**Title:Dual Convolutional Neural Networks of Ensemble learning with Attention Mechanism for Classification Task**

**usingAgricultural Pests Image Dataset**

**Abstract**

The rapid increase in global population has led to an escalated demand for food. However, the agricultural sector faces significant challenges due to substantial crop damage caused by pests, resulting in widespread food shortages. In the early stages of agricultural development, identifying these pests poses a considerable challenge. Therefore, ensuring high-quality output in agriculture necessitates accurate pest identification. Traditional methods rely on physical assessments, insect identification, and counting techniques to predict pest infestations. This study constructs pertinent models for classifying pest images: custom model 1 utilizes ResNet residual blocks with ECA attention mechanism while model 2 employs MBConvblocks from Efficientnet combined with SE attention mechanism. The objective is to validate the accuracy of these analytical models for crop pest identification.I collected Agricultural Pests dataset from Kaggle. It contain 12 different agricultural pests inclu ding ants, bees, beetles, earthworms, centipedes, moths and more. I used the above model to classify them and the results showed that the agricultural pests were classified with an accuracy of 95.2% for testing and 98.5% for training. The experimental results show that the model has high accuracy in identifying and classifying agricultural pests.

**Keywords:**CNN;AttentionMechanism;Esemblelearning;ResNet,EfficientNet;ECA;SE;Data argumentation; AgriculturalPests Image Classification.

# **Introduction**

In the agricultural sector, identifying and categorizing crop pests is a crucial task. Insects are not only common and multiply quickly, but they also seriously harm agriculture, lowering crop yields and production. About half of crop yield losses worldwide are caused by pests and agricultural diseases. Consequently, minimizing crop loss and pesticide use requires early insect detection in crops and efficient corrective action. Because insects have intricate organizational systems and a wide variety of characteristics, standard manual categorization methods are laborious and ineffective when it comes to identifying them[1]. Application of computer-assisted categorization techniques is required to help identify and manage pests more effectively, hence increasing the effectiveness of agricultural outputs[2].

Traditional manual identification methods are inefficient and costly and cannot meet the practical needs. With the development of the Internet of Things in agriculture, it has become more convenient to use farmland cameras to obtain pest images. Image recognition techniques based on computer vision can effectively reduce costs and improve recognition speed and efficiency. However, farmland images usually have more background noise, and traditional machine learning methods (e.g., support vector machines and BP neural networks) without feature preprocessing are difficult to achieve high accuracy. Considering the complexity of the farmland environment and the difficulty of feature selection, it is difficult for traditional methods to adapt to real-world environments. Meanwhile, deep learning techniques have been significantly developed in the past two years, providing new possibilities to address these challenges[2].

Deep learning techniques have experienced rapid development in the last two years, especially deep convolutional neural networks (CNNs) have achieved remarkable results in the field of image recognition. From street view recognition to vehicle detection and human motion recognition, CNNs have proven their power. In addition, CNNs have been extended to audio and video recognition and have demonstrated excellent performance in these areas as well [3]. Due to its ability to automatically extract image features, CNN is considered a versatile feature extraction tool and has begun to be applied in the agricultural field, especially for identifying agricultural pests in the farming environment.

This paper proposes a method for classifying agricultural pests that are hard to distinguish with the unaided eye. The method is based on the integration of two convolution neural network  models. The present research uses bespoke models one and two, which are based on the residual structure of the ResNet block and the blocks block of EfficientNet, respectively, to guarantee the high accuracy of the image classification model. Meanwhile, we introduced the SE (Squeeze-and-Excitation)[4] and ECA (Efficient Channel Attention) attention mechanisms to capture visual features more efficiently.We use ResNet's residual structure in Custom Model one to gain a deeper understanding of image features. Custom model two, on the other hand, makes use of EfficientNet's blocks module to accomplish more efficient information learning at various scales. We provide the SE attention mechanism and the ECA attention mechanism to dynamically adapt the network's attention to various channels and spatial information in order to improve the model's perception of significant features. This will help the model better capture essential characteristics.In the end, we improve the overall performance of the model by combining Custom Model one and Custom Model two. By incorporating the advantages of both models, this integrated approach further increases the precision of agricultural pest classification. The method used in this study offers a practical and efficient answer to the issue of agricultural pests that are hard to identify with the human eye. It also serves as a helpful resource for practice and research in related sectors.

The work presented here has resulted in the following significant contributions:

* The introduction of two custom models, a new creative model specifically designed foragricultural pests classification. This model is built upon the ResNet and EfficientNet architecture.To ensure accuracy, the model inherits the advantages of the pre-trained model while being built and customized to the uniqueness of the data set.
* I propose utilizing SE and ECA attention mechanisms. By introducing these attention mechanisms, the model becomes more robust in handling diverse agricultural pest images, thereby improving its classification performance. This approach not only enhances the perception of crucial features but also dynamically adjusts network attention to various dimensions within the image.
* Integration of the two models by setting the weights in an Weighted Average Ensemble method takes full advantage of the two models to achieve better performance, improving robustness and the generalization ability.
* Customized loss and accuracy functions enable the model to handle badly labeled data more effectively and to deliver tighter accuracy metrics that require precise predictions for all categories.

The structure of the remaining sections of this report is as follows. Section 2 provides a brief review of relevant studies. In Section 3, we discuss our proposed framework including the datasets used. In Section 4, we present the results and discussion of our experiments, showcasing the strong performance of our network on datasets and in comparison with other state-of-the-art models. Finally, in Section 5, we conclude our findings,some drawbacks and research dirctionin the future.

# **Related work**

# **2.1 Convolutional Neural Network (CNN)**

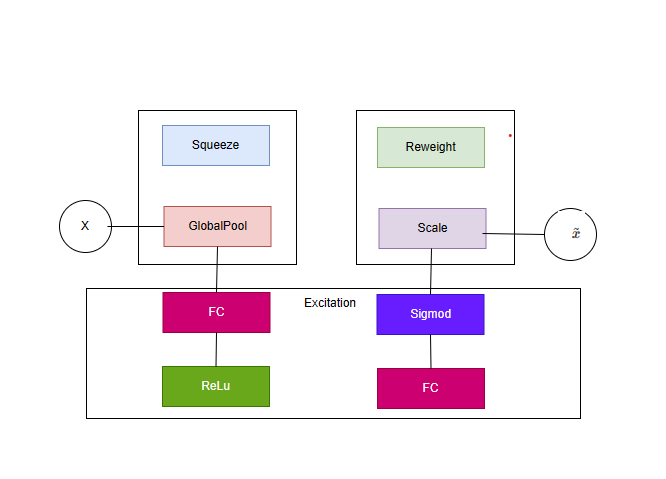
As a subset of deep neural networks, convolutional neural networks (CNNs) are mostly used for visual imagery processing, including picture identification and classification. Convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully linked layers for classification are the architectural elements that define these networks[5]. CNNs efficiently learn spatial feature hierarchies, moving from simple edges to complex patterns, which optimizes the training process by reducing the number of parameters. Their flexibility extends beyond image analysis applications, as they are useful in a variety of fields such as natural language processing, video analysis, and medical diagnosis[6].

# **2.2. Attention Mechanism**

Attention mechanism is an essential and important research direction for optimising neural networks. Attention mechanism in not only does not increase the model complexity, but also through parameter adjustment to enhance the network's attention to the essential features in the data, suppressing the background features in the data. This improves the segmentation accuracy of the prediction results, especially the segmentation accuracy of the details. Existing attention mechanisms such as SENet, BAM, CBAM, ECANetetc[7], in spite of the significant improvement in performance, however, most of the methods nowadays get better performance by increasing the complexity of the attention module, which is why the attention modules created nowadays are becoming more and more complex.

# **2.2.1 SE Attention Mechanism**

Squeezing-and-Excitation (SE) module is  a aspect of channel attention module . The main concept is that each output channel's weight and correlation can be automatically determined by adding a subnetwork to the neural network's internal architecture [8]. Subsequently, the output channels undergo weighting in accordance with the respective weights in order to optimize the crucial information channels. The SE module is organized internally into Squeeze, Congestion, and Reweight sections based on the computing demands of each section.



**Figure 1. SE Attention mechanism architecture**

# **2.2.2 ECA Attention Mechanism**

The concept of the ECA channel attention mechanism was popularized by models such as the Squeeze-Excitation Network (SENet) senet requires relatively complex re-calibration operations[9].ECA proposes a local cross-channel interaction strategy without dimensionality reduction and a method for adaptively determining the size of the one-dimensional convolution kernel, which considerably reduces the complexity of the model while achieving improved performance.ECA typically uses one-dimensional convolutions to generate channel attention maps. This approach is computationally efficient and captures local cross-channel interactions.

# **2.3 Ensemble models**

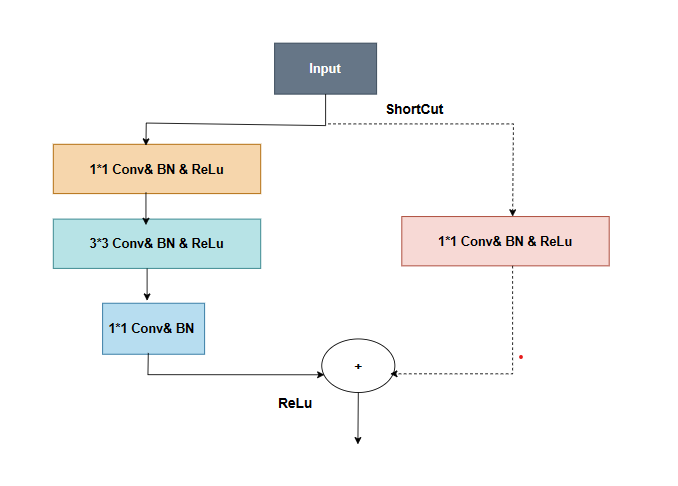
Deep learning models have achieved significant success in a wide range of applications, however, individual deep learning models can have limitations in terms of generalization, robustness and performance. Integrated models can improve the performance of a model by leveraging the strengths of multiple models while compensating for their weaknesses. It improves performance by mitigating the effects of outliers and noisy data, noise can be easily filtered out after the predictions of multiple models[10].

# **2.3.1 Weight average ensemble**

By merging the predictions of several individually trained models and averaging them, average-weighted integration is an effective machine learning integration technique that improves the accuracy and resilience of overall predictions. This approach reduces the risk of overfitting, minimises the bias and variance of individual models, and leverages the strengths of several models. Better performing models can have a greater impact because weights can be set based on model performance. Average weighted integrals are highly adaptable and effective in handling complex classifications. For instance, average-weighted integration has been demonstrated to be a useful technique for enhancing model performance in data science contests and real-world business applications, particularly when dealing with issues that are too difficult or complicated for a single model to manage[11].

# **2.4 Resnet and residual block**

The use of shortcut connection paths or jump connections in ResNets can be effective in mitigating the gradient vanishing problem[12]. This is because error signals can be propagated back to earlier layers more directly and smoothly during backpropagation. Fugure 2 shows a residual identity block which is used when the input and output dimensions are kept constant. On the other hand, the residual convolution block in Figure 3 uses convolutional layers on the shortcut path to change the dimensions so that the dimensions of the inputs and outputs match for subsequent addition operations.

