Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Improving CNN-Based Pest Recognition with a Post-Hoc Explanation of XAI

Ching-Ju Chen
National Yunlin University of Science and Technology
Ling-Wei Chen
National Cheng Kung University
Chun-Hao Yang
National Cheng Kung University
Ya-Yu Huang
National Cheng Kung University
Yueh-Min Huang (huang@mail.ncku.edu.tw)
National Cheng Kung University

Research Article

Keywords: Explainable Artificial Intelligence (XAI), Artificial Intelligence, Post-Hoc Explanation, eXplanation with Ranked Area Integrals (XRAI), Deep Learning, Pattern Features Visualization

Posted Date: August 26th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-782408/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Abstract

Deep learning is currently quite prevalent and is often used in image classification or object detection. This article adds emerging research on the use of explainable AI (XAI) in *Tessaratoma papillosa* pest identification and investigates the connotation and importance of XAI, interpretability classification standards, and neural network interpretation methods and compares the quality of interpretations between different approaches and various trade-offs. The experimental results include the data processing in the research, the establishment of training models, a comparison of the results and feature visualization methods, and the consequences of improving the training models. First, we analyzed the data processing methods of the dataset, trained the VGG16 model, and finally added a visual interpretation method to the model to visualize and explain the model identification results. The experimental results indicated that the best visual discrimination effect was obtained through eXplanation with Ranked Area Integrals (XRAI). In this study, XAI was used to obtain the factors contributing to incorrect predictions based on post-hoc explanations. Based on the inferenced result, we proposed an adjustment method for improving the model accuracy as a basis for future research to subsequently adjust and improve the model. It is hoped that the experimental results of this study can provide researchers in artificial intelligence useful information so that they can use XAI to acquire appropriate interpretations to correct recognition accuracy and drive the development of XAI.

1. Introduction

In recent years, due to the rapid development of artificial intelligence (AI), judgments made by AI have begun to indirectly or directly affect daily human life. In order to understand how artificial intelligence makes the judgments related to ordinary people, it is necessary to use some interpretation methods.

Therefore, explainable AI (XAI) plays a vital role in helping people understand how artificial intelligence can make judgments. XAI has attracted a great deal of attention in recent years. The development process, main concepts, and methods of XAI are described in a study of Arrieta et al. [1]. In light of XAI, problems often encountered in the current research can be resolved, particularly the errors that arise when using a deep learning model for image classification.

The identification target of this study is an orchard pest called *Tessaratoma papillosa* (*T. papillosa*). This pest can harm *Litchi chinensis* (litchi) and *Dimocarpus longan* (longan) by piercing and sucking, cause flower and fruit drop, twigs and young fruits to wither, and sour fruit rot in litchi in the late stage of the fruit development (Schulte et al., 2006). By using a deep learning network to classify images of *T. papillosa*, we found that twigs, dead leaves and other non-*T. papillosa* objects are often misjudged as indicating the presence of *T. papillosa*. To solve the poor classification accuracy related to *T. papillosa*, we designed an experiment based on XAI that conforms to post-hoc explanations.

First, we classified the proportion of the target objects in the image in the data set into five categories: extra small, small, medium, large and extra-large. Then, we divided it into different pixel sizes as samples.

We used VGG16 as the model architecture to train the samples. The category with the highest probability during classification was the predicted category of the image. We used four visualization methods, including a sensitivity analysis (hereafter referred to as Gradient), an integrated gradient, Grad-CAM, and XRAI, and then performed feature visualization on the recognition results to determine the reason for the incorrect predictions in order to adjust the model based on the results of the visual presentation and explain the decision made by the model to the user. Among the available methods, SmoothGrad was applied to the Gradient and integrated gradient to eliminate noises. Consequently, they became smooth and smooth integrated gradients, respectively. Therefore, a total of six visualization methods were used.

After the experiment, we classified the input image with the trained VGG16 deep learning model, compared the visualized results with the original input image and determined how the model was able to distinguish the essential information in the image. Based on the results after visualization, we deduced the following: 1) The important area judged by the model is not only on the target but sometimes is unrelated objects, and 2) even if the important area judged by the model is on the target object, there will still be some prediction errors.

We used post-hoc explanations to explain the results of the error classification and infer why prediction errors may have occurred. Two possible explanations were as follows: 1) The background of the target in the image is too messy, and 2) the features of different categories are too similar. After deducing the situations that led to incorrect identification, we made corrections and provided post-hoc explanations for the identification errors caused by the VGG16 deep learning model. By correcting various conditions that may cause classification errors, the target classification error of the deep learning model was reduced, and the classification accuracy was then improved.

2. Related Works

The purpose of this research is to explain the image classification results of the VGG16 model. Unfortunately, the computing process of the VGG16 convolutional neural network (CNN) model does not meet transparency requirements. Therefore, we used post-hoc explanation techniques to explain the VGG16 model, referring to research on visualizing features in CNNs (Melnyk et al., 2019) that considered visualization to be feasible. Thus, we used visualization to provide a post-hoc explanation of the model errors. This section describes how we used the visual explanation method in post-hoc explanation techniques to conduct research and design experiments to achieve our objective: explaining image classification results.

2.1. Interpretability vs. Explainability

Explainable AI has two main parts: a transparency design and a post-hoc explanation (Lipton 2016), as shown in Fig. 1. A detailed description is as follows:

1. The transparency design of the model reveals the function of the model and enables users to 1) understand the model structure and operating principles. For example, the structure of the decision

tree is easy to understand, 2) understand the components of the model, for example, the composition of the parameters in the logistic regression, and 3) understand the model training algorithm.
2. Post-hoc explanation: The result is used to infer the possible operating mechanism of the model. Visualization is one of ways to determine what leads to a later consequence. In other words, based on the final prediction, we tracked back to the possible factors influencing how a model generates the final prediction. For example, a saliency map makes it possible to determine which pattern features of the target in an image have more significant impacts on the model prediction.

