

Article

IPMCNet: A Lightweight Algorithm for Invasive Plant Multiclassification

Ying Chen ^{1,2,†}, Xi Qiao ^{1,2,†} , Feng Qin ^{1,2}, Hongtao Huang ^{1,2}, Bo Liu ² , Zaiyuan Li ², Conghui Liu ² ,
Quan Wang ², Fanghao Wan ² , Wanqiang Qian ² and Yiqi Huang ^{1,*}

¹ College of Mechanical Engineering, Guangxi University, Nanning 530004, China; 2111391019@st.gxu.edu.cn (Y.C.); qiaoxi@caas.cn (X.Q.); qinfeng@st.gxu.edu.cn (F.Q.); 2211301024@st.gxu.edu.cn (H.H.)

² Shenzhen Branch, Guangdong Laboratory of Lingnan Modern Agriculture, Genome Analysis Laboratory of the Ministry of Agriculture and Rural Affairs, Agricultural Genomics Institute at Shenzhen, Chinese Academy of Agricultural Sciences, Shenzhen 518120, China; liubo03@caas.cn (B.L.); lizaiyuan@caas.cn (Z.L.); liuconghui@caas.cn (C.L.); wangquan03@caas.cn (Q.W.); wanfanghao@caas.cn (F.W.); qianwanqiang@caas.cn (W.Q.)

* Correspondence: hyq@gxu.edu.cn

† These authors contributed equally to this work.

Abstract: Invasive plant species pose significant biodiversity and ecosystem threats. Real-time identification of invasive plants is a crucial prerequisite for early and timely prevention. While deep learning has shown promising results in plant recognition, the use of deep learning models often involve a large number of parameters and high data requirements for training. Unfortunately, the available data for various invasive plant species are often limited. To address this challenge, this study proposes a lightweight deep learning model called IPMCNet for the identification of multiple invasive plant species. IPMCNet attains high recognition accuracy even with limited data and exhibits strong generalizability. Simultaneously, by employing depth-wise separable convolutional kernels, splitting channels, and eliminating fully connected layer, the model's parameter count is lower than that of some existing lightweight models. Additionally, the study explores the impact of different loss functions, and the insertion of various attention modules on the model's accuracy. The experimental results reveal that, compared with eight other existing neural network models, IPMCNet achieves the highest classification accuracy of 94.52%. Furthermore, the findings suggest that focal loss is the most effective loss function. The performance of the six attention modules is suboptimal, and their insertion leads to a decrease in model accuracy.

Keywords: deep learning; plant identification; in-field detection; convolutional neural network; invasive plants



Citation: Chen, Y.; Qiao, X.; Qin, F.; Huang, H.; Liu, B.; Li, Z.; Liu, C.; Wang, Q.; Wan, F.; Qian, W.; et al. IPMCNet: A Lightweight Algorithm for Invasive Plant Multiclassification. *Agronomy* **2024**, *14*, 333. <https://doi.org/10.3390/agronomy14020333>

Academic Editor: Rodolfo Gentili

Received: 5 January 2024

Revised: 31 January 2024

Accepted: 4 February 2024

Published: 6 February 2024



Copyright: © 2024 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

Invasive alien plants are considered one of the most significant threats to biodiversity and ecosystems worldwide [1,2]. Furthermore, economic globalization and other human activities have exacerbated this threat [3]. A crucial prerequisite for early and effective control of invasive plants is real-time identification of the species. Currently, invasive plant identification relies primarily on manual identification. On the one hand, in-field identification requires a high level of expertise from identifiers, leading to high labor costs [4]. On the other hand, the sheer diversity of invasive plant species, along with the morphological similarities between some species, can result in misidentification during the recognition process. Therefore, there is a need for a more rapid, cost-effective and accurate identification method.

The development of computer vision and image processing technologies has significantly advanced plant recognition methods based on images [5]. Image recognition

methods based on traditional machine learning techniques have achieved good results in specific tasks. The authors of [6] developed an image processing system that can identify and classify various paddy plant diseases using scale invariant feature transform features and k-nearest neighbors. A rule-based semiautomatic system using the k-means concept was designed and implemented to distinguish healthy leaves from diseased leaves, utilizing three color features including color moments, color autocorrelogram, and HSV histogram, as well as three textural features namely Haralick, Gabor, and 2D DWT [7]. However, these methods rely heavily on handcrafted features, which are domain specific and lack generality. Since the concept of deep learning was introduced by Hinton [8], convolutional neural networks (CNNs) have shown strong feature extraction capabilities in image classification [9–11], detection [12–14] and segmentation [15–17], and have found extensive applications in agriculture [18–20]. The authors of [21] presented a deep learning segmentation model that is able to distinguish between different plant species at the pixel level. The authors of [22] developed convolutional neural network models to perform plant disease detection and diagnosis using simple leaf images of healthy and diseased plants. The authors of [23] designed an identification method for cash crop diseases using automatic image segmentation and deep learning with an expanded dataset, and the system achieved a correct recognition rate of more than 80% for 27 diseases of 6 crops.

Due to the large parameter count of many deep learning models, substantial data quantities are required for training to achieve good results. However, extensive image databases for many invasive plant species are scarce, increasing the susceptibility to overfitting during the training process, which impacts recognition accuracies. Thus, there is a need to reduce parameters to make the model more lightweight to address the impact of insufficient data. Invasive plant detection involves a multiclassification problem with a complex data distribution. However, in some similar multiclassification tasks, existing lightweight models perform poorly. The authors of [24] evaluated the classification performance of 35 deep learning models on 15 weed species, and the experiments showed that lightweight neural networks like MnasNet [25] and MobileNetV3-large [26] performed less effectively than models such as ResNet50 and ResNet101 [27]. The authors of [28] used deep learning models to classify 6 tomato diseases, indicating poor accuracy with the lightweight models used in the experiments. Therefore, this study constructed a lightweight neural network called IPMCNet suitable for multiclassification tasks. This model has fewer parameters than some existing lightweight neural networks, and exhibits better fitting and generalization capabilities, can fit 34 invasive plant data in multi-class tasks, achieving higher accuracy than eight other commonly used models. The application process of this algorithm involves capturing plant images using a smartphone or a drone, uploading them to a workstation, and classifying the images using the model trained on the invasive plant dataset to obtain recognition results. This enables in-field detection of invasive plants.

The main research contributions of this paper are as follows: (1) Construction of a lightweight deep learning model suitable for the identification of various invasive plant species. (2) Exploration of the impact of different loss functions, and the use of different attention modules on multiclass tasks. (3) Proposal of a method for in-field detection of invasive plants using smartphones or drones. Section 2 introduces the dataset required for the experiments, the structure of the new model, the six popular attention modules and the loss function we used. Section 3 presents the experimental environment and results, and Section 4 provides the conclusion.

2. Methods and Materials

2.1. Dataset

The in-field detection of invasive plants often involves two scenarios. On one hand, detection personnel can capture photos of plants at close range, while, on the other hand, when plants grow in locations inaccessible to personnel, the use of drones becomes necessary for image acquisition. To ensure the identification model can address the requirements of both scenarios, the dataset used to train the model is divided into two categories: images cap-

tured manually and images captured by drones. For the first scenario, we selected 24 widely distributed invasive plant species in China. These species have already caused greater ecological harm and require urgent strengthening of management. Images for each species were downloaded from the Plant Photo Bank of China (PPBC) (<https://ppbc.iplant.cn/>) (accessed on 1 June 2023) to create the first category dataset. The Plant Photo Bank of China was officially established in 2008 by the Institute of Botany, Chinese Academy of Sciences, and serves as a dedicated repository for the management of plant images. These images were expert-identified and manually captured in the field, encompassing different shooting distances and angles, suitable for the first scenario. Data for the second application scenario were collected by our laboratory using a drone, capturing 10 species. The drone is DJI Matrice 600 Pro (DJI, Guangdong, China). The camera mounted on the drone is Nikon D850 (Nikon, Tokyo, Japan), and the focal length of the camera lens is 108 mm. Drone imagery was acquired in regions with a concentrated distribution of the target species within China, with a capture height of 30 m and a shooting angle directly overhead the plants. The photos were cropped to a size of 224 × 224 to meet the data quantity requirements. The entire dataset comprises 12,025 images from 34 different species. Table 1 presents an overview of the dataset, including the labels, names and source of each species and the corresponding number of images available. Figure 1 provides illustrative examples of each plant species. Table 2 provides a comparison between the two types of datasets.

The dataset exhibits several noteworthy characteristics. First, the data downloaded from PPBC were collected by different experts, and the images used varied in terms of plant scale and size. Second, most of the data were captured in outdoor environments, resulting in complex backgrounds that could potentially interfere with the recognition process. Furthermore, the dataset comprises a wide range of plant species, with relatively low sample sizes for each category, and the average number of images per plant species is only 354. A model trained with insufficient data lacks robust generalizability. Additionally, the data distribution is imbalanced. For instance, the quantity of data for *Mimosa bimucronata* (DC.) Kuntze is 600, while *Oenothera rosea* L'Hér. ex Aiton has only 53 instances, resulting in the difference in the quantity of data being exceeded by tenfold. In such cases, the model tends to prioritize species with larger proportions, while species with fewer data points are treated as erroneous samples, potentially compromising the model's overall performance.

