

DEVELOPMENT OF DEEP NEURAL NETWORKS FOR OBJECT DETECTION SYSTEMS

MR. RUT YATIGUL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ENGINEERING IN
PRODUCTION ENGINEERING
DEPARTMENT OF PRODUCTION ENGINEERING
GRADUATE COLLEGE
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK
ACADEMIC YEAR 2025
COPYRIGHT OF KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

DEVELOPMENT OF DEEP NEURAL NETWORKS FOR OBJECT DETECTION SYSTEMS

MR. RUT YATIGUL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ENGINEERING IN
PRODUCTION ENGINEERING
DEPARTMENT OF PRODUCTION ENGINEERING
GRADUATE COLLEGE
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK
ACADEMIC YEAR 2025
COPYRIGHT OF KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK



Thesis Certificate

The Graduate College, King Mongkut's University of Technology North Bangkok

Title Development of Deep Neural Networks for Object Detection Systems

By Mr. Rut Yatigul

Accepted by the Graduate College, King Mongkut's University of Technology North Bangkok in Partial Fulfillment of the Requirements for the Master of Engineering in Production Engineering

..... Dean

(Professor Dr. Saowanit Sukparungsee)

Thesis Examination Committee

..... Chairman

(Dr. Theerasak Chanwimalueang)

..... Advisor

(Assistant Professor Dr. Wisanu Jitviriya)

..... Co-Advisor

(Assistant Professor Dr. Aran Blattler)

..... Committee

(Assistant Professor Dr. Teppakorn Sittiwanchai)

Name : Mr. Rut Yatigul
Thesis Title : Development of Deep Neural Networks for Object Detection Systems
Major Field : Production Engineering
King Mongkut's University of Technology North Bangkok
Thesis Advisor : : Assistant Professor Dr. Wisanu Jitviriya
Co-Advisor : Assistant Professor Dr. Aran Blattler
Academic Year : 2025

ABSTRACT

This research investigates the effect of adaptive pooling within CNNs and attention mechanisms, specifically the Squeeze-and-Excitation (SE) network. The experiment was divided into two stages. First, SE networks were modified by replacing their original pooling mechanism with the proposed fuzzy pooling approach. Second, the same modification was applied to the main pooling layer in CNNs. The fuzzy pooling mechanism controls a weight function that determines the fusion ratio between two pooling methods using a Laplacian feature map analysis system. The approach combines min pooling and max pooling to enhance tomato ripeness classification accuracy. The method was evaluated using a tomato ripeness dataset containing five ripeness classes: 0%, 30%, 50%, 70%, and 100%. A fixed-weight mixed pooling and traditional deterministic pooling method was also included for comparative analysis. Results demonstrate that fuzzy pooling outperformed traditional methods with zero to minimal increase in computational consumption, highlighting the effectiveness of integrating logic-based adaptive pooling into CNNs and SE networks. The proposed model achieves a balance between high performance and computational efficiency. These findings contribute to the field of image classification and provide valuable insights for improving object recognition and classification tasks.

(Total 63 Pages)

Keywords : Image Classification, Attention Mechanism, Fuzzy Pooling, Mixed Pooling, Laplacian Filter

_____ Advisor

ACKNOWLEDGEMENTS

Deep gratitude is expressed for the assistance, knowledge, care, attention, guidance, and consultation provided in various aspects, without which the completion of this thesis would not have been successful. Sincere appreciation is extended to Asst. Prof. Dr. Wisanu Jitviriya, Asst. Prof. Dr. Aran Blattler, and Asst. Prof. Dr. Teppakorn Sittiwanchai, by whom vital knowledge and guidance were imparted, and by whom this research was reviewed and corrected until its completion. The Department of Production and Robotics Engineering and the Graduate College at King Mongkut's University of Technology North Bangkok, along with all personnel by whom knowledge was shared and instrument usage was facilitated, are also gratefully acknowledged.

Finally, profound thanks are offered to parents, from whom continuous support and encouragement have always been received. Gratitude is also directed toward all teachers by whom training and knowledge were provided from the first year until graduation, and toward friends by whom the completion of this thesis was continually encouraged and supported.

Rut Yatigul

TABLE OF CONTENTS

	Page
Abstract	iv
Acknowledgements	v
Table of Contents	vi
List of Tables	viii
List of Figures	ix
List of Abbreviations	x
Chapter 1 Introduction	1
1.1 Background and Problem Statement	1
1.2 Objective	3
1.3 Scope	4
1.4 Contribution	4
Chapter 2 Literature Review	5
2.1 Tomato Ripeness Classification: Current technique	5
2.2 Convolutional Neural Networks (CNN)	6
2.3 Squeeze-and-Excitation Network	11
2.4 Research Gaps and Opportunities	16

TABLE OF CONTENTS (Continuous)

	Page
Chapter 3 Methodology	17
3.1 System Overview	17
3.2 Tomato ripeness dataset	18
3.3 Extended Squeeze and Excitation Network (ESEnet)	19
3.4 Mixed pooling	21
3.5 Laplacian filter	22
3.6 Fuzzy Pooling	25
3.7 Extended Darknet53 and Extended ResNet34	30
Chapter 4 Experimental Result	32
4.1 Stage 1: ESEnet	32
4.2 Stage 2: Extended Net	38
4.3 Implementation	44
Chapter 5 Conclusion	47
5.1 Conclusion of Research	47
5.2 Future Work and Limitations	47
Reference	49
Vita	54

LIST OF TABLES

Table		Page
3-1	Structure of Extended Darknet53	30
3-2	Structure of Extended ResNet34	31
4-3	Experimental result of ESEnet with Darknet53 on test set	33
4-4	Experimental result of ESEnet with ResNet34	34
4-5	Experimental result of Extended Darknet53	38
4-6	Experimental result of Extended ResNet34	40

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

LIST OF FIGURES

Figure		Page
1-1	Various ripeness of tomato fruit on tomato tree	1
2-2	Basic structure of Convolutional neural network	6
2-3	Example of Max-Pooling operation	9
2-4	Example of Average-Pooling operation	10
2-5	3 × 3 Avg-TopK pooling	11
2-6	A Squeeze-and-Excitation block	11
2-7	Hybrid model architecture	12
2-8	Block diagram of proposed architecture	13
2-9	The framework of Res2SE50	14
2-10	The structure of squeeze and excitation block which use in their SERN	15
2-11	Experimental result of their deblurring algorithm	15
3-12	Structure of original and modify Squeeze and Excitation	17
3-13	Sample images of each class from the tomato ripeness dataset	18
3-14	Humans focus their vision on certain part of the image from Auto_Heat_Mapper program	19
3-15	Pixel intensity variance of each feature map from ResNet34 with ESEnet	23
3-16	Pixel intensity variance of each feature map from Extended ResNet34	23
3-17	Pixel intensity variance of each feature map from Darknet53 with ESEnet	24
3-18	Pixel intensity variance of each feature map from Extended Darknet53	24
3-19	Pixel intensity variance of each feature map from VGG16 with ESEnet	25
3-20	Fuzzy membership function of ResNet34 with ESEnet	28
3-21	Fuzzy membership function of Extended ResNet34	28
3-22	Fuzzy membership function of Darknet53 with ESEnet	29
3-23	Fuzzy membership function of Extended Darknet53	29
4-24	Performance of Darknet53 + ESEnet Fuzzy Pool, Darknet53 + ESEnet Mixed Pool	33
4-25	Performance of ResNet34 + ESEnet Fuzzy Pool, Mixed Pool	35
4-26	Confusion matrices of all models on the tomato ripeness	36
4-27	Performance of Extended Darknet53 Fuzzy Pool, Mixed Pool	39
4-28	Performance of Extended ResNet34 Fuzzy Pool, Mixed Pool	41
4-29	Confusion matrices for all model on the tomato ripeness dataset	42
4-30	Hayashi laboratory tomato harvesting robot appearance, vehicle structure	44
4-31	Intel RealSense D435 depth camera of the tomato harvesting robot attache to a UFACTORY xArm 7 robot manipulator	44
4-32	Output result of tomato ripeness classification	45

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
SE	Squeeze-and-Excitation
ESEnet	Extended Squeeze-and-Excitation Network
RGB	Red Green Blue
RGBD	Red Green Blue Depth
GMac	Giga Multiply-Accumulate
FLOPs	Floating Point Operations
SGD	Stochastic Gradient Descent
ReLU	Rectified Linear Unit
HSV	Hue Saturation Value
YOLO	You Only Look Once
VGG	Visual Geometry Group
ResNet	Residual Network

พระจอมเกล้าพระนครเหนือ
มหาวิทยาลัยเทคโนโลยี

CHAPTER 1
INTRODUCTION

1.1 Background and Problem Statement

As one of the most important crops globally, tomatoes play a vital role in agriculture, and their produce is a key ingredient in many cuisines around the world [1]. Due to their rich nutritional value and appealing taste, tomatoes are consumed by millions of people worldwide. As a result, they serve as a crucial resource in both the agricultural and food industries [2].



(a) (b)

FIGURE 1-1 (a), (b) various ripeness of tomato fruit on tomato tree

In commercial production, tomatoes are usually harvested during the mature ripening stage. This is because they have sufficient firmness, an extended shelf life, and the potential to turn red after being removed from the plant [3]. Consumer preferences, especially for color and texture in which samples of different tomato ripening stage color can be seen from Figure 1-1 [4] along with the need to reduce potential damage during transportation and other supply chain processes, are the main factors influencing the decision to harvest tomatoes at this stage. Although there are many studies focusing on classifying the maturity level of tomatoes, Ikeda et al., [5] used an assigned red HSV

color range to classify fully ripe tomatoes from unripe ones. They later implemented this algorithm in a tomato robot competition. Fujinaga et al., [6] used infrared imaging to recognize tomato fruits and estimated their maturity levels using probability distributions, which were quantified by combining the color values of detected fruits with the expertise of experienced farmers. However, there are still few studies that focus on using CNNs and attention mechanisms with adaptive pooling strategies to classify the maturity level of tomatoes, which interests us since attention mechanisms have been proven by many researchers to have quite an impact on CNN model classification performance.

In the present day, Convolutional Neural Networks (CNNs) have become an essential tool for various image processing tasks [7]. One of the fields where this technology has been widely applied is agriculture. In the agricultural sector, the ability to use or apply innovative technology is very important for the industry [8]. In the past decade, many research papers [9, 10, 11] have begun integrating the concept of attention mechanisms into their work to improve the performance of computer vision tasks [12], such as object detection [13] and image segmentation [14]. Attention mechanisms can be categorized into three types: channel attention [15], spatial attention [16], and finally mixed attention [17, 18]. Even though there is a significant amount of research focusing on CNN architectures and attention mechanisms, there is still a limited number of studies addressing adaptive pooling strategies and their implementation in both CNNs and attention mechanisms. This realization motivate us to study deeper into this field to explore the effects of adaptive pooling mechanisms toward convolutional neural networks.

The main concept and contribution of our proposed method involves creating a pooling method for attention mechanisms by combining max pooling and min pooling (an inverse of max pooling) together using two different methods. The first is mixed pooling [19], where we use a weight function with a fixed weight ratio to balance the distribution of max pooling and min pooling; for this research, the ratio is set at 0.5, which equally distributes between max and min pooling. Another method is fuzzy pooling, which includes a logic-based adaptive weight adjustment system to the

existing mixed pooling. The logic behind fuzzy pooling is a texture variance analysis using a Laplacian filter, which is a filter used for edge detection, image smoothing, texture enhancement, and low-light image enhancement [20]. In this work, it analyzes the texture density level of each passing feature map. This information is later fed to fuzzy logic, which determines the texture density of the processing feature map and then assigns a proper weight ratio to the mixed pooling for performance improvement in feature extraction.

In this research, we will not only compare the raw accuracy and F1 scores of the original model with the proposed models, but we will also demonstrate the side effects of the mixed and adaptive pooling mechanisms. We believe that analyzing and concluding research with only raw model accuracy comparisons cannot represent all the discoveries found throughout this research process. Some discoveries we found are particularly interesting—for example, the differences in feature map texture intensity variance scattering of adaptive pooling that operates within Squeeze-and-Excitation Networks versus within main CNN architectures. Additionally, the model parameters of adaptive pooling and fixed weight pooling compared to the original architecture show a clear distinction between weight pooling that operates inside Squeeze-and-Excitation Networks and the pooling that operates within the main CNN architecture.

1.2 Objective

1.2.1 To study pooling and its effects on convolutional neural networks

1.2.2 To develop adaptive pooling based on image feature map texture analysis and fixed weight pooling that combines two pooling operations at equal ratios

1.2.3 To develop an image classification program that classifies tomato ripeness levels

1.2.4 To implement and test the proposed pooling system against traditional pooling operations in tomato ripeness classification tasks

1.3 Scope

1.3.1 Use a reasoning approach such as fuzzy logic to adaptively adjust the weight of the pooling system based on the level of feature map variance

1.3.2 Implement, test, and validate the adaptive and fixed pooling operations on a tomato ripeness dataset

1.3.3 Compare the new proposed method with traditional pooling and record improvements

1.3.4 Use the K-fold cross-validation method to record model accuracy across every part of the dataset

1.4 Contribution

1.4.1 This research proposed a pooling method for attention mechanism, SE-Mixed Pooling based on weight function.

1.4.2 This research compares the model accuracy of the proposed SE-Mixed Pooling with the traditional SE-Average Pooling method, and also includes SE-Max Pooling and the baseline VGG16 for a more robust comparison.

1.4.3 To validate the performance of the proposed methods, this research adopted Tomato ripeness dataset to conduct the challenging tomato ripeness classification task

CHAPTER 2
LITERATURE REVIEW

2.1 Tomato Ripeness Classification: Current technique

In present day tomato ripeness classification or application regarding tomato harvesting has been extensively researched. Ikeda et al [5]., Addresses the critical challenge of automated tomato harvesting in agricultural robotics by developing an image processing system that can detect and guide harvesting operations for tomatoes that are partially obscured by leaves, stems, and other obstacles. The authors propose a computer vision approach using RGB-D sensors and OpenCV that distinguishes between visible and hidden tomatoes through color extraction in HSV space, shape detection using Hough circles, and labeling techniques to identify connected pixel regions. The system determines harvesting strategies by analyzing the spatial relationship between red tomato regions and green obstacle regions (leaves, stems, unripe tomatoes), providing directional guidance to a cup-shaped robotic end effector to approach tomatoes from angles that avoid obstacles, or flagging tomatoes as "difficult to harvest" when multiple stems are detected. Experimental validation using an Intel RealSense D435 camera in greenhouse environments demonstrated that the system could successfully detect mature tomatoes and provide appropriate harvesting instructions for single tomato-single obstacle scenarios, though challenges remained in distinguishing between overlapping tomatoes versus tomatoes divided by obstacles, and in handling complex multi-tomato, multi-obstacle environments. The research contributes to smart agriculture solutions aimed at addressing the declining agricultural workforce by providing a practical image processing framework for automated harvesting systems, though the authors acknowledge that further development is needed to handle the complexity of real-world greenhouse environments with multiple fruits and obstacles.

Fujinaga et al [6]., presents a comprehensive system for automated tomato monitoring and harvesting that generates a "tomato growth state map" visualizing fruit maturity stages, harvest times, and spatial locations within greenhouse environments. The authors developed a robot-based system that captures RGB, depth, and infrared images to detect tomato fruits using simple machine learning methods (HOG features with SVM) that surprisingly outperformed more complex convolutional neural networks when working with limited datasets. Crucially, they quantified the expertise of experienced farmers through surveys to classify fruits into six maturity stages (Green through Red following USDA standards) and predict harvest timing across three weekly periods. Their system achieved an F1-score of 0.829 for fruit detection and demonstrated reasonable accuracy in predicting yield and short-term harvest timing when tested in actual greenhouse conditions with 30 tomato plants. While the method showed limitations in long-term harvest predictions beyond one week, this work represents a significant contribution to precision agriculture by systematically encoding farmer expertise into automated systems and providing comprehensive growth state mapping that could inform both harvesting strategies and yield predictions for commercial tomato production.

