Deep Learning Methods for Plant Image Segmentation and Classification

Jingxuan Su

Supervisor: Lyudmila Mihaylova

A Thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

Department of Automatic Control and Systems Engineering

April 16, 2024

Acknowledgements

I express my sincere gratitude to my supervisor Professor Lyudmila Mihaylova for encouragement in both my academic and personal life. I am particularly appreciative of the significant freedom she afforded me in selecting the direction and methodology of my research. She is always enthusiastic to discuss a novel approach to my research. It has been especially helpful that her inspired ideas and advice guided me to explore the direction of research. I would like to thank Mila for her valuable advice and kind help in supporting my non-academic life.

I am grateful to Dr Sean Anderson who gave me great help and valuable advice on my papers. I am truly thanks to Assoc. Prof. Charoenchai Khompatraporn and his team for collecting data and helpful comments on my papers. This collaboration with King Mongkut's University of Technology impressed me and enhanced my collaborative skills.

I would like to acknowledge the UK Royal Academy of Engineering and the Thailand Integrated Research for the funding which enables me to start my research.

Great thanks must go to my mum and all my family for their endless love. Without their constant support and encouragement, I would not have had a chance to start and insist on the whole PhD journey. Thank my boyfriend for his invaluable patience, support and accompany.

I am also appreciative of all the colleagues of Prof. Lyudmila Mihaylova's group. It is really lucky to be a member of this team. Your support made my PhD a great journey. Special thanks are due to the friends I have encountered during my stay in the United Kingdom. Thank you so much for your accompany and support.

To dedicate in memory of my late grandfather Zhixiang Liu, his endless love and wisdom continue to inspire my journey.

Abstract

Precision agriculture relies heavily on the crucial components of plant image segmentation and classification. The application of image classification is particularly related to disease identification and plant recognition, contributing to heightened accuracy and operational efficiency. Concurrently, image segmentation plays a pivotal role in extracting plant objects, facilitating yield prediction, disease localization, and weed detection.

The thesis starts with the development of innovative deep learning algorithms for autonomous precision agriculture. A novel framework for imbalanced semantic segmentation is proposed in Chapter 3, based on fully convolutional network architecture, a feature learning of weight update approach and an effective data balance scheme. Apart from the dynamic weight updates, learning holistic feature knowledge emerges as a pivotal factor in enhancing overall performance. Chapter 4 introduces a novel learning network based on the Squeeze and Excitation Network, specifically designed for fine-grained plant pathology classification. This architecture integrates label knowledge and feature knowledge to represent plant diseases, surpassing the capabilities of single learning networks. It excels in the self-distillation of additional feature knowledge, addressing potential losses after multiple convolutional layers. In Chapter 5, a novel dataset and a semi-supervised annotation method are proposed, leveraging the faster region-based convolutional neural network. A deep learning architecture for semantic segmentation is developed to navigate challenges posed by complex backgrounds, demonstrating efficacy in both practical scenarios and benchmark datasets.

All proposed methodologies undergo testing on benchmark datasets across diverse

environments, affirming their capacity for precise plant segmentation and classification.

Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial neural networks
BN	Batch Normalization
BOVW	Bag of Visual Words
CDFL	Cross dropout focal loss
CNNs	Convolutional Neural Networks
Conv	Convolutional layer
D-SegNet	Densely Connected SegNet
DenseNet	Densely Connected Convolutional Networks
FC	Fully Convolutional
FCN	Fully Convolutional Network
FN	False Negative
FPN	Feature Pyramid Network
FPs	False positives
HSD	Holistic Self-Distillation
IoU	Intersection over Union

KD	Knowledge distillation
KL	Kullback-Leibler
KNN	K-nearest neighbour
LDAM	Label-distribution-aware margin
mACC	mean accuracy
mIoU	mean Intersection over Union
R-CNN	Region-based Convolutional Neural Network
ReLu	Rectified Linear Unit
ResNet	Residual Neural Network
ROC	Receiver Operating Characteristics
RoI)	Region of Interest
RPN	Region Proposal Network
SE	Squeeze and Excitation
SE-ResNet-50	Squeeze and Excitation Residual network 50
Self KD	Self-knowledge Distillation
SENet	Squeeze and Excitation Network
SOTA	State-of-the-art
SVM	Support Vector Machine
TN	True Negative
TPs	True positives
VGG	Visual Geometry Group

Chapter 1

Introduction

Autonomous Plant Image Segmentation and Classification play a vital role in precision agriculture [7]. With the advancements in computer vision and deep learning techniques, the automation of image analysis tasks has become essential for efficient and accurate decision-making in crop management. By analyzing plant images, farmers can identify diseases, nutrient deficiencies, or pest infestations at an early stage, allowing for timely interventions and targeted treatments. This proactive approach helps prevent yield losses and ensures crop quality, leading to optimized production and reduced economic losses. Moreover, autonomous plant image segmentation and classification enable precise resource allocation. By accurately describing different components of plants, farmers can optimize the application of fertilizers, pesticides, and water. This targeted approach minimizes resource wastage, reduces environmental impact, and promotes sustainable farming practices. In summary, the indispensability of autonomous plant image segmentation and classification in precision agriculture lies in its multifaceted contributions. By automating the analysis of plant images, farmers can proficiently monitor plant health, streamline resource allocation, alleviate manual burdens, and embrace sustainable agricultural methodologies. This technology-driven paradigm shift revolutionizes crop management, fostering improvements in productivity, profitability, and the overall sustainability of modern agricultural practices.

1.1 Background and Motivation

Deep learning has revolutionized the field of precision agriculture by enabling accurate segmentation and classification of plant images. With the advancements in computer vision and deep neural networks, researchers have been able to develop sophisticated models that can analyze and understand the intricate details of plant images, leading to precise agricultural practices. For instance, the application of deep learning for plant disease detection has achieved accuracy rates exceeding 95% in some studies, highlighting its potential to significantly reduce crop losses [110]. Deep learning aids in sustainable agriculture practices by optimizing resource use and reducing waste. For instance, deep learning models that manage irrigation systems can significantly reduce water usage. A UNESCO report highlighted that smart irrigation systems could increase water efficiency by up to 70%, showcasing the critical role of deep learning in promoting sustainability[123]. One of the significant applications of deep learning in precision agriculture is plant image segmentation [88, 106]. Recent research works highlight that image segmentation methods can detect plant diseases with an accuracy rate of over 90% [106, 60]. Early detection and accurate identification of diseases can significantly reduce crop losses. In fact, the Food and Agriculture Organization (FAO) [8]estimates that up to 40% of global crop yields are lost to pests and diseases annually, underscoring the potential impact of effective detection technologies. By segmenting plant images into different regions, such as leaves, stems, fruits, and weeds, farmers can gain valuable insights into plant health, growth patterns, and weed infestation levels. Deep learning models, such as convolutional neural networks (CNNs) [57], can learn to differentiate between various plant components and accurately delineate them in images, enabling targeted interventions and resource allocation. For instance, targeted application of agricultural inputs, guided by image classification insights, can reduce water usage by up to 30% and chemical usage by 20%, contributing to more sustainable farming practices [143]. Another crucial aspect of precision agriculture is plant image classification [97, 140]. By training deep learning models on large datasets, these

models can classify plant images into different classes, such as healthy plants, diseased plants, nutrient-deficient plants, or different plant species. This information assists farmers in making informed decisions regarding crop management, disease prevention, and optimizing resource utilization.

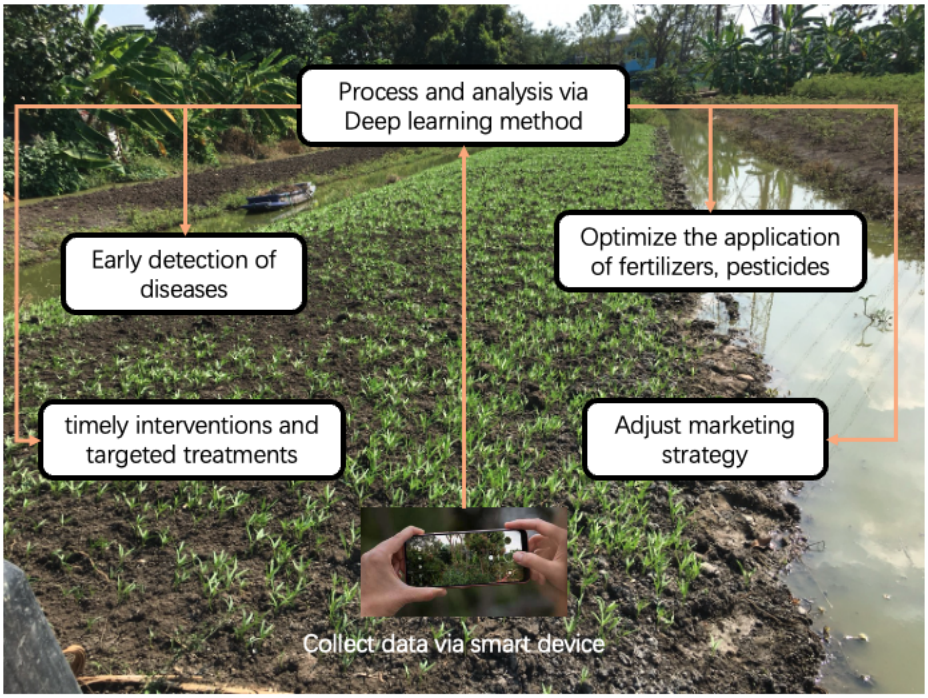


Figure 1.1: *The application of precision agriculture and deep learning.*

The incorporation of deep learning-driven plant image segmentation and classification within precision agriculture heralds a transformative approach to the cultivation and management practices of crops [7]. This innovative approach offers a multitude of benefits, transforming the landscape of agricultural practices. First and foremost, this integration empowers farmers and agricultural practitioners to monitor the health and growth status of their plants on a large scale with precision and efficiency. By harnessing the capabilities of deep learning models, early detection of diseases, stressors, or nutrient deficiencies becomes not only possible but also highly effective. As a result, swift and targeted interventions can be applied, ensuring the well-being of the crops and safeguarding against potential yield losses. Furthermore, the advantages

of deep learning in plant image segmentation and classification extend to the domain of resource management and conservation. Through the seamless integration of these technologies, precision agriculture can significantly reduce waste and environmental impact. The precise identification of areas affected by diseases or pest infestations allows for surgical and targeted spraying, minimizing the need for excessive use of chemicals. Additionally, the optimized application of fertilizers, guided by deep learning insights, ensures that nutrients are distributed efficiently, fostering healthier plants and reducing excess runoff, which can contaminate nearby ecosystems. This not only benefits the environment but also optimizes resource usage, saving both time and money for farmers and contributing to sustainable agricultural practices [46]. Lastly, deep learning models play a pivotal role in disease detection and elimination. By accurately identifying affected plants, these models enable targeted interventions, preventing the spread of diseases and reducing the competition for resources among plants. This, in turn, maximizes crop yield and quality, ensuring a bountiful harvest for farmers. Consequently, this optimizes both crop yield and quality, guaranteeing a plentiful harvest for farmers.

This thesis develops deep learning methods for autonomous plant image segmentation and classification. The proposed methods provide farmers with valuable insights and tools for effective crop management and resource allocation. These advancements contribute to more efficient and sustainable agricultural practices.

1.2 Aim and Objectives

The aim of this thesis is the accurate segmentation and classification of plant images in precision agriculture using indistinguishable plant dataset and publicly available datasets based on computer vision and deep learning methods. Validating a proposed method across different datasets ensures its robustness and ability to generalize to various real-world scenarios, beyond the specific conditions of a single dataset. The evaluation demonstrates its competitiveness and effectiveness across diverse applica-

tions. The main objectives are listed below:

- Develop deep learning algorithms for plant image classification and segmentation.
- Explore and develop deep learning algorithms which can alleviate the problem of imbalanced datasets based on comprehensive feature learning.
- Develop an effective classification framework based on knowledge distillation which reduces the model's complexity and computational costs while maintaining performance.
- Create a comprehensive plant dataset and perform detailed annotations to improve model robustness by including plant pixels under complex backgrounds.
- Develop an innovative segmentation framework trained using diverse plant images to precisely depict plant diversity across varying environmental conditions.
- Evaluate and validate thoroughly the proposed approaches over real images and benchmark datasets with respect to the state-of-the-art algorithms.

1.3 Contributions and Outline of the Thesis

The dissertation is described as six chapters. A brief overview of the content in each chapter is presented.

Chapter 2.

This chapter provides an overview of the concepts and algorithms pertaining to plant segmentation and classification. Additionally, it offers a brief introduction to the background knowledge of data annotation and detection, which are relevant to the work proposed in this thesis. The approaches to plant segmentation and classification are categorized into two main groups: (i) methods based on traditional computer vision and deep learning, and (ii) semantic segmentation algorithms and data annotation.

Both sets of techniques are comprehensively reviewed and discussed, with detailed introductions to some of the widely employed schemes.

Chapter 3.

A fully convolutional network with cross dropout focal loss for semantic segmentation analysis and comparison in this chapter. This chapter focuses on the development of fully convolutional networks and the experience of different loss functions. The loss function of segmentation algorithm updates weights through dropout. The proposed algorithm can be applied to general semantic segmentation tasks. A novel deep learning framework for semantic segmentation is proposed in the chapter.

The main contributions of this work are as follows.

- A novel improved fully convolutional network algorithm is introduced. The improvement of the fully convolutional network not only depends on the weights to adjust the loss but also keeps the statistic capability of the cross-entropy loss function.
- It is introduced that cross dropout focal loss updates weights based on the segmentation output per class after T dropout times.
- The proposed algorithm improves the segmentation performance, which is demonstrated over two popular semantic segmentation datasets, City-scapes [27] and PASCAL [38]. The results show that the improved fully convolutional network achieves better performance than the well-known fully convolutional network and other variations.

Chapter 4.

This chapter considers the problem of plant pathology classification and self-distillation methods in precision agriculture. The proposed holistic self-distillation method employs the Squeeze and Excitation Network (SENet) for feature extraction from images. SENet is an attention mechanism that adjusts the weights of features based on their importance, thereby enhancing classification accuracy. The approach begins by utilizing

SENet for feature extraction from images, followed by comprehensive self-distillation to optimize the network. Comprehensive self-distillation is a training technique that improves classification performance by using pseudo-labels generated by the network itself during training.

The main contributions of this work are as follows:

- The Holistic Self-Distillation (HSD) is a novel method to learn holistic knowledge from the teacher network through distilling feature maps and soft labels.
- The proposed HSD method employs the Squeeze and Excitation (SE) network to integrate feature information and soft labels. It can be applied on all SE networks due to similar constructions, e.g. SE-Residual networks.
- Extensive experiments are conducted on fine-grained publicly available plant pathology benchmark datasets to evaluate the performance of the HSD method. The efficiency of the HSD framework in providing a new direction of self-knowledge distillation is demonstrated.

Chapter 5.

A novel deep learning-based method is proposed for semantic segmentation of morning glory plant leaves. The method utilizes state-of-the-art Convolutional Neural Networks (CNNs) to extract hierarchical features and accurately classify each pixel of the leaf image. To facilitate the development and evaluation of various segmentation algorithms, a comprehensive benchmark dataset containing annotated morning glory plant leaf images is introduced. This dataset provides a standardized platform for researchers to compare the performance of different segmentation methods.

The main contributions of this chapter are the following:

- A new deep learning architecture, called D-SegNet is proposed, where the new feature is the use of dense blocks to augment the standard SegNet architecture. A concatenation between the layers in the dense block is introduced. The main

advantage of D-SegNet consists of the improved feature maps extracted from the images.

- A new and comprehensive dataset for the morning glory plant is collected by us and made public, which is beneficial for reproducible research. The data [79, 81], available on GitHub, contains original and mask images, which are ready for training and testing of algorithms for semantic image segmentation.

- Experimental analysis on the morning glory plant dataset and ImageCLEF (Pl@ntleaves) dataset [49] using the proposed approach and other semantic segmentation techniques.

- The performance of D-SegNet algorithm is evaluated and thoroughly validated over several metrics such as precision, recall, F1-score and Intersection over Union (IoU). Comparative results with the standard SegNet architecture are presented.

Chapter 6. This chapter summarizes the main findings for all methods and the main contributions presented in the dissertation. The direction and ideas of future work are proposed based on the previous work analyses.

1.4 Associated Publications

The author's work presented in this thesis has been published. These papers are the following:

- **Journal Papers**

  [J1] Jingxuan Su, Sean Anderson, Mahed Javed, Charoenchai Khompatraporn, Apinanthana Udomsakdigool, Lyudmila Mihaylova, "Plant leaf deep semantic segmentation and a novel benchmark dataset for morning glory plant harvesting", *Neurocomputing*, vol. 555, October 2023. Impact Factor: 6.

- **Peer-reviewed Conference Papers**

  [C1] Jingxuan Su, Sean Anderson, and Lyudmila S. Mihaylova, "A Deep Learning Method with Cross Dropout Focal Loss Function for Imbalanced Semantic Segmentation", In *Proceedings of the 2022 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, Germany, IEEE, 2022.

  [C2] Jingxuan Su, Sean Anderson, and Lyudmila S. Mihaylova, "Holistic Self-Distillation with the Squeeze and Excitation Network for Fine-grained Plant Pathology Classification", In *Proceedings of the 2023 26th International Conference on Information Fusion (FUSION)*, (pp. 1-7), Charleston, USA, IEEE, 2023.

Chapter 2

Literature Review

This chapter delves into a series of interconnected methodologies. The core contributions of this thesis are fundamentally grounded in the concepts and algorithms expounded upon within this chapter. The journey of image analysis commences with the manipulation of raw data, traversing through a sequence of distinct tasks such as segmentation and classification. Section 2.1 serves as a precursor, providing an overarching framework for understanding image processing. Section 2.2 rounds off this chapter by offering an overview of the classification method as applied in the realm of plant pathology, a subject expounded upon in greater detail in Chapter 3. The image segmentation as a pixel level classification will be reviewed following. In Section 2.3, a comprehensive survey unfolds, focusing on traditional image segmentation algorithms, which subsequently find practical application in Chapter 5. Section 2.4 unveils a panoramic exploration of contemporary deep learning approaches in the domain of image segmentation, meticulously considered in Chapters 3, 4, and 5. As a cornerstone of our work, it paves the way for modern techniques in this field.

2.1 Outline of Image Processing

Image processing involves converting an image into digital format and executing operations to extract valuable information, incorporating preprocessing and pixel classification as critical steps in this transformation [76]. The primary objective of image pre-processing is to eliminate irrelevant information in an image, restore useful real information, enhance the detectability of relevant information, and simplify data to improve the reliability of feature extraction, image segmentation, matching, and recognition. This includes tasks such as contrast enhancement and noise removal. Image augmentation is a crucial step in computer vision and plays a significant role in various applications such as medical imaging, industrial inspection, remote sensing, and plant disease detection. Image augmentation is the process of adjusting the contrast of acquired images to address issues related to variations in brightness, such as sunlight and shadows [74]. Colour transformations are employed to tackle lighting issues in image scenes. For instance, researchers used the normalized difference index (utilizing only the green and red channels) to reduce lighting effects and differentiate between plants and the background [120]. Filtering is another vital component of image augmentation. In agricultural applications, colour transformations and histogram equalization are utilized for plant leaf disease detection [162]. For instance, homomorphic filtering is a technique that minimizes lighting issues to a great extent and has been successfully applied to outdoor images in various environmental conditions [116].

Image pre-processing serves as a critical preparatory step for subsequent image analysis tasks. By eliminating noise, enhancing contrast, and addressing lighting variations, it ensures that the relevant information in an image is more pronounced and easier to detect. In the context of plant disease detection, this is particularly important as it enables accurate identification and segmentation of diseased areas on plant leaves. The application of image augmentation techniques like colour transformations and filtering has a wide range of implications beyond agriculture, including in the medical field, where it aids in the identification of anomalies in medical images, and in

industrial quality control for defect detection in manufactured products. The ability to improve the visibility of essential features in images is a fundamental aspect of computer vision, making image augmentation a vital tool in the field of image processing. Chapters 3, 4, and 5 each apply image pre-processing techniques, with comprehensive practical details elucidated in Chapter 5.

At its essence, image classification involves the assignment of a label to an image within a predefined classification set. In practical terms, our objective is to scrutinize an input image and furnish a label that accurately categorizes it. While the human visual system effortlessly discerns image classes, computers face the challenge of acquiring semantic information as seamlessly as the human eye. Traditional image classification relies on feature description and detection methods. While effective for straightforward image categorization, the complexity of real-world scenarios often overwhelms these traditional classification approaches. The image classification process encompasses key stages such as image preprocessing, extraction of image features, and the application of classifiers. Among these, image feature extraction stands out as a pivotal step. Traditional image classification approaches struggle with the enormity of image data, falling short of meeting the demands for both accuracy and speed in image classification. The advent of deep learning-based image classification methods marks a breakthrough, overcoming the limitations posed by traditional methods. Deep learning, a subset of machine learning, adeptly amalgamates low-level data features into abstract high-level representations, proving indispensable in domains like computer vision and natural language processing within the realm of artificial intelligence.

2.2 Traditional Image Classification Algorithms

Traditional image classification usually completely establishes an image recognition model, which generally includes basic steps such as input image data set, image preprocessing, feature extraction, training classifier and image classification and recognition [97]. Among traditional machine learning algorithms, image classification requires

extracting image features to describe the image, which is shown in Fig. 2.1. When the entire image is used as the input of the classification algorithm, the amount of data calculated by the algorithm is huge. Secondly, the image contains redundant information such as background, which will lead to a reduction in classification efficiency and accuracy. The main purpose of feature extraction is to reduce the dimension of the original image, map the original image to a low-dimensional feature space, and obtain low-dimensional sample features that can best reflect the essence of the image or distinguish it. After extracting different features, the features need to be fed into different machine learning algorithms as input. There are many traditional image classification algorithms, like the K-nearest neighbour (KNN) algorithm.

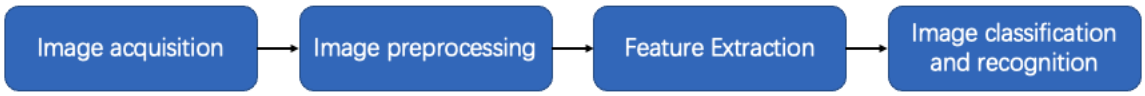


Figure 2.1: *Traditional image classification processing*

The KNN algorithm, short for K-nearest neighbor algorithm, functions as a supervised learning technique [56]. In essence, it entails identifying the K instances closest to a specific test sample A within a provided training set. Subsequently, it tallies the class counts among these K instances and assigns the class of sample A based on the class with the highest count. KNN is an online learning method, meaning each classification requires traversing all training samples. Three key elements define KNN: the value of K, the measure of distance, and the decision rule of classification. The K value is a pivotal hyperparameter directly influencing the model's performance. Optimal K value selection is crucial. A smaller K yields a more intricate and accurate model, albeit with a higher risk of overfitting, while a larger K results in a simpler model. An extreme scenario occurs when K equals the number of training samples (K=N), where the final test result corresponds to the class with the highest number of test samples, regardless of the test sample's inherent class. The distance measure determines the closeness

between a test sample and a training sample, forming the foundation for the selection of K samples. In KNN, when dealing with continuous features, the distance function typically employs either Manhattan distance (L1 distance) or Euclidean distance (L2 distance). Conversely, for discrete features, Hamming distance is commonly utilized. The classification decision rule in KNN involves selecting K training samples closest to the test sample, utilizing the concepts of K and distance mentioned earlier. The classification decision is then based on these K samples. The prevalent rule in KNN is the majority voting rule, where the class of the test sample is determined by the class with the highest count among the selected samples. However, it is important to note that this rule's efficacy heavily depends on the number of training samples. While KNN is a practical machine learning classification algorithm with a simple and easily understandable model, offering high classification accuracy for straightforward problems with minimal training time complexity, its drawbacks are apparent. The algorithm involves extensive calculations, leading to prolonged processing times, and demands significant storage space, making it inefficient as the number of feature dimensions increases.

2.3 Deep Learning Based Image Classification

In contrast to traditional image classification methods, deep learning approaches eliminate the need for manual feature description and extraction from target images. Instead, they autonomously learn features from training samples through neural networks, extracting higher-dimensional, abstract features. These features exhibit a close correlation with the classifiers, effectively addressing challenges related to manual feature extraction and classifier selection. Deep learning methods operate as end-to-end models. This chapter delves into a comprehensive analysis of key deep learning techniques in image classification, exploring the structure, advantages, and limitations of convolutional neural networks.

2.3.1 AlexNet

The concept of convolutional neural networks (CNN) comes from scientists at 1906s [146]. It lays the foundation for the development of image classification networks. The process involves taking original data as input and systematically abstracting it into feature representations for the target task layer by layer. This is achieved through a sequence of operations such as convolution, pooling, and nonlinear activation function mapping. The fundamental structure of CNN comprises an input layer, convolution layer, pooling layer, fully connected layer, and output layer. In image classification tasks, the output layer typically functions as a classifier, with commonly employed classifiers including Softmax, SVM, and others. LeCun et al. [89] proposed the LeNet-5 network, which contains 7 layers with Sigmoid activation function. It achieves a 0.8% false rate high performance on MNIST dataset. However, it has the disadvantages of small training data set size, weak generalization ability, and high training overhead. Continually, Krizhevsky et al. [84] proposed an AlexNet network with 5 convolutional layers. It uses ReLU activation function to solve the gradient vanishing problem caused by Sigmoid function since the network goes deeper. Following the emergence of the AlexNet network, the fundamental network structure is defined as a combination of convolution, ReLu nonlinear activation, MaxPooling and fully connected layers, which is shown at Fig. 2.2. The evolution of convolutional neural networks can be categorized into two distinct trends: the augmentation of network depth and the refinement of network architecture. Two GPUs are used to run separately, and the interaction between the two only exists at a specific network layer [84].

2.3.2 VGG

Simonyan et al. [149] introduced the Visual Geometry Group (VGG) network as an extension of the AlexNet, emphasizing the consequential impact of heightened network depth on ultimate performance outcomes. The VGG architecture incorporates two pivotal enhancements: 1) In contrast to AlexNet, VGG not only broadens the network

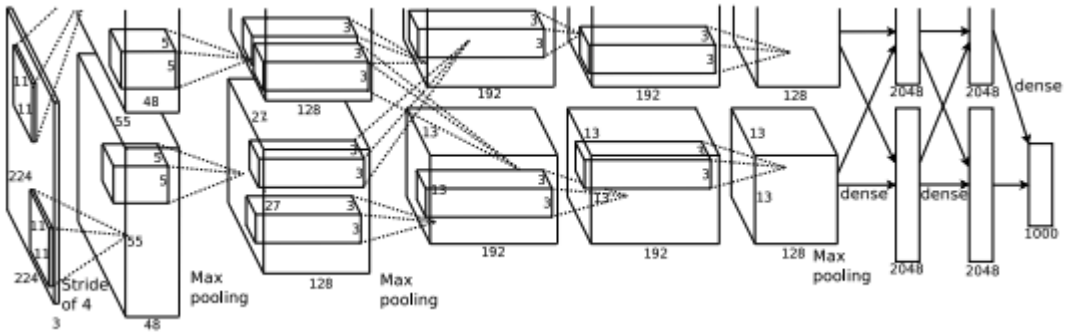


Figure 2.2: *The architecture of AlexNet [84].*

but also deepens it, exemplified by VGG-19's inclusion of 19 convolutional layers, a notable advancement beyond AlexNet's 5 layers. 2) VGG strategically underscores the effectiveness of employing multiple 3x3 convolutional filters as replacements for AlexNet's larger 7x7 or 11x11 filters. This design choice not only achieves superior performance but also mitigates computational costs. This architectural elegance has established VGG as the foundational network for seminal advancements in diverse computer vision tasks, including FCN for semantic segmentation and Faster R-CNN for object detection. The overarching objective of these architectural refinements is to augment network depth and enhance neural network efficacy while preserving a consistent receptive field.

The receptive field, as expounded by Luo et al. [100], is conceptually defined as the spatial extent within which pixels on the feature map, generated by each layer of a convolutional neural network, are correspondingly mapped back to the input image. An illustrative example is presented in Fig. 2.3. A commonly invoked analogy posits that a specific point on the feature map, relative to the size of the original image, delineates the region wherein the convolutional neural network features comprehend the input image. Opting for stacked small convolution kernels within a given receptive field proves more advantageous than employing larger counterparts. This strategic choice not only fosters the augmentation of network depth through the integration of multiple nonlinear layers but also ensures the development of a more intricate learning

model characterized by a reduced parameter count.

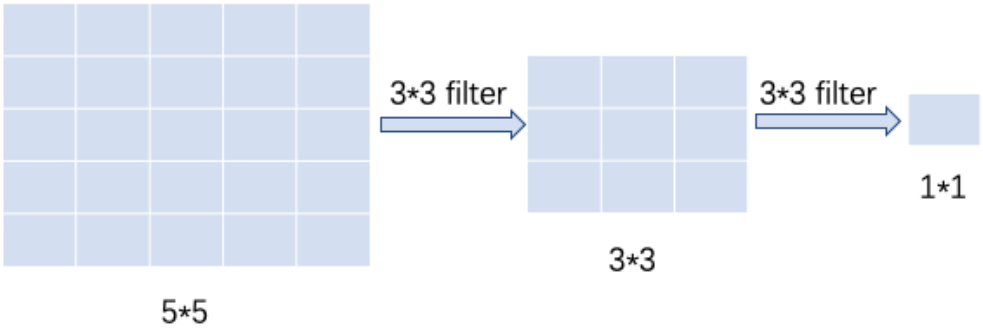


Figure 2.3: *The calculation of the receptive field is shown above.*

VGGNet distinguishes itself through the simplicity of its architectural design, characterized by consistent employment of 3x3 convolution kernels and 2x2 maximum pooling throughout the entire network. This methodological uniformity not only enhances the overall structural coherence but also serves as a pivotal element in underscoring the effectiveness of utilizing multiple small filter (3x3) convolutional layers in contrast to a singular, larger filter (5x5 or 7x7) convolutional layer. If the size of the input image is 5*5, after two 3*3 convolution kernels (with stride=1, padding=0), the receptive field size is 5*5.

In conclusion, the strengths of VGG lie in its architectural simplicity, methodological consistency, and the empirical substantiation that continuous expansion of the network's depth, particularly through the utilization of smaller convolutional filters, engenders discernible enhancements in both performance and feature representation. VGG, while exemplifying notable architectural advantages, is associated with increased computational demands and augmented parameterization, leading to heightened memory utilization. A significant proportion of the parameters is attributed to the initial fully connected layer, a facet accentuated by VGG's incorporation of three such fully connected layers. This characteristic contributes to the network's substantial resource requirements, particularly in terms of computational processing and memory alloca-

tion. The increased parameter count, predominantly emanating from the fully connected layers, necessitates a more judicious consideration of resource allocation and computational efficiency when deploying the VGG architecture in practical applications.