****

**Fugure 2. Residual identity block** **Figure 3. Residual convolutional block**

# **2.5 EfficientNet and MBConv block**

EfficientNets are a family of efficiently scaled convolutional neural networks that have become the leading models for image classification problems.EfficientNets balance the need for accuracy and efficiency by optimising in terms of reducing the model size and decreasing the complexity of the floating-point operations while still maintaining the high quality of the model's performance.At the heart of the MBConv module is the inverted residual structure. Whereas conventional residual blocks make jump connections as the number of channels increases, inverted residual blocks make jump connections as the number of channels decreases. This makes the module lightweight and suitable for resource-constrained environments[13].

# **3. Material andMethod**

## **3.1 Dataset**

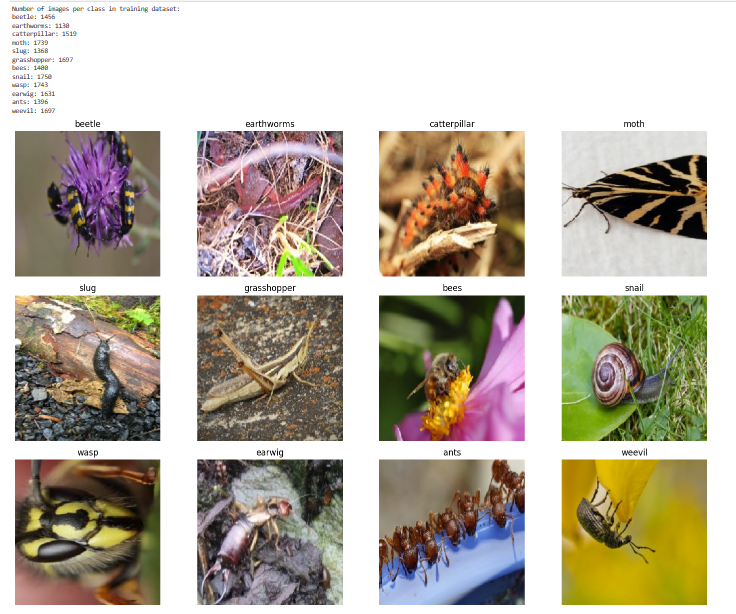
I started by downloading a dataset called Agricultural Pests from the Kaggle website, which contains 12 agricultural pests such as ants, bees, earthworms, and so on. I have put them into the training set, validation set and test set in the ratio of 70%, 15%, 15%. But then during the testing process I found that the results were not very good, then I performed data augmentation. The augmentation of the original test data enables me to evaluate the model's performance on datasets with varying levels of complexity, facilitating its adaptability for application in diverse environments. The detail will be shown below :

## **2.png3.1.1 Original Dataset**

**Figure 4. 12 Samples of different species of agricultural pest in original dataset**

Figure 4 demonstrates a compilation of various insect species and a summary table listing the number of images for each insect class in the training dataset. The table shows a collection of classes such as beetles, caterpillars, moths, slugs, grasshoppers, bees, snails, wasps, earth centipedes, ants, and weevils, with their respective image counts ranging from 227 to 351. Below the table, there are corresponding images for each category, providing a visual reference for the appearance of each insect.

## **3.1.2 Augmented Dataset**

****

**Figure 5. 12 Samples of different speciesof agricultural pest in augmented dataset**

Figure 5 demonstrates a visualization of a dataset for an insect classification task, with a focus on the training dataset. On the left side, there's a summary list that provides the number of images available for each insect category within the training dataset. The classes listed include beetles, earthworms, caterpillars, moths, slugs, grasshoppers, bees, snails, wasps, earwigs, ants, and weevils, with image counts ranging from 1,131 for beetles to 1,607 for weevils.

On the right side of the image, there are representative photographs for each of these insect categories. These images serve as visual samples of the dataset and showcase the diversity of the insect classes. For example, the beetle image shows a beetle on a purple flower, the earthworms are pictured in soil, the caterpillar on a branch, and so forth.

## **3.2 Data processing**

In the data processing, the images are first normalized to ensure their pixel values are scaled to a [0, 1] range, accelerateing the convergence of gradient descent because it ensures that all input features are in a similar range of scales.For the training dataset specifically, a series of data augmentation techniques are applied to artificially expand the diversity of the dataset. This includes random rotations of each image by up to 40 degrees, which helps the model generalize across various orientations. The images are also subject to random translations, both horizontally and vertically, by up to 20% of the image dimensions. This encourages the model to learn features that are spatially invariant, meaning it can recognize objects regardless of their position in the image frame. What’s more each image is randomly sheared, introducing slants to mimic the effect of viewing objects from different angles, and zoomed by a factor of up to 20% to simulate various scales. Horizontal flips are performed to create mirror images, which further trains the model to recognize objects regardless of their orientation.

## **3.3 ProposedModel**

## **3.3.1 Model\_1**

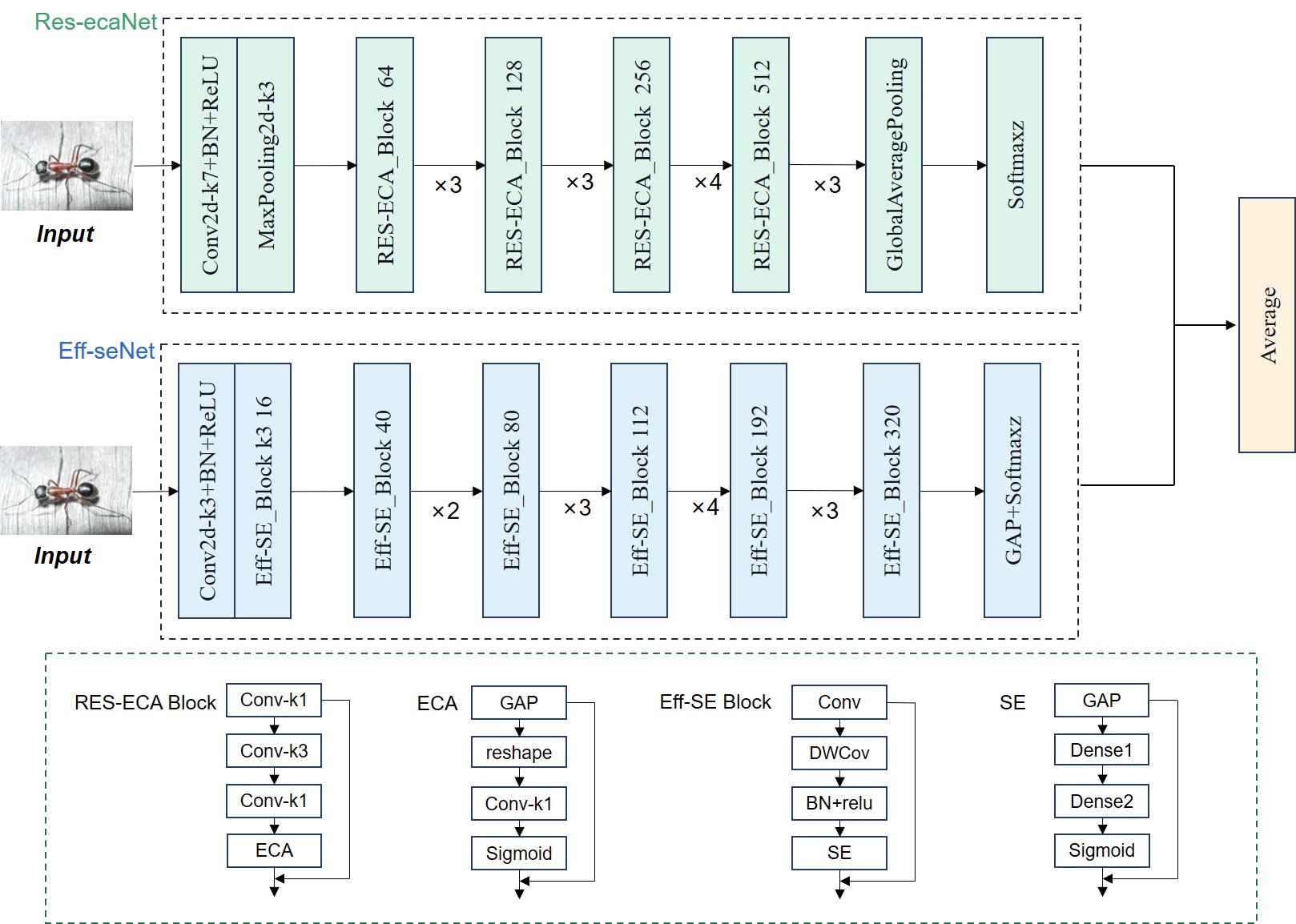
My first custom model is called ResECA, where I combine the residual block with the ECA attention mechanism. The core of the model is the residual block which is a significant structure in ResNet networks. It helps a lot in the gradient vanishing problem by passing the input directly across layers using skip connections. It also makes the model better focus on the region of interest of the image and improves the feature extraction ability of the model. Using 3 \* 3 convolution stacking can enhance multi-scale convolutional features. Each residual block performs a series of convolutions and batch normalisation and ReLU activation of the processed features. ECALayer is simultaneously integrated into the residual block, thus allowing the channel to focus on applying feature mapping. And the channel feature responses are dynamically recalibrated using global average pooling and sigmoid activation functions to generate channel-specific weights. These are used to scale the feature mapping to enhance the representation of the network to focus on more informative features.The model is demonstrated in Figure 6.

## **3.3.2 Model \_2**

My second custom model combines the fusion module with the SE attention mechanism, where the input layer accepts a 224\*224 pixel image. In this model, initial expansion is performed using a 1\*1 convolutional layer, followed by deep convolution for feature map extraction. Subsequently, the SE attention mechanism is applied to adjust the features through another 1\*1 convolutional layer, enabling connection with the original feature map. A total of 15 such modules are stacked together to constitute my proposed model. Below is the detailed description. The network starts with a 2D convolutional layer and uses batch normalisation and is normalised by the ReLU activation function. This stage is preparing for feature extraction. After that it consists of 15 EffBlock blocks. Each EffBlock block k employs depth separable convolution to filter the input efficiently, minimising the computational overhead, and the SE block is existence conditional based on the use\_se parameter. The feature mapping is refined by applying a recalibration mechanism. This is achieved through global average pooling followed by a fully connected layer that outputs channel-specific weights which are then used to scale the original feature mapping. This process enhances the model's ability to focus on the most informative features in the classification task. Throughout the model, the number of filters gradually increases and the spatial resolution decreases, which helps the network to capture more abstract features at higher levels The last layer of the sequence uses global average pooling to reduce the features for each channel to a single descriptive value. These values are then projected onto the dense layers using softmax activation to produce probability distributions for a predefined number of classes. The model is demonstrated in Figure 6.

## **3.3.3 Ensemble\_Model**

I define the input layer to match the expected input shape of (224,224,3), both models get the same input from the input layer, and then average their outputs to create a new input layer. This averaging combines the predictive power of the two models, by reducing the likelihood of overfitting and capturing different patterns learnt by the individual models.The architecture of ensemble model is showed in Figure 6.