2.2. Local Interpretability or Global Interpretability

The interpretation methods of the model can be divided into global interpretability to explain the behavior of the entire model and local interpretability to explain a single prediction, which are described as follows:

2.2.1. Global Interpretability

Global interpretability refers to focusing on how the model makes predictions based on the pattern features, the model structure, the parameters and other factors, where it focuses on the features that are important and how those features affect each other. The model has many parameters in actual situations, so it is difficult for people to imagine how the features interact to each other when obtaining prediction results. Some models with simple structures are globally interpretable. For example, the weight of the linear regression model and the decision tree model that divides the branches and obtains the predicted value of the node can both be transparently interpreted.

2.2.2. Local Interpretability

Local interpretability is more focused on a single sample or a group of samples. We can regard a model as a black box without considering the complexity of the model and only focus on observing the features in the subset containing the sample and the prediction principle of the sample based on the subset. For example, Local Interpretable Model-Agnostic Explanation (LIME) (Rai 2020) can explain why a single-sample model makes a given judgment for any model.

2.3. Practical Methods for Explaining DNNs

Because an analysis of deep neural network (DNN) models results in many challenges, many scientists have proposed practical methods to explain these models. There are many such explanations, and each explanation method has both advantages and disadvantages. The following discussion focuses on six main explanation techniques: sensitivity analysis, integrated gradients, smooth gradients, smooth integrated gradients, Grad-CAM, and XRAI.

2.3.1. Sensitivity analysis

Image data is typically represented by $\{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ as vectors. If the probability of an image being classified into class c is y_k, by adding perturbation Δx to a pixel x_n, the change Δy played by y_k can be observed, as shown in Eq. (1). If this perturbation Δx significantly impacts the final classification, it indicates a high degree of importance of pixel x_n in the model judgment:

$$\{x_1, x_2, \cdots, x_n, \cdots, x_N\} \quad (1)$$
$$\rightarrow \{x_1, x_2, \cdots, x_n + \Delta x, \cdots, x_N\}$$
$$y_k \rightarrow y_k + \Delta y$$

If one wants to determine the impact of each pixel's perturbation on the prediction, it is necessary to calculate $\left|\frac{\Delta y}{\Delta x}\right|$, which calculates gradient $\left|\frac{\partial y_k}{\partial x_n}\right|$. The gradient represents the importance of pixel x_n in the judgment category. A saliency map is drawn according to the gradient, where an area with a higher level of brightness indicates that the pixels in this area have a more significant influence on the prediction.

A sensitivity analysis (Simonyan et al. 2013) is the first scheme to use a gradient-based method to do the post-hoc explanation of the artificial intelligence model in the aftermath. The process is as follows: First assume a simple linear model $S_c(I)$ as in Eq. (2).

$$S_c(I) = \omega_c^T I + b_c \quad (2)$$

where a one-dimensional vector represents image I; c represents class; ω_c represents the weight vector; b_c is the model bias, and $S_c(I)$ is the score of the linear model. The weight ω_c shown in Eq. (2), defines the pixels' level of importance in image I.

A neural network is a very complicated nonlinear model $S_c(I)$. Therefore, we cannot apply the derivation of Eq. (1) to it. Nevertheless, we can use a first-order Taylor expansion to expand the linear model around the given image I_0 to approximate the $S_c(I)$ value, as shown in Eq. (3):

$$S_c(I) \approx \omega^T I + b \quad (3)$$

where ω is the derivative of the output value S_c in the input image I_0, as in Eq. (4), and ω is the gradient of the output S_c relative to the input image I_0.

$$\omega = \frac{\partial S_c}{\partial I}\Big|_{I_0} \quad (4)$$

Finally, a saliency map is used to present the weight ω of each pixel, so the impact of each pixel in the image on the classification result can be determined.

2.3.2. Integrated Gradients

The gradient-based interpretation method can indeed successfully explain some artificial intelligence prediction results in many cases, but there may also be some unexplainable situations. Both Simonyan et al. (2013) and Baehrens et al. (2010) mentioned that gradient-based explanations encounter unexplainable situations in saturated regions. Once the gradient enters the saturated region and approaches 0, no effective information can be obtained, as shown in Fig. 2.

Taking the gradient as the importance score, then, the gradient is close to 0 in some areas. This will be represented as a low intensity area on the saliency map. Therefore, Mukund Sundararajan et al. (2017) proposed using integrated gradients to solve problems encountered in the saturation region. Instead of gradients, all gradients are integrated as importance scores of pattern features. Using integrated gradients to calculate the integral of the gradient as the importance score can prevent the gradient from approaching 0 in some areas. The difficulty of this method is that for a given image x. However, its intensity has been fixed, so a method is needed to obtain the gradient of an image whose intensity is less than$x$.

First, suppose that the current image is x, and set a baseline image x' with an intensity of 0, which is usually a black image with all zero pixels or a random image. Then, a linear interpolation is performed between the baseline x' and the original image x to generate the intermediate image, as expressed by Eq. (5):

$$\bar{x} = x' + \alpha(x - x') \quad (5)$$

As shown in Fig. 3, when $\alpha = 0$ is the baseline image, when $\alpha = 1$, it is the original input image x. The integrated gradient is defined as in Eq. (6):

$$IntegratedGrads_i(x) = \left(x_i - x_i'\right) \times \int_{\alpha=0}^{1} \frac{\partial F\left(x' + \alpha\left(x - x'\right)\right)}{\partial x_i} d\alpha \quad (6)$$

2.3.3. Smooth Gradients and Smooth Integrated Gradients

Gradient-based interpretation methods, such as sensitivity analysis and integrated gradients, are saliency maps obtained through backpropagation. A saliency map typically has a lot of visual noise. These noises will result in the saliency map only inferring the location of the relevant area, which is not the result of a human process. The surrounding pixels of the target in the saliency map will exhibit a high degree of brightness, but these high-brightness pixels are not related to the target in the original image.

Smilkov et al. (2017) mentioned the reason for the noise may be that the derivative of the function S_c is not even continuously differentiable.

