agriculture

Article

Combining Neural Architecture Search with Knowledge Graphs in Transformer: Advancing Chili Disease Detection

Boyu Xie [1,†], Qi Su [1,†], Beilun Tang [1], Yan Li [1], Zhengwu Yang [1], Jiaoyang Wang [1], Chenxi Wang [1], Jingxian Lin [2] and Lin Li [1,*]

[1] China Agricultural University, Beijing 100083, China; xieboyu@cau.edu.cn (B.X.); suqi2021@cau.edu.cn (Q.S.); tangbl2023@163.com (B.T.); libyan_cau@163.com (Y.L.); yemyoung@cau.edu.cn (Z.Y.); wjy20180926@163.com (J.W.); wangchenxi@cau.edu.cn (C.W.)
[2] School of Computer Science and Engineering, Beihang University, Beijing 100191, China; linjingxian2018@163.com
* Correspondence: lilinlsl0726@cau.edu.cn
† These authors contributed equally to this work.

Abstract: With the advancement in modern agricultural technologies, ensuring crop health and enhancing yield have become paramount. This study aims to address potential shortcomings in the existing chili disease detection methods, particularly the absence of optimized model architecture and in-depth domain knowledge integration. By introducing a neural architecture search (NAS) and knowledge graphs, an attempt is made to bridge this gap, targeting enhanced detection accuracy and robustness. A disease detection model based on the Transformer and knowledge graphs is proposed. Upon evaluating various object detection models on edge computing platforms, it was observed that the dynamic head module surpassed the performance of the multi-head attention mechanism during data processing. The experimental results further indicated that when integrating all the data augmentation methods, the model achieved an optimal mean average precision (mAP) of 0.94. Additionally, the dynamic head module exhibited superior accuracy and recall compared to the traditional multi-head attention mechanism. In conclusion, this research offers a novel perspective and methodology for chili disease detection, with aspirations that the findings will contribute to the further advancement of modern agriculture.

Keywords: chili disease identification; knowledge graphs; Transformers; neural architecture search; focal loss

check for updates

Citation: Xie, B.; Su, Q.; Tang, B.; Li, Y.; Yang, Z.; Wang, J.; Wang, C.; Lin, J.; Li, L. Combining Neural Architecture Search with Knowledge Graphs in Transformer: Advancing Chili Disease Detection. *Agriculture* **2023**, *13*, 2025. https://doi.org/10.3390/agriculture13102025

Academic Editors: Maciej Zaborowicz and Jakub Frankowski

Received: 26 September 2023
Revised: 17 October 2023
Accepted: 17 October 2023
Published: 19 October 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Peppers, as one of the widely cultivated crops globally, not only possess significant economic value but also serve as indispensable ingredients in numerous traditional dishes [1]. However, during the growth process, peppers are vulnerable to various diseases, which can profoundly impact their yield and quality, leading to substantial economic losses for farmers and the entire agricultural supply chain [2].

Conventional crop disease detection primarily relies on agricultural experts' expertise and manual observation [3]. Such methods are time-consuming and inefficient, falling short of meeting the demands for large-scale and real-time disease detection. With the rapid advancements in information technology and computer vision [4–6], newer techniques exhibit strengths in efficiency, accuracy, and scalability, significantly enhancing the accuracy and efficiency of disease detection [7].

Zeng et al. combined convolutional neural networks and transfer learning to detect plant diseases by inspecting plant leaves, achieving an impressive accuracy rate of 99.5% [8]. Li et al. developed an MTC-YOLOv5n model for cucumber disease detection based on YOLOv5, incorporating coordinate attention (CA) and Transformer to reduce distractions and enhance model precision, further lightweighting the model for mobile

deployment [9]. Abbas, Amreen, and colleagues utilized conditional generative adversarial networks (C-GAN) to generate synthetic images of tomato plant leaves, subsequently training a DenseNet model to classify ten types of tomato diseases, achieving an accuracy rate of 97.11% [10]. Sun et al. proposed a real-time lightweight model for apple disease detection, MEAN-SSD, detecting five common apple diseases with an mAP of 83.12% and a speed of 12.53 FPS [11]. To address plant-disease identification in complex field scenarios, Wang et al. introduced a dual-stream hierarchical bilinear pooling model, primarily enhancing network layer information interaction capabilities for fine-grained recognition [12].

Knowledge graph technology, as an emerging method of data organization and representation, offers an intuitive and structured visualization of complex data relations [13]. In agriculture, knowledge graphs can consolidate information and knowledge related to crop growth, diseases, fertilization, and irrigation, offering decision-making support to farmers and agricultural experts and assisting in better crop management and disease prevention [14]. Zhou et al. created a knowledge graph for specific diseases of tomatoes and cucumbers. By integrating image modality, text modality, and knowledge graphs, an ITK-Net crop disease identification model was established, achieving 99.63% accuracy [15]. Zhu et al. addressed fruit-pest problems by first constructing a lychee knowledge graph, then using a VGG-16 model for disease and pest recognition, achieving a 94.9% accuracy rate [16]. Guan et al. constructed an agricultural knowledge graph, then used a CNN-DNN-BiLSTM network for fruit tree pest detection, comparing their results with the VGG network and BiLSTM network, showcasing the superiority of their model over traditional deep learning models [17].

Combining the knowledge graph technology with computer vision for pepper disease detection not only facilitates rapid and accurate disease identification but also offers targeted recommendations and methods for disease treatment and management [18]. For instance, using related information from the knowledge graph, specific fertilization, irrigation, and disease treatment recommendations can be provided to farmers, aiming to prevent and control diseases proactively. Furthermore, by merging computer vision and knowledge graph techniques, predictions on the occurrence, development, and spread trends of diseases can be made, granting more scientific and precise decision-making support for agricultural production and management [19].

Based on the aforementioned discussions, the primary objective of this study is to investigate the roles and impacts of a neural architecture search and knowledge graphs in chili disease detection tasks on model performance. By comparing with baseline models, this research seeks to ascertain whether these two mechanisms can enhance the model's efficacy, thereby introducing a high-precision and rapid method for chili disease detection. A chili disease identification system based on a neural architecture search and knowledge graphs was constructed, leveraging the strengths of both to elevate the efficiency and accuracy of disease detection. The main innovations and contributions are as follows:

1. A neural architecture search is applied to pepper disease image detection for the first time, automatically optimizing the model structure to achieve heightened detection accuracy.
2. A wealth of knowledge about pepper diseases is consolidated using knowledge graphs, enriching the background information and treatment recommendations for the identification results.
3. A novel method of combining Transformer in object detection is introduced and further optimized through the neural architecture search.
4. To capture subtle features in pepper disease images, a dynamic head structure is designed, and an advanced focal loss function is introduced.
5. Comprehensive experimental verification demonstrates the system's superior performance across various hardware platforms.

This research holds practical value for pepper cultivators and provides new research insights and technical references for the disease detection of other crops. Through this study, the aspiration is to propel agricultural disease detection into a more intelligent and accurate new era.

2. Related Work

2.1. Application of Neural Architecture Search in Deep Learning

The core idea of the neural architecture search (NAS) lies in the automated search for the optimal structure of deep learning models. Given the vast model space encompassing thousands of possible combinations, the goal of NAS is to identify the most performant model structure among these combinations [20].

For convolutional neural networks (CNN), which have been extensively applied to image processing tasks with remarkable success [21–23], traditional CNN models such as VGG [24] and ResNet [25] have their structures manually crafted based on researchers' insights. However, as the tasks become increasingly complex, the manual design of network architecture has become more challenging. This is where the potential of NAS is realized. NAS endeavors to explore different combinations of convolutional kernel sizes, layer counts, and connection strategies to automatically discover the most fitting CNN architecture for specific tasks [20]. The fundamental optimization problem for NAS can be expressed as [20]

$$\arg\min_{\alpha} \mathcal{L}(f(w^*(\alpha), \alpha); \mathcal{D}_{val}), \tag{1}$$

where α denotes the network structure parameters, $w^*(\alpha)$ represents the weights given the network structure parameters α, $\mathcal{L}$ is the loss function, and $\mathcal{D}_{val}$ stands for the validation set.

On the other hand, due to its self-attention mechanism, the Transformer model has shown superior performance on sequence data and has been widely adopted for natural language processing tasks [26,27]. Similar to CNNs, the structure of Transformer models can also be optimized using NAS. In NAS for Transformers, common alterations include the number of attention heads, model depth, and feed-forward neural network dimensions. For instance, through NAS, a more compact Transformer model can be discovered that maintains a performance close to the original model while significantly reducing computational requirements. The optimization problem can also be represented as in Equation (1).