2.3.3 ResNet

He et al. [63] proposed a Residual Neural Network (ResNet), namely ResNet V1, to solve the degradation problem of deep network training, which is a milestone event in the history of deep learning. In 2014, the VGG architecture comprised a total of 19 layers, whereas the subsequent ResNet introduced in 2015 featured a substantially deeper configuration, extending to 152 layers. However, it is essential to recognize that superiority in network performance is not solely contingent on increased depth. ResNet introduces innovative architectural strategies that synergize with network depth. Notably, the incorporation of residual learning constitutes a pivotal architectural innovation within ResNet. This approach strategically leverages the residual connections, enabling the network to effectively utilize its depth for enhanced learning capabilities.

Empirical evidence underscores the critical influence of network depth on model performance. An increased number of network layers theoretically enables the extraction of more intricate feature patterns, suggesting improved results with deeper models. As the depth of the network increases, its expressive capacity becomes more potent. The convolutional kernel's primary function is to extract image features, however, a single convolutional kernel is inherently limited in representing the entirety of an image. Given this constraint, the utilization of multiple convolutional kernels becomes imperative. These diverse kernels can capture distinct features within the image, thereby enhancing the model's capability to learn intricate image characteristics. Consequently, employing an ample number of convolutional kernels and parameters proves essential for effectively characterizing the nuanced aspects of the original image. Consequently, the advantages of deep networks manifest in two key aspects. The hierarchical features

become more sophisticated as the network depth increases and greater network depth correlates with heightened expressive prowess.

Nevertheless, practical experiments reveal a phenomenon known as the degradation problem in deep networks. As network depth augments, there is an observed saturation or even a decline in network accuracy.

There are two primary factors contributing to this phenomenon. The first factor is associated with the deepening of the network, leading to the vanishing or exploding of the backpropagation gradient. Backpropagation [24] serves as the mechanism for adjusting the weights of the network, encompassing parameters such as the convolutional kernel values, weights of hidden layers, and biases. These adjustments are facilitated through the computation of the gradient, which involves the calculation of partial derivatives of the objective function, the sum of squares of the difference between the predicted and true values, with respect to the weights of each layer. This computation relies on the chain rule during the backpropagation process. The gradient undergoes a series of successive multiplications, potentially resulting in significant contraction or expansion. Assuming the error gradient for each layer is a value less than 1, during backpropagation, each forward propagation is scaled by an error gradient less than 1. With increasing network depth, the cumulative effect of multiplication by values less than 1 results in a gradient that converges toward zero. Conversely, if the gradient for each layer exceeds 1, a gradient explosion occurs, impeding convergence. The second factor contributing to the challenges is the degradation problem, even when the issue of gradient vanishing is addressed. Despite the mitigation of gradient disappearance, a deep layered network may exhibit reduced effectiveness compared to a shallower network. The original layers are derived from a shallower model that has undergone training. Although the original layers with attachment layers are set a constant value as previous model, a discernible disparity still emerges between the shallow and deep layers, resulting in distinct errors. These reasons are attributed to challenges related to gradient vanishing or explosion, thereby impeding the effective training of deep learning models.

ResNet introduces the concept of residual learning as a solution to the degradation problem observed in deep networks. A residual block encompasses two distinct paths: the first is denoted as $F(x)$, representing the residual component and is aptly termed the residual path, while the second, designated as x, constitutes an identity mapping and is referred to as the shortcut. The symbol $\bigoplus$ in the accompanying diagram signifies 'element-wise addition', necessitating a prerequisite condition that the dimensions of both $F(x)$ and x, engaged in the operation, must be identical. This ensures the viability of the element-wise addition operation, crucial for maintaining consistency in the sizes of the residual and identity paths within the residual block.

Within a stacked layer structure, where several layers are sequentially arranged, let the input be denoted as x. The learned features are recorded as $H(x)$. The objective is to enable the learning of the residual, denoted as $F(x) = H(x) - x$. In essence, the original learning feature becomes $F(x) = H(x) + x$. The reason behind this approach is that learning the residual is comparatively more straightforward than directly learning the original features. In instances where the residual is zero, the stacked layer merely performs identity mapping, ensuring, at the very least, that the network's performance does not degrade. In practice, the residual is not zero, allowing the stacked layer to learn new features based on the input, consequently yielding enhanced performance. Fig. 2.4 illustrates the structure of residual learning, characterized by a shortcut connection.

ResNet incorporates an identity input x to the output, ensuring that each residual module retains access to the original input. This strategic inclusion prevents the loss of essential information in the learning process. Notably, the fundamental departure from conventional approaches lies in the ResNet's objective. Instead of expecting each layer to directly conform to the desired feature map, the residual module is designed to learn the disparity between the output and the input. This paradigm shift simplifies the learning task, requiring less information gain for effective model training.

ResNet leverages deep neural networks while concurrently circumventing challenges associated with gradient dissipation and degradation. Despite the apparent depth of ResNet, a substantial portion of its network layers does not significantly contribute

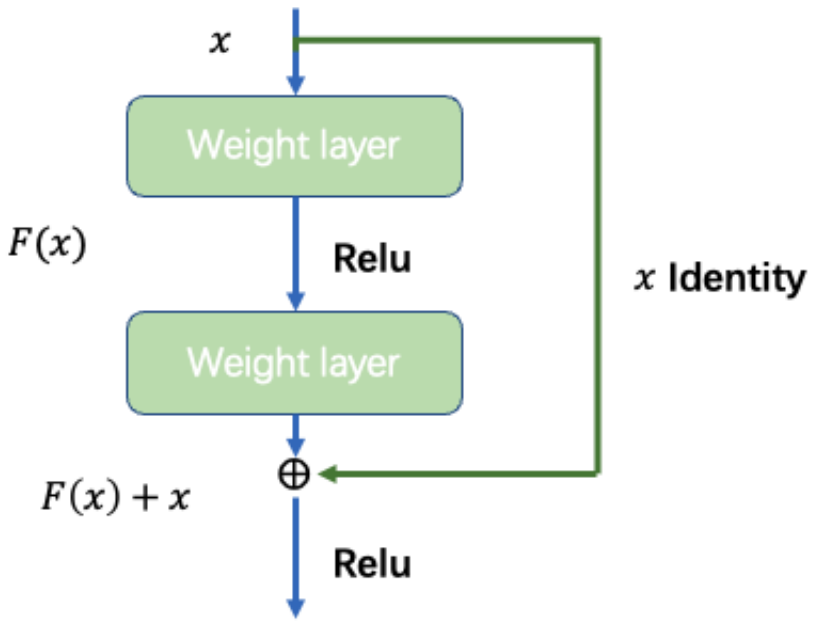


Figure 2.4: *A general residual block of residual learning.*

to the learning process due to their primary role in preventing model degradation and mitigating substantial errors. Furthermore, it is crucial to note that while the residual connections in ResNet alleviate issues related to gradient disappearance or explosion and network degradation, they do not provide a complete solution but rather a mitigating mechanism.

2.3.4 DenseNet

Cornell University's Densely Connected Convolutional Networks (DenseNet) [68] further extends the idea of ResNet, which not only provides the connection between layers but also provides the bypass connection for all previous layers. He et al. [63] posited a fundamental assumption when introducing ResNet: if a deeper network encompasses several additional layers compared to a shallower network and is proficient in learning identity mapping, the performance of the model trained by the deeper network should not be inferior to that of the shallower network. Simply, augmenting a network with ad-

ditional layers capable of learning identity mapping results in the new network's worst-case scenario being that these layers, post-training, essentially function as identity mappings without detrimentally affecting the original network's performance. When numerous layers are discarded throughout the training process, the ability of ResNet's convergence remains activated, indicating the obvious redundancy in the ResNet architecture. A similar assumption was made with the introduction of DenseNet: rather than redundantly learning features multiple times, extracting features through feature reuse offers a more efficient approach.

DenseNet employs a distinctive paradigm of dense connectivity wherein every layer within the network establishes a direct connection with its antecedent layer, facilitating the efficient reuse of features. Simultaneously, each layer is purposefully crafted to exhibit a characteristic narrowness, thereby mitigating redundancy; this design philosophy ensures that only a minimal set of features is acquired. The process of concatenation puts all output feature maps emanating from layers x_0 to x_{l-1} through channel-wise connections. The employed nonlinear transformation H in this context is a composite operation comprising Batch Normalization (BN), Rectified Linear Unit (ReLU), and a Convolutional layer (Conv) with a kernel size of 3×3 [68].

The initial impression conveyed by the term "dense connectivity" may lead one to anticipate a substantial augmentation in the parameters and computational load of the network. However, the operational efficiency of DenseNet surpasses that of other networks. This efficiency is grounded in the optimization of computational load at each network layer and the judicious reuse of features. In practice, each layer within the DenseNet architecture is tasked with acquiring a limited set of features, thereby effecting a noteworthy diminution in both parameter quantity and computational demand. This strategic reduction in the scope of feature acquisition per layer contributes significantly to the network's overall efficiency, countering the intuitive expectation of heightened computational complexity associated with dense connections.

Notably, the dense connectivity constitutes a marked departure from conventional network architectures. The extremity of this approach is exemplified by the scenario

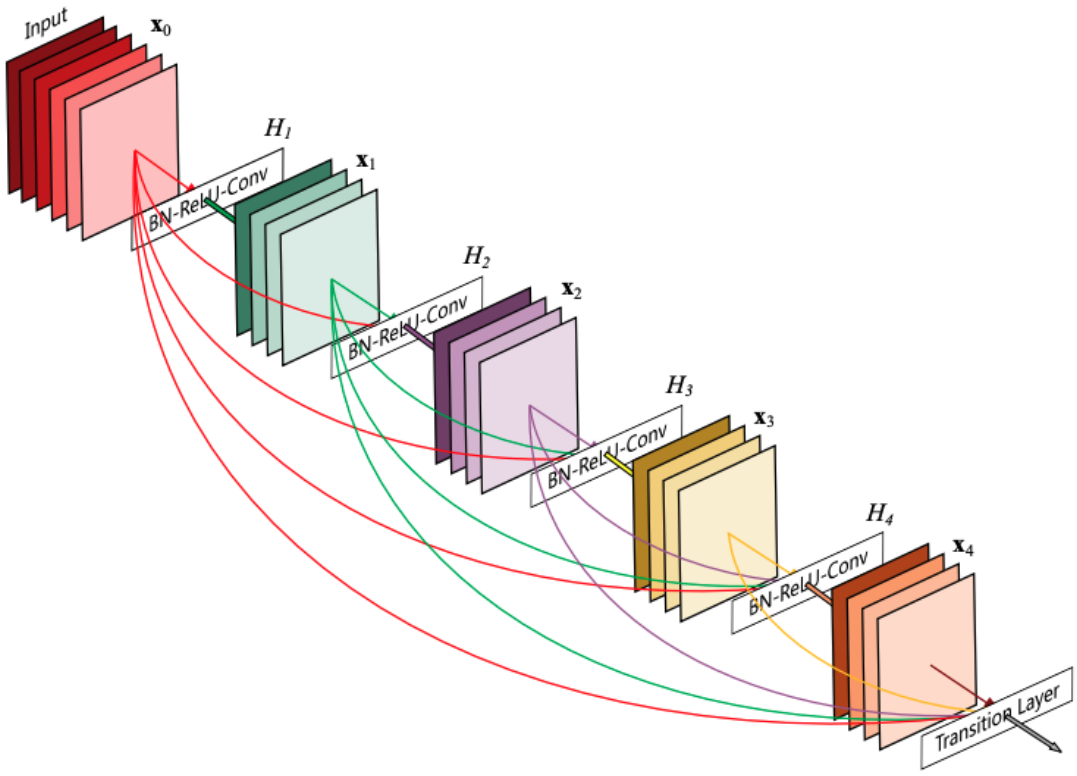


Figure 2.5: *The process of concatenation [68].*

where each layer exclusively learns a solitary feature map. These dual attributes, namely the dense connectivity and the intentional narrowness of each layer, constitute the principal difference between DenseNet and other extant networks. It is imperative to underscore that the deliberate narrowness of layers within the network would be unattainable without the incorporation of dense connectivity.

Given that the concatenation operation is requisite for the integration of feature maps from distinct layers within the DenseNet framework, it becomes imperative to uphold uniformity in feature size across these diverse layers. This condition, however, imposes constraints on the integration of down sampling procedures within the network. To circumvent this limitation and facilitate down sampling, the author strategically divides DenseNet into multiple dense blocks, as depicted below:

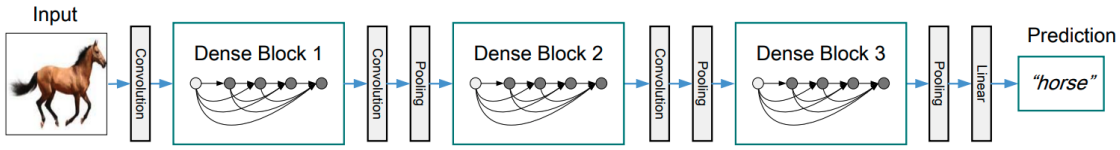


Figure 2.6: *A deep DenseNet predicts a horse image, which includes three dense blocks and transition layers [68].*

Maintaining a consistent feature size within each dense block is imperative, prompting the introduction of transition layers between distinct dense blocks to facilitate down sampling. In the experimental framework devised by the author, the transition layer is structured to include Batch Normalization (BN), a Convolutional layer (Conv) with a kernel size of 1×1, and a subsequent 2×2 average-pooling operation. This configuration ensures the effective implementation of down sampling between consecutive dense blocks. The convolution layer serves to reduce the dimension of the input feature maps to half of their original size, while the pooling layer further decreases the dimension of the feature maps by half. This collective reduction in dimension is strategically employed to curtail the size of feature maps transferred between Dense Blocks, thereby enhancing computational efficiency.

DenseNet, as an alternative convolutional neural network characterized by increased

layer depth, encompasses several notable advantages. It exhibits a reduced parameter count compared to ResNet. In attaining equivalent accuracy levels on the ImageNet classification dataset, DenseNet necessitates fewer than half the parameters required by ResNet. This characteristic bears practical significance for industry applications, wherein the deployment of more compact models not only serves to conserve bandwidth but also alleviates storage overhead substantially. Additionally, the introduction of dense connectivity augments feature reuse through bypass connections. Certain features extracted by earlier layers retain direct utility for subsequent, deeper layers in the network architecture. Even the transition layer extensively incorporates features of all layers in previous dense blocks. Moreover, the network demonstrates enhanced trainability and imparts a discernible regularization effect. It effectively mitigates issues associated with gradient vanishing and model degradation. The features derived from each layer within the neural network represent a nonlinear transformation of the input data. With increasing depth, the compounding of nonlinear functions leads to a progressive escalation in the complexity of the overall transformation. In contrast to conventional neural network classifiers that predominantly rely on the features of the final layer characterized by maximal network complexity, DenseNet possesses the distinctive ability to comprehensively exploit shallow features characterized by lower complexity. This characteristic facilitates the extraction of a smoother decision function, thereby contributing to enhanced generalization performance.

2.4 Traditional Image Segmentation Algorithms

Machine vision technology has been widely applied and researched in agriculture for the identification and detection of plants, including species and disease classification. Despite some significant challenges to be discussed below, it has demonstrated promising success in many case studies of plant pathology identification systems [61, 4]. After decades of research, machine vision has enhanced the quality management of plant pathology identification. Machine vision technology is also utilized in other agricultural

Table 2.1: *Summary of plant image classification methods*

Method Type	Traditional-based				
Specific Methods	KNN [65]	Naive Bayes[138, 115]	Support Vector Machine [18, 126]		
Advantage	Simple model and high accuracy in classifying simple problems	Insensitive to missing data and efficiency	Small number of samples needed linearly separable by kernel function mapping.		
Disadvantage	Require a lot of computation and storage space	Poor results for correlation features	Computational memory and time consumption are large when training sample is huge.		
Method Type	Deep learning-based				
Specific Methods	CNN [99]	AlexNet [19]	VGG [73]	ResNet [86]	DenseNet [159]
Advantage	It works well for images having good contrast between regions.	Share convolution kernels and reduce network parameters	Simply network structure and less parameters	Alieviates the problems of gradient vanishing and degradation when the network goes deep.	Save parameters and computation, ability of anti-overfitting, strong generalization performance
Disadvantage	Convergence to local minimum rather than global minimum	Many outputs of hidden layers are 0 which makes the network become sparse.	Expensive computing and storage	The problems can not be fully solved.	Memory usage

applications, such as grading and harvesting fruits [152, 169, 1]. Numerous researchers have developed image processing techniques and deep learning methods as guidance for machine vision, working in different fields and environments.

Threshold-based Segmentation

The fundamental concept behind the threshold method [12] involves determining one or more grayscale thresholds based on the grayscale characteristics of the image. These thresholds are then used to compare the grayscale values of each pixel within the image. Subsequently, the pixels are categorized into appropriate groups based on the results of these comparisons. Consequently, the pivotal step in this methodology is to determine the optimal grayscale threshold according to a specific criterion function. The threshold method is particularly well-suited for images in which the target and background exhibit distinct grayscale ranges. If the image contains only two classes, the target and the background, then a single threshold suffices for segmentation. In this scenario, it becomes a single-threshold segmentation. However, when the image contains multiple targets that require extraction, using a single threshold leads to mixed results. In such cases, multiple thresholds must be employed to separate each target effectively. This approach is referred to as multi-threshold segmentation. The advantage of the threshold segmentation method is straightforward and efficient in the calculation process. It relies solely on the gray values of individual pixels, which simplifies the method. The Otsu algorithm, also recognized as the threshold-based method, was introduced by the Japanese scholar Otsu in 1979 [114, 125, 41]. This algorithm offers an efficient approach to image binarization, leveraging thresholds to partition the original image into two distinct components: the foreground and the background. The essential idea of the algorithm is to maximize the inter-class variance. From the preceding discussion, it is evident that the crux of the threshold segmentation method lies in selecting the appropriate threshold. One promising direction for enhancing this method is the integration of intelligent genetic algorithms for optimizing threshold selection. This may represent the future evolution of image segmentation methods based on threshold

segmentation.

Edge-based Segmentation

Image segmentation algorithms based on edge detection [183] aim to address the segmentation challenge by identifying boundaries that separate distinct regions. This method can be considered one of the earliest and most extensively explored techniques in the field. Typically, the gray values of pixels at the boundaries of different areas exhibit significant changes. When an image is transformed from the spatial domain to the frequency domain using Fourier analysis, these boundaries correspond to high-frequency components, making edge detection a relatively straightforward algorithm. Edge detection techniques can be broadly categorized into two approaches: serial edge detection and parallel edge detection. Serial edge detection relies on the verification outcomes of prior pixels to determine if the current pixel belongs to an edge point. In contrast, parallel edge detection assesses a pixel's edge membership based on the pixel itself and some neighbouring pixels. The simplest edge detection method is the parallel differential operator method, like Canny operator [112, 164] and Sobel operator [118, 175]. It capitalizes on the abrupt transitions in pixel values between adjacent regions and utilizes first-order or second-order derivatives to identify edge points. In recent years, alternative methods have been developed, including those based on surface fitting, boundary curve fitting, reaction-diffusion equations, serial boundary search, and deformation models. The advantages and drawbacks of edge detection are noteworthy. It excels in precise edge localization and rapid processing speed. However, it falls short in ensuring edge continuity and closure. Moreover, it tends to produce numerous broken edges in high-detail areas, hindering the formation of large coherent regions and is unsuitable for dividing intricate regions into smaller, manageable fragments. These two challenges value that edge detection yields only edge points, falling short of a comprehensive image segmentation process. As such, post-processing or the integration of complementary algorithms is necessary to achieve the complete segmentation task. In future research, key areas of focus will include the selection of

adaptive thresholds for initial edge point extraction, the identification of larger regions for hierarchical image segmentation, and the development of techniques to distinguish significant edges. These aspects will prove pivotal in advancing image segmentation methodologies.

Region-based Segmentation

The region-based segmentation method [59] is a segmentation technique that involves directly identifying regions within an image. There are two fundamental approaches to region-based extraction methods: one is region growing, which initiates from a single pixel and progressively merges neighbouring pixels to form the desired segmentation region. The other approach begins with the overall image and progressively partitions it into the required segmented areas. Region growth [112] initiates from a set of seed pixels representing different growth areas. It proceeds by merging eligible neighbouring pixels into the growth area represented by the seed pixels and continues using the newly added pixels as new seed pixels. This merging process continues until no new pixels meeting the specified conditions are found. The crucial aspects of this method involve selecting the appropriate initial seed pixels and establishing sensible growth criteria. The region growth algorithm addresses three fundamental challenges: selecting or determining a set of seed pixels that accurately represent the desired area; defining the criteria for incorporating adjacent pixels during the growth process; and establishing conditions or rules for terminating the growth process. Region growing commences from a specific pixel or a set of pixels, ultimately culminating in the entire region's identification, facilitating the extraction of the desired target. Conversely, region splitting and merging [6] can be considered as the inverse procedure of region growth. Starting with the entire image, this approach involves continuously splitting it into sub-regions. Subsequently, the foreground regions are merged to isolate the foreground target that requires segmentation, ultimately resulting in the successful extraction of the target. In practical applications, it is common to combine the region growing algorithm with the region splitting and merging algorithm. This hybrid approach proves more effective

in segmenting complex scenes characterized by intricate objects or natural scenes, as well as other image segmentation scenarios where prior knowledge is limited.

The watershed algorithm [186, 31, 142] is a straightforward and intuitive region-based method for image segmentation. It operates by considering the image as a representation of topological features, akin to a landscape with mountains and lakes. In this analogy, the mountains are surrounded by water, forming a watershed. The watershed segmentation method is rooted in mathematical morphology and topology theory. It treats the image as a topological terrain, where each pixel's gray value corresponds to its elevation. In this context, each local minimum and its associated influence area are referred to as catchment basins, and the boundaries between these basins are the watersheds. The concept of watersheds is best understood through the metaphor of an immersion process. That means puncturing a small hole at the site of each local minimum and slowly submerging the entire model in water. As immersion deepens, the influence areas of local minima expand, and watersheds form where basins converge. The watershed algorithm is particularly effective at detecting weak edges in images. While it can sometimes lead to over-segmentation due to noise or subtle grayscale variations on object surfaces, it excels at ensuring the capture of closed and continuous edges. Moreover, the closed catchment basins obtained through the watershed algorithm provide analyzable characteristics of regional images.

Clustering-based segmentation

Clustering-based segmentation [26] is a prevalent technique within the realm of medical image segmentation and is akin to a statistical approach that operates independently of a training sample set. Commonly employed clustering methods encompass the K-means clustering method, the fuzzy c-means clustering algorithm, the maximum expectation algorithm, and more. Nevertheless, while the clustering method doesn't necessitate a training dataset, it crucially depends on the specification of initial parameters, and the resulting segmentation outcomes are notably susceptible to the initial parameter settings. K-means clustering [35, 113] stands as one of the most frequently employed

clustering algorithms, with its roots tracing back to signal processing. The primary objective of this algorithm is to partition data points into K clusters, identifying the centre of each cluster while minimizing a designated metric. Its paramount advantage lies in its simplicity, ease of comprehension, and expeditious processing speed. Nonetheless, a notable drawback of K-means is its applicability exclusively to continuous data, requiring the user to pre-define the number of clusters before initiating the clustering process. The K-means clustering algorithm offers a significant advantage due to its speed, simplicity, and remarkable efficiency, making it highly suitable and scalable for handling large data sets [23]. It exhibits a time complexity that approaches linearity, rendering it exceptionally well-suited for mining extensive datasets. However, the K-means algorithm also presents certain disadvantages. Foremost, it lacks explicit selection criteria for determining the number of clusters, making it a challenging task to estimate the optimal value of K. Additionally, within the K-means algorithm framework, each iteration involves traversing all the samples, resulting in substantial computational costs. Lastly, K-means is rooted in a distance-based partitioning approach, restricting its applicability to convex datasets and rendering it less suitable for clustering non-convex clusters.

2.5 Regional Proposal-based Deep Learning Segmentation

Recent years, convolutional networks drive the development of recognition [84, 149]. It improves the image classification and object detection. However, the task of semantic segmentation is different from the above tasks. It is a space-intensive prediction task in computer vision. Semantic segmentation is a classification at the pixel level. Pixels belonging to the same class are classified into one class. Therefore, semantic segmentation understands images from the pixel level. That means each pixel needs to be predicted. Previously, each pixel's label method is complicated, which has excellent defects in terms of speed and accuracy. Before deep learning methods became popu-

Table 2.2: *Summary of plant image segmentation methods*

Method Type	Advantage	Disadvantage	Specific Methods
Threshold-based	(1) Simple calculation	(1) It only depends on the pixel grey value and does not consider spatial features	Otsu threshold [114, 125, 41]
	(2) High efficiency	(2) Sensitive to noise and has low robustness	
Edge-based	(1) Accurate edge positioning	(1) The continuity and closure of edges cannot be guaranteed.	Canny edge detection [112, 164]& Sobel edge detection [118, 175]
	(2) Fast computing	(2) It is not suitable for too many edges.	
Region-based	(1) It works well for images having good contrast between regions.	(1) Complicated algorithm	Watershed [186, 31, 142]
	(2) It is suitable for complex images.	(2) Heavy computing	
Clustering-based	(1) It can eliminate noisy spots.	(1) Sensitive for the noise and grey inhomogeneity	K-means [35, 113] & Fuzzy clustering [105]
	(2) It can obtain more homogeneous regions.	(2) Difficult to determine the initial parameters	
Learning-based	(1) Strong learning ability	(1) Large-scale training data and complex networks	RNN (Recurrent Neural Networks) [133] & SVM (Support Vector Machines) [55]
	(2) High accuracy and strong portability	(2) Poor interpretability and computationally expensive	SegNet (Segmentation Networks) [5, 82]& FCN (Fully Convolutional Networks) [71, 102]

lar, semantic segmentation methods such as TextonForest and random forest classifiers were more commonly used methods. However, after the popularity of deep convolutional networks, deep learning methods have improved a lot compared to traditional methods. This section focuses on the state-of-the-art deep learning algorithms, such as Fully Convolutional Networks, and SegNet.

2.5.1 Region-based Convolutional Neural Network (R-CNN)

In 2012, Ciresan[25] used CNN to challenge the semantic segmentation task. Ciresan adopts a sliding window method, taking a small image patch centered on each pixel and inputting it into CNN to predict the semantic label of the pixel. This is a very meaningful attempt, breaking the precedent of CNN only being used for target classification, and the author also achieved state-of-art achievements that year. However, the shortcomings of this method are obvious. It is necessary to traverse each pixel to extract patches for training and prediction, which is slow and time-consuming. Meanwhile, the appropriate window size is hard to set. If it is too small, it will lack contextual information; if it is too large, it will increase the amount of calculations; there are undoubtedly a lot of redundant calculations between many windows. The shortcomings of the sliding window method also exist in the field of target detection, and researchers use region based methods to solve this problem. In the field of semantic segmentation, several algorithms based on region selection have gradually extended to the field of semantic segmentation from previous work on target detection. The R-CNN meticulously picks through the input image, confirming region proposals, and employs the formidable CNN framework to make precise predictions about the objects within each proposed region [25].

In 2015, Professor Girshick of the University of Berkeley and others jointly proposed the first deep learning model applied in the direction of target detection: Region-based Convolutional Neural Network (R-CNN)[48]. The R-CNN architecture is shown in Fig. 2.7. The main process of R-CNN is extracting 2000 region proposals using selective

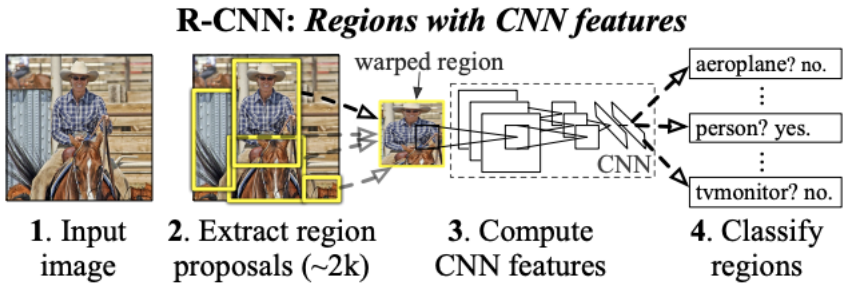


Figure 2.7: *The R-CNN framework [25].*

search[48, 168]. For each proposal, it is warped into a fixed size for a CNN model. As a feature extractor, the CNN produces a 4096-dimensional vector to fed into linear Support Vector Machine (SVM). Finally, the object is predicted in each region and classified as a class. However, there still have obvious drawbacks with R-CNN. It is time-consuming as 2000 region proposals need to be classified in each image. Testing per image expends 47 seconds. Thus, real-time implementation is impossible. There has no learning for the selective search stage, which could generate the awful candidate for region proposals.

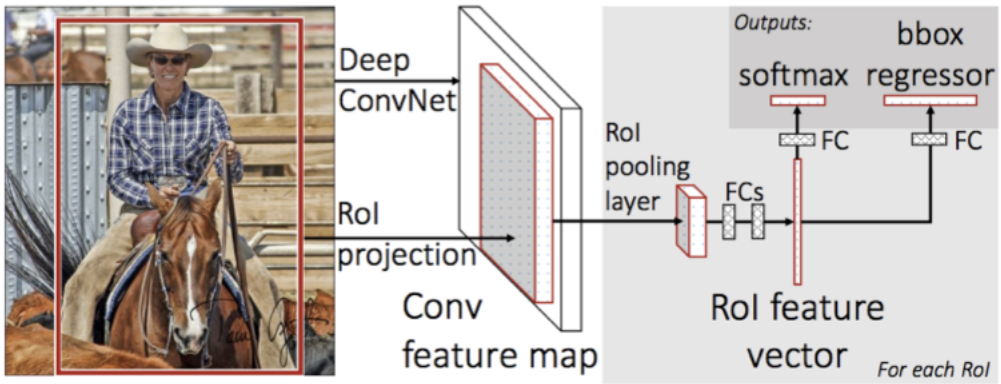


Figure 2.8: *The architecture of Fast R-CNN network [47].*

2.5.2 Fast R-CNN

In 2015, R. Girshick et al. proposed an advancement detector, Fast R-CNN[47]. It is similar to the R-CNN, but with high accuracy and faster prediction. Instead of extracting region proposals, the input image is fed into CNN that achieves a convolutional feature map. Thus, the convolution operation for each image only needs to do once. It decreases the computation times. Then, the region proposals are identified from the convolutional feature map. For each proposal, a fix-length feature vector is generated from the feature map on the Region of Interest (RoI) pooling layer, which fed into a fully connected layer. The class of proposals is predicted in the softmax layer and the offset value of the bounding box. Detector and bounding box regressor are training simultaneously, as shown in Fig. 2.8. The architecture of Fast R-CNN network starts from feature extraction, omitting the selection of region. It directly uses a neural network to perform operations on the entire image. This network can efficiently complete feature extraction and predict aim objects [48]. Fast R-CNN successfully exceeds R-CNN in performance [62]. On VOC07 dataset [37], Fast R-CNN improve 11.5% in MAP. Meanwhile, the speed is 200 times faster than R-CNN. However, the speed still is limited due to the detection of region proposals. Both of the R-CNN and Faster R-CNN algorithms identify region proposals by the selective search. The selective search slows down the processing speed. Therefore, S. Ren et al. came up with Faster R-CNN detector that erases the selective search process and detects the network's region proposals.