The SmoothGrad method randomly adds a specific degree of noise to the input image and then calculates the gradient. After several gradient calculations, the average value is taken to make the gradient change more stable in order to remove the noise. If the saliency map $M_c(x)$ is the gradient value of the output to the input, as in Eq. (7):

$$M_c(x) = \frac{\partial S_c(x)}{\partial x}, \quad (7)$$

then, SmoothGrad perturbates the input image, adds a slight difference $N\left(0, \sigma^2\right)$ to generate n perturbated images, calculates their gradient average, and then obtains a stable saliency map, as shown in Eq. (8):

$$\hat{M}_c(x) = \frac{1}{n}\sum_1^n M_c\left(x + N\left(0, \sigma^2\right)\right) \quad (8)$$

When the value of sample size n becomes larger, the saliency map has less noise and is more stable. We apply the SmoothGrad scheme to eliminate noise and generate two methods, a smooth gradient and a smooth integrated gradient.

2.3.4. Grad-CAM

Using class activation mapping (CAM) (Zhou et al. 2016), it is necessary to connect the global average pooling layer (GAP Layer) after the latest convolutional layer output and retrain the model, so this method is unsuitable for practical applications. Therefore, Selvaraju et al. (2017) improved the CAM method and proposed a Grad-CAM combining gradient information and a feature map. Regardless of whether the convolutional layer is connected to a fully connected layer or other different types of networks, the heatmap can be obtained using the Grad-CAM method without modifying the network. Grad-CAM replaces the weight value with a gradient and directly uses the output layer's output and the convolutional layer to calculate the gradient.

To improve performance details, Selvaraju et al. (2017) multiplied the result of Guided Backpropagation and the original Grad-CAM result to affect the output image and obtain a higher resolution and a heatmap with superior positioning accuracy.

2.3.5. XRAI

The XRAI method is used to extract symbolic representations of a mathematical function. A symbolic representation of the mathematical function is used as the basis of the representations for learning the neural network during training. The XRAI method is used to adjust the interpretation method used after the neural network is trained. After neural network training, the weight and deviation values are used as

the input for the interpretation method and determine how to express the neural network formula during training. This formula is then converted into a symbolic representation in the XRAI method.

In the study of *XRAI: Explainable Representations through AI* (Christiann et al., 2020), it is indicated that Boolean functions and low-order polynomials can be used as examples to perform offline training on synthetic data to explain different types of functions. Unlike with integrated gradients, XRAI also evaluates the overlapping area of the image to reconstruct a saliency map that highlights the relevant area of the image instead of pixels.

Kapishnikov et al. (2019) first performed oversegmentation on an image, repeatedly tested the importance of each area, and merged smaller regions into larger regions based on the attribution score. The results of the experiment confirmed that this strategy can produce significant regions with high quality, tight boundaries, and the performance of XRAI was better than other existing post-explanatory methods. More importantly, XRAI can be used with any DNN-based model as long as the input features can be classified through a similarity calculation (for example, the color similarity in the image).

Kapishnikov et al. (2019) proved that under a general neural network model, for the purpose of comparing post-interpretations of an ImageNet data set, XRAI is more effective. This interpretable method is often applied to models that accept image input, such as natural images of any real scene containing multiple objects.

The XRAI algorithm proposed by Christiann et al. (2020) is as follows:

Given image *I*, model *f* and attribution method *g*

Over-segment *I* to segments $s \in S$

Get attribution map $A = g(f, I)$

Let saliency mask $M = 0$, trajectory $T = []$

while $S \neq \emptyset$ *and* $area(M) < area(I)$ **do**

for $s \in S$ **do**

Compute gain[2]: $g_s = \sum_{i \in s \setminus M} \frac{A_i}{area(s \setminus M)}$

end for

$\hat{s} = \underset{s}{argmax} g_s$

$S = S / \hat{s}$

$M = M \cup \hat{s}$

Add *M* to list *T*

end while

return *T*

XRAI uses integrated gradients to satisfy sensitivity-N (Ancona et al., 2017), where the sum of all its input parameters will be equal to the input of the *softmax* value subtracted from the *benchmark softmax* value. XRAI starts with an empty mask and then selectively and continuously adds the most profitable block in the total attribution of each block until the complete image is obtained as a mask or all the blocks that can be added have been used. The density of the trajectory of masks obtained from the calculation above is regarded as an essential order when sorting the blocks, which means that image blocks contributing to the prediction category should have a high positive attribution. Blocks that are not related to the prediction should have an attribution close to zero. Blocks that contain competing types should have a negative attribution.

3. Research Methods

After putting the XAI method into perspective, we referred to two studies on image classification of plant diseases and insect pests (Vaishnnave et al., 2020; Ji et al., 2020) and then designed an experiment to improve the classification accuracy of *T. papillosa* images. A diagram of the architecture is shown in Fig. 4, which shows how the model judges the input images and predicts their category through XAI. Six visualization methods were employed to visualize the block5-conv3 layer in VGG16 for post-hoc explanations of how the model made judgments. The six visualization methods are as follows: sensitivity analysis, integrated gradients, smooth gradients, smooth integrated gradients, Grad-CAM, and XRAI.

Afterward, by comparing the results predicted by the model with the visualization results, we determine the reasons for the correctness of the image error prediction in the experiment.

The approach used in a study of XAI conducted by Arrieta et al. (2020) was employed in the present research for the purpose of recognizing litchi stink bugs based on deep learning networks. XAI demonstrates a high degree of transparency. In transparent machine learning models, it is necessary to have the ability to interpret the model while designing the model to meet the definition of transparency, including requirements for simulability, comprehensibility, and algorithm transparency. The model design of the deep learning network cannot meet the requirements for transparency due to its many layers and degree of complexity. We found that post-hoc explanation methods are more suitable for explaining the research content of *T. papillosa* than transparency methods. Therefore, we used six post-hoc explanation methods, including a sensitivity analysis, integrated gradients, smooth gradients, smooth integrated gradients, Grad-CAM and XRAI, to explain classification errors related to *T. papillosa*.

