. Therefore, using various image analytics methods feature detection from the UAV-captured images plays a vital role for conducting these PA tasks. To enhance the effectiveness of the UAV-based image analysis, this study investigates the performance of various classical feature detection algorithms on the UAV-captured images of tomato fields. This study also identifies the standard benchmarks of the evaluation metric used in the feature detection methods. The evaluation considers challenging conditions such as rotation, illumination variation, and scaling. Results show that Oriented FAST and Rotated BRIEF (ORB) and Speeded Up Robust Features (SURF) among the classical methods demonstrated better performance under all these environmental conditions. However, considering the limitations in existing feature detection techniques this study also suggests that integrating classical feature detection with deep learning approaches could significantly improve real-time feature detection efficiency.

Keywords: Precision Agriculture, UAV-Captured Images, Computer Vision, Object Detection, Feature Detection, Image Overlapping

Introduction

Farming is a complex sector that considers factors like soil, crop types, weeds, temperatures, etc. to maximize the yield. It requires planting, watching the weather, and applying the required amounts of water, nutrients, and pesticides, manually inspecting fields for sights of stress or pest infestation. In general, farmers are unsure about which type of fertilizer to use (organic or conventional) to meet the needs of their land. Soil degradation caused by insufficient and unbalanced fertilization results in nutrient mining and the emergence of second-generation nutrient management issues. According to a study by the Associated Chambers of Commerce and Industry of India crop yield fluctuates so often that these conventional (ASSOCHAM), annual crop losses due to pests amount to Rs. 50,000 cores (Swati, 2014). Furthermore, resources and time.

To boost quality crop productivity, remote sensing techniques such as satellite and the UAV offer an effective way to serve small to large-scale operations and assess crop health. Such remote sensing techniques can be used to pinpoint areas of crop stress to determine when, where, and how much water, fertilizer, and pesticides are needed to produce a healthy crop (Fawakherji *et al.*, 2021). PA analytics and relevant research bring efficiency to agriculture by producing healthier crops. This helps to minimize losses and maximize production to boost profits. UAV-based photogrammetric is becoming a popular field, especially in PA as it has the following advantages over traditional ways of inspecting fields for sights of stress (Cheng *et al.*, 2010; Pacot and Marcos, 2018; Rokhmana, 2015).

The adoption of Unmanned Aerial Vehicles (UAVs) for precision agriculture systems offers numerous compelling advantages. First, there are significant cost savings from reduced expenses in constructing the platform infrastructure. Operationally, UAVs provide adaptable and swift responsiveness to changing field conditions. From a data perspective, they offer the capability to acquire high-resolution images and accurate positioning information essential for precision farming. Technologically, UAVs enable the implementation of missions that involve high risks and advanced technology without endangering human operators. Regulatory advantages include the fact that most

countries do not require airspace control permits for low-altitude flights, as seen in nations like China and Bangladesh.

Due to these benefits, building PA systems using the UAV has become a hot topic worldwide. However, utilizing the UAV-captured raw images in developing PA systems does not always provide fruitful results. It requires preprocessing of images before using them as input in the PA system. However, in the last couple of years, several initiatives have been presented. These initiatives can be categorized into two groups: (i) classical methods and (ii) Convolutional Neural Network (CNN)-based methods. CNN-based methods are more suitable when huge volume of data is required to be processed to train the analytical model. On the other hand, in situations when computational resources are restricted, datasets are tiny, or there are particular applications that need accurate feature matching, classical feature detection algorithms like Scale-invariant feature transform (SIFT), SURF, etc. may perform better than CNN-based methods. Classical methods offer explicit feature descriptors that are simpler to understand, demand less computing power, and do not require a lot of training data. Classical methods provide high precision and reliable performance under various conditions, making them suitable for specific applications where CNNs might not be optimal.

In case of feature detection algorithms, the main challenge lies in the Keypoint (it is notable that Key Points, Keypoints and Feature Points can be used interchangeably) identification or feature detection in the UAV-captured image. Keypoints provide a substantial quantity of crucial information in an image. Since it may reduce misalignment faults in the final stitched image, accurate extraction of these Keypoints - is necessary for image stitching (Lindeberg, 2012; Bay *et al.*, 2006; Lavin and Gray, 2015; Alahi *et al.*, 2012; Calonder *et al.*, 2010; Leutenegger *et al.*, 2011; Alcantarilla *et al.*, 2012; Rublee *et al.*, 2011). However, a comprehensive comparative study on feature detection is lacking in the literature. To address this gap, we conducted an exploratory study of existing state-of-the-art feature detection methods. This paper outlines several notable contributions, including a meticulous comparison of contemporary, cutting-edge feature detection methods, specifically those employed for detecting features in images captured by UAVs. Furthermore, this paper concludes by delineating the limitations observed in some existing feature detection algorithms, emphasizing their challenges in handling image overlapping analysis in UAV-captured imagery.

This paper aims at researchers and academics, as well as both non-commercial and commercial entities with a vested interest in exploring, developing, or choosing feature detection methods for the PA and the UAV-based technology, focusing on comparative analyses of classical feature detection methods under varied agricultural conditions.

The remainder of the paper is organized as follows. Section *Research Methodology* describes the research methodology we followed to conduct this review study. Section *Feature Detection Approach* illustrates the classical feature detection algorithms. Section *Classical Approaches vs. Deep Learning Approaches* highlights the influence of classical feature detection methods and the deep learning-based methods. Section *Data Collection and Preprocessing* details the data collection and preprocessing approach used in this investigation. Section *Evaluation Metrics* illustrates the evaluation metrics to compare the performance of feature detection methods. Section *Experimental Data* presents the experimental framework. Results and discussions of this review study are explained in Section *Results and Discussion*. Finally, Section *Conclusion* states the concluding remarks and future research direction of feature detection methods for PA.

Methodology

This review study has been carried out by the steps shown in Figure 1.

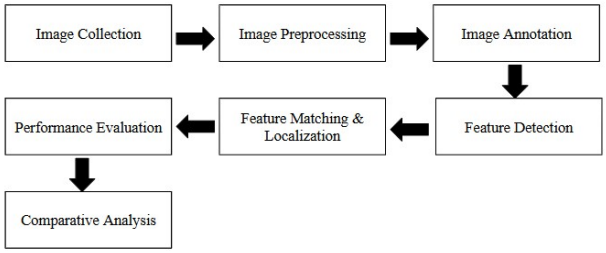


Fig. 1: Steps of the methodology followed in this study

Image Collection: Using a UAV to gather high-resolution aerial images of a target region, such as a tomato field or an agricultural field, is the first stage of this study. A UAV fitted with an RGB camera was used. Image capturing follows predetermined flight routes to guarantee constant and thorough coverage of the region. Neutral density (ND) filters are also used for controlling lightning conditions. Raw aerial images with crucial metadata, such as timestamp, exposure, resolution, focal length, etc., are the result of this stage and are necessary for further processing and analysis.

Image Preprocessing: The main goal of the *Image Preprocessing* is to improve image quality for precise annotation and analysis. Image preprocessing is performed in several steps including cropping, resizing, or tiling of the images to make them uniform and consistent. Other improvement tasks such as contrast correction and noise reduction are also performed to increase visual clarity. Image normalization or grayscale conversion processes are also applied to make all images in a standard and consistent form to make them ready for next steps of the processing pipelines.

Image Annotation: In this stage, specific Keypoints or areas of interests within the preprocessed images

(such as crops/fruits, leaves, stems, crop disease, weeds, landmarks, etc.) are labeled using any suitable annotation tool. Common image annotation tools are Roboflow, LabelImg, or CVAT. In this review study, Roboflow is used for image annotation, as it is user friendly and the annotated image can be downloaded in different formats such as Pascal VOC, YOLO TXT, TF Record and COCO JSON. In our experiment, we converted our annotated images as COCO JSON.

Feature Detection: In this stage, classical feature detection methods such as SIFT, SURF, ORB, Binary Robust Invariant Scalable Keypoints (BRISK), BRIEF, FREAK, KAZE, and AKAZE have been applied to the processed image dataset to identify recognizable and repeating keypoints in all images. Tools such as Python and OpenCV are used to implement this step. This process results in a collection of keypoints for each image aligned with the corresponding descriptor. These collection of keypoints are ultimately used for feature matching and localization.

Feature Matching and Localization: This stage is used to understand the spatial linkages and to facilitate precise localizations for identifying similarity between features across the images. Matching techniques such as Brute Force matcher or FLANN are frequently used in conjunction with RANSAC to remove outliers and to increase the match reliability. This step results in matching keypoint pairs and calculated transformation matrices (such as affine or homography) that characterize the spatial alignment of images.

Performance Evaluation: The goal of the sixth phase, performance evaluation, is to quantitatively evaluate the efficacy of feature detection methods. This involves computing a number of metrics, including precision, recall, matching robustness, false positive rate, detection rate, repeatability rate, mean localization error (MLE), and matching robustness. These measurements offer a detailed understanding of the precision, consistency, and dependability of the algorithms. Usually OpenCV, custom Python scripts, or Pycocotools are used for evaluation, especially compared to object detection tasks, to ensure the system satisfies the performance requirements for the intended use.

Comparative Analysis: The last stage of this review study is to compare the performance of the feature detection methods applied to our collected UAV-captured tomato image dataset. In this step, all feature detection methods are simulated on different sets of datasets based on different environmental scenarios. In this study, we consider three factors, such as rotation, scaling, and illumination. Performance are evaluated 8 evaluation metrics.

Classical Feature Detection Approaches

For feature detection, several feature detection algorithms have been introduced such as SURF (Bay *et al.*, 2006), SIFT (Lindeberg, 2012), and BRISK (Leutenegger *et al.*, 2011), and Maximally Stable Extremal Region (MSER) (Cen *et al.*, 2019). The feature detection algorithms that are evaluated in this study are summarized in the following sections.

SIFT

SIFT is a computer vision method that finds and describes distinctive features in images (Lindeberg, 2012). It works by detecting key points that are invariant to changes in scale, rotation, and lighting conditions. These key points are then described by their local image gradients, creating unique representations that can be matched across different images for tasks such as object recognition and image stitching. SIFT is widely used as a result of its robustness and effectiveness in various applications.