In summary, NAS provides an effective method for automatically optimizing deep learning models such as CNNs and Transformers. By facilitating automated search processes, not only can NAS identify high-performing model architectures, but it can also save significant time and effort for researchers. With the further development of NAS techniques, it is anticipated that more high-performance, computationally efficient deep learning models will emerge.

2.2. Application of Knowledge Graphs in Agricultural Tasks

Knowledge graphs, as structured knowledge organization methods, have been increasingly recognized in agricultural tasks [28]. The knowledge ecosystem in agriculture is intricate, encompassing soil types, climatic conditions, crop varieties, and pest species. The strategic combination of this information determines the ultimate result of agricultural production. Knowledge graphs can structure and visualize this data, providing potent decision support for agricultural production and research.

Consider the core agricultural task of disease prediction. Traditional methods [3] largely rely on empirical knowledge, whereas knowledge graphs integrate multi-faceted data, such as historical records, soil testing outcomes, and weather forecasts, offering a more precise model for disease prediction.

2.2.1. Data Annotation Process

The data related to diseases are first gathered from various sources, potentially including reports from agricultural departments, research papers from experts, and field experiment data. These datasets are then subjected to preprocessing tasks, such as data cleaning and format conversion. Subsequently, with the aid of expert knowledge and semi-automated tools, these datasets are annotated to establish relationships (e.g., causality or correlation) between various factors, such as soil type or climatic conditions and diseases.

2.2.2. Model Input

When constructing knowledge graph models, inputs mainly comprise numerical or categorical information of various factors, such as soil type (sandy, clay, or loamy) and climatic conditions (temperature, humidity, rainfall, etc.). Moreover, historical records of disease occurrences, such as the incidence rate or disease type from the previous quarter, can also be integrated.

2.2.3. Model Output

The model output predominantly pertains to predictions related to disease occurrences, which include the likelihood of the disease manifesting, potential disease types, and probabilities associated with each type. These outputs can offer farmers targeted preventive and treatment recommendations.

Mathematically, the construction of a knowledge graph can be perceived as a graph model, where nodes represent various factors or diseases, and edges signify their relationships. For disease prediction, a probabilistic model, such as a Bayesian network, can be devised to depict the probabilistic relationships between various factors and diseases. Specifically, given the observed values of factors x, the probability of disease occurrence can be expressed as

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}, \tag{2}$$

where y represents the event of disease occurrence, $P(x|y)$ denotes the probability of observing factor x given the disease occurrence, $P(y)$ is the prior probability of disease occurrence, and $P(x)$ is the marginal probability of factor x.

To conclude, knowledge graphs have found broad applications in agricultural tasks [29–31], especially in disease prediction. By organizing and integrating diverse information in a structured manner, knowledge graphs not only enhance the accuracy of disease prediction but also deliver robust decision support for agricultural production and research.

3. Materials

3.1. Data Entry for Knowledge Graphs

Knowledge graphs have demonstrated their immense value in various tasks within the current AI research, especially in the identification of chili pepper diseases [29], where they can provide rich semantic background knowledge. Detailed below is the methodology employed to construct and utilize the knowledge graph to aid in disease detection from image datasets.

3.1.1. Knowledge Graph Construction

Initially, the core entities of the knowledge graph were determined, including "Disease", "Pathogen", "Affected Part", and "Treatment Method". These entities are vital factors in disease identification and treatment. Each entity possesses associated attributes, such as the "Name", "Incubation Period", and "Typical Symptoms" of a disease. Subsequently, relationships between these entities were established. For instance, a "Pathogen" might "Cause" a certain "Disease", and a "Disease" might "Affect" a certain "Part", as illustrated in Figure 1.

$$\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{A}\} \tag{3}$$

where $\mathcal{E}$ represents the set of entities, $\mathcal{R}$ denotes the set of relationships, and $\mathcal{A}$ stands for the set of attributes.

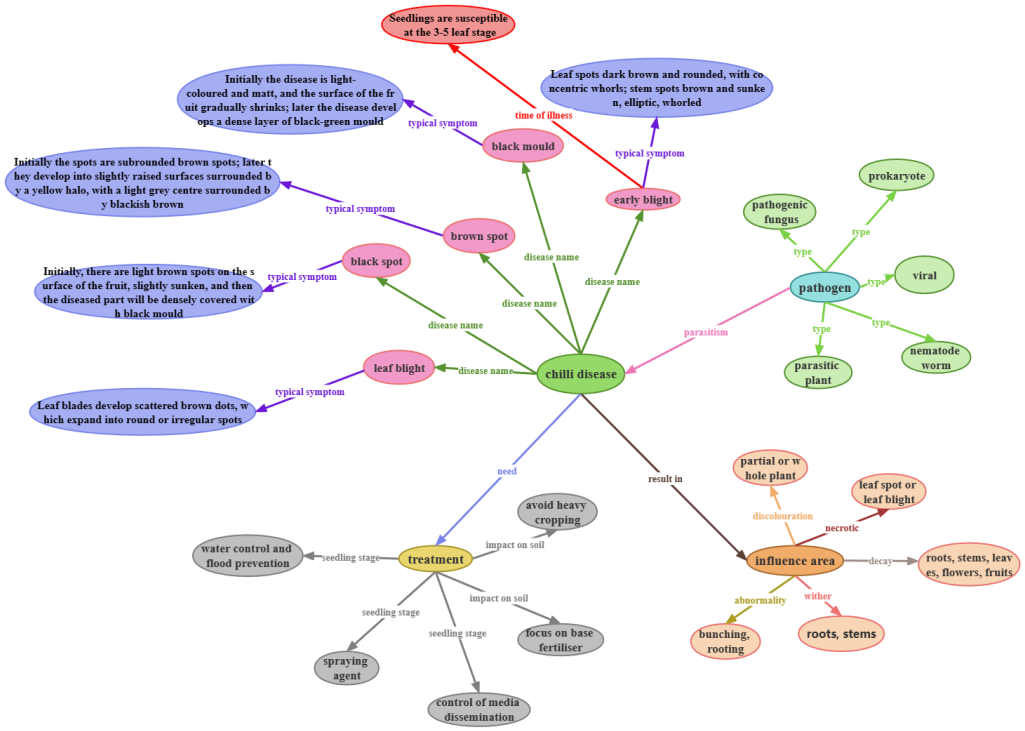

Figure 1. Illustration of the knowledge graphs generated in this paper. Leaf blight: This can be caused by various pathogens in different crops, but in chili peppers, it might be due to Alternaria solani or Phytophthora capsici. Black spot: Black spot is typically associated with roses and is caused by the fungus Diplocarpon rosae. In chili peppers, a disease with similar symptoms might be caused by a different pathogen, so it is important to accurately diagnose the disease. Brown spot: This could refer to bacterial leaf spot caused by Xanthomonas campestris pv. vesicatoria in chili peppers. Black mold: This is usually referring to the sooty mold that grows on the honeydew produced by insects. Early blight: This is typically caused by the fungus Alternaria solani.

3.1.2. Knowledge Graph Application

The knowledge graph provides not only detailed information about chili pepper diseases but also equips the model with semantic background knowledge. When a suspected disease region is detected in an image by the model, this region is associated with the disease entity in the knowledge graph to gather more information about that disease. For instance, upon detecting a disease, information such as its typical symptoms, potential pathogens, affected parts, and recommended treatment methods can be retrieved from the knowledge graph. To realize this functionality, a mapping function M was defined that takes the model's output and associates it with the knowledge graph [28]:

$$\mathcal{I} = M(\mathcal{O}, \mathcal{G}) \tag{4}$$

where $\mathcal{O}$ is the model's output, $\mathcal{G}$ is the knowledge graph, and $\mathcal{I}$ is the information associated with the knowledge graph.

3.1.3. Adapting Image Datasets

During model training on the image dataset, both image annotations and information from the knowledge graph were utilized as auxiliary inputs. Specifically, for each image, the disease entities and attributes related to them were retrieved and input into the model along with the image. To facilitate this functionality, an input function I was defined,

which takes the image data and its related knowledge graph information, generating the model's input:

$$x' = I(x, \mathcal{I}) \tag{5}$$

where x is the image data, $\mathcal{I}$ is the information related to the knowledge graph, and x' is the model's input. In conclusion, the knowledge graph plays a pivotal role in chili pepper disease identification. It enriches the model with semantic background knowledge and amplifies the model's inference capabilities.