3.1. Sensitivity Analysis

A sensitivity analysis (Simonyan et al., 2013) is the first method using a gradient-based form. The Gradient will be referred to as a sensitivity analysis in the following discussion. The importance of each pixel to the prediction result can be observed using a saliency map drawn by calculating the magnitude of the gradient value. When a pixel's brightness in the saliency map is higher, this indicates that it has higher importance related to the output result.

3.2. Integrated Gradients

Sundararajan et al. (2017) used integrated gradients to solve the saturating gradient problem caused by the gradient-based method. If a gradient is replaced with the integral of the gradient as the importance score, this will not result in the gradient value being 0. Sundararajan et al. (2017) assumed a current image x and a baseline image x', which usually is a black image with an intensity of 0. Then, the linear interpolation was applied to the baseline and the original image x to generate the intermediate image.

3.3. Smooth Gradient and Smooth Integrated Gradient

SmoothGrad (Smilkov et al. 2017) used the addition of noise to eliminate noise, making the saliency map more conducive to the interpretation of the model. By randomly adding different degrees of noise to the input image, the gradient of the noise image can be calculated and averaged. This procedure will make the gradient change more stable so as to remove the noise in the saliency map. Both the Gradient and the integrated gradient use SmoothGrad to eliminate noise and become the so-called smooth gradient and smooth integrated gradient.

3.4. Grad-CAM

In a study conducted by Zhou et al. (2016), CAM was connected to the GAP Layer in the last convolutional layer and the model was retrained. This kind of operation is complicated, so it is infeasible for practical applications. The Grad-CAM in Selvaraju et al. (2017) does not need to modify the network architecture. It calculates the gradient based on the output of the output layer and the output of the convolutional layer. We thus could use this gradient to replace the weight value in CAM to save the time required for model retraining.

3.5. XRAI

Kapishnikov et al. (2019) demonstrated XRAI saliency maps where pixels were replaced with regions. The interpretability of the XRAIs was better than other existing post-hoc explanation methods and could be incorporated into any DNN-based model. In our experiment, an XRAI heat map was created, and the top 30% of the blocks that were most relevant to the prediction result were selected to interpret the correlations.

4. Experimental Results And Discussion

In this study, we used a camera to collect images of *T. papillosa* located on the top of trees, and the images were appropriately cut and then classified. The training data set was divided into eight categories:

eggs of *T. papillosa*, *T. papillosa* 30 mins prior to birth, *T. papillosa* larva 30 mins after birth, juveniles of *T. papillosa*, *T. papillosa*, branch, leaf, and longan. We used the VGG16 neural network model with six post-explanation methods, including Gradients, integrated gradients, smooth gradients, smooth integrated gradients, Grad-CAM, and XRAI and applied them to the trained model. Finally, the advantages and disadvantages of the six post-hoc explanation methods were compared according to the relationship between the importance of pattern features and the prediction results.

4.1. Data Processing

In this research, 987 images of *T. papillosa* collected with a camera comprised the data set. The proportion of the target classification object in the image could be divided into five scales: extra small, small, medium, large, and extra-large, as shown in Fig. 5. The description of the five scales is as follows:

1. Extra small: The target occupies a tiny proportion of the image. The image is cut with 150x150 pixels, for a total of 58 samples.
2. Small: The target occupies a small proportion of the image. The image is cut by 300x300 pixels, for a total of 224 samples.
3. Medium: The target occupies a moderate proportion of the image, and the image is cut with 600x600 pixels, for a total of 574 samples.
4. Large: The target occupies a more significant proportion of the picture, and the image is cut with 1000x1000 pixels, for a total of 119 samples.
5. Extra-large: The target occupies a large proportion in the picture and remains uncut, for a total of 12 samples.

Images at five different scales were segmented by cutting the image into many small images and making the target in each small image progressively clearer after cutting, as shown in Fig. 6. After the segmentation was completed, these images were classified into eight different types, labeled from 0 to 7 for eggs of *T. papillosa*, *T. papillosa* 30 mins prior to birth, *T. papillosa* larva 30 mins after birth, juvenile of *T. papillosa*, *T. papillosa*, branch, leaf, and longan, for a total of eight categories, as shown in Fig. 7.

To maintain the integrity of the data set, fuzzy images or images containing multiple targets were deleted and excluded from the data set. After cutting and classifying, there were a total of 6,983 samples available for model training using VGG16. The sample numbers for the 8 types of target images in the data set are detailed in Table 1.

Table 1
The 8 types of image samples of the target in the data set

Label	Category	Image Samples
0	Eggs of *T. papillosa*	87
1	30 mins prior to birth	108
2	larva 30 mins after birth	178
3	Juvenile of *T. papillosa*	257
4	*T. papillosa*	586
5	Branch	275
6	Leaf	4,868
7	Longan	627

4.2. Establishment of the Training Model

In this work, the XAI method was used to do the analysis of the identification of *T. papillosa* based on deep learning networks. A single image taken by an UAV was classified into one of eight types (eggs of *T. papillosa*, *T. papillosa* 30 mins prior to birth, *T. papillosa* larva 30 mins after birth, juvenile of *T. papillosa*, *T. papillosa*, branch, leaf, and longan). Since VGG16 is better than other neural network models in image classification, it was used for *T. papillosa* images.

First, we used the weights of 1 million images being trained on ImageNet as the pre-training models and proceeded with training. The final output layer is changed to eight categories. The category with the highest probability is taken as the predicted category of the image.

We divided the image into a training set and a testing set and then split 20% of the samples from the training set as a validation set. After continuous testing, we used the Adam optimizer. The learning rate is 0.000001 as the basic parameters of the experiment in this work. The schematic diagram of the model architecture is shown in Fig. 8.

After the VGG16 model was trained, the accuracy of the training set was 81.1%; the accuracy of the validation set was 69.1%, and the accuracy of the training set was 68.8%, as shown in Fig. 9(a). Figure 9(b) shows the confusion matrix obtained by predicting the testing set using the self-trained VGG16 model. According to the prediction results of this confusion matrix, the accuracy of the self-training model in the test set was not good. Most larva were predicted to be *T. papillosa*, and a leaf was predicted to be *T. papillosa* or a branch.

