YOLO-TMHC: An Optimized Model Balancing Precision and Efficiency for Tibetan Medicinal Herb Recognition

Xinlong Hu
[1]*School of Information Science and Technology*
XiZang University
Lhasa,XiZang Autonomous Region
erhuer2@stu.utibet.edu.cn

Yue Shen
[1]*School of Information Science and Technology*
XiZang University
Lhasa,XiZang Autonomous Region
15889001427@163.com

Nuo Qun*
Collaborative Innovation Center for XiZang Informatization by MOE and XiZang Autonomous Region
Lhasa,XiZang Autonomous Region
q_nuo@utibet.edu.cn
*Corresponding author

Ji De
Key Laboratory of Biodiversity and Environment on the Qinghai-Tibetan Plateau, Ministry of Education
Lhasa,XiZang Autonomous Region
dekyi1981@utibet.edu.cn

***Abstract*— Accurate recognition of Tibetan medicinal herbs is crucial for the research and application of modern traditional Chinese medicine. However, existing manual recognition methods face challenges such as high labor intensity and high misidentification rates. Therefore, this study proposes a deep learning-based classification model for Tibetan medicinal herbs, named YOLO-TMHC. YOLO-TMHC is built upon YOLOv11n and incorporates various improvement strategies to meet the high accuracy requirements of practical applications. Specifically, the C3k2 module of the backbone network is improved by adding a concatenated diverse branch block (DBB) module. By combining multi-scale convolution, sequential convolution, and average pooling operations, the model is able to learn more comprehensive and specific target features. After the connection operation in the neck structure, an efficient bidirectional feature pyramid network (BiFPN) is introduced to achieve bidirectional fusion and weighted optimization of multi-scale features, further enhancing the model's ability to extract key features in object detection. Additionally, a CAFM attention mechanism is added after each C3k2 module in the neck to further improve the model's learning of key features. This study uses a dataset based on 12 categories of Tibetan medicinal herbs, consisting of 29,384 images. Experimental results demonstrate that the proposed YOLO-TMHC achieves the best recognition performance on this herbal dataset. The mAP@0.5 of YOLO-TMHC is 98.14%, which represents a 1.32% improvement over YOLOv11n in mAP@0.5. Moreover, YOLO-TMHC recognizes a single image in just 28.6 ms. The experimental results show that YOLO-TMHC outperforms currently mainstream recognition models and achieves high-precision recognition of Tibetan medicinal herbs. This study provides technical support for the development of modern herbal recognition and edge recognition devices.**

Keywords—Tibetan medicinal herb recognition,Deep learning, YOLOv11, Attention module, Real-time recognition

I. Introduction

Tibetan medicinal herbs, as a crucial component of Tibetan medicine, have a long-standing history and play a vital role in promoting health, particularly in the western regions of China 10[1]. In recent years, these herbs have garnered increasing attention for their potential contribution to modern traditional Chinese medicine (TCM) and global herbal medicine research [2,3]. Currently, over 3,000 species of Tibetan medicinal herbs are known, and rapid identification of these herbs is essential for effective diagnosis and treatment[4]. However, the identification of Tibetan medicinal herbs faces significant challenges due to their wide variety and the complexities involved in distinguishing between them. Traditional methods of herb identification, which rely heavily on manual classification, face numerous limitations due to the diverse textures and morphologies of the herbs [5]. These methods are not only time-consuming but also prone to errors, making them inefficient and incapable of meeting the demands for large-scale herb identification. Therefore, these limitations hinder the widespread application and development of Tibetan medicinal herbs in both research and clinical settings [6,7].

With the rapid advancement of deep learning technologies, convolutional neural networks (CNNs) and other machine learning algorithms have provided a promising solution to the challenges posed by the diversity and complexity of Tibetan medicinal herb identification [8]. These methods, which excel in image classification and pattern recognition, have demonstrated great potential in automating and accelerating the identification process. By leveraging large datasets and advanced model architectures, deep learning techniques can learn to extract critical features from herb images, enabling more accurate and efficient identification. This offers a significant advantage over traditional methods [9,10]. The application of deep learning for Tibetan medicinal herb identification opens up new opportunities for enhancing the efficiency and scalability of herb classification, thus supporting the broader application of Tibetan medicine in modern healthcare and research.

In recent years, scholars have begun to apply deep learning techniques for non-destructive identification of medicinal herbs. Hajam et al. [11] summarized recent methods for herb identification, emphasizing the wide applicability of artificial intelligence technologies in this field. Vo et al. [12] collected images of 10 medicinal herbs from Vietnam and constructed a dataset of approximately 10,000 images. They further trained a deep lear ning model based on VGG16, achieving an accuracy of 93.6%. Roopashree et al. [13] created a dataset of 40 Indian medicinal herbs with a total of around 2,500 images, and used pre-trained models such as VGG16, VGG19, InceptionV3, and Xception for recognition, achieving an accuracy of 97.5%. Quoc et al. [14] collected images of 100 therapeutic herbs, retaining about 10,000 images, and trained a VGG16 model, ultimately achieving an average accuracy of 99.275% for all the herbs. However, current research in medicinal herb identification based on deep learning technologies faces several challenges. Specifically, most studies have collected fewer than 10,000 images, which often leads to models that are overfitted due to the relatively low difficulty of the recognition task. Moreover, mainstream research still relies on older models like VGG networks, which are now considered outdated [15,16]. Therefore, there is a pressing need to develop a more robust and high-performance recognition model to address the challenges of herb identification effectively.

The YOLO (You Only Look Once) series of algorithms has gained widespread adoption in various object detection tasks due to its end-to-end training approach and outstanding real-time detection performance [17-21]. As the YOLO series has evolved, YOLOv7 and YOLOv8 have focused on optimizing detection accuracy, achieving significant breakthroughs in small object detection and performance under complex backgrounds [22, 23]. In comparison, YOLOv10 is an upgraded version of YOLOv8 that further enhances the model's capability to detect multi-scale objects. Additionally, YOLOv10 introduces optimizations in model size and computational efficiency, making it more suitable for resource-constrained devices [24]. With the ongoing developments, the YOLO algorithm has now reached version 11, continuing to push the boundaries of detection precision, speed, and adaptability.

Therefore, this study builds upon YOLOv11 and introduces several improvements to better suit the herb recognition task. The improvement strategies specifically include enhancing the C3k2 module of the backbone network by adding a concatenated diverse branch block (DBB) module. This modification enables the model to capture more detailed and diverse target features through the integration of multi-scale convolutions, sequential convolutions, and average pooling operations. Additionally, after the connection operation in the neck structure, an efficient bidirectional feature pyramid network (BiFPN) is introduced to perform bidirectional fusion and weighted optimization of multi-scale features, thus improving the model's ability to extract crucial features for object detection. Furthermore, a CAFM attention mechanism is incorporated after each C3k2 module in the neck structure to enhance the model's attention to key features, boosting its performance in complex and diverse recognition tasks. These improvements collectively contribute to the model's robustness, accuracy, and efficiency in recognizing Tibetan medicinal herbs.