3.2. Image Dataset Collection and Annotation

For the training of the chili pepper disease identification model, a substantial amount of annotated image data were necessary. Initially, a plethora of chili images were gathered from multiple online agricultural databases, as shown in Table 1. These images covered different growth stages, lighting conditions, and shooting angles, ensuring data diversity, as showcased in Figure 2.

Table 1. Distribution of the images in the dataset used in this paper.

Kind	Number of Images before Augmentation	Number of Images after Augmentation
Black Mold	331	852
Brown Spot	219	719
Black Spot	486	1046
Leaf Blight	173	683
Early Blight	320	905



Figure 2. Samples of chili pepper disease images dataset in this paper. (Leaf blight, black spot, brown spot, black mold, early blight.)

Following data collection, a team comprising agricultural experts and data annotators was assembled. Using annotation tools, they annotated each image for the location and category of diseases. Each disease region was represented with a bounding box, accompanied by a specific disease name, as illustrated in Figure 3.

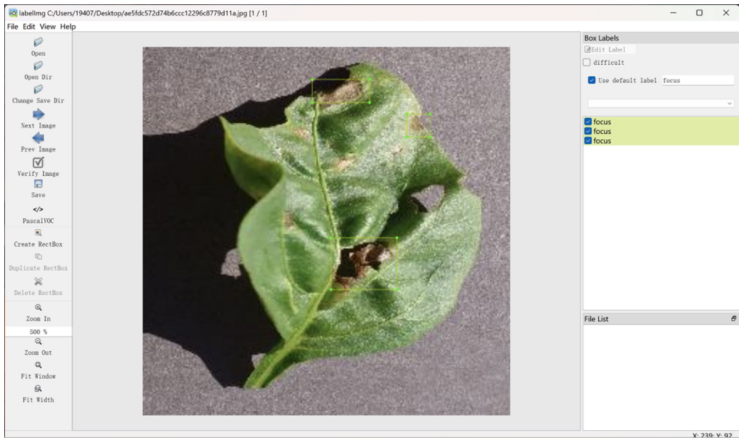

Figure 3. Illustration of the annotation screenshot by LabelImg [32] application.

Mathematically, this representation can be expressed as

$$D = \{(x_i, y_i)\}_{i=1}^{N} \tag{6}$$

where x_i is the i-th image, y_i denotes its corresponding disease annotation, inclusive of bounding boxes and category labels, and N is the total number of images.

3.3. Dataset Augmentation

Data augmentation is a common technique in deep learning, allowing for increased data diversity without actually expanding the dataset, thereby enhancing the model's generalization capabilities. Given the characteristics of agricultural images, such as variations in lighting and obstructions, it was decided to employ data augmentation to bolster the model's robustness against these factors. Several augmentation techniques were applied, including random cropping, rotation, scaling, brightness, and contrast adjustments, as depicted in Figure 4.

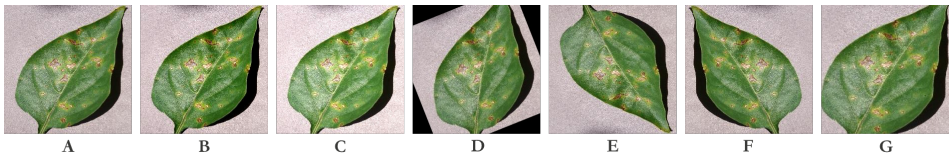

Figure 4. Demo of different augmentation methods used in this paper. (**A**) is the original image; (**B**) is contract augmentation; (**C**) is brightness augmentation; (**D**) is rotation augmentation; (**E**) is flipping vertically augmentation; (**F**) is flipping horizontally augmentation; (**G**) is cropping augmentation.

Mathematically, given an image x, an augmentation function $\mathcal{T}$ was defined that takes image x and produces the augmented image x':

$$x' = \mathcal{T}(x) \tag{7}$$

Through data augmentation, a vast number of images slightly different from the original yet retaining the same semantic essence can be generated. This not only amplifies the volume of training data but also aids the model in learning more robust features, thereby enhancing its performance in real-world scenarios. In summary, through the construction of the knowledge graph and the collection, annotation, and augmentation of datasets, a rich and diverse training dataset was provided for the task of chili pepper disease identification. This laid a solid foundation for the training and evaluation of the model, ensuring commendable results in practical applications.

4. Proposed Method

4.1. Overall

A comprehensive framework that integrates both the Transformer and knowledge graph models is proposed in this study, aiming for efficient and accurate detection of chili pepper diseases. To fully exploit both the image data of chili pepper diseases and the related knowledge information, a two-stage model, has been designed, as depicted in Figure 5.

Initially, image features of chili pepper diseases are extracted using the Transformer model after a CNN module, as shown in Figure 5. Subsequently, by incorporating the knowledge graph model, related knowledge information is integrated to provide a more comprehensive and precise decision support for disease identification. The input to the model is twofold: the first being the image data of chilies, encompassing images of both healthy and diseased peppers; the second pertains to knowledge information related to diseases, which might encompass aspects such as disease types, pathogens, influencing factors, and mechanisms of disease onset. The model output is the identification result of the chili pepper diseases, covering disease type, the likelihood of occurrence, and related knowledge information. These outputs can offer farmers targeted prevention and treatment recommendations.

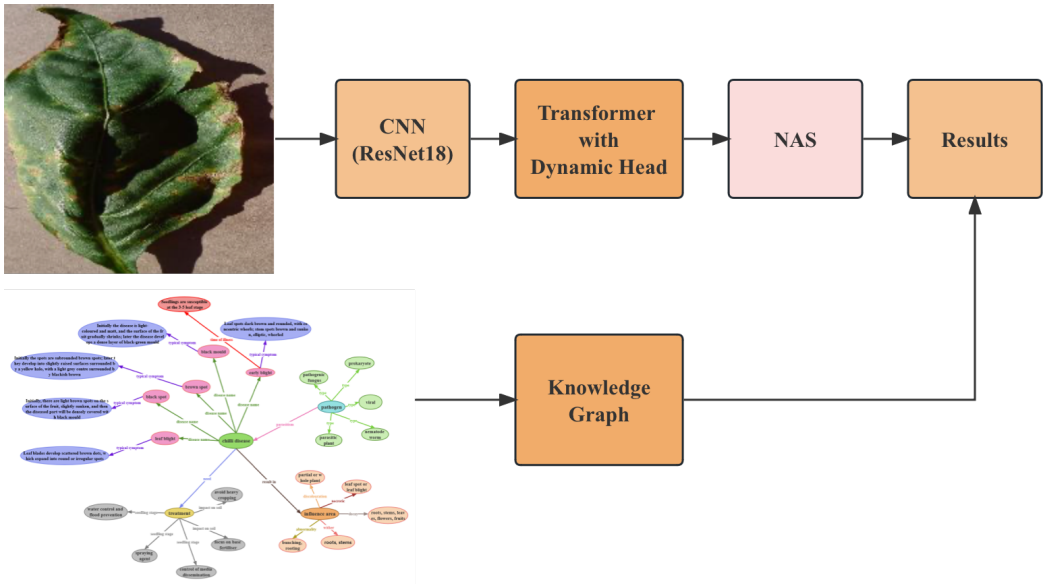

Figure 5. Illustration of the whole method proposed in this paper.

4.2. NAS for Performance Optimization

To further enhance the performance of the model, neural architecture search (NAS) technology is applied for the structural search of the entire model, as shown in Figure 6. Specifically, a search space is first defined, encompassing various potential structural configurations of the Transformer model, such as its depth, number of attention heads, and dimensions of the feed-forward neural network. Then, using NAS, the model structure best suited for the chili pepper disease detection task is autonomously sought. Furthermore, to integrate the knowledge graph model, structural configurations related to knowledge information, such as the embedding methods for knowledge nodes and the association methods between knowledge and features, are also incorporated into the search space. Through the autonomous search with NAS, not only can the most performant model structure be found, but the optimal way to integrate knowledge information with features can also be determined.