**Figure 6 The architecture of the ensemble model**

## **3.3.4 Custom loss and accuracy function**

:The custom 'custom\_categorical\_crossentropy' is the categorical cross-entropy loss with label smoothing. Using the following equation:

(1)

Epsilon is a very small positive number used to avoid numerical instability such as division by zero. When calculating cross-entropy, the value of y\_pred is limited to the range [epsilon, 1 - epsilon]. Apply label smoothing to the true label y\_true. This is done by transferring a portion of the confidence from the correct category to other categories. Correcting the predicted probability is done using the following equation

=clip (2)

Then the categorical cross-entropy was calculated using:

CCE=−∑​×log() (3)

Here the loss is calculated for each sample and then averaged over all the samples.

The custom ‘accuracy’ :

tf.argmax(y\_true, axis=-1) (4)

It finds the category index with the maximum probability for each sample in the true labels. axis=-1 specifies that the operation is performed on the last axis, i.e., to find the maximum in the category probability distribution for each sample.

tf.argmax(y\_pred, axis=-1) (5)

It performs the same operation in the output of the model prediction to find the category index with the highest prediction probability.

tf.equal(tf.argmax(y\_true, axis=-1), tf.argmax(y\_pred, axis=-1)) (6)

It uses tf.equal to compare the indexes of the categories found by both. Returns True if the predicted category is the same as the true category, False otherwise.

tf.reduce\_mean(tf.cast(correct\_predictions, tf.float32)) (7)

This step first converts the Boolean values True (correct predictions) and False (incorrect predictions) to floating point numbers (1.0 and 0.0, respectively). Then, the average of all predictions is calculated using tf.reduce\_mean to get the overall accuracy.

## **3.4 Evaluation Strategy**

These metrics used in this report include include loss, accuracy, ROC-AUC curves, recall, precision, sensitivity, specificity, confusion matrices and f1 scores, and precision recall curves. The metrics are mathematically represented as follows:

|  |  |
| --- | --- |
|  | (4) |
|  | (5) |
| * Sensitivity/ | (6) |
|  | (7)  (8) |
|  |  |
|  |  |

## **3.5. Environment Execution**

## The Python-based models in this study were implemented on a robust Lenovo Legion Y7000P laptop, which is equipped with 32 GB of RAM to efficiently manage large datasets and extensive processing tasks. The system is powered by an Intel® Core(TM) i7-10875H CPU that operates at a base clock speed of 2.30 GHz, capable of handling demanding computational requirements. Complementing the processor, the laptop includes an NVIDIA GeForce RTX 2060 graphics card with 6 GB of dedicated memory, providing excellent GPU acceleration for machine learning and graphic-intensive applications. The laptop's operating system is Windows 11, offering the latest features and security updates. However, due to the size of the dataset I chose to rent a cloud server at Featurize to run it.Finally, the computational setup for the model training is composed of an NVIDIA A4000 GPU with 16.9 GB of dedicated memory, an Intel® Xeon® E5-2680 v4 CPU, 30.1 GB of system RAM, and 451.0 GB of storage capacity. This configuration delivers robust processing capabilities required for complex computational tasks.

# **4.Experimental Results**

This section aims to explain the remarkable results achieved by the models developed in the experiments conducted on the Agricultrual pests dataset. Firstly, a detailed comparative analysis of the classification results on the Agricultrual pests dataset is presented in terms of a single model and a ensemble model to highlight the superior accuracy of the ensemble model. Subsequently, experiments were conducted to explore the impact of various hyperparameters on model performance. These experiments compared different settings of batch size and learning rate to identify the configurations that yield optimal performance in the classification task, while keeping all other conditions constant. The purpose of this paragraph is to describe the remarkable results achieved by the models developed in the experiments conducted on the Agricultrual pests dataset. Firstly, a detailed comparative analysis of the classification results on the Agricultrual pests dataset is presented in terms of a single model and a composite model to highlight the superior accuracy of the composite model. Subsequently, in order to explore the impact of various hyperparameters on model performance, experiments were conducted based on different batch\_size, dropout, weight and learning\_rate to compare hyperparameters with different hyperparameter setting values and to identify the hyperparameter configurations that would yield the optimal performance in the classification task while keeping all other conditions constant. In addition to this, I will also compare the effect of the original and augmented data on the performance of the model. In order to verify the excellent performance of the model in Agricultrual pests classification task, comparison experiments with existing models such as LeNet-5, InceptionresnetV2, ResNet50, etc. have been conducted. The results indicate that the model proposed in this paper demonstrates better classification results on the same dataset compared to these pre-trained models, further confirming its excellent performance on the Agricultural pest dataset.

**\***All tests use the augmented dataset except the part called ‘Ensemble model using original dataset’.

# **4.1.1** **The CNN model\_1**

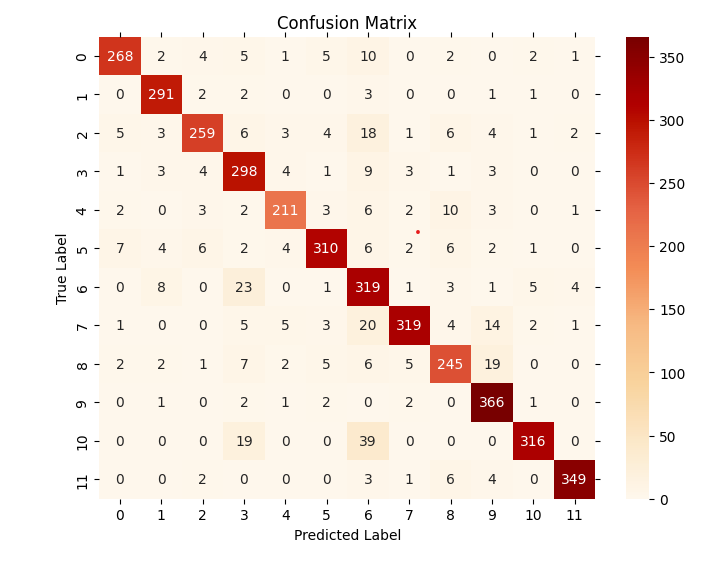
# **A. Accuracy and Loss**

**Figure 7. The accuracy and loss of model\_1**

Figure 7 demonstrates a steady improvement and stabilisation of the training accuracy, indicating that the model is converging reasonably well on the training data. The validation accuracy is also improving at the same time, but stagnates and fluctuates at a level below the training accuracy. As for the losses, both the training and validation losses decrease over time, however, the validation loss shows some fluctuations and is not as stable as the training loss. This shows a mild degree of overfitting, which could be specifically due to the model not generalising well to the validation data.

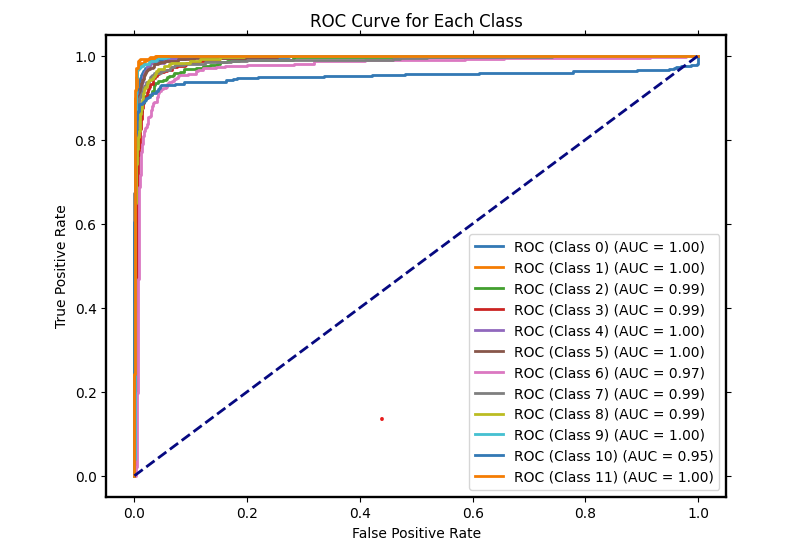
# **B. Confusion Matrix**

Figure 8 shows the Confusion Matrix for this test.Confusion matrix is a tool used to evaluate the performance of a classification model, it shows the number of correct and incorrect classifications, the rows represent the true label and the columns represent the predicted label. The non-diagonal lines show the incorrect classifications. The colours vary from light red to dark red, the darker the colour the larger the number, in this case we want the diagonal to be dark.



**Figure 8. Confusion Matrix of model\_1**

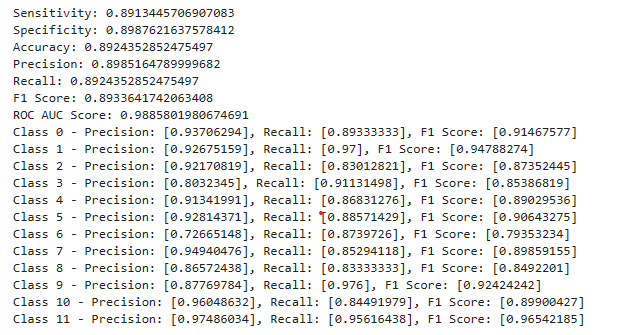
# **C. ROC-AUC Curve**



**Figure 9. ROC-AUC curve of model\_1**

Figure 9 shows ROC-AUC curve of this test, the ROC curve is a graphical representation of the performance of the classification model at different threshold settings.The AUC provides a single value to summarise this performance, with 1.0 representing perfection and 0.5 representing no discriminative power. Overall, this AUC-ROC plot indicates that the model has excellent classification ability for each individual category. The high AUC values for all categories indicate that the model is able to highly discriminate between positive and negative categories, making it a powerful model for this multi-category classification task.

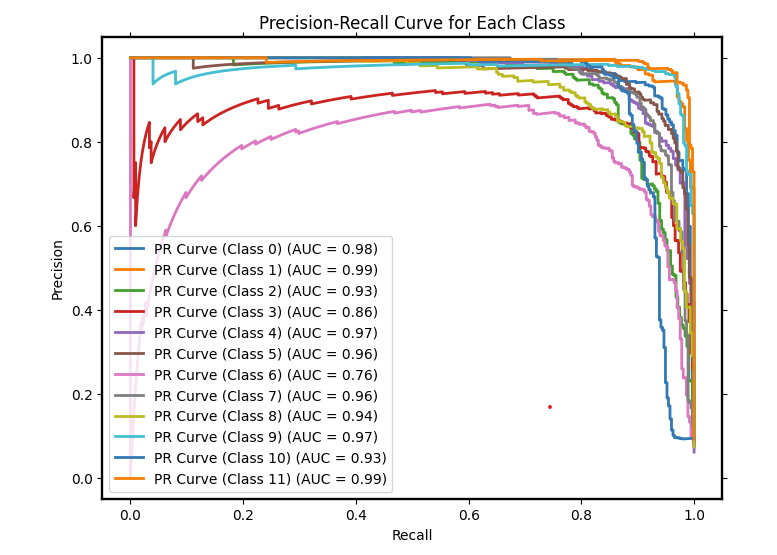
# **D. Sensitivity and Specificity et al.**



**Figure 10. Precision, Recall, F1 Score, Sensitivity, Specificity and ROC-AUC socre of model\_1**

Figure 10 indicates Precision, Recall, F1 Score, Sensitivity, Specificity and ROC-AUC socre of this test .Recall measures the model's ability to correctly identify both positive instances and avoid negative errors, while Precision represents the proportion of accurate predictions within a given category. The F1 score is a harmonized mean of Recall and Precision, providing a single metric that balances both the model's sensitivity and precision. A higher F1 score indicates superior model performance, as it signifies both high accuracy and thoroughness in the model's predictions.