SIFT performs in several steps: *Identify keypoints* at multiple scales by detecting local extrema; *Keypoint Localization*, which refines the keypoint positions and scales using a 3D quadratic function; *Orientation Assignment*, which assigns a dominant orientation to each keypoint based on gradient magnitudes and orientations; *Descriptor Generation*, which creates robust descriptors by capturing the gradient distribution around each keypoint; and *Descriptor Matching*, which compares keypoint descriptors using distance metrics like Euclidean distance for tasks such as image alignment and object recognition.

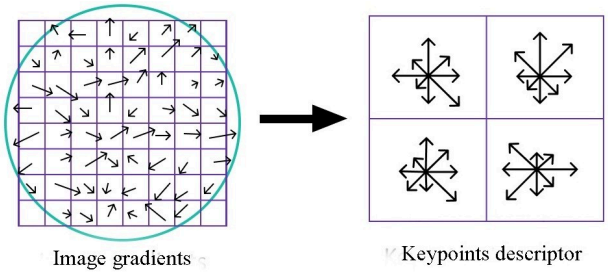


Fig. 2: Generation of descriptor array from a sample set of gradients (Lowe, 2004)

The left portion of Figure 2 shows the image gradients produced in the orientation assignment. During the orientation step, the grids are partitioned into four segments. Later, all segments of the gradients were merged on the basis of the individual direction of these segments. The right part of the image shows the descriptors measured using the gradients (Lowe, 2004). Two parameters such as the peak threshold and the edge threshold control the SIFT (Lindeberg, 2012) detector. The peak threshold eliminates too-small (in absolute value) peaks from the DoG (i.e. Difference of Gaussian) scale space. For increasing the peak threshold, fewer features are obtained. On the other hand, the edge threshold removes DoG scale space peaks with very small curvature (these peaks generate poorly localized frames). By increasing the edge threshold, more features are obtained.

SURF

Speeded Up Robust Features (SURF) is an algorithm for detecting and describing local features in images, designed to be faster than SIFT. This algorithm detects keypoints using the Hessian matrix for quick detection. It employs box filters to efficiently handle different scales, approximating Gaussian smoothing. For rotation invariance, it determines the dominant orientation of each keypoint using Haar wavelet responses. It then generates robust descriptors by computing Haar wavelet responses in subregions around each keypoint. Finally, it matches keypoints between images by comparing their descriptors using distance metrics as the Euclidean distance.

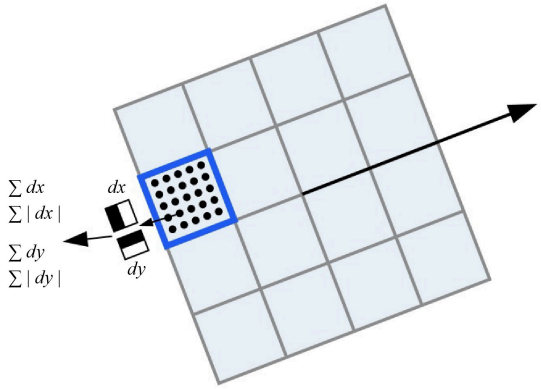


Fig. 3: Wavelet responses over the sub-region (Bay *et al.*, 2006)

As illustrated in Figure 3, *dx*, and *dy* represent pixels in each subregion. The absolute values of *dx* and *dy*, respectively, are summarized at the same time to reflect the intensity change of the descriptors. A four-dimensional vector feature *V* serves as a representation of each sub-region.

BRISK

Autonomous Systems Lab at ETH Zurich, Switzerland introduced Binary Robust Invariant Scalable Keypoints (BRISK) which is a point-feature detector and descriptor (Leutenegger *et al.*, 2011; Bojanic *et al.*, 2019). BRISK can achieve minimum processing complexity by deploying a special scale-space FAST-based detector (Lavin and Gray, 2015; Tyagi, 2019). It uses binary descriptors for feature description, which makes it highly efficient in terms of memory usage and computational speed. This is particularly useful for real-time applications and large-scale image processing. The steps of the BRISK (Leutenegger *et al.*, 2011) algorithm are described in the following.

BRISK identifies the interest points or keypoints in an image. These keypoints are distinct locations in the image that can be reliably found in different images of the same scene, despite changes in viewpoint, lighting, or occlusion. BRISK generates a bit-string description through intensity comparisons collected by targeted sampling within each keypoint neighborhood (Cen *et al.*, 2019). BRISK constructs a scale-space pyramid as shown in Figure 4, which involves creating a series of images at different levels of detail (scales). This allows BRISK to detect keypoints at multiple scales, making the algorithm robust to changes in object size or distance from the camera. Once the keypoints are identified, BRISK computes a binary descriptor for each keypoint. Unlike SIFT and SURF, which use floating-point descriptors, BRISK uses binary strings to represent the local image patch around each keypoint. This binary representation makes BRISK more memory-efficient and faster to compute.

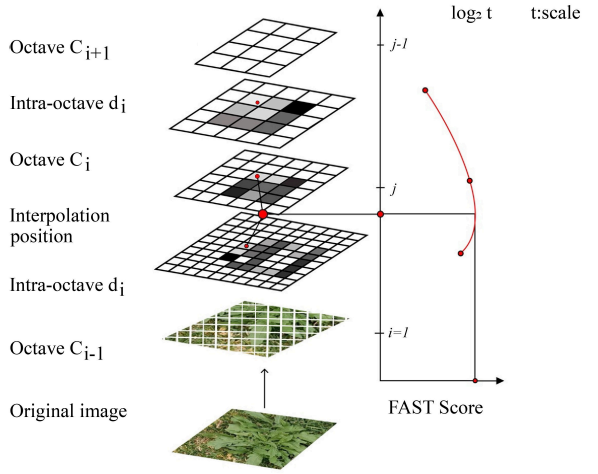


Fig. 4: Scale pyramid space with n number of octaves and n number of intra-octaves for keypoints detection at i represents the level

BRISK aims to be invariant to changes in rotation and scale, meaning that the same keypoint should be detected even if the object is rotated or scaled. This is achieved by employing a scale-invariant detector and descriptor that adapt to the local image structure. BRISK is designed to be restrained to noise, occlusion, and other image distortions. It achieves robustness through a combination of its detection and description methods, which are designed to handle various challenging conditions.

ORB

The Oriented FAST (Huang *et al.*, 2018) and Rotated BRIEF (Calonder *et al.*, 2010) (ORB) algorithm is presented by enhancing the integration of the FAST (Lavin and Gray, 2015) and BRIEF algorithms (Calonder *et al.*, 2010; Huang *et al.*, 2018). This makes ORB a feature detection as well as a feature description algorithm. The main aim of ORB is resource conservation. ORB computes oriented BRIEF (Calonder *et al.*, 2010) characteristics and adds a quick and accurate orientation component to FAST (Lavin and Gray, 2015). However, ORB uses the centroid approach to determine

FAST's orientation (Lavin and Gray, 2015). A de-correlation method for BRIEF (Calonder *et al.*, 2010) features under Rotation Invariance is also suggested by the authors of ORB, which might enhance performance in nearest-neighbor applications (Tian, 2013).

ORB performs image-matching in three steps: Feature Point Extraction, Orientation Assignment, Generating Feature Point Descriptors, and Feature Point Matching. In Feature Point Extraction, ORB algorithm uses the improved FAST (Lavin and Gray, 2015) algorithm to detect feature points. In this method, a pixel is more likely to be a corner point if it differs greatly from its neighboring pixels. The process begins by selecting a pixel in the image (as it presented by *p* in Fig. 5) and assuming its brightness. A brightness threshold is set, and 16 surrounding pixels are chosen within a small radius. The brightness of these surrounding pixels is compared to the center pixel. If a specific number of consecutive surrounding pixels are brighter or darker than the center by a defined threshold, the center pixel is considered a feature point. Initially, only four specific pixels are tested for efficiency. If at least three of these pass the threshold, all 16 surrounding pixels are evaluated to confirm if the pixel is an interest point. Every pixel in the image undergoes this iterative process. This method is optimized by starting with a subset of pixels to reduce unnecessary calculations and improve speed.

After this, ORB detects the keypoints in two sub-steps such as Feature Point Detection (ORB uses the FAST (Lavin and Gray, 2015) algorithm to quickly detect keypoints in an image.) and FAST (Lavin and Gray, 2015) Corner Point Computation (it refers only to comparing differences in brightness between pixels. Therefore, the number of corner points becomes large and uncertain).

In the step of Orientation Assignment, ORB assigns an orientation to each keypoint by computing the intensity centroid around the keypoint. This makes the features rotation-invariant. Feature Point Descriptors are generated in the third step. After extracting oriented FAST (Lavin and Gray, 2015) keypoints, the ORB algorithm uses an improved version of the BRIEF (Calonder *et al.*, 2010) algorithm to compute a descriptor for each keypoint. BRIEF (Calonder *et al.*, 2010) is characterized by a binary vector descriptor, with its vector composed of multiple 0 and 1. Finally, ORB performs *Descriptor Matching* steps. Binary descriptors are matched between images using the *Hamming distance*, which counts the number of differing bits.

FREAK

The Human visual system, specifically the retina, serves as the source of inspiration for FREAK (Fast Retina Keypoint) (Alahi *et al.*, 2012). First of all, a cascade of binary strings is created by effectively comparing picture intensities over a retinal sample pattern. FREAK (Alahi *et al.*, 2012) is a dual descriptor that is computed based on brightness comparison experiments conducted over a significant amount of samples on an interesting point (Alahi *et al.*, 2012). FREAK algorithm performs the feature detection in a number of steps.

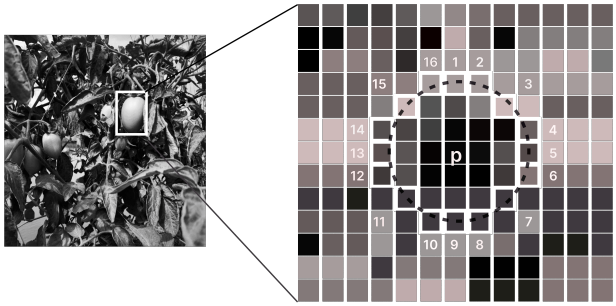


Fig. 5: Feature Detection in Image Patch (Lavin and Gray, 2015)

The first step of the FREAK algorithm involves generating a sampling pattern. At this stage, a Gaussian kernel is used to smooth *N* points in a sample around a particular keypoint. To simulate human retina behavior that is identical to the behavior of the human visual system, the kernel size is varied depending on the position of the sampling point. The centroids of the receptive fields are thus illustrated by the FREAK (Alahi *et al.*, 2012) descriptor sampling sites.