Table 1. The labels, names, source, and number of 34 invasive plants. 'P' represents Plant Photo Bank of China, and 'D' represents Drone.

	Name	Source	Num		Name	Source	Num
0	<i>Bidens Pilosa</i> L.	D	400	17	<i>Sonchus oleraceus</i> L.	P	389
1	<i>Pistia stratiotes</i> L.	P	411	18	<i>Lantana camara</i> L.	D	382
2	<i>Eupatorium odoratum</i> L.	D	400	19	<i>Sphagneticola trilobata</i> (L.) Pruski	P	400
3	<i>Erigeron acris</i> L.	D	400	20	<i>Galinsoga parviflora</i> Cav.	P	400
4	<i>Oenothera rosea</i> L'Hér. ex Aiton.	P	53	21	<i>Cabomba caroliniana</i> A. Gray	P	258
5	<i>Eichhornia crassipes</i> (Mart.) Solms	P	400	22	<i>Mikania micrantha</i> Kunth in Humb. et al.	D	336
6	<i>Mimosa bimucronata</i> (DC.) Kuntze	D	600	23	<i>Amaranthus caudatus</i> L.	P	273
7	<i>Mimosa pudica</i> L.	P	400	24	<i>Ipomoea cairica</i> (L.) Sweet	P	400
8	<i>Melinis repens</i> (Willd.) Zizka	D	400	25	<i>Erigeron canadensis</i> L.	P	400
9	<i>Myriophyllum verticillatum</i> L.	D	400	26	<i>Euphorbia cyathophora</i> Murr.	P	81
10	<i>Spartina alterniflora</i> Loisel.	D	400	27	<i>Erigeron annuus</i> (L.) Pers.	P	400
11	<i>Flaveria bidentis</i> (L.) Kuntze	P	89	28	<i>Parthenium hysterophorus</i> L.	P	400
12	<i>Ageratum conyzoides</i> L.	P	149	29	<i>Cuscuta campestris</i> Yunck.	P	400
13	<i>Nicandra physalodes</i> (L.) Gaertn.	P	400	30	<i>Ipomoea purpurea</i> (L.) Roth	P	400
14	<i>Synedrella nodiflora</i> (L.) Gaertn.	P	222	31	<i>Tithonia diversifolia</i> (Hemsl.) A. Gray	P	400
15	<i>Solanum aculeatissimum</i> Jacquem.	P	400	32	<i>Crotalaria pallida</i> Blanco	P	400
16	<i>Alternanthera philoxeroides</i> (Mart.) Griseb.	D	400	33	<i>Ageratina adenophora</i> (Spreng.) R. M. King and H. Ro	P	382
	Total						12,025



Figure 1. Diagrams and names of 34 invasive plant species, arranged in the order of the species' label sequence.

Table 2. Comparison of data from the Plant Photo Bank of China and drone.

Source	Diagram	Size	Shooting Angle	Images Quantity
Plant Photo Bank of China		Various	Various	Some species have less than 100 images
Drone		224 × 224	Directly above	Each species has more than 300 images

2.2. Proposed IPMCNet Model

In response to the complex backgrounds, limited data per species and data imbalance issues are present in the studied dataset, we proposed corresponding solutions, as

illustrated in Figure 2. In the presence of intricate image backgrounds, the features of small target plants are susceptible to being overlooked by the models. Consequently, the use of small-scale convolutional kernels becomes imperative. Additionally, deepening convolutional neural networks enhances their fitting capabilities, thereby elevating the recognition accuracy. Furthermore, the insertion of attention modules is considered to heighten the focus on target regions while reducing interference from surrounding areas. In cases where insufficient data lead to overfitting, the adoption of depthwise separable convolutions, or the removal of fully connected layers, proves instrumental in significantly reducing the number of model parameters, thus preventing overfitting. Simultaneously, the segmentation of channels, followed by distinct convolutional operations, can also reduce the number of parameters and the computational load. Addressing data distribution imbalances involves considering loss functions that assign greater weights to underrepresented classes, ensuring that the model better identifies species with fewer data. We designed IPMCNet based on some of the solutions above. The overall structure of this neural network is shown in Figure 3; it consists of modules such as the DWBlock and Stem.

Complex background	Insufficient data	Data imbalance
Use 1×1 convolutions	Use depthwise separable convolutions	Replace the loss function
Deepen the model	Remove the fully connected layer	
Use the attention module	Split channel	

Figure 2. The data characteristics and methods taken to improve recognition accuracy.

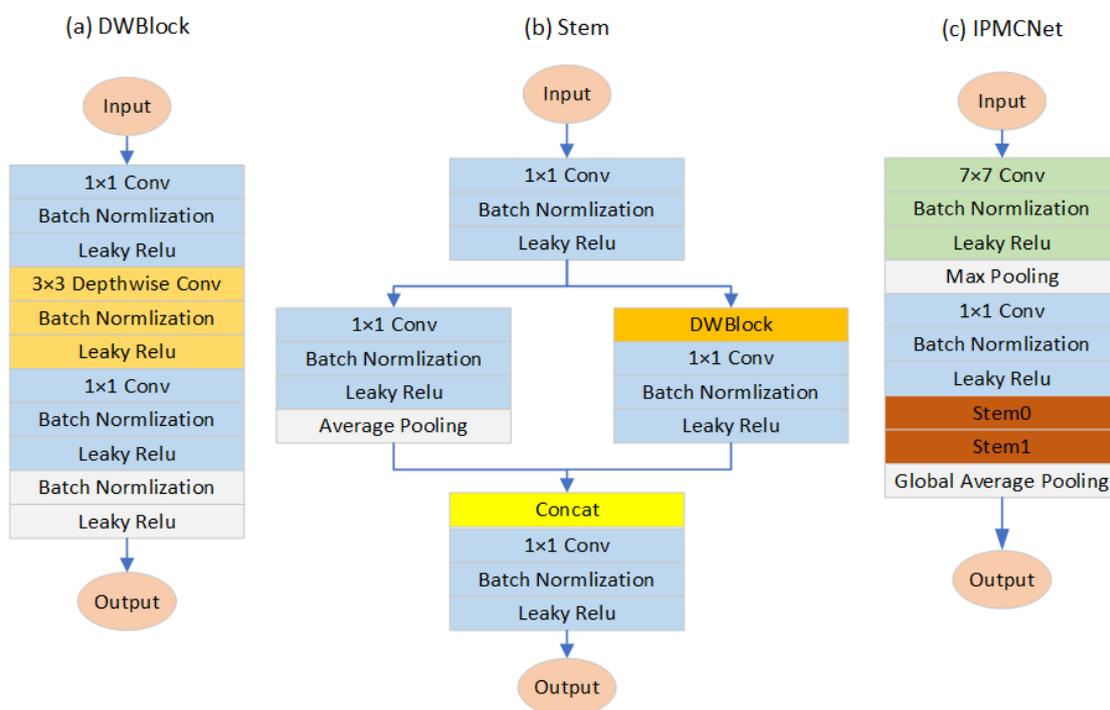


Figure 3. Overall structure of the IPMCNet model. (a) DWBlock and (b) Stem are the diagrams of the two main modules in the model, (c) IPMCNet is the diagram of the entire model.

2.2.1. DWBlock

Within the DWBlock, we initially employ 1×1 convolutional kernels [29] to change the channel numbers, ensuring effective feature extraction while reducing the computational load. Batch normalization [30] is applied after each convolutional layer to standardize the

output values, following a standard normal distribution with a mean of 0 and a variance of 1. This accelerates network training and convergence, prevents gradient vanishing, and mitigates overfitting. To avoid the issue of the ReLU function outputting zero in the negative value range [31], a leaky ReLU [32] activation function is used. For simplicity, the combination of convolutional layers, batch normalization layers and activation function layers are collectively referred to as the CBR module. Next, inspired by depthwise separable convolutions [33], we use a channel convolution with a 3×3 kernel size, followed by a pointwise convolution. This approach effectively reduces the number of parameters, enhances the local feature extraction capabilities, and facilitates the fusion and combination of features from different channels. Finally, batch normalization and an activation function are applied sequentially. The structure of the DWBlock is shown in Figure 3a.

2.2.2. Stem

In the Stem section, as shown in Figure 3b, a CBR module with a 1×1 convolutional kernel is initially used, followed by the output branching into two paths. In this structure, the branching is different from that of the ResNet shortcut [27], where ResNet directly adds the input to the output. Instead, this branching is inspired by CSPNet [34], where the feature map's channel count is split into two parts, each undergoing different convolution operations. One half is input to a CBR module with a 1×1 convolutional kernel size, while the other half passes through another CBR module before entering several DWBlocks. Subsequently, the results are concatenated with the results of the two convolution operations. Finally, a CBR module with a 1×1 convolutional kernel is used to adjust the channel count.