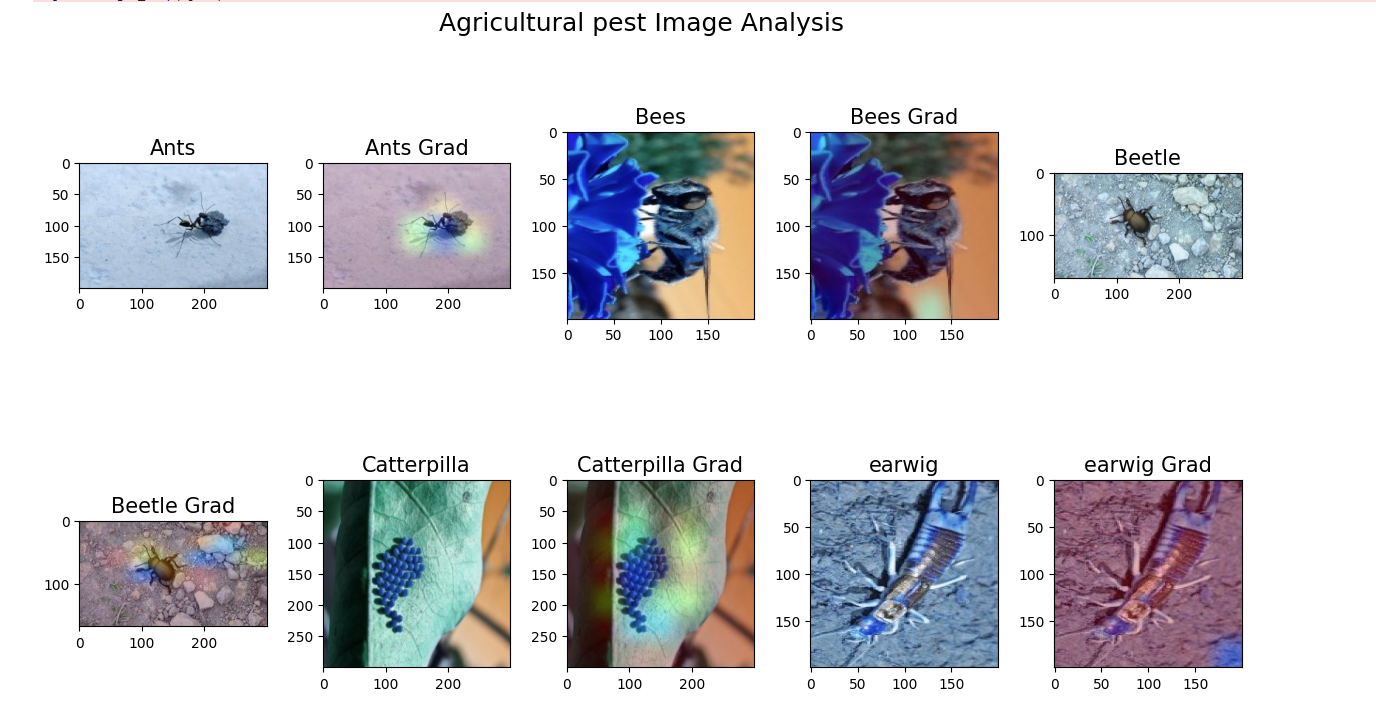
# **E. Precision-Recall Curve**



**Figure 11. Precision-Recall Curve of model\_1**

Figure 11 shows the Precision-Recall Curve of this test. Plotting accuracy (y-axis) versus recall (x-axis) at various threshold values creates the PR curve. A perfect classifier will have 100% recall and 100% precision, or a point at (1,1). The classifier is more accurate and thorough the closer it is to the graph's upper right edge.

# **F. CAM**



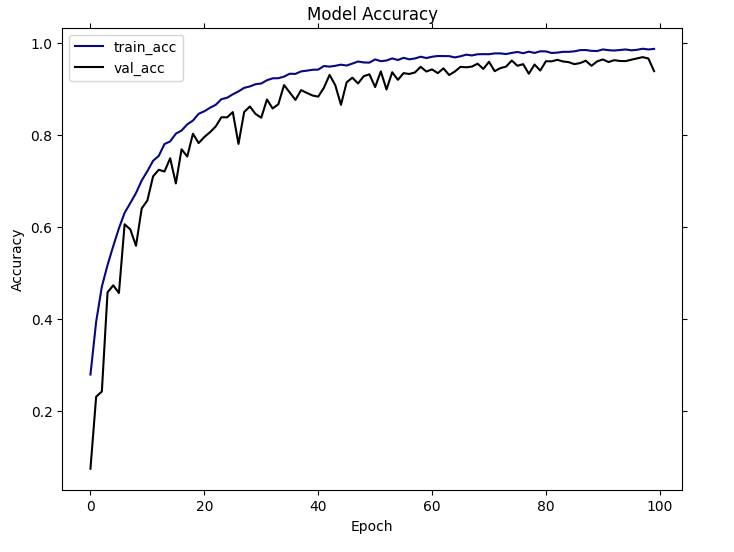
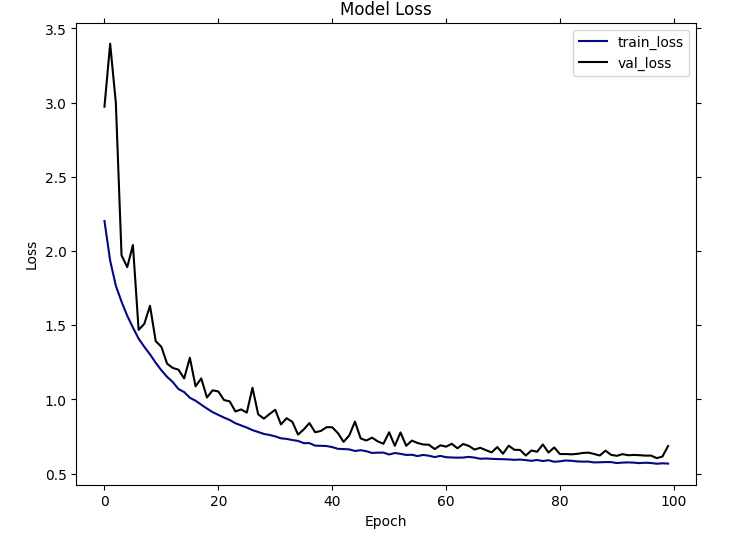
**Figure 12. CAM of model\_1**

Figure 12 shows the CAM chart of Agricultral pests. A gradient map with vibrant colours corresponds to every photograph of a pest. These pictures could be produced using a visualisation method (like Grad-CAM, gradient-weighted class activation mapping) to show the areas that the model concentrates on while identifying or classifying images. Brighter or warmer colours inside these areas suggest that the model gives these areas greater consideration.

# **4.1.2** **The CNN model\_2**

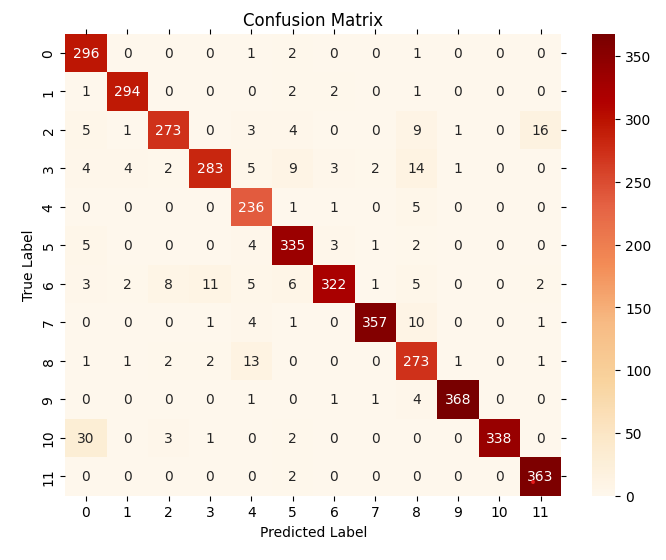
# **A. Accuracy and Loss**

The accuracy and loss of the model\_2 is shown in Figure 13, which training accuracy reach 98.64%

and validation accuracy reach 93.5%.

**Figure 13. Accuracy and Loss of model\_2**

# **B. Confusion Matrix**

The confusion matrix of the model\_2 is demonstrated in Figure 14.

**Figure 14. Confusion Matrix of model\_2**

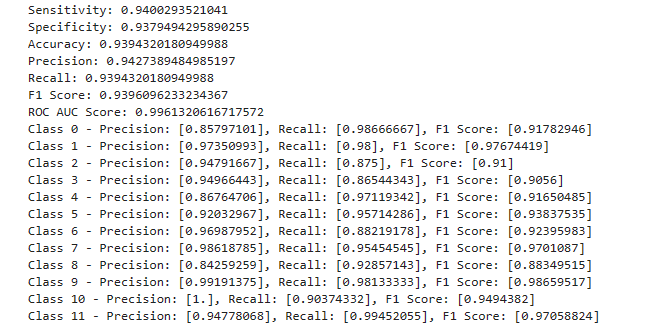
# **C. ROC-AUC Curve**

The ROC-AUC Curve of the model\_2 is demonstrated in Figure 15.

**Figure 15. ROC-AUC Curve of model\_2**

# **D. Sensitivity and Specificity et al.**

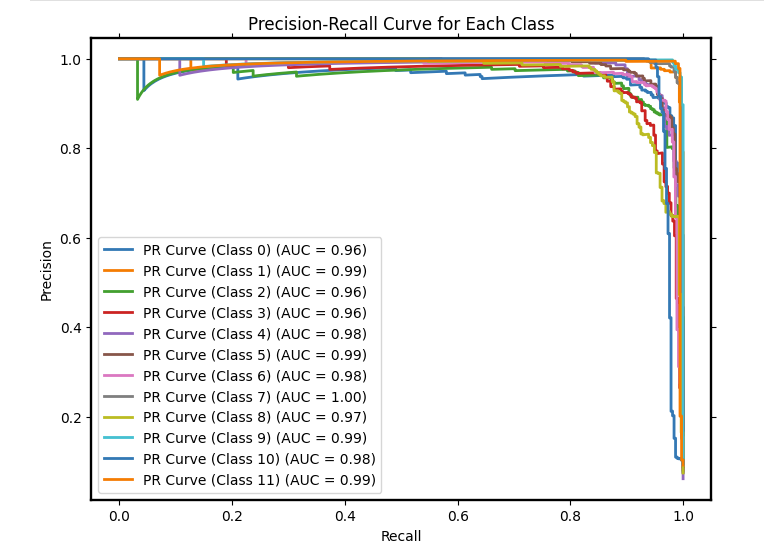
The Precision, Recall, F1 Score, Sensitivity, Specificity and ROC-AUC socre of the model\_2 is demonstrated in Figure 16.