To put the basis of the self-training model's prediction into perspective, we used a post-hoc explanation method to explain the self-training model to explain the model classification errors.

4.3. Visualizing Prediction Pattern Features

From the prediction results of the VGG16 training model, we found that many targets were classified into the wrong category. We used the Gradients, integrated gradients, smooth gradients, smooth integrated gradients, Grad-CAM, and XRAI visualization model methods to discover the reason for these classification errors. The last convolution layer, the block5-conv3 layer, of the model was visualized to determine how the model worked. By adjusting the model parameters through the results of the visualization effects, we could determine the reasons for the model's prediction errors and explain the decisions made by the model to users.

As shown in Fig. 10 and Fig. 11, different visualization methods were used to represent the correct and incorrect predictions of eight target objects. It can be observed that the pixels or blocks with high importance fall on the target in the saliency map where the prediction of each category is correct (Fig. 10). Therefore, the visualization method explains the model well. However, the model can be improved to classify the image into the correct category based on the pattern features of the target. Thus, two phenomena can be observed based on Fig. 11.

1. In some saliency maps, the essential pixels or areas not only fall on the target but also on other unrelated objects, as shown in Fig. 12. Taking juvenile *T. papillosa* and *T. papillosa* as an example, it can be observed that not only the features of juvenile and *T. papillosa* are given attention in the remarkable picture, but the leaves next to them are also regarded as essential features. It is believed that the background images in the data set are too complicated, which leads to the model learning other types of features during model training.

1. Although important pixels or blocks fall on the target, they are still mispredicted, as shown in Fig. 13. Taking *T. papillosa* 30 mins prior to birth and *T. papillosa* larva 30 mins after birth as examples, there are targets in the important part of the saliency map, but they were still predicted as the wrong category. We believe that the features among different categories are too similar. It is recommended that images with higher resolution be used rather than out-of-focus as training samples so that the model can learn the pattern features of each category correctly.

4.4. Comparison of the Pattern Features in the Visualization Methods

Although the visualized Gradient and integrated gradient saliency maps can provide high feature correlations, there will be still a lot of noise. Smooth gradient and smooth integrated gradients, which eliminate noise, can make observations of essential features easier. Grad-CAM overlays the heat map on the original image, which stresses the critical points of a feature. XRAI regionally expresses essential features, indicating that it is the best visualization method because the features that affect the prediction results can be seen immediately.

As shown in Fig. 14, we preserved the top 30% of the XRAI saliency map most relevant to the prediction and removed the remaining less relevant 70% of the area. The XRAI method significantly improved the judgment efficiency, making the observation of essential features more intuitive.

4.5. Improving the results of the training model

We visualized the last convolutional layer, called the block5-conv3 layer, of the original VGG16 model based on the post-hoc explanation method to explore the reasons for the model's prediction errors. It is believed that too many low-resolution, out-of-focus, and cluttered background images in the training data set caused the model prediction errors. Therefore, the training data set is adjusted to remove bad images, retrain the model, and re-predict the test set.

The parameters of the improved model were the same as those of the original self-training model. After enhancing the model, the accuracy of the training set was 92.5%; the accuracy of the validation set was 75.6%, and the accuracy of the training set was 72.5%, as shown in Fig. 15 (a), where the test accuracy of the improved model was higher than that of the original model.

After comparing the confusion matrix of the original model Fig. 9 (b) and the improved model Fig. 15 (b), we found that there were fewer prediction errors. At the same time, we also found the prediction accuracy of the improved model was improved with respect to the original model. The improvements in accuracy are shown in Table 2 and Fig. 16. Therefore, we concluded that the selected data set has a significant influence on the model's prediction.

We used the post-hoc explanation to visualize the output pattern features of the convolutional layer to help lead to an understanding of the model's prediction standards and then improve the model to obtain better predictions.

Table 2
Comparison of the accuracy of the original and improved VGG16 model

Accuracy	Training set	Validation set	Testing set
Original	0.811	0.691	0.688
Improved	0.925	0.756	0.725

5. Conclusions And Prospect

After visualizing the classification results of *T. papillosa*, we inferred the possible reasons for classification errors as follows: 1) The background of the data set image is cluttered, which leads to the learning of other non-target features during model training. 2) The different characteristics of the categories are not distinguishable. Therefore, we concluded that high-resolution images should be selected in future data set images. The number of training samples for each category should be

increased so that the model can learn the characteristics of each target category more correctly, which can improve the target weight of the feature of the class in the model. We suggest that in the future, when performing model training for image classification, the content of the experiment can be improved to reduce classification errors and improve accuracy. Furthermore, the experiment and image classification accuracy can be indeed improved by the inferences made by explainable artificial intelligence.

Declarations

Declarations

Funding

This research is supported by the Ministry of Science and Technology, Taiwan, R.O.C. under Grant no. MOST 109-2321-B-067F-001- and MOST 110-2321-B-067F-001-.

Conflicts of interest/Competing interests

The authors declare that they have no competing interests.

Authors' contributions

Ching-Ju Chen conceived and designed the research framework. She is also responsible for data analysis and the draft of the paper; Ling-Wei Chen carried out the experiment by following the proposed framework. She also provided the description of experiment procedure in the draft; Chun-Hao Yang collected the data and did comparisons; Ya-Yu Huang collected the data, including literature survey; Yueh-Min Huang supervised the research process and edited the draft.

Ethics approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Consent to participate

No need consent to participate.

Consent for publication

Not applicable.