2.2.3. Proposed Method

The overall structure of IPMCNet is shown in Figure 3c. This process begins with a CBR module with a 7×7 convolutional kernel size, followed by the use of max pooling to reduce the number of parameters, simplify the network complexity, alleviate the excessive sensitivity of the convolutional layers to their position and increase the generalizability of the model. Subsequently, a 1×1 convolutional module is used, and its output serves as the input for two Stem modules. Stem0 is a Stem module that employs 3 DWBlocks, while Stem1 utilizes 4 DWBlocks. The utilization of two Stem modules effectively deepens the neural network model, thereby enhancing its feature extraction capabilities. Finally, global average pooling replaces the commonly used fully connected layers as the ultimate output. The global average pooling layer does not require parameters, avoiding overfitting [35]. It also accumulates spatial information, enhancing the model's robustness to spatial variations in the input.

2.2.4. Attention Module

Attention modules are crucial tools utilized to enhance a model's efficiency in processing information and performing deep learning tasks. To mitigate the impact of complex backgrounds on image model recognition capabilities, six different attention modules, namely, the squeeze-and-excitation (SE) [36], efficient channel attention module [37] (ECA), shuffle attention (SA) [38], normalization-based attention module (NAM) [39], coordinate attention (CA) [40] and convolutional block attention module (CBAM) [41], are tested with the model.

SE calculates the importance of each channel in the feature maps and assigns a weight to each feature based on this importance, allowing the neural network to focus on specific feature channels. ECA introduces a non-dimensional reduction local inter-channel interaction strategy, effectively avoiding the impact of dimensionality reduction on channel attention learning. CBAM addresses the limitation of the SEs, which consider only channels and neglect spatial information. CBAM first employs a structure similar to that of SE, generating different channel weights. Then, all the feature maps are compressed into a single feature map, and the spatial feature weights are calculated. The SA module divides the input feature maps into multiple groups, integrating channel and spatial attention into

a block for each group using Shuffle units and facilitating information communication among different sub-features through the “channel shuffle” operator. CA encodes channel relationships and long-range dependencies with precise positional information, enabling the network to focus on significant regions with a lower computational cost. NAM re-designs the channel and spatial attention sub-modules, utilizing the contribution factors of weights to enhance the attention mechanism without the use of fully connected layers and convolutional layers.

2.3. Loss Function

The loss function is employed to quantify the disparities or errors between the model’s predicted outcomes and the actual results. In the realm of deep learning, the gradients of the loss function, with respect to the model’s parameters, are computed through the backpropagation algorithm. These gradients are subsequently utilized for parameter updates to enhance model optimization. The loss function serves as a guiding framework for the adjustment of the model parameters. The different choices of a loss function exert varying impacts on the training and performance of the model. Depending on the problem’s unique characteristics and requirements, the selection of an appropriate loss function can greatly contribute to optimizing model performance.

The recognition of invasive plants constitutes a multiclass classification problem. Typically, multiclass classification problems make use of the cross entropy loss function and softmax is commonly used as the activation function.

$$\text{Cross Entropy Loss} = -\log(p_t), \quad (1)$$

t represents the true class of the sample, and p_t represents the probability value output by softmax when the predicted class is the same as the true class.

However, a primary drawback of the cross entropy loss function is its underlying assumption that all classes are equally learned. In cases of imbalanced class distributions during training, species with fewer samples encounter challenges in feature extraction for supervised algorithm learning, leading to subpar predictive performances for minority classes. Given the varying distributions and quantities of invasive plants, imbalanced dataset distributions are commonly encountered.

Consequently, in our experiments, we employ the focal loss [42] as the chosen loss function. The focal loss excels in addressing the imbalanced sample classification issue. It is capable of discerning samples based on their relative difficulty and assigns different loss weights to each sample. Specifically, it assigns smaller weights to easily differentiable samples and larger weights to those that are more challenging, thereby increasing the recognition accuracy. The following is the formula for focal loss:

$$\text{Focal Loss} = -\alpha_t(1 - p_t)^\gamma \log(p_t), \quad (2)$$

t represents the true class of the sample, and p_t represents the probability value output by softmax when the predicted class is the same as the true class. α_t is the weighting factor for the true class, and γ is the hyperparameter for adjusting class weights.

2.4. In-Field Detection System

To achieve identification of invasive plants in the wild, we propose a method that combines smartphones, drones, and cloud computing. The overall flowchart of the proposed method is depicted in Figure 4. The first step involves training the model. The dataset is sent to a computer workstation to undergo preprocessing, which includes resizing and data augmentation. IPMCNet is trained by these processed data. Subsequently, the trained model can be utilized for practical applications. When the target plants are in proximity to the researcher, the researcher can use smartphones to capture photos of the target plants. These photos are then uploaded to the workstation, where they undergo resizing and normalization. When plants are situated in locations challenging for the researcher to reach,

images are first collected using a drone. These images are uploaded to the workstation by a smartphone or a laptop, then they undergo cropping and normalization. The preprocessed images are fed into the pretrained model for prediction, and the results are transmitted back to the researcher's smartphone or laptop through the network. This allows researchers to quickly determine the types of invasive plants present at a specific location within a relatively short timeframe.

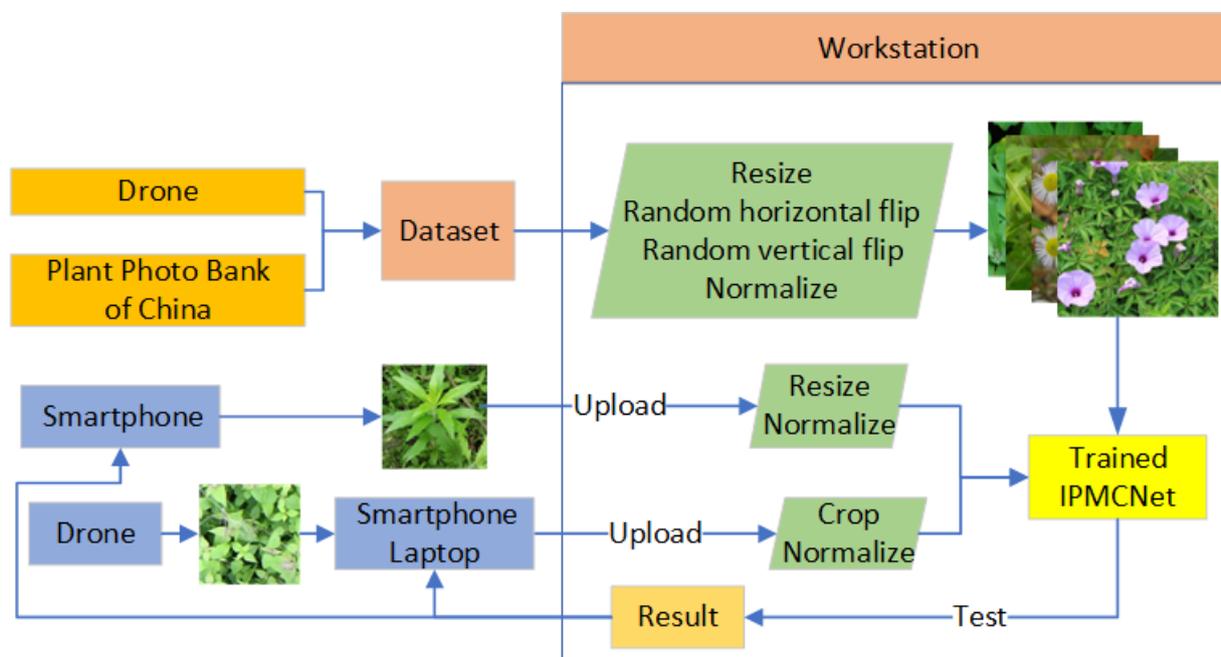


Figure 4. The overall flowchart of the in-field detection method.

3. Results and Discussion

3.1. Experimental Setup

The data were divided into three parts at a 7:2:1 ratio for training, validation, and testing. The training set data were employed to determine the parameters of the model. The validation set, independent of the training set, was used to assess the model during training. This validation process aids in providing information that may be useful for adjusting hyperparameters, particularly to assess the robustness of the model. Following the training and validation of the model, we utilized the model to predict the outputs of the data in the test set. Since the data are sourced from the internet and come in varying sizes, it is necessary to resize the data to 224×224 before feeding it into the model to ensure that each batch of data has a consistent size during deep learning training. Data augmentation is employed to increase the diversity of the data, reduce the model's dependency on specific attributes and enhance its performance and generalization capability. In this study, random horizontal flipping and random vertical flipping were used for data augmentation. The images were then normalized.