2.2 Convolutional Neural networks (CNN)

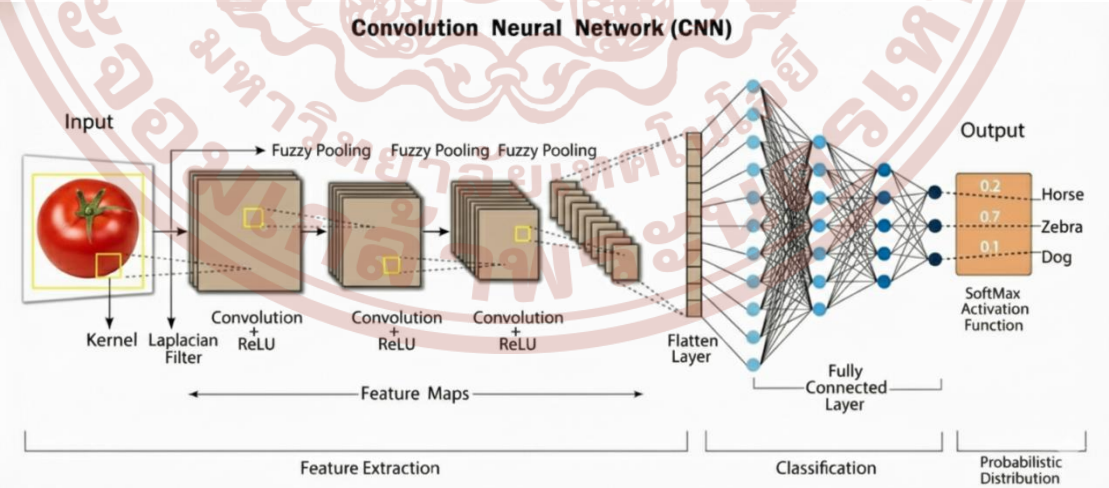


FIGURE 2-2 Basic structure of Convolutional neural network

In the present day, Convolutional Neural Networks (CNNs) in which its basic structure can be seen from Figure 2-2 have become an essential tool for various image processing tasks [7], representing a fundamental shift in how machines perceive and interpret visual information. One of the fields where this technology has been widely applied is agriculture, where the integration of computer vision systems has revolutionized traditional farming practices. In the agricultural sector, the ability to use or apply innovative technology is very important for the industry [8], as it enables precision farming, automated crop monitoring, pest detection, and yield optimization that were previously impossible with conventional methods. In the past decade, Recent literature [9, 10, 11] demonstrates that embedding attention modules significantly boosts accuracy across visual processing domains [12], particularly within applications like identifying objects [13] and partitioning images [14]. These attention mechanisms have emerged as a critical component in modern deep learning architectures, drawing inspiration from human visual attention systems that naturally focus on relevant parts of a scene while filtering out irrelevant information. Attention mechanisms can be categorized into three distinct types: channel attention [15], spatial attention [16], and finally mixed attention [17, 18]. Each type addresses different aspects of feature representation and processing within neural networks. SENet, channel attention emphasizes more distinctive features by modeling inter-channel relationships through global context aggregation, whereas spatial attention excels at capturing fine-grained details by focusing on various spatial locations within feature maps and identifying which regions contain the most relevant information for the task at hand. By combining both mechanisms, the mixed attention approach can simultaneously focus on feature map areas that have both dense and sparse textures, creating a more comprehensive understanding of the input data that leverages both global channel-wise statistics and local spatial relationships. Focusing more on channel attention, this mechanism can be used to improve the feature selection process by allowing the network to weigh the importance of each channel differently through learned parameters that adapt during training. It helps the model assign its attention to informative features while giving less attention to less useful ones, which technically means guiding the network to focus on important features that can contribute more to its task and reducing computational waste on irrelevant information. In practice, this leads to different treatment of the artificial neural network between important and unimportant features, creating a hierarchical feature representation that prioritizes discriminative characteristics, which is especially beneficial in scenarios with complex or noisy inputs where signal-to-noise ratio becomes a critical factor. As research goes on, further improvement of channel attention mechanisms could lead to a better performance model with enhanced robustness and generalization capabilities across diverse datasets and application domains.

2.2.1 Convolutional Layer

Convolutional layers are fundamentally designed to identify and extract localized patterns that frequently occur in natural images [19]. In general, the convolutional layer generates feature maps by linear convolutional filter followed by nonlinear activation functions, such as ReLU [20], sigmoid, tanh, etc. In this layer kth output feature map y_k can be calculated as follows:

$$y_k = f(\omega_k * x) \tag{2-1}$$

where x denotes the input image, ω_k stands for the convolutional filter associated with the kth feature map, $*$ indicate the 2D convolution operator which is used to calculate the inner product of the filter template at every location in the input image, and $f(\cdot)$ is the nonlinear activation function.

2.2.2 Non-linear Transformation Layer

To enhance visual recognition performance in CNNs, integrating rectifying non-linear transformations is a well-established strategy [21]. At their core, these layers induce a localized competitive dynamic among spatially aligned features distributed across multiple channels. Generally, this non-linearity is achieved through one of two primary methods. The first approach, known as local response normalization [22], calculates a standardized activation y_{kij} for the spatial coordinates (i, j) within the $k-th$ feature map, defined by the following equation:

$$y_{kij} = \frac{x_{kij}}{\left(1+\frac{\alpha}{N}\cdot\sum_{l=k-\frac{N}{2}}^{k+\frac{N}{2}}(x_{lij})^2\right)^{\beta}} \tag{2-2}$$

where the sum runs over N adjacent feature maps at the same spatial location, and the parameters of α and β can be determined using a validation set.

Another is the local contrast normalization [21] with the normalized output y_{kij} at the position (i, j) in feature map k as

$$y_{kij} = \frac{x_{kij}}{\left(1+\frac{\alpha}{M_1\cdot M_2}\sum_{p=i-\frac{M_1}{2}}^{i+\frac{M_1}{2}}\sum_{q=j-\frac{M_2}{2}}^{j+\frac{M_2}{2}}(x_{kpq}-m_{kij})^2\right)^{\beta}} \tag{2-3}$$

where the local contrast is computed within a local $M_1 \times M_2$ region with the center at (i, j), and m_{kij} is the mean of all x value within the above $M_1 \times M_2$ region in the kth feature map as computed as

$$m_{kij} = \frac{1}{M_1\cdot M_2}\cdot\sum_{p=i-\frac{M_1}{2}}^{i+\frac{M_1}{2}}\sum_{p=i-\frac{M_1}{2}}^{j+\frac{M_2}{2}}x_{kpq} \tag{2-4}$$

2.2.3 Feature Pooling Layer

In a CNN, the pooling layer operates by summarizing the outputs of neighboring groups of neurons within a specific kernel map. By pooling over these local neighborhoods, it strategically reduces the spatial resolution of the feature maps generated by the previous layer. This transformation converts the joint feature representation into a more compact and highly usable form. Consequently, it retains critical information while discarding extraneous details, providing the network with enhanced robustness to clutter and invariance to shifts in position, lighting conditions, or other input distortions.

2.2.3.1 Max Poolin g

To minimize variability by down-sampling convolutional outputs, a max-pooling operation [23] is frequently utilized. This mechanism works by isolating and passing forward only the highest value from every cluster of activations pixel.

$$p_{j,m} = \max\left(h_{j,(m-1)N+r}\right) \tag{2-5}$$

where $N \in \{1\ until\ R\}$ defines a pooling stride that facilitates overlapping between adjacent pooling windows when $N < R$. By compressing the K convolutional bands, the pooling layer effectively lowers the overall spatial dimensionality to $M = \frac{K-R}{N+1}$ band which the layer that comes as a result will be $p = [p_1, \ldots, p_m] \in R^{M.J}$

An example of Max-Pooling operation is shown in Figure 2-3.



FIGURE 2-3 Example of Max-Pooling operation

2.2.3.2 Average Pooling

The idea of average or mean pooling and extracting the feature, firstly introduce in [24] that is the first convolution based neural network. As shown in Figure 2-4. and (2-6), an average pooling layer perform down-sampling by dividing the input into rectangular pooling region and computing the average value of each region.

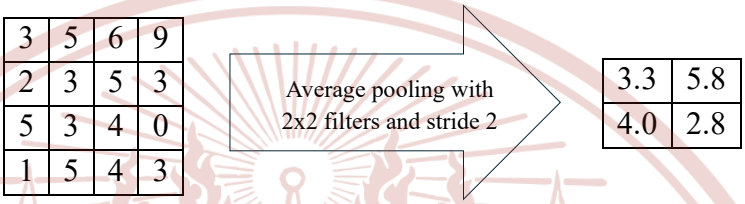


FIGURE 2-4 Example of Average-Pooling operation

$$y_{kij} = \frac{1}{|\mathcal{R}_{ij}|} \sum_{(p,q)\in\mathcal{R}_{ij}} x_{kpq} \tag{2-6}$$

2.2.3.3 L_p Pooling

Sermanet et al. [25] proposed the concept of L_p pooling claimed that its generalization ability is better than max pooling. In this pooling, a weighted average of inputs in pooling region. It is represented as given in (2-7)

$$s_j = \left(\frac{1}{|R_j|}\sum_{i\in R_j} a_i^p\right)^{\frac{1}{p}} \tag{2-7}$$

where $s_j$represent the output of the pooling operator at location j, a_i is the feature value at location I within the pooling region R_j. The value of p varies between 1 and ∞. When $p = 1$, L_p operator behaves as average pooling and at $p = \infty$ it lead to max-pooling. For L_p pooling, $p > 1$ is examined as a trade-off between average and max pooling.

2.2.3.4 Avg-TopK pooling

The Avg-TopK pooling [26] method aims to eliminate the drawback of both max and average pooling methods. It is an approximate value that represents all pixel by selecting the top k highest entries from input data. The average value of the k pixels with the highest values is determined. (2-8) represents the mathematical expression of the Avg-TopK pooling method.

$$F_{Avg-TopK(X)} = \frac{1}{k}\sum_{i=1}^{k} Y_i \tag{2-8}$$

X represents the set of element with pixel according to the pool size value selected from the data coming from the convolution layer. Yi presents the i-th pixel value. F(Avg-TopK) represent the average of the k highest items.

Figure 2-5 shows the operations for K=3 in Avg-TopK pooling operation with a filter size of 3 × 3 from 6 × 6 pixel input.

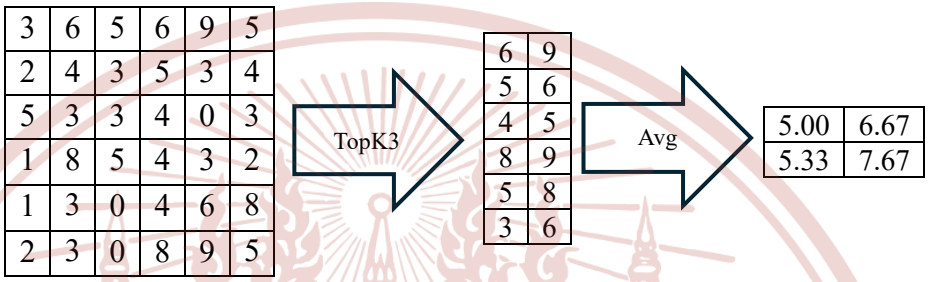


FIGURE 2-5 3 × 3 Avg-TopK pooling

As you can see, although max pooling and average pooling are the most common and popular types of pooling operations used in CNNs, there are many more new pooling methods that have been proposed which claim to be able to enhance the performance of traditional pooling methods. An interesting observation is that many of these new pooling methods, like Avg-TopK pooling, at their core still rely on the basic operations of traditional pooling like average and max pooling, but the new pooling methods tend to combine traditional pooling with various methods to achieve better overall model performance.

2.3 Squeeze-and-Excitation Network

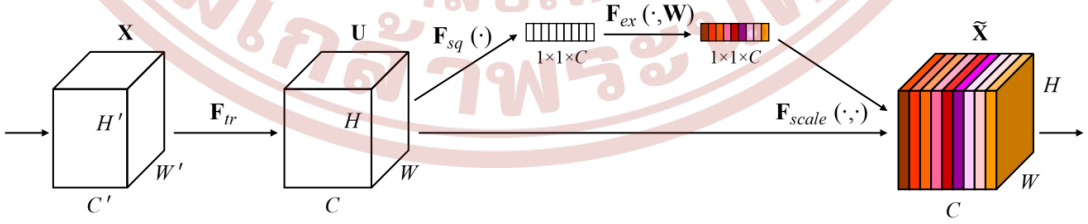


FIGURE 2-6 A Squeeze-and-Excitation block [15]

Squeeze-and-Excitation Network which structure can be observed from Figure 2-6 [15], is a popular example of channel attention mechanisms that improve CNN performance by adaptively reweighting channel features through a bottleneck

architecture that first compresses global information and then expands it back to generate attention weights, emphasizing important information and suppressing less useful signals while maintaining computational efficiency. The SE block achieves this through a two-step process: squeeze operation that aggregates feature maps across spatial dimensions, and excitation operation that learns channel-wise attention weights through fully connected layers. Even though there is a significant amount of research focusing on CNN architectures and attention mechanisms, there is still a limited number of studies addressing adaptive pooling strategies and their implementation in both CNNs and attention mechanisms, particularly in the context of dynamic feature aggregation that can adapt to varying input characteristics and task requirements.

Squeeze-and-Excitation Network has also been extensively researched and implemented in many applications.

Armağan et al [27]. presents a hybrid deep learning model for multi-class skin lesion classification that combines DenseNet121 and EfficientNetV2-B0 architectures with Squeeze and Excitation (SE) attention mechanisms in which its structure can be observe from Figure 2-7. The study addresses the critical need for early and accurate skin cancer diagnosis by leveraging dermoscopic images from the ISIC 2019 and ISIC 2020 datasets, encompassing eight different types of skin lesions including melanoma, nevus, basal cell carcinoma, and others. The researchers demonstrate that integrating SE attention mechanisms significantly improves classification performance, with the EfficientNetV2-B0 + SE model achieving 96.04% accuracy (0.74% improvement) and 96.12% AUC, while the DenseNet121 + SE model showed a substantial 2.45% increase in AUC from 92.16% to 94.61%. The hybrid approach effectively addresses the challenges of imbalanced datasets and similar-appearing lesions by enabling the model to focus on relevant features while suppressing irrelevant information, ultimately contributing to more reliable automated skin cancer detection systems that could assist healthcare professionals in early diagnosis and treatment decisions.

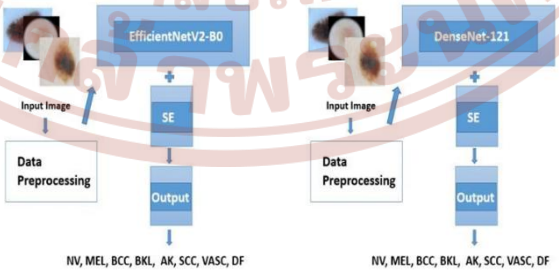


FIGURE 2-7 Hybrid model architecture [27]

Maurya and Chand [28] presents RFB-SE/LinkNet, a novel encoder-decoder architecture for road scene segmentation in autonomous vehicle applications which architecture structure can be observed from Figure 2-8. The authors address the challenging task of semantic segmentation in unstructured driving environments by combining a pre-trained SE-ResNeXt model as the feature extractor with a Receptive Field Block (RFB) enhanced with Squeeze-and-Excitation (SE) modules for improved multi-scale feature representation. The proposed framework leverages the SE attention mechanism to model channel interdependencies while the RFB module captures objects of varying sizes through different receptive fields and dilated convolutions. Evaluated on the India Driving Lite (IDDLite) dataset, which represents the complexities of Indian road conditions with seven semantic classes, the RFB-SE/LinkNet achieves 81.0% mean Intersection over Union (mIoU), outperforming existing state-of-the-art models including U-Net, LinkNet, DLinkNet, and EffUNet. The research demonstrates that combining attention mechanisms with multi-scale feature extraction significantly enhances the model's ability to segment complex road scenes, contributing valuable insights for autonomous vehicle navigation systems operating in challenging, unstructured environments.