Creating the descriptor is the next step in the FREAK algorithm (Alahi *et al.*, 2012). This descriptor is built via intensity comparisons between various pairs of smoothed sample locations, such as the centers of receptive fields. Later in the Orientation Normalization step (Alahi *et al.*, 2012), descriptors are evaluated using several selected sampling pairs that are symmetrically arranged around the sampling pattern's center. The total of the differences between two component's identical elements can be used to compute their Manhattan distance from one another. This distance presents the distribution of the intensity difference of each keypoints. This orientation makes FREAK rotationinvariant.

Freak utilizes binary descriptors that are compact and convenient for the application with a constraint of low memory and bandwidth. It performs faster feature detection compared to other conventional descriptors such as SIFT or SURF.

BRIEF

BRIEF, or Binary Robust Independent Elementary Features, was introduced by Calonder (Calonder *et al.*, 2010). BRIEF uses a sampling pattern with 128, 256, or 512 comparisons, with sample points randomly selected from an isotropic Gaussian distribution centered at the feature position (equating to 128, 256, or 512 bits). Researches (Alahi *et al.*, 2012) influenced by BRIEF (Calonder *et al.*, 2010) demonstrates that image patches can be efficiently identified based on a sizable number of

pairwise intensity comparisons. This serves as the basis for BRIEF (Calonder *et al.*, 2010).

BRIEF uses straightforward binary comparisons between pixels in an image patch (Calonder *et al.*, 2010). We know that image patch refers to the surroundings of a pixel. The dimension of a patch is a square of pixels' height and width. The first pixel in each random pair is chosen at random from a Gaussian distribution with a Stranded Deviation or spread that is centered on the keypoint (Figure 6). Each comparison is a binary test (0 or 1), resulting in a compact and efficient binary string. A Gaussian distribution centered on the first pixel with a standard deviation or spread of sigma by two is used to generate the second random pixel in the pair. A value of 1 is attributed to the relevant bit if the brightness of the first pixel exceeds that of the second one. Conversely, a value of 0 is assigned.

In many ways, the performance of BRIEF (Calonder *et al.*, 2010) is comparable to SIFT (Lindeberg, 2012), including its resilience to lighting, blur, and perspective distortion. However, it is highly vulnerable to in-plane rotation.

KAZE

KAZE (Accelerated Segment Test with K-means and Enhanced Descriptors) (Alcantarilla *et al.*, 2012) algorithm is a complex feature detection and description algorithm used in computer vision. Initial step of KAZE is Scale Space Construction. In order to detect features at different scales, KAZE (Alcantarilla *et al.*, 2012) constructs a nonlinear scale space. This scale space is formed by convolving the image with Gaussian kernels at different scales, a fundamental operation in image processing. To identify the local structure, KAZE (Alcantarilla *et al.*, 2012) algorithm computes the gradient of the image. Using partial derivatives, the gradient is created to show how the intensity of the image varies in different directions.

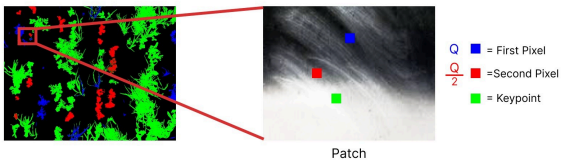


Fig. 6: Keypoints detection by Binary comparison between pixels in image patches (Tyagi, 2019)

After that Extremal regions in the nonlinear scale space are identified by KAZE (Alcantarilla *et al.*, 2012). A region in the image where the gradient magnitude is at its maximum or smallest is known as an Extremal region (Matas *et al.*, 2004). Extremal regions are measured using scale-normalized Laplacian as a base. The extremal regions maximize or minimize the determinant of the Hessian matrix (AL-Rammahi, 2007).

Keypoint descriptors are computed by KAZE (Alcantarilla *et al.*, 2012) after keypoints have been located and identified. These descriptors are essential for feature matching because they capture the local image information around each keypoint. By choosing extremal regions and extracting their coordinates and scale, KAZE (Alcantarilla *et al.*, 2012) localizes keypoints. To express the local image structure, KAZE (Alcantarilla *et al.*, 2012) employs a descriptor that is similar to the Local Binary Pattern (LBP) code. In a neighboring region, this descriptor reflects patterns of intensity variations. For each sample point, LBP code is measured for every sample.

To organize related LBP-like descriptors into clusters, KAZE (Alcantarilla *et al.*, 2012) employs K-Means clustering. The feature vectors' dimensionality is decreased by substituting cluster centers for the descriptors. The scale-space construction demonstrates how KAZE (Alcantarilla *et al.*, 2012) achieves scale-Invariance by taking keypoints into account at various scales. Although KAZE (Alcantarilla *et al.*, 2012) does not naturally offer rotation Invariance, extra methods can be used, including an estimate of the orientation of keypoints and keypoints matching across multiple orientations.

AKAZE

Accelerated KAZE (AKAZE), which constructs a scale space via nonlinear diffusion, is regarded as one of the first algorithms to discover features (Kalms *et al.*, 2017). Contrast factor computation, non-linear scale-space construction, and feature detection make up three components of the AKAZE algorithm. Scale-space representation using 3 octaves and 4 levels is illustrated in Figure 7 based on non-linear diffusion. This image illustrates how an image is blurred gradually in various scales and octaves.

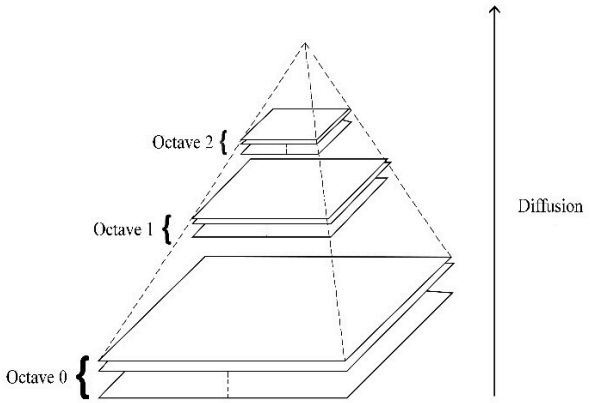


Fig. 7: AKAZE's scale-space representation (Kalms *et al.*, 2017)

Contrast Factor Computation enhances keypoint detection by evaluating the local contrast around each keypoint. This involves calculating the intensity differences between the keypoint and its surrounding pixels, often derived from the gradient magnitude, which

measures the change in intensity. Keypoints with higher gradient magnitudes indicate more significant features. AKAZE uses this contrast factor to apply a threshold, discarding keypoints with lower contrast to retain only the most prominent and reliable ones. This process improves the algorithm's robustness and accuracy, especially in challenging conditions like varying lighting or noisy environments.

On the other hand, Nonlinear Scale-Space Building for AKAZE algorithm refers to a process that involves creating a multi-scale representation of an image using nonlinear diffusion filtering. This is different from the traditional linear Gaussian scale-space used in other algorithms like SIFT. This diffusion technique enhances the preservation of edges and important details, leading to more accurate and robust feature detection. AKAZE (Kalms *et al.*, 2017) uses a pyramidal framework for its scale-space (Figure 7). It consists of octaves, with each octave containing sub-levels. Each successive octave is one-fourth size of the previous octave.

Classical Approaches vs. Deep Learning Approaches

Image features such as local features and global features are important in selecting either classical algorithms or deep learning algorithms of feature detection. Global features represent characteristics of an entire image (Chen *et al.*, 2021). These features capture information about the overall content and structure of the image. Examples of global features include color histograms, texture descriptors (e.g., Gabor filters) (Mehrotra *et al.*, 1992), shape descriptors (e.g., Hu moments) (Huang and Leng, 2010), and deep learning features extracted from the entire image using CNNs (Purwono *et al.*, 2023; Krizhevsky *et al.*, 2012). Global features are useful for tasks such as image classification, scene recognition, and image retrieval. On the other hand, local features represent distinctive regions or keypoints within an image. These features capture information about specific parts of the image, such as corners, edges, or blobs, and are often invariant to changes in scale, rotation, and illumination. Examples of local feature descriptors include SIFT, SURF, ORB, and BRISK. Local features are commonly used for tasks such as image matching, object detection, image registration, and panorama stitching. Local features typically outperform global features, as they excel in identifying significant visual characteristics within an image. Similarly, methods based on local features offer improved classification or retrieval performance and possess strong discriminative power in addressing most computer vision challenges compared to global features (Utomo *et al.*, 2021).

Moreover, deep learning-based techniques for real-time feature detection from images leverage deep learning models to extract meaningful features or patterns from visual data in real time. Some widely used neural network techniques used for real-time feature detection from images are CNNs (Purwono *et al.*, 2023; Krizhevsky *et al.*, 2012), Region-Based CNNs (R-CNNs) (Girshick *et al.*, 2016), Single Shot MultiBox Detector (SSD) (Liu *et al.*, 2016), You Only Look Once (YOLO) (Redmon *et al.*, 2016), Feature Pyramid Networks (FPNs) (Lin *et al.*, 2017), MobileNet (Howard *et al.*, 2017), and so on. These neural network techniques enable real-time feature detection from images across a wide range of applications, from surveillance systems and autonomous vehicles to augmented reality and medical imaging. They provide the capability to analyze and understand visual information rapidly and accurately, facilitating intelligent decision-making and automation in diverse domains.