**Figure 16. Precision, Recall, F1 Score, Sensitivity, Specificity and ROC-AUC socre of of model\_2**

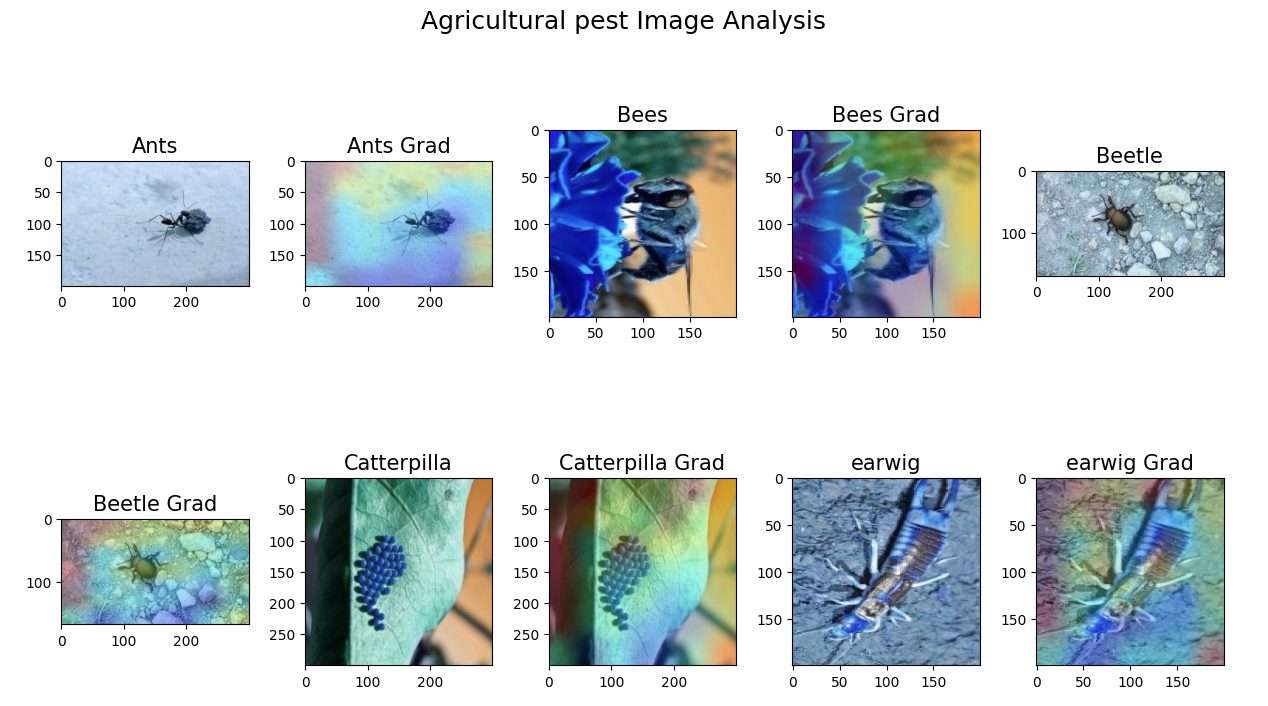
# **E.** **Precision-Recall Curve**

The Precision-Recall Curve of the model\_2 is demonstrated in Figure 17.



**Figure 17. Precision-Recall Curve of model\_2**

# **E. CAM**



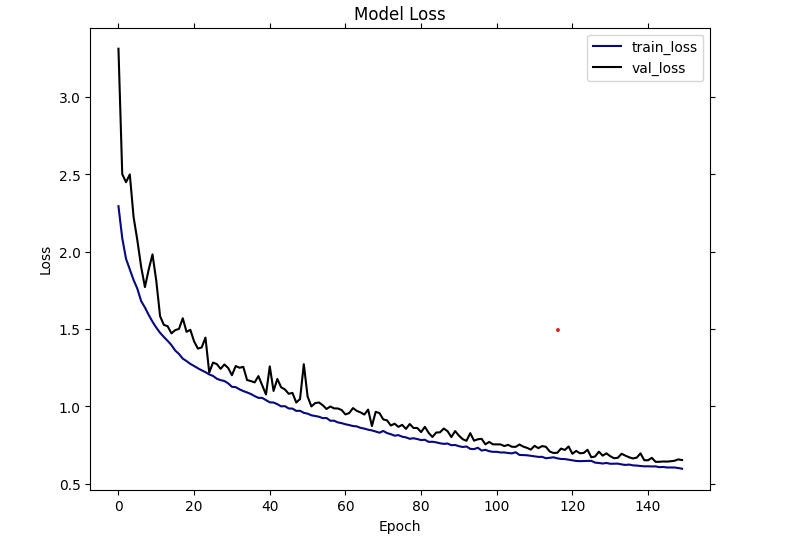
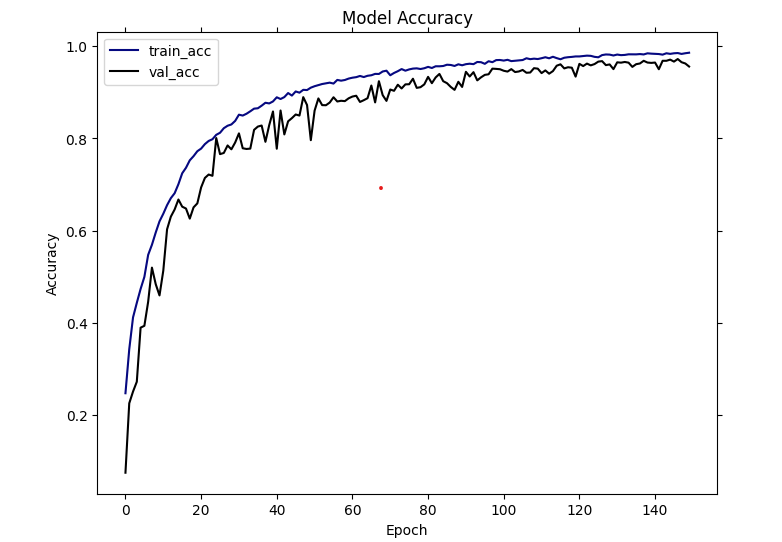
**Figure 18. CAM of model\_2**

# Figure 18 shows the CAM chart of Agricultral pests of model\_2.

# **4.1.3.** **The ensemble\_model**

# **A. Accuracy and Loss**

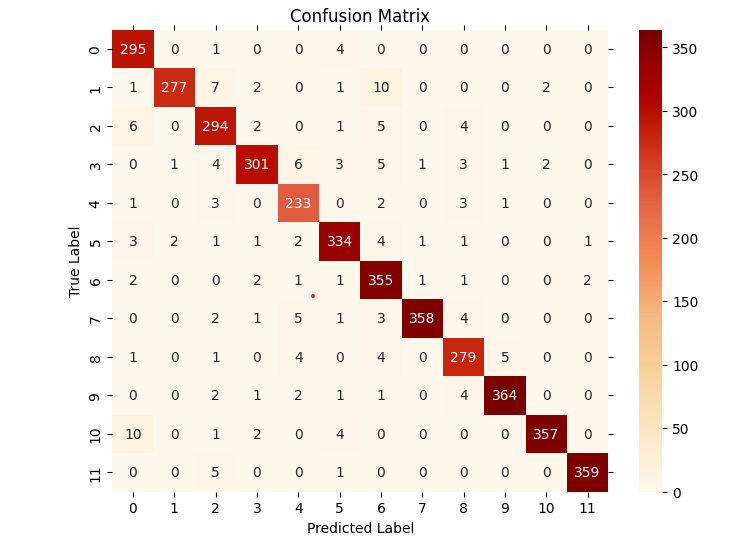
The accuracy and loss of the ensemble model is shown in Figure 19, which training accuracy reach 98.57%

and validation accuracy reach 95.2%.

**Figure 19. Accuracy and Loss of ensemble model**

# **B. Confusion Matrix**

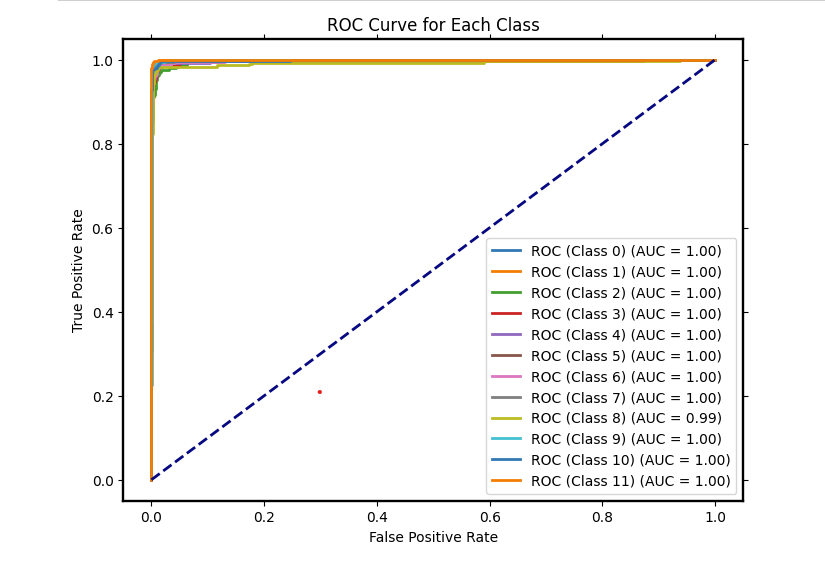
The confusion matrix of the ensemble model is demonstrated in Figure 20.

****

**Figure 20. Confusion Matrix of ensemble model**

# **C. ROC-AUC Curve**

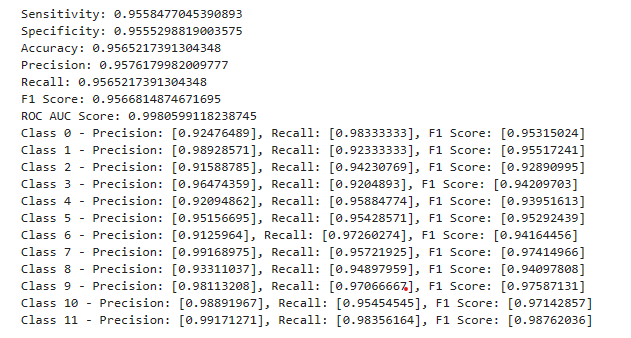
The ROC-AUC curve of the ensemble model is demonstrated in Figure 21.