The focal loss is used as the loss function for model training, with the setting $\alpha_t = 1$ and $\gamma = 2$. The Adam optimizer [43], known for its high efficiency and low memory requirements, is used for model optimization. To prevent the learning rate from being too large, causing the loss function to directly overshoot the global optimum and making it difficult for the model to converge, the learning rate for the model is set at 0.0005. To avoid the problem of model underfitting due to insufficient training, each experiment was iterated 100 times, ensuring that all the models were fully trained. Due to the GPU memory constraints, model training was carried out with a batch size of 16.

All the procedures mentioned above were implemented and developed using the PyTorch (1.12.1+cu116) environment. The PC's GPU is an NVIDIA GeForce GTX 1080 (NVIDIA, Santa Clara, CA, USA).

3.2. Evaluation of the Proposed Model

In this section, we present the classification results of the IPMCNet model using images from the test dataset. To assess the classification performance, we calculated the precision, recall, specificity, and accuracy for each class separately. Precision is the proportion of correctly predicted positive samples among all the instances predicted as positive. Recall represents the proportion of correctly predicted positive samples among all the positive samples. Specificity is the proportion of true negative samples among the samples that were actually negative. The accuracy refers to the proportion of correctly predicted samples among all the samples. Higher values suggest that the model has better classification prediction capabilities. The formulas for the four evaluation indicators are as follows:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (4)$$

$$\text{Specificity} = \frac{TN}{TN + FP}, \quad (5)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \quad (6)$$

TP represents the true positive samples, TN represents the true negative samples, FP represents the false positive samples and FN represents the false negative samples.

The quantitative evaluation results of the IPMCNet model on the test set are presented in Table 3 and Figure 5. It is evident that the IPMCNet model exhibits a remarkable performance in identifying various invasive plants. The results indicate that most of the samples in each category are accurately recognized. The average precision, recall, specificity, and accuracy for all the species were 93.66%, 93.98%, 99.83%, and 99.68%, respectively.

Table 3. IPMCNet classification test results for 34 invasive plants.

Species	Num	Pr. (%)	Re. (%)	Sp. (%)	Acc. (%)
<i>Bidens Pilosa</i> L.	80	98.77	100.00	99.96	99.96
<i>Pistia stratiotes</i> L.	83	96.25	92.77	99.87	99.62
<i>Eupatorium odoratum</i> L.	80	100.00	98.75	100.00	99.96
<i>Erigeron acris</i> L.	80	100.00	100.00	100.00	100.00
<i>Oenothera rosea</i> L'Hér. ex Aiton.	11	100.00	90.91	100.00	99.96
<i>Eichhornia crassipes</i> (Mart.) Solms	80	100.00	95.00	100.00	99.83
<i>Mimosa bimucronata</i> (DC.) Kuntze	80	100.00	98.75	100.00	99.96
<i>Mimosa pudica</i> L.	80	95.00	95.00	99.83	99.66
<i>Melinis repens</i> (Willd.) Zizka	80	100.00	100.00	100.00	100.00
<i>Myriophyllum verticillatum</i> L.	80	100.00	100.00	100.00	100.00
<i>Spartina alterniflora</i> Loisel.	80	100.00	100.00	100.00	100.00
<i>Flaveria bidentis</i> (L.) Kuntze	18	73.68	77.78	99.79	99.62
<i>Ageratum conyzoides</i> L.	30	75.76	83.33	99.66	99.45
<i>Nicandra physalodes</i> (L.) Gaertn.	80	96.25	96.25	99.87	99.75
<i>Synedrella nodiflora</i> (L.) Gaertn.	45	80.00	88.89	99.57	99.37
<i>Solanum aculeatissimum</i> Jacquem.	80	87.34	86.25	99.56	99.11
<i>Alternanthera philoxeroides</i> (Mart.) Griseb.	80	98.77	100.00	99.96	99.96
<i>Sonchus oleraceus</i> L.	78	92.50	94.87	99.74	99.58
<i>Lantana camara</i> L.	77	100.00	100.00	100.00	100.00
<i>Sphagneticola trilobata</i> (L.) Pruski	80	97.30	90.00	99.91	99.58
<i>Galinsoga parviflora</i> Cav.	80	96.25	96.25	99.87	99.75

Table 3. Cont.

Species	Num	Pr. (%)	Re. (%)	Sp. (%)	Acc. (%)
<i>Cabomba caroliniana</i> A. Gray	52	84.75	96.15	99.61	99.54
<i>Mikania micrantha</i> Kunth in Humb. et al.	68	100.00	100.00	100.00	100.00
<i>Amaranthus caudatus</i> L.	55	94.83	100.00	99.87	99.87
<i>Ipomoea cairica</i> (L.) Sweet	80	95.24	100.00	99.83	99.83
<i>Erigeron canadensis</i> L.	80	86.30	78.75	99.56	98.86
<i>Euphorbia cyathophora</i> Murr.	17	89.47	100.00	99.92	99.92
<i>Erigeron annuus</i> (L.) Pers.	80	98.75	98.75	99.96	99.92
<i>Parthenium hysterophorus</i> L.	80	78.57	96.25	99.08	98.99
<i>Cuscuta campestris</i> Yunck.	80	100.00	97.50	100.00	99.92
<i>Ipomoea purpurea</i> (L.) Roth	80	100.00	92.50	100.00	99.75
<i>Tithonia diversifolia</i> (Hemsl.) A. Gray	80	84.71	90.00	99.43	99.11
<i>Crotalaria pallida</i> Blanco	80	91.30	78.75	99.74	99.03
<i>Ageratina adenophora</i> (Spreng.) R. M. King and H. Ro	77	92.65	81.82	99.78	99.20
Avg (UAV)	78	99.75	99.75	99.99	99.98
Avg (Plant Photo Bank of China)	66	91.12	91.57	99.77	99.55
Avg (Total)	70	93.66	93.98	99.83	99.68

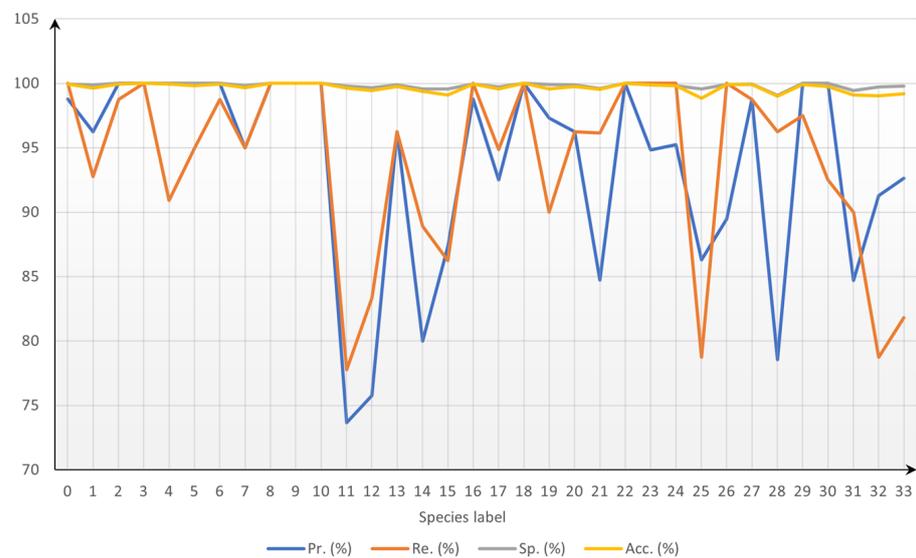


Figure 5. A line chart of the classification results. The blue line represents precision, the orange represents recall, the gray represents specificity, and the yellow represents accuracy.

Among the 24 plant species obtained from the PPBC, the average precision and recall were 91.12% and 91.57%. This is notably lower than the 10 species collected by drone. Furthermore, species with precision below 80% include *Flaveria bidentis* (L.) Kuntze, *Ageratum conyzoides* L., *Synedrella nodiflora* (L.) Gaertn., *Parthenium hysterophorus* L., and species with recall below 80% are *Flaveria bidentis* (L.) Kuntze, *Erigeron canadensis* L., *Crotalaria pallida* Blanco. All data for these species are sourced exclusively from the PPBC. The suboptimal performance of the PPBC data can be attributed to several factors: (1) A lower quantity of images. (2) Diverse shooting locations for images. Under insufficient data conditions, images of different organs of the same species may be misclassified as different species. (3) Resizing of images of varying sizes to 224×224 during preprocessing, leading to varying degrees of deformation in recognition targets and increasing the difficulty of identification.