In PA, real-time image analysis is crucial for some tasks such as crop monitoring, disease detection, yield estimation, and irrigation management. To perform real-time image analysis effectively, it is essential to use image feature detection algorithms that are computationally efficient, robust to environmental variations, and capable of detecting relevant features for agricultural applications. Here are some image feature detection algorithms that can be used for real-time image analysis in PA: classical algorithms like FAST (Lavin and Gray, 2015), ORB (Rublee *et al.*, 2011), SURF (Bay *et al.*, 2006), SIFT (Lindeberg, 2012) and CNN-based approaches. FAST (Lavin and Gray, 2015) can be used for crop monitoring and plant identification. ORB features can be used for tasks such as recognition of plant species, detection of weeds, and estimation of crop yield. SURF (Bay *et al.*, 2006) features can be used for tasks such as crop disease detection, soil moisture estimation, and crop health monitoring. SIFT (Lindeberg, 2012) can be used for real-time image analysis tasks in PA, such as plant phenotyping, leaf counting, and fruit detection. While these algorithms excel in detecting and describing local features in images, they are not directly integrated into CNNs due to their handcrafted nature and fixed feature extraction process. Madhuri *et al.*, presents a hybrid model combining Multi-Head Attention-based Bi-Directional Gated Recurrent Unit (M-Bi-GRU) with CNN. In this algorithm method, the Adaptive Reptile Search Optimization (ARSO) algorithm is employed for feature selection to enhance prediction accuracy. Lin Fudong *et al.*, in their research proposed a deep learning model called MMST-ViT. This model uses a multi-modal spatial-temporal vision transformer to combine satellite imagery and meteorological data. As a result of this combination, the authors claim that this model can accurately detect the impact of agricultural yields influenced by both short-term and long-term climate change.

In their research paper, Yewle and Karakus introduce the RicEns-Net model (Yewle and Karakus, 2024). Using a deep ensemble structure, this model integrates weather measurements, optical remote sensing data, and synthetic

aperture radar (SAR). By narrowing down more than 100 predictors to 15 essential traits, feature selection improved prediction accuracy. Lei Zhang presents another deep learning model in their article (Zhang *et al.*, 2024). This model improves the accuracy of yield estimation by processing multi-source data, such as climate, EVI, LAI, and solar-induced chlorophyll fluorescence (SIF), by combining CNN and Bi-Directional Long Short-Term Memory (BiLSTM) networks. Vignesh *et al.* (2023) in their study proposed a model for predicting agricultural yields considering crop and environmental parameters. This proposed model used a Discrete Deep Belief Network (DBN) with a VGG Net classifier. They also used the Tweak Chick Swarm Optimization technique to improve it. Zhou *et al.* (2023) present an alternative algorithm for the prediction of the wheat yield. In this model, agronomic variables are incorporated with multi-temporal spectral information from the UAV-captured images using Random Forest method.

However, classical feature detection algorithms can still complement CNNs in several ways, such as prepossessing, data augmentation, fine tuning, hybrid architecture, and transfer learning. There are several attempts (Utomo *et al.*, 2021; Tsourounis *et al.*, 2022; Chen *et al.*, 2021) in the last couple of years for a hybrid approach by combining the classical algorithm and deep learning algorithm for efficient real-time feature detection. The choice of image feature detection algorithm depends on factors such as specific task requirements, computational resources, and environmental conditions. By selecting the appropriate algorithm and optimizing its implementation, real-time image analysis can be achieved effectively in PA applications.

Data Collection and Preprocessing

Data Collection

In our study, we used a custom image dataset captured by the UAV. After collecting, blurry and poor quality images were filtered out and formed a dataset of 750 raw images along with another 3000 images captured using ND filters (750 images for each filters) of the UAV. The details of the image data set collected are illustrated in Table 1.

Image Augmentation

Since captured image dataset is smaller comparatively, we augmented the images to increase the size of the dataset. This is required to improve the performance, generalization and robustness of the models. To augment the raw dataset, we consider three factors, such as rotation, illumination, and scaling for image augmentation.

We consider clockwise 90^0, clockwise 180^0, and counterclockwise 90^0 rotations for rotation-based image augmentation.

We scaled up the raw images by 2× and 3×, and scaled down by 2× for scaling-based image augmentation.

Table 1: Data Collection Details

Item	Description
Image Type	Tomato Field
Number of Images	750 raw images (Images are captured by the UAV both from the above and from the side) 3000 images using the UAV filters
Number of Augmented Images	4500 (using scale and brightness as augmentation factors)
Location	Tomato Fields located in Terokhada, Khulna District, Bangladesh. Coordinates: Latitude: $22^{\circ}57'35.24''$ N, Longitude: $89^{\circ}40'2.39''$ E.
Data Collection Medium	DJI 2*S* UAV
Diversity Condition	Illumination, rotation, scaling, and occlusions.
Illuminance Conditions	It was a regular sunny day (7 February, 2024) in the Winter season. Illuminance levels during capturing images were: $28570 \pm 150 LUX (8:00 AM - 11:00 AM)$, $46810 \pm 90 LUX (12:00 PM - 1:30 PM)$, and $24560 \pm 110 LUX (2:00 PM - 3:30 PM)$.
Filtering Term	Dataset consists of 90% or more annotated images. *ND*4, *ND*8, *ND*16, and *ND*32 are used to reduce light by factors of 1/4, 1/8. 1/16, and 1/32 during image capture.
Photo Resolution	750 pixels X 750 pixels
Image Type	RGB
Auto Orient	Applied

Image Annotation

Roboflow was used for image annotation. Roboflow was selected for image annotation because of its easy-to-use web interface, support for various annotation types (including polygons and bounding boxes), and integrated tools for format conversion and data augmentation. By facilitating effective labeling, tracking dataset versions, and exporting in formats compatible with widely used machine learning frameworks, it streamlines the process of preparing datasets. We use 7 classes to label the images. These classes are “tomato”, “weed”, “water”, “soil”, “stem”, “leaf”, and “other” (Figure 8).

Image Normalization

All annotated and raw images are normalized using Min-Max Normalization. OpenCV was used for normalization of all images before using the images as input.

Evaluation Metrics

In this study, eight of the widely used existing feature detection algorithms were investigated. To understand their feature detection capacity and compare their performance, 8 evaluation metrics such as number of *matching keypoints*, *percentage of consistent keypoints*,

repeatability rate, *matching robustness*, *detection rate*, *false positive rate*, *mean localization error*, and *standard deviation of mean localization error (MLE)* were evaluated. These evaluation metrics are illustrated below:

Consistent Keypoints

If keypoint K_i from the image I_i needs to be extracted, it must be required to identify keypoints that are consistent in all images I_1, I_2, I_3, ..., I_n of the same class. In this case, following formula can be used to measure the consistent keypoints:

$$ConsistentKeypoints = \{(K_i, K_j) \mid \text{dist}(D_i, D_j) < \varepsilon \wedge \|p_i - T(p_j)\| < \delta\},$$

where, D_i,D_j = descriptors of keypoints K_i,K_j , ε = a threshold, p_i, p_j are coordinates of keypoints, T is a transformation matrix aligning image j to image i and δ is a spatial threshold.

Repeatability Rate

In feature detection, the repeatability rate measures how reliably a detector detects the same physical points (also known as keypoints or features) in several images of the same scene, particularly when subjected to various transformations (such as scale, rotation, illumination, etc.).

$$RepeatabilityRate = \frac{N_{\text{repeatable}}}{min(N_1, N_2)}$$

where, $N_{\text{repeatable}}$ = number of matching keypoints between an original image and its transformed images.

N_1 and N_2 = number of keypoints in the original image and the transformed images, respectively.

Matching Robustness

The ratio of correct matches to total matches under various circumstances is known as *Matching Robustness* (Edstedt *et al.*, 2024).

$$MatchingRobustness = \frac{N_{\text{correctmatches}}}{N_{\text{totalmatches}}}$$

where, $N_{\text{correctmatches}}$ = Number of keypoint matches that are geometrically correct. $N_{\text{totalmatches}}$ = All matches returned by the descriptor matching algorithm.

False Positive Rate

The False Positive Rate (FPR) in feature detection measures the frequency with which a detector misidentifies a feature that should not be matched or detected (Padilla *et al.*, 2020).

$$FPR = \frac{N_{\text{falsematches}}}{N_{\text{totalmatches}}}$$

where, $N_{\text{falsematches}}$ = keypoints that are incorrectly matched,

$N_{\text{totalmatches}}$ = Total keypoint pairs returned by the matcher.

Detection Rate

In the context of feature detection, the Detection Rate quantifies the ability of a feature detector to correctly identify true keypoints in an image, often in comparison to ground truth (Padilla *et al.*, 2020).

$$DetectionRate = \frac{N_{\text{truepositives}}}{N_{\text{truepositives}} + N_{\text{falsenegatives}}}$$

$N_{\text{truepositives}}$ = number of correctly detected keypoints $N_{\text{falsenegatives}}$ = number of ground truth keypoints that are not detected correctly.

Mean Localization Error

A widely used evaluation metric in feature detection is the Mean Localization Error (MLE), which calculates the average separation between the identified keypoints and their matching ground truth keypoints (Dai *et al.*, 2025; Frigieri *et al.*, 2017; Oksuz *et al.*, 2018). MLE can be represented by the following formula:

$$MLE = \frac{1}{N}\sum_{i=1}^{n} \|p_i - p_i^{gt}\|$$

where, N =Number of matched keypoints, p_i = position of the i_{th} detected keypoints, $p^{\text{gt}}{}_i$ = position of the corresponding i_{th} ground truth keypoints, and $\|p_i - p_i^f\|$= Euclidean distance e_i.

Standard Deviation of Mean Localization Error

The variance in the localization accuracy of identified keypoints in relation to ground truth positions is measured by the standard deviation of MLE which is calculated by the following formula (Frigieri *et al.*, 2017):

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(e_i - MLE)^2}$$

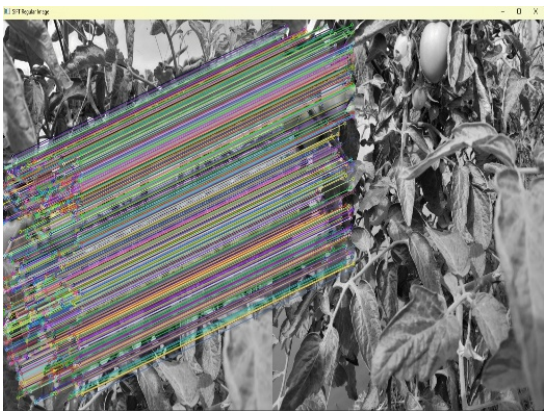
Fig. 8: Sample annotated image

Experimental Data

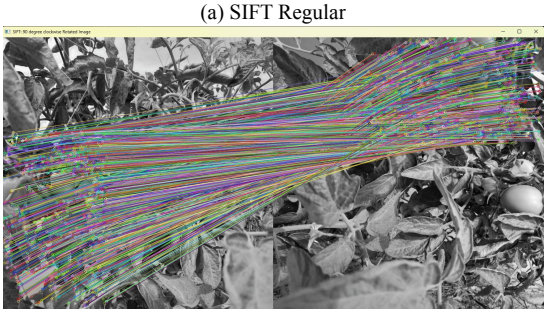
In this experiment, three influential factors such as *Rotation*, *Scale*, and *Illumination* are considered. Additionally, to conduct the experiment we have used a system with 16 GB of RAM and a Quad-Core Intel CPU with a 3.0 GHz clock speed.