**Figure 21. ROC-AUC Curve of ensemble model**

# **D. Sensitivity and Specificity et al.**

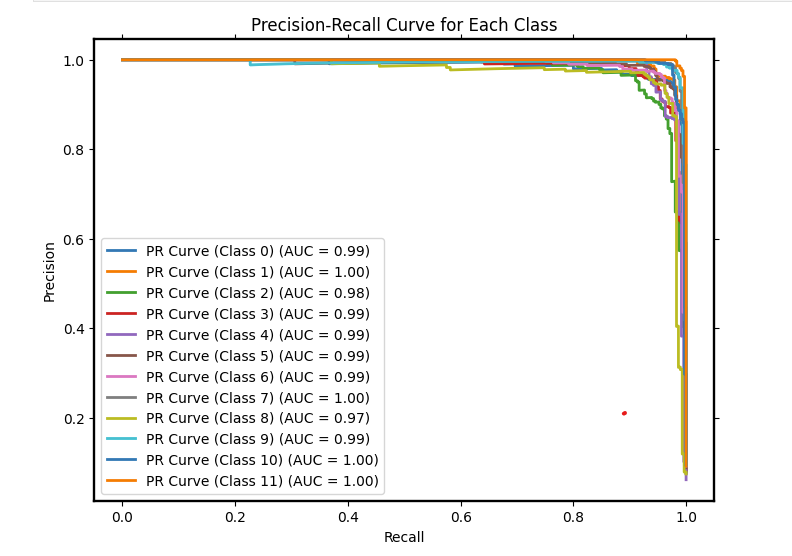
The Precision, Recall, F1 Score, Sensitivity, Specificity and ROC-AUC socre of the ensemble model is demonstrated in Figure 22.

****

**Figure 22. Precision, Recall, F1 Score, Sensitivity, Specificity and ROC-AUC socre of ensemble\_model**

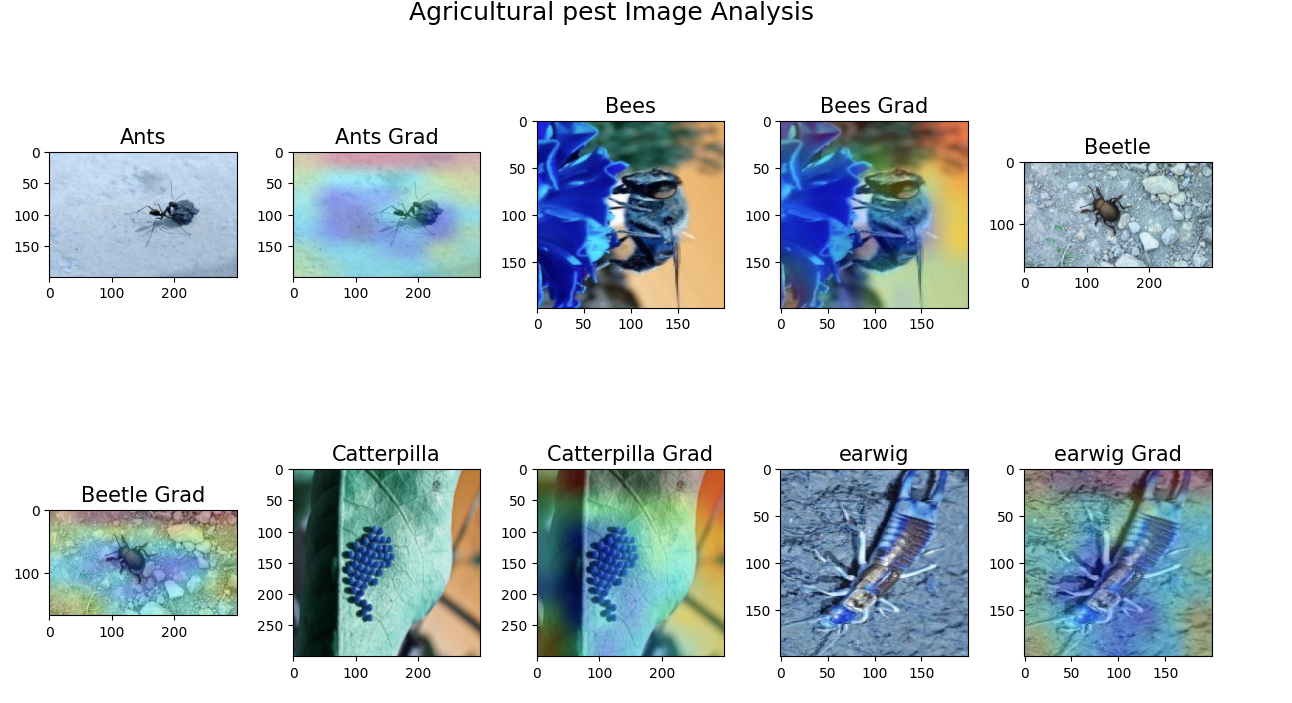
# **E. Precision-Recall Curve**

The Precision-Recall Curve of the ensemble model is demonstrated in Figure 23.



**Figure 23. Precision-Recall Curve of ensemble model**

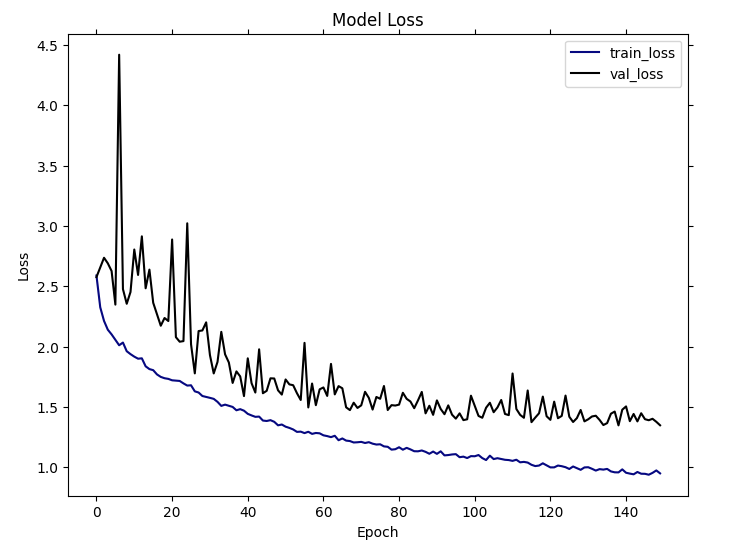
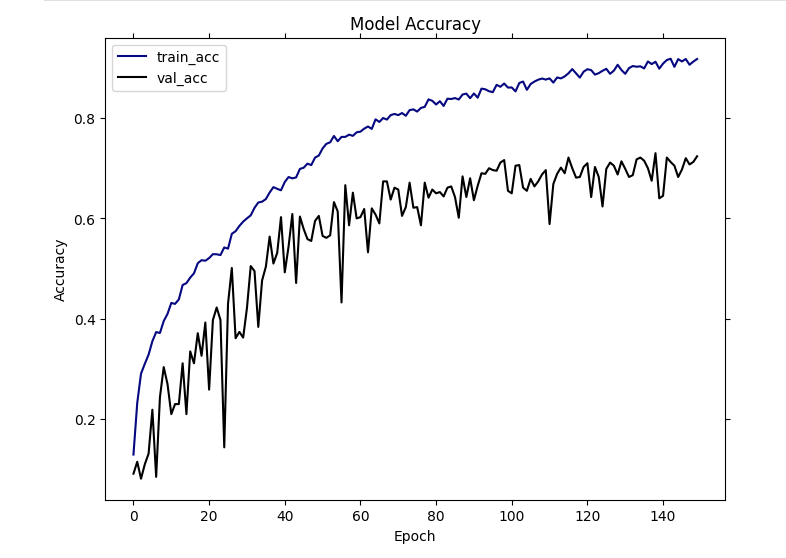
# **E. CAM**

****

**Figure 24. CAM of ensemble model**

# **4.1.4.** **Ensemble model using original dataset**

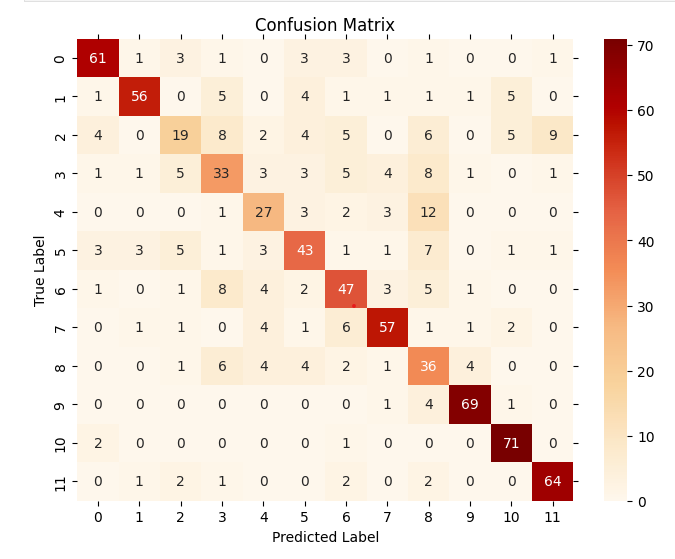
# **A. Accuracy and Loss**

The Precision-Recall Curve of the ensemble model in this test is demonstrated in Figure 25.

**Figure 25. Accuracy and Loss of ensemble model using original dataset**

# **B. Confusion Matrix**

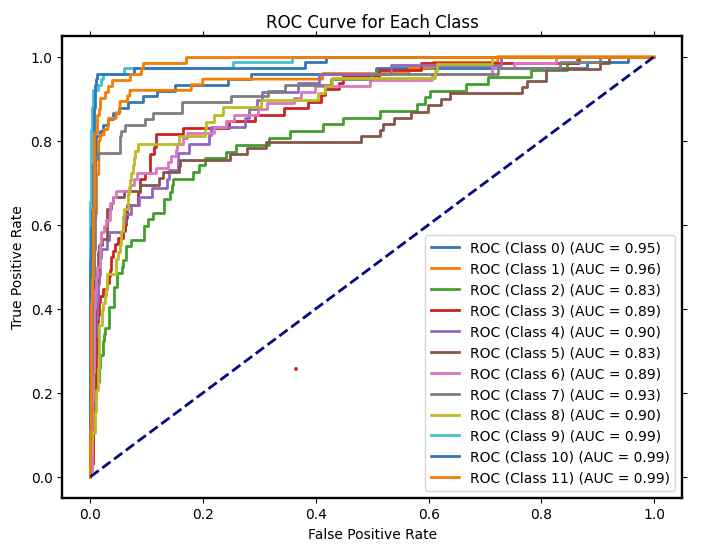
The Confusion Matrix of the ensemble model in this test is demonstrated in Figure 26.



**Figure 26. Confusion Matrix of ensemble model using original dataset**

# **C. ROC-AUC Curve**

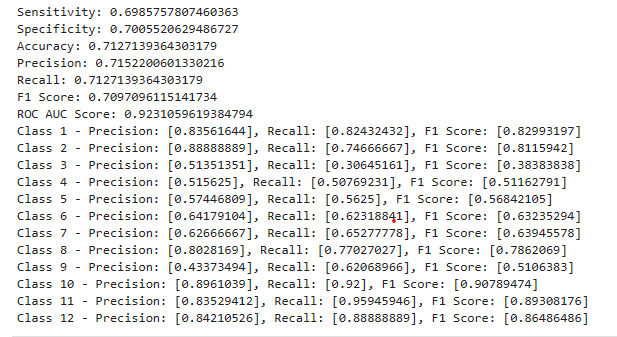
The ROC-AUC Curve of the ensemble model in this test is demonstrated in Figure 27.