References

1. Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A et al (2020) Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible

AI. Information Fusion 58:82–115

2. Schulte MJ, Martin K, Sauerborn J (2006) Effects of azadirachtin injection in litchi trees (Litchi chinensis Sonn.) on the litchi stink bug (Tessaratoma papillosa Drury) in northern Thailand. J Pest Sci 79(4):241–250
3. Melnyk P, You Z, Li K (2019) A High-Performance CNN Method for Offline Handwritten Chinese Character Recognition and Visualization. Soft Comput 24:7977–7987
4. Lipton ZC (2016) The mythos of model interpretability. ICML Workshop on Human Interpretability in Machine Learning, New York
5. Rai A (2020) Explainable AI: from black box to glass box. J Acad Mark Sci 48:137–141
6. Simonyan K, Vedaldi A, Zisserman A, Andrew Z (2013) Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034
7. Baehrens D, Schroeter T, Harmeling S, Kawanabe M, Hansen K, Mueller KB (2010) How to explain individual classification decisions. The Journal of Machine Learning Research 11:1803–1831
8. Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. International Conference on Machine Learning. Proceedings of Machine Learning Research
9. Smilkov D, Thorat N, Kim B, Viégas F (2017) Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825
10. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. Proceedings of the IEEE conference on computer vision and pattern recognition
11. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE international conference on computer vision
12. Christiann B, Marton S, Stuckenschmidt H (2020) xRAI: Explainable Representations through AI. arXiv preprint arXiv:2012.06006ss
13. Kapishnikov A, Bolukbasi T, Viégas F, Terry M (2019) Xrai: Better attributions through regions. Proceedings of the IEEE/CVF International Conference on Computer Vision
14. Ancona M, Ceolini E, Ztireli C, Gross M (2017) Towards better understanding of gradient-based attribution methods for Deep Neural Networks. arXiv:1711.06104
15. Vaishnnave MP, Suganya Devi K, Ganeshkumar P (2020) Automatic method for classification of groundnut diseases using deep convolutional neural network. Soft Comput 24:16347–16360
16. Ji M, Zhang K, Wu Q et al (2020) Multi-label learning for crop leaf diseases recognition and severity estimation based on convolutional neural networks. Soft Comput 24:15327–15340