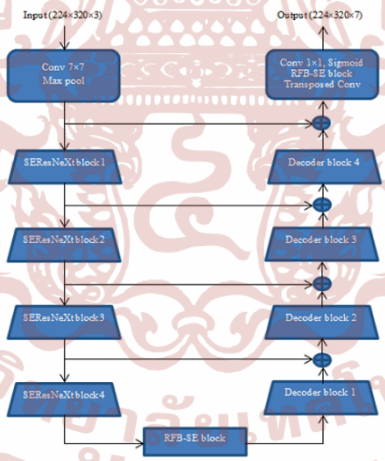


FIGURE 2-8 Block diagram of proposed architecture [28]

Wang et al [29]. proposes Res2SE50, an enhanced ResNet50 model for remote sensing image scene classification that integrates two novel attention mechanisms to address limitations in extracting local texture and global semantic features from complex satellite imagery. The model which structure can be observed from Figure 2-9 incorporates a Depth-wise Separable Convolution Squeeze-and-Excitation (DSE) module that replaces traditional fully-connected layers with depth-wise separable convolutions to enhance global channel features while preserving spatial information,

and a Channel Squeeze-and-Excitation (CSE) module that uses adaptive average pooling and 1×1 convolutions to generate local channel attention weights. Evaluated on the AID and NWPU-RESISC45 benchmark datasets, Res2SE50 achieved superior performance with accuracies of 94.09% and 91.52% respectively using only 20% training data, outperforming the baseline ResNet50 by 0.56-1.67% and surpassing state-of-the-art methods like LHNet by 0.79-1.63%. The key contribution lies in the cascaded integration of dual attention mechanisms that effectively capture both spatial and channel-wise features, with Grad-CAM visualizations confirming the model's enhanced ability to focus on discriminative scene features, establishing a new baseline for remote sensing scene classification and demonstrating the effectiveness of combining multiple attention types for improved feature discrimination in complex visual scenes.

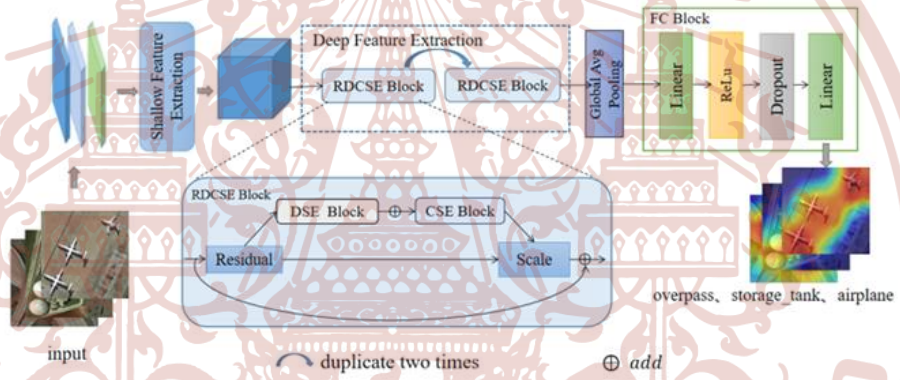


FIGURE 2-9 The framework of Res2SE50 [29]

Wu et al [30]. proposes SESRN (Squeeze and Excitation Scale-Recurrent Network), a novel convolutional neural network architecture for single image blind deblurring that integrates Squeeze and Excitation (SE) modules into a multi-scale recurrent framework in which system structure can be observe from Figure 2-10. The authors address the limitations of traditional optimization-based deblurring methods and existing CNN approaches by incorporating SE modules into residual blocks (SE-ResBlocks) within a scale-recurrent network structure, which processes images at multiple resolutions to progressively recover sharp images from blurry inputs. The SE modules enhance useful features while suppressing irrelevant information through channel-wise attention mechanisms, using global average pooling for feature compression followed by fully connected layers with bottleneck architecture to generate attention weights. Experimental evaluation on the GoPro and Köhler datasets which result can be observe from Figure 2-11 demonstrates that SESRN outperforms state-of-the-art methods including those by Nah et al., Kupyn et al. (DeblurGAN), and Tao et al. (SRN), achieving superior Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index

(SSIM) scores of 30.47 and 0.9363 respectively on the GoPro dataset, with only a minimal 7‰ increase in model parameters. The key contribution lies in demonstrating how attention mechanisms originally designed for image classification can be effectively adapted for image restoration tasks, specifically showing that SE modules can enhance the quality of deblurred images both quantitatively and visually by enabling better feature discrimination and detail recovery in complex motion blur scenarios.

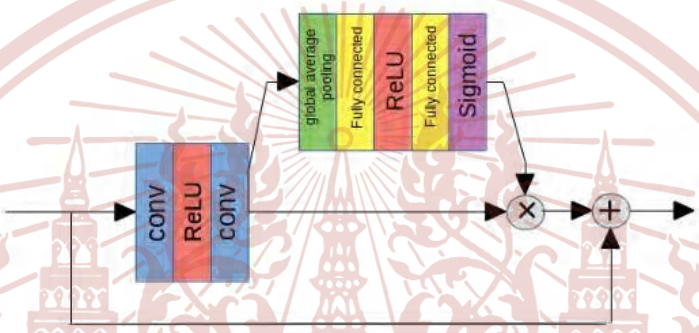


FIGURE 2-10 The structure of squeeze and excitation block which use in their SERN [30]



FIGURE 2-11 Experimental result of their deblurring algorithm [30]

2.4 Research Gaps and Opportunities

As can be seen from [27, 28, 29, 30], although the Squeeze and Excitation network has been extensively researched, the pooling mechanism that operates inside it has rarely been studied. This realization motivated us to conduct our research focusing on this section of the Squeeze and Excitation network. The main concept and contribution of our proposed method involves creating a pooling method for attention mechanisms by combining max pooling and min pooling (an inverse of max pooling) together using two different methods, addressing the limitation that traditional pooling operations only capture either maximum or average activations without considering the full spectrum of feature responses. The first is mixed pooling [31], where we use a weight function with a fixed weight ratio to balance the distribution of max pooling and min pooling, creating a hybrid approach that captures both prominent features (through max pooling) and suppressed features (through min pooling); for this research, the ratio is set at 0.5, which equally distributes between max and min pooling to ensure balanced feature representation without bias toward either extreme activation values. Another method is fuzzy pooling, which includes a logic-based adaptive weight adjustment system to the existing mixed pooling, incorporating intelligent decision-making capabilities that can dynamically adjust the pooling strategy based on input characteristics. The logic behind fuzzy pooling is a texture variance analysis using a Laplacian filter, which is a second-derivative filter used for edge detection, image smoothing, texture enhancement, and low-light image enhancement [32], providing robust texture characterization through its sensitivity to rapid intensity changes. In this work, it analyzes the texture density level of each passing feature map by computing the variance of Laplacian responses across spatial locations, quantifying the complexity and richness of local patterns within the feature representation. This information is later fed to fuzzy logic, which determines the texture density of the processing feature map through membership functions that classify texture characteristics into categories such as smooth, moderate, and highly textured regions, and then assigns a proper weight ratio to the mixed pooling for performance improvement in feature extraction by optimizing the balance between max and min pooling based on the identified texture properties, ultimately leading to more adaptive and context-aware feature aggregation.

CHAPTER 3
METHODOLOGY

3.1 System Overview

In this research, we will divide the experiments into two stages. In the first stage, we will focus mainly on the Squeeze and Excitation Network in which its structure can be observed from Figure 3-12. We will replace the average pooling layer at the head of the Squeeze and Excitation Network which implemented in ResNet34 and Darknet53 with an existing pooling technique—first with max pooling, and then with our proposed methods: mixed pooling and fuzzy pooling.

For mixed pooling, we will use a weight ratio of 0.5, which will share the usage between max pooling and min pooling equally. For fuzzy pooling, we will use the Mamdani fuzzy logic technique and set the range of feature map texture variance input by extracting the variance range captured from the tomato ripeness dataset after the model has been trained for one epoch, which will cover the entire training dataset. The output will range from zero to one depending on the feature map texture variance level. If the variance is high, the fuzzy logic will put more weight toward max pooling; otherwise, it will put more weight toward min pooling

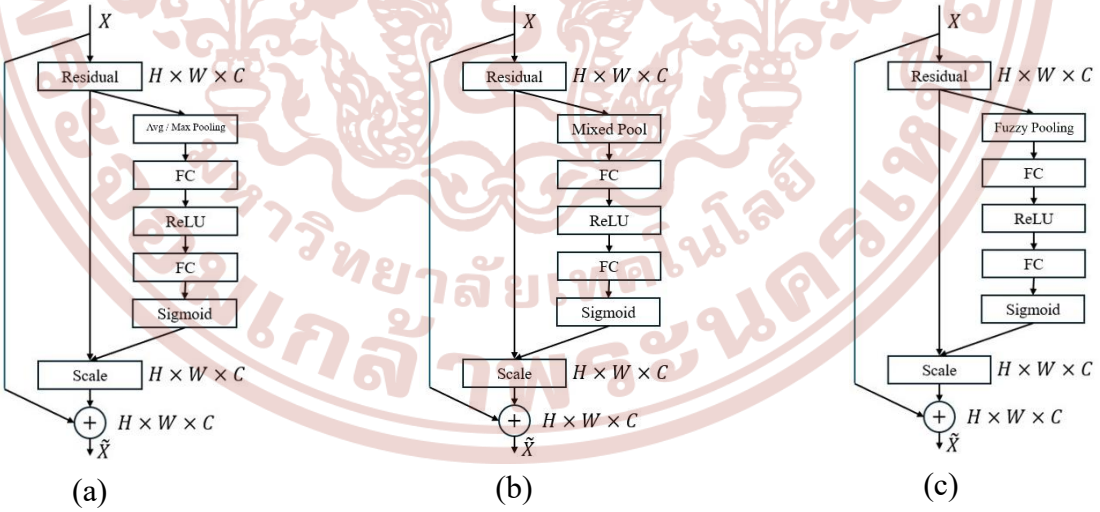


FIGURE 3-12 Structure of (a) original and (b) modify Squeeze and Excitation Network

In the second stage of the experiment, we will implement our proposed methods on the main CNN architectures by replacing the original pooling layers within ResNet34 and DarkNet53 with mixed and fuzzy pooling. For mixed pooling, we will still use the

same weight ratio of 0.5, providing balanced usage between max and min pooling. For fuzzy pooling, the membership function will be set using the same method as in the first stage.

3.2 Tomato ripeness dataset

In this work, we selected tomato ripeness as the benchmark dataset, as our long-term goal is to develop an improved object detection algorithm for tomato-picking robots. The ability to accurately identify and pick fully ripe tomatoes is crucial for performance. (Details regarding the tomato-picking robot itself are beyond the scope of this research.)



(a) (b) (c) (d) (e)

FIGURE 3-13 Sample images of each class from the tomato ripeness dataset: (a) 0% ripe (unripe), (b) 30% ripe, (c) 50% ripe, (d) 70% ripe, (e) 100% ripe.

The tomato ripeness dataset contains five classes (0%, 30%, 50%, 70%, and 100%) as shown in Figure 3-13, representing different ripeness levels of tomatoes. The dataset consists of a total of 3,263 images, with 2,935 images used for the training set and 328 images for the test set. The validation set is automatically generated from the training set using the K-fold cross-validation method, which will be explained in later sections.

We also implemented a class weighting system in the dataset due to class imbalance. This ensures that all the classes have equal importance on gradient updates when model is training regardless of how many samples we have from each class in the training data. This prevents models from predicting the more frequent class more often just because it's more common. The details of the sample quantities for each class in the training set are as follows: (0% Ripe, 100% Unripe): 732 samples, (30% Ripe, 70% Unripe): 547 samples, (50% Ripe, 50% Unripe): 488 samples, (70% Ripe, 30% Unripe): 668 samples, (100% Ripe, 0% Unripe): 495 samples.

The training environment used to conduct this research is as follows: CPU – 12th Gen Intel® Core™ i9-12900H 14-Core Processor; GPU – Nvidia GeForce RTX 3090 (24GB VRAM); RAM – 16 GB. The operating system used is Ubuntu 22.04.

3.3 Extended Squeeze and Excitation Network (ESEnet)

As can be seen from Figure 3-14 humans and animals generally do not process an entire scene all at once when visualizing something; instead, they tend to focus on specific parts through a selective attention process that has evolved as an efficient biological mechanism. This natural attention system allows organisms to direct their cognitive resources toward the most relevant information while filtering out unnecessary details, enabling them to navigate complex visual environments effectively.

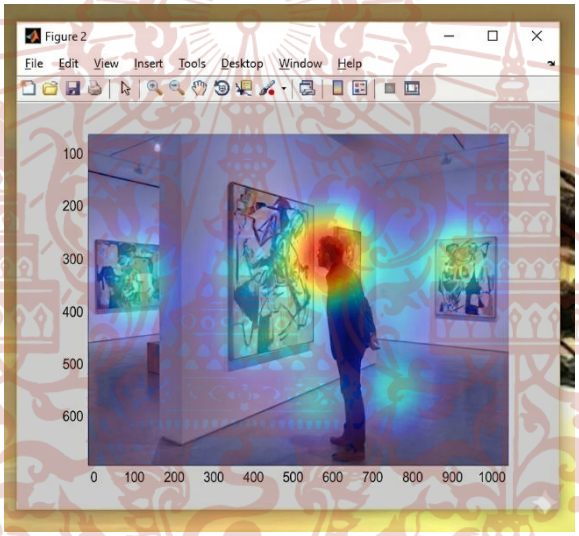


FIGURE 3-14 Humans focus their vision on certain part of the image from Auto_Heat_Mapper program

This concept is similar to how attention mechanisms—such as spatial or channel-based attention—operate, focusing only on important pixels and disregarding irrelevant ones without processing all pixels with equal computational weight. These computational attention mechanisms essentially mimic this biological process by learning to identify and emphasize the most informative parts of the input data during neural network processing.

In the original research paper, the main purpose of the Squeeze-and-Excitation Network was to improve the classification accuracy of CNNs in large-scale image classification problems due to its ability to model contextual information [33] by explicitly capturing channel interdependencies. The SE mechanism addresses the limitation that traditional CNNs treat all feature channels equally, regardless of their actual relevance to the specific task at hand. By introducing this channel attention mechanism, SE-Net enables networks to adaptively recalibrate feature responses, emphasizing informative channels while suppressing less useful ones based on global

context, which proved particularly effective for complex image classification tasks with diverse visual patterns.

For multi-class classification of tomato maturity, the Darknet53, VGG16 and Resnet34 model was utilized, with the Squeeze-and-Excitation Network integrated into it to improve classification accuracy by enabling the networks to adaptively focus on the most discriminative features for maturity assessment. The Squeeze-and-Excitation Network was introduced by Hu et al. in 2018 [15], and it focuses on combining spatial and channel-based information while suppressing irrelevant features through a parameter-efficient attention mechanism that enhances the representational capacity of convolutional networks. The operation of the Squeeze-and-Excitation Network can be separated into two distinct computational stages that work sequentially to achieve channel-wise feature recalibration

- **Squeeze**: This stage uses global average pooling to compress the spatial dimensions of each feature map into a single value per channel, effectively aggregating spatial information across the entire feature map. It captures global context and summarizes how important each channel is by computing the mean activation value, which serves as a compact descriptor of the channel's overall response pattern across all spatial locations.

- **Excitation**: The squeezed information is passed through a small neural network (with two fully connected layers separated by a ReLU activation and a bottleneck reduction ratio) to generate **channel-wise weights** that range between 0 and 1 through a sigmoid activation function. These learned weights are then used to **scale** (recalibrate) the original feature maps through element-wise multiplication, emphasizing important channels and suppressing less useful ones based on the global context understanding derived from the squeeze operation.