2.5.3 Faster R-CNN

The Faster R-CNN proposed in 2016 made breakthrough progress that replaced the most time-consuming and fatal part of its predecessors: the selective search algorithm. Faster R-CNN is an end-to-end real-time detector [131]. This object detection system is composed of a deep fully convolutional network, Region Proposal Network (RPN), and Fast R-CNN detector, as shown in Fig. 2.9. It replaces the selective search al-

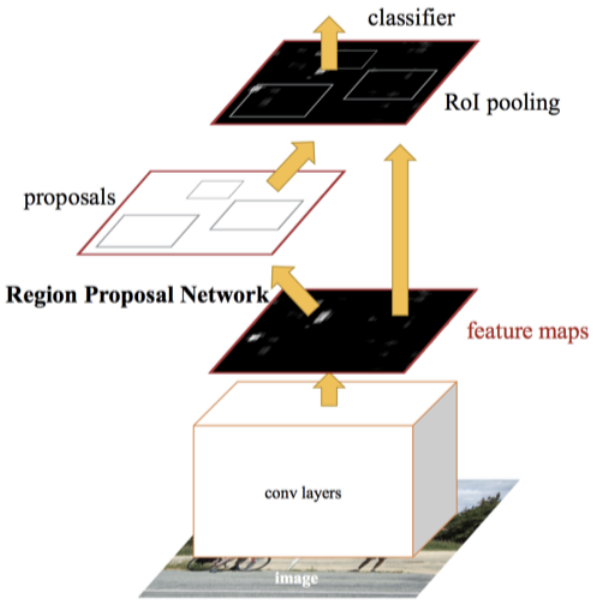


Figure 2.9: *The diagram of Faster R-CNN network [131].*

gorithm with region proposal network to select regions and reduces the time from 2s to 10ms [131]. As a CNN based object detection approach, Faster R-CNN extra feature maps from an image using fundamental convolutional layers. The feature map is shared with the RPN layer and the Fully Convolutional (FC) layer. RPN generates the region proposals, which decides the property of anchors, positive or negative. The bounding box regression will fix anchors to get the precise proposals. The RoI pooling layer collects feature maps from convolutional layers and fixed proposals. After integrating this information, the proposal feature maps are extracted, which are conveyed to FC layers. Finally, it is calculated the proposal class and obtained the final precise bounding box simultaneously. The RPN, as the main contribution, brings down the processing time of region proposals. It tells the Fast R-CNN were to detect using the 'attention mechanisms. Meanwhile, it shares layers to the following detection steps. Although Faster R-CNN breaks the speed bottleneck of algorithms mentioned above, the redundancy computation still exists in the following detection stage.

2.6 Encoder-Decoder Based Deep Learning Segmentation

The Encoder-Decoder based Segmentation Model is a popular architecture in deep learning for semantic segmentation tasks. It comprises two main components: the encoder and the decoder. The encoder component is typically a convolutional neural network (CNN) that processes the input image in a hierarchical manner. It extracts high-level features by passing the input image through a series of convolutional layers. Each layer captures increasingly complex and abstract features from the image. The decoder component is designed to generate a segmentation mask or map that corresponds to the original input image. It takes the high-level features extracted by the encoder and progressively upsamples or expands them using techniques such as transposed convolutions or bilinear interpolation. The goal is to recover the spatial information lost during the encoding process. The final output of the model is a segmentation map, where each pixel is assigned a label corresponding to a particular class or class. The encoder-decoder architecture allows for end-to-end training, where the model learns to extract meaningful features from the input image and reconstruct the segmentation mask. This architecture is effective for segmentation tasks as it combines the feature extraction capabilities of the encoder with the spatial recovery capabilities of the decoder, enabling accurate and detailed segmentation of objects or regions within the input image.

2.6.1 Fully Convolutional Network

Typically, in convolutional neural network (CNN) architectures, several fully connected layers follow the convolutional layer to transform the feature map into a fixed-length feature vector. The conventional CNN structure, as exemplified by AlexNet, is well-suited for tasks such as image-level classification and regression. These tasks inherently seek to derive a numerical depiction, specifically a probability distribution, for the entire

input image. For instance, in the case of AlexNet's ImageNet model, the output is a 1000-dimensional vector signifying the probability of the input image belonging to each class, determined through softmax normalization.

Fully Convolutional Network (FCN) [96] addresses semantic segmentation challenges by enabling pixel-level classification of images. In classical CNN architectures, the standard fully connected layer condenses the original image, converting a two-dimensional matrix into a one-dimensional vector, causing the loss of valuable spatial information. This yields a scalar output denoting the classification label. In contrast, FCN is adept at handling inputs of varying dimensions. Through the use of deconvolution layers, it undertakes upsampling on the feature map derived from the final convolutional layer, thereby restoring the output dimensions to match the original image. Particularly noteworthy is FCN facilitates the generation of predictions for each pixel, all the while preserving the spatial information inherent in the original input image. Ultimately, the pixel-wise classification is executed on the upsampled feature map. Fig.2.10 denotes the schematic of fully convolutional neural network. Fully convolutional networks can efficiently upsampling via deconvolutional layers and improve upsampling roughness through skip connection. This network structure largely helps in dense predictions like semantic segmentation [96]. However, directly sampling the image size from the final feature map results in a notably coarse accuracy. This phenomenon stems from the deeper layers of the network, which can learn intricate features but, concurrently, risk the loss of crucial spatial location information. Notably, shallower layers retain more precise location information in their output. To enhance results, the skip connection is introduced that combines both deeper and shallower layer outputs, which is shown in Fig. 2.11. The process of feature extraction is referred to as the encoder, which is the phase in FCN where the preceding feature maps become smaller. The process of upsampling or deconvolution carried out later is referred to as the decoder. Within the decoder, the image is restored to its original size [96].

In essence, FCN distinguishes itself from CNN by substituting the final fully con-

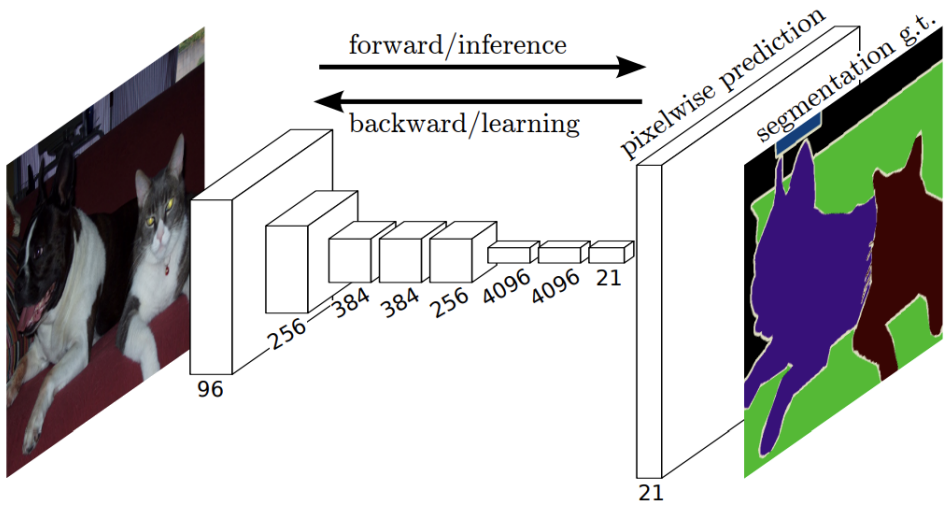


Figure 2.10: *The Fully convolutional networks [96].*

nected layer of CNN with a convolutional layer, resulting in an output that comprises an image with assigned labels. The FCN recovers the categorization of each pixel from abstract features, thereby extending the classification from the image level to the pixel level. The principal distinction between fully connected layers and convolutional layers lies in the fact that neurons within the convolutional layer establish connections solely with local regions in the input data, with parameter sharing among neurons within the same convolutional column. Despite this structural variance, both layer types share a commonality in their computation of dot products by neurons, rendering their functional forms analogous. Consequently, it becomes plausible to reciprocally convert between these layer types. For any given convolutional layer, a corresponding fully connected layer exists capable of executing an equivalent forward propagation function. The weight matrix associated with such a fully connected layer is characterized by its substantial size, comprising predominantly zero elements with specific non-zero blocks. Notably, within the majority of these non-zero blocks, the elements share uniform values. Conversely, any fully connected layer can be transformed into a convolutional layer. Specifically, the input data volume's size is treated as a filter volume, resembling the spatial arrangement of the original fully connected layer. This

transformation ensures that the output from the convolutional layer aligns closely with the output of the initial fully connected layer.

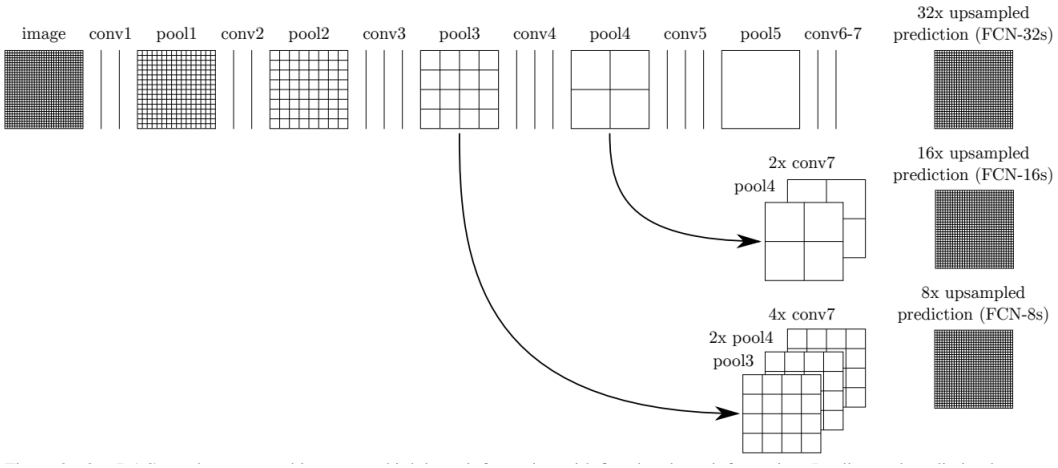


Figure 2.11: *The process of feature extraction [96].*

The forefront of deep learning models for semantic segmentation is largely characterized by architectures rooted in the FCN, which contributes to the substantial advancement in overall accuracy. Functioning as a pioneering model in semantic segmentation, FCN is purposefully constructed as an end-to-end network tasked with predicting pixel-level information, effectively translating input pixels into corresponding output pixels. This innovation marked a paradigm shift by fundamentally altering the conventional approach that necessitated window-based transformations for semantic segmentation tasks, effectively transforming the task into an image classification problem. Notably, FCN eliminates the reliance on fully connected layers traditionally employed in image classification, like CNNs, opting instead for the exclusive use of convolutional layers throughout the network architecture. However, the results of FCN, even with an 8x upsampling, remain insufficiently precise. While this represents a notable improvement compared to a 32x upsampling, the results remain characterized by blurriness and smoothness, lacking sensitivity to intricate details within the image. The classification of each pixel occurs without a comprehensive consideration of inter-pixel relationships. Moreover, the network omits the conventional spatial reg-

ularization step commonly applied in pixel classification-based segmentation methods, resulting in a notable absence of spatial consistency.

2.6.2 SegNet

SegNet is an open-source project focused on image segmentation [10], developed by the team at Cambridge University. This initiative aims to accurately identify and segment areas within images where various objects are present, such as cars, roads, pedestrians, and more, down to the pixel level. The implementation of image segmentation utilizes a convolutional neural network, primarily consisting of two key components: the encoder and the decoder. The encoder follows the VGG16 architecture and primarily analyzes object-related information. On the other hand, the decoder translates the analyzed information into the final image representation, assigning each pixel a specific colour or label corresponding to its respective object information. Subsequent chapters will delve into the expansion and enhancement of the SegNet model.

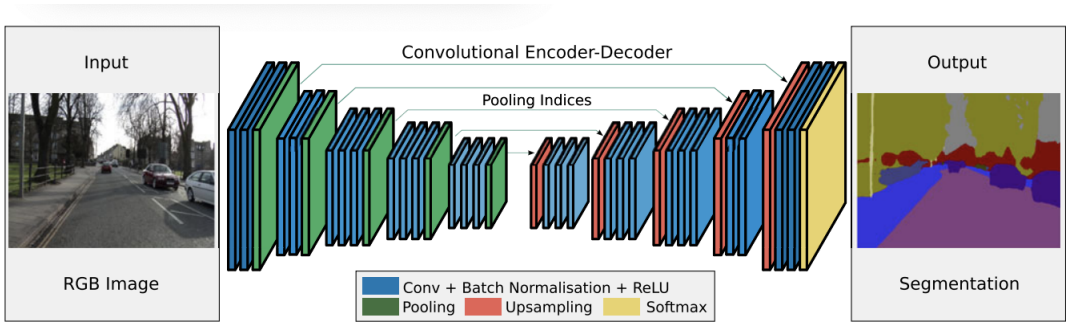


Figure 2.12: *The architecture of SegNet [10].*

As depicted in the above figure, SegNet is a symmetric network comprising an encoder (on the left) and a decoder (on the right). Upon input of an RGB image, the network categorizes objects within the image (e.g., "road," "car," "building," etc.) by leveraging the semantic information associated with these objects. Ultimately, it produces a segmented image representing the identified objects [10]. Although it boasts a sophisticated name, the encoder is essentially a series of convolutional networks. The network primarily comprises volume-based layers, pooling layers, and BatchNormaliza-

tion layers. The volume-based layer extracts local features from the image, the pooling layer reduces the image dimensions and conveys scale-invariant features to the subsequent layer, and BatchNormalization normalizes the distribution of training images while expediting the learning process.

To sum up, the encoder's role involves categorizing and scrutinizing the image's low-level local pixel values to derive high-level semantic information ("car," "road," "pedestrian"). The decoder then assimilates this semantic information, associating each object with its corresponding pixel points and representing each object with a distinct colour.

With the encoder having acquired comprehensive objects and approximate positional information, the subsequent task involves mapping these objects to precise pixels. This task is handled by the decoder. The decoder involves upsampling the downsized feature image and subsequently applying convolutional processing to the upsampled image. This process aims to enhance the geometric shape of the objects and compensate for the loss of details caused by the pooling layer in the encoder, which reduces the object's granularity. The encoder analyzes the image to determine the type of object present in a specific area, while the decoder identifies the pixels in the original image that correspond to this object. Through this process, an image is successfully segmented. Following each round of maximum pooling and downsampling, the spatial resolution of the feature map decreases, which poses a challenge for accurately segmenting boundary contours. As a solution, one option is to store all feature maps, but this would significantly strain memory resources. Consequently, the SegNet proposes a more memory-efficient approach by exclusively recording the indices of the maximum-pooled values. While this storage method may result in a minor loss of precision, it remains suitable for practical applications with more modest memory constraints.

SegNet employs a technique called Pooling Indices to retain the original information of pooling points. During processing in the encoder's pooling layer, the system records the source area of each 1×1 feature point pooled from the preceding 2×2 area. This information is referred to as Pooling Indices in the research paper. In the decoder

phase, these Pooling Indices come into play. Given SegNet's symmetrical structure, the corresponding pooling layer's Pooling Indices can be utilized to determine where a specific 1×1 feature point should be positioned within the upsampled 2×2 area during the upsampling of the feature map in the decoder. The final layer of the encoder comprises the pooling layer, continuing the downsampling process. With this, the encoder's role concludes, and the decoding phase ensues. In essence, each decoding layer undertakes the inverse of the encoding process, albeit with some distinctions. This process is illustrated in the figure below. Initially, the prior pooling results are reinstated based on the stored maximum pooling positions. Upon this restoration, the feature map expands in size, necessitating the placement of zeros in other locations. Subsequently, the resulting feature map undergoes convolution with the kernel, generating a dense feature map, which is then subjected to batch processing. These steps are iteratively applied to each mapping.

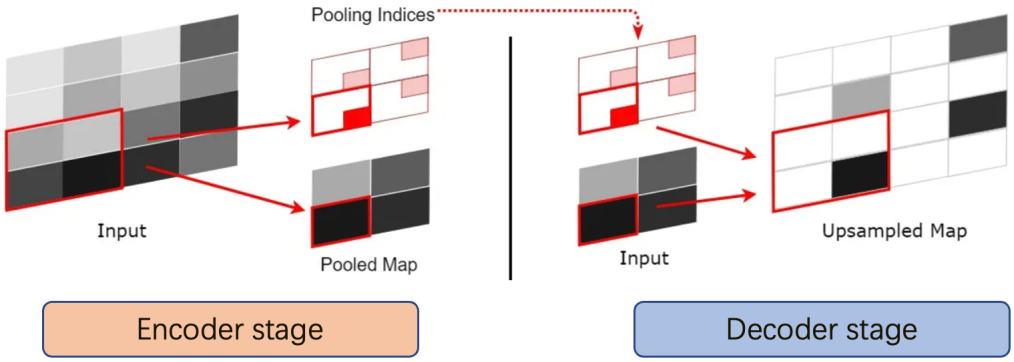


Figure 2.13: *The computation procedure of SegNet.*

2.7 Evaluation Metrics

The mean Intersection over Union (IoU) [132] characterizes the balance between precision and recall performance measures. This section also shows both precision and recall results and demonstrates that the performance of the FCN with the cross dropout focal loss function gives very good segmentation results.

- **Mean Intersection over Union**, $mIoU$: In semantic segmentation, this evaluation metric calculates the intersection ratio of two sets. These two sets are annotated data and predicted outputs [44]. It is computed by

$$mIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}}, \tag{2.1}$$

  where p_{ij} and p_{ji} represents false positive and false negatives for class i and class j respectively. The value of p_{ii} is the number of true positives. The value of k is the total number of classes.

- **Mean Accuracy**: It computes two sets, which are the number of the correct pixels p_{ii} and the total number of pixels per class [44]. After getting per-class accuracy, the mean accuracy $mAcc$ averages the total $k+1$ classes:

$$mAcc = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}}. \tag{2.2}$$

- **Precision** It [106] refers to the proportion of the total number of true positives (TPs) divided by the sum of all TPs and false positives (FPs):

$$Precision = \frac{TP}{TP + FP}. \tag{2.3}$$

  The average of per-class precision [155] is

$$Precision_M = \frac{1}{l} \sum_{i=1}^{l} Precision, \tag{2.4}$$

  which is the arithmetic mean of all the summary precision values by the number of classes l.

- **Recall** It [106] defines the number of correct positive predictions, which are achieved from all the positive predictions. False negatives and true positives

denote total samples:

$$Recall = \frac{TP}{TP + FN}. \tag{2.5}$$

The average per-class recall is identified as

$$Recall_M = \frac{1}{l} \sum_{i=1}^{l} Recall. \tag{2.6}$$

- **F1-score** The Recall_M [155] focuses on the per-class effectiveness of class labels. A good model expects to get high values on both precision and recall. However, it is difficult to decide the model performance when the precision and recall are reaching different extremums. Thus, it is necessary to use the F1-score in the evaluation. The F1-score_M [155], on the other hand, summarizes the Precision_M and the recall of a classifier system into a single metric

$$F1 - score_M = 2 \times \frac{(Precision_M \times Recall_M)}{(Precision_M + Recall_M)}. \tag{2.7}$$

- **TP rate and FP rate** The TP rate reflects the model's ability to correctly identify positive cases (e.g., diseased samples). It measures the proportion of actual positives that are correctly identified as positive by the model out of all actual positives. The FP rate reflects the model's tendency to incorrectly identify negative cases (e.g., healthy samples) as positive. It measures the proportion of actual negatives that are wrongly labeled as positive by the model out of all actual negatives. These rates vary by adjusting the model's decision threshold, which determines the probability score at which outcomes are classified as positive or negative.

$$FP\_rate = \frac{FP}{FP + TN},$$
$$TP\_rate = \frac{TP}{TP + FN}. \tag{2.8}$$

Chapter 3

Explore Loss Function for Imbalanced Data Problem in Semantic Segmentation

Deep learning methods have proven their potential in semantic segmentation. However, they depend on the data quality and training process. Usually, the data corresponding to the objects to be segmented are of different sizes and this creates difficulties for the segmentation method. Objects are segmented and associated with classes during the training process. Data imbalance is a challenging problem, which often results in unsatisfactory segmentation performance. This chapter proposes a solution to this task based on a novel deep learning approach based on the fully convolutional network, which expresses well the learning weight of features providing a balance of the uniform distribution of classes within the dataset. The performance of the considered fully convolutional network (FCN) with different various is considered and carefully evaluated. The proposed method improves efficiently the semantic segmentation performance over other well-known methods. It is demonstrated on Cityscapes and PASCAL VOC 2010 publicly available datasets. The implementation is over relatively large data sets. The

achieved mean accuracy of the proposed improvement network with cross dropout focal loss function (CDFL) on Cityscapes dataset is 76.41% and on PASCAL VOC 2010 dataset is 79.63% which is approximately 2.5% improvement compared with the network implemented with the improvement functions.

3.1 Imbalanced Semantic Segmentation Analysis

Semantic segmentation is defined as the pixel-level classification [173, 44, 57]. The results depend to a large extent on the dataset balance. When one label data is in the minority class while millions of labels are in the majority class, it can lead to a slight bias or severe imbalance in the predictive results [72]. This means data imbalance is a fundamental problem in semantic segmentation tasks, which restricts the accuracy and precision of the image segmentation. The imbalanced dataset poses a challenge for prediction since many semantic segmentation algorithms assume an equal number of each class. However, in semantic segmentation datasets, the class imbalance is inevitable. For instance, in Cityscapes datasets [27], a traffic sign and a person are considered as the minority of the segmentation classes, while a large number of classes correspond to buildings, roads and the sky. Commonly, models need to pay sufficient attention to the minority of classes for safety reasons. However, models naturally have a bias towards the majority classes in the training process, which leads to low accuracy and precision results, especially on a small number of classes. The choice of the loss function and how it is linked to the labels plays an important role in improving the image segmentation performance.

A series of significant works show how to alleviate the impact of the data imbalance on the segmentation results [44, 127]. The majority of the methods focus on the design of loss functions that consider well both the minority and majority classes. There are three types of loss functions [165, 77, 151]. Firstly, the region-based loss function directly optimizes the intersection-over-union (IoU) [129]. This type of loss function mainly applies to medical segmentation. Secondly, the statistics-balanced loss

This chapter corresponds to the publication-Jingxuan Su, Sean Anderson, and Lyudmila S. Mihaylova, "A Deep Learning Method with Cross Dropout Focal Loss Function for Imbalanced Semantic Segmentation", In *Proceedings of the 2022 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, Germany, IEEE, 2022.

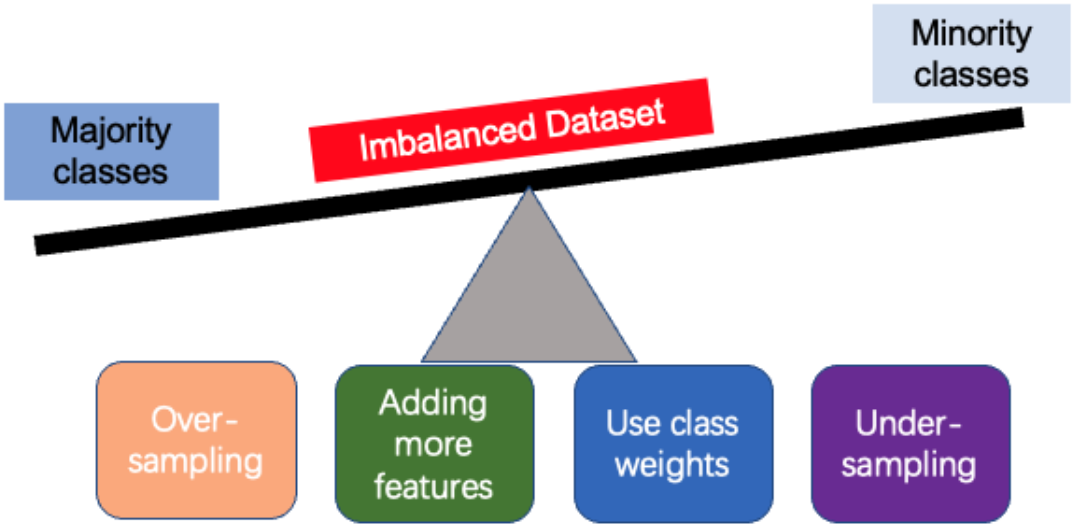


Figure 3.1: *The sketch of imbalanced dataset problem.*

function adjusts the weight of class distribution based on its margin or size, i.e. class-balanced loss [28] and a label-distribution-aware margin (LDAM) loss function [17]. It encourages overfull false positives in the small number of classes. However, this approach could undermine the learning capability in feature extraction [191]. Thirdly, the performance-balanced loss function adds factors to weight the distribution of each class, i.e. as it is in the focal loss function [91]. However, its applications face challenges sometimes [28] since it cannot balance between the small and large number of classes that up-weight the minority class [191].

This work develops a novel data-balanced driven semantic segmentation solution consisting of a fully connected convolutional neural network and a cross dropout focal loss function. The cross dropout focal loss function down-weights, respectively up-weighs a class based on the output for this class. Unlike the statistics-balanced losses, the cross dropout focal loss has dynamic weight components based on per-class network outputs, compared to the statistics-balanced losses. In our experiments, the cross dropout focal loss can effectively address data imbalance and improves the accuracy

This chapter corresponds to the publication-Jingxuan Su, Sean Anderson, and Lyudmila S. Mihaylova, "A Deep Learning Method with Cross Dropout Focal Loss Function for Imbalanced Semantic Segmentation", In *Proceedings of the 2022 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, Germany, IEEE, 2022.

and IoU.

3.2 The Fully Convolutional Network Architecture for Semantic Segmentation

Deep learning methods have witnessed a significant surge in popularity for semantic segmentation tasks [96, 20, 44]. However, despite the advancements, challenges persist in achieving precise pixel-level image segmentation, primarily due to the presence of imbalanced datasets. Among the pioneering algorithms, the fully convolutional network (FCN) [96] has emerged as a versatile leader and is frequently chosen as the core component in numerous deep learning approaches [29, 145]. FCNs are characterized by their use of skip layers, enabling them to learn representations effectively. Furthermore, the efficiency of fully convolutional networks extends to other well-known architectures such as U-Net [135] and SegNet [10, 9]. These architectures possess various attractive properties, including smooth predictions and straightforward visualizations of feature activations in the pixel label space [60, 88].

To balance computational efficiency and accuracy, this work picks the fully convolutional network architecture as the deep learning backbone. Nonetheless, despite the capabilities of these state-of-the-art networks, they still encounter challenges in solving the data imbalance problem inherent in semantic segmentation tasks.

The quality of the training datasets plays a pivotal role in achieving effective semantic segmentation models. Deep learning methods, such as FCN [96], U-Net [135], and more recent techniques like Deeplab [20], often face the issue of class imbalance. When neural networks are trained predominantly on easy examples, the learning process may suffer, leading to overall suboptimal accuracy. To address this problem, increasing the number of hard examples has been a common approach [39, 148, 170]. However, in contrast to these existing works, this work proposes a novel solution: the cross dropout focal loss function, which effectively tackles the data imbalance issue without requiring complex computations or extensive sampling.

Various loss functions have been utilized for semantic segmentation tasks. Among them, the cross-entropy [181] measures the difference between two probability distributions and has found widespread use. The weighted cross-entropy [122] addresses the imbalance problem by assigning appropriate weights to positive and negative examples, resulting in improved performance compared to conventional cross-entropy for imbalanced classes. Another approach, the balanced cross entropy [176], is motivated by the weighted cross-entropy and optimizes the utilization of the number of samples in each class. The focal loss function [91], on the other hand, allows training on a sparse set of hard examples and has shown effectiveness in object detection tasks.

The proposal of these various loss functions stems from the motivation to improve the weighting of class labels [103, 72]. However, it's worth noting that some of these loss functions may lead to the introduction of excessive false positives and adversarial results [165].

Evaluation metrics are essential in the context of segmentation networks, and the Intersection over Union (IoU) is one of the most commonly used indicators. Lovasz Softmax [11] directly optimizes the IoU using the Lovasz convex extension, while the Dice similarity coefficient [161] controls the trade-off between false positives and negatives in image segmentation. In contrast to these methods, our proposed cross dropout focal loss function takes a different approach by considering the balance between different classes and not solely relying on label weights.

In conclusion, deep learning methods have gained immense popularity for semantic segmentation tasks, but imbalanced datasets present ongoing challenges in achieving accurate pixel-level image segmentation. The fully convolutional network architecture is employed as the backbone while introducing a novel cross dropout focal loss function to address the data imbalance issue. Various loss functions have been explored, each with its own merits, driven by the motivation to improve the weighting of class labels. By carefully considering the balance between different classes, our proposed approach aims to achieve more accurate and efficient semantic segmentation results.

3.3 The Proposed Method in Multi-class Segmentation

3.3.1 Cross Entropy for Multi-class Segmentation

Cross-entropy serves as a crucial metric for assessing the disparity between the probability distribution derived from current training data and the actual distribution. It quantifies the gap between the observed output probabilities and the desired output probabilities. In essence, a lower cross-entropy value indicates a greater similarity between the two probability distributions. The cross-entropy [72] has been widely applied in many semantic segmentation tasks [72, 106]. It uses the number of pixels for each class to optimize the geometric mean confidence of each weighted class. The formula for the cross-entropy CE is the following:

$$CE = -\sum_{c=1}^{M} y_c \log\left(p_c\right), \tag{3.1}$$

where M denotes the class number, p_c represents the corresponding value of the c-th class in the output of the softmax activation function, and y_c denotes the value of true predictions in the class c. If the class of prediction and label are the same, then the value of 1 is assigned, otherwise, it is 0. However, this approach with the cross-entropy has an obvious drawback that it applies to a balanced dataset. When the number of pixels in the minority class is much smaller than the number of pixels in the majority class for the same image, the $y_c = 0$ in the function will dominate. Thus, the number of pixels influences the value of y_c. In other words, if the number of $y_c = 0$ is much larger than the number of $y_c = 1$, this situation will make the model heavily biased towards the main label which results in poor results.