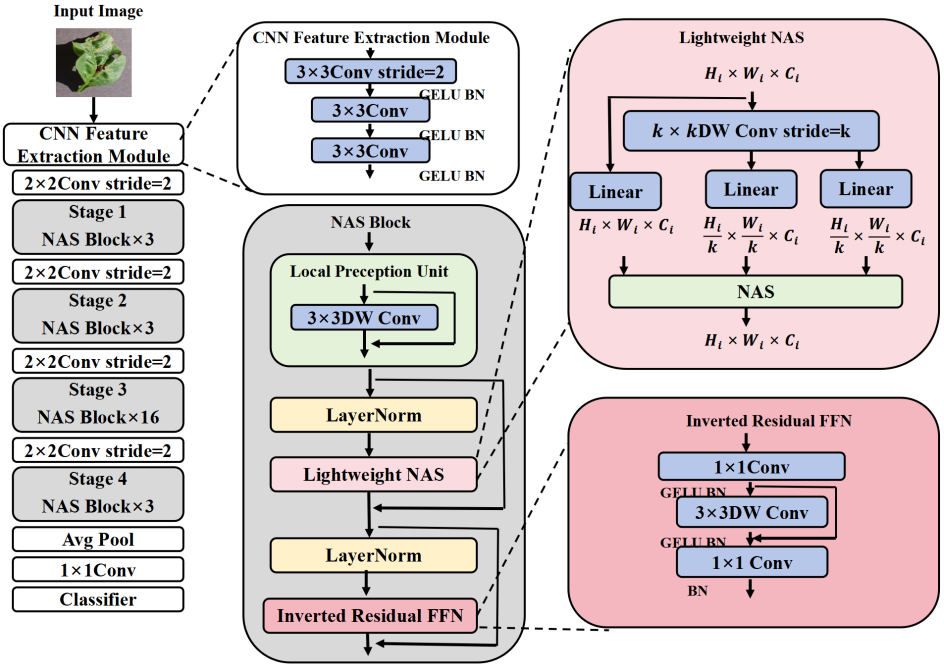

Figure 6. Illustration of the NAS module used in this paper. GELU [33] means Gaussian error linear units. The NAS block details are shown in the gray bounding box.

In summary, the method proposed in this study is a comprehensive framework that integrates both the Transformer and knowledge graph models. Through its two-phase model design, it harnesses both the image data of chili pepper diseases and the related knowledge information, providing comprehensive and precise decision support for disease identification. Moreover, by employing NAS technology, the model's performance is further optimized, adapting it more closely to the characteristics and demands of the chili pepper disease detection task.

4.3. Integration of Knowledge Graph

The value of knowledge graphs in various AI applications is gradually gaining recognition among researchers [30,31]. In the task of chili pepper disease identification, the knowledge graph can equip the model with rich prior knowledge and background information, aiding the model in better understanding and identifying diseases. The input for this module is the raw features from the object detection model and the entity and relationship information related to chilies in the knowledge graph, as shown in Figure 1. The output is the enhanced features after integrating with the knowledge graph, as shown in Figure 1.

Entities and relationships related to chilies are first extracted from the knowledge graph to build a disease-attribute subgraph. Then, a graph neural network (GNN) [29] is used to encode this subgraph, obtaining the embedding representation for each disease. Mathematically, this process can be represented as

$$h_v^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} W^{(l)} h_u^{(l)}\right), \tag{8}$$

where $h_v^{(l)}$ denotes the embedding of node v at layer l, $\mathcal{N}(v)$ represents the set of neighbors of node v, $W^{(l)}$ is the weight matrix at layer l, and σ is an activation function. Subsequently, the obtained disease embeddings are fused with the raw features from the object detection model through a fully connected layer, mathematically expressed as

$$f' = \mathrm{ReLU}(W_f f + b_f + h), \tag{9}$$

where f is the raw feature, h is the disease embedding, and W_f and b_f are the weight matrix and bias, respectively. The design of this fusion module is driven by the intent to leverage prior knowledge and background information from the knowledge graph to enhance the model's comprehension capability. Conventional object detection models only learn features from images and lack a deep understanding of the reasons behind and impacts of the diseases. The knowledge graph, on the other hand, can provide the model with this invaluable information, aiding the model in better distinguishing between various diseases, thereby enhancing identification accuracy. It not only equips the model with rich information from the knowledge graph, enhancing its understanding of the diseases, but also introduces a novel, more potent feature representation method, allowing the model to learn features not just from images but also from the knowledge graph. Finally, integrating the knowledge graph provides a more stable and robust feature representation, ensuring the model's robust performance even in the face of noisy or incomplete data.

4.4. Transformer in Focus Detection Task

The recent computer vision research has extensively focused on the Transformer model due to its unique self-attention mechanism. Initially, Transformers were designed for handling natural language, aiming to capture long-range dependencies in texts. However, it was discovered by researchers that its self-attention property is also highly suited for image processing tasks, especially in scenarios that necessitate capturing long-distance relationships between different parts of an image. In the chili pepper disease identification task presented in this study, morphological features of diseases can appear anywhere in the image, and there might exist correlations or structural dependencies between these

locations. For instance, the onset of a disease on one side of the chili could imply the emergence of disease symptoms on the opposite side. Traditional CNN models, focusing primarily on local features, might miss such global, long-distance dependencies. This is where the Transformer's uniqueness in object detection, especially in chili pepper disease identification, comes into play.

In the design of this study, a CNN, ResNet18 [25], is initially utilized to extract basic image features, which are then fed as inputs to the Transformer module [26], as depicted in Figure 7.

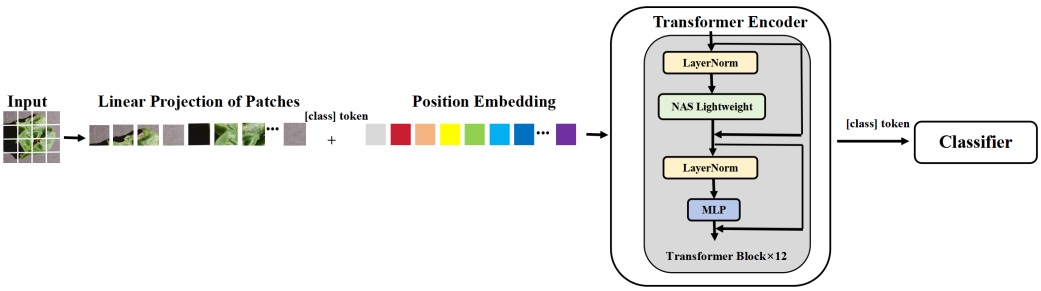

Figure 7. Illustration of the Transformer architecture used in our model.

The Transformer module [26] converts the feature map into a sequential format, with each pixel point acting as an element in the sequence. These elements undergo processing via the self-attention mechanism, resulting in new feature representations. These new features not only amalgamate local information but also integrate global, long-range information. Specifically, the self-attention mechanism in the Transformer can be mathematically represented as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{10}$$

where Q, K, and V are the query, key, and value derived from the input features via linear transformation, respectively, and d_k denotes the dimension of the key. Adopting this design, which combines the local feature extraction capabilities of CNN and the long-distance relationship capturing abilities of the Transformer, allows for a more accurate identification of chili disease features. Furthermore, applying NAS on this structure can further optimize the model, automatically searching for the network configuration most suited for chili disease identification, thereby enhancing the accuracy of identification.

4.4.1. Dynamic Head for Tiny Focus Feature

In object detection, especially in detecting tiny focus features, the conventional multi-head self-attention [26] used in Transformer models might encounter certain limitations. To more precisely capture these small yet pivotal features, a design called "Dynamic Head" is proposed, intended to replace the original multi-head attention mechanism, as shown in Figure 8.

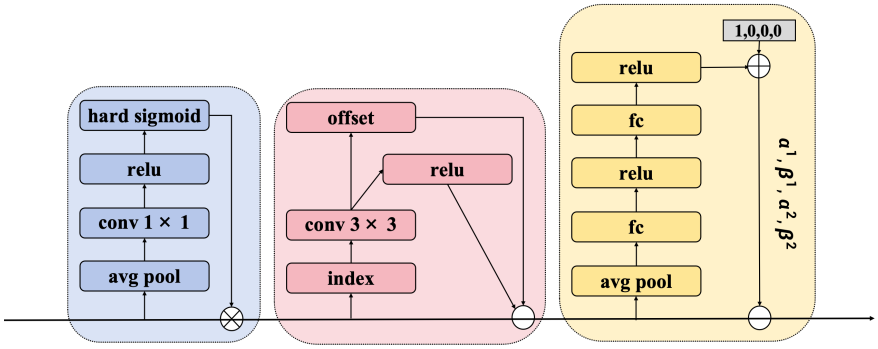

Figure 8. Illustration of the dynamic attention head proposed in this paper. The hard sigmoid is a simplified version of the sigmoid function, which accelerates computation through linear approximation.