Classical feature detection methods are evaluated using Python and the OpenCV library. Hyper-parameters of the algorithms are also fine-tuned to identify consistent accuracy. Besides, the consistency threshold value of 20-100 is used in this investigation. The consistency threshold is a criterion that measures how consistent a match is across multiple views or frames. The consistency threshold is often defined on the basis of geometric constraints or similarity measures between keypoints. In this research, we have investigated Rotation Invariance, Illumination Invariance, and Scale Invariance to identify the value of evaluation metrics.

Rotation Invariance: In computer vision and image processing, Rotation Invariance is vital for tasks like object recognition, where an algorithm must identify an object regardless of its orientation in an image. This allows for more robust and versatile PA applications, ensuring that the system's performance is not hindered by the orientation of the objects it analyzes. Therefore, in our experiment, we have used the rotated image with different angles such as 90^0, 180^0, and 270^0. For example, Fig. 9 and 10 represent the keypoint matching for the rotated images using SIFT.

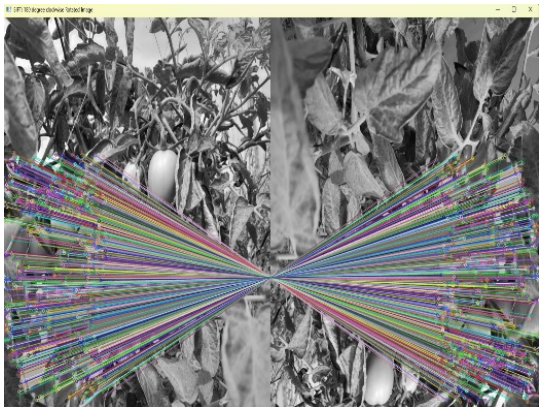


(a) SIFT Regular

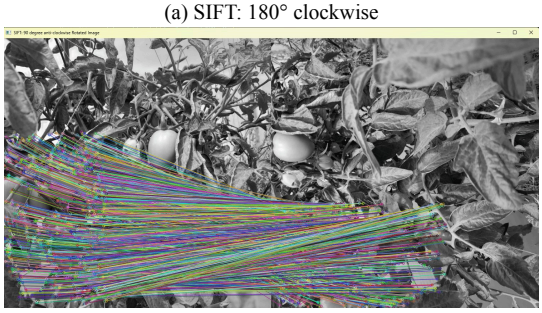


(b) SIFT: 90° clockwise

Fig. 9: Keypoint matching between rotated paired images using SIFT



(a) SIFT: 180° clockwise



(b) SIFT: 270° anticlockwise

Fig. 10: Keypoint Matching between rotated paired images using SIFT

Each row represents a distinct rotation angle, while columns present corresponding feature detection performance metrics. Notably, Rotation Invariance proves pivotal, as it ensures consistent feature identification regardless of object orientation. The values in Table 2 underscore the algorithm's robustness across various rotations, emphasizing its ability to maintain reliable feature detection under different angular perspectives. This comprehensive analysis in the table reaffirms the significance of Rotation Invariance in bolstering the overall efficacy of feature detection algorithms in complex visual environments.

Scale Invariance: The Table 3 elucidates the significance of Scale Invariance in feature detection, showing performance metrics across different scales. Rows represent varied scales, while columns display the corresponding feature detection results of the contending algorithms. The values in Table 3 underscore the algorithm's consistent ability to identify features across diverse scales, ensuring reliable performance in scenarios where object dimensions may vary. Fig. 11 represents images with different scales.

Since Feature Detection involves identifying and extracting meaningful patterns or structures from data, Scale Invariance ensures that these features can be reliably detected regardless of changes in their size or scale. This property is essential in scenarios where the size of objects or patterns within an image may vary and a consistent detection approach is desired. Therefore, we have tested the algorithms on different scales to check

DOI: 10.3844/jcssp.2025.2153.2170

their robustness. We scaled the images (Figure 11) to (−)30% (3 times scaled down), (+)200% (2 times scaled up) and (+)300% (3 times scaled up) in our investigation. The results of our experiment are stated in Table 3.

Table 2: Comparing Feature Detection Methods Using Rotated Images (Bold Values Signify Desirable/Near-Desirable Metrics)

Rotation Inv.	Evaluation Metrics	SIFT	SURF	ORB	FREAK	BRIEF	BRISK	KAZE	AKAZE
Base	Matching Keypoints (Unit: count)	3437	4458	234	3666	1174	**5040**	3437	2917
	Consistent Keypoints (%)	58.85	61.5	**63.33>**	50.34	49.41	36.4	58.85	42.2
	Repeatability Rate (%)	73.78	**82.09**	70.6	60.45	69.87	58.5	53.79	51.97
	Matching Robustness (%)	58.85	**67.5**	64.6	49.4	51.12	62.4	58.85	42.2
	Detection Rate (%)	58.85	**65.25**	60.6	50.34	49.42	56.4	65.85	55.2
	False Positive Rate (%)	41.18	39.2	**29.4**	49.66	50.58	43.6	41.15	47.8
	Std of Localization Error (%)	2.07	1.2	**0.022**	2.06	0.12	**0.03**	4.08	**0.007**
0° clockwise	Matching Keypoints	3485	4126	234	2839	655	**5022**	3485	2771
	Consistent Keypoints (%)	59.67	**61.0**	64.01	40.03	42.2	31.02	59.67	42.2
	Repeatability Rate (%)	**74.82**	**75.5**	71.6	55.22	60	61.35	54.5	64.82
	Matching Robustness (%)	59.67	**72.22**	63.16	50.93	42.5	51.02	59.67	57.06
	Detection Rate (%)	59.67	**63.33**	**61.6**	55.5	50	38.98	40.33	59.94
	False Positive Rate (%)	40.35	46.11	**38.4**	55.5	47.7	**38.98**	40.33	39.94
	Std of Localization Error (%)	7.64	7.1	**4.63**	6.09	9.6	4.6	7.64	**4.92**
180° clockwise	Matching Keypoints	3459	4089	234	2455	870	**5005**	3459	2862
Consistent Keypoints (%)	59.23	**67.12**	**68.7**	44.05	50.43	40.97	59.23	48.43	
Repeatability Rate (%)	69.26	**74.49**	**71.0**	58.16	67.12	61.29	47.26	49.78	
Matching Robustness (%)	59.23	70.9	**62.6**	59.15	39.5	**60.97**	59.23	40.43	
Detection Rate (%)	59.23	58.05	**67.9**	60.015	54.5	**60.97**	59.23	54.43	
False Positive Rate (%)	40.77	44.89	**39.4**	39.99	53.54	49.03	40.77	**39.57**	
Std of Localization Error (%)	4.95	4.56	**0.44**	**3.0**	1.05	9.09	2.95	4.43	
90° anti-clockwise	Matching Keypoints	3327	**3802**	234	2768	110	5024	3327	2773
	Consistent Keypoints (%)	56.97	**58.26**	**57.45**	50.00	47.03	31.35	34.45	57.21
	Repeatability Rate (%)	**71.43**	**72.44**	64.6	60.25	66.49	41.78	69.43	68.88
	Matching Robustness (%)	56.97	**64.57**	**65.6**	50.06	49.1	61.35	56.97	57.21
	Detection Rate (%)	56.97	**64.88**	**63.16**	50.06	55.2	61.35	56.97	57.21
	False Positive Rate (%)	51.5	43.03	**39.4**	49.99	52.45	38.65	43.11	**42.8**
	Std of Localization Error (Unit: pixels)	4.85	5.57	**0.57**	5.0	**2.12**	5.52	4.85	7.95

Table 3: Comparing Feature Detection Methods Using Scaled Images (Bold Values Signify Desirable/Near-Desirable Metrics)

Scale Inv.	Evaluation Metrics	SIFT	SURF	ORB	FREAK	BRIEF	BRISK	KAZE	AKAZE
Base Image	Matching Keypoints (Unit: count)	3982	**4817**	2927	1174	4746	240	3380	2779
	Consistent Keypoints (%)	58.85	**61.5**	**63.33**	50.34	49.41	36.4	58.85	42.2
	Repeatability Rate (%)	68.1	**75.11**	**71.4**	55.83	60.14	52.15	49.17	49.88
	Matching Robustness (%)	59.04	**60.91**	**61.11**	52.51	48.33	57.79	58.02	54.45
	Detection Rate (%)	**63.33**	**70.25**	62.8	54.41	50.55	60.9	66.0	61.87
	False Positive Rate (%)	38.7	**33.9**	**32.32**	52.07	56.29	41.19	47.99	44.19
	Std of Localization Error (%)	0.81	0.52	**0.024**	2.0	1.05	**0.044**	0.66	0.24
(+200% Image)	Matching Keypoints	2196	**4125**	1442	516	2946	235	1777	1231
	Consistent Keypoints (%)	**50.3**	47.15	**55.0**	42.39	40.15	29.6	41.0	46.71
	Repeatability Rate (%)	71.50	**72.91**	**73.9**	62.05	59.91	61.5	57.05	49.88
	Matching Robustness (%)	61.11	**69.15**	**70.6**	59.12	56.6	61.73	66.33	57.13
	Detection Rate (%)	66.33	**70.92**	**71.3**	59.52	58.34	62.7	64.22	56.19
	False Positive Rate (%)	54.23	**51.4**	**46.66**	55.23	61.33	60.32	61.34	59.9
	Std of Localization Error (%)	**0.081**	**0.073**	**0.057**	0.15	0.116	0.46	2.41	2.04
(+300% scaled-up Image)	Matching Keypoints	8972	9676	7298	2923	**20869**	500	10228	10177
	Consistent Keypoints (%)	81.56	**86.6>**	**87.91**	82.2	80.78	89.2	85.33	79.3
	Repeatability Rate (%)	70.0	74.4	**82.33**	79.3	80.86	80.0	**81.19**	72.8
	Matching Robustness (%)	86.8	**87.32**	**89.06**	82.0	80.23	83.77	80.65	72.3
	Detection Rate (%)	86	87.2	**90.06**	82.1	79.86	85.5	**88.01**	84.66
	False Positive Rate (%)	20.0	**18.5**	**12.94**	22.3	19.14	20.11	24.52	30.0
	Std of Localization Error (%)	4.22	3.94	**0.2**	**0.41**	5.98	10.0	3.13	5.23
(+33% scaled-up Image)	Matching Keypoints	7549	8881	5283	2959	**10776**	153	5718	7763
	Consistent Keypoints (%)	41.57	**46.5**	**48.3**	22.61	33.52	28.2	25.91	21.48
	Repeatability Rate (%)	32.9	**27.18**	30.99	36.16	34.241	30.26	**29.9**	30.6
	Matching Robustness (%)	32.9	**39.52**	**45.99**	16.16	26.09	22.2	26.1	20.71
	Detection Rate (%)	32.9	**35.02**	**38.99**	16.16	17.241	10.2	29.91	23.59
	False Positive Rate (%)	**67.09**	**61.88**	69.01	75.84	71.76	79.8	70.1	79.4
	Std of Localization Error (Unit: pixels)	4.22	3.94	**0.2**	**0.41**	5.98	10	3.13	5.23