Besides the standard Squeeze-and-Excitation Network, this work also introduces an improved Extended Squeeze-and-Excitation Network (ESEnet) by changing its squeeze operation from average pooling to mixed pooling and fuzzy pooling, based on image texture variance level analysis using advanced pooling strategies. This modification addresses the limitation that global average pooling may lose important spatial information, particularly in regions with varying texture characteristics. This enhancement is under the assumption that in high texture intensity areas of the feature map, max pooling is more suitable as a texture extractor due to its ability to extract sharp edges and high-contrast texture areas by selecting the maximum activation values, while min pooling is used to balance out the max pooling in areas with smooth transitions in texture intensity by capturing the minimum responses that may contain complementary information. This will be controlled by a weighting function, which uses a fixed ratio (typically 0.5) for mixed pooling to ensure balanced feature representation, and an adaptive ratio for fuzzy pooling that dynamically adjusts based

on the texture density analysis performed using Laplacian filtering to optimize the pooling strategy according to local feature map characteristics.

3.4 Mixed pooling

In general CNN applications, max pooling and average pooling are among the most commonly used pooling operations due to their computational efficiency and simplicity in implementation across various network architectures. Max pooling operates by selecting the maximum value within each pooling window, which helps preserve the most prominent features and provides translation invariance, while average pooling computes the mean of all values in the pooling window, offering a smoother representation that can help reduce overfitting. However, in the Squeeze-and-Excitation Network, the authors of the original paper [15] state that, based on their comprehensive experiments on the ImageNet dataset [34], average pooling slightly outperforms max pooling in terms of error rate—22.28% compared to 22.57% for top-1 error, and 6.03% compared to 6.09% for top-5 error when integrated into the squeeze operation. This performance difference, while seemingly modest, represents a statistically significant improvement in large-scale image classification tasks where even small accuracy gains can be meaningful. The superior performance of average pooling in this context can be attributed to its ability to provide a more stable and comprehensive global descriptor by considering all spatial locations equally, rather than focusing solely on the maximum responses which might introduce noise or lose important contextual information. Therefore, average pooling is selected as the standard pooling operation for the Squeeze-and-Excitation Network, establishing it as the baseline approach for subsequent research and modifications in attention-based pooling strategies.

$$u = \lambda f_{max}(x) + (1 - \lambda) f_{min}(x) \tag{3-9}$$

For mixed pooling, we introduce a weighting function (3-9) to combine max and min pooling, using λ as a weighting parameter to control the contribution of each pooling operation in the final aggregated result. This linear combination allows for flexible control over how much emphasis is placed on the extreme values (maximum) versus the suppressed values (minimum) within each feature map region. Notably, we choose min pooling instead of average pooling to pair with max pooling, creating a complementary relationship between two opposite pooling strategies. This decision is made to increase the contrast between the feature extraction behaviors of max and min pooling, which would be less pronounced if average pooling were used instead, since average pooling represents a middle ground that might dilute the distinctive characteristics that each extreme pooling operation can capture. Max pooling excels at highlighting dominant features and strong activations, while min pooling captures the baseline or suppressed responses that might contain valuable information about feature

absence or weak signals, creating a more comprehensive feature representation when combined.

For the weight in mixed pooling, we choose 0.5 as the standard weighting factor, combining max and min pooling in equal proportions without bias toward either pooling strategy. This choice is made to investigate the effect of an equal combination of min and max pooling on the performance of the Extended Squeeze and Excitation Network, serving as a baseline configuration that ensures neither pooling operation dominates the feature extraction process. The equal weighting approach provides a balanced representation that captures both the prominent features through max pooling and the complementary information through min pooling, allowing the network to learn from the full spectrum of feature responses rather than being biased toward only the strongest or weakest activations within each channel.

3.5 Laplacian filter

Laplacian filter is one of the classical simple filters widely used in digital image processing applications. It is a high pass filter from signal point of view [32], which means it allows high-frequency components to pass through while attenuating low-frequency components, making it effective for detecting sharp transitions in image data. Laplacian filters will highlight regions of rapid intensity change in an image and are particularly good at finding fine details in an image [35], such as edges, corners, and texture boundaries that represent important structural information. The filter operates as a second-derivative operator that measures the rate of change of the gradient, making it highly sensitive to areas where pixel intensities change abruptly, which typically correspond to object boundaries and texture variations.

The Laplacian filter $L(x, y)$ of an image with pixel intensity value $I(x, y)$ is given by:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial x^2} \tag{3-10}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -8 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3-11}$$

This work used a Laplacian filter to measure the texture density of each feature map during the forward pass. The extracted information is then passed to a fuzzy logic system, which assigns an appropriate pooling weight ratio

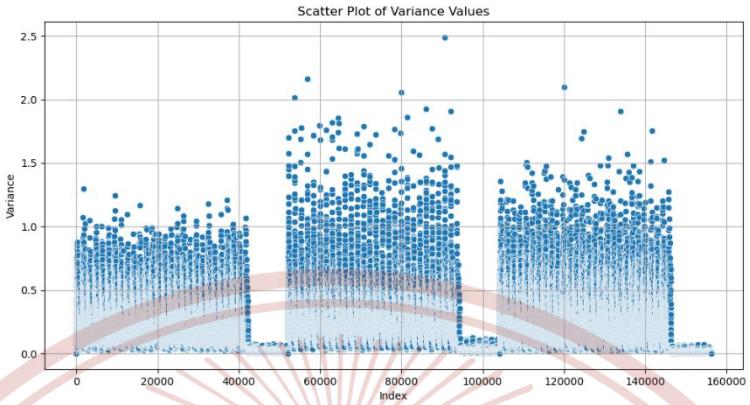


FIGURE 3-15 Pixel intensity variance of each feature map from ResNet34 with ESEnet

From Figure 3-15, we can observe the operation of the Laplacian filter in analyzing the pixel intensity of each feature map of tomato ripeness dataset. This information is crucial for the operation of the fuzzy logic system. Figure 3-15 presents the pixel intensity variance of ResNet34 with ESEnet. The data shown in Figure 3-15 originates from the fuzzy pooling block operating within the ESEnet. One interesting observation is that the pixel intensity variance captured from the fuzzy pooling block directly integrated into the architecture shown in Figure 3-16 appears more widely distributed and exhibits a broader range of values compared to the fuzzy pooling block embedded within the ESEnet. This may occur because feature maps that pass through many convolutional layers tend to develop more diverse characteristics, unlike the fuzzy pooling blocks that are directly connected to each convolutional layer in the ESEnet.

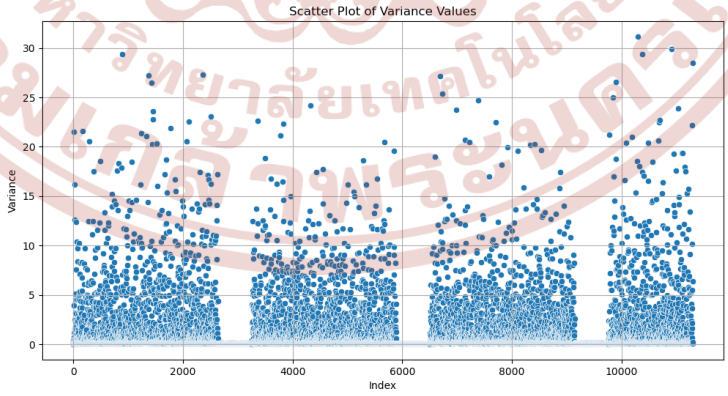


FIGURE 3-16 Pixel intensity variance of each feature map from Extended ResNet34

The Laplacian filter is also applied to Darknet53, both in the Extended Darknet53 (Darknet53 with fuzzy pooling) and in Darknet53 with ESEnet, to measure pixel

intensity in the feature maps and provide input to the fuzzy logic system as a weight function ratio λ.

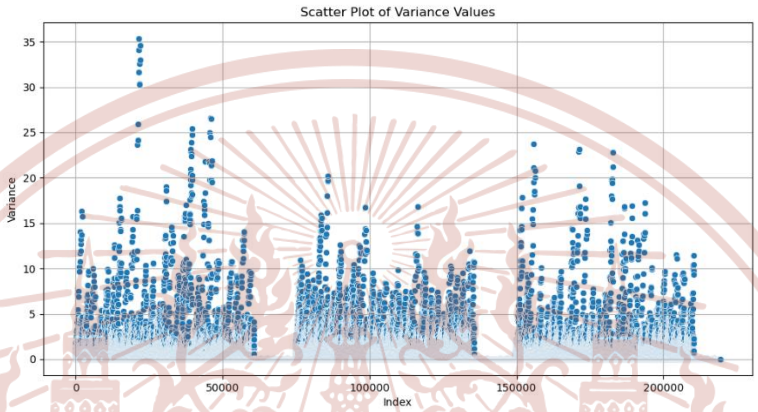


FIGURE 3-17 Pixel intensity variance of each feature map from Darknet53 with ESEnet

Figure 3-17 represents the pixel intensity variance of feature maps that pass through Darknet53 with ESEnet. When compared to its ResNet34 counterpart, the range of pixel intensity variance appears to be wider. This indicates that, even when using the same dataset, different CNN architectures can produce feature maps with distinct characteristics.

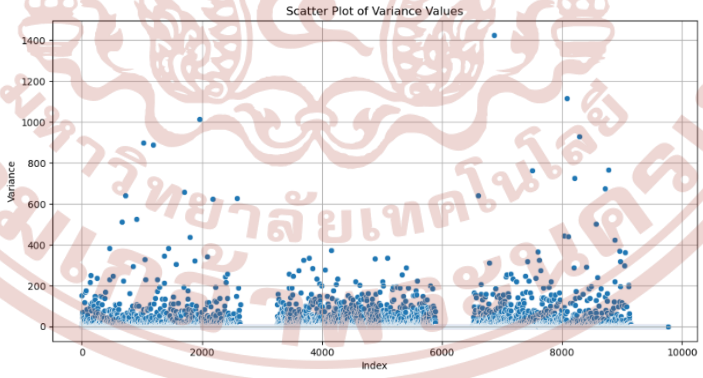


FIGURE 3-18 Pixel intensity variance of each feature map from Extended Darknet53

As seen in Figure 3-18, the pixel intensity variance range of the Extended Darknet53 is wider compared to Darknet53 ESEnet. This behavior is consistent with its ResNet34 counterpart, reinforcing the theory that the placement of the adaptive pooling block influences the characteristics of the feature maps it observes.

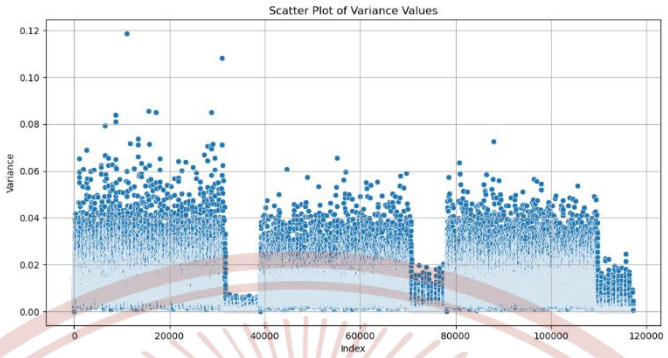


FIGURE 3-19 Pixel intensity variance of each feature map from VGG16 with ESEnet

The VGG16 architecture was also implemented with ESEnet from Figure 3-19 You can observe that the feature map texture variance levels extracted from Laplacian filter have a range from 0.00-0.04, which was the smallest among all three architectures tested. This can be explained by the fact that VGG16 is also the smallest of the three architectures, containing only 13 convolutional layers compared to ResNet34 (34 convolutional layers) and Darknet53 (53 convolutional layers).

With fewer convolutional layers, the amount of features that the architecture can extract is more limited, as each layer contributes to learning increasingly complex patterns. This results in lower texture variance in the feature maps, indicating more uniform feature representations across different input regions compared to the deeper networks. The use of pixel intensity variance information within the fuzzy logic system will be discussed in the next section.

The Laplacian filter plays a crucial role in this research, as it serves as the indicator for the fuzzy logic system to determine whether to give more weight to max pooling or min pooling. This assumes that in regions of the feature map with low texture intensity, priority should be given to min pooling, while in areas with rich texture intensity, max pooling should be prioritized as the main pooling operation. This decision-making process is made possible through texture density analysis using the Laplacian filter.

3.6 Fuzzy Pooling

Fuzzy logic is an advanced control strategy designed to handle highly nonlinear systems [36]. The term "fuzzy" refers to the inherent ambiguity or uncertainty present in many real-world situations, where conditions cannot be strictly classified as true or false. Instead, fuzzy logic offers a range of values between true and false, allowing for greater flexibility in identifying the most appropriate solution to a given problem. In this work, fuzzy logic is introduced to adaptively calibrate the pooling weight function. The input to the fuzzy logic system is the texture level of the feature map, represented

as texture variance. A higher variance indicates that the feature map contains sharp edges or low-level features. In such cases, fuzzy logic prioritizes max pooling, based on the assumption that max pooling is more effective at capturing sharp edges and areas with strong texture on plain backgrounds [37]. Conversely, in low-variance cases, fuzzy logic prioritizes min pooling, which is an inverse version of max pooling.

Formula that used to describe system dynamics of fuzzy logic is described as follows

$$\mu_F(x;a,b,c) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \\ 0, & x \ge c \end{cases} \quad (3\text{-}12)$$

(3-12) represents the general formula for the triangular fuzzy membership function. The symbol μ represent the membership function. which indicates how much a crisp input (in our case, variance or lambda) belongs to a fuzzy such as low, middle or high.

$$\mu_{low}(v) = \mu_{tri}(v;0,0,2) \quad (3\text{-}13)$$

$$\mu_{middle}(v) = \mu_{tri}(v;0,2,4) \quad (3\text{-}14)$$

$$\mu_{high}(v) = \mu_{tri}(v;2,4,4) \quad (3\text{-}15)$$

(3-13), (3-14), and (3-15) represent the variance membership functions. In this work, we define the variance level of the feature map as low, middle, and high. We measure the variance by collecting data from the Laplacian filter attached to the ESE block, which is connected to each residual block of Darknet53 and ResNet34 and Extended Darknet53 and Extended Resnet34 during the training process. We then analyze the data distribution, remove outliers, and define the upper and lower boundaries of the clustered variance values. These boundaries are used to assign the fuzzy membership functions for variance. Note that if the dataset or CNN architecture changes, these membership functions must be readjusted to suit the new environment.

$$\mu_{\lambda,low}(\lambda) = \mu_{tri}(\lambda;0,0,0.5) \quad (3\text{-}16)$$

$$\mu_{\lambda,middle}(\lambda) = \mu_{tri}(\lambda; 0,0.5,1) \tag{3-17}$$

$$\mu_{\lambda,high}(\lambda) = \mu_{tri}(\lambda; 0.5,1,1) \tag{3-18}$$

(3-16), (3-17), and (3-18) represent the membership functions of the lambda value, which will later be used to tune the weight function of the real-time adaptive pooling operation. In this case, since lambda serves as a weight function descriptor, its range is limited to between 0 and 1.

$$\lambda_{low}(\lambda) = \min\left(\mu_{v,low}(v), \mu_{\lambda,low}(\lambda)\right) \tag{3-19}$$

$$\lambda_{middle}(\lambda) = \min\left(\mu_{v,middle}(v), \mu_{\lambda,middle}(\lambda)\right) \tag{3-20}$$

$$\lambda_{high}(\lambda) = \min\left(\mu_{v,high}(v), \mu_{\lambda,high}(\lambda)\right) \tag{3-21}$$

(3-19), (3-20), and (3-21) represent the activation functions. These formulas follow the structure of the *Scikit-Fuzzy* library, which is the fuzzy logic framework used in this work. Specifically, the three activation functions $\lambda_{low}(\lambda), \lambda_{middle}(\lambda), \lambda_{high}(\lambda)$ are used to determine the degree to which each fuzzy rule contributes to the final output. These functions play a key role in modulating the weight ratio of the adaptive pooling mechanism based on the inferred importance of the feature map regions.

$$\lambda^* = \frac{\int \mu_a(\lambda)\cdot\lambda \, d\lambda}{\int \mu_a(\lambda) \, d\lambda} \tag{3-22}$$

Lastly, Equation (3-22) represents the centroid-of-area defuzzification function used in Mamdani fuzzy logic. This step produces a crisp λ value, which is then fed into the weight function.