The balanced cross entropy (BCE) [72] adds a weight parameter for each class to solve the data imbalance problem. The balanced cross entropy BCE for muti-segmentation is represented with the equation:

$$BCE = -\sum_{c=1}^{M} w_c y_c \log\left(p_c\right). \tag{3.2}$$

The weight parameter w_c calculation formula is $w_c = \frac{N-N_c}{N}$, where N denotes the total number of pixels, and N_c shows the number of pixels in the ground truth per class. The variable y_c still has the same meaning as in the cross-entropy expression. In this way, the balanced cross entropy can represent well the different classes with different weights for small or large classes. However, it did not consider the easy-hard imbalance in per class. The balanced cross entropy cannot address the data imbalance issue effectively when facing a big semantic segmentation dataset.

3.3.2 Focal loss for Multi-class Segmentation

Focal loss was first proposed by He Kaiming and was initially used in the image field to solve model performance problems caused by data imbalance [91]. In contrast to balanced cross-entropy, focal loss aims to address the issue of model training resulting from sample imbalance. The latter introduces weight factors into the loss function, considering the distribution of samples. The former, on the other hand, tackles the challenge of classifying samples by focusing the loss on those that are particularly difficult to distinguish.

Focal loss addresses model training challenges arising from sample imbalance by starting with the perspective of classifying difficult samples. The issue associated with sample imbalance is that classes with a limited number of samples are inherently harder to classify. Consequently, by prioritizing challenging samples in terms of classification difficulty, focal loss effectively alleviates the problem of low classification accuracy within classes having few samples.

It is worth noting that challenging samples are not exclusive to classes with limited samples. In other words, focal loss not only mitigates the sample imbalance problem but also contributes to enhancing the overall model performance. However, simply directing the loss toward difficult-to-classify samples is insufficient for effective model

training, as the model parameter updates during training rely on the gradients of the loss function.

In [91] the focal loss function is proposed for binary segmentation. The idea for the focal loss is inspired by the cross-entropy. The focal loss has two hyperparameters, γ and α that are introduced for balancing between the easy and hard examples. This work extends the focal loss to the multi-segmentation task. The activation function can only be the softmax [91, 109] function. The multi-focal loss with the softmax function FL_{softmax} is defined as:

$$FL_{\text{softmax}} = -\sum_{c=1}^{M} \alpha_{\text{c}} \left(1 - \text{p}_{\text{c}}\right)^{\gamma} \log\left(\text{p}_{\text{c}}\right), \tag{3.3}$$

where α_{c} indicates the weight of the c-th class label, p_c denotes the output of the c-th class after the softmax function. The value of p_c can reflect the degree of difficulty of the sample in segmentation. When $p_c > 0.5$, it belongs to an easy-segmented region, otherwise is a hard-segmented region. If the value of p_c is big, the prediction results will be more accurate. The parameter γ adjusts the rate of easy label down-weighted labels. The parameter α represents the adjustment weight of the corresponding positive sample. However, this loss function only considers the easy-hard imbalance, without considering the imbalance class.

3.4 Cross Dropout Focal Loss for Multi-class Segmentation

The proposed dropout cross focal loss function aims to improve the model performance and weight well the balance per class for easy-hard segmentation. In the proposed approach the input data are considered with T dropout times into the segmentation architecture. Thanks to the Monte Carlo dropout procedure [42], deep neural network output $\hat{y}_t$ will be different after dropout at each time. Then an indicator variable $u(\hat{y})$ is introduced which depends on the network predicted output $\hat{y}_t$ and can be expressed

by the following equation:

$$u(\hat{y}) \approx \frac{1}{T}\sum_{t=1}^{T}(\hat{y}_t)^2 - \left(\frac{1}{T}\sum_{t=1}^{T}\hat{y}_t\right)^2. \tag{3.4}$$

The value of the indicator variable $u(\hat{y})$ represents the easy-hard degree of segmentation from the dropout output per class perspective. Motivated by the focal loss, the value of the indicator variable $u(\hat{y})$ can replace the modulating factor $(1-p_t)$ from equation (3.3). Thus, the focal loss is updated to the dropout focal (DF) loss shown in the following equation:

$$DF = -\frac{1}{N}\sum_{i=1}^{N}\alpha_{\mathrm{i}}\left(u_i(\hat{y})\right)^{\gamma}\log\left(\hat{y}_i\right). \tag{3.5}$$

When the value of $u(\hat{y})$ is close to 0, the value of the dropout focal loss function will reduce. That means the easy-segmented labels are down-weighted. In this work, the T value was set equal to 5 to keep an efficient computational time and sufficient accuracy.

The cross-entropy and focal loss functions face challenges with the imbalanced dataset. Thus, a novel loss function is proposed, the cross dropout focal loss (CDFL). Based on the cross-entropy, the dropout focal loss is added with a weighted index ω as a modulating factor to solve the data imbalance problem. The cross dropout focal loss $CDFL$ is represented with the following equation:

$$\begin{aligned} CDFL &= CE + \omega DF \\ &= -\sum_{i=1}^{N} y_i \log\left(p_i\right) - \omega[\frac{1}{N}\sum_{i=1}^{N}\alpha_{\mathrm{i}}\left(u_i(\hat{y})\right)^{\gamma}\log\left(\hat{y}_i\right)]. \end{aligned} \tag{3.6}$$

This work sets up the values of the γ and α parameters respectively equal to 2 and 0.75, which is the same choice as in [91]. The weight factor ω balances the impact of the cross-entropy and of the dropout focal loss. For the purpose of performance validation, various values of ω were tested, including 0.1, 0.01, and 0.001. It was found that

the best performance was obtained when the value of ω was set to 0.01. As a result, the cross dropout focal loss function avoids excessive weighting of hard-segmented examples or of the minority classes which could cause undesirable results. Meanwhile, the $CDFL$ provides a suitable training weight for the different inputs. Therefore, the cross dropout focal loss achieves a higher balancing ability than the cross-entropy. It can handle both class and easy-hard segmentation imbalance situations.

3.5 Implementation and Analysis

The experiments are performed with the Ubuntu 20.04 system. The server environment uses Python 3.7, Pytorch 1.12.1 and CUDA 10.1.

3.5.1 Datasets and Implementation Details

In order to evaluate the proposed loss function, the performance of a FCN with different loss functions is evaluated over two popular semantic segmentation datasets, Cityscapes for outdoor driving and PASCAL VOC 2010.

Cityscapes

Cityscapes [27] is a popular data set for semantic segmentation, which comprises urban street scenes. It is a large-scale driving database that contains fine annotated data and coarse annotated data of around 25000. There are 8 groups with 30 classes. Data was captured from 50 cities under different environmental conditions. In this work, the dataset adopts 3475 fine annotations images for train and validation sets and 1525 images for the test set. It has 19 classes shown in Fig. 3.4.

The data imbalance within Cityscapes dataset can be inferred from a few key points. The dataset encompasses images from 50 different cities and under various environmental conditions. This diversity is crucial for developing robust semantic segmentation models but can also introduce data imbalance if certain environmental conditions or city-specific features are underrepresented. With 19 classes but a fixed total number

of images (3,475 for training/validation and 1,525 for testing), there is an inherent risk of data imbalance if the distribution of images across these classes is not even. For instance, some classes might be overrepresented in the dataset, while others might have significantly fewer examples. The dataset contains both fine and coarse annotations, with a specific focus on fine annotations for training and validation. If the fine annotations are not evenly distributed across the 19 classes, this could lead to a data imbalance where models might perform better on classes with more detailed annotations compared to those with less.

PASCAL VOC 2010

Pattern Analysis, Statical Modeling and Computational Learning (PASCAL) Visual Object Classes (VOC) [38] is a computer vision challenge for five different competitions and provides ground truth annotated datasets. This work only focuses on the PASCAL VOC 2010, which is a two dimensional (2D) segmentation dataset. Especially, the dataset supports pixel-level segmentation. It contains 540 classes divided into 3 groups (objects, stuff, and hybrids). The dataset contains 4998 images for training and 1550 for validation. It has 20 classes: aeroplane, bag, bed, bedclothes, bench, bicycle, bird, boat, book, bottle, building, bus, cabinet, car, cat, ceiling, chair, cloth, computer, cow and others.

Considering the dataset's characteristics, several factors suggest the potential for data imbalance within the PASCAL VOC 2010 dataset. While the dataset encompasses 540 classes divided into 3 groups, it specifically focuses on 20 classes for segmentation tasks. This selection process might not equally represent the diversity within the larger group of classes, potentially leading to an imbalance. The inherent challenge in datasets with a wide range of classes, like the PASCAL VOC 2010, is ensuring that each class is adequately represented. With 20 classes identified for segmentation, the degree of imbalance would depend on how evenly the images are distributed across these classes. If certain classes, such as 'aeroplane' or 'cat', have significantly more images compared to others like 'bench' or 'cloth', this could lead to a model bias

towards the more represented classes. The classes included vary significantly in terms of visual complexity and size. For example, 'building' and 'bus' are likely to be larger and more visually complex than 'book' or 'bottle'. This variability can contribute to data imbalance, as simpler objects might be easier to segment and therefore might appear more frequently or with more annotations compared to complex ones.

Evaluation metrics

The cross dropout focal loss based on FCN [96] is implemented for the segmentation task on the two above mentioned datasets. In semantic segmentation, the mean accuracy (mACC) and the mean IoU (mIoU) [17, 191, 95, 78] are important metrics. Here this work employs them to evaluate the image semantic segmentation performance. The next subsection presents results over the considered public data and evaluates the segmentation results.

3.5.2 Validation Results and Analysis

This work conducted a comprehensive evaluation by comparing the performance of the fully convolutional network using three widely recognized loss functions for imbalanced semantic segmentation. The loss functions under scrutiny were the cross-entropy, focal loss [91], and Lovasz Softmax loss function [11]. However, our research introduced a novel loss function, which outperformed the others on two diverse datasets.

Table 3.1: *Quantitative FCN performance with different losses on Cityscapes*

loss	mIoU (%)	mAcc (%)	mPre (%)	mRec (%)
Cross-entropy	66.51	76.33	80.78	77.86
Focal loss	62.1	74.47	79.25	72.75
Lovasz softmax loss	57.14	70.51	75.22	70.51
CDFL	**66.62**	**76.41**	**81.23**	**78.11**

Table 3.1 serves as a valuable repository of segmentation results obtained from the Cityscapes outdoor driving dataset. These results offer critical insights into the perfor-

(a) The Frankfurt cityroad from Cityscapes database

(b) The fine annotation image of Frankfurt cityroad

(c) The Lindau cityroad from Cityscapes

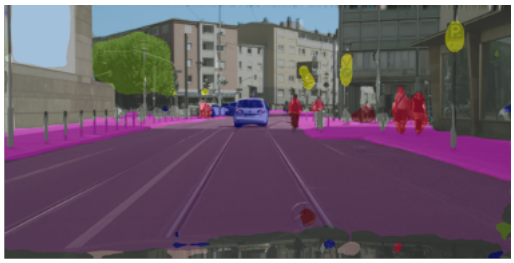
(d) The fine annotation image of Lindau cityroad

(e) The Munster cityroad from Cityscapes

(f) The fine annotation image of Munster cityroad

Figure 3.2: *The original cityroad images and the corresponding fine annotation images*

(a) The segmented Frankfurt cityroad image by FCN model with cross entropy loss



(b) The segmented Frankfurt cityroad image by FCN model with focal loss

(c) The segmented Frankfurt cityroad image by FCN model with cross dropout focal loss

(d) The segmented Lindau cityroad image by FCN model with cross entropy loss

(e) The segmented Lindau cityroad image by FCN model with focal loss

(f) The segmented Lindau cityroad image by FCN model with cross dropout focal loss

(g) The segmented Munster cityroad image by FCN model with cross entropy loss

(h) The segmented Munster cityroad image by FCN model with focal loss

(i) The segmented Munster cityroad image by FCN model with cross dropout focal loss

Figure 3.3: *Visualization of segmentation results on Cityscapes with FCN.*

mance of various loss functions in the realm of imbalanced semantic segmentation. This work assessed these metrics across several dimensions, including the mean accuracy and Intersection over Union (IoU). The outcomes were rather striking, as they revealed significant enhancements in segmentation quality when employing the cross-entropy, focal loss, and cross dropout focal loss, in contrast to the Lovasz softmax loss. This empirical evidence highlights the importance of selecting the appropriate loss function in the context of image segmentation, especially when dealing with imbalanced datasets.

Cross-entropy has long been a foundational loss function in the field of deep learning, and its effectiveness was clearly evident in our results. However, the introduction of the focal loss, a concept introduced by Lin in 2017 [91], showed promise in further improving the quality of segmentation. This loss function, designed to address the problem of class imbalance, gives more weight to misclassified examples, thereby concentrating the network's attention on the challenging instances, which ultimately boosts the overall segmentation performance. Moreover, the cross dropout focal loss, an innovative hybrid approach that incorporates the best aspects of both cross-entropy and focal loss, emerged as a standout performer in our evaluation.

What sets the cross dropout focal loss (CDFL) apart from the other loss functions is its remarkable balance between precision and recall. With a precision rate of 81.23% and a recall rate of 78.11%, CDFL demonstrates an admirable ability to strike a harmonious equilibrium between correctly identifying relevant objects in the images (precision) and avoiding false negatives (recall). This characteristic is of immense significance, especially in applications where false positives or missed detections can have significant consequences.

In the context of semantic segmentation, achieving high precision is crucial to ensure that the identified objects are indeed the objects of interest, and CDFL's precision rate attests to its ability to excel in this regard. Similarly, its robust recall rate indicates that CDFL is capable of capturing a substantial portion of the true positives, minimizing the risk of overlooking important objects or regions within the images.

The importance of these findings is not only applicable to the scope of our research

but extends to broader applications in computer vision, autonomous driving, and any domain where accurate object recognition in images is pivotal. Choosing the most suitable loss function can be the point of success in machine learning tasks, and our evaluation underlines the potential of the cross dropout focal loss to significantly impact the field.

In conclusion, the results presented in Table 3.1 underscore the promising advancements in semantic segmentation achieved by the cross-entropy, focal loss, and cross dropout focal loss when compared to the Lovasz softmax loss. Among these, the cross dropout focal loss, with its remarkable precision and recall rates, has the potential to achieve high performance in other image segmentation tasks, offering a valuable tool for researchers and practitioners seeking to enhance the quality and accuracy of their computer vision applications. The implications of these findings extend to a wide array of real-world scenarios, making this research a crucial stepping stone toward more effective and reliable image segmentation solutions.

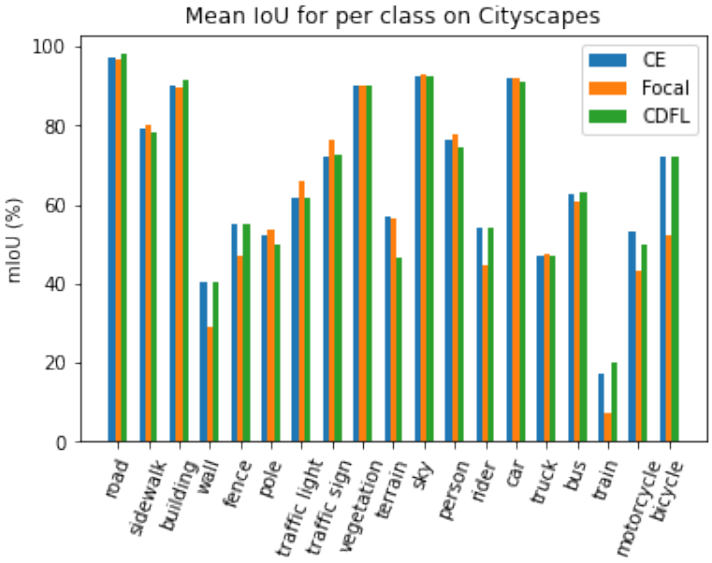


Figure 3.4: *Mean IoU per class on Cityscapes with FCN*

In terms of overall performance, the cross dropout focal loss function emerged as the top performer while maintaining a commendable mean IoU and accuracy. Fig. 3.4 displays the mean IoU values for individual classes, showcasing the cross dropout focal

loss's ability to effectively enhance the weight of smaller classes like trains and buildings while maintaining high precision and recall.

Furthermore, this work visualized the segmentation results in Fig. 3.3, illustrating that both the cross-entropy and focal loss produced incorrect predictions, misclassifying the black class as the blue class at the bottom. On the other hand, the focal loss and cross dropout focal loss exhibited precise predictions, especially evident in the second row of Fig. 3.3, where traffic signs were accurately identified.

To further illustrate the benefits of the Cross Dropout Focal Loss, a histogram graph is constructed in Fig. 3.4, showcasing its ability to encourage correct predictions for small classes, such as trains.

Overall, the novel Cross Dropout Focal Loss has emerged as a powerful tool for improving semantic segmentation performance. It not only outperforms other loss functions while maintaining good mean IoU and accuracy but also excels in accurately predicting small classes, significantly enhancing the overall segmentation results.

Table 3.2: *Quantitative FCN performance with different losses on PASCAL VOC 2010*

Algorithms	mIoU (%)	mAcc (%)	mPre (%)	mRec (%)
FCN + Cross-entrop(Original)	66.72	76.85	80.32	76.23
FCN + Focal loss	62.45	75.74	76.45	70.57
FCN + Lovasz	59.43	64.16	69.56	63.24
FCN + CDFL(Ours)	67.74	79.63	**81.85**	**79.63**

This work further shows the performance of the proposed loss function and of other state-of-the-art loss functions on the PASCAL VOC 2010 segmentation dataset. The table below (Table 3.2) presents the performance metrics for four distinct loss functions, each evaluated in the context of imbalanced semantic segmentation. These metrics, which include mIoU (mean Intersection over Union), accuracy, precision, and recall, serve as essential benchmarks to gauge the effectiveness of the loss functions in question. The cross-entropy loss function delivered a commendable mIoU of 66.72%, indicating its effectiveness in achieving a balanced intersection over union metric. With an accu-

racy of 76.85%, it showcases its ability to correctly classify objects in the segmentation task. The precision value of 80.32% demonstrates its capability to minimize false positives, while a recall of 76.23% underscores its efficiency in minimizing false negatives. Focal loss, a loss function specifically designed to address class imbalance, yielded a slightly lower mIoU at 62.45%. Its accuracy of 75.74% signifies its competence in accurate classification. Precision and recall scores of 76.45% and 70.57%, respectively, show its effectiveness in both minimizing false positives and false negatives, albeit at a slightly lower level than cross-entropy. The Lovasz softmax loss function achieved a mIoU of 59.43%, signaling a moderate performance in terms of intersection over union. While its accuracy is at 64.16%, indicating a reasonable level of correct classifications, precision and recall values of 69.56% and 63.24%, respectively, signify its capacity to mitigate false positives and false negatives, though not as effectively as the previous two loss functions. The CDFL loss function demonstrates impressive results with an mIoU of 67.74%. It excels in accuracy, achieving 79.63%, showcasing its strong classification capabilities. With a precision of 81.85%, it minimizes false positives effectively. Furthermore, the recall value is also 79.63%, highlighting its ability to minimize false negatives. CDFL combines the strengths of cross-entropy and focal loss, resulting in a balanced and high-performing loss function.

In summary, the choice of loss function is a critical determinant in the success of image segmentation tasks. In this evaluation, CDFL emerges as the standout performer, offering a remarkable balance between precision and recall. It not only achieves a high mIoU and accuracy but also excels in minimizing false positives and false negatives. These findings have significant implications for the field of computer vision, as they underscore the potential of loss functions to significantly impact the quality and accuracy of segmentation results. Researchers and practitioners can consider these results when selecting the most suitable loss function for their specific applications.

3.6 Summary

This chapter introduces a novel approach known as the data-balanced fully convolutional network (FCN) algorithm, specifically tailored for semantic image segmentation. The original FCN network, when combined with innovative Cross Dropout Focal Loss (CDFL), serves as a compelling solution to mitigate the challenges posed by imbalances in class datasets, thereby enhancing the overall model performance. What sets this loss function apart is its unique design perspective, which originates from the model's output. It incorporates dynamic weights that adapt to the relative classes, effectively segmenting the corresponding objects within images. The Cross Dropout Focal Loss offers several notable advantages:

1) Effective Data Balancing: It efficiently addresses the data imbalance issue by dynamically weighting the dropout output for each class. This adaptive approach ensures that underrepresented classes receive the attention they deserve, fostering a more equitable and accurate segmentation.

2) Dynamic Weight Updates: The loss function dynamically updates weights based on the results of multiple dropout iterations (denoted as T). These dropout results serve as a reflection of the segmentation's level of complexity, helping in the generation of well-suited weightings that optimize the segmentation process.

3) Enhanced Performance: Validation of the proposed Cross Dropout Focal Loss on Cityscapes and PASCAL datasets reveals substantial performance gains. With an approximate 2.5% improvement in accuracy when compared to state-of-the-art loss functions, it showcases the robustness and effectiveness of our novel approach.

This chapter's findings not only underscore the importance of addressing data imbalances in semantic image segmentation but also provide a clear path towards significantly improving model accuracy and robustness. The data-balanced fully convolutional network algorithm represents a groundbreaking development in this field, offering a practical and efficient solution for enhancing the quality of image segmentation results. The following chapter will solve the fine-grained classification to discover

the learning strategy in deep learning methods.

Chapter 4

Explore Learning Strategy with the Squeeze and Excitation Network for Fine-grained Plant Pathology Classification

Fine-grained plant pathology classification is an important task for precision agriculture, but at the same time, it is challenging due to the subtle differences in plant classes. Variances in the lighting conditions, position, and stages of disease symptoms usually lead to degradation of classification accuracy. Knowledge distillation is a popular method to improve the model performance to deal with the indistinguishable image classification problem. It aims to have a well-optimised small student network guided by a large teacher network. Existing knowledge distillation methods mainly consider training a teacher network that needs a high storage space and considerable computing resources. Self-knowledge distillation methods have been proposed to distil knowledge from the same network. Although self-knowledge distillation saves time and space compared with knowledge distillation, it only learns label knowledge. This chapter pro-

posed a novel self-distillation method to recognize the fine-grained plant class, which considers holistic knowledge based on the Squeeze and Excitation Network. This new method is labelled as holistic self-distillation because it captures knowledge through spatial features and labels. The performance validation of the proposed approach is performed on two public fine-grained plant datasets: Plant Pathology 2021 and Plant Pathology 2020 with the accuracy of 98.22% and 90.72% respectively. The experiments present on the state-of-the-art algorithm (ResNet-50). The classification results demonstrate the effectiveness of the proposed approach with respect to accuracy.

4.1 Precision Agriculture for Fine-grained Plant Pathology

Precision agriculture seeks to improve the production of plants and control environmental variations such as diseases that impact the production and quality of plants [54, 190]. Plant classification is an important technological challenge in precision agriculture [2] that aims to classify different subordinate classes under coarse large classes, e.g. plant diseases [163]. Plant classification tasks can be subdivided into coarse-grained and fine-grained [117] images. While coarse-grained image classification is interested in representing generic classes characterised with a large degree of dissimilarities, fine-grained image classification is a sub-field of object recognition that aims at representing classes with a large degree of similarity and is concerned with the problem of distinguishing between images of closely related entities, for instance, different species of plants from the same class. The focus work of this chapter is on fine-grained plant class classification for plants.

Fine-grained image classification [180] is a process that begins by discerning fundamental classes and subsequently delves into finer, more specific subclasses. It encompasses the ability to differentiate between various plant diseases, bird species, car models, dog breeds, and more. This approach is currently being applied in both industrial and real-world settings, serving a diverse array of business requirements and

This chapter corresponds to the publication-Jingxuan Su, Sean Anderson, and Lyudmila S. Mihaylova, "Holistic Self-Distillation with the Squeeze and Excitation Network for Fine-grained Plant Pathology Classification", In *Proceedings of the 2023 26th International Conference on Information Fusion (FUSION)*, (pp. 1-7), Charleston, USA, IEEE, 2023.

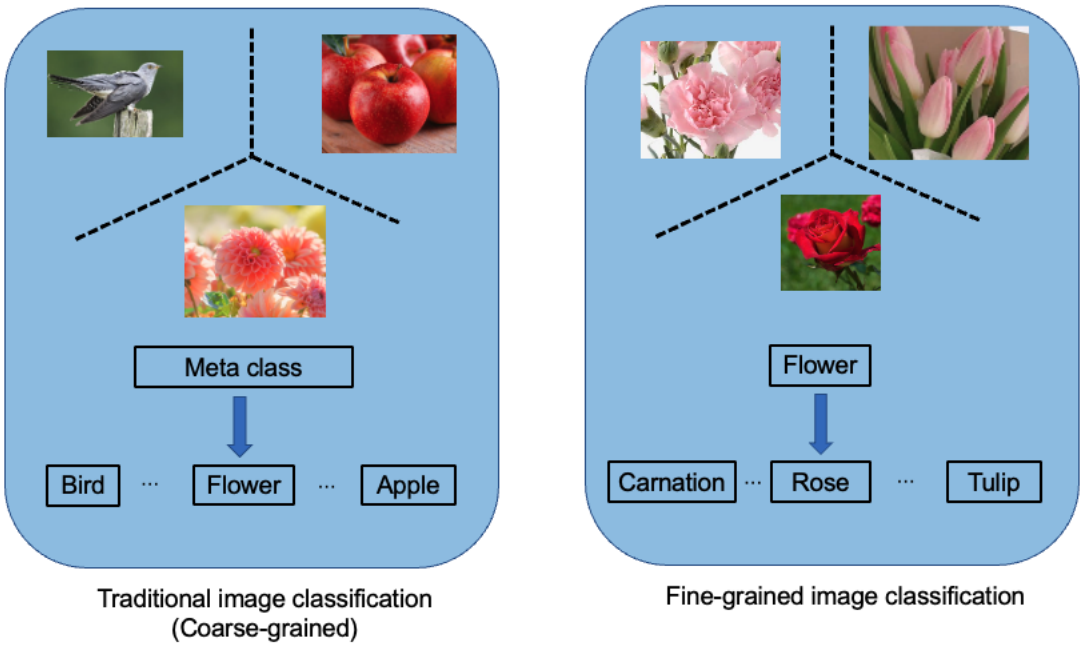


Figure 4.1: *The coarse-grained and fine-grained image classification.*

application scenarios. Fine-grained images pose a unique challenge compared to their coarse-grained counterparts. These images exhibit greater similarity in appearance and characteristics within the same broad class. Moreover, factors such as posture, perspective, illumination, occlusion, and background interference come into play during data collection, introducing substantial variations between classes and comparatively minor distinctions within classes. Intuitively, the fine-grained plant classes look very similar and are hard to distinguish, as shown in Fig. 4.4. Specifically, the inter-class variance is much smaller than the intra-class variance. Apparently, the fine-grained plant dataset increases the difficulty of classification. Moreover, the classification performance could directly affect society communities such as the farmers. The misclassification of plant diseases can lead to improper use of chemicals, decreased yield, and potentially harming the entire farm [158, 119]. Currently, manual scouting based disease classification is time–consuming and expensive. While many deep learning methods have achieved remarkable success in classification [85, 93, 94, 156], their application to fine-grained plant classification is still less satisfactory. This situation is even worse for great pathology

variances due to genetic variations, and light conditions.

These complexities significantly heighten the difficulty of accurate classification in the fine-grained image domain. The journey of fine-grained image classification research has evolved over an extensive period since its inception. Confronted with this intricate challenge, researchers have dedicated themselves to a comprehensive exploration and refinement of approaches, primarily building upon coarse-grained image classification models.

This research trajectory can be broadly categorized into two primary branches: traditional algorithms founded on feature extraction and algorithms rooted in deep learning [174]. In the early stages, algorithms reliant on feature extraction encountered notable limitations, primarily stemming from their constrained capacity to express intricate features. However, in recent years, the advent of deep learning has ushered in a remarkable transformation. The formidable feature extraction capabilities inherent in neural networks have propelled significant advancements within the realm of fine-grained image classification. This newfound prowess has accelerated progress in the field, marking a pivotal turning point in the pursuit of more accurate and nuanced image classification.

Enhanced fine-grained image classification algorithms have evolved over time, departing from early approaches rooted in artificial features that emphasized the analysis of local image attributes [174]. Initially, specific local characteristics were isolated from the image, followed by their encoding via relevant models. This method, however, proved cumbersome and somewhat limited in its expressive capacity, primarily because it failed to account for the interplay between various local attributes and their spatial relationships with global features. Consequently, these limitations hindered the attainment of satisfactory results.

To enhance classification accuracy, researchers introduced the notion of the "Bag of Visual Words" (BOVW) based on local attributes [179]. This approach involves quantifying the image's overall information, using quantified image segments as visual words, and describing the image's content through the distribution of these visual

words. The bag-of-words model was integrated with a range of feature extraction techniques, resulting in some progress. Nevertheless, practical application requirements remained unmet, and the construction of a bag of words remained a complex process necessitating additional work.

Both local attributes and visual word bags exhibited limited connections to global features, focusing their semantic mining efforts on specific image regions. In response, the concept of feature localization emerged, which involves leveraging key points' positional information to uncover the most valuable image data. The incorporation of location information did improve classification accuracy to some extent. However, acquiring accurate location data demands high-precision algorithms and precise manual annotation, which can be costly.

Deep learning provides the CNN model has achieved a historic breakthrough, and its effect greatly exceeds that of traditional methods. The difficulty of fine-grained plant classification comes from identifying subtle feature differences in particular regions. Residual network[63] as a state-of-the-art algorithm provides an effective architecture in general image classification. Squeeze and Excitation (SE) networks [67] have been proposed to focus on the feature details of specific regions, which won first place at the ILSVRC 2017 classification [136]. The main contribution of SE networks consists in the introduced Squeeze and Excitation (SE) block that finds the interdependencies between channels and adaptively pays attention to important features. The SE block can be stacked with any convolutional neural network, such as SE-ResNet-50, SE-Inception and others [67]. The SE network trains the binary assigned data (named hard label [153]). However, the performance of SE network may be restricted, since hard labels cannot provide sufficient feature information and the spatial features are lost in the SE block.

Knowledge distillation (KD) methods [64] aim at providing a well-optimised small student network guided by a large teacher network. The KD guides the student to learn the probability of each class (named soft labels [187, 189]) generated by the teacher network. Existing KD methods mainly consider training a teacher network that needs a

high storage space and considerable computing resources. Self-KD methods [50] have been proposed to distil their own knowledge without a pretrained teacher network. These approaches help the network to enhance classification performance. However, these methods often rely on extra networks and soft labels to capture additional knowledge, which loses the spatial features.

To address these challenges in existing classification methods, a novel self-distillation approach is proposed, Holistic Self-Distillation (HSD). The proposed approach is designed to extract spatial feature information before the SE block. The HSD is demonstrated superior to state-of-the-art (SOTA) method and other SE network approaches on plant image classification tasks. Extensive experiments on two public datasets further show the superiority of HSD in learning knowledge comprehensively from spatial feature information and soft labels.