The confusion matrix depicted in Figure 6 compares the true class to the predicted class. The vertical axis of the confusion matrix is the predicted label, and the horizontal axis is the true label. The species names corresponding to the serial numbers in the images can be found in Table 1. The number of predictions that accurately match the category level of the test data is represented by the diagonal matrix values, while the off-diagonal elements correspond to inaccurate predictions. Notably, accurately identifying plants belonging within *Asteraceae* in the dataset is more challenging. For instance, 9 images of *Erigeron canadensis* L. were identified as *Parthenium hysterophorus* L. Moreover, 5 images of *Ageratina adenophora* (Spreng.) R. M. King and H. Rob. was misclassified as *Parthenium hysterophorus* L., and 4 images of *Sphagneticola trilobata* (L.) Pruski were identified as *Tithonia diversifolia* (Hemsl.) A. Gray. An additional solution involves training the model with more diverse species of *Asteraceae* that share similar morphologies.

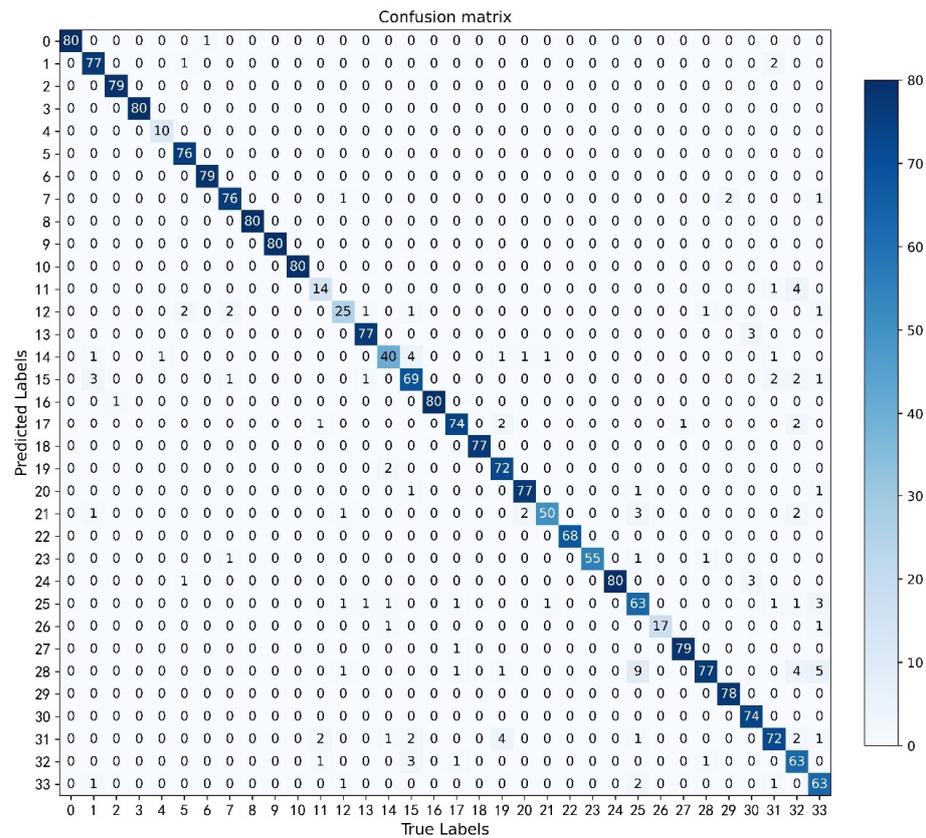


Figure 6. The confusion matrix generated from the classification results. The vertical axis of the confusion matrix is the predicted label, and the horizontal axis is the true label.

In order to more visually display the training and validation process of the model, Figure 7 shows the accuracy and loss curves. Before approximately the 50th epoch, both the training and validation accuracies were relatively low and showed an upward trend, indicating that the current model could not effectively fit the data and was in an underfitting state. After approximately the 60th epoch, the training accuracy curve surpassed the validation accuracy curve with a slight upward trend, while the validation accuracy curve exhibited a slight downward trend, suggesting that the model is in a mild overfitting state. In future training iterations of the model, limiting the number of epochs to around 60 can help reduce the negative impact of overfitting on accuracy and conserve computational resources.

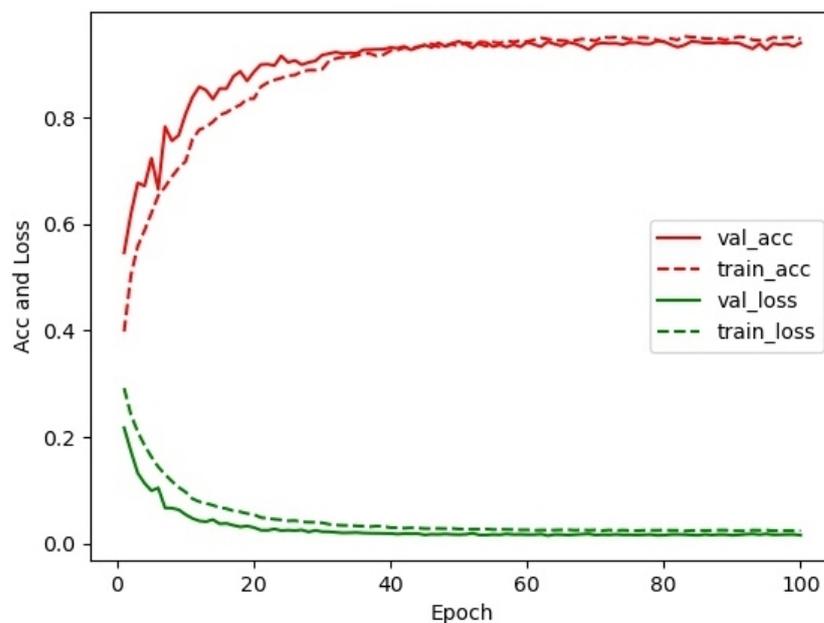


Figure 7. Accuracy and loss curves during model training and validation.

3.3. Comparative Analysis of the Different Models

We trained, validated, and tested four classic neural network models, EfficientNet_V2_S [44], ResNet101, ResNet50 and ConvNext_Tiny [45], as well as four lightweight neural network models, MobileNetV3, MnasNet1_3, SqueezeNet [46] and ShuffleNet_V2_X1_0 [47], using the same dataset and training environment with identical parameters. We then compared the results with those of the newly developed IPMCNet. Our experimental results demonstrate that the IPMCNet model achieved a Top-1 accuracy of 94.52%, with the second-lowest number of parameters among all the models (IPMCNet: 1.32 M, SqueezeNet: 1.25 M).

Notably, IPMCNet had 20.14 M fewer parameters than did the second-lowest parameter count model, EfficientNet_v2_s, among the existing classic neural networks. Compared with existing lightweight models, IPMCNet's accuracy was 2.66% greater than that of the second-best model, MnasNet1_3. This indicates that IPMCNet offers significant advantages over the existing models in terms of both high accuracy and low parameter count. Figure 8 shows a comparison of the accuracy and parameters among the different models. The higher the recognition accuracy of the model is, the closer its position is to the top in the graph. The fewer model parameters there are, the closer the position is to the left side of the image. IPMCNet is located at the top left of the chart.

The accuracy and loss curves for the training and validation of the nine models are illustrated in Figure 9. The red line represents accuracy, the green line represents loss, the solid lines denote the validation process and the dashed lines denote the training process. As indicated by the red dashed lines, after approximately 50–60 epochs of training, each model achieved an accuracy of more than 90% on the training dataset, indicating a strong fitting capability to the training data. However, the solid red lines are consistently below the red dashed lines, indicating that the accuracy during the validation process is lower than that during the training process, indicating the occurrence of overfitting. Comparing the nine plots, it is evident that IPMCNet exhibits the least degree of overfitting and demonstrates the strongest generalization ability among the models.

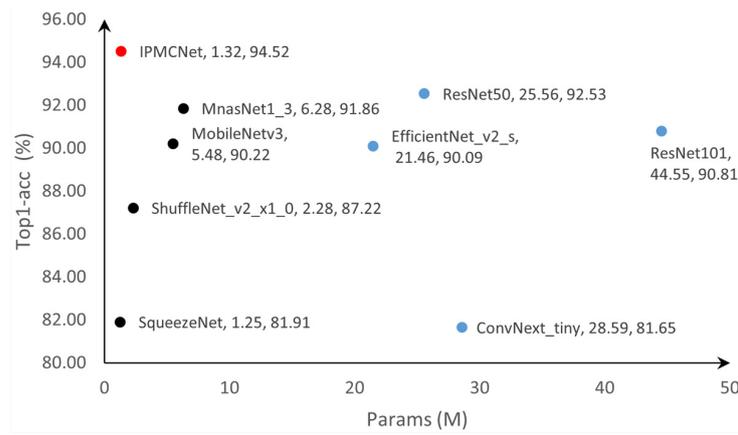


Figure 8. Parameter quantities and Top1-acc of each model. The red dots represent the proposed IPMCNet, the black dots represent lightweight models, and the blue dots represent nonlightweight classical models.