The main contributions of this study are as follows:

(1)This study collected images of 12 categories of Tibetan medicinal herbs, constructing a dataset containing a total of 29,384 herb images.

(2)To effectively address the challenges of Tibetan medicinal herb recognition, this study is based on YOLOv11 and incorporates various improvement strategies, further developing the YOLO-TMHC Tibetan medicinal herb recognition model.

(3)Extensive experiments demonstrate that YOLO-TMHC achieves the best performance compared to currently state-of-the-art recognition models, offering a good balance in terms of recognition speed, model size, and recognition accuracy. This provides technical support for the development of subsequent edge devices.

II. Related Works

A. Development of object detection

The field of object detection has undergone a significant transformation, shifting from traditional handcrafted feature-based methods to deep learning-driven approaches. Earlier techniques, such as SIFT, HOG, and Haar features, relied on manually designed descriptors combined with classical classifiers. While effective in controlled scenarios, these methods struggled with complex environments due to their limited adaptability. Similarly, conventional image-based object detection techniques, including threshold segmentation and edge detection, were constrained by fixed feature extraction rules, making it difficult to achieve high-precision detection across diverse conditions. Traditional detection pipelines involved generating candidate regions, extracting features, and classifying objects, often requiring extensive manual tuning, which limited their scalability and real-world applicability. Despite their relevance in specific domains, deep learning-based methods have now become the dominant paradigm, as they automatically learn feature representations from data, significantly improving detection accuracy and robustness.

With the rapid advancement of deep learning, object detection has evolved towards learning feature representations directly from large-scale datasets, improving both precision and efficiency. Detection algorithms are generally classified into two-stage and one-stage approaches. Two-stage detectors first generate candidate regions before refining and classifying them, whereas one-stage detectors predict object categories and locations in a single step, offering higher computational efficiency. Among them, the YOLO series has gained widespread adoption due to its balance between detection speed and accuracy.

The YOLO framework, initially introduced by Redmon et al., has continuously evolved through multiple iterations. YOLOv2 and YOLOv3 introduced enhanced feature extraction and improved training strategies, while YOLOv4 optimized detection efficiency [19]. Subsequent versions, including YOLOv5, YOLOv6 with its EfficientNet-based backbone, and YOLOv7 leveraging model reparameterization techniques, further improved performance [20-23]. Recent advancements such as YOLOv9 and YOLOv10 continue to refine accuracy and computational efficiency, making YOLO one of the most widely used object detection frameworks across various domains, including robotics, autonomous driving, and surveillance [24]. The latest version, YOLOv11, incorporates architectural optimizations and advanced attention mechanisms to enhance detection performance, particularly in complex scenes and for small-object detection [25].

III. Materials and Methods

A. *YOLOv11*

YOLOv11, released by Ultralytics on September 30, 2024, is an object detection algorithm. Compared to previous YOLO models, YOLOv11 achieves significant improvements in both accuracy and speed . The network architecture, as shown in Fig 1, is primarily composed of the backbone network, neck network, and detection head. Key innovations of the model include an optimized backbone network, enhanced feature fusion mechanisms, an efficient detection head design, and improvements for small object detection . With these advancements, YOLOv11 improves detection accuracy while maintaining real-time performance .

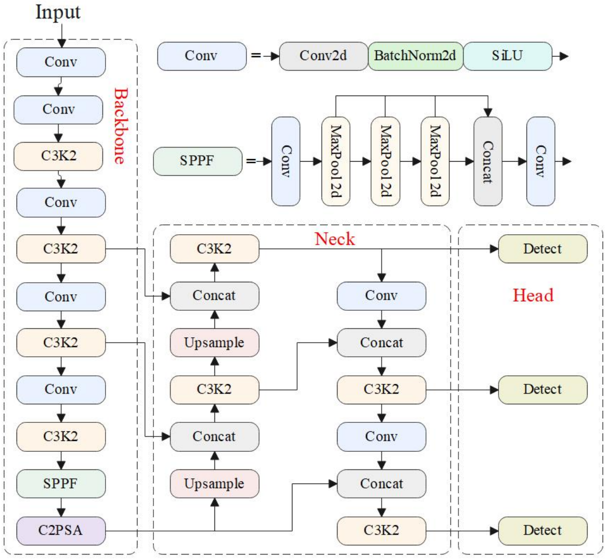


Fig 1. The network architecture of YOLOv11.

YOLOv11 utilizes an improved version of CSPDarknet53 as its backbone network, generating feature maps at different scales through five downsampling operations, labeled P1 to P5. Building on this, YOLOv11 replaces the original C2f module with the C3K2 module. The CBS (Convolution, Batch Normalization, and SiLU activation) module in the backbone network first performs convolution, followed by batch normalization, and finally applies the SiLU activation function to enhance the output. Additionally, the backbone network introduces the Spatial Pyramid Pooling Fast Module (SPFF), which pools the feature maps to a fixed size, thereby increasing the diversity of feature representations. The C2PSA module (with Pyramid Slice Attention mechanism) further strengthens the feature extraction capability. Its PSA mechanism uses a multi-level design, improving upon the SE attention mechanism and making it more suitable for handling multi-level features.

The neck network adopts the PAN-FPN structure, which enhances the fusion of shallow positional information and deep semantic information through a bottom-up path, addressing the limitations of object localization information in the FPN structure. YOLOv11's detection head employs a decoupled structure, with independent branches used for predicting class and location information. Depending on the task, different loss functions are selected: binary cross-entropy loss (BCELoss) for classification tasks and distribution focal loss (DFL) with CIoU for bounding box regression. Furthermore, two DWConv operations are added to the classification detection head, significantly reducing both the parameter count and computational cost. Overall, YOLOv11 enhances real-time object detection performance through a series of innovative designs and optimizations, demonstrating its strong potential in various computer vision tasks.

B. *BiFPN*

The core idea of the BiFPN module is to optimize the multi-scale feature representation problem in object detection through efficient bidirectional cross-scale connections and weighted feature fusion. Multi-scale feature fusion is one of the key challenges in object detection, as targets can vary significantly in scale. Traditional detectors often face performance bottlenecks when dealing with objects of different scales . To address this issue, the BiFPN module introduces learnable weights to dynamically adjust the importance of features at different scales, thereby achieving higher accuracy in the detection of multi-scale targets .

Traditional Feature Pyramid Networks (FPN) make predictions based on pyramid features extracted by the backbone network, using a top-down path to integrate multi-scale features. However, FPN does not fully consider the transfer of features from lower to higher levels, which can lead to the insufficient propagation of fine-grained information in high-level features. To address this issue, PANet (Path Aggregation Network) adds a bottom-up path on top of FPN, further enhancing the feature propagation capability. The traditional FPN aggregates multi-scale features in a top-down manner, and the formula is as follows:

$$P_7^{out} = \mathrm{Conv}(P_7^{in}) \tag{1}$$

$$P_6^{out} = \mathrm{Conv}(P_6^{in} + \mathrm{Resize}(P_7^{out})) \tag{2}$$

$$\cdots$$

$$P_3^{out} = \mathrm{Conv}(P_3^{in} + \mathrm{Resize}(P_4^{out})) \tag{3}$$

Where $P_3^{in}, \ldots P_7^{in}$ are the input features. $P_3^{out}, \ldots P_7^{out}$ are the output features. Resize is usually an upsampling or downsampling op for resolution matching, and Conv is usually a convolutional op for feature processing.