4.2 Squeeze and Excitation Networks

Inspired by the significant improvements of the Squeeze and Excitation network in feature spatial encoding and classification tasks, the Squeeze and Excitation Residual network 50 (SE-ResNet-50) [67, 75, 32] is applied on the fine-grained classification to extract the holistic feature knowledge. It consists of two main parts, Residual framework [63] and Squeeze and Excitation block. This network captures the interdependencies between feature channels that obtain the importance of each feature channel through learning. The core idea is that useful features are promoted and the other features are suppressed. Fig. 4.2 shows the schematic of SE-ResNet-50 with feature maps computation. Formally, the Squeeze operation $\mathbf{F}_{sq}(\mu)$ transforms the size of feature map $H \times W \times C$ to the size of feature map $1 \times 1 \times C$, which is calculated by:

$$\mathbf{F}_{sq}(\mu_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u(i,j), \tag{4.1}$$

where H denotes the height, W is the width and C is the channel dimension of the feature map and μ is the shrink operate.

The Excitation component's role is to ascertain the characteristic weight of each channel, necessitating three key considerations: It must exhibit flexibility to ensure the acquired weights hold a higher value. It should maintain simplicity to prevent a substantial decrease in network training speed following the incorporation of SE blocks. The interplay between channels should be non-exclusive, meaning that the learned features can enhance significant attributes while mitigating less essential ones. According to the above requirements, SE blocks use a two-layer fully connected gate mechanism. The calculation method of the gated unit s (i.e. the feature vector $1\times1\times C$ in Fig. 4.2) is expressed as:

$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \sigma(g(\mathbf{z}, \mathbf{W})) = \sigma\left(\mathbf{W}_2\delta\left(\mathbf{W}_1\mathbf{z}\right)\right) \tag{4.2}$$

where σ denotes sigmoid function, δ is ReLu function. $\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{r}\times C}$, $\mathbf{W}_2 \in \mathbb{R}^{C\times\frac{C}{r}}$ are the weight matrices of the two fully connected layers respectively. r is the number of hidden layer nodes in the middle layer, the reference paper [67] points out that this value is 16. After obtaining the gate control unit s, the final output $\tilde{\mathbf{X}}$ is expressed as s and $\mathbf{M}$ that is a group matrix of μ. The scale operation $\mathbf{F}_{\text{scale}}(\cdot,\cdot)$ in Fig. 4.2:

$$\tilde{x}_c = \mathbf{F}_{\text{scale}}\ (\mu_c, s_c) = s_c \cdot \mu_c \tag{4.3}$$

where $\tilde{x}_c$ is a feature map of a feature channel of $\tilde{\mathbf{X}}$, $\tilde{\mathbf{X}} = [\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_c]$, the activation s_c is a scalar value in the gating unit $\mathbf{s}$. $\mathbf{F}_{\text{scale}}\ (\mu_c, s_c)$ denotes channel-wise multiplication between the s_c and the feature map μ_c.

The aforementioned content constitutes the entirety of the SE blocks algorithm. SE blocks can be comprehended from two distinct viewpoints: SE blocks learn dynamic priors for each Feature Map. SE blocks can be viewed as attention mechanisms applied in the context of feature maps. This is due to the core principle of the attention mechanism, which also involves learning a set of weights.

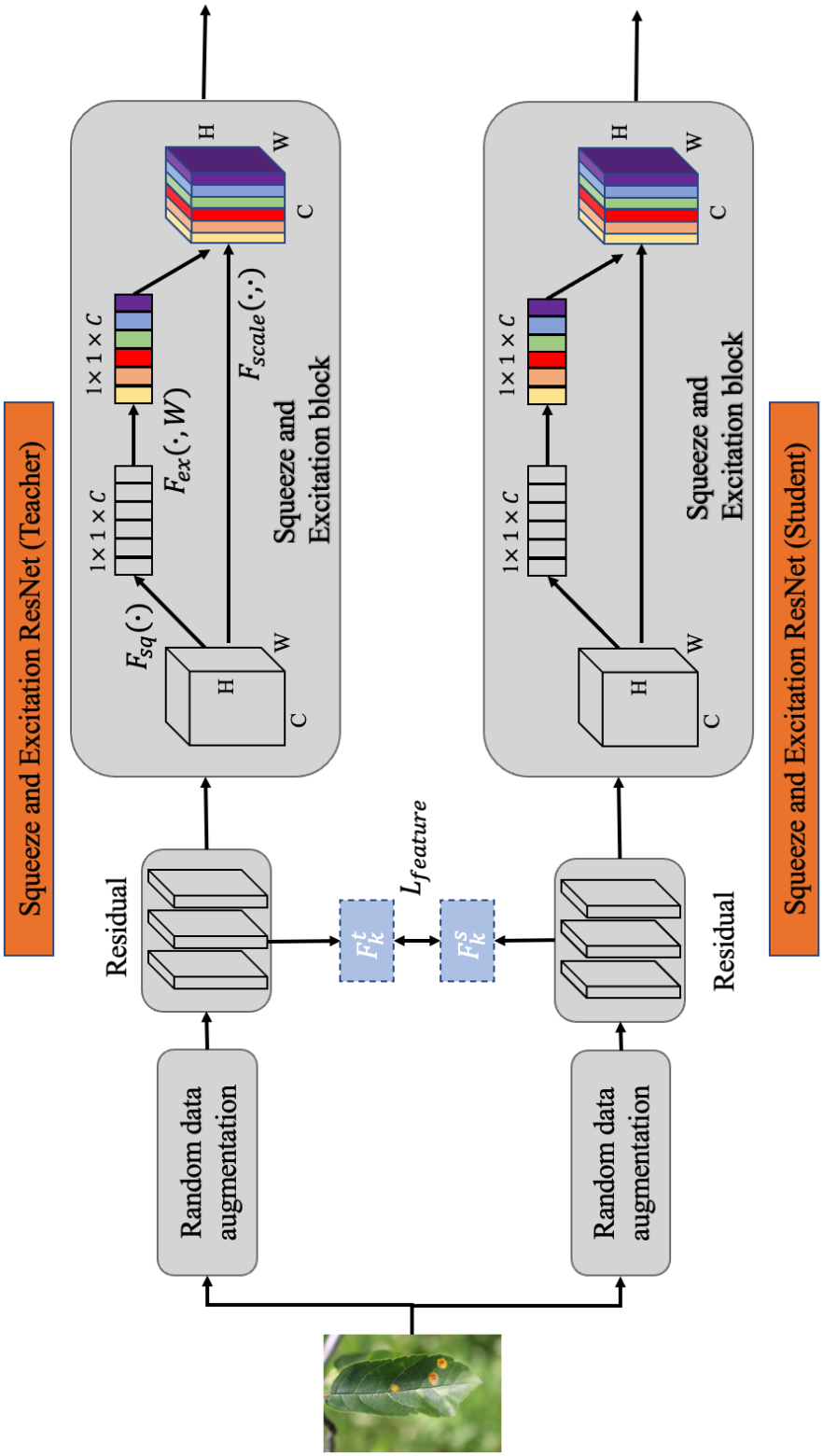


Figure 4.2: *The architecture of the fusion features based on Squeeze and Excitation Residual network.*

4.3 Knowledge Distillation

Knowledge distillation (KD) is a robust technique employed in network compression, encompassing the utilization of a sophisticated pre-trained teacher network to impart a guiding signal for training a more lightweight student network [16, 184]. This approach addresses the challenge of producing efficient models without significant loss in performance. Within KD, two primary methodologies emerge: logits distillation and feature distillation, both geared towards conveying the teacher network's knowledge to the student network. Logits distillation, sometimes referred to as soft-label or target distillation, is a well-known approach.

A seminal work by Hinton et al. [64] introduced the concept of using the softmax outputs of the teacher network as soft labels for training the student network. The crux of this method revolves around minimizing the Kullback-Leibler (KL) divergence between the soft labels generated by the teacher network and the hard labels produced by the student network. This technique allows the student network to gain insights into the teacher network's decision-making process, leading to enhanced performance. On the other hand, feature distillation focuses on learning the intermediate-level features of the teacher network and transferring them to the student network [134]. This approach delves deeper into the teacher's insights by considering not only the final predictions but also the internal representations. By doing so, feature distillation seeks to achieve a more comprehensive transfer of knowledge.

While knowledge distillation offers a potent solution for network compression, it's worth noting that the training of the teacher network itself can be demanding in terms of time and computational resources [144]. This factor sometimes limits the practicality of knowledge distillation in scenarios where resource constraints are prominent. Despite this limitation, KD's benefits are clear: it facilitates the creation of more efficient models with the potential to rival the performance of their larger counterparts. In conclusion, knowledge distillation serves as a bridge between complex teacher networks and leaner student networks. By leveraging insights from the teacher's soft labels

or intermediate features, the student network can learn in a more efficient manner. However, the overhead of training the teacher network remains a consideration in the broader application of this technique, warranting exploration of strategies to mitigate such challenges while reaping the benefits of network compression.

4.4 Self-knowledge Distillation

In the realm of enhancing effectiveness and improving performance, Self-knowledge Distillation (Self KD) offers a unique approach that eschews the need for training additional networks [187, 50]. Instead, Self KD harnesses self-generated knowledge to elevate the capabilities of the student network. This approach is particularly intriguing as it taps into the inherent insights and nuances within the data itself to bolster the learning process. The crux of Self KD lies in its utilization of mixed soft and hard labels to train the student network. When relying solely on hard labels, there is a risk of losing valuable information present in the original data. This susceptibility to overfitting not only hampers the model's ability to generalize but also leads to a decrement in overall performance. Soft labels emerge as a key solution to this quandary, acting as a bridge to counteract the degradation of model generalization. These soft labels provide supplementary knowledge, furnishing the model with additional cues about the relationships between closely related labels, thereby contributing to improved learning outcomes [172, 21].

Diverse strategies have been devised to enhance Self KD methodologies. For instance, a self-attention distillation approach [66] leverages attention maps as soft targets, enriching the learning process for tasks like lane detection. Snapshot distillation [178] emerges as a potent technique to prevent underfitting, effectively amplifying the dissimilarity between teacher and student networks, thereby facilitating more robust learning. A novel Self KD method has even been proposed, which involves redefining the probabilities of soft labels throughout the training process [58]. These variations in self-distillation techniques [157, 182, 192] all revolve around the concept

of soft labels and regularization, highlighting their significance in enhancing knowledge transfer. However, a caveat emerges when dealing with deeper teacher networks. As the teacher network's depth increases, the knowledge encapsulated within both soft and hard labels becomes less sufficient. The complexity of deeper architectures calls for more sophisticated mechanisms to ensure effective knowledge transfer. This opens up avenues for future research to explore innovative approaches that can bridge the gap between complex teacher networks and the learning capacity of student networks.

In summation, Self KD presents a compelling proposition by tapping into self-generated knowledge to enhance learning. Through the strategic use of mixed soft and hard labels, this approach not only combats overfitting but also enriches the learning process. As the field of Self KD advances, addressing the challenges posed by deeper teacher networks will undoubtedly be a key focal point, with the potential to unlock even more effective knowledge transfer mechanisms. However, the classical self-knowledge distillation focuses only on soft label knowledge distillation [64]. The student network could ignore spatial feature information. Therefore, holistic self-distillation is proposed to learn the knowledge of the teacher network from both soft labels and spatial features.

4.5 Holistic Self-Distillation

Consider a batch of the K-class labelled dataset $\mathbf{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where N represents the number of training instance in the dataset, $\mathbf{x}_i$ is the input data and y_i is the corresponding label of $\mathbf{x}_i$.

The hard labels are fed into the Squeeze and Excitation network $H(y_i, \mathbf{p}_i)$. The cross-entropy loss function is defined as follows

$$\mathbf{L}_{CE} = \frac{1}{n} \sum_{i=1}^{n} H(y_i, \mathbf{p}_i). \tag{4.4}$$

The predictive distribution $\mathbf{p}_i$ is computed through the softmax layer that compares

the logit $f_k(\mathbf{x}_i)$ with other logits. It is formulated as

$$\mathbf{p}_i(k) = \frac{\exp\left(f_k\left(\mathbf{x}_i\right)/\tau\right)}{\sum_{j=1}^{K}\exp\left(f_j\left(\mathbf{x}_i\right)/\tau\right)}, \tag{4.5}$$

where $f_k(\mathbf{x}_i)$ represents the corresponding logit of the k and the temperature constant τ is normally set to 1, , which represents the standard Softmax function without any modification. Setting the temperature τ to 1 in the Softmax function during knowledge distillation ensures the preservation of the teacher model's original prediction confidence levels, serving as a baseline for evaluating the effects of temperature scaling. This standard setting avoids the over-smoothing of probability distributions, maintaining the integrity of the teacher's high-confidence predictions. It is particularly useful in scenarios where the direct guidance of the teacher model's raw predictions is deemed optimal for the student's learning. Using τ=1 allows for a clear comparison between the original teacher model outputs and the effects of applying temperature scaling to distil knowledge to the student model.

Using the Kullback-Leibler (KL) divergence, it optimizes the student network [64], which minimizes the loss between soft label $\mathbf{p}_i^t$ and $\mathbf{p}_i^s$ generated by student and teacher respectively:

$$\mathbf{L}_{KD} = \frac{1}{n}\sum_{i=1}^{n}\tau^2 \cdot D_{KL}\left(\mathbf{p}_i^s \| \mathbf{p}_i^t\right). \tag{4.6}$$

Feature maps often contain the context and spatial information of images. Instead of training mixed soft and hard labels alone, our proposed method utilizes feature map information. The proposed method encourages the student network to learn discriminative features between soft label x_i and hard label x_j. Motivated by the hint loss from FitNet [134], the squared l_2-norm is employed for teacher feature maps $\left\{\mathbf{F}_k^t(\boldsymbol{x}) \in \mathbb{R}^{H\times W\times C}\right\}_{k=1}^{K}$ and student feature maps $\left\{\mathbf{F}_k^s(\boldsymbol{x}) \in \mathbb{R}^{H\times W\times C}\right\}_{k=1}^{K}$. Since the same size, the feature maps can directly compute with each other. Consequently, feature fusion is defined as:

$$\mathbf{L}_{feature} = \sum_{k=1}^{K} \frac{1}{HWC} \left\| \mathbf{F}_k^t(\boldsymbol{x}) - \mathbf{F}_k^s(\boldsymbol{x}) \right\|^2. \tag{4.7}$$

A good student network is able to learn holistic knowledge from feature fusion and probabilities of soft labels. The student network is trained to optimize two stages of loss:

$$\begin{aligned} \mathbf{L}_{stage1} &= \mathbf{L}_{CE} + \mathbf{L}_{KD}, \\ \mathbf{L}_{stage2} &= \mathbf{L}_{CE} + \mathbf{L}_{feature}. \end{aligned} \tag{4.8}$$

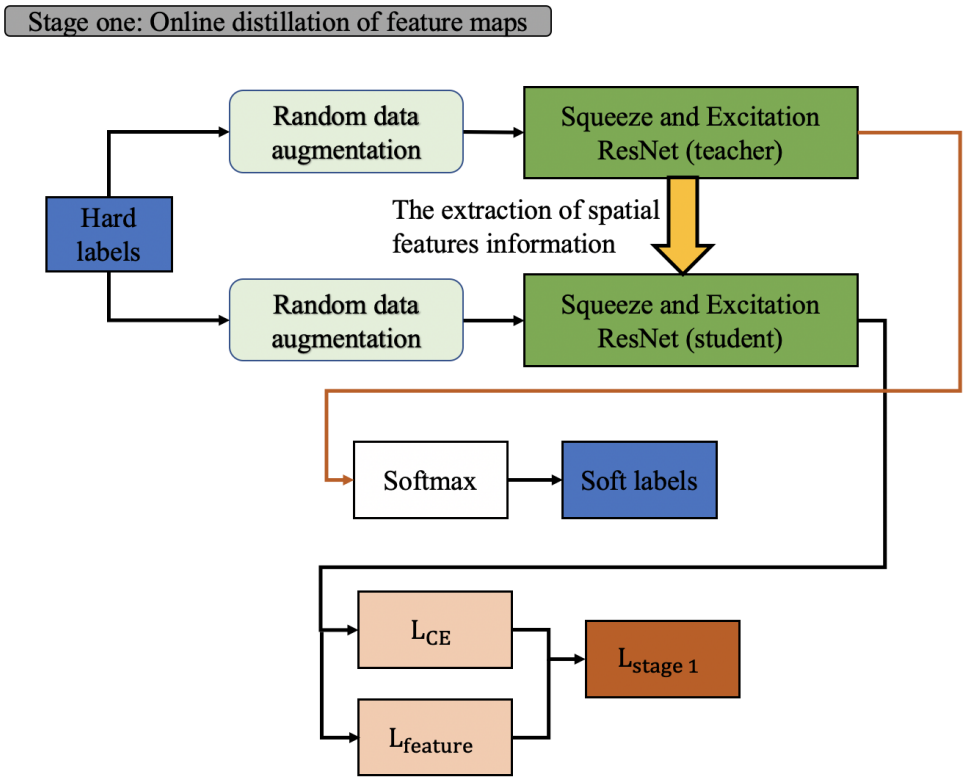
The $\mathbf{L}_{CE}$ is the cross-entropy (CE) loss between hard labels and results. In short, the Squeeze and Excitation network distil soft labels and feature maps. The Squeeze and Excitation network is trained by a new training dataset with mixed soft and hard labels. Meanwhile, the distilled feature map is involved in the loss function. The whole training process is the holistic distillation visualized in Fig. 4.3.

4.6 Experimental Results and Discussion

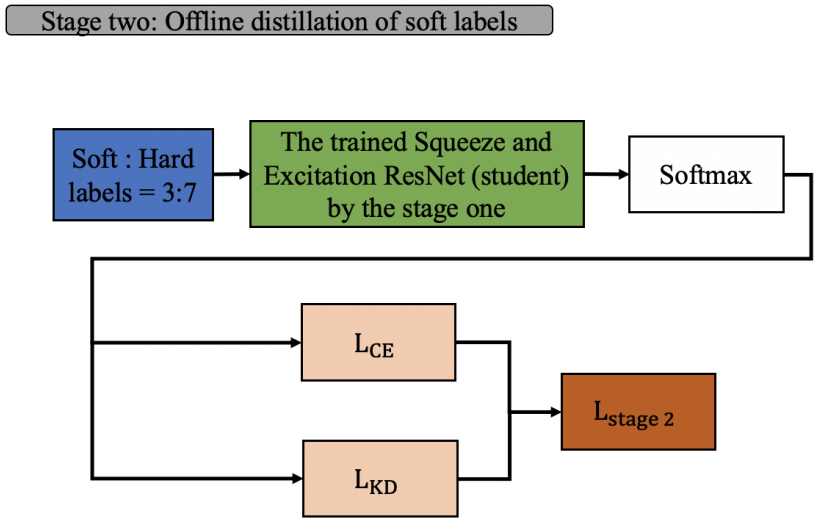
4.6.1 Datasets and Implementation Details

The plant pathology datasets [163] are available at the Kaggle community and are a part of the Computer Vision and Pattern Recognition (CVPR) Fine-Grained Visual Categorization (FGVC) workshop 2020 and 2021. The Plant Pathology 2020 dataset contains 3,651 high-quality RGB images of four apple foliar classes: healthy, scab, rust and multiple diseases. These images are captured under different illumination, angle, surface and noise conditions (Fig. 4.4). The plant pathology of FGVC 2021 increased the images to the number of 23,249 and added two classes of disease powdery mildew and frog eye leaf spot.

The proposed method is evaluated over these two datasets. The 3,651 images of Plant Pathology 2020 are used to train the model. The model performance is tested on the hidden dataset of the Kaggle leaderboard. The Plant Pathology 2021 dataset is



(a)



(b)

Figure 4.3: *The diagram of the holistic self-distillation method*

divided into train and test data with a ratio of 6:4. The teacher network is essentially the same as the SE-ResNet-50. The network is used in all experiments and is pre-trained by ImageNet [137].

In the first stage, the networks of teacher and student have trained simultaneity through the same dataset with random data augmentation. The 12 types of data augmentation are randomly applied, such as compose, resize, random brightness, different blur and flip etc. The networks will generate different feature maps for the same image. The student network is trained by minimising feature loss. Meanwhile, the teacher network generates soft labels. Then, this work adopt 30% soft label and 70% hard label to train the student network that is pre-trained by stage one. The whole stage is named holistic self-distillation.

Table 4.1: *A performance of different classes on Plant Pathology 2021.*

classes	Precision (%)	Recall (%)	F1 score (%)	Test number
Healthy	93.34	97.69	95.46	1950
Multiple diseases	86.56	73.91	79.74	1142
Powdery mildew	90.65	89.45	90.04	455
Scab	92.73	93.34	93.03	1846
Rust	88.58	89.54	89.05	736
Frog eye leaf spot	88.30	92.44	90.32	1323
Macro avg	90.03	89.39	89.61	7452
Weighted avg	90.62	90.73	90.58	7452

4.6.2 Performance Validation Results and Analysis

his section will test the performance of the holistic self-distillation on Plant Pathology 2020 and 2021 datasets with the SE-ResNet-50 network. All these experiments were run under the PyTorch framework over two NVIDIA Tesla K80 GPUs.

The performance of the method is shown with different datasets in Table 4.2. ResNet-50 (SOTA) achieved accuracies of 97.34% for Plant Pathology 2020 and 89.98% for Plant Pathology 2021, serving as a strong baseline for comparison. SE (teacher model) shows a slight improvement over the SOTA, with accuracies of 97.96% and

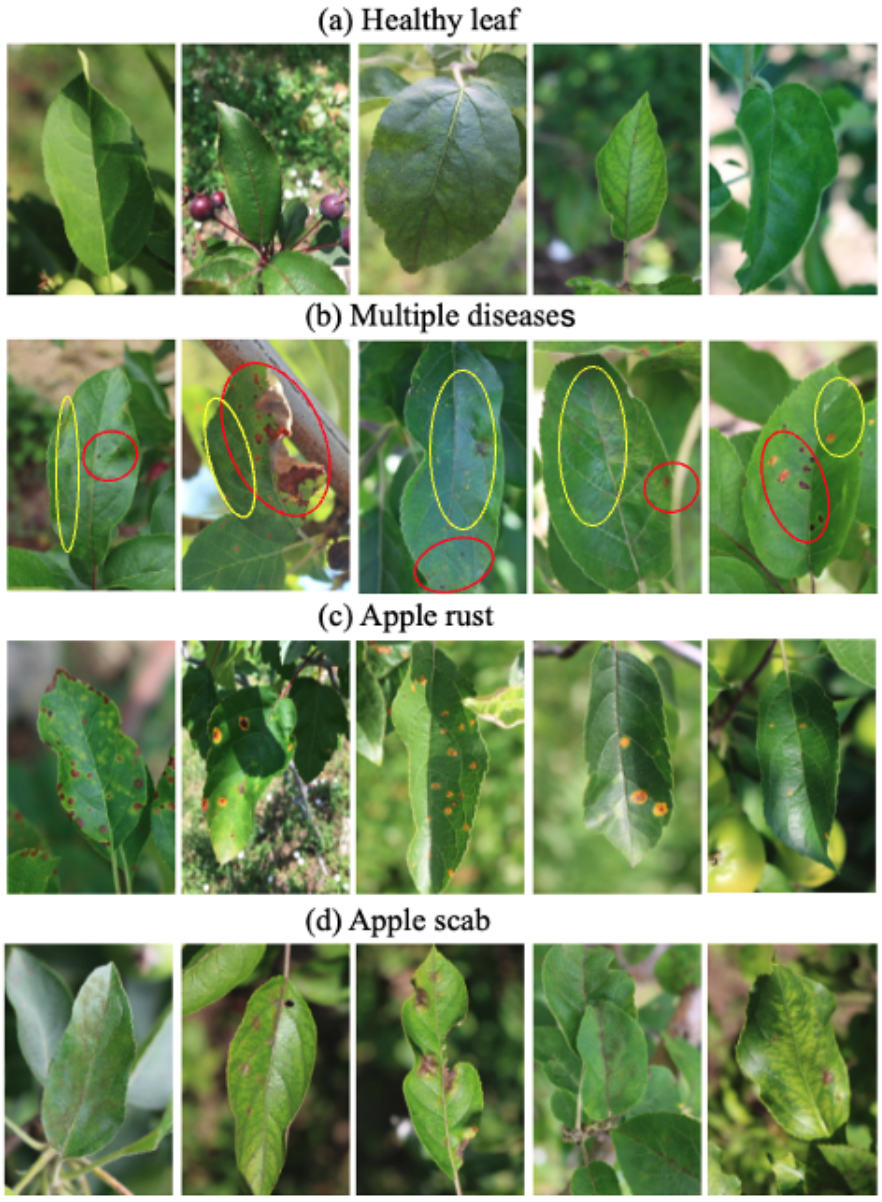


Figure 4.4: *Sample images from the fine-grain plant pathology datasets [163] showing the different symptoms (a) healthy leaf, (b)multiple diseases(red with rust, yellow with scab), (c) apple rust, (d)apple scab.*

90.48% for the 2020 and 2021 datasets respectively, indicating the effectiveness of the Squeeze and Excitation networks in capturing relevant features. SE + KD (Knowledge Distillation) marginally improves upon the teacher model, reaching accuracies of 97.97% for 2020 and 90.51% for 2021, demonstrating that traditional knowledge distillation contributes to performance enhancement. SE + HSD (Holistic Self-Distillation), the proposed method, outperforms all the other models with accuracies of 98.22% for 2020 and 90.72% for 2021. This indicates a significant improvement and highlights the effectiveness of the holistic self-distillation approach in leveraging both spatial features and soft label knowledge for fine-grained classification tasks. The incremental improvements in accuracy from the ResNet-50 model to the SE + HSD model illustrate the effectiveness of integrating Squeeze and Excitation networks with knowledge distillation techniques, and especially the proposed holistic self-distillation method. The highest accuracy achieved by the proposed method underscores its potential for enhancing fine-grained classification performance, particularly in the challenging domain of plant pathology where subtle differences between classes must be accurately discerned. This analysis reinforces the contribution of the proposed method to advancing the state-of-the-art in plant pathology classification.

Table 4.2: *A performance comparison on Plant Pathology 2020 and 2021 in terms of accuracy (%).*

Method	Plant Pathology 2020	Plant Pathology 2021
ResNet-50 (SOTA)	97.34%	89.98%
SE (teacher)	97.96	90.48
SE + KD	97.97	90.51
SE + HSD	**98.22**	**90.72**

Table 4.1 shows the experimental results of the HSD method for each class in Plant Pathology 2021. Three metrics [52] are applied to each class, which is computed by True Positive (TP), False Positive (TP), True Negative (TN), and False Negative (FN). The precision [177] indicates the predicted positive is the true positive. The recall [51] represents the correct prediction in positive samples. The F1 score finds a balance

between both precision and recall. Combining the above metrics, the macro average computes the arithmetic mean of the metrics of each class. The weighted average takes into account the weight of each class [124]. Among them, HSD achieves brilliant performance in all the classes. The healthy class gets the best results within three metrics over 1,950 test images. The multiple diseases class are prone to be misclassified.

This work further visualizes the performance of the HSD method in Fig. 4.5. The Receiver Operating Characteristics (ROC) curve is usually used to measure the performance of a model by True Positive (TP) rate and False Positive (FP) rate [69]. The ROC curve has robustness even though the imbalanced positive and false samples hardly change the shape of curves [107].

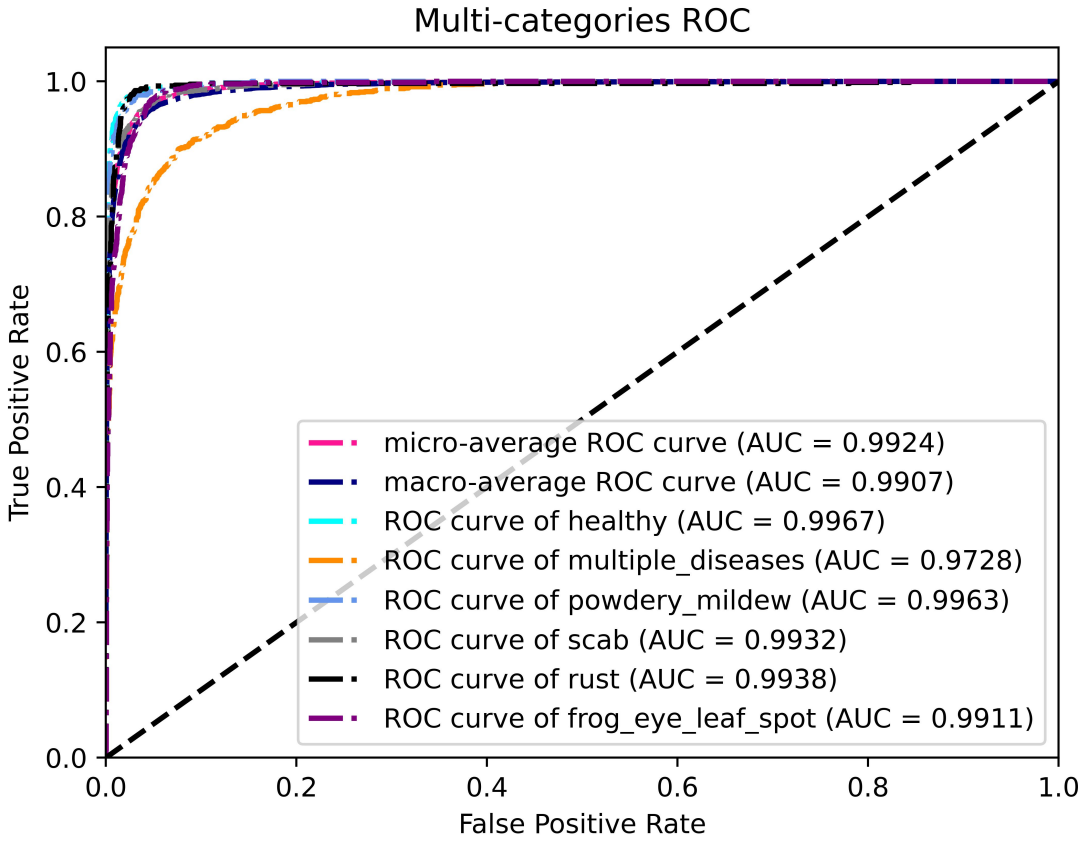


Figure 4.5: *The ROC curves of the Plant Pathology 2021. The AUC (Area Under Curve) is defined as the area under the ROC curve [107].*

Apparently, if the ROC curve closes to the upper left corner with a high value of

TP and a low value of FP, it represents the high performance of the classifier. As shown in curves of Fig. 4.5, the HSD method can effectively classify the diseases with robust ability. The multiple diseases ROC curve is obvious fluctuations that match the class accuracy in Table 4.1. the macro-average and micro-average ROC curves are calculated to evaluate the overall characteristics.