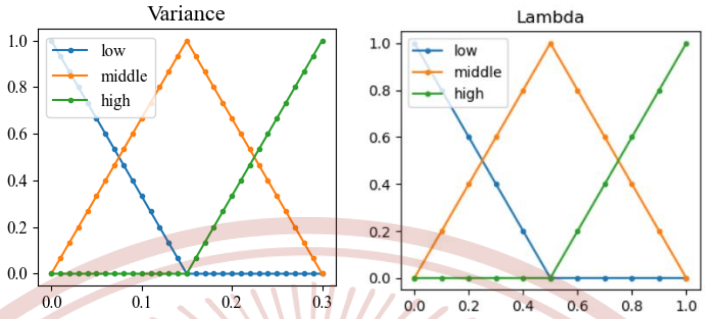


FIGURE 3-20 Fuzzy membership function of ResNet34 with ESEnet

Figure3-20 shows the fuzzy membership function of ResNet34 with ESEnet. The Mamdani fuzzy logic system with triangular membership functions is the type of fuzzy logic used throughout this work.

The input membership function in Figure 3-20. is defined based on pixel intensity variance in the feature map, which, for ResNet34 with ESEnet, can be observed in Figure 3-15. It may be noted that although many variance data points in Figure 3-15. exceed 1.0, the input to the fuzzy membership function is capped at 1.0. This is due to several reasons. First, outlier data points—those that lie far outside the main distribution—are discarded. Second, to define an appropriate cutoff value for the fixed fuzzy membership function, extensive experiments were conducted involving many trials and errors to determine the value that yields the best classification accuracy.

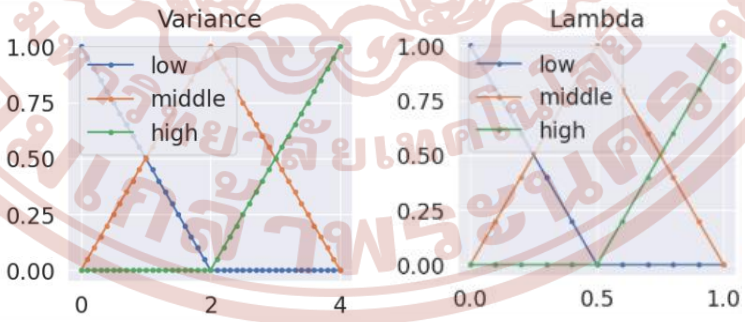


FIGURE 3-21 Fuzzy membership function of Extended ResNet34

The fuzzy membership function for the Extended ResNet34 is shown in Figure3-21. As previously explained in Section 4.1, due to the placement of the fuzzy pooling at the back of ResNet34, the range of recorded pixel intensity variance is significantly greater than that of ResNet34 with ESEnet. Interestingly, the experimental results show that the Extended ResNet34 demonstrates greater improvements over the original ResNet34

compared to ResNet34 with ESEnet. The details of the experiment will be presented in a later section.

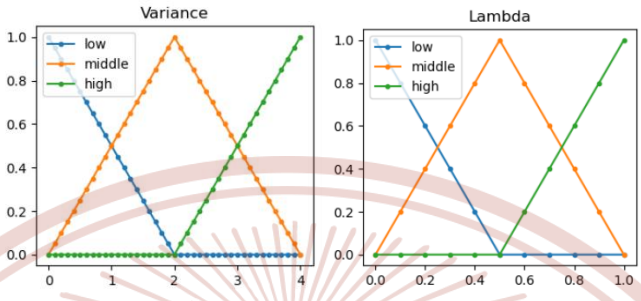


FIGURE 3-22 Fuzzy membership function of Darknet53 with ESEnet

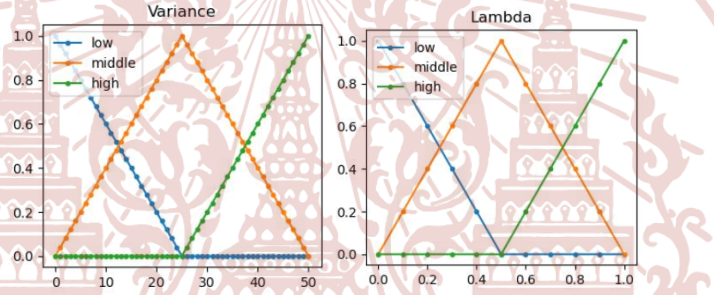


FIGURE 3-23 Fuzzy membership function of Extended Darknet53

As can be seen from Figure 3-22 and 3-23, and as mentioned in the previous section, both Darknet53 with ESEnet and Extended Darknet53 share a similar characteristic—an increase in the range of feature map intensity variance between the ESEnet and Extended architectures. A significant increase in feature map intensity variance is observed, similar to what was seen with ResNet34.

Additionally, Extended Darknet53 appears to show greater improvement over the original architecture compared to Darknet53 with ESEnet. This enhanced performance can be attributed to the Extended architecture's ability to capture more diverse and discriminative features through its modified structure. The increased variance range indicates that the Extended Darknet53 produces more varied feature representations across different spatial regions, suggesting better adaptability to complex input patterns. This improved feature diversity likely contributes to enhanced model performance, as the network can better distinguish between subtle differences in input data. The superior performance of Extended Darknet53 over the ESENet variant demonstrates the effectiveness of architectural extensions in leveraging the deep network's inherent capacity for complex feature learning.

3.7 Extended Darknet53 and Extended ResNet34

TABLE 3-1 Structure of Extended Darknet53

Type	*Filter*	*Size*	*Output*
Convolutional Convolutional	32 64	3 × 3 3 × 3 / 2	256 × 256 128 × 128
Convolutional Convolutional Residual	32 64	1 × 1 3 × 3	128 × 128
Convolutional	128	3 × 3 / 2	64 × 64
Convolutional Convolutional Residual	64 128	1 × 1 3 × 3	64 × 64
Convolutional	256	3 × 3 / 2	32 × 32
Convolutional Convolutional Residual	128 256	1 × 1 3 × 3	32 × 32
Convolutional	512	3 × 3 / 2	16 × 16
Convolutional Convolutional Residual	256 512	1 × 1 3 × 3	16 × 16
Convolutional	1024	3 × 3 / 2	8 × 8
Convolutional Convolutional Residual	512 1024	1 × 1 3 × 3	8 × 8
Fuzzy/Mixed pool		Global	
Connected		1000	
Softmax			

Darknet53 is the main convolutional neural network architecture used in YOLOv3 [38]. It is a continuous improvement over YOLOv2 and its predecessor, Darknet19. The architecture employs successive 3×3 and 1×1 convolutional layers and introduces shortcut connections. Darknet53 is larger than Darknet19, consisting of 53 convolutional layers—hence the name "Darknet53". In this work, we introduce a new Extended Darknet53 by replacing its original average pooling layer with our proposed mixed and fuzzy pooling layer, The fuzzy pooling layer has also been retuned based on the input pixel intensity variance range, as shown in Figure 3-18, while the mixed pooling layer has a fixed weight ratio set at 0.5. Table 3-1 presents the complete structure of our proposed Extended Darknet53. The only difference from the original architecture is the pooling layer, which has been replaced with fuzzy and mixed pooling, while the rest of the architecture remains unchanged.

ResNet34 (Residual Network) is a state-of-the-art architecture proposed by He et al. [39] from Microsoft Research Asia. The core design concept involves introducing a shortcut mechanism that directly connects the inputs to the outputs. This approach

effectively addresses the issue of performance degradation that typically arises as network depth increases. In this research, we have replaced the average pooling layer—originally located between the classification sequence and the fully connected layer—with mixed and fuzzy pooling. The fuzzy pooling layer has been tuned based on the input pixel intensity variance range, as shown in Figure 3-21, while the mixed pooling layer uses a fixed weight ratio of 0.5.

TABLE 3-2 Structure of Extended ResNet34

layer name	*output size*	*34-layer*
conv1	122×122	$7 \times 7,\ 64,\ \text{stride } 2$
conv2_x	56×56	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3,\ 64 \\ 3 \times 3,\ 64 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3,\ 128 \\ 3 \times 3,\ 128 \end{bmatrix} \times 4$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3,\ 256 \\ 3 \times 3,\ 256 \end{bmatrix} \times 6$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3,\ 512 \\ 3 \times 3,\ 512 \end{bmatrix} \times 3$
	1×1	Fuzzy/Mixed pool, 5-d fc, softmax

As can be seen from Table 3-2, only the average pooling layer has been replaced with adaptive pooling, while the max pooling layer at the conv2_x stage remains unchanged. The purpose of introducing Extended Darknet53 and Extended ResNet34 into the experiment is to further demonstrate the robustness of adaptive pooling (fuzzy and mixed pooling). In this case, the adaptive pooling operations are applied directly to the main pooling layers of the CNN, rather than being confined within the Squeeze-and-Excitation Network. This allows for a potentially greater impact on the system, as the pooling operations now influence the core structure of the network. As a result, we can observe not only the effects of adaptive pooling on attention mechanisms like the Squeeze-and-Excitation Network, but also its influence on the main CNN architectures, such as Darknet53 and ResNet34.

After this, the results will be examined by comparing Extended Darknet53 and Extended ResNet34. A deep analysis will be conducted on the results obtained from both architectures to explain the behavior of adaptive pooling when applied to different CNN architectures. Additionally, we will analyze its behavior when applied to attention mechanisms such as ESEnet and compare it with the original Squeeze-and-Excitation Network.

CHAPTER 4
EXPERIMENTAL RESULT

Measuring model performance is a challenging task, as it is essential to ensure that the results presented and any observed performance improvements are genuinely due to the model architecture itself, rather than noise, variance in the training process, or mere chance. To address this, our work incorporates K-fold cross-validation, dividing the dataset into 10 equal folds. This allows all models to be trained and validated on all parts of the dataset [40]. We report the fold with the highest F1 score to represent the model's performance. The F1 score is a widely used metric for evaluating CNN architectures across various fields [41]. Additionally, we fixed the random seed at 42 to ensure reproducibility, enabling future researchers to replicate our results.

4.1 Stage1: ESEnet

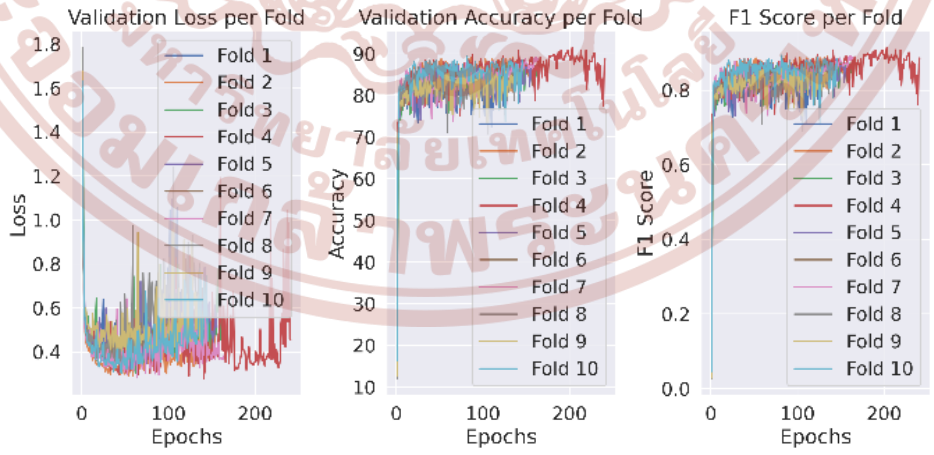
We will divide experimental into two stages, stage1 will present the result of ESEnet and compare with the original Squeeze-and-Excitation Network and, we also replaced the average pooling operation with max pooling to observe the differences when applying another deterministic pooling method. Two CNN architectures are used as the base for ESEnet: Darknet53 and ResNet34. Both architectures adopt almost the same training hyperparameters, starting with a batch size of 96, a learning rate of 0.0001 for Darknet53 and 0.001 for ResNet34, the SGD optimizer, and 1,000 training epochs with early stopping based on validation loss (patience = 100). The SE squeeze ratio was set to 16. The outcomes of our work are presented in Table. 4-3 and Table. 4-4.

The analysis of model performance on the tomato ripeness dataset revealed several insightful findings. First let's look at Darknet53 Table. 4-3 presents both the average and peak F1 scores measured on the test set across all folds for each model variant, including the original Darknet53, Darknet53+SE (Average Pool), Darknet53+SE (Max Pool), and the proposed Darknet53+ESE (Mixed Pool) and Darknet53+ESE (Fuzzy Pool). The proposed Darknet53+ESE (Fuzzy Pool) achieved the highest peak F1 score of 0.8746 and the highest average F1 score of 0.8422, slightly outperforming the Darknet53+SE (Average Pool) model, which recorded a peak F1 score of 0.8530 and an average F1 score of 0.8350 and original Darknet53 model with peak F1 score at 0.8497 and average F1 score at 0.8381. This suggests that integrating fuzzy pooling with the Squeeze-and-Excitation (SE) block can lead to a modest improvement in Darknet53 model performance. Closely following was Darknet53+ESE (Mixed Pool), which yielded a peak F1 score of 0.8566 and an average F1 score of 0.8385, still slightly outperforming both Darknet53+SE (Average Pool) and the original Darknet53. Interestingly, Darknet53+SE (Max Pool) performs the worst in terms of average F1 score at 0.8342 across all folds; however, it achieves the third-best peak F1 score at 0.8558 across all folds, being outperformed only by Darknet53+ESE (Mixed Pool) and Darknet53+ESE (Fuzzy Pool).

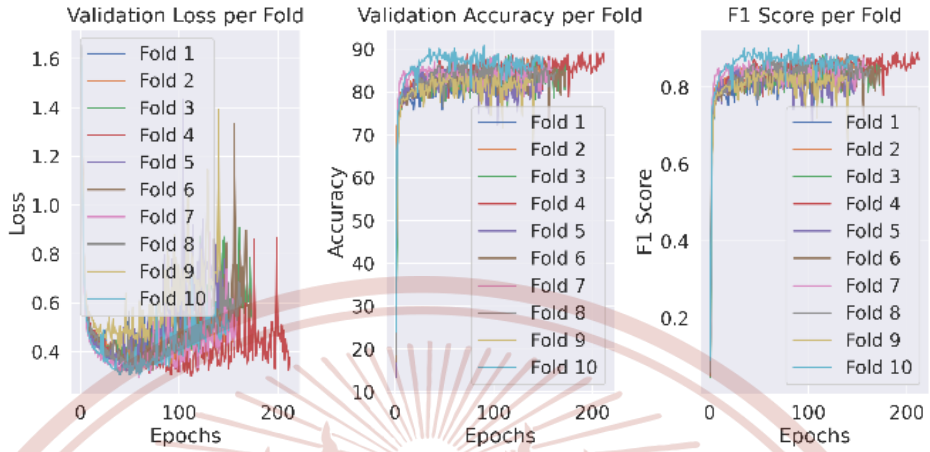
Figure 4-24 (a), (b) represents the model performance on validation set of ESEnet with Figure 4-24 (a) fuzzy pool and Figure 4-24 (b) mixed pool using tomato ripeness dataset we will measure validation loss, validation accuracy and validation F1 score of every folds. When compared between Figure 4-24 (a), (b) one interesting observation is that ESEnet with fuzzy pool tend to take longer to reach its steady state than mixed pool this could stem from many reasons such as in the number of operation it need to take to process data for one iteration for mixed pool since its fixed itself using a assign weight ratio there is no further step when data passing through Squeeze-and-Excitation block however for fuzzy pool when data reach the squeeze process it will transfer the data to Laplacian filter to analysis feature map texture density and after that send to fuzzy logic to process and assign a weight ratio these process may cause the training to take longer before reach the steady state.