Unlike the unidirectional fusion in FPN and PANet, the BiFPN module repeatedly performs both top-down and bottom-up feature fusion, allowing low-level features to better interact with high-level features and preventing information loss. This bidirectional fusion mechanism effectively improves the model's ability to recognize objects of different scales, particularly enhancing its sensitivity to small objects and targets that are farther from the camera, especially in complex scenarios. In addition, another major advantage of the BiFPN module is its introduction of a learnable weight mechanism, which allows the importance of features from different scales to be dynamically adjusted during the fusion process. This weight learning enables the model to intelligently select which features are most critical for object detection, thereby achieving more efficient feature fusion across different scales. This design not only improves the model's performance but also ensures computational efficiency, maki ng it particularly suitable for resource-constrained environments, such as edge computing and mobile devices. The formula for the two fused features at level 6 of BiFPN is as follows:

$$P_6^{td} = \mathrm{Conv}\left(\frac{\omega_1 \cdot P_6^{in} + \omega_2 \cdot \mathrm{Resize}(P_7^{in})}{\omega_1 + \omega_2 + \epsilon}\right) \quad (4)$$

$$P_6^{out} = \mathrm{Conv}\left(\frac{\omega_1' \cdot P_6^{in} + \omega_2' \cdot P_6^{td} + \omega_3' \cdot \mathrm{Resize}(P_5^{out})}{\omega_1' + \omega_2' + \epsilon}\right) \quad (5)$$

Where ω_i is a learnable weight that can be a scalar (per feature), a vector (per channel), or a multi-dimensional tensor (per pixel). $\epsilon = 0.0001$ is a small value to avoid numerical instability. P_6^{td} is the intermediate feature at level 6 on the top-down pathway. P_6^{out} is the output feature at level 6 on the bottom-up pathway.

In the implementation of the BiFPN module, feature fusion is accomplished through multiple convolution operations and weighted summation. During each feature fusion step, features from different levels are weighted through both top-down and bottom-up operations, with the weights continuously adjusted throughout the fusion process, ultimately resulting in a refined and efficient multi-scale feature representation. Incorporating the BiFPN module in the neck optimizes the importance of different input features through learnable weights. This dynamic optimization mechanism greatly improves the multi-scale feature fusion efficiency, enabling the model to better handle the challenges of tibetan herb densities, weather conditions, and lighting in real-world applications, thereby enhancing the model's robustness. The structure diagram of FPN is shown in Fig 2a, and the structure diagram of BiFPN is shown in Fig 2b.

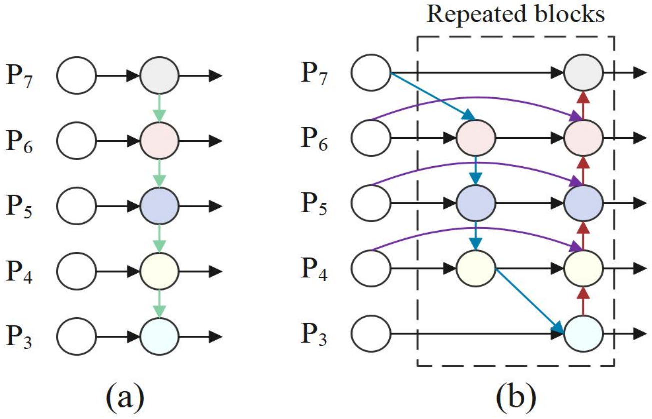


Fig 2. The structure diagram of (a) FPN and (b) BiFPN.

C. CAFM module

In the field of image denoising, effectively combining global and local features is crucial for improving denoising performance. CNNs excel at extracting local features but have limitations in capturing long-range dependencies, while Transformer-based models, which rely on attention mechanisms, are proficient at extracting global features but are relatively weaker in focusing on local details. Against this backdrop, the CAFM was developed to fully integrate the strengths of both approaches, enabling more efficient feature modeling and denoising.

CAFM consists of a collaborative local branch and global branch. In the local branch, to enhance cross-channel information interaction and integration, 1 × 1 convolutions are first applied to adjust the channel dimension. This step effectively controls the number of channels in the feature map, allowing subsequent operations to be performed at an appropriate dimension, reducing computational cost while highlighting key information. Then, a channel shuffling operation is performed, which groups the input tensor along the channel dimension and applies depthwise separable convolutions within each group to deeply mix channel information. The groups are then merged, further enriching the diversity of local features and enabling better capture of local details in the image. Finally, a 3 × 3 × 3 convolution is used to extract features, further refining the local features and providing high-quality local feature representations for subsequent fusion.

$$X_{reduce} = \mathrm{Conv}_{1*1}\left(X_{in} . C \rightarrow \frac{C}{r}\right) \quad (6)$$

$$X_{shuffled} = ChannelShuffle(X_{reduce} \cdot g) \quad (7)$$

$$X_{local} = \mathrm{Conv}_{3*3*3}\left(X_{shuffled}\right) \quad (8)$$

The global branch, on the other hand, leverages the attention mechanism to capture long-range feature dependencies. Specifically, 1 × 1 convolutions and 3 × 3 depthwise separable convolutions are first used to generate queries (Q), keys (K), and values (V). This approach not only reduces computational complexity but also effectively extracts key information from the input features, providing a foundation for the attention calculation. Next, Q and K undergo reshaping and interaction calculations to obtain the attention map. During this process, a learnable scaling parameter α is introduced to control the magnitude of the matrix multiplication in the attention computation, ensuring that the attention mechanism focuses on important feature regions and accurately captures global feature information. Finally, the output of the attention mechanism is added to the input features, achieving an initial fusion of global and local features, with the output containing both local details and global semantic information.

$$\mathrm{Attention} = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) * \alpha \quad (9)$$

$$X_{global} = X_{in} + \mathrm{Attention} * V \quad (10)$$

$$X_{out} = \mathrm{Concat}(X_{local}, X_{global}) + X_{in} \quad (11)$$

In the actual Tibetan medicinal herb image denoising task, the CAFM plays a critical role. For noisy Tibetan medicinal herb images, the local branch focuses on local regions of the image, extracting fine details such as edges and textures, which are crucial for restoring the integrity of the local structure. Meanwhile, the global branch, taking a macroscopic approach, integrates the overall information of the image, such as correlations between different regions, using the attention mechanism. This eff ectively removes noise and further enhances the image quality. The structure diagram of CAFM is shown in Fig 3.

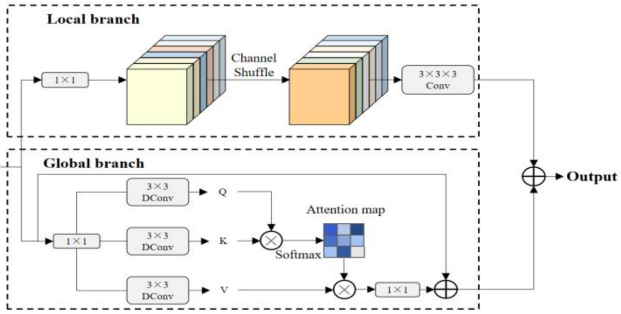


Fig 3. The structure diagram of CAFM.