Additionally, the confusion matrix is used to visualize the performance of the proposed method. Each row of the confusion matrix indicates the true label and each column indicates the predicted label [171]. As seen in Fig. 4.6, the confusion matrix serves as a vital instrument in data analysis, offering a comprehensive view of the relationships between classes within a given dataset.

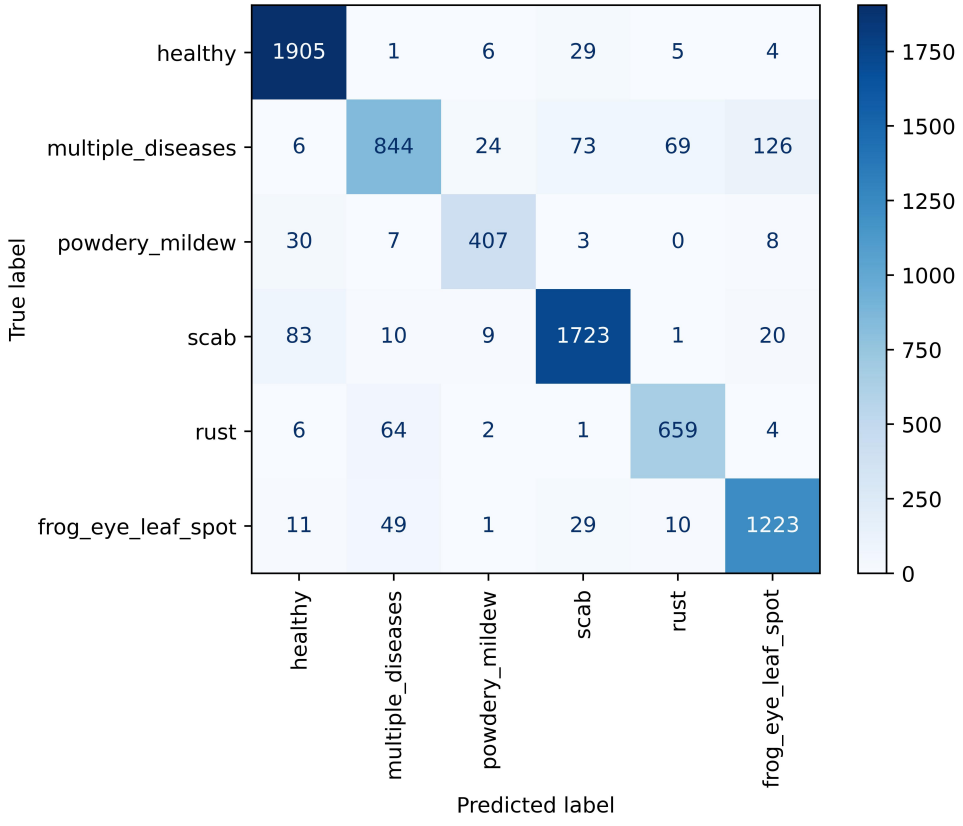


Figure 4.6: *The confusion matrix on the Plant Pathology 2021 dataset.*

In the context of the Plant Pathology 2021 dataset, this matrix provides valuable insights into the performance of an algorithm designed for disease classification. In

the matrix, the diagonal represents true predictions, effectively showcasing instances where the algorithm correctly identified the disease class. For instance, it's noteworthy that the rust disease class exhibits strong performance, with 1,723 images correctly classified. This indicates a high level of accuracy in recognizing rust disease. However, it's also crucial to examine the elements outside the diagonal, as they represent cases of misclassification. For rust disease, these misclassifications are particularly interesting. The matrix shows that 73 images were misclassified as multiple diseases, suggesting that the algorithm sometimes struggles to distinguish rust disease from other classes. This is an area that may warrant further investigation and algorithm refinement. This issue isn't unique to rust disease. The confusion matrix reveals that there are other classes that are susceptible to misclassification. For example, it's evident that powdery mildew and scab diseases can be mistaken for one another. Similarly, instances of healthy plants are sometimes incorrectly labelled as having frog eye leaf spot. These misclassifications underscore the complexity of disease classification, where subtle visual cues can lead to erroneous associations.

Notably, all diseases in the dataset seem to be prone to misidentification as multiple diseases. This suggests the existence of shared visual characteristics among different plant diseases. Identifying these shared traits and developing more sophisticated algorithms capable of making nuanced distinctions is an ongoing challenge in this field. What's particularly promising is that the HSD method method performs well across all classes. This could be attributed to its learning ability of distillation to handle the intricacies of disease classification and reduce the risk of misclassifications.

4.7 Summary

This chapter delves into the innovative concept of holistic self-distillation (HSD), a method that emerges as a promising solution to address the longstanding bottleneck associated with fine-grained classification within the realm of plant pathology. The main idea is to leverage the capabilities of a squeeze and excitation network in order to

tackle the intricate challenges posed by this specialized field. Fine-grained classification is a crucial task in plant pathology, one that demands a high level of precision and an in-depth understanding of subtle differences among various plant diseases. Achieving this precision often requires advanced machine learning techniques, and HSD presents a novel approach to meet these demands. The HSD method utilizes the Squeeze and Excitation (SE) network to take a unique approach by incorporating both soft and hard labels from the training dataset. These labels are used to guide the network towards a more comprehensive understanding of the subtle characteristics that distinguish different plant diseases. It is important to note that the feature of the HSD network is not only learned from label information but also extracted from an SE teacher network. This teacher network acts as a source of valuable feature knowledge, which is then distilled into the SE student network. One of the most significant advantages of the HSD method is its ability to simultaneously harness label knowledge and feature knowledge. While label knowledge helps the network understand the specific classes and classifications of plant diseases, feature knowledge enables the network to identify the key characteristics and patterns that differentiate these diseases. This dual-source learning strategy enhances the network's comprehension of fine-grained distinctions, making it a highly effective tool in plant pathology. The student network, which is trained using the HSD method, becomes a master of holistic knowledge. By combining label and feature knowledge, it is equipped to explore intricate details within the dataset. This holistic approach sets the HSD method apart from traditional classification techniques and empowers it to make nuanced and accurate predictions regarding plant diseases. To validate the efficacy of the HSD method, it has been rigorously tested on two public plant pathology datasets. The results have been nothing short of impressive, demonstrating that this approach significantly enhances the performance of fine-grained classification. These positive outcomes underscore the potential of the HSD method in contributing to the advancement of plant pathology research and applications.

In conclusion, the Holistic Self-Distillation (HSD) method, built upon the foun-

dation of the squeeze and excitation network, represents a significant leap forward in the field of fine-grained plant pathology classification. By combining label and feature knowledge and training the student network to grasp holistic insights, HSD shows great promise in the way of disease recognition and classification in the plant sciences. Its successful application in public Plant Pathology 2021 and 2022 datasets with the accuracy of 98.22% and 90.72% respectively, and ongoing research will undoubtedly bring forth even more innovative self-distillation deep learning methods, pushing the boundaries of knowledge and expertise in plant pathology. After exploring the loss function and learning strategy in deep learning methods, the following chapter will design a novel deep neural network framework to solve the semantic segmentation task.

Chapter 5

Explore Deep Learning Architecture for Semantic Segmentation in Automating Morning Glory Plant Harvesting

Computer vision and deep learning have made substantial progress in the areas of agriculture and smart farming, particularly for enhancing crop production using image segmentation techniques for crop yield prediction. Further improvements to crop yield prediction results can be achieved by developing accurate and efficient methods. In response to such demands, this chapter proposes a novel convolutional neural network architecture, called Densely Connected SegNet (D-SegNet) and demonstrates its advantages on plant segmentation using a new morning glory plant dataset, and also on a complimentary publicly available dataset to promote research in this direction. The D-SegNet is evaluated using 10-fold cross validation. It achieves performance better than the state-of-the-art SegNet algorithm. The evaluated precision, recall and F1-score values are 98.20%, 90.64% and 94.26%, respectively, for the morning glory plant dataset.

The intersection over union (IoU) value in the image segmentation tasks is 90.56%. A series of experiments on the morning glory plant dataset as well as on the publicly available dataset were conducted. The results show that the proposed method achieves accurate segmentation results and can be useful for assessing the plant weight during harvesting. In summary, this new plant segmentation network, D-SegNet, could form an important component of future cloud-based machine learning systems to predict crop yield from noisy smartphone images taken in the field.

5.1 Semantic Segmentation Based on Deep Learning Methods

As the global population increases it is vital to tackle the challenges of agricultural production to improve food security and make production efficient [46]. Smart farming [167] is one promising area that supports the growth of agriculture production and its efficiency by leveraging technological advances in sensing, monitoring, and artificial intelligence [7, 140]. Within the broad area of smart farming, crop planning, monitoring, and yield prediction [111] have been considered some of the dominant challenges in promoting coordinated efficiency and maximizing the economic potential of modernizing agriculture.

Yield prediction is one particularly important aspect of an agricultural business for the following reasons. The buyers of fresh products often plan the procurement based on the sale estimates or customers' orders. Suppose the buyers are aware that there will be a shortage or excess of production from the farmers. In that case, they can plan ahead of time by seeking additional products from retailers and buying or selling the excess to other buyers. By gaining more accurate yield information, the buyers can manage their incoming stock effectively. This can result in keeping the price for the farmers high and at the same time the management cost for the buyers low. With an accurate yield prediction, the buyer of the product can save costs through better planning their sourcing strategy and simultaneously better serve the needs of the

This chapter corresponds to the publication-Jingxuan Su, Sean Anderson, Mahed Javed, Charoenchai Khompatraporn, Apinanthana Udomsakdigool, Lyudmila Mihaylova, "Plant leaf deep semantic segmentation and a novel benchmark dataset for morning glory plant harvesting", *Neurocomputing*, vol. 555, October 2023.

customers. This further promotes the buyer's business, leading to close collaboration between farmers and buyers.

This chapter presents a solution for crop yield prediction using computer vision and image processing methods. Image processing methods can assist precision agriculture, for example in plant phonology [166], automatic segmentation of leaf images [175, 15] and yield prediction [140]. However, current methods for segmentation are not sufficiently accurate to be reliable and useful in yield prediction [3], especially using simple methods such as k-means clustering [80]. There is a wealth of machine learning methods for image analysis [30, 141] - from support vector machines (SVM) [55] and artificial neural networks (ANN) [139] to deep learning (DL) [87]. The conventional neural learning methods [108] mentioned above aim at solving different tasks, including object detection and segmentation. Due to the accuracy demands of yield prediction, image-based, semantic segmentation at a pixel level is an appealing approach and is becoming popular in this application domain [83, 13]. However, standard convolutional neural networks [53] face often problems such as slow convergence, and loss of vital feature information when using several convolutional layers, leading to inaccurate segmentation results. Thus, an optimized network architecture needs to be proposed. At present, most semantic segmentation methods conduct research on expanding the depth [149, 160] or width [154, 185] of network architectures. Deep learning methods [188, 45], as data-driven methods, provide efficient solutions for big data and automated feature extraction, with high performance and accuracy [141, 150].

This chapter proposes a new type of semantic segmentation architecture for yield prediction, which comprises encoder-decoder [22] and dense block [68] structures. The encoder is comprised of thirteen convolutional layers of a VGG-16 network [149]. Each encoder is linked to a decoder and hence there are thirteen corresponding decoders. Typically, the convolutional layers of the encoder comprise batch normalization and rectified linear unit (ReLU) non-linear operations, followed by non-overlapping max pooling and sub-sampling layers. The sparse encoding due to the pooling process is up-sampling in the decoder using the max pooling indices in the encoding sequence.

This has the important advantage of retaining class boundary details in the segmented images and also reducing the total number of model parameters. The model is trained end to end using the stochastic gradient descent method [14].

The structure of a dense block includes a concatenation between layers. The concatenation enhances feature reuse, which directly connects every layer. This connection pattern not only propagates features to the next layer but to every layer in a dense block. A novel deep learning architecture is proposed here for dense semantic segmentation, which extends the standard SegNet architecture with dense blocks, hence this architecture is called D-SegNet. This new type of architecture is demonstrated to outperform the state-of-the-art algorithms and achieve high accuracy on crop yield prediction.

5.2 D-SegNet Architecture

This section presents a new architecture called D-SegNet for semantic segmentation, which augments the encoder-decoder architecture of SegNet [22] with dense blocks [68]. The D-SegNet architecture combines the improved capability of feature extraction from the dense block with the computational efficiency of the SegNet encoder-decoder architecture, including efficient memory use and fast computation. The encoders are based on the 13 convolutional layers of the VGG-16 network [149]. The decoder places corresponding layers in reverse. Dense blocks are added after each encoder or decoder.

5.2.1 Encoder-Decoder Framework

The encoder-decoder architecture of D-SegNet is illustrated in Fig. 5.1. The encoder consists of five blocks which include 'Convolution (Conv)+Batch Normalisation (BN)+ Rectified Linear (ReLU)' operations and they form the Downsample Blocks. In each Downsample Block, a convolutional layer uses a 3×3 kernel and a stride equal to 1. Batch normalization is then applied to the output of the convolutional layer. Meanwhile, the max pooling with 2×2 window and stride 2 achieve translation invariance.

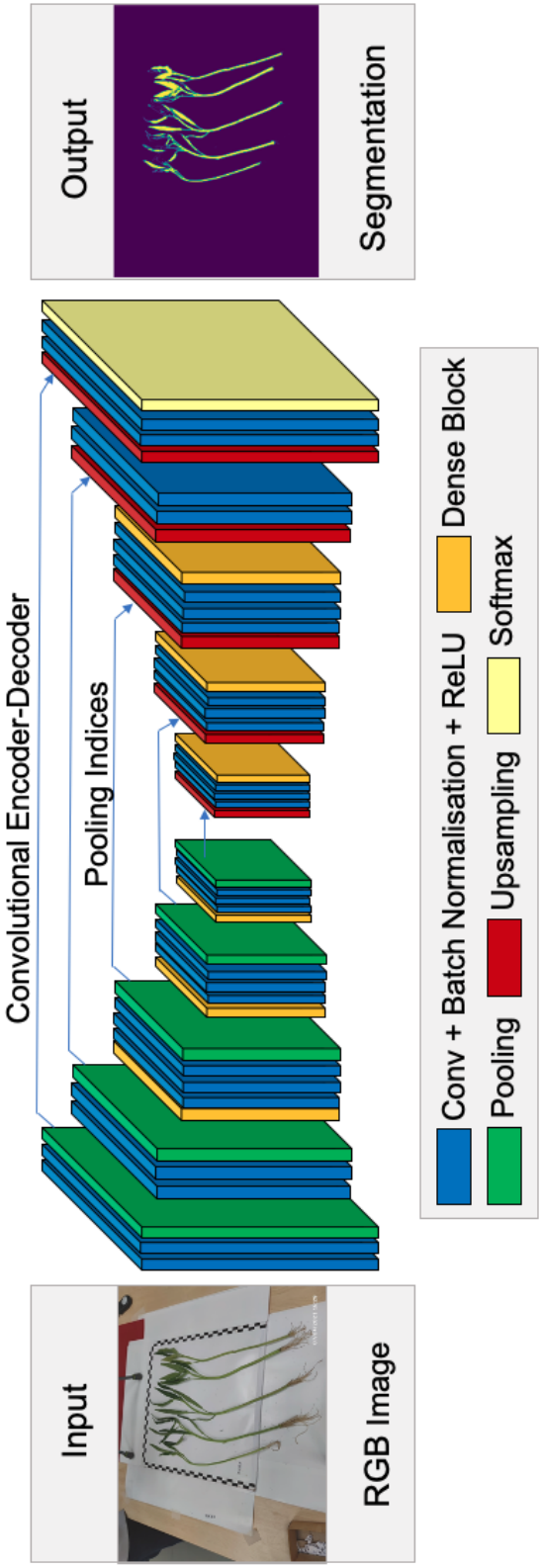


Figure 5.1: *A schematic of the D-SegNet architecture.*

Therefore, the size of feature maps is changed regularly from 224×224 to 7×7 after the image is passed through five Downsample Blocks. The role of the encoder is to generate feature maps with semantic information.

The decoder which contains five Upsample Blocks upsamples the feature maps from the encoder output by using the memorized max-pooling indices to produce sparse feature maps. The size of these sparse feature maps is rescaled to the size of the original image. The purpose of the rescale operation in the decoder is to map the sparse feature maps to the input image to implement pixel-by-pixel classification.

5.2.2 Dense Block

Recent CNNs benefit from very deep convolution layers to capture rich feature representations, due to the fact that 'the deeper the network, the better [63]. However, training deep neural networks may need a huge amount of time and computational resources, due to redundant feature maps. The dense block proposed in [68] is considered an effective method to address this challenge. It encourages feature reuse and makes it efficient to train a very deep network. The dense block improves the flow of features throughout the network by connecting all layers with each other. In addition, it also enhances feature propagation. The dense connectivity in a dense block can be formulated as follows:

$$x_d = H_d\left([x_0, x_1, \ldots, x_{d-1}]\right), \tag{5.1}$$

where x_0 to x_d are feature maps from layer 0 to layer d in a dense block, respectively. Here $[x_0, x_1, \ldots, x_{d-1}]$ refers to the concatenation of feature maps from x_0 to x_{d-1}, H_d denotes a non-linear transformation, including BN [70], ReLU, and Conv operations. The BN size is set up to 4 to keep a lightweight network. In this way, each layer within a dense block has direct connections with all subsequent layers, as shown in Fig. 5.2. According to (5.1), the channel number of feature maps in the d^{th} layer of each dense block is $k_0 + k \times (d-1)$, where k_0 is the number of input channels. The growth rate is

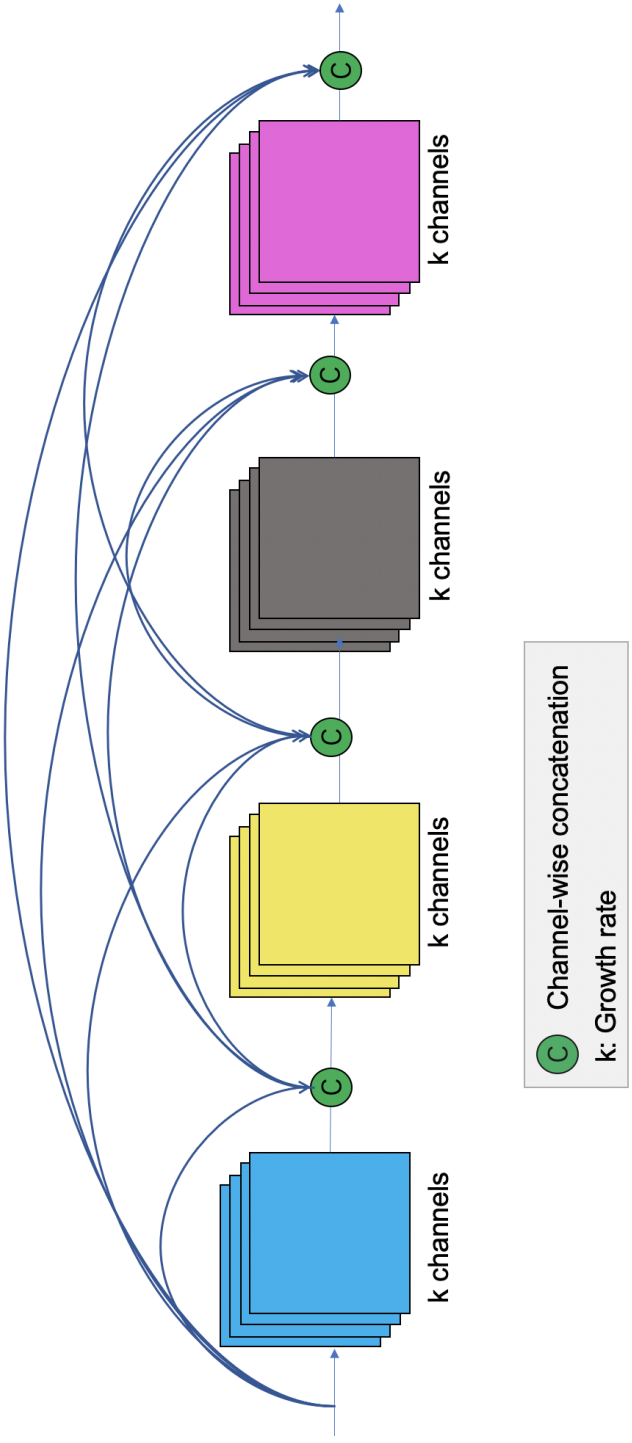


Figure 5.2: *The general dense block model proposed in this chapter. Feature map sizes match within each block.*

termed k, following the same notation as in [68]. In [68], due to the feature map size, k is set up equal to 32. Each dense block has 4 layers with a growth rate of $k = 32$. Table. 5.1 gives all parameters of the D-SegNet algorithm.

5.2.3 Optimal Semantic Segmentation Model (D-SegNet)

The D-SegNet as a novel architecture has an outstanding framework and computational capability. This powerful segmentation engine consists of a deep convolutional encoder-decoder architecture, dense blocks, and a pixel-level classification layer. Table. 5.1 lists the details of D-SegNet architecture. From the Table. 5.1, Dense Block is directly connected to Downsample/Upsample Blocks. Specifically, 1×1 *conv* layers are set before the 3×3 *conv* layers to avoid increasing the model parameters. Therefore, the D-SegNet is a lightweight network architecture. (see Table. 5.1). The number of feature maps of an output remains $4 \cdot k$, which is the same as in [68]. Note that the feature map size only changes in the Downsample/Upsample Blocks. Thus, the feature map size remains the same in all dense blocks, which contains large spatial information in small feature maps.

Different from other semantic segmentation architectures, the D-SegNet encourages feature reuse and prevents gradient vanishing problems. The main reason is that the D-SegNet is a lightweight network, which benefits from dense blocks. Its network structure is narrow, and only needs a few parameters. The number of feature maps of each convolutional layer output in the dense block is very small, instead of hundreds of thousands of outputs like in other networks. Within the dense block, each layer has direct access to the gradients from the loss function and the original input signal. On the other hand, a dense connection is equivalent to directly connecting an input and loss at each layer. Thus, it can mitigate the phenomenon of gradient vanishing when the depth of the network is deep.

Table 5.1: *D-SegNet architecture for plant segmentation. The growth rate of k for the whole network is 32. Note that each Up/Down sampling block shown in the table corresponds to the sequence Conv+BN+ReLU+Pooling or Upsampling.*

Layers	Output size	D-SegNet (k=32)
Input layer	224×224	
Downsample Block (1)	112×112	3×3 *conv*, stride 1, 2×2 max pool, stride 2
Downsample Block (2)	56×56	3×3 *conv*, stride 1, 2×2 max pool, stride 2
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \ conv \\ 3 \times 3 \ conv \end{bmatrix} \times 12$
Downsample Block (3)	28×28	3×3 *conv*, stride 1, 2×2 max pool, stride 2
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \ conv \\ 3 \times 3 \ conv \end{bmatrix} \times 24$
Downsample Block (4)	14×14	3×3 *conv*, stride 1, 2×2 max pool, stride 2
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \ conv \\ 3 \times 3 \ conv \end{bmatrix} \times 16$
Downsample Block (5)	7×7	3×3 *conv*, stride 1, 2×2 max pool, stride 2
Upsample Block (1)	14×14	3×3 *conv*, stride 1, 2×2 max pool, stride 2
Dense Block (4)	14×14	$\begin{bmatrix} 1 \times 1 \ conv \\ 3 \times 3 \ conv \end{bmatrix} \times 16$
Upsample Block (2)	28×28	3×3 *conv*, stride 1, 2×2 max pool, stride 2
Dense Block (5)	28×28	$\begin{bmatrix} 1 \times 1 \ conv \\ 3 \times 3 \ conv \end{bmatrix} \times 8$
Upsample Block (3)	56×56	3×3 *conv*, stride 1, 2×2 max pool, stride 2
Dense Block (6)	56×56	$\begin{bmatrix} 1 \times 1 \ conv \\ 3 \times 3 \ conv \end{bmatrix} \times 4$
Upsample Block (4)	112×112	3×3 *conv*, stride 1, 2×2 max pool, stride 2
Upsample Block (5)	224×224	3×3 *conv*, stride 1, 2×2 max pool, stride 2
Output layer	224×224	

5.2.4 Loss Functions Used for D-SegNet Training

This section discusses the loss functions used to train the D-SegNet architectures. The output from the proposed architecture is a four-dimensional (4D) tensor $Z \in \mathbb{R}^{N,C,H,W}$, where N denotes the batch size of the output tensor, C represents the number of channels (or depth), H is the height and W the tensor width. Color images have three channels (i.e. C=3) for each channel red, green, and blue, also known as the red, green, blue (RGB) representation. For brevity, Z is considered to be a two-dimensional (2D) matrix of size N, C, containing elements $z_{n,c}$, which is a 2D matrix of size H, W. Prior to computing the cross-entropy loss, the segmented label map output Z needs to be converted to probabilities through the softmax function $\phi_{soft}(z_{n,c})$ in the compression layer as follows

$$\phi_{soft}(z_{n,c}) = \frac{\exp\left(z_{n,c}\right)}{\sum_{i=1}^{C} \exp\left(z_{n,i}\right)}. \tag{5.2}$$

The threshold value of the softmax function is set up to 0.6. The output of the softmax layer is then used to compute the cross-entropy loss $\mathcal{L}_{cross}$ which can be calculated from

$$\mathcal{L}_{cross}(z_{n,c}, \hat{y}_{n,c}) = -\sum_{c=1}^{C} \hat{y}_{n,c} \log(\phi_{soft}(z_{n,c})). \tag{5.3}$$

The softmax output is computed by considering the exponent of the output $z_{n,c}$ from the final layer. This is divided by the sum of the exponential outputs across the channel dimension C. Then, considering the outputs from the softmax layer $\phi_{soft}(z_{n,c})$, the respective target label for that output $\hat{y}_{n,c}$ is used to compute the cross-entropy loss. Here, the term *log* refers to the natural logarithm of the softmax layer element.

The cross-entropy loss is averaged across all N mini-batches of segmented label outputs and target labels, e.g. $\sum_{n=1}^{N} \mathcal{L}_{cross}$. The averaged loss is then backpropagated for updating the layers of the D-SegNet end-to-end using the chain rule and the update rule. The update rule sums the old weights and biases of a specific layer with the differential of the cross-entropy error with respect to the weights and biases of all

the convolutional layers multiplied by the learning rate. The learning rate set during training of the D-SegNet is 0.1.

5.3 Experimental Results and Discussion

This section describes the data used to evaluate the D-SegNet architecture and the data pre-processing steps. In order to evaluate the performance of D-SegNet, the image data needs to be annotated, which is a complex and time-consuming process, involving the training-validation data. To annotate the data, a semi-automatic labeling method was used consisting of Faster Region-based Convolutional Neural Network (Faster R-CNN) [131] and K-Means clustering [80], which are described specifically in the following Section 3.1.2. This approach improves the efficiency and accuracy of annotation. The original and labelled dataset are available on GitHub [79, 81] .

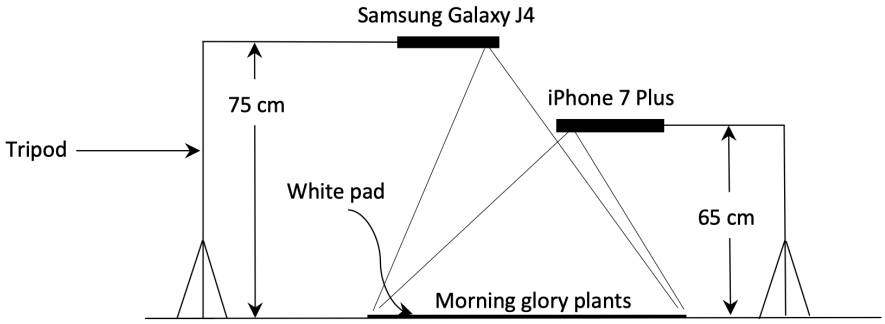
5.3.1 Morning Glory Plant Images

A total of 2,018 images of the morning glory plant were collected during different phases of the plant growth. These images and the developed deep learning approaches are aimed at helping farmers to decide when to harvest the plant, at the period when the plant is with the desired size and weight. In each of the images, there are 4 or 5 morning glory plants. The plants were laid on white space with tick marks on three edges to indicate the size of the plants. Each side tick mark is 50 centimetres long. The plants in each image were weighed and the total weight was recorded (in grams). The hardware specifications used for data collection and their relevant distances from the target (height) are summarised in Table 5.2.

Each image was taken by two smartphones, a Samsung Galaxy J4 and an iPhone 7 plus. The Samsung Galaxy J4 was installed at 65 centimetres above the plants, while the iPhone 7 Plus was installed at 75 centimetres above the plants. The images are of size $4,032 \times 3,024$ or $4,128 \times 3,096$. The Samsung Galaxy J4 mobile phone has a 13 MP rear camera with $f/1.9$ aperture, and the iPhone 7 Plus has a 12 MP rear camera

(a) (b)

(c)



(d)

Figure 5.3: *Environment and construction of image acquisition. (a)Plant culture environment; (b) and (c) Mature plant; (d) Side view of image acquisition construction.*

Table 5.2: *The specification of the data collection devices and their distances from the target*

Device	Height	Illuminance	Rear camera	Aperture
Samsung Galaxy J4	65cm	620-680 LUX	13MP	$f/1.9$
iPhone 7 plus	75cm	620-680 LUX	12MP	$f/1.8$

with $f/1.8$ aperture. Fig. 5.3 shows the plant environment and image acquisition device. All images were taken in a laboratory with an illuminance of approximately 620-680 LUX. The light was mainly from the fluorescent lightbulbs installed in the laboratory with some effect from natural light through the windows. There were only a few images in that the illuminance that fell below or above the range mentioned. This could perhaps be due to the sensitivity of the device or the shadow from the experimenters.

The experiments of automatic plant segmentation are conducted on this dataset. The dataset was collected by the team from King Mongkut's University of Technology Thonburi, at the geographical location with latitude and longitude 13.39°N, 100.29°E) in Thailand.

Image Pre-processing

In recent years, deep learning models have made remarkable strides in computer vision tasks [34]. Typically, they heavily rely on extensive data and powerful computation resources [147]. However, numerous fields have limited data availability, such as farm imagery. A widely adopted solution is data augmentation, encompassing various algorithms like geometric transformations and kernel filters. Data augmentation significantly enhances model robustness and generalization capabilities.This section primarily focuses on geometric transformations, known for their ease of implementation and safety for project images [84]. For example, rotations and flips maintain the safety of cat and dog images, but alter the digital numbers to 6 and 9. Hence, label-preserving transformations are prioritized in geometric augmentation. Horizontal axis flipping is a standard practice, validated on datasets like ImageNet and MNIST. Rotation augmentation involves rotating the image to the right or left along its axis. Translation shifts the image horizontally or vertically, effectively mitigating positional bias.