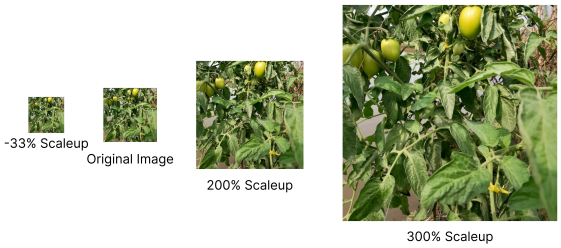


Fig. 11: Average Scale Invariance of feature matching for different scales-up image

Illumination Invariance: Illumination Invariance is a pivotal aspect of feature detection in computer vision and image processing. The challenge arises from the fact that lighting conditions in an environment can significantly affect the appearance of objects, which makes feature detection algorithms crucial to be robust to variations in illumination. Illumination Invariance ensures that these algorithms can reliably identify and extract features from images, regardless of changes in lighting intensity or direction.

To address illumination variations, feature detection algorithms incorporate techniques that normalize or compensate for changes in brightness, contrast, and shadows. In our investigation, we have captured images under different lighting conditions, as shown in Figure 12. Table 4 shows our experimental result based on different lighting conditions such as (+)50% high, (−)25% low, and (−)50% low exposures. This investigation was simulated for 50 epochs. Overexposed and highly underexposed images were consistently found to result in lower accepted values of the evaluation metrics. However, moderately underexposed images (in this investigation, images captured using *ND*4 filters) were found to result in consistent accepted values of the evaluation metrics.

(a) Illumination invariance: sample image 1 with different illumination

(b) Illumination Invariance: sample image 2 with different illumination

Fig. 12: Sample images (tomato) with different brightness for identifying illumination invariance

Table 4: Comparing Feature Detection Methods Using Illuminated Images (Bold Values Signify Desirable/Near-Desirable Metrics)

Illumination	Evaluation Metrics	SIFT	SURF	ORB	FREAK	BRIEF	BRISK	KAZE	AKAZE
Base Image	Matching Keypoints (Unit: count)	3982	**4889**	2927	1118	4746	240	3380	2779
	Consistent Keypoints (%)	78.1	**81.42**	**83.9**	77.1	75.6	74.1	80.4	77.6
	Repeatability Rate (%)	**37.16**	30.9	39.96	27.76	30 .59	32.6	28.2	28.29
	Matching Robustness (%)	37.16	**45.23**	**48.96**	27.76	39.59	32.6	28.2	25.31
	Detection Rate (%)	37.16	41.14	**43.36**	27.76	30.59	32.6	38.2	38.28
	False Positive Rate (%)	62.84	47.9	**39.04**	72.24	89.41	77.4	**41.8**	61.72
	Std of Localization Error (%)	0.81	0.52	**0.024**	0.4	**0.05**	0.043	2.66	4.1
25% Low Exposure	Matching Keypoints	2221	4077	1636	581	**3144**	280	1821	1433
	Consistent Keypoints (%)	63.6	**64.2**	**71.14**	39.64	44.77	42.6	52.34	50.15
	Repeatability Rate (%)	**33.018**	22.05	**38.15**	21.18	20.99	20.6	26.9	30.35
	Matching Robustness (%)	29.35	**46.55**	**50.76**	41.12	40.2	30.6	42.59	40.34
	Detection Rate (%)	29.35	**41.05**	**47.6**	40.12	40.0	40.6	35.72	30.34
	False Positive Rate (%)	50.65	49.08	**39.27**	59.88	59.91	49.4	**44.28**	49.7
	Std of Localization Error (%)	5.08	**3.073**	**2.011**	5.0	6.036	7.047	7.29	6.11
50% Low Exposure	Matching Keypoints	3089	**9823**	2771	630	250	193	1897	1678
	Consistent Keypoints (%)	38.26	**41.2**	**46.94**	38.3	33.98	31.0	28.79	30.99
	Repeatability Rate (%)	**37.79**	30.0	**42.34**	26.45	20.66	31.0	32.59	27.8
	Matching Robustness (%)	33.12	**39.6**	**41.34**	26.45	23.66	31.0	32.59	37.8
	Detection Rate (%)	37.79	41.5	**52.34**	26.45	40.66	44.3	**48.59**	47.8
	False Positive Rate (%)	62.21	60.09	**59.28**	73.55	66.34	65.3	**47.41**	52.2
	Std of Localization Error (%)	**0.16**	0.25	**0.02**	0.75	0.37	0.189	1.44	0.55
50% High Exposure	Matching Keypoints	1890	**8909**	1360	168	1524	171	909	892
	Consistent Keypoints (%)	45.69	48.5	51.15	46.84	40.26	40.2	36.32	41.69
	Repeatability Rate (%)	29.48	28.88	28.57	21.71	30.36	30.2	35.19	32.53
	Matching Robustness (%)	39.48	42.38	**42.6**	25.30	28.26	29.2	36.32	37.7
	Detection Rate (%)	30.48	35.7	**38.58**	26.71	27.26	20.2	35.9	**37.7**
	False Positive Rate (%)	60.52	60.09	**57.21**	**57.29**	54.74	57.8	63.68	68.3
	Std of Localization Error (Unit: pixels)	2.56	3.94	**0.24**	5.63	6.072		9.6	3.03

Results and Discussion

Evaluating feature detection algorithms requires accounting for various factors, such as scale changes, rotation, illumination variations, noise, blur, and occlusions. In this study, we focus specifically on scale changes, rotation, and illumination variations, as these factors significantly impact object detection in overlapping images. In the following, we present the experimental results based on the three key aspects.

Table 2 presents the measured values of various evaluation metrics for the base images, as well as 90° clockwise, 180° clockwise, and 90° counter-clockwise rotated images. When clockwise-rotated images were considered, the impact of rotation invariance on the evaluation metrics in SIFT was minimal. while the impact on the SURF, AKAZE, KAZE, BRISK, BRIEF, and FREAK was higher and their performance was sharply dropped. For example, the evaluation metric, such as *repeatability*, *detection rate*, *matching robustness*, and *consistent keypoints* decrease dramatically. In contrast, *false positive rate* and *MLE* increases for clockwise rotated images. However, ORB performs moderately and consistently in all simulations.

In contrast , for counter-clockwise rotated (90^0*and* 180^0) images, SIFT performed consistently for all evaluation metrics, while SURF, AKAZE, BRISK, and KAZE performed moderately. In this case, when counter-clockwise 180^0 rotated images were considered, ORB performance was dropped more than 50%.

Similarly, Table 3 provides the measured values of different evaluation metrics for base images and scaled images, including (−)33% scaled-down, +200% scaled up, and (+300%) scaled-up images. In this investigation, the evaluation results show that the values of *repeatability rate, detection rate, consistent keypoints*, and *matching robustness* decreased sharply for down-scaled images for all of the feature detection algorithms. In contrast, the *false positive rate* and *MLE* were increased for down-scaled images. Although the values of the evaluation metrics were better for SIFT and SURF compared to the values of other feature detection algorithms, ORB demonstrated consistent and moderate values for all scaling levels.

Additionally, Table 4 displays the measured values of the evaluation metrics of the feature detection algorithms on the illumination invariance influenced by variable brightness exposures. Our experiment shows that a higher illumination invariance offers better repeatability, matching robustness, and lower false positives for feature detectors. That means, *detection rate, repeatability*, and *consistent keypoints* were dropped while *false positive rate* and *MLE* were decreased for both underexposed and overexposed images. Finally, it can be concluded that gradient-based feature detectors (e.g. SIFT, SURF) are better illumination-invariant compared to the binary feature detectors (e.g. ORB, BRIEF). However, in this investigation, ORB performed consistently in all scenarios and factors considered.

Table 5: Benchmarks of Evaluation Metrics Required by Object Detection Techniques

Features	Requirements
Consistent Keypoints	The consistent keypoints metric means the reliability of keypoints detected under various scenarios including changes in scale, rotation, and illumination. Usually, a higher level of consistent keypoints is preferable. For object detection algorithms High: >75%, Moderate: 50%-75%, Low: <50%.
Repeatability	A Repeatability Rate of 80% or more is typically regarded as good performance in practice. However, the application parameters, the type of data, and the particular difficulties presented by the surroundings or objects being identified can all affect the allowable range of repeatability rate. For object detection algorithms High: >80%, Moderate: 60%-80%, Low: <60%.
Matching Robustness	A higher Matching Robustness is preferable since it shows that the algorithm can construct feature correspondences with accuracy and dependability. The range of acceptable Matching Robustness can change depending on the particular objectives of the feature-matching assignment. For object detection algorithms High: >80%, Moderate: 60%-80%, Low: <60%.
Detection Rate	Generally, a higher Detection Rate of keypoints is desirable, as it indicates that the algorithm effectively captures a larger proportion of true positive instances in image data. For object detection algorithms High: >80%, Moderate: 60%-80%, Low: <60%.
False Positive Rate	A lower False Positive Rate is generally preferable since it shows that the feature detection algorithm being used is less likely to identify things incorrectly. For object detection algorithms High: >10%, Moderate: 5%-10%, Low: <5%.
Mean Localization Error	This metric evaluates the degree to which the identified keypoints correspond to their actual location within the image. Usually, the Mean Localization Error is between one and three pixels. For object detection algorithms High: >3 pixels, Moderate: 1-3 pixels, Low: <1 pixels.
Std of MLE	Since it shows that the mean localization errors are less variable and more constant, a lower standard deviation is preferred. A low standard deviation indicates that the algorithm delivers accurate and reliable feature localization consistent across various scenarios. The mean localization error standard deviations are commonly given as pixel values, with a standard deviation of less than 1 pixel typically regarded as satisfactory performance. For object detection algorithms High: >15%, Medium to High: 05%15%, Medium: 5%-10%. Low: <5%.