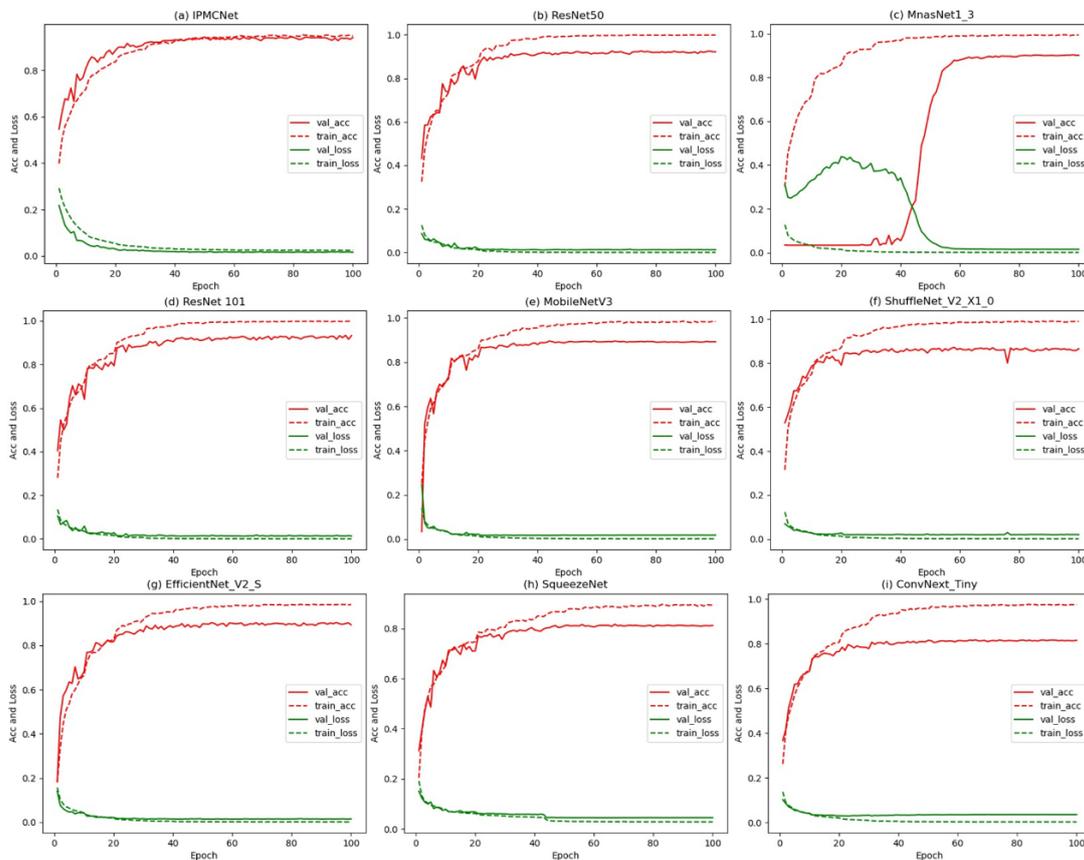


Figure 9. The accuracy and loss curves for the training and validation of the nine models. The charts are arranged from (a–i) in descending order of test accuracy.

3.4. Comparison of Results Using Different Loss Functions

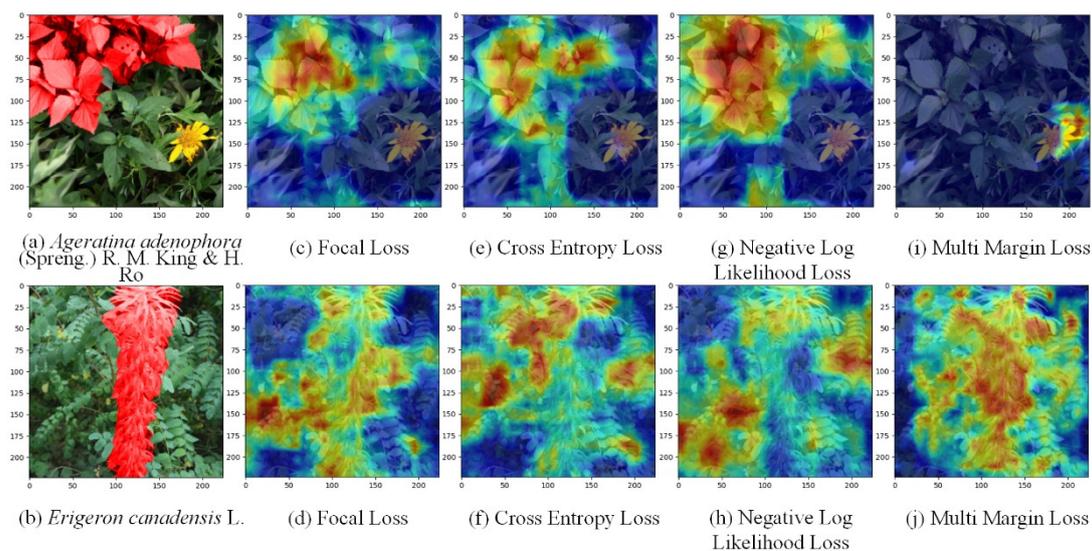
To explore the impact of different loss functions on model training, in addition to the focal loss, we trained the model using common loss functions such as cross entropy loss, multi-margin loss and negative log likelihood loss (NLLoss), which are equally suitable for addressing class imbalances. The experimental results are presented in Table 4 below. The Top-1 accuracies for cross entropy loss, NLLoss and multi-margin loss were 93.38%, 89.75% and 70.94%, respectively, which are lower than the 94.52% accuracy achieved with focal loss. It is evident that the focal loss performs better in handling these multiclass, class-imbalanced data in this experiment.

Table 4. Top1-acc of IPMCNet trained using different loss functions.

Loss Function	Top1-acc (%)
Focal Loss	94.52
Cross Entropy Loss	93.38
Negative Log Likelihood Loss	89.75
Multi-Margin Loss	70.94

To visually demonstrate the feature extraction capability of models trained with different loss functions, we utilized gradient-weighted class activation mapping (Grad-CAM) [48] to visualize the final feature extraction results. Grad-CAM is a technique that shows which parts of a deep neural network contribute the most to the prediction results. The output feature map of the last convolutional layer in the neural network has the greatest impact on the classification results, Grad-CAM can calculate the weight of each channel by performing global average pooling on the gradient of the last convolutional layer. These weights are then multiplied by the feature map to generate a class activation map (CAM), where each pixel represents the importance of that pixel region for the classification result. Red indicates that the pixel area is the most important to the classification result and is followed by yellow, the blue indicates that the pixel area has no impact on the classification result. We compared the focal points of different models on the same images.

Figure 10a,b show two images with complex backgrounds from the *Ageratina adenophora* (Spreng.) R. M. King and H. Ro and *Erigeron canadensis* L. datasets, the location of the target plant is marked in red. Figure 10c–j show the class activation maps generated by models trained with focal loss, cross entropy loss, NLLoss, and multi-margin loss for IPMCNet. In the CAM of *Ageratina adenophora* (Spreng.) R. M. King and H. Ro, trained with focal loss, cross entropy loss and negative log likelihood loss, as shown in Figure 10c,e,g, respectively, all the models successfully identified the position of *Ageratina adenophora* (Spreng.) R. M. King and H. Ro; however, the model trained with focal loss focused more accurately on the region. The model trained with the multi-margin loss focused on incorrect areas, as shown in Figure 10i. In the CAM of *Erigeron canadensis* L., models trained with cross entropy loss and NLLoss mainly focused on background plants. Models trained with focal loss and multi-margin loss assigned higher weights to the target plants, as shown in Figure 10d,j; however, the model trained with multi-margin loss had more misidentified regions than did the focal loss-trained model. In conclusion, the focal loss model exhibited the strongest ability to extract features from images of these two different types of invasive plants.

**Figure 10.** Grad-CAM for models trained with different loss functions.

3.5. Comparison of Results Using Different Attention Modules

To mitigate the adverse impact of complex image backgrounds on classification accuracies, and to enable the model to focus on object recognition, we conducted six experiments by incorporating six types of attention modules, SE, ECA, SA, NAM, CBAM, and CA into IPMCNet. These modules were inserted after the last convolutional block of the DWBlock so that they could assign different weights to the previously extracted feature maps before output. Top1-acc and params of IPMCNet using different attention models are presented in Figure 11. It is evident that the insertion of the NAM yields the highest Top-1 accuracy (93.01%). However, this approach falls short of the performance of the model without any attention modules. After the insertion of attention modules, there was no significant increase in the model's parameter count, and it is challenging to observe a clear correlation between the changes in parameter count and Top-1 accuracy with the addition of different attention modules.