Traditional multi-head self-attention aims to allow the model to simultaneously capture multiple different feature relationships. Specifically, each "head" independently executes self-attention operations, thereby focusing on different parts of the input features. This can be mathematically expressed as

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_n)W^O, \quad (11)$$

where each head_i represents an independent self-attention operation as previously described, and W^O is the output linear transformation matrix. However, in the dynamic head design, instead of using a fixed number of "heads" for the self-attention operations, the number and weights of the "heads" are dynamically adjusted based on the content of the input features. Specifically, a weight coefficient is introduced for each "head", which adjusts dynamically based on the input features, allowing some "heads" to have higher weights when dealing with tiny focus features. Mathematically, the dynamic head can be expressed as

$$\text{DynamicHead}(Q, K, V) = \text{Concat}(\alpha_1 \times \text{head}_1, \alpha_2 \times \text{head}_2, \ldots, \alpha_n \times \text{head}_n)W^O, \quad (12)$$

where α_i is the weight coefficient for the ith "head", and it functions based on the input features. The design rationale behind the dynamic head originates from the observation that not all "heads" are equally important when processing tiny focus features; some "heads" might be more adept at capturing such features while others might overlook them. By introducing dynamic weights, the model can prioritize the more relevant "heads", achieving a more accurate capture of tiny focus features. Compared to the traditional multi-head self-attention, the dynamic head offers the following advantages:

1. Precise feature capture: Through dynamic weights, the model can place greater emphasis on those "heads" that are beneficial for capturing tiny focus features, thereby improving identification accuracy.
2. Enhanced model flexibility: The dynamic head is not limited to the detection of tiny focus features but is also applicable to other types of object detection tasks. This is because it can dynamically adjust the "head" weights based on input feature content, making the model more adaptive to the current task.

In conclusion, the dynamic head offers a novel and more potent feature extraction mechanism for chili pepper disease identification. It is believed that through this design, the model's accuracy and robustness can be further enhanced.

4.4.2. Advanced Focal Loss Function

In object detection tasks, especially with imbalanced data distributions, the classic cross-entropy loss may lead to a model preference for frequently occurring background classes at the expense of minority target classes, such as chili diseases [34]. To address this issue, an advanced focal loss function is proposed in this study. The original design of the focal loss function was intended to increase the weight of samples misclassified by the model, ensuring that these samples receive greater attention during training. It is mathematically defined as

$$\text{FL}(p_t) = -(1 - p_t)^\gamma \log(p_t), \quad (13)$$

where p_t is the model's predicted probability for the positive class, and γ is a tuning parameter used to control the rate of weight increase. However, the original focal loss function might still fail to capture some critical, hard-to-classify samples in certain cases. To further accentuate the model's focus on these samples, an advanced version of the focal loss function is introduced. A new parameter, α, is incorporated into the original focal loss function, combined with the sample's class imbalance. The specific form is

$$\text{AFL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t), \quad (14)$$

where α_t is a coefficient related to the sample's class distribution, used to further enhance the weight of hard-to-classify samples. Such a design was chosen since in imbalanced data distributions, the hard-to-classify samples tend to be key and valuable. By introducing α_t, the model can place more emphasis on these samples during training, thereby improving the model's generalization capability. Mathematically, α_t can be defined as

$$\alpha_t = \frac{N_{\text{neg}}}{N_{\text{pos}} + N_{\text{neg}}}, \tag{15}$$

where N_{pos} and N_{neg} are the number of positive and negative samples, respectively. Compared to the original focal loss function, the advanced focal loss function offers the following advantages:

1. Enhanced ability to handle class imbalance: By introducing α_t, the weight of hard-to-classify samples can be further emphasized, ensuring that the model focuses more on these samples during training.
2. Improved generalization capability: In imbalanced data distributions, hard-to-classify samples are often key and valuable. By utilizing the advanced focal loss function, the model's focus on these samples during training can be accentuated, thereby enhancing its generalization capability.
3. Greater flexibility in loss adjustment: Compared to the original focal loss function, the advanced focal loss introduces a new parameter, α_t, providing flexibility to adjust the loss function according to specific task requirements, leading to improved training outcomes.

In summary, the advanced focal loss function offers a novel and more potent loss design for chili disease identification. It is believed that through this design, the accuracy and robustness of the model can be further enhanced.

4.5. Experiment Design

To provide a comprehensive and objective assessment of the proposed chili disease identification system, which is based on neural architecture search and knowledge graphs, a series of experiments have been designed. The subsequent sections detail the experimental design.

4.5.1. Experiment Platform

In this research, all experiments were conducted on the Linux operating system platform. To ensure efficient code execution and rapid model development, Python was chosen as the primary development language, owing to its extensive use in the fields of data science and machine learning. To build and test our model, we utilized several popular Python libraries. First, we employed the PyTorch library, version 1.8.0, which is an open-source deep learning framework that offers flexible and efficient model training and evaluation capabilities. Additionally, for data processing and analysis, we used NumPy (version 1.19.5) and Pandas (version 1.2.3), both of which provide a plethora of handy tools and functions. For the visualization of our model and the presentation of results, we employed Matplotlib (version 3.4.1) to generate high-quality graphics.

4.5.2. Dataset Partition and Baseline

The chili disease dataset was initially partitioned. Adhering to conventional data-splitting principles and aiming to ensure training stability, the dataset was divided into training, validation, and test sets at a ratio of 8:1:1. The training set is utilized for model training and parameter updates, the validation set for performance validation and hyperparameter tuning, and the test set for the final evaluation of the model performance. To comprehensively evaluate the proposed model, several popular models in the object detection domain were chosen as baselines, including YOLOv5 [35], YOLOv8 [36], DETR [37], SSD [38], and EfficientDet [39]. These models have demonstrated remarkable performance in object detection tasks, thereby serving as suitable performance benchmarks. Notably,

these models span various technological trends from real-time detection using the YOLO series to the Transformer-based DETR, offering a comprehensive perspective to evaluate the proposed approach.

4.5.3. Optimizer Selection and Hyperparameter Settings

The choice of optimizer plays a crucial role in influencing the training speed and final performance of the model. In these experiments, the Adam [40] optimizer was chosen due to its ability to adaptively adjust learning rates and its proven effectiveness across various tasks, combining the advantages of Momentum and RMSProp. The selection of hyperparameters is also a pivotal aspect of experimental design. Both grid search and random search strategies were employed to find the optimal hyperparameter combination, with various combinations validated on the validation set. Ultimately, a learning rate of 0.001, batch size of 32, and weight decay of 0.0005 were selected, as they exhibited the best performance on the validation set.

4.5.4. Ablation Study Design

To further validate the effectiveness and significance of each component within the proposed model, a series of ablation studies were conducted. These studies included variations such as: a model without the use of knowledge graphs; a model without the use of neural architecture search; the application of the original multi-head attention mechanism instead of the introduced dynamic head; and the utilization of cross-entropy loss instead of the advanced focal loss function proposed. Through these ablation studies, a deeper understanding of the role of each component within the model can be obtained, offering valuable insights for further research and improvements.

4.5.5. Experiment Metric

To objectively and comprehensively evaluate the performance of the proposed chili disease identification system, multiple evaluation metrics were employed, including precision, recall, mAP, and FPS [7].

1. Precision
   Precision measures the proportion of positive predictions that are actually correct. It reflects the accuracy of the model's predictions, indicating how many of the predicted positive samples are true positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{16}$$

   Here, TP represents the number of true positives, while FP indicates false positives. For chili disease identification, a high precision implies that the model has a low rate of false alarms when identifying diseases.
2. Recall
   Recall indicates the proportion of actual positive samples that are correctly predicted. It captures the model's capability to retrieve relevant instances, revealing how many of all positive samples are accurately predicted by the model.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

   In this equation, FN denotes the number of false negatives. Recall is particularly important for chili disease identification as a high recall ensures that most diseases are detected, mitigating potential agricultural losses.
3. Mean Average Precision (mAP)
   mAP computes the average precision at varying levels of recall, commonly employed in object detection tasks. For each recall level, precision is calculated, and then an average of these precisions is taken.

$$\text{mAP} = \frac{1}{|\mathcal{R}|} \int_{r \in \mathcal{R}} P(r)\, dr \tag{18}$$

Here, $\mathcal{R}$ represents the set of recall values, and $P(r)$ indicates the precision at recall level r. mAP offers a holistic measure of the model's accuracy and recall capabilities.

4. Frames Per Second (FPS)
FPS serves as an indicator of the model's real-time capability, denoting the number of frames the model can process per second.

$$\text{FPS} = \frac{1}{\text{Time per frame}} \tag{19}$$

For chili disease identification, a high FPS suggests that the model can swiftly process images, offering timely disease detection results in practical applications.