DOI: 10.3844/jcssp.2025.2153.2170

Table 6: Metrics-wise evaluation results

Approaches	Metrics						
	Matching Keypoints	Detection Rate	False Positive Rate	Consistent Keypoints	Repeatability Rate	Matching Robustness	Mean Localization Error
SIFT (Lindeberg, 2012)	◐	◐	●	◐	◐	◔	◐
SURF (Bay *et al*., 2006)	●	◔	◐	◔	◔	◔	◔
ORB (Rublee *et al*., 2011)	◔	●	◔	◔	◔	◐	●
FREAK (Alahi *et al*., 2012)	◐	◐	◐	◔	◐	◔	◐
BRIEF (Calonder *et al*., 2010)	◐	◐	●	◐	◐	◐	◐
BRISK (Leutenegger *et al*., 2011)	◐	◔	◔	◐	◐	●	◐
KAZE (Alcantarilla *et al*., 2012)	◔	◐	◐	◐	◐	◔	◐
AKAZE (Kalms *et al*., 2017)	◔	◐	●	◔	◐	◐	◐

● Highly Desirable ◔ Medium to High Desirable ◐ Medium Desirable ○ Low Desirable

Based on the above experimental analysis, we find that the ideal value of the evaluation metrics for measuring the performance of the feature detection algorithms depends on some particular requirements of the application and the quality of the data being processed. Widely used benchmarks of different evaluation matrices for object detection is stated in the Table 5. From our investigation, we find the performance of these metrics for different algorithms stated in Table 6. This table shows that the ORB performed better compared to other algorithms. In Tables 2, 3, and 4, bold faced values show the first and second best values of the evaluation metrics for the algorithms in our investigation. This investigation also found that SURF is the second best performer following the ORB algorithm.

Limitations of Feature Detection Approaches

Despite the growing popularity of the use of the UAV, there are still some significant issues with the UAV-captured image data processing (Pacot and Marcos, 2018; Rokhmana, 2015; Mizotin *et al.*, 2010). In this study, we find the following significant limitations in the UAV-captured image data processing, especially in the field of PA. Since the UAV takes images from various angles, the number of matching keypoints for overlapping images could differ. Rotation invariance, illumination invariance, and scale invariance must therefore be further investigated in order to determine the comparable number of matching keypoints from the overlapping images.

The quality or informativeness of each keypoint may degrade as the number of keypoints increases. Finding the right balance between quantity and quality can be difficult. In this study, it was found that if the number of keypoints (*nfeatures*) for ORB is set to 2000 or more, it floods the image with keypoints including less informative, noisy, and redundant regions. Similarly, decreasing the parameter *hessianThreshold* (threshold for keypoint detector response) to 400 allows SURF to detect fewer but higher-quality keypoints, while setting it to 100 detects more but lower-quality keypoints.

When keypoints are difficult to distinguish in a scene with repeating patterns, feature detectors may have trouble matching, causing ambiguities. In our experiment, ORB identified several similar corners on leaves, stems, or tomatoes in a row by assigning similar binary descriptors but failed to detect unique features, leading to false matches. Further research on feature detection algorithms is needed to improve repeatability.

Existing feature detection algorithms lack robustness in diverse environmental scenarios with variations in lighting, airflow, weather, and clutter. For instance, in this study, ORB was unable to detect matching keypoints under shadowed and overexposed images.

There is an absence of occlusion-aware feature identification algorithms that can identify only occluded features, determine their significance, and distinguish occluded from unoccluded features. In this study, when a ripe tomato was partially covered by a leaf, SIFT did not always detect the tomato's keypoints.

There may be very little forward overlap since the attitude angles (Hirakoso *et al.*, 2016) between adjacent UAV-captured images are substantially larger than those in conventional aerial images. Feature matching in this scenario presents a significant challenge for detectors dealing with PA-related issues.

The lateral overlap degree may not be sufficient for image mosaicing since flight paths are curved.

To make feature detectors more robust in real-world scenarios, data augmentation techniques must be improved. These techniques involve training detectors with synthetic data that include occlusions and overlapping objects.

Feature detection algorithms should ensure temporal consistency in feature detection and the ability to track features across temporarily occluded or overlapped frames.

Parameter sensitivity is a key challenge in feature detection. Some algorithms require fine-tuning of

parameters, and their sensitivity affects performance. For example, ORB performance may vary significantly due to *nfeatures* (number of keypoints to retain) and *scaleFactor* (pyramid decimation ratio), which affect detection density and robustness. Finding optimal parameters for diverse datasets is challenging.

Feature detection algorithms may lack semantic understanding, leading to the detection of keypoints in irrelevant or non-informative regions. Improving semantic relevance remains an ongoing challenge.

Ensuring that feature detection algorithms generalize well across diverse datasets remains a persistent challenge.

Algorithms that perform well on one type of data may struggle when applied to different scenes or domains.

Addressing these limitations is crucial to advancing feature detection algorithms to perform effectively in complex and dynamic real-world environments, where occlusions and overlapping objects are common challenges. Solutions to these problems can significantly impact various applications, including object recognition, tracking, augmented reality, and robotics, to improve PA performance.

Hence, various preprocessing should be carried out before image processing to prevent the aforementioned issues and ensure that the images are suitable for mosaicing and mapping. The overlapping image analysis and current approaches to solve the overlap problem for the UAV-captured images are kept open for future work.

Combined Use of Classical and DL-Based Approaches

Although classical feature detection algorithms are highly effective and robust, they are not typically used as a deep neural network for real-time image feature detection in PA. Since these classical algorithms themselves are not typically used directly within deep learning models, their feature descriptors can be incorporated into deep learning architectures for tasks such as image retrieval, image classification, and object detection. Here is how these algorithms can be used in conjunction with deep learning for image feature detection algorithms:

Feature extraction: Classical algorithms extract keypoint locations and feature descriptors from images. These keypoints and descriptors capture important information about distinctive regions in the image, such as corners, edges, and texture patterns.

Feature Matching: Once keypoints and descriptors are extracted from multiple images, they can be matched to find the corresponding points between images. This is useful for tasks such as image registration, where the goal is to align different views of the same scene.

Training Data Generation: Descriptors generated from these approaches can be used to generate training data for deep learning models. For example, we can extract SIFT descriptors from an image dataset and use them as input features for training a deep neural network.

Feature Fusion: These descriptors can be used with other types of features (e.g., deep learning-based features such as CNN activation) to improve the performance of tasks such as image classification or object detection. Fusion techniques can include concatenating feature vectors, using attention mechanisms, or combining feature maps at different network layers.

Fine-tuning Pretrained Models: Descriptors extracted by the classical Computer Vision approaches can be used to fine-tune pre-trained deep-learning models for specific tasks. For example, we can use SIFT descriptors as additional input channels to a CNN and fine-tune the network's weights on a new dataset with limited labeled data.

Hybrid Approaches: Researchers have explored hybrid approaches that combine handcrafted feature descriptors like FAST with deep learning architectures. These approaches leverage the complementary strengths of handcrafted and learned features to improve performance in tasks such as image matching, object tracking, and visual localization.

Conclusion

Remote sensing technology like the UAV is being popular in PA especially crop detection, crop yield predition, leaf disease detection, weed detection, forecasting harvesting period, etc. Therefore, increasing the efficiency of the UAV-based image analyzing methods is crucial. In line with this goal, this study explored the feature detection algorithms (as depicted in Section 2) to analyze their performance using the UAV-captured images of the tomato field. In this review study, standard benchmarks (Table 5) are also identified for the evaluation metrics of these feature detection methods. In our experiment, we have considered three factors such as rotation, illumination, and scaling for selected images. In the future, we shall also deploy different machine learning algorithms (Adnan and Islam 2016, 2017; Adnan *et al.*, 2021) for the purpose of parameter tuning in different application contexts. After evaluating these methods, we have identified their ability (Table 6) and limitations (stated in Section 7.1) to detect features in the images of a tomato field. We found that ORB and SURF among the classical feature detection methods performed better in feature detection in all three scenarios (i.e. for rotated, scaled, and illuminated images). Our investigation also found that fusion of the classical feature detection methods with deep learning methods may enhance the efficiency in feature detection, particularly in the real-time process. Therefore, further studies are required to resolve these limitations.

Data Availability

The dataset is publicly available at https://github.com/sarwar-ku/Dataset-Tomato.

Acknowledgment

This research has been carried out as the part of a post graduate project who cooperates with the Department of Computer Science and Engineering, Jashore University of Science and Technology, Bangladesh.

Funding Information

This research did not receive any specific grant from funding agencies for this study.

Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors Contributions

Muhammad Sarwar Jahan Morshed: Conceptualization, Data curation, implementation and experiment, formal analysis, visualization, Writing-original draft, Writing-review, and editing.

Md. Nasim Adnan: Data curation, visualization, Writing-review, editing, and supervision.

Md. Rafiqul Islam: Data curation, visualization, Writing-review, and editing.

Ethics

This study does not involve human participants, human data, or animals. The images used were captured solely of agricultural fields (tomato crops) using the UAV, in compliance with local regulations governing drone usage. No ethical approval was required. All procedures performed were in accordance with applicable institutional, national, and international guidelines and regulations.