D. DBB module

In the evolution of Convolutional Neural Networks (ConvNets), balancing performance improvement with inference efficiency has been a central challenge. While innovative architectures such as the Inception series have demonstrated that multi-branch topologies and multi-scale path combinations can enrich feature spaces and boost performance, their complexity often leads to slower inference times. To address this issue, the DBB was introduced, providing a novel solution for optimizing ConvNet performance .

As a fundamental component of ConvNets, DBB employs a multi-branch topology that integrates multi-scale convolutions, sequential convolutions, and average pooling operations .The mathematical expression of its multi-branch feature fusion is:

$$X_{DBB} = Conv_{1*1}\left(Concat\left(Conv_{3*3}(X), Conv_{5*5}(X), Conv_{3*3}(Conv_{3*3}(X)), AvgPool(X)\right)\right) \quad (12)$$

Multi-scale convolutions capture diverse features at various resolutions, sequential convolutions progressively refine features, and average pooling aggregates local features—reducing dimensionality while enhancing robustness. These operations expand the feature space synergistically. Another key advantage lies in DBB's structural reparameterization mechanism. During inference, it reparameterizes into a single convolutional layer via:

$$W_{reparam} = W_{1*1} * Concat(W_{3*3}, W_{5*5}, W_{3*3}^{seq}, W_{avg}) + b_{reparam} \quad (13)$$

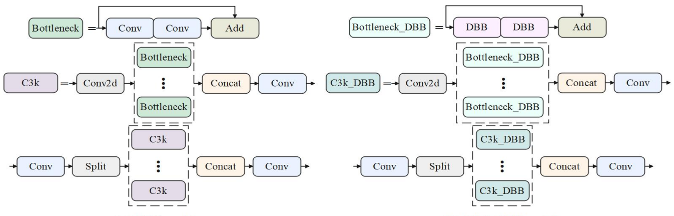
This adjusts kernel parameters and biases for seamless integration into ConvNet architectures, ensuring efficient inference. In this study, the C3K2_DBB module improves the C3K2 structure, with its feature integration defined as:

$$X_{C3K2\_DBB} = Conv_{1*1}\left(Split\left(DBB(X)\right)\right) + Conv_{3*3}(X) \quad (14)$$

The workflow involves: input feature maps passing through a standard convolutional layer for preliminary extraction, splitting via a Split operation for subsequent processing, feature extraction through stacked DBB modules, and final merging/integration via concatenation and a standard convolutional layer to complete feature extraction.

These components create diverse learning paths that enhance feature extraction. Multi-scale convolutions capture features at various resolutions, akin to viewing an object through different focal lengths, offering the model multiple perspectives. Sequential convolutions progressively refine features, uncovering critical information within the data. Average pooling effectively aggregates local features, reducing dimensionality while enhancing robustness. These operations synergistically expand the feature space, allowing the model to learn more comprehensive and representative features, thereby improving performance. Another key advantage of DBB is its unique structural reparameterization mechanism. During training, DBB leverages its complex structure to help the model learn rich feature representations. During inference, it can be reparameterized into a single convolutional layer, maintaining high efficiency. This transformation is based on the linear properties of convolutions and a set of transformation rules, which adjust kernel parameters and biases for seamless integration into traditional ConvNet architectures. This process ensures efficient inference without additional computational cost or time overhead. By enabling flexible switching between training and inference phases, DBB significantly enhances model performance without altering the overall architecture or inference efficiency, making it a valuable tool for ConvNet optimization. Its potential is evident across various computer vision tasks, such as image classification, object detection, and semantic segmentation, offering new avenues for innovation and application in deep learning.

In this study, the C3K2_DBB module is an innovative improvement of the C3K2 structure, aiming to enhance the model's recognition performance by incorporating the DBB module. The workflow of the C3K2_DBB module is as follows: First, the input feature map is passed through a standard convolutional layer to extract preliminary feature representations of the target. Next, the feature map is split into multiple parts using a Split operation for subsequent processing. The split feature maps then undergo feature extraction through several stacked DBB modules. Finally, the extracted features are merged via a concatenation operation and integrated using a standard convolutional layer, completing the feature extraction process. The structure of the C3K2 module is shown in Fig 4a, and the structure of the C3K2_DBB module is shown in Fig 4b.



(a) C3k2 module

(b) C3k2_DBB module

Fig 4. The structure diagram of (a) C3K2 and (b) C3K2_DBB.

E. YOLO-TMHC

The YOLO-TMHC model enhances the YOLOv11 target detection network by replacing the backbone's C3k2 with C3k2_DBB, incorporating the DBB module to optimize feature learning. DBB, with its multi-branch topology, integrates multi-scale convolution, sequential convolution, and average pooling, enriching the feature space and enabling the model to learn more comprehensive and representative features, similar to the Incepti on architecture. After the neck connection, a BiFPN module is added, focusing on efficient bidirectional cross-scale connections and weighted feature fusion, addressing the critical issue of multi-scale feature representation in target detection. Additionally, the CAFM attention mechanism is integrated after each C3k2 block in the neck, reinforcing the learning of valuable features. The model's depth is increased by adding four CAFM layers, further improving its performance. The network architecture of YOLO-TMHC is shown in Fig 5.

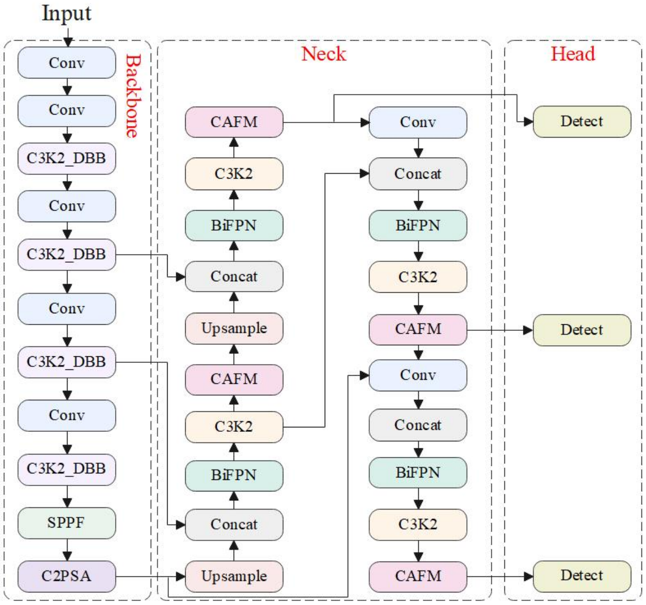


Fig 5. The network architecture of YOLO-TMHC.