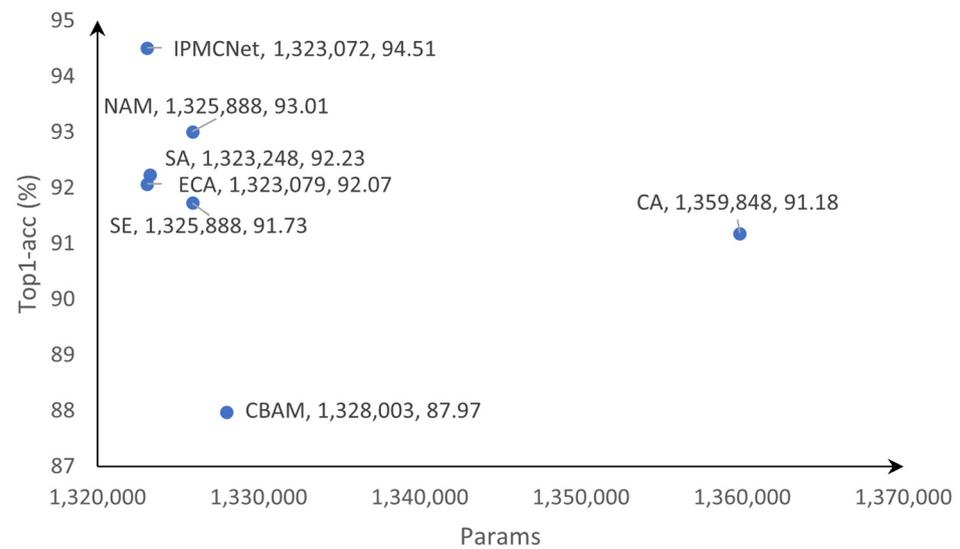


Figure 11. Top1-acc and parameter quantities of IPMCNet using different attention models.

We also employed Grad-CAM to visually illustrate the feature extraction results of the different models. Figure 12a,b present two images with complex backgrounds from the *Ageratina adenophora* (Spreng.) R. M. King and H. Ro and *Erigeron canadensis* L., the location of the target plant is marked in red. Figure 12c–p depict the class activation maps generated by a trained IPMCNet and a trained IPMCNet with SE, ECA, SA, NAM, CA and CBAM. In the CAM of *Ageratina adenophora* (Spreng.) R. M. King and H. Ro, only the model without any attention modules accurately identified the position of *Ageratina adenophora* (Spreng.) R. M. King and H. Ro. The model employing the NAM exhibited more focus on the background, excluding the target plant. The IPMCNet model with the inserted CA module exhibited attention across most areas of the image but focused on the edges, as shown in Figure 12m. Models with the ECA, SA, SE, and CBAM modules showed misaligned attention points. In the CAM of *Erigeron canadensis* L., IPMCNet achieved higher recognition accuracy than IPMCNet with the inserted NAM, ECA, and SE modules, focusing on fewer background plants. The model using the SA module concentrated only on a small portion of the target plant. After the CA and CBAM modules were inserted, the model's attention shifted away from *Erigeron canadensis* L., highlighting other plants, resulting in a recognition performance slightly worse than that of the former cases, as shown in Figure 12n,p. In conclusion, models without attention modules demonstrated the strongest feature extraction capabilities on images of these two different types of invasive plants. This outcome may be attributed to the limited dataset size and excessive noise, with attention modules excessively focusing on noise or irrelevant information, resulting in a decline in recognition accuracy.

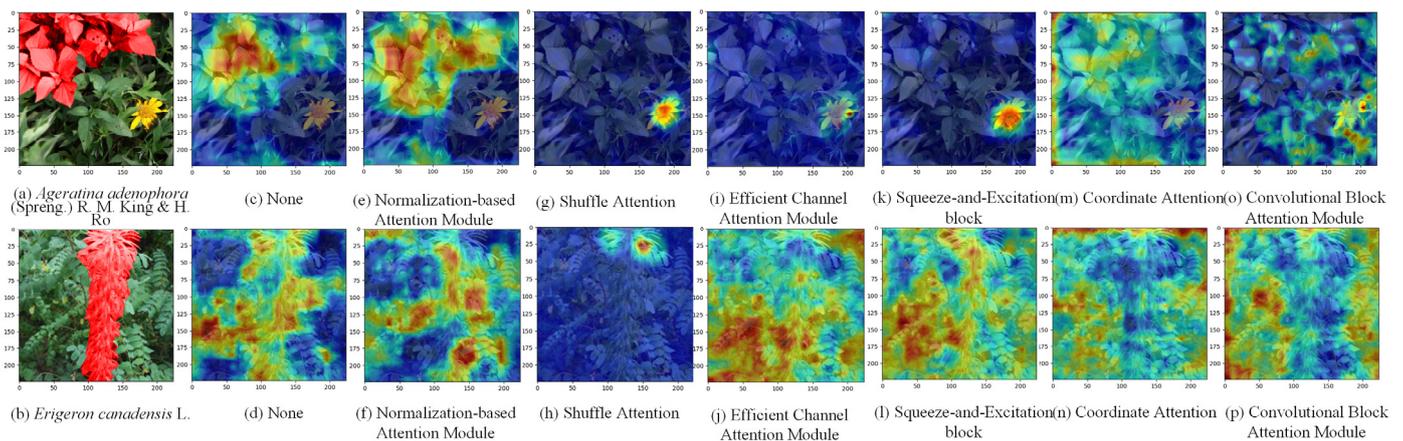


Figure 12. Grad-CAM for models using different attention modules.

3.6. Advantages and Improvements of IPMCNet

In summary, IPMCNet exhibits the following advantages in practical applications: (1) Highest accuracy in identification. Given the rapid spread of invasive plants, the more accurate in-field detection of invasive plants, the more timely control measures can be implemented, and the fewer economic losses caused by the spread of invasive plants. (2) Strong generalization ability of the model. In scenarios with insufficient data, a diverse range of species, and varied image capture angles and distances with complex backgrounds, the model demonstrates the lowest degree of overfitting. (3) Fewer parameters. This characteristic makes the model's training process less demanding on computer hardware and enables fast identification. In the future, the model can be embedded in mobile phones and other portable devices to automate the detection of invasive plants.

For further enhancement of the model's performance, improvements can be explored in two directions: model optimization and dataset optimization. In terms of model optimization, considerations may include (1) utilizing more efficient and lightweight structures, such as incorporating channel shuffle from ShuffleNet to reduce the number of model parameters, and (2) designing a two-stage neural network for detection and classification. Employing detection boxes of the same shape and integrating concepts from object detection models like Faster R-CNN [49] can help identify core regions for classification, reducing the adverse effects of varying image shapes and complex backgrounds. In terms of dataset optimization, efforts could focus on collecting data with more diverse angles and distances, capturing various organs of the same species. Additionally, gathering data for more similar species, such as those challenging to identify within the *Asteraceae* in the experiment, could be beneficial.

4. Conclusions

To mitigate the threats posed by invasive plants, timely and accurate identification of these species is of paramount importance. Consequently, the development of a self-efficient, cost-effective method for invasive plant detection holds significant practical significance. In recent years, deep learning models have demonstrated remarkable image recognition task performances. However, classical CNN models come with a substantial number of parameters and require large datasets for training to ensure accuracy. Unfortunately, most available invasive plant image datasets are limited in size, leading to suboptimal recognition performance of existing models. Therefore, there is a need to develop a CNN model that combines high recognition accuracies with lightweight characteristics.

In this study, we introduce a novel network architecture named IPMCNet for invasive plant recognition, which exhibits a low parameter count. The experimental results indicate that the proposed approach effectively identifies 34 different invasive plant species. Compared with commonly used neural network models, including EfficientNet_V2_S (90.09%), ResNet101 (90.81%), ResNet50 (92.53%), ConvNext_Tiny (81.65%), MobileNetV3

(90.22%), MnasNet1_3 (91.86%), SqueezeNet (81.91%), and ShuffleNet_V2_X1_0 (87.22%), it achieves the highest accuracy (94.52%). Additionally, the model has a parameter count of 1.32 M, ranking as the second lowest among all models, surpassed only by SqueezeNet (1.25 M). Furthermore, the model exhibits excellent generalization across various shooting environments, different image sizes and varying vegetation scales.

To mitigate the adverse effects of data imbalances, four different loss functions, focal loss, cross entropy loss, negative log likelihood loss and multi-margin loss were employed in the experiments. It was observed that the model trained with focal loss achieved the highest accuracy. To enhance the algorithm's capability to handle complex background images, six distinct attention modules including the squeeze-and-excitation block, efficient channel attention module, shuffle attention, normalization-based attention module, coordinate attention, and convolutional block attention module were incorporated into the model. The results indicated that all six modules led to a decrease in the model's accuracy.