In conclusion, the evaluation metrics employed in this study encompass the model's accuracy, recall capabilities, and real-time processing ability, providing a comprehensive and objective assessment standard. Particularly for chili disease identification, high accuracy and recall ensure timely and precise disease detection, while a high FPS guarantees real-time application, thus, offering farmers timely and effective disease control recommendations.

5. Results and Discussion

In machine learning and computer vision research, the evaluation and comparison of models serve as pivotal components for assessing their efficacy and robustness. The primary aim of this section is to evaluate and compare the performance of various object detection models on chili disease identification tasks, thereby offering theoretical and empirical foundations for practical applications. By employing consistent evaluation metrics—precision, recall, and mAP—an unbiased and objective assessment of the strengths and weaknesses of each model is achieved.

5.1. Detection Results

The experimental results from this study reveal that the model developed in this research achieved the best scores on the evaluation metrics of precision, recall, and mAP, being 0.95, 0.91, and 0.94, respectively, outperforming other baseline models, as shown in Tables 2 and 3.

Table 2. Detection results of different models.

Model	Precision	Recall	mAP
YOLOv5 [35]	0.89	0.87	0.88
YOLOv8 [36]	0.88	0.86	0.87
SSD [38]	0.86	0.83	0.85
DETR [37]	0.90	0.88	0.89
EfficientDet [39]	0.87	0.85	0.86
Ours	0.95	0.91	0.94

Table 3. Detection results of different chili disease types using our model.

Kind	Precision	Recall	mAP
Black Mold	0.93	0.90	0.92
Brown Spot	0.93	0.88	0.91
Black Spot	0.95	0.92	0.95
Leaf Blight	0.97	0.92	0.96
Early Blight	0.97	0.92	0.96

Among them, DETR demonstrated commendable performance, second only to the proposed model. This success can be attributed to its design based on the Transformer architecture, which is adept at capturing the contextual information of images, providing

a more comprehensive perspective for chili disease detection. The YOLO series models (YOLOv5 and YOLOv8) also showcased noteworthy performance by predicting bounding boxes and categories in a single forward pass, ensuring the model's real-time capability and accuracy. However, the SSD model exhibited the most modest performance among all, possibly due to its inadequate handling of small or highly overlapping objects. EfficientDet, being a model focusing on efficiency, still holds considerable application value in scenarios with limited computational resources, despite its slightly inferior performance compared to the others. In summary, integrating knowledge graphs with deep learning techniques led to significant improvements in chili disease identification, and emerging detection models such as DETR also showcased vast potential.

5.2. Test on Different Edge-Platform

The objective of this experiment was to evaluate and compare the real-time performance of various object detection models across multiple edge computing platforms. These platforms encompassed common smartphone models such as Huawei P40 and iPhone 13, as well as microcomputers such as the Jetson Nano and Raspberry Pi. By assessing the frames per second (FPS) performance of these models on different hardware, insights were gained into the potential real-world performance of each model, especially in resource-constrained scenarios. The results are presented in Table 4.

Table 4. FPS comparison of different detection models on different hardware platform. Generally, we believe that if a model achieves a processing speed of 30FPS, it can be considered to have met the requirements for real-time monitoring [3]. On the Huawei P40, the implementation of the model in this paper was achieved using AI-related API interfaces provided by Google, with Java being the development language. On the iPhone, the development was carried out in the Swift language using Apple's Xcode software 14.0, as shown in Figure 9. For both the Jetson Nano and Raspberry Pi, since they run on a Linux system, their implementation is the same as that on servers. Both are developed using the Python language based on the PyTorch framework.

Model	Huawei P40	Jetson Nano	Raspberry Pi	iPhone 13
YOLOv5 [35]	31	49	11	28
YOLOv8 [36]	29	44	12	29
SSD [38]	13	27	-	-
DETR [37]	3	15	-	-
EfficientDet [39]	28	45	13	19
Ours	33	58	13	31

These findings are reflective of the inherent design nuances and optimization levels of each model. The YOLO series, renowned for its streamlined design and efficient forward computation, ensured commendable real-time performance across platforms. EfficientDet, on the other hand, sought a balance between efficiency and accuracy, rendering it slightly less optimal in certain resource-limited environments. While SSD aimed for efficiency during its design phase, it failed to match the expectations on some edge devices, possibly due to its multi-scale features and intricate default bounding box computations. As for DETR, its Transformer-based design excelled in capturing contextual information from images but at the cost of increased computational complexity, leading to a pronounced reduction in FPS on some edge devices.

Delving deeper into their mathematical constructs, the YOLO series fundamentally simplifies object detection to a regression problem, thereby eliminating the complexities of multi-stage computations and ensuring exemplary FPS. DETR, equipped with a Transformer structure, encompasses extensive matrix computations and self-attention mechanisms. This computational burden is particularly pronounced on edge devices, resulting in compromised performance. EfficientDet, meanwhile, endeavors to strike a balance between model size and computational intricacies. While it might occasionally fall short of YOLO's performance, its utility remains in resource-constrained settings.

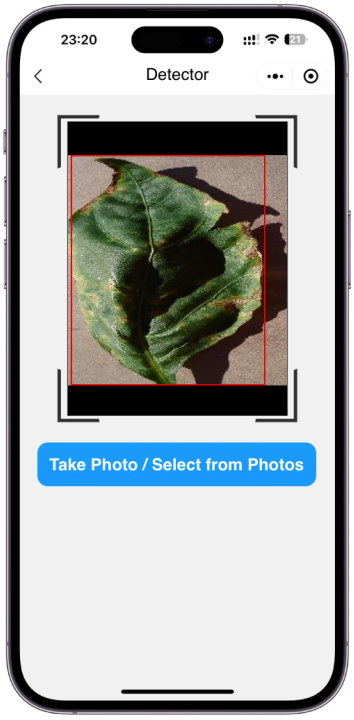

Figure 9. Our method running on iPhone.

In conclusion, the varied performance of these models on diverse edge computing platforms stems from their unique design philosophies and underlying mathematical constructs. These insights offer valuable guidance for future research, aiding researchers and engineers in judicious model selection and optimization to cater to practical application needs.

5.3. Ablation Study on Different Dataset Augmentation Methods

The primary objective of this experiment was to evaluate and compare the effects of various dataset augmentation techniques on model performance. Dataset augmentation is commonly employed to expand the training set, enhance model generalization, and mitigate issues arising from insufficient data or overfitting. By systematically applying and combining various augmentation methods, such as flipping, cropping, resizing, and brightness adjustment, insights into their specific impacts on model performance can be gained. This paves the way for determining the optimal augmentation strategies for practical applications. The experimental results are presented in Table 5.

Table 5. Ablation experiment results of different dataset augmentation methods on our model.

Flipping	Cropping	Resize	Brightness	mAP
-	-	-	-	0.88
-	✓	✓	✓	0.93
✓	-	✓	✓	0.89
✓	✓	-	✓	0.93
✓	✓	✓	-	0.91
✓	✓	✓	✓	0.94

From the results table, it can be discerned that different augmentation techniques exert distinct effects on model performance. Specifically:

1. Without any augmentation, the model achieved an mAP of 0.88.
2. Employing solely cropping, resizing, and brightness adjustment, the model's mAP rose to 0.93, indicating that these three techniques significantly bolstered the model's performance.

3. With only flipping, resizing, and brightness adjustment, the mAP reached 0.89. This score, slightly higher than without any augmentation, is nonetheless inferior compared to the effect of cropping, suggesting that flipping might not always be as effective as cropping in certain contexts.
4. Retaining all methods but resizing, the mAP still remained at 0.93, possibly implying that, in the presence of other augmentations, the impact of resizing becomes less pronounced.
5. When excluding brightness adjustment but maintaining other methods, the mAP was 0.91, highlighting the contribution of brightness adjustment to model performance.
6. Integrating all augmentation techniques, the model achieved its best mAP of 0.94.

These findings underscore the pivotal role of data augmentation in enhancing model performance, particularly the cropping, resizing, and brightness adjustment methods, which seem to have a more pronounced effect on performance.

5.4. Ablation Study on Different Loss Functions

The primary objective of this study was to evaluate and compare the impact of different loss functions on model performance. Loss functions serve as a pivotal component in machine learning and deep learning training, determining how the parameters of the model are optimized and how features are learned during the training process. By contrasting various loss functions, insights can be garnered regarding their distinct roles and effects during model training, offering theoretical guidance for model selection and optimization in practical applications.