IV. Results

A. Data Set

Tibetan medicinal herbs are classified into thirteen major categories, including: Animal-based, Dunbu (Wetland Herb)-based, E (Dryland Herb)-based, Stone-based, Tree-based, Soil-based, Salt-Alkali-based, Treasure-based, Juice Essence-based, Crop-based, Fire-based, Processed, and Water-based. Since Water-based medicinal herbs are primarily composed of liquids, they are difficult to effectively recognize through image recognition. Therefore, this study focuses on the remaining twelve categories, establishing an image dataset with a total of 29,384 images. The data used in this experiment were collected from online sources. Sample images from the twelve categories of medicinal herbs are shown in Fig 6.

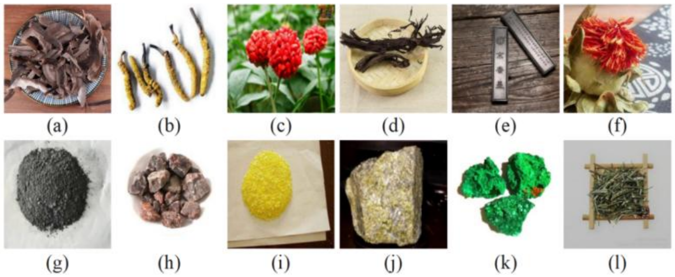


Fig 6. The sample images from the twelve categories of medicinal herbs, including: (a) Animal-based, (b) Dunbu (Wetland Herb)-based, (c) E (Dryland Herb)-based, (d) Stone-based, (e) Tree-based, (f) Soil-based, (g) Salt-Alkali-based, (h) Treasure-based, (i) Juice Essence-based, (j) Crop-based, (k) Fire-based, and (l) Processed.

The dataset was divided into three subsets: training, testing, and validation, with proportions of 80%, 10%, and 10%, respectively. The details of the dataset split are shown in Table 1. The analysis diagram of dataset category instances distribution, object location and size characteristics is shown in Fig 7. The visualization analysis of dataset category quantity, object coordinates and size distribution for YOLO is shown in Fig 8.

Table 1. The specific configuration of the Tibetan Medicinal Herb Dataset.

Tibetan Medicinal Herb	Images	Training	Test	Val
Animal-based	1545	1236	155	155
Dunbu (Wetland Herb)-based	1742	1394	174	174
E (Dryland Herb)-based	3099	2479	310	310
Stone-based	3667	2934	367	367
Tree-based	3048	2438	305	305
Soil-based	2211	1769	221	221
Salt-Alkali-based	360	288	36	36
Treasure-based	6631	5305	663	663
Juice Essence-based	3197	2558	320	320
Crop-based	3092	2474	309	309
Fire-based	486	389	49	49
Processed	306	245	31	31
Total	29384	23507	2938	2938

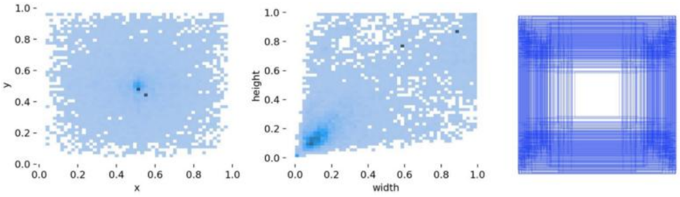


Fig 7. The analysis diagram of dataset category instances distribution, object location and size characteristics.

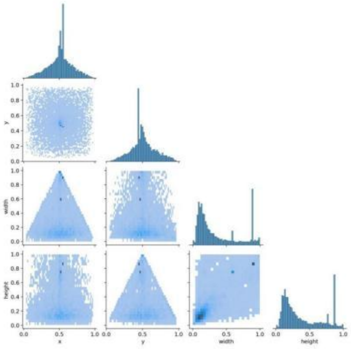


Fig 8. The visualization analysis of dataset category quantity, object coordinates and size distribution for YOLO.

B. Experimet Stadards

The configuration of the training platform in this study is shown in Table 2. During model training, the dynamic initialization learning rate was set to 0.001, the number of training epochs was 100, and the batch size was 16.

TABLE 2. EXPERIMENTAL PARAMETER CONFIGURATION.

Experimental environment	Configuration parameters
CPU	AMD EPYC7T83 64-Core Processor
GPU	RTX 4090(24GB)
Operating system	ubuntu18.04
Software platform	PyCharm
Software packages	PyTorch1.9.0, CUDA 11.1, Python 3.8
Memory	90G

In this study, Precision, Recall, and mean average precision at IoU 0.5 (mAP@0.5) are used as the main evaluation metrics for the performance of Tibetan medicine detection. Precision represents the proportion of correctly detected Tibetan medicine items among all detected instances, assessing the accuracy of the model. Recall measures the proportion of actual Tibetan medicine items that are correctly detected, evaluating the model's ability to avoid missing detections. Mean average precision considers both precision and recall, evaluating the model's detection performance at various thresholds, with particular focus on the mAP@0.5 condition. The indicators are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{15}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{16}$$

$$\text{AP} = \int_0^1 Precision(Recall)\,\mathrm{d}\,Recall \tag{17}$$

$$\text{mAP@0.5} = \frac{1}{N}\sum_{\text{i}=1}^{N} AP \tag{18}$$

Where TN, TP, and FN are the sample numbers of True negative (TN), True positive (TP), and False negative (FN). N is the total number of identification types.

Additionally, as part of the evaluation for lightweight models, the experiment also considers model size, complexity, and real-time performance. Model size is assessed by the number of parameters, which indicates the number of trainable adjustable parameters. The number of parameters is used to assess the model's size, reflecting the count of trainable adjustable parameters, and serves as an indicator of the model's deployment cost. GFLOPs is used to measure model complexity, reflecting the amount of floating-point operations required for each forward pass. FPS is used to assess real-time performance, indicating the number of image frames the model can process per second. These metrics provide a comprehensive evaluation of the lightweight Tibetan medicine detection model's accuracy, computational efficiency, and real-time capability.

C. Comparison experiments of YOLO series algorithms

In this study, we performed comparative experiments on several optimized YOLO models to evaluate their performance in Tibetan medicine herb detection tasks. The models selected for comparison include YOLOv4-tiny, YOLOv5s, YOLOv7-tiny, YOLOv8n, YOLOv10n, and YOLOv11n. These lightweight YOLO models were chosen for their balance between detection accuracy and computational efficiency, aiming to achieve faster inference speeds while maintaining a high level of performance. The models were compared across key metrics such as Precision, Recall, mAP@0.5, and the number of parameters. The results from the comparative experiment are summarized in Table 3.

From Table 3, it is clear that YOLOv11n outperforms all other models, achieving the highest mAP@0.5 of 96.82%. This is a 4.24%, 6.14%, 3.24%, and 0.16% improvement over YOLOv4-tiny, YOLOv5s, YOLOv7-tiny, and YOLOv8n, respectively. YOLOv11n also excelled in Precision (96.10%) and Recall (96.35%), outperforming the other models in these metrics as well. Despite its high performance, YOLOv11n remains lightweight, with only 2.6 × 106 parameters, which is the smallest among the models tested. This low parameter count indicates that YOLOv11n is highly efficient in terms of both computational resources and inference speed.