TABLE 4-3 Experimental result of ESEnet with Darknet53 on test set

Model architecture	***Average F1 score (across all folds)***	***Peak F1 score (across all folds)***	***Model FLOPs (GMac)***	***Model Parameters (M)***
Darknet53	0.8381	0.8497	9.32	40.59
Darknet53 + SE (Average pool)	0.8350	0.8530	9.33	41.45
Darknet53 + SE (Max pool)	0.8342	0.8558	9.33	41.45
Darknet53 + ESE (Mixed pool)	0.8385	0.8566	9.33	41.45
Darknet53 + ESE (Fuzzy pool)	**0.8422**	**0.8746**	9.33	41.45



(a)



(b)

FIGURE 4-24 Performance of (a) Darknet53 + ESEnet Fuzzy Pool, (b) Darknet53 + ESEnet Mixed Pool

Next, let's look at ResNet34 based on the results in Table. 4-4 In terms of peak F1 score across all 10 folds, the model with the highest peak F1 score is ResNet34 + ESE (Fuzzy Pool) at 0.8661, slightly outperforming ResNet34 + SE (Average Pool), which achieved a peak F1 score of 0.8630, and the original ResNet34, which scored 0.8439. In terms of average F1 score across the 10 folds, ResNet34 + ESE (Mixed Pool) achieved the highest average F1 score of 0.8353, closely followed by ResNet34 + ESE (Fuzzy Pool) with a score of 0.8313. Both of these models outperformed ResNet34 + SE (Average Pool), which scored 0.8284, and the original ResNet34, which scored 0.8130. ResNet34 + SE (Max Pool) achieved a peak F1 score of 0.8472, slightly outperforming the original ResNet34, but underperforming compared to all other models. In terms of average F1 score, ResNet34 + SE (Max Pool) scored 0.8209, again slightly higher than the original ResNet34 but still lower than the rest, highlighting the instability of deterministic pooling. While deterministic pooling like average pool or max pool may perform well in certain settings, when its environment get change there is a very high change that those certain pooling may not perform as well as we expected while an adaptive pooling from the result that shown on both table show a high stable F1 score even across different CNN architectures.

TABLE 4-4 Experimental result of ESEnet with ResNet34

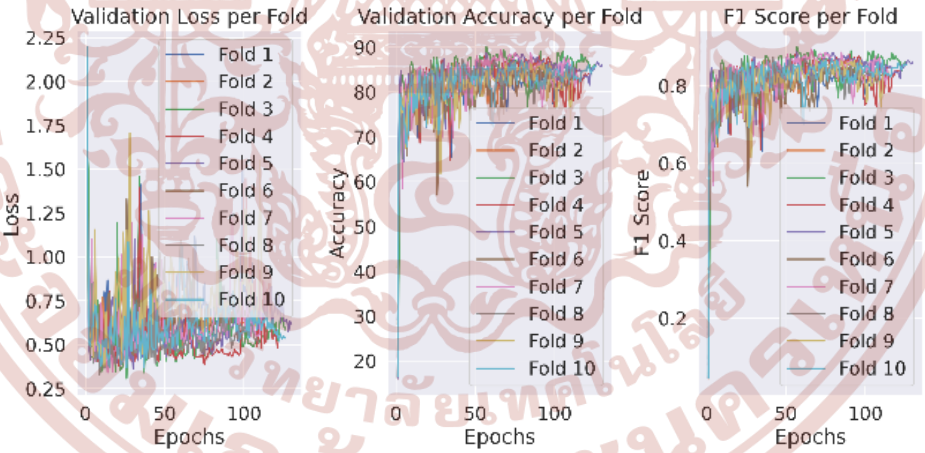
Model architecture	***Average F1 score (across all folds)***	***Peak F1 score (across all folds)***	***Model FLOPs (GMac)***	***Model Parameters (M)***
ResNet34	0.8130	0.8439	3.68	21.3

ResNet34 + SE (Average pool)	0.8284	0.8630	3.69	21.61
ResNet34 + SE (Max pool)	0.8209	0.8472	3.69	21.61
ResNet34 + ESE (Mixed pool)	**0.8353**	0.8635	3.69	21.61
ResNet34 + ESE (Fuzzy pool)	0.8313	**0.8661**	3.69	21.61

In conclusion, replacing deterministic pooling with mixed pool or fuzzy pool in the Squeeze-and-Excitation Network's "squeeze" operation improves feature extraction performance. These enhanced features lead to more accurate and robust classification results by better adapting to varying feature distributions and highlighting critical information.

Figure 4-25 illustrates the model performance on the validation set of ESEnet with (a) fuzzy pool and (b) mixed pool using the tomato ripeness dataset. From our observations, ESEnet with fuzzy pool tends to take longer to reach a steady state compared to the mixed pool. This behavior is also observed in Darknet53, for the same reason: the fuzzy pool architecture is more complex than the mixed pool, making data processing slower. Nevertheless, both methods achieve satisfactory results and outperform traditional methods.

(a)



(b)

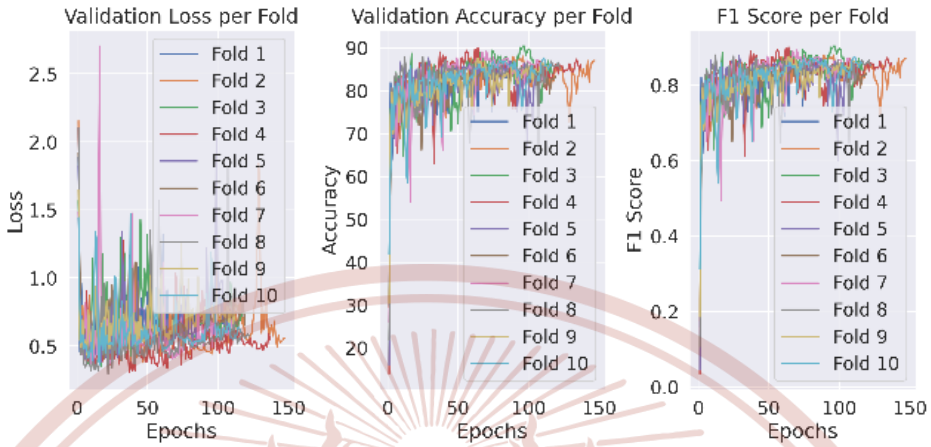
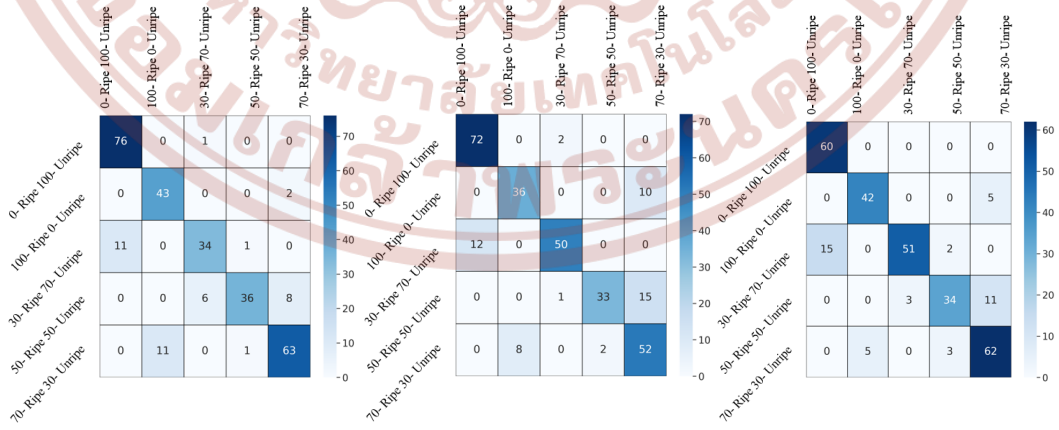


FIGURE 4-25 Performance of ResNet34 + ESEnet (a) Fuzzy Pool, (b) Mixed Pool

Figure 4-26 showed a confusion matrices of all models on the tomato ripeness dataset the matric was captured at fold with the maximum F1 score from each model. For general analysis, we can see that each class in the confusion matrix indicates good classification accuracy. The class “0-Ripe 100-Unripe” appears to have the highest number of true positive predictions compared to other classes across all methods and architecture types. This is due to the high number of data samples in the dataset—732 samples in the training set and 81 samples in the test set. The next best-performing class is “70-Ripe 30-Unripe,” which also shows a high number of correct predictions. This class has 668 samples in the training set and 74 samples in the test set, making it the second largest class in terms of dataset quantity among all classes. We can observe a slight improvement in true positive predictions for many classes when using Darknet53 + ESE (Fuzzy Pool) and ResNet34 + ESE (Fuzzy Pool) compared to traditional methods such as Darknet53, ResNet34, Darknet53 + SE (Average Pool), and ResNet34 + SE (Average Pool).



Darknet53 + ESE (Fuzzy pool) Darknet53 + ESE (Mixed pool) Darknet53 + ESE (Max pool)

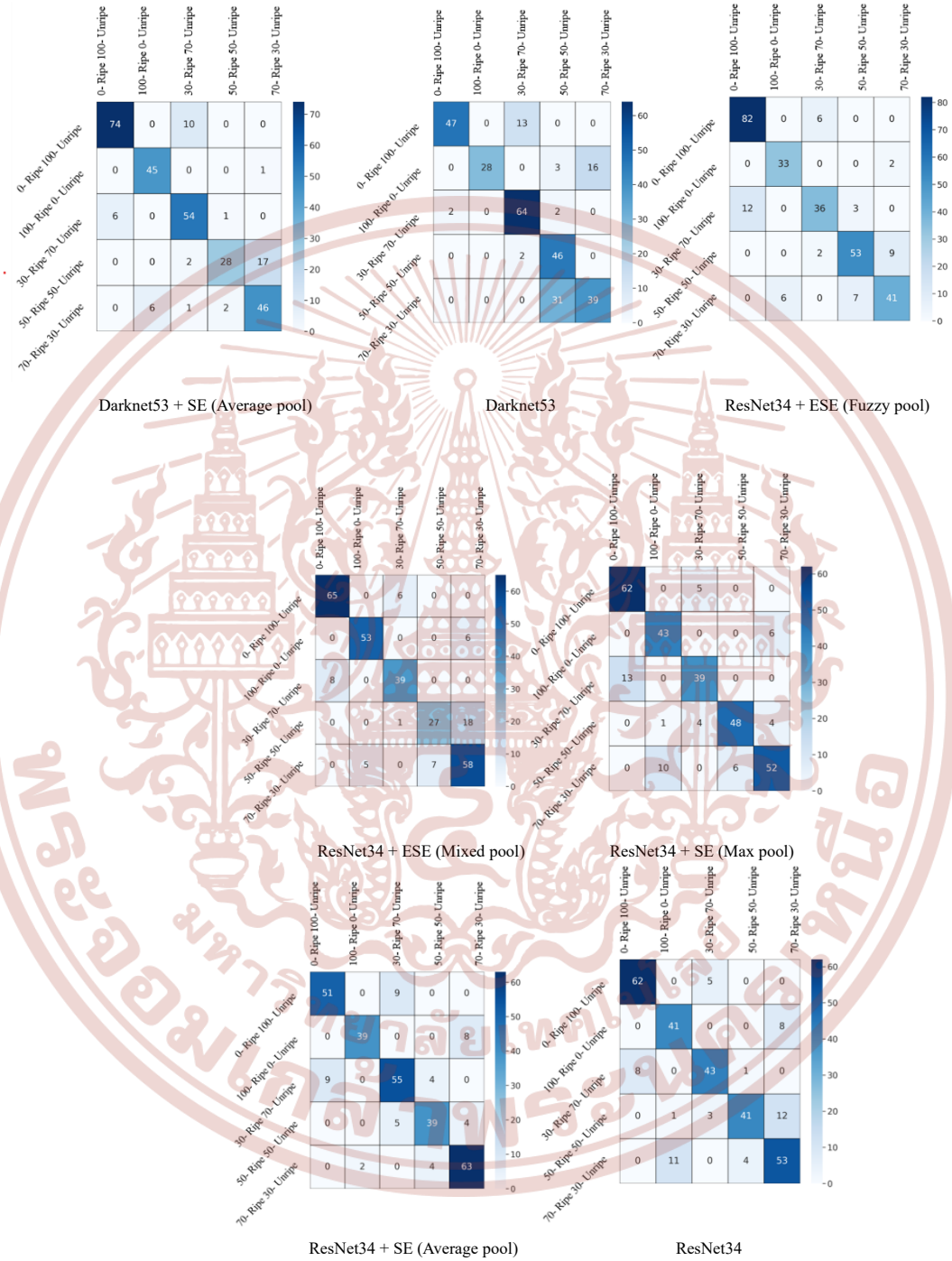


FIGURE 4-26 Confusion matrices of all models on the tomato ripeness

4.2 Stage2: Extended Net

This section is the second stage we will perform an experiment on Extended Darknet53 and Extended ResNet34 and compare with the original ResNet34 and Darknet53, we also replaced the average pooling operation in both of its main architecture with max pooling to observe the differences when applying another deterministic pooling method. All the CNN architecture in the second stage of the experiment will adopt the same training hyperparameter, starting with a batch size of 96, a learning rate of 0.0001, the SGD optimizer, and 1,000 training epochs with early stopping based on validation loss patience will be set at 100 for both architectures. Squeeze-and-Excitation Network has been removed from this stage of the experiment to observe the pure effect of adaptive pooling on the CNN architecture.

TABLE 4-5 Experimental result of Extended Darknet53

Model architecture	*Average F1 score (across all folds)*	*Peak F1 score (across all folds)*	*Model FLOPs (GMac)*	*Model Parameters (M)*
Darknet53	0.8381	0.8497	9.32	40.59
Darknet53 + Max pool	0.8302	0.8441	9.32	40.59
Darknet53 + Mixed pool	0.8345	0.8468	9.32	40.59
Darknet53 + Fuzzy pool	**0.8478**	**0.8720**	9.32	40.59

As can be observed from Table 4-5, Darknet53 + Fuzzy Pool achieves the best average and peak F1 scores of 0.8478 and 0.8720, respectively, compared to the original Darknet53, which records an average and peak F1 score of 0.8381. This result demonstrates the improved robustness of the adaptive pooling algorithm, even when it operates as a standalone block and is not tied to a Squeeze-and-Excitation Network. As for Darknet53 + Mixed Pool, in this experimental setting, while it outperforms Darknet53 + Max Pool, it does not surpass the original Darknet53, achieving an average F1 score of 0.8345 and a peak F1 score of 0.8468. This result can be explained by the fact that mixed pooling is still a deterministic method—it uses a fixed lambda weight ratio to fuse min and max pooling, and this lambda value must be manually tuned. What we aim to highlight in this experiment, by using a specific lambda value, is that under certain settings, this weight ratio can improve model accuracy. For example, in Section 7.1, mixed pooling used in ResNet34 + ESEnet and Darknet53 + ESEnet outperforms the original architectures in both average and peak F1 scores. To summarize, in certain settings where the weight ratio has been tuned, mixed pooling can perform well—

sometimes even outperforming the original architecture. However, it is recommended that mixed pooling should be fine-tuned each time there is a modification in the architecture for optimal performance. In terms of efficiency, Darknet53 + Fuzzy Pool and Darknet53 + Mixed Pool show a zero percent increase in computational consumption compared to the original architecture. This is a very interesting observation, as in the previous section, both mixed and fuzzy pooling implemented on ESEnet showed a slight increase in computational cost—with model FLOPs of 9.32 GMac for the original Darknet53 and 9.33 GMac for the versions with mixed and fuzzy pooling, and model parameters increasing from 40.59M on Darknet53 to 41.45M on the mixed and fuzzy pool variants in ESEnet. A similar slight increase in computational cost was observed in the ResNet architecture with Squeeze-and-Excitation Networks. From this experiment, we can confirm that the increase in computational consumption arises from the integration of Squeeze-and-Excitation Networks, not adaptive pooling. You can already see a hint of this increase in ESEnet + Average Pool and ESEnet + Max Pool, even before the implementation of adaptive pooling. The experiments in this section further reinforce the theory that adaptive pooling incurs almost zero computational overhead.