Author Contributions: Conceptualization, Y.C., X.Q., F.Q., H.H., B.L., Z.L., C.L., Q.W., F.W., W.Q. and Y.H.; Data curation, Y.C., X.Q. and H.H.; Formal analysis, Y.C.; Funding acquisition, X.Q., F.W., W.Q. and Y.H.; Investigation, X.Q. and Y.H.; Methodology, Y.C.; Project administration, X.Q., B.L., Z.L., C.L., Q.W., F.W., W.Q. and Y.H.; Resources, X.Q., F.W., W.Q. and Y.H.; Software, Y.C. and F.Q.; Supervision, X.Q., F.W., W.Q. and Y.H.; Validation, Y.C.; Visualization, Y.C.; Writing—original draft, Y.C.; Writing—review and editing, Y.C., X.Q., F.Q., H.H., B.L., Z.L., C.L., Q.W., F.W., W.Q. and Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key Research and Development Program of China (2023YFC2605200 and 2023YFC2605202), the National Natural Science Foundation of China (32272633), the Guangxi Natural Science Foundation of China (2021JJA130221), Shenzhen Science and Technology Program (KCXFZ20230731093259009) and the Agricultural Science and Technology Innovation Program.

Data Availability Statement: All the data mentioned in the paper are available from the corresponding author.

Acknowledgments: Acknowledgement for the data support from Plant Data Center of Chinese Academy of Sciences (<https://www.plantplus.cn> (accessed on 1 June 2023)).

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

References

1. Nguyen, D.T.C.; Tran, T.V.; Nguyen, T.T.T.; Nguyen, D.H.; Alhassan, M.; Lee, T. New frontiers of invasive plants for biosynthesis of nanoparticles towards biomedical applications: A review. *Sci. Total Environ.* **2023**, *857*, 159278. [CrossRef]
2. Dyrmann, M.; Mortensen, A.K.; Linneberg, L.; Hoye, T.T.; Bjerge, K. Camera Assisted Roadside Monitoring for Invasive Alien Plant Species Using Deep Learning. *Sensors* **2021**, *21*, 6126. [CrossRef]
3. Qian, W.Q.; Huang, Y.Q.; Liu, Q.; Fan, W.; Sun, Z.Y.; Dong, H.; Wan, F.H.; Qiao, X. UAV and a deep convolutional neural network for monitoring invasive alien plants in the wild. *Comput. Electron. Agric.* **2020**, *174*, 105519. [CrossRef]
4. Wäldchen, J.; Rzanny, M.; Seeland, M.; Mäder, P. Automated plant species identification—Trends and future directions. *PLoS Comput. Biol.* **2018**, *14*, e1005993. [CrossRef] [PubMed]
5. Chen, Y.; Huang, Y.; Zhang, Z.; Wang, Z.; Liu, B.; Liu, C.; Huang, C.; Dong, S.; Pu, X.; Wan, F.; et al. Plant image recognition with deep learning: A review. *Comput. Electron. Agric.* **2023**, *212*, 108072. [CrossRef]
6. Jagan, K.; Balasubramanian, M.; Palanivel, S. Detection and Recognition of Diseases from Paddy Plant Leaf Images. *Int. J. Comput. Appl.* **2016**, *144*, 34–41. [CrossRef]
7. Kaur, S.; Pandey, S.; Goel, S. Semi-automatic leaf disease detection and classification system for soybean culture. *IET Image Process.* **2018**, *12*, 1038–1048. [CrossRef]
8. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. AcM* **2017**, *60*, 84–90. [CrossRef]
9. Kalampokas, T.; Vrochidou, E.; Papakostas, G.A.; Pachidis, T.; Kaburlasos, V.G. Grape stem detection using regression convolutional neural networks. *Comput. Electron. Agric.* **2021**, *186*, 106220. [CrossRef]
10. Too, E.C.; Li, Y.; Njuki, S.; Liu, Y. A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **2019**, *161*, 272–279. [CrossRef]
11. Rodrigues, L.; Magalhaes, S.A.; da Silva, D.Q.; dos Santos, F.N.; Cunha, M. Computer Vision and Deep Learning as Tools for Leveraging Dynamic Phenological Classification in Vegetable Crops. *Agronomy* **2023**, *13*, 463. [CrossRef]

12. Gao, J.; French, A.P.; Pound, M.P.; He, Y.; Pridmore, T.P.; Pieters, J.G. Deep convolutional neural networks for image-based *Convolvulus sepium* detection in sugar beet fields. *Plant Methods* **2020**, *16*, 29. [[CrossRef](#)]
13. Hassan, S.M.; Maji, A.K. Plant Disease Identification Using a Novel Convolutional Neural Network. *IEEE Access* **2022**, *10*, 5390–5401. [[CrossRef](#)]
14. Saleem, M.H.; Potgieter, J.; Arif, K.M. Weed Detection by Faster RCNN Model: An Enhanced Anchor Box Approach. *Agronomy* **2022**, *12*, 1580. [[CrossRef](#)]
15. Das, M.; Bais, A. DeepVeg: Deep Learning Model for Segmentation of Weed, Canola, and Canola Flea Beetle Damage. *IEEE Access* **2021**, *9*, 119367–119380. [[CrossRef](#)]
16. Yuheng, S.; Hao, Y. Image Segmentation Algorithms Overview. *arXiv* **2017**, arXiv:1707.02051.
17. Li, S.; Li, B.; Li, J.; Liu, B.; Li, X. Semantic Segmentation Algorithm of Rice Small Target Based on Deep Learning. *Agriculture* **2022**, *12*, 1232. [[CrossRef](#)]
18. Tang, Y.; Chen, M.; Wang, C.; Luo, L.; Li, J.; Lian, G.; Zou, X. Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review. *Front. Plant Sci.* **2020**, *11*, 510. [[CrossRef](#)] [[PubMed](#)]
19. Teimouri, N.; Dyrmann, M.; Nielsen, P.R.; Mathiassen, S.K.; Somerville, G.J.; Jorgensen, R.N. Weed Growth Stage Estimator Using Deep Convolutional Neural Networks. *Sensors* **2018**, *18*, 1580. [[CrossRef](#)]
20. Darwin, B.; Dharmaraj, P.; Prince, S.; Popescu, D.E.; Hemanth, D.J. Recognition of Bloom/Yield in Crop Images Using Deep Learning Models for Smart Agriculture: A Review. *Agronomy* **2021**, *11*, 646. [[CrossRef](#)]
21. Picon, A.; San-Emeterio, M.G.; Bereciartua-Perez, A.; Klukas, C.; Eggers, T.; Navarra-Mestre, R. Deep learning-based segmentation of multiple species of weeds and corn crop using synthetic and real image datasets. *Comput. Electron. Agric.* **2022**, *194*, 106719. [[CrossRef](#)]
22. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [[CrossRef](#)]
23. Xiong, Y.; Liang, L.; Wang, L.; She, J.; Wu, M. Identification of cash crop diseases using automatic image segmentation algorithm and deep learning with expanded dataset. *Comput. Electron. Agric.* **2020**, *177*, 105712. [[CrossRef](#)]
24. Chen, D.; Lu, Y.; Li, Z.; Young, S. Performance evaluation of deep transfer learning on multi-class identification of common weed species in cotton production systems. *Comput. Electron. Agric.* **2022**, *198*, 107091. [[CrossRef](#)]
25. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. MnasNet: Platform-Aware Neural Architecture Search for Mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
26. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.-C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 770–778.
28. Zhang, N.; Wu, H.; Zhu, H.; Deng, Y.; Han, X. Tomato Disease Classification and Identification Method Based on Multimodal Fusion Deep Learning. *Agriculture* **2022**, *12*, 2014. [[CrossRef](#)]
29. Lin, M.; Chen, Q.; Yan, S. Network In Network. *arXiv* **2014**, arXiv:1312.4400v3.
30. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
31. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv* **2015**, arXiv:1505.00853.
32. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; p. 3.
33. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv* **2017**, arXiv:1610.02357.
34. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Electr Network, Seattle, WA, USA, 14–19 June 2020; pp. 1571–1580.
35. Xu, Y.; Zhao, B.; Zhai, Y.; Chen, Q.; Zhou, Y. Maize Diseases Identification Method Based on Multi-Scale Convolutional Global Pooling Neural Network. *IEEE Access* **2021**, *9*, 27959–27970. [[CrossRef](#)]
36. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [[CrossRef](#)] [[PubMed](#)]
37. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539.
38. Zhang, Q.L.; Yang, Y.B. SA-Net: Shuffle Attention for Deep Convolutional Neural Networks. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 2235–2239.
39. Liu, Y.; Shao, Z.; Teng, Y.; Hoffmann, N. NAM: Normalization-based Attention Module. *arXiv* **2021**, arXiv:2111.12419.

40. Hou, Q.B.; Zhou, D.Q.; Feng, J.S. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Electr Network, Virtual, 19–25 June 2021; pp. 13708–13717.
41. Woo, S.; Park, J.; Lee, J.-Y. CBAM: Convolutional Block Attention Module. *arXiv* **2018**, arXiv:1807.06521v2.
42. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2018**, arXiv:1708.02002.
43. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
44. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2020**, arXiv:1905.11946v5.
45. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 11966–11976.
46. Iandola, F.; Han, S.; Moskewicz, M.; Ashraf, K.; Dally, W.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
47. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
48. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [[CrossRef](#)]
49. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:1506.01497. [[CrossRef](#)]

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.