In contrast, YOLOv8n, which had the smallest parameter count of 3.0 × 106, achieved impressive Precision (95.84%) and Recall (96.26%), with an mAP@0.5 of 96.66%, making it a strong contender for real-time applications. However, it fell slightly behind YOLOv11n in overall performance. YOLOv7-tiny, YOLOv5s, and YOLOv4-tiny, while still effective, showed relatively lower mAP@0.5 scores and higher parameter counts, indicating a trade-off between detection accuracy and model size.

In summary, YOLOv11n emerged as the best-performing model in terms of both detection accuracy and computational efficiency for Tibetan medicine herb detection. Its high mAP@0.5, combined with a compact model size, makes it ideal for deployment in resource-constrained environments. Based on these findings, YOLOv11n was selected as the base model for further enhancements and model optimization in this research. The comparison charts of data details for several groups of YOLO models are shown in Fig 9.

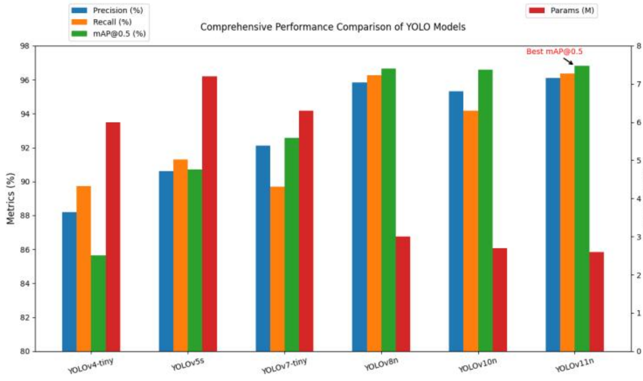


Fig 9. The comparison charts of data details for several groups of YOLO models.

TABLE 3. THE COMPARISON EXPERIMENT OF YOLO SERIES ALGORITHMS.

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	Params (106)
YOLOv4-tiny	88.20	89.72	85.63	6.0
YOLOv5s	90.60	91.29	90.72	7.2
YOLOv7-tiny	92.11	89.71	92.58	6.3
YOLOv8n	95.84	96.26	96.66	3.0
YOLOv10n	95.31	94.19	96.59	2.7
YOLOv11n	96.10	96.35	96.82	2.6

D. Ablation Experiment

To evaluate the contribution of each proposed enhancement in this study, several modules were added to the YOLOv11n baseline model, and ablation experiments were conducted. Specifically, the DBB, CAFM, and BiFPN modules were integrated into YOLOv11n to investigate their effects on the performance of the model in target detection tasks. The results of these ablation experiments are shown in Table 4.

The results demonstrate that each module provided improvements in model performance. After incorporating the DBB module into YOLOv11n, Precision, Recall, and mAP@0.5 increased by 0.30%, 0.33%, and 0.30%, respectively, showing a clear enhancement. This suggests that the DBB module, by introducing a diverse branching structure, enriched the feature learning capabilities of the model, improving its overall accuracy. Incorporating the CAFM module into the YOLOv11n architecture further improved the model's performance. Specifically, Precision, Recall, and mAP@0.5 increased by 0.41%, 0.12%, and 0.17%, respectively. The results highlight that the CAFM module, with its attention mechanism, helped the model focus on more discriminative features, leading to better target detection performance. Adding the BiFPN module resulted in a notable performance boost, especially in terms of mAP@0.5, which rose by 0.06%. Precision and Recall increased by 0.17% and 0.03%, respectively. The BiFPN module effectively improved the multi-scale feature fusion process, ensuring that features from different resolutions were integrated more efficiently, contributing to better detection results. When DBB and CAFM were combined, the model's Precision, Recall, and mAP@0.5 improved by 0.64%, 0.30%, and 0.72%, respectively. This indicates that both modules work synergistically to refine feature extraction and enhance feature attention, making the model more robust in diverse scenarios. Similarly, the combination of DBB and BiFPN improved the model's performance, with Precision increasing by 0.68%, Recall by 0.23%, and mAP@0.5 by 0.80%. The results show that DBB and BiFPN together not only improved feature diversity but also enhanced the fusion of multi-scale features, leading to higher overall performance. When all three modules （DBB, CAFM, and BiFPN) were integrated into YOLOv11n, the model achieved the best results, with a Precision of 97.12%, Recall of 97.03%, and mAP@0.5 of 98.14%. This combination demonstrated a significant improvement in all performance metrics, proving the effectiveness of these modules in boosting the detection capabilities of YOLOv11n.

The ablation experiments confirm that the incremental addition of DBB, CAFM, and BiFPN modules consistently improved the model's performance, with the combination of all three modules resulting in the highest detection accuracy. These findings highlight the substantial benefits of integrating diverse feature extraction, attention mechanisms, and multi-scale feature fusion in enhancing model performance for target detection tasks.

TABLE 4. THE RESULTS OF THE ABLATION EXPERIMENT.

YOLOv11n	DBB	CAFM	Precision (%)	Recall (%)	mAP@0.5 (%)
√			96.10	96.35	96.82
√	√		96.40	96.68	97.12
√		√	96.51	96.22	96.99
√			96.27	96.38	96.88
√	√	√	96.74	96.50	97.84
√	√		96.98	96.58	97.62
√		√	96.78	96.89	97.81
√	√	√	97.12	97.03	98.14

E. Comparison of different attention modules

To systematically analyze the influence of different attention mechanisms, experiments were carried out by incorporating various attention modules into the model. Specifically, attention mechanisms such as ECA (Efficient Channel Attention), SE (Squeeze-and-Excitation), CBAM (Convolutional Block Attention Module), CA (Coordinate Attention), and the proposed CAFM (ours) were embedded. The experimental results of introducing different attention mechanisms are shown in Table5.

Analysis shows that CAFM (ours) excels in all three indicators. In terms of mAP@0.5, CAFM achieves 98.14%, outperforming ECA, SE, CBAM, and CA by 1.44%, 1.02%, 0.66%, and 0.76%, respectively. For Precision, CAFM reaches 97.12%, highlighting its superior capability in accurate positive prediction. In Recall, CAFM's 97.03% also surpasses other mechanisms, reflecting stronger detection ability for positive instances. Overall, the proposed CAFM attention mechanism significantly enhances the model's detection accuracy, demonstrating better performance than traditional attention modules in Precision, Recall, and mAP@0.5, which verifies its effectiveness in improving model performance.

TABLE 5. RESULTS OF INTRODUCING DIFFERENT ATTENTION MECHANISMS.

Group	Model	Precision (%)	Recall (%)	mAP@0.5 (%)
1	+ECA	96.80	96.66	96.70
2	+SE	96.69	96.68	97.12
3	+CBAM	97.01	96.62	97.48
4	+CA	96.77	96.78	97.38

5	+CAFM (ours)	97.12	97.03	98.14

F. *Comparison of model size and recognition speed*

In this section, we evaluate the model size and recognition speed of the YOLO-TMHC algorithm to assess its computational efficiency and suitability for real-time deployment. The results of the model size and recognition speed test are shown in Table 6.

The YOLO-TMHC model has a parameter count of 3.5 million, with an average recognition time of 28.6 ms per image. This results in a high FPS of 35.0, demonstrating the model's ability to perform efficient and rapid target detection. The model also achieves a GFLOPs value of 8.6, further indicating its computational efficiency. These results suggest that YOLO-TMHC achieves a good balance between model size and recognition speed, making it suitable for applications requiring both high performance and low resource consumption.