Figures

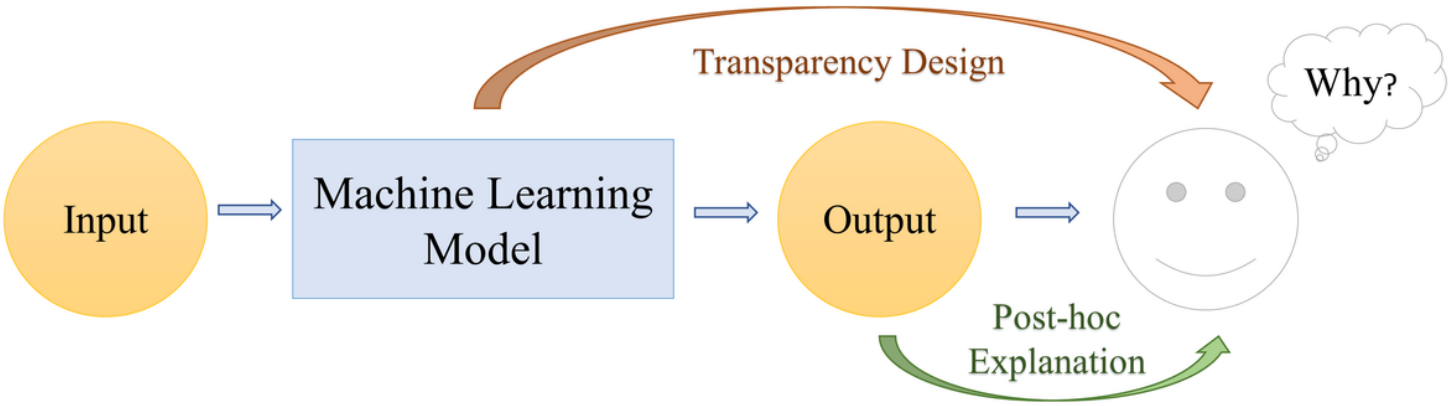


Figure 1

Two significant parts of XAI: Transparency design and the post-hoc explanation

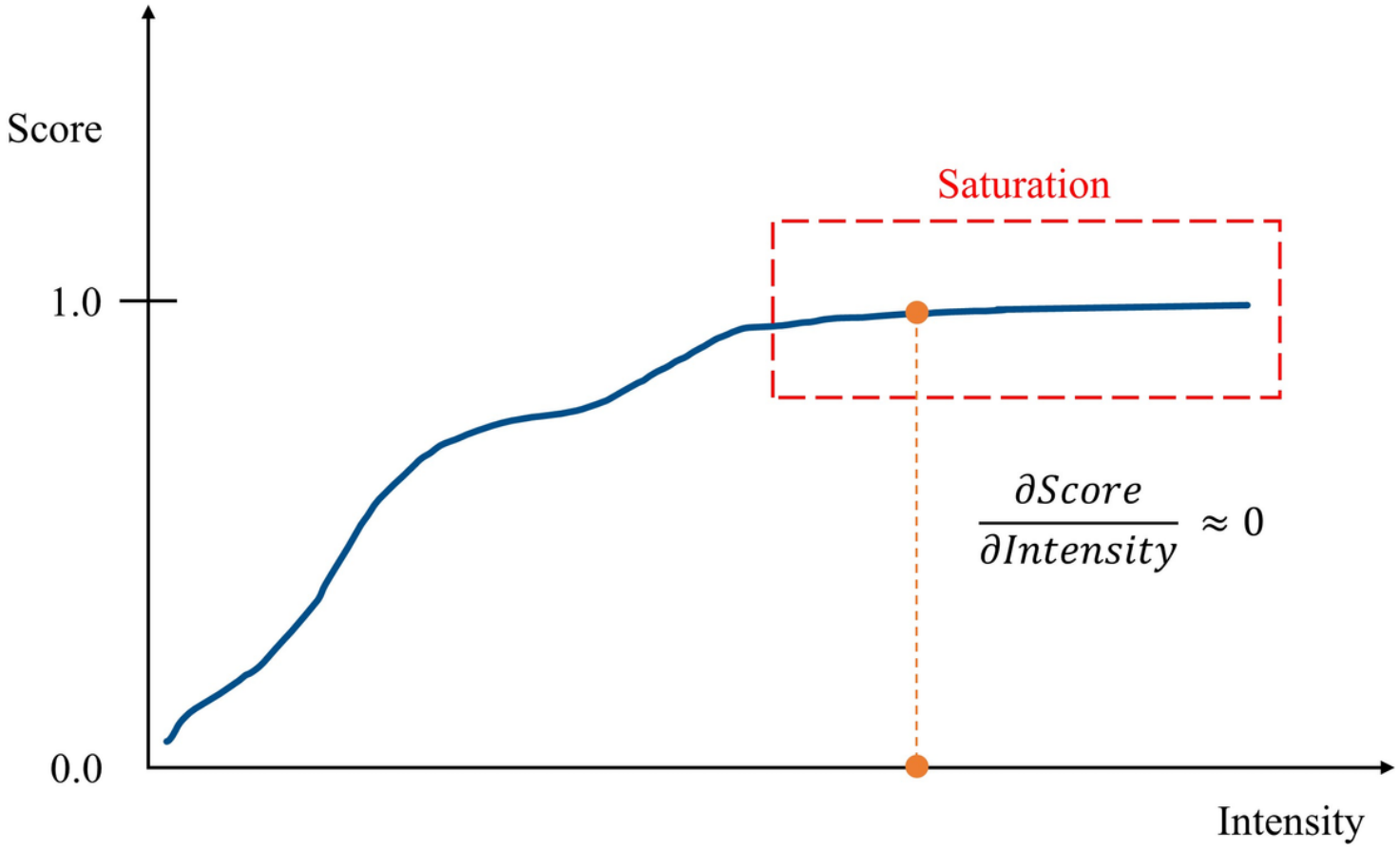


Figure 2

The gradient in the saturation region approaching 0

Figure 3

Interpolated images generated between the baseline image and the original input image using linear interpolation

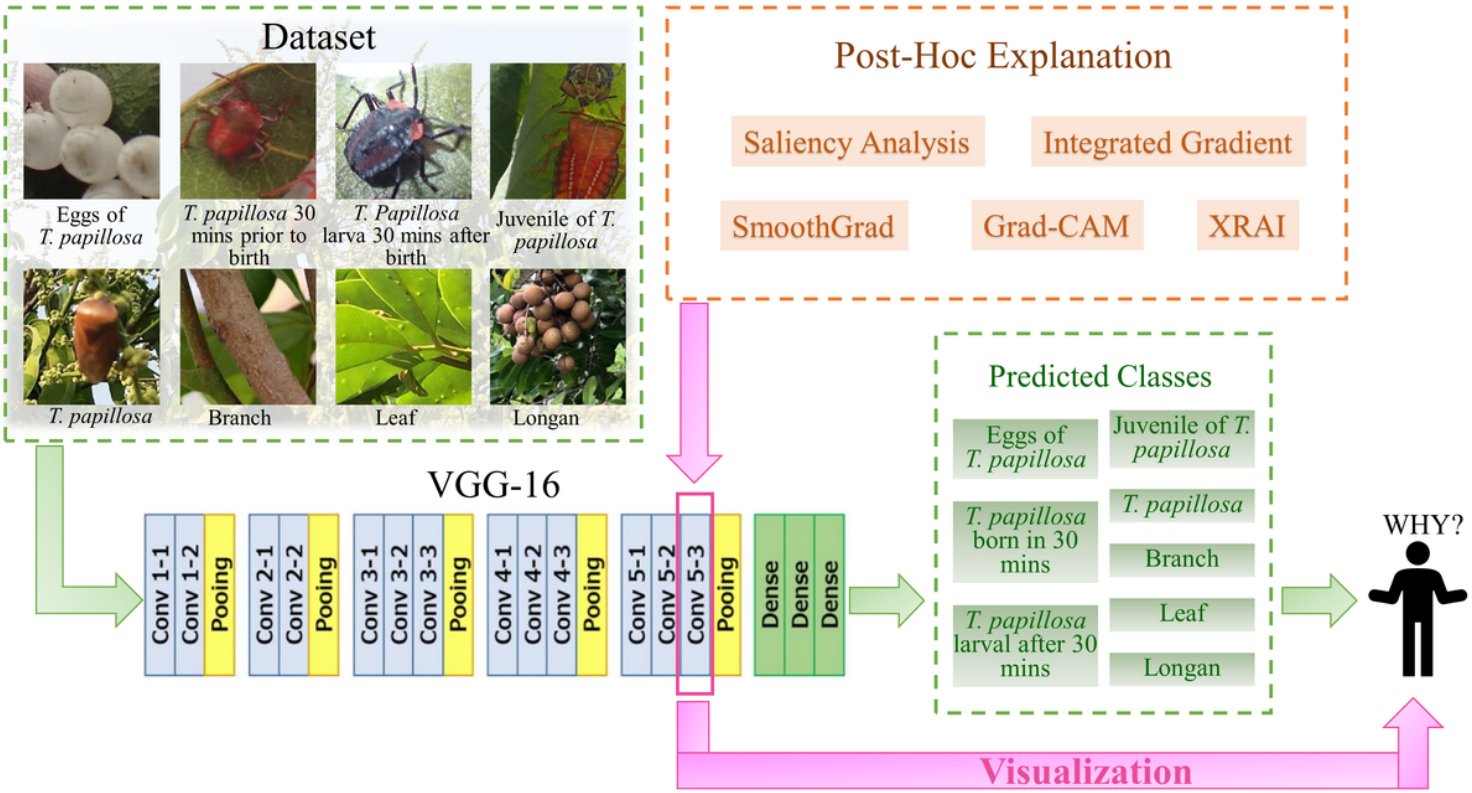


Figure 4

Diagram of the experimental architecture

Figure 5

The classification is based on the proportion of the size of the target identifier T. papillosa in the picture.