References

Adnan, M. N., Ip, R. H. L., Bewong, M., & Islam, M. Z. (2021). BDF: A new decision forest algorithm. *Information Sciences*, *569*, 687–705. https://doi.org/10.1016/j.ins.2021.05.017

Adnan, M. N., & Islam, M. Z. (2017). Forest PA : Constructing a decision forest by penalizing attributes used in previous trees. *Expert Systems with Applications*, *89*, 389–403. https://doi.org/10.1016/j.eswa.2017.08.002

Adnan, Md. N., & Islam, Md. Z. (2016). Forest cern: A new decision forest building technique. *Lecture Notes in Computer Science*, *9651*, 304–315. https://doi.org/10.1007/978-3-319-31753-3_25

Alahi, A., Ortiz, R., & Vandergheynst, P. (2012). FREAK: Fast Retina Keypoint. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 510–517. https://doi.org/10.1109/cvpr.2012.6247715

Alcantarilla, P. F., Bartoli, A., & Davison, A. J. (2012). Kaze features. *Springer*, 214–227. https://doi.org/10.1007/978-3-642-33783-3_16

AL-Rammahi, A. M. H. (2007). On studying hessian matrix with applications. *Scientific Journal of Qadisya University*, *2*(2), 69–76.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded Up Robust Features. *Computer Vision – ECCV*, *3951*, 404–417. https://doi.org/10.1007/11744023_32

Bojanic, D., Bartol, K., Pribanic, T., Petkovic, T., Donoso, Y. D., & Mas, J. S. (2019). On the Comparison of Classic and Deep Keypoint Detector and Descriptor Methods. *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 64–69. https://doi.org/10.1109/ispa.2019.8868792

Cen, C., Li, R., & Xu, X. (2019). Fast robust matching algorithm based on BRISK and GMS. *Journal of Physics: Conference Series*, *1237*(2), 022070. https://doi.org/10.1088/1742-6596/1237/2/022070

Chen, L., Rottensteiner, F., & Heipke, C. (2021). Feature detection and description for image matching: from hand-crafted design to deep learning. *Geo-Spatial Information Science*, *24*(1), 58–74. https://doi.org/10.1080/10095020.2020.1843376

Cheng, X., Jinling, W., & Yaming, X. (2010). overlap analysis of the images from unmanned aerial vehicles". *2010 International Conference on Electrical and Control Engineering*, 1459–1462. https://doi.org/10.1109/ICECE.2010.360

Dai, S., Bai, T., & Zhao, Y. (2025). Keypoint Detection and 3D Localization Method for Ridge-Cultivated Strawberry Harvesting Robots. *Agriculture*, *15*(4), 372. https://doi.org/10.3390/agriculture15040372

Edstedt, J., Sun, Q., Bökman, G., Wadenbäck, M., & Felsberg, M. (2024). RoMa: Robust Dense Feature Matching. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19790–19800. https://doi.org/10.1109/cvpr52733.2024.01871

Fawakherji, M., Potena, C., Pretto, A., Bloisi, D. D., & Nardi, D. (2021). Multi-Spectral Image Synthesis for Crop/Weed Segmentation in Precision Farming. *Robotics and Autonomous Systems*, *146*, 103861. https://doi.org/10.1016/j.robot.2021.103861

Frigieri, E., Borghi, G., Vezzani, R., & Cucchiara, R. (2017). Fast and Accurate Facial Landmark Localization in Depth Images for In-Car Applications. *Image Analysis and Processing - ICIAP 2017*, *10484*, 539–549. https://doi.org/10.1007/978-3-319-68560-1_48

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2016). Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Vol. 38, Issue 1, pp. 142–158). https://doi.org/10.1109/tpami.2015.2437384

Hirakoso, N., Tajima, K., Andou, M., Matsumoto, A., & Shigematsu, Y. (2016). Study on specification of attitude angle for small satellite by lunar outline extraction. *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 288–293. https://doi.org/10.1109/sice.2016.7749256

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, Tobias, Andreetto, Marco, & Adam, Hartwig. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv:1704.04861v1*. https://doi.org/10.48550/arXiv.1704.04861

Huang, J., Zhou, G., Zhou, X., & Zhang, R. (2018). A New FPGA Architecture of FAST and BRIEF Algorithm for On-Board Corner Detection and Matching. *Sensors*, *18*(4), 1014. https://doi.org/10.3390/s18041014

Huang, Z., & Leng, J. (2010). Analysis of hu's moment invariants on image scaling and rotation. *Proceedings of the 2010 2nd International Conference on Computer Engineering and Technology (ICCET)*. https://doi.org/10.1109/ICCET.2010.5485542

Kalms, L., Mohamed, K., & Göhringer, D. (2017). Accelerated Embedded AKAZE Feature Detection Algorithm on FPGA. *Proceedings of the 8th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies*, 1–6. https://doi.org/10.1145/3120895.3120898

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet Classification With Deep Convolutional Neural Networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Lavin, A., & Gray, S. (2015). Fast algorithms for convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, 4013–4021. https://doi.org/10.1109/CVPR.2016.437

Lepetit, V., & Fua, P. (2006). Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(9), 1465–1479. https://doi.org/10.1109/tpami.2006.188

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary Robust invariant scalable keypoints. *2011 International Conference on Computer Vision*, 2548–2555. https://doi.org/10.1109/iccv.2011.6126542

Lin, F., Crawford, S., Guillot, K., Zhang, Y., Chen, Y., Yuan, X., Chen, L., Williams, S., Minvielle, R., Xiao, X., Gholson, D., Ashwell, N., Setiyono, T., Tubana, B., Peng, L., Bayoumi, M., & Tzeng, N.-F. (2023). MMST-ViT: Climate Change-aware Crop Yield Prediction via Multi-Modal Spatial-Temporal Vision Transformer. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 5751–5761. https://doi.org/10.1109/iccv51070.2023.00531

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936–944. https://doi.org/10.1109/cvpr.2017.106

Lindeberg, T. (2012). Scale Invariant Feature Transform. *Scholarpedia*, *7*(5), 10491. https://doi.org/10.4249/scholarpedia.10491

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. *Computer Vision – ECCV 2016*, *9905*, 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. https://doi.org/10.1023/b:visi.0000029664.99615.94

Madhuri, J., Indiramma, M., & Nagarathna, N. (2025). M-Bi-GRU-CNN: a hybrid deep learning model with optimized feature selection for enhanced crop yield prediction. *Multimedia Tools and Applications*, *84*, 39787–39811. https://doi.org/10.1007/s11042-025-20747-9

Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, *22*(10), 761–767. https://doi.org/10.1016/j.imavis.2004.02.006

Mehrotra, R., Namuduri, K. R., & Ranganathan, N. (1992). Gabor filter-based edge detection. *Pattern Recognition*, *25*(12), 1479–1494. https://doi.org/10.1016/0031-3203(92)90121-x

Mizotin, M., Krivovyaz, G., Velizhev, A., Chernyavskiy, A., & Sechin, A. (2010). Robust matching of aerial images with low overlap. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 13–18.

Oksuz, K., Cam, B. C., Akbas, E., & Kalkan, S. (2018). Localization Recall Precision (LRP): A New Performance Metric for Object Detection. *Computer Vision – ECCV 2018*, *1211*, 521–537. https://doi.org/10.1007/978-3-030-01234-2_31

Pacot, M. P., & Marcos, N. (2018). Feature-based stitching algorithm of multiple overlapping images from unmanned aerial vehicle system. *Asian Journal of Basic and Applied Sciences*.

Padilla, R., Netto, S., & Silva, E. (2020). A survey on performance metrics for object-detection algorithms. *Journal of Applied Computing (or ArXiv Preprint, Depending on Citation)*. https://doi.org/10.48550/arXiv.2006.04181

Purwono, P., Ma'arif, A., Rahmaniar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Haq, Q. M. ul. (2023). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, *2*(4), 739–748. https://doi.org/10.31763/ijrcs.v2i4.888

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/cvpr.2016.91

Rokhmana, C. A. (2015). The Potential of UAV-based Remote Sensing for Supporting Precision Agriculture in Indonesia. *Procedia Environmental Sciences*, *24*, 245–253. https://doi.org/10.1016/j.proenv.2015.03.032

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision*, 2564–2571. https://doi.org/10.1109/iccv.2011.6126544

Swati, M. (2014). Assocham study suggests annual crop losses worth rs 50,000 crore due to pests in india. *The Times of India*. https://timesofindia.indiatimes.com/business/india-business/assocham-study-suggests-annual-crop-losses-worth-rs-50000-crore-due-to-pests-in-india/articleshow/30363563.cms

Tian, D. (2013). A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, *8*(4), 385–395.

Tsourounis, D., Kastaniotis, D., Theoharatos, C., Kazantzidis, A., & Economou, G. (2022). SIFT-CNN: When Convolutional Neural Networks Meet Dense SIFT Descriptors for Image and Sequence Classification. *Journal of Imaging*, *8*(10), 256. https://doi.org/10.3390/jimaging8100256

Tyagi, D. (2019). Introduction to SURF (Speeded-Up Robust Features). *Medium*. Retrieved from https://medium.com/data-breach/introduction-to-surf-speeded-up-robust-features-c7396d6e7c4e

Utomo, A., Juniawan, E. F., Lioe, V., & Santika, D. D. (2021). Local Features Based Deep Learning for Mammographic Image Classification: In Comparison to CNN Models. *Procedia Computer Science*, *179*, 169–176. https://doi.org/10.1016/j.procs.2020.12.022

Vignesh, K., Askarunisa, A., & M. Abirami, A. (2023). Optimized Deep Learning Methods for Crop Yield Prediction. *Computer Systems Science and Engineering*, *44*(2), 1051–1067. https://doi.org/10.32604/csse.2023.024475

Yewle, A. D., & Karakus, O. (2024). Multi-modal data fusion and deep ensemble learning for accurate crop yield prediction. *ArXiv:2502.06062*. https://doi.org/10.48550/arXiv.2502.06062

Zhang, L., Li, C., Wu, X., Xiang, H., Jiao, Y., & Chai, H. (2024). BO-CNN-BiLSTM deep learning model integrating multisource remote sensing data for improving winter wheat yield estimation. *Frontiers in Plant Science*, *15*, 1500499. https://doi.org/10.3389/fpls.2024.1500499

Zhou, H., Yang, J., Lou, W., Sheng, L., Li, D., & Hu, H. (2023). Improving grain yield prediction through fusion of multi-temporal spectral features and agronomic trait parameters derived from UAV imagery. *Frontiers in Plant Science*, *14*, 1217448. https://doi.org/10.3389/fpls.2023.1217448