**Figure 27. ROC-AUC Curve of ensemble model using original dataset**

# **D. Sensitivity and Specificity et al.**

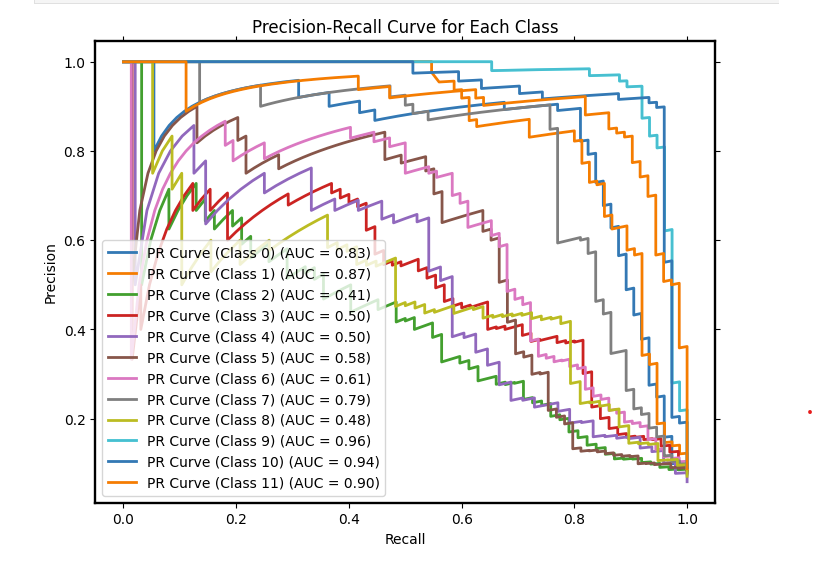
The Precision, Recall, F1 Score, Sensitivity, Specificity and ROC-AUC socre of the ensemble model in this test is demonstrated in Figure 28.



**Figure 28. Precision, Recall, F1 Score, Sensitivity, Specificity and ROC-AUC socre of ensemble model using original dataset**

# **E.** **Precision-Recall Curve**

The Precision-Recall Curve of the ensemble model in this test is demonstrated in Figure 29.



**Figure 29. Precision-Recall Curve of ensemble model using original dataset**

**4.2.** **Performance Results using the dataset**

**4.2.1 Performance Results with different batch\_size**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Batch\_size** | **accuracy** | **loss** | **recall** | **precision** | **F1-score** | **sensitivityi** | **specificity** | **ROC-AUC score** |
| **8** | **0.8143** | **1.3562** | **0.8116** | **0.8169** | **0.8101** | **0.8145** | **0.8134** | **0.8592** |
| **16** | **0.9126** | **0.8257** | **0.9133** | **0.9114** | **0.9139** | **0.9151** | **0.9102** | **0.9421** |
| **32** | **0.9565** | **0.6592** | **0.9562** | **0.9576** | **0.9566** | **0.9558** | **0.9555** | **0.9908** |

**Table 1. The performance with different batch\_size**

Table 1 shows accuracy, loss, recall,precision,F1-score,sensitivity, specificity and ROC-AUC score with different batch\_size. According to the table, it is clear that 32 is the best batch\_size for this model.

**4.2.2 Performance Results with data augmentation dataset for training and original dataset**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **dataset** | **accuracy** | **loss** | **recall** | **precision** | **F1-score** | **sensitivityi** | **specificity** | **ROC-AUC score** |
| **original** | **0.7127** | **1.3562** | **0.7127** | **0.7152** | **0.7097** | **0.8145** | **0.7005** | **0.9231** |
| **augmented** | **0.9565** | **0.6592** | **0.9562** | **0.9576** | **0.9566** | **0.9558** | **0.9555** | **0.9908** |

**Table 2. The performance with different dataset**

The ROC-AOC Curve, Precision-Recall Curve and confusion matrix of both them is showed in Figure 23,24,26 and Figure 18,19,21.

## **4.3. Discussion**

## **4.3.1 Comparision between the ensemble model using orginal and augmented dataset**

## **Result comparision and analysis:**

With the enhanced dataset, the test set accuracy increased from 72% to 95%, while the overfitting problem was well overcome. The loss have decreased at the same time.

Comparing their confusion matrices shows a higher number of correct predictions and fewer misclassifications per class. The distribution and degree of misclassification in the confusion matrix of the model for the orginal dataset is more chaotic. In contrast, the confusion matrix for the model with the augmented dataset indicates that the model has a stronger discriminative ability for the classes.The pronounced contrast between the darker diagonal cells and the lighter off-diagonal cells in the confusion matrix suggests that the model exhibits superior generalization capabilities.

It is clear that the ROC Curves have perfect AUC scores of 1.0 for all classes, which can be recognized as an exceptional level of classification performance after using the augmented dataset.

By comparing the Precision-Recall (PR) curves plot, it is very noticeable that the AUC scores of the model using the augmented dataset are closer to the upper right hand corner of the curve, which suggests that the model maintains a high level of precision at all recall levels. In contrast, the AUC scores of the models using the original dataset vary significantly, demonstrating the variability of model performance across classes.

The overall sensitivity, specificity, accuracy, and F1 score are moderate to high, with sensitivity at approximately 0.69, specificity at 0.70, accuracy at 0.71, F1 score at 0.71 and ROC AUC score is very at 0.92 for ensemble model using original dataset. As for ensemble model using augmented dataset, the overall sensitivity, specificity, accuracy, precision, recall, and F1 score are all very high, close to or above 0.95, indicating excellent model performance across these metrics. The ROC AUC score is nearly perfect at 0.99.

## **Summary:**

This is because I did the data augmentation such as rotated, scaled, clipped, etc., which enhances the diversity of the training samples and can help the model to learn more generalised features rather than memorising data samples. In addition to this, this also brings it closer to the real world. This is because there are many images in the real world that vary depending on factors such as the angle at which they were taken or the intensity of the light. This enhancement in model generalization consequently bolsters the model's applicability and performance in real-world scenarios.

In order to be able to obtain more detailed features the first convolutional neural network consists of 13 residual blocks plus the ECA attention mechanism, with 3 more convolutional layers in each residual block. Not only does each convolutional layer have a certain number of parameters, but the convolutional layers in the residual connections also increase the number of parameters.The ECA attention mechanism also has additional parameters, so the number of parameters for the whole model is not small. The second model uses a mixture of 1x1 convolution and deepdepthwise convolutions However, due to the depth of the network, the number of parameters is larger, especially due to the extension and projection layers in each block. The depth of the model greatly increases its complexity. After ensembling the two models into one, model becomes more complex. So we need more data for effective training to help the model learn.when we need to train such a complex model from scratch . That’s also reason why I will use augmented dataset.

## **4.3.2 Construction process of model\_1**

I made many attempts before building out this final model. Initially I used a sequential model containing 5 convolutional layers and used the batch normalisation max pooling plus SE attention mechanism along with fully connected layers. But it didn't work very well and the accuracy of the training stayed constant over time at around 15%. I thought that the model was not complex enough to learn the features of the image, and then I tried to customise the InceptionNet plus SE attention mechanism. Upon integrating InceptionNet into the architecture, I observed a substantial enhancement in accuracy, reaching as high as 83%. However, this improvement comes at the cost of increased model complexity,attributable to the depth of InceptionNet.In order to be compatible with the two factors of complexity and performance, I used the residual block. Although the depth will increase the complexity, the jump connection can improve the overall efficiency. In addition to this, the problem of vanishing gradients can also be solved. After that, I found that the number of parameters of the model is relatively large, and I used a lightweight attention mechanism-ECA attention mechanism, which avoids manipulating the dimensionality of the channel by processing the channel directly through a one-dimensional convolutional kernel. This reduces the number of parameters and computational complexity while also providing high computational efficiency.

## **4.3.3 Construction process of model\_2**

Based on the first model, I tried to customise EfficientNet\_b7 combined with SE attention mechanism. Despite its complexity and the considerable computational resources it demands, my efforts to integrate this model with model\_1 and experiment with various weighting schemes, such as (0.3, 0.7), (0.4, 0.6), and (0.2, 0.8), were not successful in effectively amalgamating the strengths of both models. So I used the MBConv block from EfficientNet instead, along with the SE attention mechanism, and I discovered that the results greatly improved the training accuracy to 98% without overfitting. At this point I wanted to use the same idea I used in building the first model, which is to replace the SE module with the ECA attention mechanism to reduce the model parameters. But the performance did not significantly improve the effect of what. Therefore, the final model was built as MBConv blocks with SE attention mechanism.

## **4.3.4 Process of ensembling model**

Integration methods include voting method, average additive method and stacking method, I used average additive method because both of them have similar characteristics and both of them have high performance so average ensemble is a good choice. Subsequently, I conducted experiments with different weight combinations, including (0.3, 0.7), (0.4, 0.6), and (0.2, 0.8), to assess their impact on performance. Through this rigorous testing process, it became evident that an equal weighting of (0.5, 0.5) for both models yielded the most effective performance, optimally leveraging the strengths of each model in the ensemble.

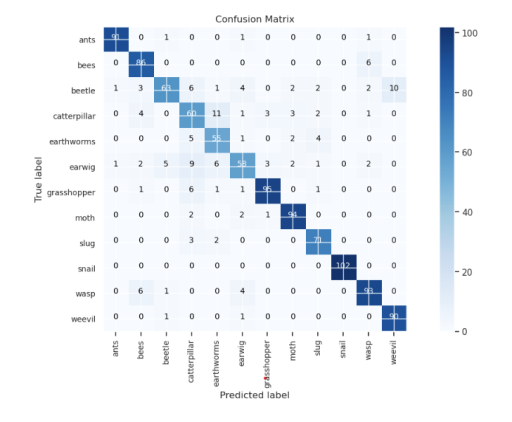
## **4.3.5 Found of learning rate**

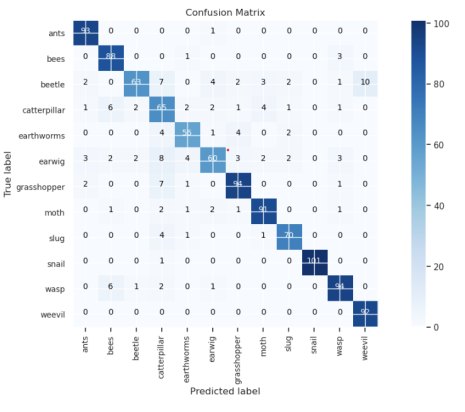
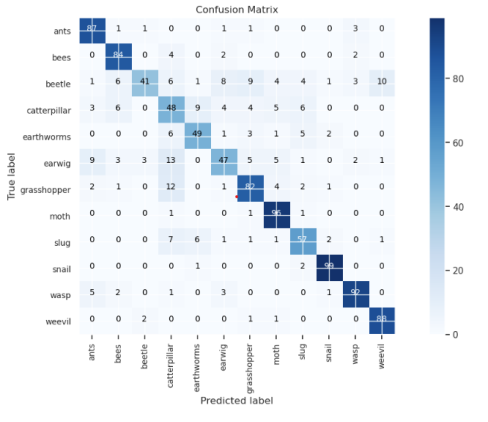
The fixed learning rate did not exhibit stable convergence to the optimal value during the later stages of my training, necessitating manual fine-tuning to identify the optimal learning rate, a process that proved excessively cumbersome. Consequently, I employed an exponential decay learning rate adjustment strategy for achieving adaptability. Specifically, I set the step size to 10000 and the decay rate to 0.9. This approach enables a balanced exploration between high and low learning rates: while a higher learning rate facilitates rapid exploration, a lower one aids in refining weight tuning. As a result, this enhances both stability and convergence during model training.