From the experimental data presented in Table 6, it can be inferred that varying loss functions considerably influence model performance. Specifically, the advanced focal loss outperforms in all metrics, achieving an mAP of 0.94, underscoring its efficacy for this task. The original focal loss also delivers commendable results with an mAP of 0.91, largely attributed to its design purpose of addressing class imbalance issues. Conversely, AP loss and DR loss present closely aligned performance, albeit marginally trailing behind focal loss, with mAPs of 0.88 and 0.87, respectively. Analytically, the focal loss accentuates the optimization of the model by emphasizing harder-to-classify samples, proving particularly advantageous for tasks potentially grappling with class imbalances. While AP Loss and DR Loss might exhibit promising results for certain tasks, they seem less effective than focal loss for this specific endeavor.

Table 6. Ablation experiment results of different loss functions on the proposed method.

Loss Function	Precision	Recall	mAP
AP Loss [41]	0.91	0.86	0.88
DR Loss [42]	0.90	0.85	0.87
Focal Loss [34]	0.93	0.87	0.91
Advanced Focal Loss	0.95	0.91	0.94

Mathematically dissecting the models reveals that the AP loss, a probability-based loss function, predominantly focuses on differentiation between positive and negative cases. It might render satisfactory outcomes when there is a balanced distribution of positive and negative samples. However, in scenarios involving class imbalances or other intricate factors, its efficacy might be overshadowed by other loss functions. DR Loss, used in conjunction with the Adam optimizer [40], incorporates gradient momentum and the second moment, contributing to a more stabilized optimization process. Nevertheless, this stability might come at the expense of performance. From a mathematical perspective regarding the focal loss, it augments the weight for samples mispredicted by the model. This means the model tends to pay more attention to challenging samples, which becomes crucial in situations with class imbalances.

5.5. Ablation Study on Dynamic Head Module

The aim of this experiment was to evaluate and compare the impacts of different attention mechanisms on model performance. Attention mechanisms play a pivotal role in deep learning, especially when processing sequential and image data. By contrasting various attention mechanisms, such as multi-head attention and the dynamic head module, insights into their distinct roles and effects during model training can be gleaned. A deep understanding of these mechanisms offers a theoretical foundation for the selection and optimization of models in practical applications. The results are presented in Table 7.

Table 7. Ablation experiment results of different attention mechanism on the proposed method.

Attention Mechanism	Precision	Recall	mAP
Multi-Head [26]	0.93	0.88	0.91
Dynamic Head Module	0.95	0.91	0.94

According to Table 7, it is evident that different attention mechanisms distinctly affect model performance. Specifically, the dynamic head module outperforms the multi-head attention mechanism across all metrics, achieving an mAP of 0.94, while the multi-head attention records an mAP of 0.91. Originating from the Transformer architecture, the multi-head attention mechanism processes information in parallel by segmenting the input into multiple distinct subspaces. Each subspace possesses its own weights, enabling the model to simultaneously attend to various information segments. While this mechanism aids the model in capturing a myriad of features and patterns within the data, it may also introduce some redundancy. In contrast, the dynamic head module presents a more flexible mechanism. It can dynamically adjust attention weights based on data characteristics, thus, capturing critical features with more specificity. This dynamism allows the model to better adapt to various data and scenarios, especially when the data exhibit intricate patterns or noise. This adaptability is a plausible reason why the dynamic head module surpasses the performance of the multi-head attention.

From a mathematical perspective, the multi-head attention mechanism processes multiple subspaces' information in parallel through matrix operations. Although this parallel processing boosts efficiency, it might introduce redundancy, leading to dispersed weights, which could compromise the model's performance. On the other hand, the dynamic head module pays closer attention to the main features within the data, dynamically adjusting weights to amplify the influence of these features, thereby enhancing the model's precision.

In conclusion, different attention mechanisms possess unique characteristics and outcomes when processing data. While the multi-head attention mechanism can process information across multiple subspaces in parallel, it might lead to weight dispersion, affecting the model's performance. Conversely, the dynamic head module, by dynamically adjusting weights, captures key features more specifically, enhancing the model's performance. Such insights provide valuable guidance for researchers and engineers when choosing and optimizing models.

5.6. Ablation Study on NAS and Knowledge Graph

The primary objective of the experimental design was to investigate the role and impact of neural architecture search (NAS) and knowledge graphs on the task of chili disease detection. By comparing both with the baseline model, it was discerned whether these mechanisms could enhance the model's performance, thereby elucidating their significance in crop disease detection.

As observed in Table 8, the baseline model, devoid of any attention mechanism, exhibited performances of 0.83, 0.85, and 0.84 in precision, recall, and mAP, respectively. This served as the benchmark for subsequent comparisons. With the exclusive utilization of NAS, all evaluation metrics displayed an increase. Such findings suggest that NAS can effectively optimize the model structure, thereby enhancing its performance. A notable advantage of NAS is its capability to automatically search for an optimal model architecture,

hence identifying the most suitable model for a specific task. In this context, NAS likely pinpointed distinct features and patterns particularly apt for chili disease detection, leading to heightened accuracy and recall. In the scenario where only the knowledge graph was applied, there was an improvement in the model's performance, though potentially not as pronounced as with NAS. The primary role of the knowledge graph lies in its ability to consolidate domain knowledge, assisting the model in better understanding and interpreting data. Within the realm of chili disease detection, the knowledge graph might encompass diverse information pertinent to the disease, such as pathogens, symptoms, and growth environment. Such information can aid the model in more accurately identifying diseases. Notably, when the model integrated both NAS and the knowledge graph, all indicators experienced a significant surge. This underscores, to a certain extent, that NAS and the knowledge graph are complementary. Their concurrent application to the model can yield superior performance enhancements, indicating the considerable benefits of considering both model architecture optimization and domain knowledge integration for tasks such as chili disease detection.

Table 8. Ablation experiment results of NAS and knowledge graph.

Attention Mechanism	Precision	Recall	mAP
None (baseline)	0.83	0.85	0.84
Only NAS	0.90	0.88	0.88
Only Knowledge Graph	0.89	0.87	0.89
Both	0.95	0.91	0.94

From a mathematical standpoint, NAS primarily focuses on the optimization of the model's structure, ensuring the model's ability to capture the most valuable features from the data. Conversely, the knowledge graph emphasizes the model's semantic understanding, ensuring precise judgments in the intricate backdrop of crop diseases. Their combination equips the model with a robust discriminatory capability, making it exemplary in the task of chili disease detection. In conclusion, future crop disease detection tasks should contemplate the concurrent use of NAS and knowledge graphs to achieve heightened detection accuracy and robustness. Moreover, with the integration of more crop disease data and domain knowledge, the potential of the knowledge graph may further unfold, paving the way for significant breakthroughs in crop disease detection.

5.7. Limitations and Future Works

In the experiments conducted on edge computing platforms, a variety of common smartphones and microcomputers were covered. However, considering the rapid hardware updates, the hardware platforms used might not fully represent future devices. Additionally, for different application scenarios, a broader spectrum of hardware platforms might be considered. In this study, primary attention was given to the multi-head attention and dynamic head module attention mechanisms. Although the dynamic head module performed excellently in tests, it does not imply that it is suitable for all tasks or scenarios. The loss functions mentioned in the text, such as AP loss, DR loss, and focal loss, might not encompass all potential loss functions. Different loss functions might yield varied results under different tasks and data distributions.

Given the swift advancement in hardware technology, the future research should consider a more diverse range of new hardware platforms, including newly emerged chips and modules specifically designed for machine learning and microcomputers with higher computational capabilities. Apart from multi-head attention and dynamic head modules, many other attention mechanisms warrant exploration, such as axial attention and sparse attention. Delving deeper into these novel attention mechanisms could further enhance model performance. Regarding loss functions, future efforts might attempt to design new loss functions or combine and adjust existing ones to accommodate various tasks and data distributions.

6. Conclusions

In this study, a chili disease detection method based on deep learning is presented. Initially, the performance of various object detection models was evaluated on edge computing platforms. Through experiments, it was found that the dynamic head module and the multi-head attention mechanism exhibited distinct characteristics and performances in data processing. Notably, the dynamic head module, owing to its flexible nature, surpassed the multi-head attention mechanism in terms of performance. Furthermore, to optimize model performance, different data augmentation strategies and loss functions and their impact on model performance were explored. The experimental results indicated that when all the data augmentation methods were integrated, the model achieved the best mAP, reaching 0.94. Regarding the loss functions, the dynamic head module demonstrated higher precision and recall compared to the traditional multi-head attention mechanism.