TABLE 6. THE MODEL SIZE AND RECOGNITION SPEED OF YOLO-TMHC

Model	Params (106)	Average recognition time (ms)	FPS	GFLOPs (G)
YOLO-TMHC	3.5	28.6	35.0	8.6

G. *YOLO-TMHC Performance Testing*

This section presents the performance testing results of YOLO-TMHC. Fig 10 compares the mAP@0.5 training curves of YOLO-TMHC and YOLOv11n. The red curve represents YOLO-TMHC, and the blue curve corresponds to YOLOv11n. From the training curves, it is evident that YOLO-TMHC consistently outperforms YOLOv11n across the entire training period, highlighting the positive impact of the enhancements introduced in this study. Furthermore, four random images from the test set were selected and input into YOLO-TMHC for evaluation. The results are shown in Fig 11. These results demonstrate that YOLO-TMHC is capable of accurately detecting and bounding the target objects, with high detection precision and robustness, showcasing the model's strong performance in practical scenarios.

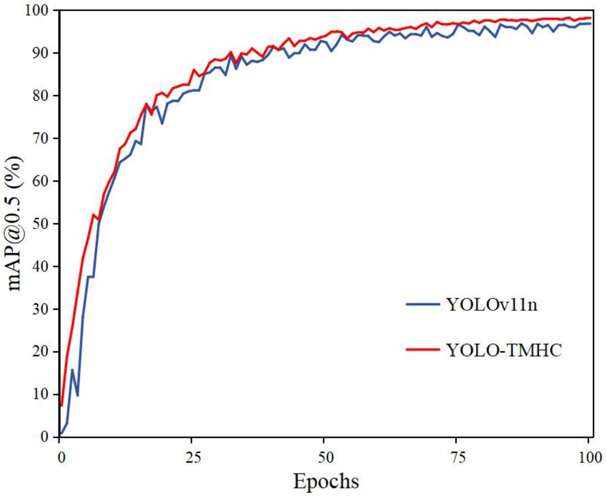


Fig 10. The training curves of YOLOv11n and YOLO-TMHC.

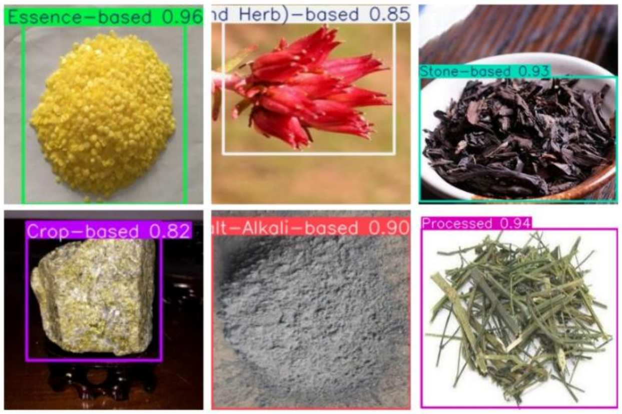


Fig 11. The testing results of YOLO-TMHC.

V. DISCUSSION

In this study, we proposed the YOLO-TMHC model for the detection of Tibetan medicine ingredients, aiming to enhance both the detection accuracy and computational efficiency. The model was evaluated against a variety of existing object detection models, and YOLO-TMHC demonstrated significant improvements, achieving a mAP@0.5 of 98.14%. In comparison to traditional models, YOLO-TMHC excelled in Precision, Recall, and overall detection performance, highlighting its ability to accurately identify Tibetan medicine ingredients in complex scenarios. These results emphasize that YOLO-TMHC offers a promising solution for real-time, high-precision detection tasks, particularly in environments with limited computational resources, thereby advancing the field of Tibetan medicine detection in both research and practical applications. In the future, as deep learning technology continues to advance rapidly, the development of more promising models will open up new possibilities for various applications. In the field of Tibetan medicine recognition, a critical area of research will be the creation of a more lightweight and robust recognition model. Such a model will be essential to enhance the performance and scalability of Tibetan medicine recognition systems. This will be one of the key research directions moving forward. However, several challenges remain in the recognition of Tibetan medicines. One of the most significant obstacles is the large variety of medicinal herbs, which makes it difficult to construct comprehensive datasets for training recognition models. Additionally, many Tibetan medicines exist in powder form, making it more challenging to aggregate and recognize their features effectively. These challenges require more advanced techniques to handle different forms of medicinal products. Moreover, there is currently limited research focused specifically on Tibetan medicine recognition, and studies addressing the recognition of medicinal products remain scarce. Further research is needed to deepen the understanding of Tibetan medicine identification, and progress in this field will require sustained efforts.

Looking ahead, it will also be essential to consider the recognition environment, such as the identification platform. By placing Tibetan medicines within a highly practical recognition platform, we can better simulate real-world recognition scenarios, thereby improving the applicability of the recognition model in real-life situations. This will help bridge the gap between theoretical models and their practical applications.

VI. Conclusions

This study presents the YOLO-TMHC algorithm, specifically developed for accurate Tibetan medicine recognition. By integrating key enhancement modules, we significantly improved the model's detection capability and robustness. Experimental results demonstrate that YOLO-TMHC outperforms the baseline YOLOv11n, achieving an mAP@0.5 of 98.14%, with an improvement of 1.32%. The model also maintains a high processing speed, with an average recognition time of 28.6 ms per image and an FPS of 35.0. These findings indicate that YOLO-TMHC achieves a remarkable balance between high recognition accuracy and real-time performance, making it a promising solution for Tibetan medicine recognition in practical applications. This study provides valuable insights for the development of efficient and robust recognition systems, contributing to the modernization and digitization of Tibetan medicine practices. Future work will focus on further optimizing model performance and exploring its adaptability in diverse environments, expanding its application potential in real-world scenarios.

Acknowledgment

This work was supported by the Key Project of National Natural Science Foundation of China (62436006), the National Natural Science Foundation of China Youth Fund (62406257) and the Science and Technology Plan Project of Xizang Autonomous Region (XZ202501ZY0101)

References

[1] Zhao, F.; Zhang, J.; Liu, Q.; Liang, C.; Zhang, S.; Li, M. Fast Quality Detection of Astragalus Slices Using FA-SD-YOLO. Agriculture 2024, 14, 2194. https://doi.org/10.3390/agriculture14122194

[2] Li, L.; Wang, C.; Li, W.; Chen, J. Hyperspectral image classification by AdaBoost weighted composite kernel extreme learning machines. Neurocomputing 2018, 275, 1725–1733.

[3] Zhao, M.; Yang, B.; Li, L.; Si, Y.; Chang, M.; Ma, S.; Li, R.; Wang, Y.; Zhang, Y. Efficacy of Modified Huangqi Chifeng decoction in alleviating renal fibrosis in rats with IgA nephropathy by inhibiting the TGF-β1/Smad3 signaling pathway through exosome regulation. J. Ethnopharmacol. 2022, 285, 114795.