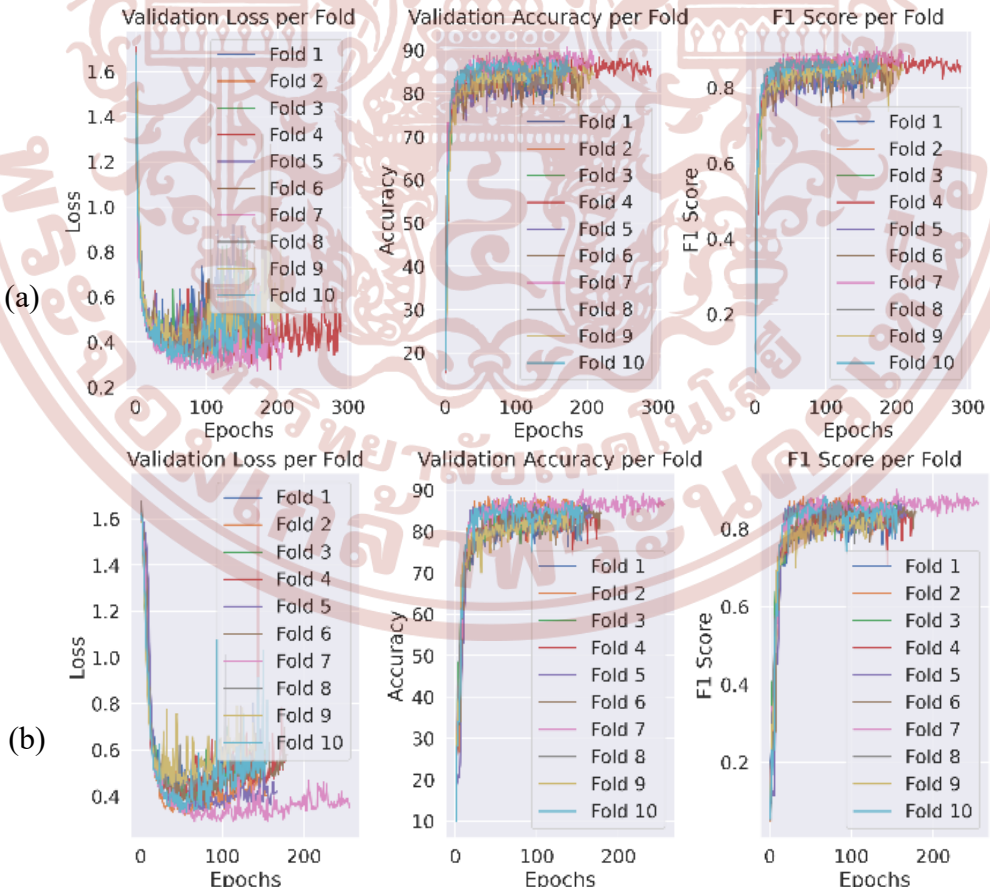


FIGURE 4-27 Performance of Extended Darknet53 (a) Fuzzy Pool, (b) Mixed Pool

Figure 4-27 shows the validation loss, accuracy, and F1 score of the Extended Darknet53 with fuzzy and mixed pooling across different folds. While some folds show signs of overfitting, this is not a concern due to the use of early stopping, which monitors validation loss and stops training once it stops improving. The reported F1 score is taken from the epoch with the lowest validation loss, ensuring that overfitted epochs do not affect the final evaluation. This approach guarantees that the results reflect the model's best generalization performance. Another interesting observation is that, similar to Darknet53 + ESEnet, fuzzy pool usually takes longer to complete training compared to mixed pool, due to its more complex system depth, as previously explained in Section 7.1.

TABLE 4-6 Experimental result of Extended ResNet34

Model architecture	***Average F1 score (across all folds)***	***Peak F1 score (across all folds)***	***Model FLOPs (GMac)***	***Model Parameters (M)***
ResNet34	0.8240	0.8443	3.68	21.3
ResNet34 + Max pool	0.8148	0.8301	3.68	21.3
ResNet34 + Mixed pool	0.8204	0.8491	3.68	21.3
ResNet34 + Fuzzy pool	**0.8344**	**0.8498**	3.68	21.3

From Table 4-6 Extended ResNet34 displayed a similar pattern of improvement for the adaptive pooling algorithm. For the fuzzy pool version, the average F1 score across all folds was achieved at 0.8344 and the peak F1 score was achieved at 0.8498, which outperformed both the original ResNet34 (which achieved an average F1 score across all folds at 0.8240 and a peak F1 score at 0.8443) and ResNet34 + Max pool (which achieved an average F1 score at 0.8148 and peak F1 score at 0.8301). From stage 2 of the experiment, it can be observed that the max pool variants of both Darknet53 and ResNet34 performed quite poorly, both in terms of average and peak F1 scores, despite their good performance in some settings of ESEnet. For example, ESEnet with Darknet53 max pool variant outperformed both average pool and original Darknet53 variants. This highlights our assumption that deterministic pooling methods have a more limited range of usage compared to adaptive pooling, which can automatically adjust itself to be suitable according to its training environment. Extended ResNet34 (Mixed pool) followed closely behind its fuzzy pool variant in terms of peak F1 score at 0.8491, outperforming both max pool and original variants of ResNet34. However, in terms of average F1 score, while it was able to outperform the max pool variant at 0.8204, it still slightly lagged behind the original ResNet34, which achieved

an average F1 score at 0.8240. This highlights the assumption that for mixed pool, even though it is a fusion weight system, its operational principle is still deterministic pooling due to the fact that the weight ratio has to be manually fixed and assigned for the whole training process and cannot tune itself during the training process. In terms of computational consumption, both mixed pool and fuzzy pool variants of Extended ResNet34 consume no more than both the original ResNet34 and ResNet34 + Max pool (model FLOPs = 3.68 GMac and model parameters = 21.3 M for all models).

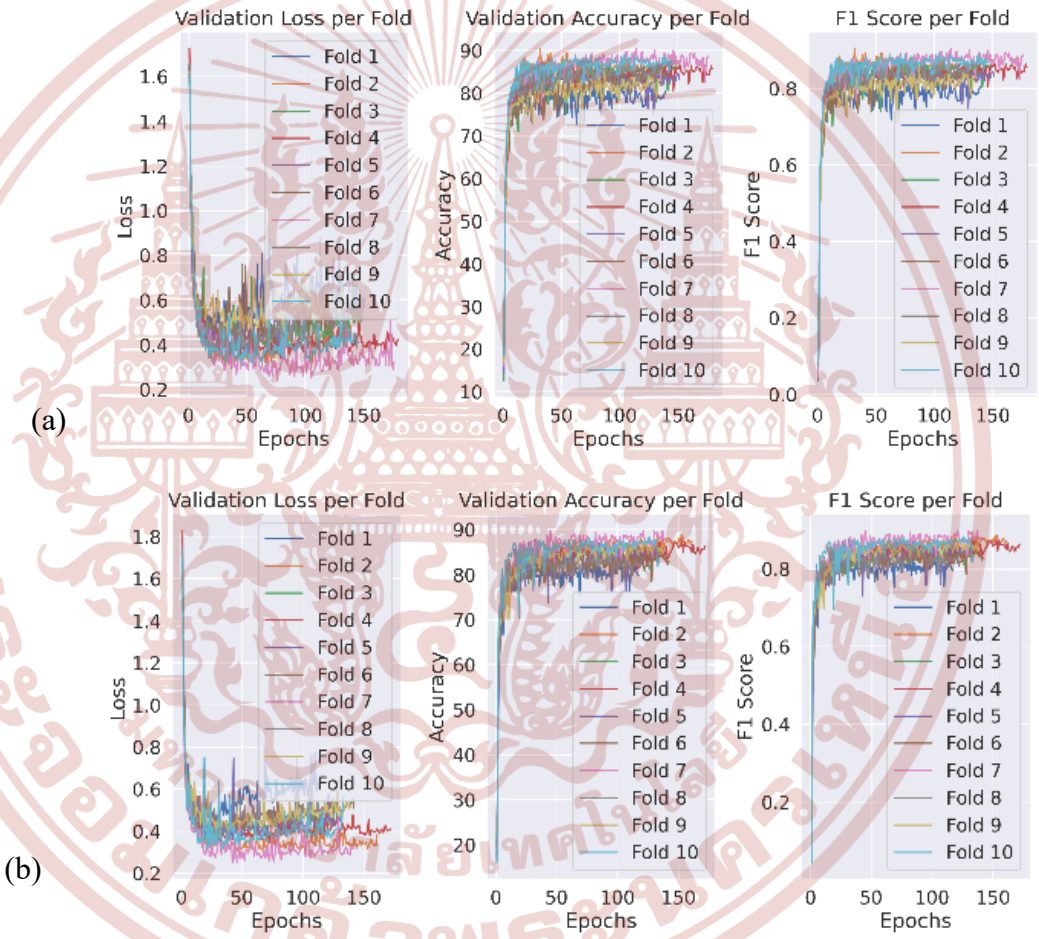
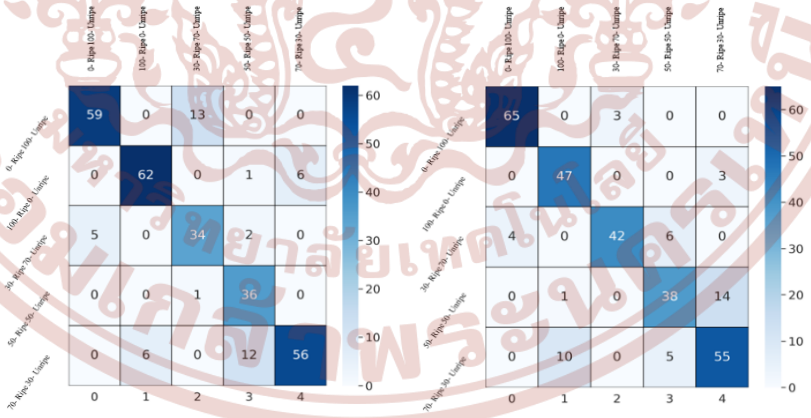


FIGURE 4-28 Performance of Extended ResNet34 (a) Fuzzy Pool, (b) Mixed Pool

In terms of validation loss, accuracy, and F1 score, it can be observed from Figure 4-28 that we can see some training folds that overfit, but in general, almost all folds have a normal downward slope validation loss graph. Even for the overfitted folds, we still have early stopping implemented, which will stop when it reaches 100 patience and will capture the model at the epoch with the lowest validation loss, as explained in the previous section.

To summarize the results of Extended Net, both CNN architectures that we selected for the experiment output a similar momentum of results, which is an improvement in model accuracy of the proposed fuzzy pool. This improvement was constant across architectures, highlighting the generalization ability of fuzzy pool in improving model accuracy. Mixed pool, under a correctly tuned weight ratio, can outperform traditional models, but due to its requirement for tuning every time there is a change in training environment or model architecture, the overall generalization ability is still not as good as fuzzy pool, which can be directly implemented into any system without the requirement for parameter tuning every time it must be used.

Figure 4-29 shows the confusion matrices of all models on the tomato ripeness dataset, captured at the fold with maximum F1 score from each model. The results are similar to ESEnet results in Figure 4-26, with high true positive predictions for class "0-Ripe 100- Unripe." This occurs for two main reasons: first, this class contains 732 samples (the highest number), and second, it has distinct characteristics (fully green tomatoes only). The uniformity of green coloration provides clear visual markers that enable more reliable feature extraction and classification. Other classes like "50-Ripe 50-Unripe" contain varied images with orange, red, and deep red tomatoes with orange spots. These non-uniform characteristics make classification difficult and cause confusion with similar classes like "70-Ripe 30-Unripe" (light red and orange tomatoes). The overlapping color ranges and transitional states between ripeness levels create ambiguous visual boundaries that challenge the models' discriminative capabilities. "Original Darknet53 (Average pool)" clearly demonstrates this issue, misclassifying "70-Ripe 30-Unripe" as "50-Ripe 50-Unripe" in 31 samples (almost half of its 39 correct classifications).



Extended Darknet53 (Fuzzy pool) Extended Darknet53 (Mixed pool)

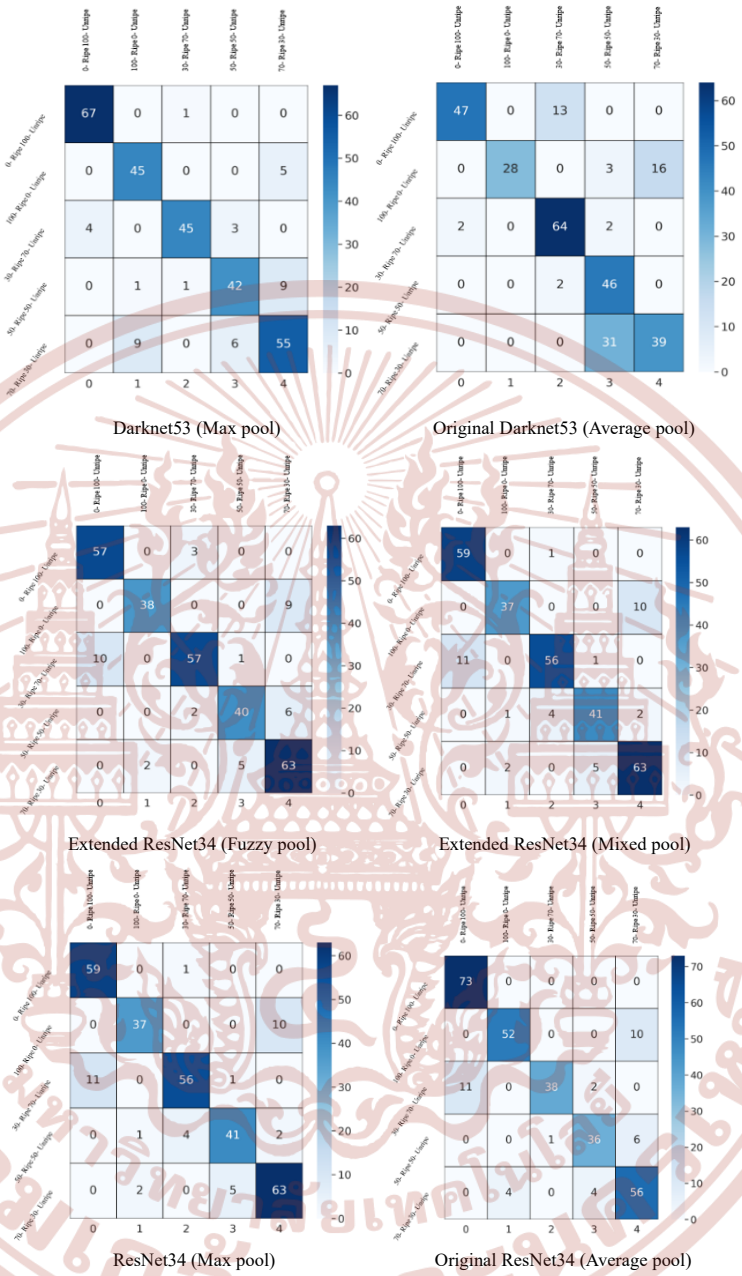


FIGURE 4-29 Confusion matrices for all model on the tomato ripeness dataset

It also misclassifies many as "100-Ripe 0-Unripe" due to similar visual characteristics. Interestingly, ResNet34 shows fewer misclassifications between "70-Ripe 30-Unripe" and "50-Ripe 50-Unripe," suggesting that different CNN architectures affect classification performance even with the same dataset. All architectures perform well, with slight advantages for Extended ResNet34 and Extended Darknet53.