Generally, data augmentation includes the generation of additional images based on random flips, flexible rotation and different illumination. Fig. 5.4 shows the augmented data. Fig. 5.4(a) and Fig. 5.4(b) have different noise and illumination backgrounds,

(a) (b) (c)

Figure 5.4: *Morning glory plant images: (a) original image, (b) image with added noise and (c) image with a different illumination background. Images (b) and (c) are obtained as a result of the data augmentation.*

respectively. Fig. 5.4(c) is obtained after a rotation of the original image at 180°. These operations expand the original data set from 2,018 to 6,054 images, which will help to train and evaluate the segmentation networks.

Image Data Annotation

The overlap and thickness of leaves make manual annotation difficult and create challenges to the training of convolutional neural networks. Thus, a semi-automatic annotation method is applied, whose main steps are summarized in Fig. 5.5.

The Faster R-CNN algorithm [131] is applied to detect the plant. Faster R-CNN is a state-of-the-art object detection network, which consists of region-based proposal algorithms able to find the object location [40]. R-CNN is composed of a feature extraction network followed by two networks. The feature extraction network is a pre-trained CNN that forms the feature map. The first network ResNet-50 [63] is the feature extraction network of the detector. The second network is a region proposal network (RPN), which generates object proposals. The RPN decides the positive or negative of anchors by the softmax function. The bounding box regression is used to fix anchors and get precise accuracy.

The region of interesting (ROI) pooling layer processes features maps and proposals for further prediction. The area outside of the region box is set up as a black background. Next, the k-means clustering algorithm [80, 98] is applied to annotate the

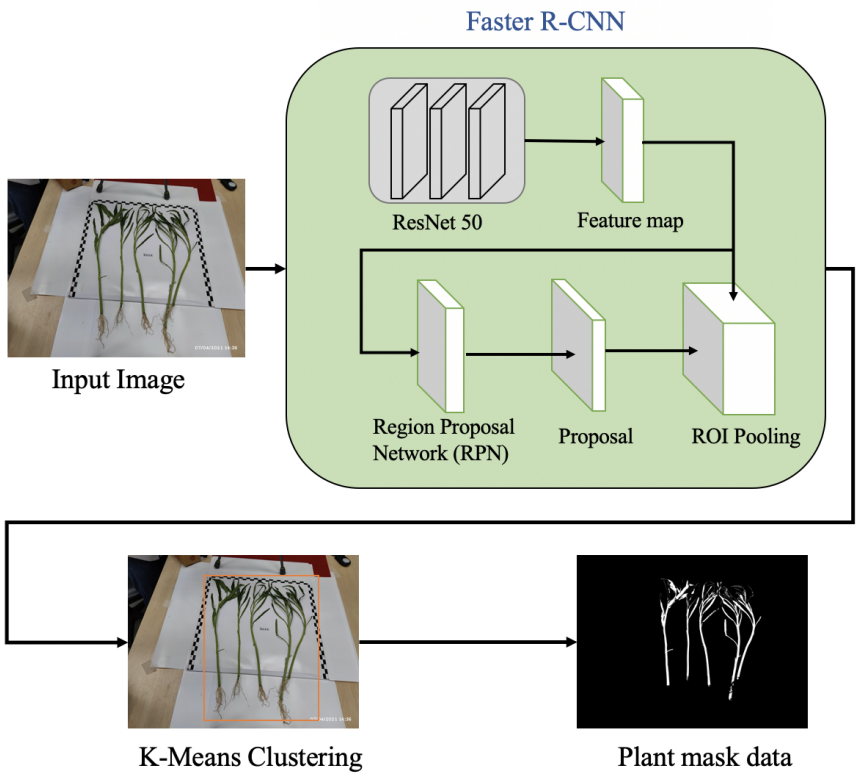


Figure 5.5: *Image data annotation steps*

(a) (b)

Figure 5.6: *Images of the morning glory plant. (a) Original image; (b) Annotated image.*

Table 5.3: *The pseudo code of k-clustering*

Algorithm 1 K-means clustering

1: Initialize the image as input pixels p(x, y) and set the number of k

repeat

2: Calculate the Euclidean distance d between centre and each pixel, as the following equation.

$$d = \|p(x, y) - c_k\|$$

3: According to the distance d all pixels are sort firstly to the nearest cluster

4: The centroid need be recalculated with the pixels of each cluster

$$c_k = \frac{1}{k} \sum_{y \in c_k} \sum_{x \in c_k} p(x, y)$$

until The error tends to converge

5: Resolute the pixels to the image

leaves within the bounding box area. Clustering, as a fundamental algorithm, partitions images into specific groups. Within this algorithm, there are various branches, with the most popular being k-means clustering. K-means clustering is a classical unsupervised learning algorithm recognized for its speed and straightforward computation [33]. In k-means clustering, it randomly selects k clusters as centroids from all classes. This algorithm comprises two distinct modules. Firstly, k centroids need to be calculated. Secondly, each point is assigned to the cluster of the nearest centroid. The Euclidean distance is the most commonly used method to determine centroid distances. Let's demonstrate the algorithm using an image. Despite the ease of implementation, k-means still exhibits shortcomings in terms of both quality and computation [90]. The primary drawback lies in the determination of the number of clusters, k. The quality of segmentation relies on the arbitrary choice of k. Additionally, computational complexity, which depends on the data volume, k, and iteration numbers, also needs to be taken into account. In this work, a fixed number k is set equal to 2.

To avoid complex data processing [101], the annotated data consider leaf pixels as white and the other as black, as shown in Fig. 5.6. It is saved in a PNG format with high resolution $4,128 \times 3,096$. The high-resolution image is stored but it is not processed by the convolutional neural network since such an image would require a large amount of memory space. This would require significant computational power from the GPU. Therefore, the size of original and annotated images is processed as

$224 \times 224 \times 3$ and 224×224.

(a) (b)

Figure 5.7: *ImageCLEF dataset: a herbarium sheet*

5.3.2 Benchmark Dataset

A popular public dataset from ImageCLEF 2021 (Pl@ntLeaves), [49] is used to evaluate our proposed segmentation method. This dataset consists of leaf herbarium and comprises 1956 images, from which 1190 images are used for training, and 256 for testing. Sample leaf images are shown in Fig. 5.7.

5.3.3 K-fold Cross Validation

Cross-validation is a resampling method that is used to evaluate models on a limited data sample [130]. In K-fold cross-validation, the entire dataset is split into K groups randomly. For every fold, one out of K subsets is chosen as the validation set and K-1

subsets are used for training. It helps to avoid the overfitting problem and to improve

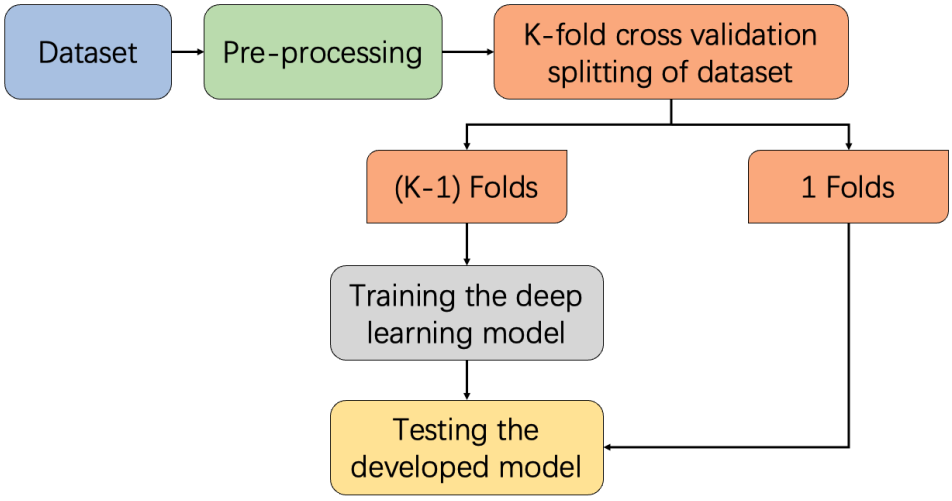


Figure 5.8: *The process of K-fold cross validation in training the deep learning model*

model performance with a small dataset. The diagram of K-folder cross validation is shown in Fig. 5.8.

Experiments are conducted for 10-fold cross-validation with the morning glory plant dataset and ImageCLEF dataset. Each dataset is divided into 10 subsets stochastically. For each cross-validation iteration, the model is trained using 9 subsets and is tested to the other subset as the validation set. The procedure is then repeated for 10-folder cross-validation. Each subsample is used once as a validation subset.

5.4 Results and Discussion

5.4.1 Evaluation Metrics

This section presents the evaluation metrics used for the performance validation of the D-SegNet algorithm. These include precision, recall, and the F1 score. The precision-recall metrics [51] allow further evaluation for classification beyond simple accuracy measures that do not take into account the problem of class imbalance.

First, the number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) samples are calculated. The TP are predictions that are classified correctly for the inspected class. The TN are other predictions that are correct but for negative samples (i.e. not inspected). Then, false positives are miss-classified negative samples and the true negatives are negative samples that are correctly classified. In addition, the IoU is also an important metric and it is used to evaluate the algorithms' performance.

The precision metric [36] is used to compute the number of correctly predicted positive classes of a classification system. The recall metric is used to define the number of correct positive predictions that are achievable from all of the positive predictions. The model evaluation should not only use the statistic metrics but also need a qualitative evaluation from the actual segmentation view. The intersection over union (IoU) [129] as an important evaluation index in semantic segmentation measures the overlap of the ground truth and prediction region. The IoU is generally calculated based on classes, which is to accumulate the IoU value of each class. The IoU value is to average the sum IoU results of each class to obtain a global evaluation. Therefore, the IoU is actually the mean value, that is, the average crossover ratio (mean IoU).

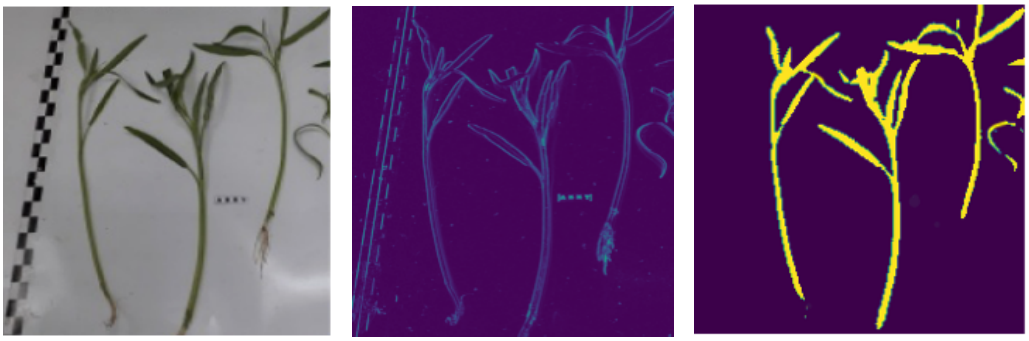
In the next section, the performance of the segmented results is discussed in detail while comparing different architectures.

5.4.2 Experimental Results

In order to evaluate the performance of the D-SegNet algorithm, it is compared with traditional edge-based segmentation and with the standard SegNet algorithm.

From Fig. 5.9, D-SegNet provides quite clear contours of the segmented plant, whereas the classical Sobel edge-based segmentation [43] visually gives less accurate results. Morphological operations [104, 121] are used to connect the edge pixels into meaningful edges.

There are two steps in morphological operations: erosion and dilation. Dilation



(a) (b) (c)

Figure 5.9: *Segmentation results. (a) Original image; (b) Segmentation based on edge detection with Sobel operator* ; (c) D-SegNet

Table 5.4: *Performance of different segmentation methods in morning glory plant*

Metrics	Based on edge Detection	SegNet-Basic	SegNet	D-SegNet
Precision_M	67.38%	96.57%	97.48%	**98.31%**
Recall_M	62.25%	88.74%	89.64%	**90.92%**
F1-score_M	64.71%	92.48%	93.39%	**94.47%**

expands the image boundaries to make sure the boundary pixels are connected. Erosion shrinks the image pixels slightly, which removes noise pixels from the object boundary. Finally, the object is segmented from the image. However, it cannot provide a coherent and precise leaf shape. Although this does not mean traditional methods become obsolete, D-SegNet indeed solves complex segmentation problems with super-human accuracy. The big data training and plentiful feature computation provide descriptive and salient features to predict the underlying patterns. Thereby D-SegNet gets a better performance than the traditional computer vision method.

In this chapter, the proposed approach, SegNet-Basic [9] and SegNet [10] methods are trained. In order to provide a fair comparison of the performance of the considered deep learning architectures, the same GPU and environment system configurations are applied to all of them. A 10-fold cross-validation strategy is applied to each data set. Then 9 folds are used for training, and the other one fold is used for testing. The

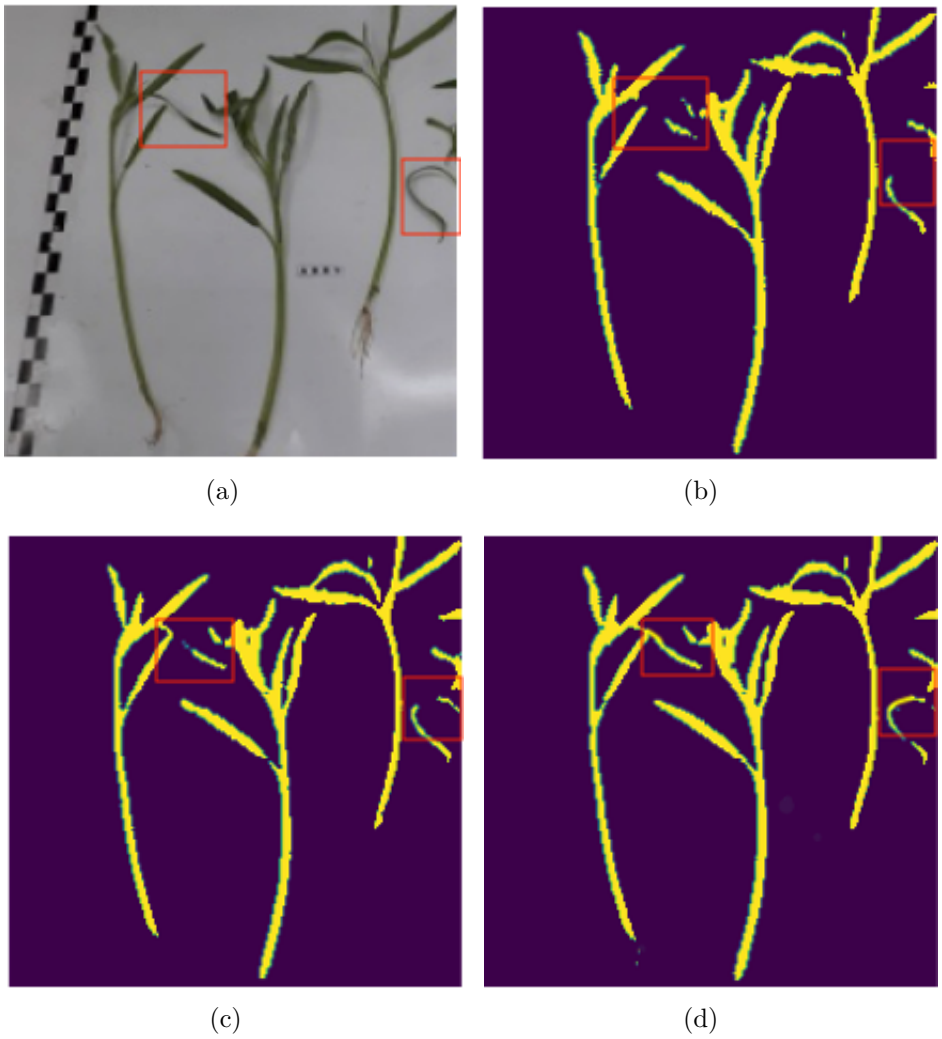


Figure 5.10: *Results of transitional steps in proposed model for automating the segmentation process. (a) Original image; (b) SegNet-Basic; (c) SegNet; (d) D-SegNet;*

Table 5.5: *Performance of different segmentation methods in ImageCLEF (Pl@ntleaves) dataset*

Metrics	Based on edge detection	SegNet-Basic	SegNet	D-SegNet
Precision$_M$	46.35%	90.98%	93.79%	**96.57%**
Recall$_M$	41.68%	84.87%	87.03%	**88.54%**
F1-score$_M$	43.89%	87.81%	90.28%	**92.38%**

procedure is repeated 10 times to make sure each fold has been used for testing once.

Fig. 5.10 displays a group of segmented images. Based on subjective observation, the two framed parts in each figure show different segmentation results. SegNet-Basic and SegNet are hard to segment the curve leaf, while D-SegNet gives a clear shape. It is expected that the proposed approach can capture many features through the dense block. Obviously, the D-SegNet achieves higher performance than the other algorithms. The SegNet-Basic [9] is a more lightweight network than SegNet, which comprises 4 pairs of encoder-decoders. As shown in Fig. 5.10(b), SegNet-Basic can take the coarse contour. Several pixels of leaf edge have been lost. In addition, SegNet as the state-of-the-art algorithm [10] is selected for comparison. It is a fully convolutional network with a pixel-level classification network that has 5 pairs of encoder-decoders. SegNet segments well the plant from the lab background compared with SegNet-Basic as seen in Fig. 5.10(c). However, when SegNet compares with the D-SegNet segmentation method in this chapter, it can be found that the SegNet segmentation method loses some details, as shown in Fig. 5.10(c). The SegNet method is sensitive to the close areas between plants, leaves overlapping and interference with light conditions. Thus, SegNet loses some features on leaves, and it cannot segment the special position of the plant in the original image. In this chapter, the D-Segnet method solved these problems, which can extract plant boundary contour from the complex background accurately.

Tables 5.4 and 5.6 list average quantitative segmentation results calculated over 10-fold cross-validation on the morning glory plant dataset. The numerical results denote precision, recall, F1-score and IoU in the four segmentation methods. There is a significant difference between segmentation based on edge detection and other deep learning algorithms. All mean metric values of deep learning algorithms are higher than that of the traditional image processing algorithm. The performance of D-SegNet outperforms that of SegNet-Basic and SegNet with respect to both visual and numerical results. Both SegNet-Basic and SegNet are encoder-decoder networks. The size of the feature map is going small after each convolution. The common problem is

Network	mean IoU
Based on edge detection	56.37%
SegNet-Basic	90.56%
SegNet	88.58%
D-SegNet	**90.64%**

Table 5.6: *Comparison of different segmentation methods in performance*

the convolution operation between layers may lose important features. The D-SegNet is designed to break this bottleneck. This is thanks to the concatenation between layers of the dense block in the D-SegNet. It makes the feature extracted from the encoder-decoder, which directly connects to each layer in the dense block. The D-Segnet will learn more features than the SegNet-Basic and SegNet, which makes D-SegNet lead the performance competition. The performance differences are obvious from Table 5.5. It shows the mean values of precision, recall and F1-score with four different algorithms on the ImageCLEF dataset. The D-SegNet gives accurate segmentation results on both datasets.

5.4.3 Additional Considerations

The accuracy of plant pixel identification in smartphone images is a fundamental concern. These images capture plants at different stages of growth, and leaves often appear in a non-uniform manner due to their discrete growth patterns. Each group of plants is photographed four times, resulting in multiple images for analysis. However, the inherent randomness in leaf arrangement can lead to variations in pixel labelling within the same group of plants. This complexity is further compounded when leaves overlap, as pixel counts for overlapping leaves are consolidated into a single entity. Understanding the relationship between plant pixels and weight prediction is important for effective automatic yield estimation.

Addressing this intricate issue requires a multifaceted approach, and two primary methods have been proposed. First, the challenge of labelling overlapped leaves pixel by pixel is acknowledged as a cumbersome task, prompting alternative strategies. Sec-

ond, the sample diversity within each group of plants is significantly increased. This increase in sample diversity stems from the multiple images captured from various plant positions within the same group. This strategy leverages the variation in leaf arrangement to enhance the robustness of the pixel labelling process.

The D-SegNet algorithm, as introduced in this research, excels at segmenting plants in their mature stages, surpassing the performance of state-of-the-art algorithms like SegNet [10]. However, it is crucial to note that the algorithm doesn't comprehensively capture plant growth at every stage. The timing of harvesting continues to rely on manual labour, as the morning glory plant's physical shape undergoes dynamic changes as it matures. Future research aims to delve deeper into the diverse morphological shapes that the morning glory plant exhibits. Understanding the morphological and physiological characteristics of the plant at various growth stages is pivotal for more precise and automated harvesting strategies.

This research is the reliance on smartphone images. The utilization of smartphone imagery underscores the potential of this technology in agriculture, given its widespread accessibility and affordability to a broad spectrum of farmers. The envisioned applications extend beyond yield prediction, as images captured by farmers can be seamlessly integrated into automated systems designed for predictive analytics. This initial study showcases the feasibility of processing noisy smartphone images and using them to predict yields with reasonably high accuracy. It constitutes the first step toward developing a comprehensive and automated system for yield prediction, one that holds immense promise for enhancing smart-farm planning and monitoring practices.

Furthermore, the D-SegNet algorithm exhibits versatility beyond yield prediction. Its capabilities extend to weed detection, disease identification and differentiation, and various related tasks within the realm of precision agriculture. The algorithm's adaptability and robust performance make it a valuable asset in tackling multifaceted challenges associated with crop management.

To ensure a robust and fair comparison, the evaluation metrics in this research are meticulously averaged over a 10-fold cross-validation process. The results and eval-

uation techniques employed in this study firmly substantiate the superiority of the proposed approaches. Not only does the D-SegNet algorithm excel in its segmentation capabilities, but it also shines in terms of reliability. These findings bolster the algorithm's potential as a pivotal tool in the field of agriculture, where precision and accuracy are paramount for effective yield estimation, disease control, and sustainable farming practices.

5.5 Summary

This chapter introduces a deep learning-based segmentation method known as D-SegNet, which holds significant promise for applications in automated yield prediction and precision agriculture. In contrast to traditional computer vision approaches, D-SegNet excels in providing pixel-level segmentation, offering a more detailed and accurate analysis of agricultural data.

One of the standout features of D-SegNet is its employment of a dense block structure, which significantly enhances feature propagation. This architecture surpasses existing deep learning algorithms like SegNet, establishing itself as a superior tool for segmentation tasks. The concatenation of feature maps within D-SegNet further improves the extraction of object information, making it particularly well-suited for handling sequential data. The network, once trained, exhibits remarkable proficiency in identifying intricate segmentation details, including very small and overlapping leaves within plant images. To validate the effectiveness of D-SegNet, comprehensive experiments were conducted, employing four metrics for performance evaluation and comparison against competing methods. The results were striking, with D-SegNet achieving precision, recall, and F1-score values of 0.9820, 0.9064, and 0.9426, respectively. Furthermore, the Intersection over Union (IoU) metric was calculated for 2,421 untrained plant images, yielding a remarkable score of 0.9056. These findings unequivocally establish D-SegNet as a highly accurate and efficient tool for plant segmentation, substantially outperforming prior methodologies.

The focus of this research is on the morning glory plant, a staple crop in the agriculture industry, and its potential applications for yield prediction. This research holds particular promise for small-scale and community-based farms, offering them a powerful tool to estimate their crop yields accurately. Accurate yield predictions enable the buyers of agricultural products to optimize their sourcing strategies, reduce costs, and better meet the demands of their customers. This, in turn, promotes a healthier and more cooperative relationship between farmers and buyers, fostering a mutually beneficial partnership in the agricultural sector.

Experiments show that D-SegNet has the potential to improve the field of precision agriculture and yield prediction. This advanced plant segmentation network could become an integral component of machine learning systems, enabling the precise estimation of crop yields from smartphone images captured in the field. Such technology has the potential to enhance the efficiency and profitability of agricultural operations, benefiting farmers, buyers, and the entire agricultural supply chain.

Chapter 6

Conclusions

6.1 Summary and Contributions

The thesis presents a comprehensive exploration of deep learning techniques applied to autonomous plant image segmentation and classification. This research domain has gained remarkable traction due to its distinct advantages when compared to conventional image processing methods for plant imagery. In contrast to the traditional coarse feature extraction, deep learning-based approaches revolutionize image segmentation and classification by harnessing the power of precision convolution calculations. These approaches excel in capturing an array of features, employing various methodologies. This innovative approach equips farmers and retailers with a valuable tool for seamlessly adapting their plant management and marketing strategies. Moreover, these deep learning models outshine their predecessors by delivering superior precision in plant image segmentation and classification. This marks a significant departure from the labor-intensive and time-consuming practices associated with traditional image processing and manual inspection. The results are not only faster but also more accurate in the realm of plant image analysis.

While the promising deep learning techniques for plant image segmentation and classification are advancing rapidly, several challenging problems remain to be ad-

dressed. The inherent similarity and complex backgrounds in plant images increase the difficulty of accurate segmentation and classification. Furthermore, enhancing prediction performance is a daunting task, owing to numerous sources of interference. Among these challenges, dealing with overlapped plants or imbalanced data poses one of the most formidable obstacles. Overlapping plants obscure the objects, blending features from different plants and making distinctions challenging. Additionally, distinguishing plants in an imbalanced dataset, which predominantly emphasizes learning features from a large amount of data while neglecting features with limited occurrences, is equally demanding. Furthermore, the fine-grained classification tasks in plant pathology present their own set of challenges, characterized by high intra-class variation and low inter-class differences.

To address the issues outlined above and elevate the performance of plant image segmentation and classification, this thesis introduces several frameworks rooted in deep learning methods. Although deep learning algorithms have achieved remarkable performance, this thesis aims to optimize the deep learning framework, focusing on refining the loss function, enhancing layer design, and improving learning strategies. These optimizations are summarized as follows:

In Chapter 3, a deep learning method with a novel loss function is proposed based on the fully convolutional network architecture specifically for addressing imbalanced datasets in semantic segmentation tasks. The cross dropout focal loss function systematically updates the weights assigned to each class based on the outcomes of dropout iterations. the proposed method dynamically adjusts the importance of dropout outputs from each class, effectively capturing the gradient of segmentation difficulty. This dynamic weighting approach ensures that small, subtle features are treated with the same significance as the more prominent, obvious features, leading to more comprehensive and accurate segmentation results. The framework performance is assessed on two distinct benchmark datasets. The results are promising, indicating an overall improvement of 2.5% in segmentation accuracy when applied to larger datasets.

In Chapter 4, a novel learning strategy of deep learning algorithm is developed

based on the squeeze and excitation (SE) network for the precise classification of fine-grained plant pathology. Fine-grained images of plant diseases pose a unique challenge due to the subtle symptoms they exhibit on the leaves, often defying accurate classification even by human experts. The holistic self-distillation imparts both soft and hard labels to guide the SE network in capturing the intricate nuances of these diseases. In this process, the teacher SE network imparts valuable feature knowledge, which is then transferred to the student SE network, along with crucial hard label information. This dual-source learning strategy equips the SE network with the ability to assimilate both label knowledge and feature knowledge, resulting in a robust classification framework. This powerful SE network can now distinguish between various plant diseases within the complex environmental context, moving beyond mere error diagnosis and significantly contributing to the overall classification accuracy. Consequently, experiments demonstrate that the proposed method surpasses the performance of other state-of-the-art methods, establishing its superiority in the realm of fine-grained plant pathology classification.

In Chapter 5, an innovative deep learning framework and a meticulously curated morning glory dataset take centre stage as the focal points of our research, addressing the challenging task of semantic segmentation. The dataset has been meticulously curated from original plant data, and meticulously captured using two distinct devices operating under identical environmental conditions, ensuring a comprehensive and reliable source of information. The corresponding labelled data is thoughtfully generated using a semi-supervised methodology, providing an essential foundation for our semantic segmentation approach. The D-SegNet framework embraces an encoder-decoder architecture enriched with dense blocks, significantly enhancing the propagation of essential features throughout the network. This design choice allows for robust and efficient information extraction, especially advantageous when dealing with sequential data or complex plant structures. A key innovation lies in the concatenation of feature maps, which augments the extraction of critical object-specific information, enhancing the model's ability to discriminate between different plant components and structures.

This unique feature makes our framework particularly well-suited for handling the intricate and intricate nuances within the plant world. The proposed framework to the test, subjecting it to rigorous evaluation on two distinct plant datasets. The results speak to its promise and potential, as it consistently demonstrates improved segmentation performance compared to existing methods. This not only emphasizes the effectiveness of our D-SegNet framework but also opens up possibilities for enhancing the understanding and analysis of plant images and beyond.

6.2 Future Work

In this section, some promising deep learning algorithms are able to deal with sparse data and at the same time estimate the uncertainties of the developed solutions. Different ways of fusion of the features from the images will be considered. Multimodal information fusion is another area of future work, especially visual and audio features in changeable dynamic environments.

- **Evaluating the uncertainty of the D-SegNet on the area of sparse datasets.** A sparse image dataset refers to an image dataset that contains many sparse regions. Only a small portion of pixels contain useful information, while other pixels are typically vacant or contain background. This kind of dataset commonly arises in image segmentation tasks. For instance, the COCO dataset [92] encompasses a substantial number of images for object detection and image segmentation, where objects usually manifest solely in a fraction of the image, while the rest constitutes the background. This results in sparsity since the majority of pixels lack object-related information. Handling such datasets necessitates specialized image segmentation and object detection algorithms to effectively extract information and address the sparse regions. The model may generate indistinct predictions for segmented objects, emphasizing the need for robust uncertainty assessment.

- **The holistic self-distillation can be used to classify tasks with distinct features,** especially those whose characteristics pose a challenge for a conventional model to differentiate. In theory, the holistic self-distillation learning strategy holds the potential to enhance the performance of any model. The proposed approach harnesses holistic insights directly from the framework itself and leverages self-teaching through labelling. It offers models the opportunity to leverage their own knowledge and self-learn, resulting in enhanced performance. This form of self-guided learning fosters a robust and self-sufficient model, capable of outperforming the initial student framework and achieving state-of-the-art results.

- **Integrating the Feature Pyramid Network (FPN) into the D-SegNet framework to attain multi-scale segmentation capabilities.** Multiscale image segmentation is a methodology geared towards enhancing the precision of object or region segmentation in images by simultaneously incorporating data from various scales or resolutions. This approach is particularly adept at addressing intricate scenarios in plant segmentation tasks, where objects exhibit varying sizes, shapes, and boundaries in different areas of the image. To facilitate this process, image features, including colour, texture, edges, and more, are meticulously extracted at each scale. These extracted features play a pivotal role in enabling algorithms to gain a deeper comprehension of the structural nuances of objects across diverse scales. Furthermore, the feature information extracted at different scales is amalgamated to generate the ultimate segmentation output. The Feature Pyramid Network (FPN) follows the principle of simultaneously integrating data from multiple scales, thereby enhancing the accuracy and resilience of segmentation. This adaptability equips the algorithm to effectively handle a multitude of scenes and objects with varying characteristics.