## **4.4 Fair comparison Withothe Deep Learning Models**

To exemplify the superiority of my model, I compared this model to MobileNet, InceptionV3 and ResNet50. This is represented by the evaluation metrics in Table 3.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | accuracy | loss | recall | precision | F1-score | sensitivity | specificity | ROC-AUC SCORE |
| ResNet50V2 | 0.7916 | 0.70892 | 0.7916 | 0.7949 | 0.7846 | 0.7925 | 0.7751 | 0.8512 |
| InceptionresnetV2 | 0.8717 | 0.3998 | 0.8700 | 0.8700 | 0.8614 | 0.8516 | 0.8754 | 0.9265 |
| EfficientNet-V2 | 0.8799 | 0.3991 | 0.8851 | 0.8894 | 0.8736 | 0.8709 | 0.8822 | 0.9337 |
| My model | 0.9565 | 0.6592 | 0.9562 | 0.9576 | 0.9566 | 0.9558 | 0.9555 | 0.9908 |

**Table 3. Matrix of coparision between my model and pre-trained model**



**Figure 30. Confusion Matrix of EfficientNetV2, ResNet50V2 and InceptionResNetV2**

I've developed a model that exhibits superior performance compared to established architectures such as ResNet50V2, InceptionresnetV2, and EfficientNet-V2, particularly across metrics like accuracy, precision, recall, F1 score, sensitivity, specificity, and ROC-AUC scores. The key to this advancement lies in the strategic incorporation of both ECA and SE attention mechanisms. These modules effectively deepen the network's focus on salient features, enabling a nuanced adaptation at the channel level without significantly burdening computational resources.

The ECA mechanism, specifically, sharpens the model's focus during training, enhancing the discriminative power of features while concurrently streamlining the parameter space. Additionally, the dense residual connections within the model are crucial for preserving detail-rich representations. The fusion of these elements with the SE attention mechanism, executed through depthwise separable convolutions, not only preserves but augments the model's performance. This synergy allows for the capture of complex patterns with a relatively modest computational footprint, embodying the essence of efficient and deep feature learning.

But my model is likely to be more complex and take longer to train compared to them, so limited resources are a drawback.

## **4.5 Indirect comparison With Existing Literature**

Litureatures[14] ‘s model makes use of a modified version of the ResNet-50 deep convolutional neural network. The idea of transfer learning is incorporated, and ImageNet's pre-trained weights are used. The technique increases computing efficiency and accuracy of feature extraction while automating the process. With 23,989,124 parameters and layers designed for feature extraction, the enhanced ResNet-50 leverages maximum pooling to minimise input size and thus lower computational costs. Because of its architecture's jump connections, which address the issue of vanishing gradients, deep learning models are easier to train.Compared to my model my model has less parameter which is more easy to compute, but in the future I could draw some advantages from the transfer learning in this model.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Literatures | accuracy | loss | recall | precision | F1-score | sensitivity | specificity |
| Shahet al[14]. | 0.9681 | 0.001489 | x | x | x | x | x |
| A. P. P and S. E. N[15] | 0.963 | x | 0.96 | 0.97 | 0.96 | x | x |
| Ullah et al.[16] | 1.0 | x | 1.0 | 1.0 | 1.0 | x | x |
| Y et al[17]. | 0.8626 | 0.54 | 0.8547 | 0.8621 | x | x | x |
| Q et al[18]. | 0.9597 | 0.2154 | 0.9469 | 0.9512 | 0.96.1 | x | x |
| My model | 0.9565 | 0.6592 | 0.9562 | 0.9576 | 0.9566 | 0.9558 | 0.9555 |

**Table 4.Compare with related literature**

# **5.Conclusion**

This paper presents an integrated model based on convolutional neural networks and SE,ECA attention mechanisms that shows high results in the task of classifying agricultural pests from different backgrounds. The ensemble model integrates two distinct convolutional neural networks with disparate structures to extract localized features from the image. In this model, a combination of SE attention mechanism and ECA attention mechanism is employed, wherein the SE attention mechanism focuses on recalibrating channel feature responses to enhance overall model performance.This enables the network to prioritize more informative features, thereby allowing the model to focus on valuable channel information and enhancing feature representation by capturing lightweight cross-channel interactions. The ECA method avoids dimensionality reduction and simplifies model complexity by resizing kernels, leading to significant improvements in performance across various visual tasks and enhancing the discriminative power of the model. Additionally, this paper employs dataset augmentation techniques to diversify training data, enabling adaptation of the model for different environments. This approach offers a convenient solution for real-life identification and classification of agricultural pests, facilitating targeted spraying strategies for farmers. However, it should be noted that this proposed method has certain limitations such as high computational complexity resulting in longer training times due to limited resources. Future research should focus on simplifying the model architecture to improve training efficiency and reduce overall time requirements.

**References**

[1] G. Rajeswarappa, S. Depuru, and S. Sirisala, ‘Crop Pests Identification based on Fusion CNN Model: A Deep Learning’, in *2023 8th International Conference on Communication and Electronics Systems (ICCES)*, Jun. 2023, pp. 968–974. doi: 10.1109/ICCES57224.2023.10192693.

[2] ‘Pest identification via deep residual learning in complex background’, *Comput. Electron. Agric.*, vol. 141, pp. 351–356, Sep. 2017, doi: 10.1016/j.compag.2017.08.005.

[3] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, ‘Beyond short snippets: Deep networks for video classification’, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 4694–4702. doi: 10.1109/CVPR.2015.7299101.

[4] W. Xia, F. Yu, H. Wang, and R. Hong, ‘A high-precision lightweight smoke detection model based on SE attention mechanism’, in *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, Jan. 2022, pp. 941–944. doi: 10.1109/ICCECE54139.2022.9712739.

[5] R. R. Rajalaxmi, E. Gothai, S. V, S. Vignesh, and T. Varun, ‘Vision based Fall Detection using Optimized Convolutional Neural Network’, in *2022 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2022, pp. 01–06. doi: 10.1109/ICCCI54379.2022.9740859.

[6] R. Surya and E. Gautama, ‘Cassava Leaf Disease Detection Using Convolutional Neural Networks’, in *2020 6th International Conference on Science in Information Technology (ICSITech)*, Oct. 2020, pp. 97–102. doi: 10.1109/ICSITech49800.2020.9392051.

[7] Y. He, S. Zhou, and X. Quan, ‘Remote Sensing Image Scene Classification Based on ECA Attention Mechanism Convolutional Neural Network’, in *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, Oct. 2022, pp. 1265–1269. doi: 10.1109/ICCASIT55263.2022.9987089.

[8] X. Fan, Y. Wang, J. Liu, Z. Yin, and Z. Yang, ‘Agricultural pest detection algorithm based on attention mechanism’, in *2022 IEEE 22nd International Conference on Communication Technology (ICCT)*, Nov. 2022, pp. 1709–1713. doi: 10.1109/ICCT56141.2022.10072957.

[9] W. Yang and H. Li, ‘Continuous Sign Language Recognition Method Based on ECA and Time Attention Mechanism’, in *2023 IEEE 6th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, Aug. 2023, pp. 480–484. doi: 10.1109/PRAI59366.2023.10332074.

[10] S. Abimannan, E.-S. M. El-Alfy, Y.-S. Chang, S. Hussain, S. Shukla, and D. Satheesh, ‘Ensemble Multifeatured Deep Learning Models and Applications: A Survey’, *IEEE Access*, vol. 11, pp. 107194–107217, 2023, doi: 10.1109/ACCESS.2023.3320042.

[11] U. Vignesh and B. S. Chokkalingam, ‘EnC-SVMWEL: Ensemble Approach using CNN and SVM Weighted Average Ensemble Learning for Sugarcane Leaf Disease Detection’, in *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, Mar. 2023, pp. 1663–1668. doi: 10.1109/ICSCDS56580.2023.10104818.

[12] U. Archana, A. Khan, A. Sudarshanam, C. Sathya, A. K. Koshariya, and R. Krishnamoorthy, ‘Plant Disease Detection using ResNet’, in *2023 International Conference on Inventive Computation Technologies (ICICT)*, Apr. 2023, pp. 614–618. doi: 10.1109/ICICT57646.2023.10133938.

[13] V.-T. Hoang and K.-H. Jo, ‘Practical Analysis on Architecture of EfficientNet’, in *2021 14th International Conference on Human System Interaction (HSI)*, Jul. 2021, pp. 1–4. doi: 10.1109/HSI52170.2021.9538782.

[14] D. Shah, R. Gupta, K. Patel, D. Jariwala, and J. Kanani, ‘Deep Learning based Pest Classification in Soybean crop using Residual Network-50’, in *2022 IEEE 2nd International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, Dec. 2022, pp. 1–5. doi: 10.1109/iSSSC56467.2022.10051424.

[15] A. P. P and S. E. N, ‘Paddy Pest Classification Using Deep Learning Based Features’, in *2022 Second International Conference on Next Generation Intelligent Systems (ICNGIS)*, Jul. 2022, pp. 1–5. doi: 10.1109/ICNGIS54955.2022.10079825.

[16] N. Ullah, J. A. Khan, L. A. Alharbi, A. Raza, W. Khan, and I. Ahmad, ‘An Efficient Approach for Crops Pests Recognition and Classification Based on Novel DeepPestNet Deep Learning Model’, *IEEE Access*, vol. 10, pp. 73019–73032, 2022, doi: 10.1109/ACCESS.2022.3189676.

[17] Y. Song, X. Duan, Y. Ren, J. Xu, L. Luo, and D. Li, ‘Identification of the Agricultural Pests Based on Deep Learning Models’, in *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, Nov. 2019, pp. 195–198. doi: 10.1109/MLBDBI48998.2019.00044.

[18] Q. Dai, X. Cheng, Y. Qiao, and Y. Zhang, ‘Agricultural Pest Super-Resolution and Identification With Attention Enhanced Residual and Dense Fusion Generative and Adversarial Network’, *IEEE Access*, vol. 8, pp. 81943–81959, 2020, doi: 10.1109/ACCESS.2020.2991552.