4.3 Implementation

The Tomato-Harvesting competition has been organized since 2014 [42] in Kitakyushu, Japan, with the aim of developing agricultural robots, arousing the interest of young researchers in robotics in agriculture, and giving back the developed technology to the agricultural field [43]. Hayashi laboratory at Kyushu Institute of Technology has developed a tomato harvesting robot and has been constantly participating in the Tomato-Harvesting competition since 2014.

This work has been implemented and tested on the tomato harvesting robot developed by Hayashi laboratory by utilizing its hand camera, which is placed at the end effector, to perform close-up classification of tomato ripeness levels.



(a) (b)

FIGURE 4-30 Hayashi laboratory tomato harvesting robot (a) appearance, (b) vehicle structure



FIGURE 4-31 Intel RealSense D435 depth camera of the tomato harvesting robot attached to a UFACTORY xArm 7 robot manipulator

The camera that will be used for this experiment is the Intel RealSense D435, which is an RGBD camera attached to the end effector of UFACTORY xArm 7 robot manipulator. This setup was also used by the tomato harvesting robot to detect and measure the depth from the end effector to the tomato fruit and use the manipulator to reach and grasp, but for this work we will only use its RGB information. The model and classification program have been moved from the training PC, whose specifications were already declared in section 6, to the author's own laptop for demonstration, which has the following specifications:

Lenovo IdeaPad Gaming3 15ACH6, CPU – AMD Ryzen 5 5600H with Radeon Graphics; GPU – Nvidia GeForce RTX 3060 Laptop GPU (6GB VRAM); RAM – 16 GB. The operating system used is Ubuntu 22.04.

As shown in Figure 4-31, the tomato fruit will be attached to a rack covered with fake tomato leaf background to simulate the real background the robot will see during competition. We will classify one tomato at a time because in reality, the hand camera will only be used for precise measurement of tomato fruit at close distance, since the tomato harvesting robot will use a Kinect DK v2 depth camera at the robot base to perform rough overall tomato detection.

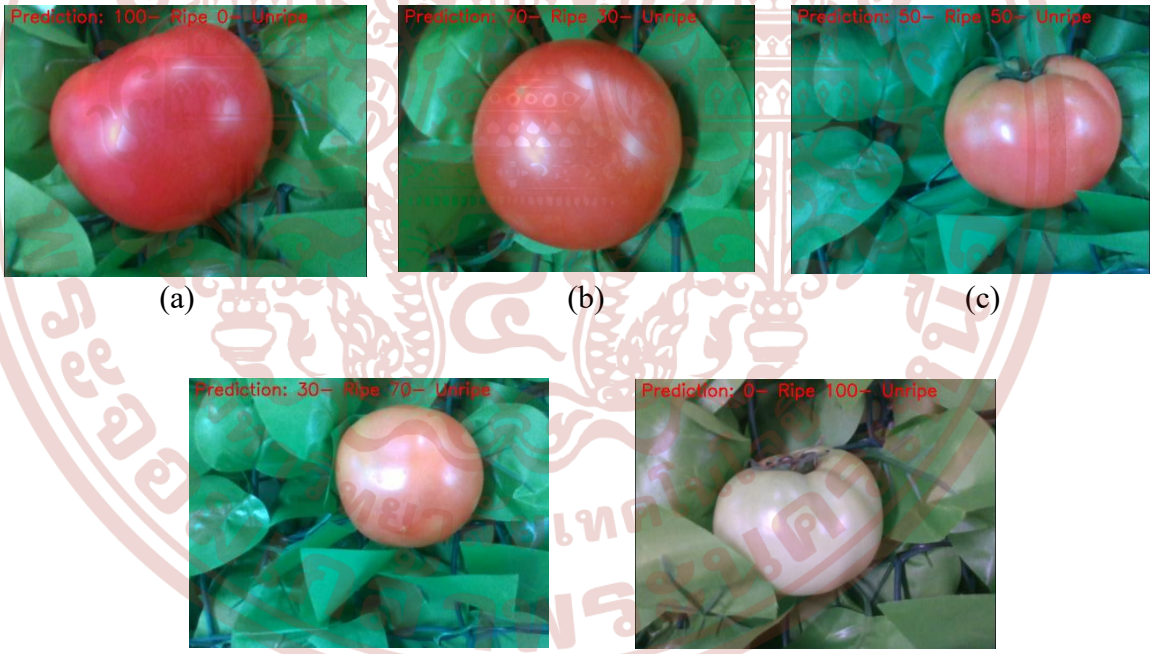


FIGURE 4-32 (a), (b), (c), (d), (e) Output result of tomato ripeness classification

Figure 4-32 shows a classification result of tomato ripeness using a real tomato fruit as a sample for the classification program. In terms of qualitative analysis from what we observed in real experiments, we found that a deep red tomato (100% Ripe, 0% Unripe) has a very robust classification process with only minor misclassification

observed. The same goes for fully green tomatoes (0% Ripe, 100% Unripe), which only have minor misclassification. This can be explained by the fact that these two classes of tomatoes have very distinct characteristics, such as high color contrast from other classes. While there are still quite a few misclassifications observable in the three other classes (30% Ripe, 70% Unripe), (50% Ripe, 50% Unripe), and (70% Ripe, 30% Unripe), this makes sense due to the fact that these three classes share quite similar characteristics in color, where all three classes only have slight differences in shade of red, which could cause misclassification to occur.

พระจอมเกล้าพระนครเหนือ
มหาวิทยาลัยเทคโนโลยี

CHAPTER 5
CONCLUSION

5.1 Conclusion of research

We have proposed an adaptive pooling method for analyzing the ripeness of tomato fruit based on feature map texture intensity analysis. The system has been implemented on CNNs by replacing their pooling layers. To prove its robustness, we implemented it on two CNN architectures: ResNet34 and Darknet53, both of which yielded positive results with improved classification accuracy (Darknet53 + Fuzzy pool achieved average F1 score across all folds at 0.8478 outperform its baseline model which achieved average F1 score across all folds at 0.8381) and (ResNet34 + Fuzzy pool achieved average F1 score across all folds at 0.8344 outperform its baseline model which achieved average F1 score across all folds at 0.8240) . In addition to CNNs, we also implemented the adaptive pooling system on the attention mechanism "Squeeze-and-Excitation Networks." This aims to fill the research gap clearly identified in previous studies, which stated that the average pooling used in Squeeze-and-Excitation Networks could be replaced and further investigated. This implementation also yielded positive results, showing improved classification accuracy on both CNN models where Squeeze-and-Excitation Networks were implemented, compared to traditional methods using average pooling (Darknet53 + ESE (Fuzzy pool and Mixed pool) achieved average F1 score across all folds at 0.8422 and 0.8385 outperform its baseline model which achieved average F1 score across all folds at 0.8381 and Darknet53 + baseline SE (Average pool) at 0.8350) and (ResNet34 + ESE (Fuzzy pool and Mixed pool) achieved average F1 score across all folds at 0.8313 and 0.8353 outperform its baseline model which achieved average F1 score across all folds at 0.8130 and ResNet34 + baseline SE (Average pool) at 0.8284). We introduced K-fold cross-validation to our result collection process, dividing the dataset into ten different folds and training on all parts of the dataset. This approach ensures that any improvement recorded in this research stems from actual model performance rather than random noise occurring during the training process. The algorithm was also implemented on a tomato harvesting robot to analyze qualitative results during real-world operation.

5.2 Future work and Limitation

In future work, we plan to introduce additional datasets to further analyze the robustness of our adaptive pooling system and expand its application beyond agricultural imaging analysis. We also intend to integrate more CNN architectures into the experiment to further validate the robustness of the adaptive pooling system, including MobileNet, VGG-16, and EfficientNet. Additionally, we plan to replace the

fuzzy logic component with an artificial neural network that can learn and dynamically adjust the weight ratios based on classification accuracy feedback.

As for limitations, this research only conducts performance evaluation on a tomato ripeness dataset since we want to define a specific application that this research will be implemented on. However, further datasets could be tested for robustness of the experimental results. More CNN architectures could also be employed to further test the improvement's robustness across various CNN architectures.



REFERENCES

1. Bergougnoux, V. “The history of tomato: From domestication to biopharming.” Biotechnology Advances. 32(1) (2014) : 170-189.

2. Sattar, S., et al. “Tomatoes Unveiled: A Comprehensive Exploration from Cultivation to Culinary and Nutritional Significance.” Qeios. (2024).

3. Bapat, V. A., et al. “Ripening of fleshy fruit: molecular insight and the role of ethylene.” Biotechnology advances. 28(1) (2010) : 94-107.

4. Oltman, A. E., Jervis, S. M., Drake, M. A. “Consumer attitudes and preferences for fresh market tomatoes.” Journal of food science. 79(10) (2014) : S2091-S2097.

5. Ikeda, T., et al. “Tomato Recognition for Harvesting Robots Considering Overlapping Leaves and Stems.” J. Robot. Mechatron. 33(6) (2021) : 1274-1283.

6. Fujinaga, T., Yasukawa, S., Ishii, K. “Tomato Growth State Map for the Automation of Monitoring and Harvesting.” J. Robot. Mechatron. 32(6) (2020) : 1279-1291.

7. Goto, T., Ishigami, G. “CNN-Based Terrain Classification with Moisture Content Using RGB-IR Images.” J. Robot. Mechatron. 33(6) (2021) : 1294-1302.

8. Yoshida, T., Fukao, T., Hasegawa, T. “Fast Detection of Tomato Peduncle Using Point Cloud with a Harvesting Robot.” J. Robot. Mechatron. 30(2) (2018) : 180-186.

9. Tian, Y., Li, X., Ma, S. “Lightweight Encoder with Attention Mechanism for Pipe Recognition Network.” J. Robot. Mechatron. 36(2) (2024) : 343-352.

10. He, H., et al. “An Intent Recognition Method for Aerial Swarm Based on Attention Pooling Mechanism.” J. Adv. Comput. Intell. Intell. Inform. 29(1) (2025) : 5-11.

11. Yang, T., Chan, A. B. “Visual tracking via dynamic memory networks.” IEEE transactions on pattern analysis and machine intelligence. 43(1) (2019) : 360-374.

12. Liang, L., et al. “Improvement of residual attention network for image classification.” International Conference on Intelligent Science and Big Data Engineering. (October 2019) : 529-539.

13. Khan, F. S., Van de Weijer, J., Vanrell, M. “Modulating shape features by color attention for object recognition.” International Journal of Computer Vision. 98 (2012) : 49-64.

14. Fu, J., et al. “Dual attention network for scene segmentation.” Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2019) : 3146-3154.

15. Hu, J., Shen, L., Sun, G. “Squeeze-and-excitation networks.” Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) : 7132-7141.

16. Jaderberg, M., Simonyan, K., Zisserman, A. “Spatial transformer networks.” Advances in neural information processing systems. 28 (2015).

17. Woo, S., et al. “Cbam: Convolutional block attention module.” Proceedings of the European conference on computer vision (ECCV). (2018) : 3-19.

18. Li, X., et al. “Mixed attention mechanism for small-sample fine-grained image classification.” 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). (November 2019) : 80-85.

19. Zeiler, M. D. “Hierarchical convolutional deep learning in computer vision.” PhD thesis, New York University. (2014).

20. Nair, V., Hinton, G. E. “Rectified linear units improve restricted boltzmann machines.” ICML. (2010) : 807-814.

21. Jarrett, K., et al. “What is the best multi-stage architecture for object recognition?” ICCV. (2009) : 2146-2153.

22. Hinton, G. E., et al. “Improving neural networks by preventing co-adaptation of feature detectors.” arXiv. (No date provided).

23. Ranzato, M. A., Boureau, Y.-L., Cun, Y. L. “Sparse feature learning for deep belief networks.” Advances in neural information processing systems. (2008).

24. LeCun, Y., et al. “Handwritten digit recognition with a back-propagation network.” Advances in neural information processing systems. (1990).

25. Sermanet, P., Chintala, S., LeCun, Y. “Convolutional neural networks applied to house numbers digit classification.” Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012). (2012).

26. Özdemir, C. “Avg-topk: A new pooling method for convolutional neural networks.” Expert Systems with Applications. 223 (2023) : 119892.

27. Armağan, S., Gündoğan, E., Kaya, M. “Classification of Skin Lesions using Squeeze and Excitation Attention based Hybrid Model of DenseNet and EfficientNet.” 2024 International Conference on Decision Aid Sciences and Applications (DASA). (December 2024) : 1-5.

28. Maurya, A., Chand, S. “Exploiting Pre-trained encoder with receptive fields and squeeze-excitation module for road segmentation.” 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN). (August 2021) : 139-143.

29. Wang, Z., et al. “Enhancing ResNet50 with Depth-wise Separable Convolution Squeeze-and-Excitation and Channel Squeeze-and-Excitation Modules for Accurate Remote Sensing Image Scene Classification.” 2024 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML). (November 2024) : 80-84.

30. Wu, J., et al. “Convolutional neural network with Squeeze and Excitation modules for image blind deblurring.” 2020 Information Communication Technologies Conference (ICTC). (May 2020) : 338-345.

31. Zafar, A., et al. “A comparison of pooling methods for convolutional neural networks.” Applied Sciences. 12(17) (2022) : 8643.

32. Gong, Y., et al. “Quarter Laplacian filter for edge aware image processing.” 2021 IEEE International Conference on Image Processing (ICIP). (September 2021) : 1959-1963.

33. Soydaner, D. “Attention mechanism in neural networks: where it comes and where it goes.” Neural Computing and Applications. 34(16) (2022) : 13371-13385.

34. Deng, J., et al. “Imagenet: A large-scale hierarchical image database.” In 2009 IEEE conference on computer vision and pattern recognition, pp. 248-255. Ieee, 2009.

35. Bhairannawar, S., et al. “Color image enhancement using Laplacian filter and contrast limited adaptive histogram equalization.” 2017 Innovations in Power and Advanced Computing Technologies (i-PACT). (April 2017) : 1-5

36. Prasoetphon, T., et al. “Fuzzy-PID vs. PID Controllers: A Comparative Study on DC Motor Speed Control.” 2024 1st International Conference on Robotics, Engineering, Science, and Technology (RESTCON). (February 2024) : 64-68.

37. Nirthika, R., et al. “Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study.” Neural Computing and Applications. 34(7) (2022) : 5321-5347.

38. Redmon, J., Farhadi, A. “Yolov3: An incremental improvement.” arXiv preprint arXiv:1804.02767. (2018).

39. He, K., et al. “Deep residual learning for image recognition.” Proceedings of the IEEE conference on computer vision and pattern recognition. (2016).

40. Sittiwanchai, T., Khawnuan, U., Yodpijit, N. “Enhancing human activity recognition with lightweight CNN models and integrated blocks.” Science, Engineering and Health Studies. (2024) : 24040001-24040001.

41. Khawnuan, U., Sittiwanchai, T., Yodpijit, N. “Convolutional neural network for wearable fall detection systems.” Science, Engineering and Health Studies. (2023) : 23040002-23040002.

42. Ishii, K., et al. “Report on the 8th tomato-harvesting competition toward smart agriculture.” Proceedings of International Conference on Artificial Life and Robotics. (2023) : 486-490.

43. Matsuo, T., et al. “Toward Smart Tomato Greenhouse The 6th Tomato-Harvesting-Robot Competition and Regulation Changes Aiming at Practical Application.” Journal of Robotics, Networking and Artificial Life. 9(1) (2022) : 13-19.

VITA

Name	Mr. Rut Yatigul
Thesis Title	Development of Deep Neural Networks for Object Detection Systems
Major Field	Production Engineering
Biography	Mr. Rut Yatigul was born on September 2, 2000. He received his vocational certificate in Technical Education (Electrical and Electronics) (English Program) in 2019 and his Bachelor's degree in Robotics Engineering and Automation Systems in 2023 from King Mongkut's University of Technology North Bangkok, Thailand. He is currently a Master's student in the Department of Production and Robotics Engineering at King Mongkut's University of Technology North Bangkok, Thailand.