[4] Dai, G.; Fan, J.; Dewi, C. ITF-WPI: Image and text based cross-modal feature fusion model for wolfberry pest recognition. Comput. Electron. Agric. 2023, 212, 108129.

[5] Li, D.; Zhao, Z.; Yin, Y.; Zhao, C. Research on the Classification of Sun-Dried Wild Ginseng Based on an Improved ResNeXt50 Model. Appl. Sci. 2024, 14, 10613. https://doi.org/10.3390/app142210613

[6] Yu, X.; Feng, X.; Zhang, J.; Huang, J.; Zhang, Q. Research progress on chemical constituents and pharmacological effects of Panax ginseng. Res. Ginseng 2019, 31, 47–51.

[7] Wang, F. Research on Processing Technology and Products of Ginseng Root. China Food Semimonthly Mag. 2024, 53, 124–126.

[8] Kong, F.; Xu, S.; Lu, H.; Cao, S.; Liu, J.; Li, Z.; Sun, W. Exploring Key Technologies for Intelligent Production of Authentic Ginseng, Rooted in Its Three Major Values. Spec. Wild Econ. Anim. Plant Res.

[9] Zhou, C.; Zhao, F.; Di, J.; Cao, S.; Zhang, C.; Zhang, H.; Kong, F. The Application of Mechanized Production in Ginseng Planting and Origin Processing. Spec. Res. 2022, 44, 161–163.

[10] Wang, Q.; Dong, N.; Yang, Y.; Liu, P. Key TCM identification techniques based on image processing and deep learning. Autom. Instrum. 2023, 43, 30–35.

[11] M Hajam, T Arif, A Khanday, Mudasir A Wani, M Asim, “AI-Driven Pattern Recognition in Medicinal Plants: A Comprehensive Review and Comparative Analysis,” Computers, Materials & Continua, 81(2): 2077--2131. doi: 10.32604/cmc.2024.057136.

[12] A. H. Vo, H. T. Dang, B. T. Nguyen, and V. H. Pham, “Vietnamese herbal plant recognition using deep convolutional features,” Int. J. Mach. Learn. Comput, vol. 9, no. 3, pp. 363–367, 2019. doi:10.18178/ijmlc.2019.9.3.811.

[13] S. Roopashree and J. Anitha, “DeepHerb: A vision based system for medicinal plants using xception features,” IEEE Access, vol. 9, pp. 135927–135941, 2021. doi: 10.1109/ACCESS.2021.3116207.

[14] T. N. Quoc and V. T. Hoang, "Traditional Vietnamese Herbal Medicine Image Recognition by CNN," 2023 15th International Conference on Knowledge and Smart Technology (KST), Phuket, Thailand, 2023, pp. 1-5, doi: 10.1109/KST57286.2023.10086725.

[15] J. Abdollahi, “Identification of medicinal plants in ardabil using deep learning: Identification of medicinal plants using deep learning,” in 2022 27th Int. Comput. Conf., Comput. Soc. Iran (CSICC), 2022, pp. 1–6. doi: 10.1109/CSICC55295.2022.9780493.

[16] H. Zhou, F. Luo, H. Zhuang, Z. Weng, X. Gong and Z. Lin, "Attention Multihop Graph and Multiscale Convolutional Fusion Network for Hyperspectral Image Classification," in IEEE Transactions on Geoscience and Remote Sensing, vol. 61, pp. 1-14, 2023, Art no. 5508614, doi: 10.1109/TGRS.2023.3265879.

[17] Li, X.; Duan, W.; Fu, X.; Lv, X. R-SABMNet: A YOLOv8-Based Model for Oriented SAR Ship Detection with Spatial Adaptive Aggregation. Remote Sens. 2025, 17, 551.

[18] Zhu, H.; Xie, Y.; Huang, H.; Jing, C.; Rong, Y.; Wang, C. DB-YOLO: A duplicate bilateral YOLO network for multi-scale ship detection in SAR images. Sensors 2021, 21, 8146.

[19] Zhang, X.; Xuan, C.; Ma, Y.; Su, H.; Zhang, M. Biometric facial identification using attention module optimized YOLOv4 for sheep. Comput. Electron. Agric. 2022. 203,107452. https://doi.org/10.1016/j.compag.2022.107452

[20] Huang, J.; Wang, K.; Hou, Y.; Wang, J. LW-YOLO11: A Lightweight Arbitrary-Oriented Ship Detection Method Based on Improved YOLO11. Sensors 2025, 25, 65.

[21] Jocher, G.; Qiu, J.; Chaurasia, A. Ultralytics YOLO (Version 8.0.0) [Computer Software]. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 9 October 2024).

[22] Meiyun Chen, Min Li, Qianxue Wang, Xiuhua Cao, ADS-YOLO: An enhanced YOLO framework for high-speed MLCCs defect detection, Infrared Physics & Technology, 145, 2025, 105733, 1350-4495. https://doi.org/10.1016/j.infrared.2025.105733.

[23] Hezheng Wang, Jiahui Liu, Jian Zhao, Jianzhong Zhang, Dong Zhao, Precision and speed: LSOD-YOLO for lightweight small object detection, Expert Systems with Applications, 269, 2025, 126440, 0957-4174. https://doi.org/10.1016/j.eswa.2025.126440.

[24] Kotteswaran Venkatesan, Muthunayagam Muthulakshmi, Balaji Prasanalakshmi, Elangovan Karthickeien, Harshini Pabbisetty, Rahayu Syarifah Bahiyah, Comparative analysis of resource-efficient YOLO models for rapid and accurate recognition of intestinal parasitic eggs in stool microscopy, Intelligence-Based Medicine, 11, 2025, 100212, 2666-5212. https://doi.org/10.1016/j.ibmed.2025.100212

[25] Zhang, L.; Sun, Z.; Tao, H.; Wang, M.; Yi, W. Research on Mine-Personnel Helmet Detection Based on Multi-Strategy-Improved YOLOv11. Sensors 2025, 25, 170.

[26] Zhou, M.; Wan, X.; Yang, Y.; Zhang, J.; Li, S.; Zhou, S.; Jiang, X. EBR-YOLO: A Lightweight Detection Method for Non-Motorized Vehicles Based on Drone Aerial Images. Sensors 2025, 25, 196. https://doi.org/10.3390/s25010196

[27] Gautam, D.; Mawardi, Z.; Elliott, L.; Loewensteiner, D.; Whiteside, T.; Brooks, S. Detection of Invasive Species (Siam Weed) Using Drone-Based Imaging and YOLO Deep Learning Model. Remote Sens. 2025, 17, 120. https://doi.org/10.3390/rs17010120

[28] Li, H.; Guo, C.; Yang, Z.; Chai, J.; Shi, Y.; Liu, J.; Zhang, K.; Liu, D.; Xu, Y. Design of field real-time target spraying system based on improved YOLOv5. Front. Plant Sci. 2022, 13, 1072631.

[29] Shao, D.; Liu, Y.; Liu, G.; Wang, N.; Chen, P.; Yu, J.; Liang, G. YOLOv7scb: A Small-Target Object Detection Method for Fire Smoke Inspection. Fire 2025, 8, 62. https://doi.org/10.3390/fire8020062