Summarizing the core contributions of this study: First, a comprehensive deep learning framework for chili disease detection is introduced, encompassing every step from data preprocessing to model training and evaluation. Second, through a multi-faceted ablation study, various factors influencing model performance, such as data augmentation strategies, loss functions, and attention mechanisms, were revealed, offering valuable insights for future research in the field. Finally, this study not only presents an effective solution for chili disease detection but also provides insights and references for disease detection in other crops. In essence, this work offers a fresh perspective and approach to chili disease detection in modern agriculture, bridging the gap between traditional agricultural techniques and contemporary computer vision technologies. It is hoped that the findings of this study can be further expanded into practical agricultural production, contributing significantly to the advancement of modern agriculture.

Author Contributions: Conceptualization, B.X. and Q.S.; methodology, B.X. and J.W.; software, Y.L. and C.W.; validation, B.X., Q.S. and Y.L.; formal analysis, Q.S. and B.T.; resources, B.T. and Z.Y.; data curation, B.T., Z.Y., J.W. and C.W.; writing—original draft, B.X., Q.S., B.T., Y.L., Z.Y., J.W., C.W., J.L. and L.L.; writing—review and editing, L.L.; visualization, Z.Y. and J.W.; supervision, J.L.; project administration, J.L. and L.L.; funding acquisition, L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ro, N.Y.; Sebastin, R.; Hur, O.S.; Cho, G.T.; Geum, B.; Lee, Y.J.; Kang, B.C. Evaluation of Anthracnose Resistance in Pepper (*Capsicum* spp.) Genetic Resources. *Horticulturae* **2021**, *7*, 460. [CrossRef]
2. Fidan, H.; Yildiz, K.; Sarikaya, P. Molecular detection of resistance-breaking strain Cucumber mosaic virus (rbCMV) (Cucumovirus; Bromoviridae) on resistant commercial pepper cultivars in Turkey. *J. Phytopathol.* **2023**, *171*, 234–241. [CrossRef]
3. Zhang, Y.; Wa, S.; Liu, Y.; Zhou, X.; Sun, P.; Ma, Q. High-accuracy detection of maize leaf diseases CNN based on multi-pathway activation function module. *Remote Sens.* **2021**, *13*, 4218. [CrossRef]
4. Zhang, Y.; Wang, H.; Xu, R.; Yang, X.; Wang, Y.; Liu, Y. High-Precision Seedling Detection Model Based on Multi-Activation Layer and Depth-Separable Convolution Using Images Acquired by Drones. *Drones* **2022**, *6*, 152. [CrossRef]
5. Zhang, Y.; He, S.; Wa, S.; Zong, Z.; Lin, J.; Fan, D.; Fu, J.; Lv, C. Symmetry GAN Detection Network: An Automatic One-Stage High-Accuracy Detection Network for Various Types of Lesions on CT Images. *Symmetry* **2022**, *14*, 234. [CrossRef]
6. Zhang, Y.; Liu, X.; Wa, S.; Liu, Y.; Kang, J.; Lv, C. GenU-Net++: An Automatic Intracranial Brain Tumors Segmentation Algorithm on 3D Image Series with High Performance. *Symmetry* **2021**, *13*, 2395. [CrossRef]
7. Zhang, Y.; Wa, S.; Zhang, L.; Lv, C. Automatic plant disease detection based on tranvolution detection network with GAN modules using leaf images. *Front. Plant Sci.* **2022**, *13*, 875693. [CrossRef]
8. Zeng, Y.; Zhao, Y.; Yu, Y.; Tang, Y.; Tang, Y. Pepper Disease Detection Model Based on Convolutional Neural Network and Transfer Learning. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *792*, 012001. [CrossRef]
9. Li, S.; Li, K.; Qiao, Y.; Zhang, L. A multi-scale cucumber disease detection method in natural scenes based on YOLOv5. *Comput. Electron. Agric.* **2022**, *202*, 107363. [CrossRef]
10. Abbas, A.; Jain, S.; Gour, M.; Vankudothu, S. Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Comput. Electron. Agric.* **2021**, *187*, 106279. [CrossRef]

11. Sun, H.; Xu, H.; Liu, B.; He, D.; He, J.; Zhang, H.; Geng, N. MEAN-SSD: A novel real-time detector for apple leaf diseases using improved light-weight convolutional neural networks. *Comput. Electron. Agric.* **2021**, *189*, 106379. [CrossRef]
12. Wang, D.; Wang, J.; Ren, Z.; Li, W. DHBP: A dual-stream hierarchical bilinear pooling model for plant disease multi-task classification. *Comput. Electron. Agric.* **2022**, *195*, 106788. [CrossRef]
13. Peng, C.; Xia, F.; Naseriparsa, M.; Osborne, F. Knowledge Graphs: Opportunities and Challenges. *Artif. Intell. Rev.* **2023**, *56*, 13071–13102. [CrossRef] [PubMed]
14. Qiao, B.; Zou, Z.; Huang, Y.; Fang, K.; Zhu, X.; Chen, Y. A joint model for entity and relation extraction based on BERT. *Neural Comput. Appl.* **2022**, *34*, 3471–3481. [CrossRef]
15. Zhou, J.; Li, J.; Wang, C.; Wu, H.; Zhao, C.; Teng, G. Crop disease identification and interpretation method based on multimodal deep learning. *Comput. Electron. Agric.* **2021**, *189*, 106408. [CrossRef]
16. Zhu, D.; Xie, L.; Chen, B.; Tan, J.; Deng, R.; Zheng, Y.; Hu, Q.; Mustafa, R.; Chen, W.; Yi, S.; et al. Knowledge graph and deep learning based pest detection and identification system for fruit quality. *Internet Things* **2023**, *21*, 100649. [CrossRef]
17. Guan, L.; Zhang, J.; Geng, C. Diagnosis of Fruit Tree Diseases and Pests Based on Agricultural Knowledge Graph. *J. Phys. Conf. Ser.* **2021**, *1865*, 042052. [CrossRef]
18. Yu, C.; Wang, F.; Liu, Y.H.; An, L. Research on knowledge graph alignment model based on deep learning. *Expert Syst. Appl.* **2021**, *186*, 115768. [CrossRef]
19. Meng, X.; Yang, Y.; Qi, H.; Li, D.; Lu, Y.; Huang, G.; Zhang, J. Construction and Application of a Tree Knowledge Graph. In Proceedings of the 2021 IEEE/ACIS 19th International Conference on Computer and Information Science (ICIS), Shanghai, China, 23–25 June 2021. [CrossRef]
20. Zoph, B.; Le, Q.V. Neural architecture search with reinforcemenDenseNett learning. *arXiv* **2016**, arXiv:1611.01578.
21. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
22. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
23. Huang, G.; Liu, Z.; Laurens, V.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Computer Society, 2016, Las Vegas, NV, USA, 27–30 June 2016.
24. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
27. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
28. Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Philip, S.Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 494–514. [CrossRef]
29. Chen, Y.; Kuang, J.; Cheng, D.; Zheng, J.; Gao, M.; Zhou, A. AgriKG: An agricultural knowledge graph and its applications. In Proceedings of the Database Systems for Advanced Applications: DASFAA 2019 International Workshops: BDMS, BDQM, and GDMA, Chiang Mai, Thailand, 22–25 April 2019; Proceedings 24; Springer: Cham, Switzerland, 2019; pp. 533–537.
30. Qin, H.; Yao, Y. Agriculture knowledge graph construction and application. *J. Phys. Conf. Ser.* **2021**, *1756*, 012010. [CrossRef]
31. Chenglin, Q.; Qing, S.; Pengzhou, Z.; Hui, Y. Cn-MAKG: China meteorology and agriculture knowledge graph construction based on semi-structured data. In Proceedings of the 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), Singapore, 6-8 June 2018; pp. 692–696.
32. Blok, P.M.; Polder, G.; Peller, J.; van Daalen, T. *OPTIMA-RGB Colour Images and Multispectral Images (Including LabelImg Annotations)*; Wageningen University & Research: Wageningen, The Netherlands, 2022.
33. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
34. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
35. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
36. Terven, J.; Cordova-Esparza, D. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv* **2023**, arXiv:2304.00501.
37. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 213–229.

38. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
39. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
40. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
41. Chen, K.; Lin, W.; Li, J.; See, J.; Wang, J.; Zou, J. AP-loss for accurate one-stage object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3782–3798. [CrossRef]
42. Qian, Q.; Chen, L.; Li, H.; Jin, R. Dr loss: Improving object detection by distributional ranking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12164–12172.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.