- **Incorporating visual-audio features into the existing framework of plant to enable multimodal feature fusion.** Multimodal fusion represents a potent strategy for building cross-media information retrieval systems. This technology empowers users to initiate a query using a plant image and seamlessly access associated information, encompassing both audio and textual data. It brings together the power of both visual and audio information to offer a more comprehensive and holistic understanding of plant-related content. Whether it is audio clips describing the plant's unique features or textual descriptions of its growth stages, this technology empowers users to access a complete and multi-dimensional perspective of the plant in question. It significantly encourages farmers to move beyond the constraints of traditional, empirically driven methods. Instead, it motivates them to delve deeper into the realms of professional agricultural knowledge, lead-

ing to more informed decisions when it comes to tasks like seed sowing, pesticide application, and plant care. Simultaneously, it equips agricultural students with a powerful tool to swiftly and accurately grasp the characteristics and contextual background of plants within the real-world farming environment.

Bibliography

[1] M. Abbasgholipour, M. Omid, A. Keyhani, and S. S. Mohtasebi. Color image segmentation with genetic algorithm in a raisin sorting system based on machine vision in variable conditions. *Expert Systems with Applications*, 38(4):3671–3678, 2011.

[2] H. S. Abdullahi, R. Sheriff, and F. Mahieddine. Convolution neural network in precision agriculture for plant image recognition and classification. In *Proceedings of the Seventh International Conference on Innovative Computing Technology (INTECH)*, volume 10, pages 256–272. Ieee, 2017.

[3] A. Aggelopoulou, D. Bochtis, S. Fountas, K. C. Swain, T. Gemtos, and G. Nanos. Yield prediction in apple orchards based on image processing. *Precision Agriculture*, 12(3):448–456, 2011.

[4] A. Ahmad, D. Saraswat, and A. El Gamal. A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. *Smart Agricultural Technology*, 3:100083, 2023.

[5] S. Aich and I. Stavness. Leaf counting with deep convolutional and deconvolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2080–2089, 2017.

[6] S. Angelina, L. P. Suresh, and S. K. Veni. Image segmentation based on genetic algorithm for region growth and region merging. In *Proceedings of International*

Conference on Computing, Electronics and Electrical Technologies (ICCEET), pages 970–974. IEEE, 2012.

[7] M. H. Anisi, G. Abdul-Salaam, and A. H. Abdullah. A survey of wireless sensor network approaches and their energy consumption for monitoring farm fields in precision agriculture. *Precision Agriculture*, 16(2):216–238, 2015.

[8] S. V. Anstalt. Food and agriculture organization of the united nations. 2013.

[9] V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.

[10] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[11] M. Berman, A. R. Triki, and M. B. Blaschko. The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018.

[12] K. Bhargavi and S. Jyothi. A survey on threshold based segmentation technique in image processing. *International Journal of Innovative Research and Development*, 3(12):234–239, 2014.

[13] P. Bosilj, E. Aptoula, T. Duckett, and G. Cielniak. Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture. *Journal of Field Robotics*, 37(1):7–19, 2020.

[14] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'*, pages 177–186. Springer, 2010.

[15] G. J. Brindha and E. Gopi. An hierarchical approach for automatic segmentation

of leaf images with similar background using kernel smoothing based Gaussian process regression. *Ecological Informatics*, 62:101323, 2021.

[16] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541, 2006.

[17] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in Neural Information Processing Systems*, 32, 2019.

[18] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408:189–215, 2020.

[19] H.-C. Chen, A. M. Widodo, A. Wisnujati, M. Rahaman, J. C.-W. Lin, L. Chen, and C.-E. Weng. Alexnet convolutional neural network for disease detection and classification of tomato leaf. *Electronics*, 11(6):951, 2022.

[20] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[21] P. Chen, S. Liu, H. Zhao, and J. Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5008–5017, 2021.

[22] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[23] K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen. Fuzzy c-means clus-

tering with spatial information for image segmentation. *Computerized Medical Imaging and Graphics*, 30(1):9–15, 2006.

[24] M. Cilimkovic. Neural networks and back propagation algorithm. *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin*, 15(1), 2015.

[25] D. Ciresan, A. Giusti, L. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in Neural Information Processing Systems*, 25, 2012.

[26] G. B. Coleman and H. C. Andrews. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979.

[27] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[28] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.

[29] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in Neural Information Processing Systems*, 29, 2016.

[30] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar. A survey of deep learning and its applications: a new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 27(4):1071–1092, 2020.

[31] P. Deepa and S. N. Geethalakshmi. Improved watershed segmentation for apple fruit grading. In *Proceedings of International Conference on Process Automation, Control and Computing*, pages 1–5, 2011.

[32] R. Deng, M. Tao, H. Xing, X. Yang, C. Liu, K. Liao, and L. Qi. Automatic diagnosis of rice diseases using deep learning. *Frontiers in Plant Science*, 12:701038, 2021.

[33] N. Dhanachandra, K. Manglem, and Y. J. Chanu. Image segmentation using k-means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54:764–771, 2015.

[34] A. Dorward and E. Chirwa. A review of methods for estimating yield and production impacts. Technical report, Centre for Development, Environment and Policy, SOAS, University of London, UK, 2010.

[35] S. R. Dubey, P. Dixit, N. Singh, and J. P. Gupta. Infected fruit part detection using k-means clustering segmentation technique. *International Journal of Interactive Multimedia and Artificial Intelligence ...*, 2013.

[36] M. Everingham, S. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[37] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[38] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[39] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *Proceedings of the IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition, pages 2241–2248. Ieee, 2010.

[40] C. Feng, D. Zhao, and M. Huang. Image segmentation and bias correction using local inhomogeneous intensity clustering (linc): a region-based level set method. *Neurocomputing*, 219:107–129, 2017.

[41] L. Fu, E. Tola, A. Al-Mallahi, R. Li, and Y. Cui. A novel image processing algorithm to separate linearly clustered kiwifruits. *Biosystems Engineering*, 183:184–195, 2019.

[42] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

[43] W. Gao, X. Zhang, L. Yang, and H. Liu. An improved Sobel edge detection. In *Proceedings of the 3rd International Conference on Computer Science and Information Technology*, volume 5, pages 67–71. IEEE, 2010.

[44] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[45] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, 70:41–65, 2018.

[46] R. Gebbers and V. I. Adamchuk. Precision agriculture and food security. *Science*, 327(5967):828–831, 2010.

[47] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[48] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional

networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, 2015.

[49] H. Goëau, P. Bonnet, and A. Joly. Overview of plantclef 2021: cross-domain plant identification. In *Working Notes of CLEF 2021-Conference and Labs of the Evaluation Forum*, volume 2936, pages 1422–1436, 2021.

[50] J. Gou, B. Yu, S. J. Maybank, and D. Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

[51] C. Goutte and E. Gaussier. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In *Proceedings of Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23.*, pages 345–359. Springer, 2005.

[52] M. Grandini, E. Bagli, and G. Visani. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*, 2020.

[53] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.

[54] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[55] J. M. Guerrero, G. Pajares, M. Montalvo, J. Romeo, and M. Guijarro. Support vector machines for crop/weeds identification in maize fields. *Expert Systems with Applications*, 39(12):11149–11155, 2012.

[56] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer. Knn model-based approach in classification. In *Proceedings of On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Confer-*

ences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7., pages 986–996. Springer, 2003.

[57] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew. A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, 7(2):87–93, 2018.

[58] S. Hahn and H. Choi. Self-knowledge distillation in natural language processing. *arXiv preprint arXiv:1908.01851*, 2019.

[59] E. Hamuda, M. Glavin, and E. Jones. A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture*, 125:184–199, 2016.

[60] S. Hao, Y. Zhou, and Y. Guo. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:302–321, 2020.

[61] S. M. Hassan, K. Amitab, M. Jasinski, Z. Leonowicz, E. Jasinska, T. Novak, and A. K. Maji. A survey on different plant diseases detection using machine learning techniques. *Electronics*, 11(17):2641, 2022.

[62] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.

[63] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[64] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[65] E. Hossain, M. F. Hossain, and M. A. Rahaman. A color and texture based approach for the detection and classification of plant leaf disease using knn clas-

sifier. In *Proceedings of International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1–6. IEEE, 2019.

[66] Y. Hou, Z. Ma, C. Liu, and C. C. Loy. Learning lightweight lane detection CNNs by self attention distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1013–1021, 2019.

[67] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[68] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[69] J. Huang and C. X. Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005.

[70] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of International Conference on Machine Learning*, pages 448–456. PMLR, 2015.

[71] Y. Itzhaky, G. Farjon, F. Khoroshevsky, A. Shpigler, and A. Bar-Hillel. Leaf counting: Multiple scale regression and detection using deep cnns. In *Proceedings of BMVC-British Machine Vision Conference*, volume 328. Newcastle, 2018.

[72] S. Jadon. A survey of loss functions for semantic segmentation. In *Proceedings of the IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.

[73] P. Jasitha, M. Dileep, and M. Divya. Venation based plant leaves classification using googlenet and vgg. In *Proceedings of the 4th International Conference*

on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), pages 715–719. IEEE, 2019.

[74] G. Jeon. Color image enhancement by histogram equalization in heterogeneous color space. *Int. J. Multimedia Ubiquitous Eng*, 9(7):309–318, 2014.

[75] Y. Jiang, L. Chen, H. Zhang, and X. Xiao. Breast cancer histopathological image classification using convolutional neural networks with small se-resnet module. *PloS One*, 14(3):e0214587, 2019.

[76] L. Jiao and J. Zhao. A survey on the new generation of deep learning in image processing. *IEEE Access*, 7:172231–172263, 2019.

[77] J. M. Johnson and T. M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[78] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

[79] S. Kanokwan and C. Khompatraporn. Morning glory plant datset. Website, April 2021. `https://github.com/S-KANOKWAN?tab=repositories`.

[80] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002.

[81] C. Kittipong and C. Khompatraporn. Morning glory plant datsets. Website, August 2021. `https://github.com/CH-KITTIPONG?tab=repositories`.

[82] S. Kolhar and J. Jagtap. Convolutional neural network based encoder-decoder architectures for semantic segmentation of plants. *Ecological Informatics*, 64:101373, 2021.

[83] D. Komura and S. Ishikawa. Machine learning methods for histopathological image analysis. *Computational and Structural Biotechnology Journal*, 16:34–42, 2018.

[84] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[85] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[86] V. Kumar, H. Arora, J. Sisodia, et al. Resnet-based approach for detection and classification of plant leaf diseases. In *Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 495–502. IEEE, 2020.

[87] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782, 2017.

[88] F. Lateef and Y. Ruichek. Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, 338:321–348, 2019.

[89] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[90] A. Likas, N. Vlassis, and J. J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451–461, 2003.

[91] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[92] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proceedings of*

Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12., pages 740–755. Springer, 2014.

[93] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457, 2015.

[94] X. Liu, T. Xia, J. Wang, Y. Yang, F. Zhou, and Y. Lin. Fully convolutional attention networks for fine-grained recognition. *arXiv preprint arXiv:1603.06765*, 2016.

[95] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019.

[96] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[97] D. Lu and Q. Weng. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5):823–870, 2007.

[98] H. Lu, Q. Gao, X. Zhang, and W. Xia. A multi-view clustering framework via integrating k-means and graph-cut. *Neurocomputing*, 501:609–617, 2022.

[99] J. Lu, L. Tan, and H. Jiang. Review on convolutional neural network (cnn) applied to plant leaf disease classification. *Agriculture*, 11(8):707, 2021.

[100] W. Luo, Y. Li, R. Urtasun, and R. Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 29, 2016.

[101] L. Ma, M. Wang, J. Dong, and K. Peng. A novel distributed detection framework

for quality-related faults in industrial plant-wide processes. *Neurocomputing*, 492:126–136, 2022.

[102] X. Ma, X. Deng, L. Qi, Y. Jiang, H. Li, Y. Wang, and X. Xing. Fully convolutional network for rice seedling and weed image segmentation at the seedling stage in paddy fields. *PloS one*, 14(4):e0215676, 2019.

[103] D. G. MacArthur, S. Balasubramanian, A. Frankish, N. Huang, J. Morris, K. Walter, L. Jostins, L. Habegger, J. K. Pickrell, S. B. Montgomery, et al. A systematic survey of loss-of-function variants in human protein-coding genes. *Science*, 335(6070):823–828, 2012.

[104] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, 1990.

[105] G. E. Meyer, J. C. Neto, D. D. Jones, and T. W. Hindman. Intensified fuzzy clusters for classifying plant, soil, and residue regions of interest from color images. *Computers and Electronics in Agriculture*, 42(3):161–180, 2004.

[106] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542, 2022.

[107] J. Muschelli III. Roc and auc with a binary predictor: a potentially misleading metric. *Journal of Classification*, 37(3):696–708, 2020.

[108] J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, and A. Valenzuela. A review of convolutional neural network applied to fruit image processing. *Applied Sciences*, 10(10):3443, 2020.

[109] K. Nemoto, R. Hamaguchi, T. Imaizumi, and S. Hikosaka. Classification of rare building change using cnn with multi-class focal loss. In *Proceedings of IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 4663–4666. IEEE, 2018.

[110] L. C. Ngugi, M. Abelwahab, and M. Abo-Zahhad. Recent advances in image processing techniques for automated leaf pest and disease recognition–a review. *Information Processing in Agriculture*, 8(1):27–51, 2021.

[111] A. Nigam, S. Garg, A. Agrawal, and P. Agrawal. Crop yield prediction using machine learning algorithms. In *Proceedings of the Fifth International Conference on Image Information Processing (ICIIP)*, pages 125–130. IEEE, 2019.

[112] M.-E. Nilsback. *An automatic visual flora-segmentation and classification of flower images*. PhD thesis, Oxford University, 2009.

[113] X. Niu, M. Wang, X. Chen, S. Guo, H. Zhang, and D. He. Image segmentation algorithm for disease detection of wheat leaves. In *Proceedings of the International Conference on Advanced Mechatronic Systems*, pages 270–273, 2014.

[114] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[115] F. R. F. Padao and E. A. Maravillas. Using naïve bayesian method for plant leaf classification based on shape and texture features. In *Proceedings of 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, pages 1–5. IEEE, 2015.

[116] G. Pajares, J. J. Ruz, and J. M. de la Cruz. Performance analysis of homomorphic systems for image change detection. In *Proceedings of Pattern Recognition and Image Analysis: Second Iberian Conference, IbPRIA 2005, Estoril, Portugal, June 7-9*, pages 563–570. Springer, 2005.

[117] M. Palmer, H. T. Dang, and C. Fellbaum. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(2):137–163, 2007.

[118] H. N. Patel, R. K. Jain, and M. V. Joshi. Automatic segmentation and yield measurement of fruit using shape analysis. *International Journal of Computer Applications*, 45(7):19–24, 2012.

[119] A. Peil, V. G. Bus, K. Geider, K. Richter, H. Flachowsky, and M.-V. Hanke. Improvement of fire blight resistance in apple and pear. *Int J Plant Breed*, 3(1):1–27, 2009.

[120] A. Perez, F. Lopez, J. Benlloch, and S. Christensen. Colour and shape analysis techniques for weed detection in cereal fields. *Computers and Electronics in Agriculture*, 25(3):197–212, 2000.

[121] M. Pesaresi and J. A. Benediktsson. A new approach for the morphological segmentation of high-resolution satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 39(2):309–320, 2001.

[122] V. Pihur, S. Datta, and S. Datta. Weighted rank aggregation of cluster validation measures: a Monte Carlo cross-entropy approach. *Bioinformatics*, 23(13):1607–1615, 2007.

[123] T. Pointet. The united nations world water development report 2022 on groundwater, a synthesis. *LHB*, 108(1):2090867, 2022.

[124] D. M. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[125] E. Prasetyo, R. D. Adityo, N. Suciati, and C. Fatichah. Mango leaf image segmentation on hsv and ycbcr color spaces using otsu thresholding. In *Proceedings of 3rd International Conference on Science and Technology-Computer (ICST)*, pages 99–103. IEEE, 2017.

[126] C. A. Priya, T. Balasaravanan, and A. S. Thanamani. An efficient leaf recognition algorithm for plant classification using support vector machine. In *Proceedings*

of International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012), pages 428–432. IEEE, 2012.

[127] E. Puyol-Antón, B. Ruijsink, S. K. Piechnik, S. Neubauer, S. E. Petersen, R. Razavi, and A. P. King. Fairness in cardiac mr image analysis: an investigation of bias due to data imbalance in deep learning based segmentation. In *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 413–423. Springer, 2021.

[128] H. Qin, R. Gong, X. Liu, M. Shen, Z. Wei, F. Yu, and J. Song. Forward and backward information retention for accurate binary neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2250–2259, 2020.

[129] M. A. Rahman and Y. Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *Proceedings of International Symposium on Visual Computing*, pages 234–244. Springer, 2016.

[130] P. Refaeilzadeh, L. Tang, and H. Liu. Cross-validation. *Encyclopedia of Database Systems*, 5:532–538, 2009.

[131] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28:91–99, 2015.

[132] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019.

[133] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In *Proceedings of Computer Vision–ECCV : 14th European Conference, Amsterdam, The Netherlands, October 11-14.*, pages 312–329. Springer, 2016.

[134] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[135] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015.

[136] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[137] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

[138] M. M. Saritas and A. Yasar. Performance analysis of ann and naive bayes classification algorithm for data classification. *International Journal of Intelligent Systems and Applications in Engineering*, 7(2):88–91, 2019.

[139] M. A. Sartin, A. C. Da Silva, and C. Kappes. Image segmentation with artificial neural network for nutrient deficiency in cotton crop. *Journal of Computer Science*, pages 1084–1093, 2014.

[140] A. Savla, N. Israni, P. Dhawan, A. Mandholia, H. Bhadada, and S. Bhardwaj. Survey of classification algorithms for formulating yield prediction accuracy in precision agriculture. In *Proceedings of the International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pages 1–7. IEEE, 2015.

[141] L. Saxena and L. Armstrong. A survey of image processing techniques for agriculture. In *Proceedings of Asian Federation for Information Technology in Agri-*

culture, pages 401–413. Australian Society of Information and Communication Technologies in Agriculture, 2014.

[142] H. Scharr, M. Minervini, A. P. French, C. Klukas, D. M. Kramer, X. Liu, I. Luengo, J.-M. Pape, G. Polder, D. Vukadinovic, et al. Leaf segmentation in plant phenotyping: a collation study. *Machine Vision and Applications*, 27:585–606, 2016.

[143] U. Shafi, R. Mumtaz, J. García-Nieto, S. A. Hassan, S. A. R. Zaidi, and N. Iqbal. Precision agriculture techniques and practices: From considerations to applications. *Sensors*, 19(17):3796, 2019.

[144] Y. Shen, L. Xu, Y. Yang, Y. Li, and Y. Guo. Self-distillation from the last mini-batch for consistency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11943–11952, 2022.

[145] J. Sherrah. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. *arXiv preprint arXiv:1606.02585*, 2016.

[146] C. S. Sherrington. Observations on the scratch-reflex in the spinal dog. *The Journal of Physiology*, 34(1-2):1, 1906.

[147] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[148] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016.

[149] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[150] A. Singh, B. Ganapathysubramanian, A. K. Singh, and S. Sarkar. Machine learning for high-throughput stress phenotyping in plants. *Trends in Plant Science*, 21(2):110–124, 2016.

[151] A. Singh and A. Purohit. A survey on methods for solving data imbalance problem for classification. *International Journal of Computer Applications*, 127(15):37–41, 2015.

[152] D. C. Slaughter and R. C. Harrell. Discriminating fruit for robotic harvest using color in natural outdoor scenes. *Transactions of the ASAE*, 32(2):757–0763, 1989.

[153] J. Snedeker, L. Gleitman, et al. Why it is hard to label our concepts. *Weaving a Lexicon*, 257294, Cambridge, MA: MIT Press, 2004.

[154] S. G. Sodjinou, V. Mohammadi, A. T. S. Mahama, and P. Gouton. A deep semantic segmentation-based algorithm to segment crops and weeds in agronomic color images. *Information Processing in Agriculture*, 2021.

[155] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[156] J. Su, S. Anderson, and L. S. Mihaylova. A deep learning method with cross dropout focal loss function for imbalanced semantic segmentation. In *Proceedings of Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6. IEEE, 2022.

[157] D. Sun, A. Yao, A. Zhou, and H. Zhao. Deeply-supervised knowledge synergy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6997–7006, 2019.

[158] T. B. Sutton, H. S. Aldwinckle, A. M. Agnello, and J. F. Walgenbach. *Compendium of apple and pear diseases and pests*. Am Phytopath Society, 2014.

[159] A. Swaminathan, C. Varun, S. Kalaivani, et al. Multiple plant leaf disease classification using densenet-121 architecture. *Int. J. Electr. Eng. Technol*, 12:38–57, 2021.

[160] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[161] S. A. Taghanaki, Y. Zheng, S. K. Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu, and G. Hamarneh. Combo loss: Handling input and output imbalance in multi-organ segmentation. *Computerized Medical Imaging and Graphics*, 75:24–33, 2019.

[162] K. Thangadurai and K. Padmavathi. Computer vision image enhancement for plant leaves disease detection. In *Proceedings of World Congress on Computing and Communication Technologies*, pages 173–175. IEEE, 2014.

[163] R. Thapa, K. Zhang, N. Snavely, S. Belongie, and A. Khan. The plant pathology challenge 2020 data set to classify foliar disease of apples. *Applications in Plant Sciences*, 8(9):e11390, 2020.

[164] R. Thendral, A. Suhasini, and N. Senthil. A comparative analysis of edge and color based segmentation for orange fruit recognition. In *Proceedings of International Conference on Communication and Signal Processing*, pages 463–466. IEEE, 2014.

[165] J. Tian, N. C. Mithun, Z. Seymour, H.-p. Chiu, and Z. Kira. Recall loss for imbalanced image classification and semantic segmentation. In *Proceedings of the International Conference on Learning Representations*. ICLR, 2021.

[166] S. A. Tsaftaris, M. Minervini, and H. Scharr. Machine learning for plant phenotyping needs image processing. *Trends in Plant Science*, 21(12):989–991, 2016.

[167] A. C. Tyagi. Towards a second green revolution. *Irrigation and Drainage*, 65(4):388–389, 2016.

[168] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders. Segmentation as selective search for object recognition. In *Proceedings of International Conference on Computer Vision*, pages 1879–1886. IEEE, 2011.

[169] E. Van Henten, B. v. Van Tuijl, J. Hemming, J. Kornet, J. Bontsema, and E. Van Os. Field test of an autonomous cucumber picking robot. *Biosystems engineering*, 86(3):305–313, 2003.

[170] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).*, volume 1, pages I–I. IEEE, 2001.

[171] S. Visa, B. Ramsay, A. L. Ralescu, and E. Van Der Knaap. Confusion matrix-based feature selection. In *Proceedings of the 22nd Midwest Artificial Intelligence and Cognitive Science (MAICS)*, volume 710, pages 120–127, 2011.

[172] L. Wang and K.-J. Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:3048–3068, 2021.

[173] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1451–1460. Ieee, 2018.

[174] Y. Wang and Z. Wang. A survey of recent work on fine-grained image classification techniques. *Journal of Visual Communication and Image Representation*, 59:210–214, 2019.

[175] Z. Wang, K. Wang, F. Yang, S. Pan, and Y. Han. Image segmentation of overlapping leaves based on chan–vese model and sobel operator. *Information Processing in Agriculture*, 5(1):1–10, 2018.

[176] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1395–1403, 2015.

[177] R. Yacouby and D. Axman. Probabilistic extension of precision, recall, and F1 score for more thorough evaluation of classification models. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 79–91, 2020.

[178] C. Yang, L. Xie, C. Su, and A. L. Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2859–2868, 2019.

[179] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval*, pages 197–206, 2007.

[180] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, and L. Wang. Learning to navigate for fine-grained classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 420–435, 2018.

[181] M. Yi-de, L. Qing, and Q. Zhi-Bai. Automated image segmentation using improved pcnn model based on cross-entropy. In *Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Processing.*, pages 743–746. IEEE, 2004.

[182] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911, 2020.

[183] S. Yuheng and Y. Hao. Image segmentation algorithms overview. *arXiv preprint arXiv:1707.02051*, 2017.

[184] S. Yun, J. Park, K. Lee, and J. Shin. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13876–13885, 2020.

[185] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[186] Q. Zeng, Y. Miao, C. Liu, and S. Wang. Algorithm based on marker-controlled watershed transform for overlapping plant fruit segmentation. *Optical Engineering*, 48(2):027201–027201, 2009.

[187] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722, 2019.

[188] Q. Zhang, Y. Liu, C. Gong, Y. Chen, and H. Yu. Applications of deep learning for dense scenes analysis in agriculture: A review. *Sensors*, 20(5):1520, 2020.

[189] Z. Zhang and M. Sabuncu. Self-distillation as instance-specific label smoothing. *Advances in Neural Information Processing Systems*, 33:2184–2195, 2020.

[190] Y.-Y. Zheng, J.-L. Kong, X.-B. Jin, X.-Y. Wang, T.-L. Su, and M. Zuo. Cropdeep: The crop vision dataset for deep-learning-based classification and detection in precision agriculture. *Sensors*, 19(5):1058, 2019.

[191] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen. BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9719–9728, 2020.

[192] X. Zhu, S. Gong, et al. Knowledge distillation by on-the-fly native ensemble. *Advances in Neural Information Processing Systems*, 31:7528–7538, 2018.

Appendices

Appendix A

Forward Propagation

This appendix presents a comprehensive description of the forward and backward propagation equations for VGG-16 network, a component integral to the design of D-SegNet presented in Chapter 5.

VGG-16 has a total of 13 convolutional layers, 5 pooling layers and 3 fully connected layers. The first two layers of fully connected networks use dropout and L2 regularization to prevent overfitting, and use batch gradient descent and Momentum to use cross entropy as Target loss for training optimization.

Given the number of network nodes (convolution kernels) in layer l as n^l, the kernel between the layer l and $l-1$ as $\mathrm{k}^l_{\mathrm{p,q}}$, b^1_p denotes the bias of the p node at layer l. The weight of fully connected network at layer l is W^1, z^1 presents the forward input without through activity function at layer l, while a^1 represents the forward input through activity function at layer l.

The convolutional operates at layer l in math following:

$$z^l_p(i,j) = \sum_{q=1}^{n^{l-1}} \sum_{\substack{u=-1 \\ a^l_p(i,j)}}^{l} \sum_{v=-1}^{l} a^{l-1}_q(i-u, j-v) k^l_{p,q}(u,v) + b^l_p, \tag{A.1}$$

$$a^l_p(i,j) = \mathrm{ReLU}\left(z^l_p(i,j)\right).$$

The max pooling function is

$$\mathrm{z_p^l(i,j) = \max\left(a_p^{l-1}(2i-u, 2j-v)\right)\, u, v \in \{0,1\}.} \tag{A.2}$$

A feature map of size $7 \times 7 \times 512$ is obtained after 18 times the above operation, which needs to be converted into a 25,088-dimensional vector as the input of the fully connected layer. This output is a_{18}:

$$\mathrm{a^{18} = F\left(\left\{z_p^{18}\right\}_{p=1,2,\cdots,512}\right).} \tag{A.3}$$

The front two layers of fully connected network adopt dropout, set as d. The connection of layer l is indicated r^l, which follows Bernoulli distribution [128]:

$$\mathrm{r^l \sim\ Bernoulli\ (d).} \tag{A.4}$$

Thus, the process of forward propagation can be expressed as

$$\begin{aligned} \tilde{a}^l &= r^l \odot a^l, \\ z^{l+1} &= W^{l+1}\tilde{a}^l + b^{l+1}, \\ a^{l+1} &= \mathrm{ReLU}\left(z^{l+1}\right). \end{aligned} \tag{A.5}$$

The activity function of the output layer is softmax:

$$a_i^L = \mathrm{softmax}\left(z_i^L\right) = \frac{e^{z_i^L}}{\sum_{k=1}^{n^L} e^{z_k^L}}. \tag{A.6}$$

The loss function picks cross entropy:

$$\mathrm{L = -\sum_{i=1}^{n^L} y_i \log a_i^L.} \tag{A.7}$$

Appendix B

Back Propagation

The δ^1 error of the layer l indicates the gradient of loss for the forward input of layer l as $\frac{\partial \mathrm{L}}{\partial \mathrm{z}^l}$. The partial derivative of softmax is formulated as:

when $i = j$, it is

$$\frac{\partial}{\partial z_j}\left(\frac{e^{z_j}}{\sum_{k=1}^{n} e^{z_k}}\right) = \frac{e^{z_j}\sum_{k=1}^{n} e^{z_k} - \left(e^{z_j}\right)^2}{\left(\sum_{k=1}^{n} e^{z_k}\right)^2} = a_j\left(1 - a_j\right); \tag{B.1}$$

when $i \neq j$, it is

$$\frac{\partial}{\partial z_j}\left(\frac{e^{z_i}}{\sum_{k=1}^{n} e^{z_k}}\right) = \frac{-e^{z_i}e^{z_j}}{\left(\sum_{k=1}^{n} e^{z_k}\right)^2} = -a_i a_j. \tag{B.2}$$

The error of the output layer at node j is

$$
\begin{aligned}
\delta_j^L &= \frac{\partial L}{\partial z_j^L} \\
&= \sum_{i=1}^{n^L} \frac{\partial L}{\partial a_i^L} \frac{\partial a_i^L}{\partial z_j^L} \\
&= \frac{\partial L}{\partial a_j^L} \frac{\partial a_j^L}{\partial z_j^L} + \sum_{i \neq j} \frac{\partial L}{\partial a_i^L} \frac{\partial a_i^L}{\partial z_j^L} \\
&= -\frac{y_j}{a_j^L} a_j^L \left(1 - a_j^L\right) + \sum_{i \neq j} -\frac{y_i}{a_i^L} \left(-a_i^L a_j^L\right) \\
&= -y_j \left(1 - a_j^L\right) + a_j^L \sum_{i \neq j} y_i \\
&= a_j^L - y_j.
\end{aligned}
\tag{B.3}
$$

Thus, the error of the back propagation at layer$l$ is

$$
\delta^l = \left(\mathrm{W}^{l+1}\right)^{\mathrm{T}} \delta^{l+1} \odot \mathrm{r}^1 \odot \mathrm{ReLU}\left(\mathrm{z}^l\right)'. \tag{B.4}
$$

The error of back propagation from the fully connected layer to the pooling layer is

$$
\delta^{18} = \mathrm{F}^{-1}\left(\left(\mathrm{W}^{19}\right)^{\mathrm{T}} \delta^{19}\right). \tag{B.5}
$$