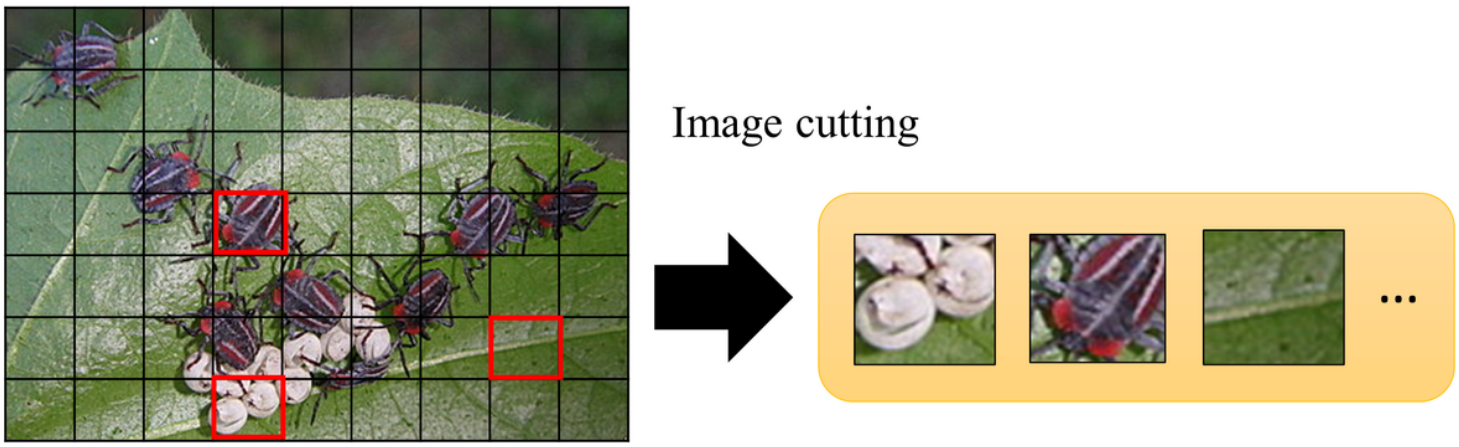


Figure 6

Cutting an image into many small images

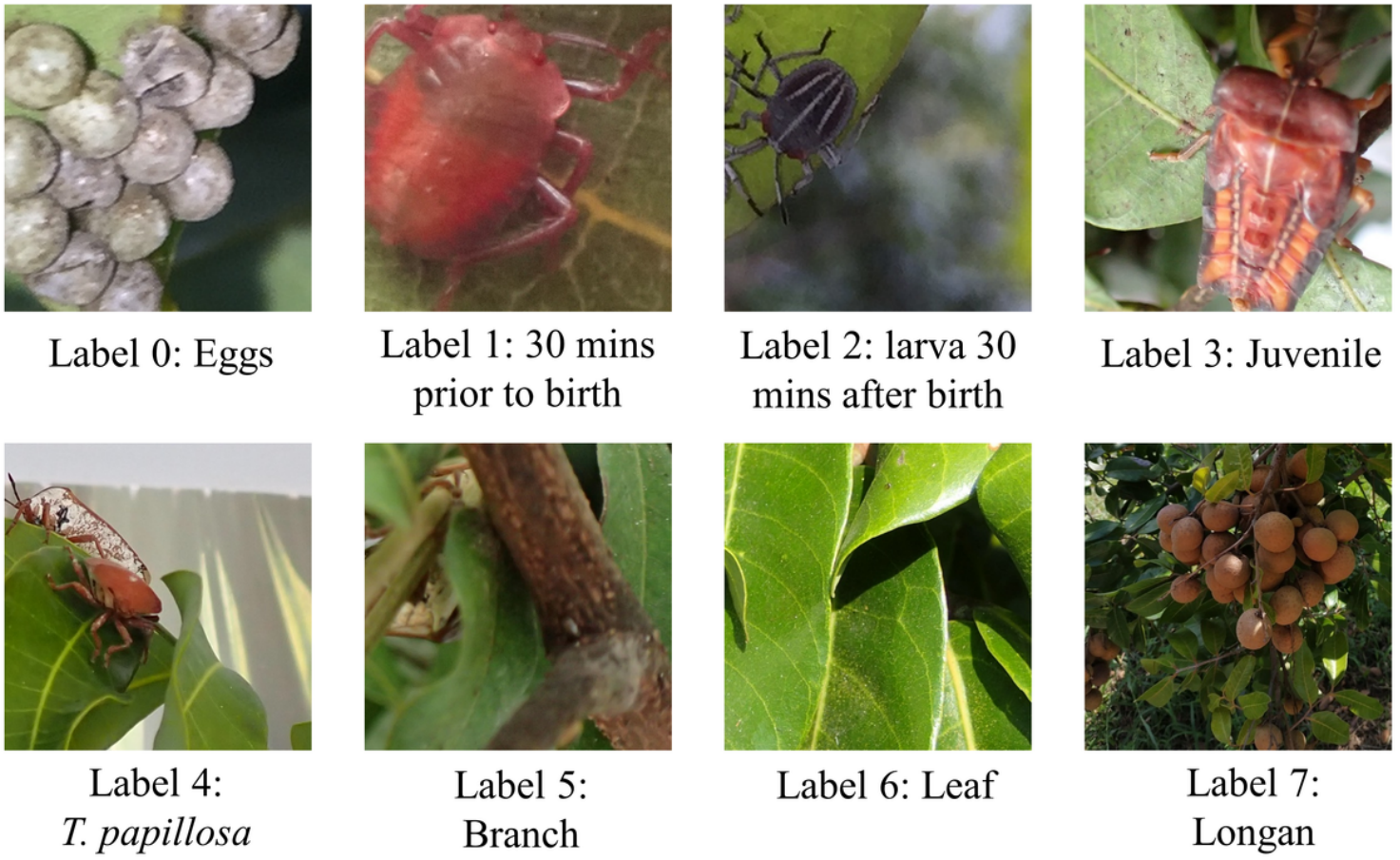


Figure 7

The target images after cutting in each category

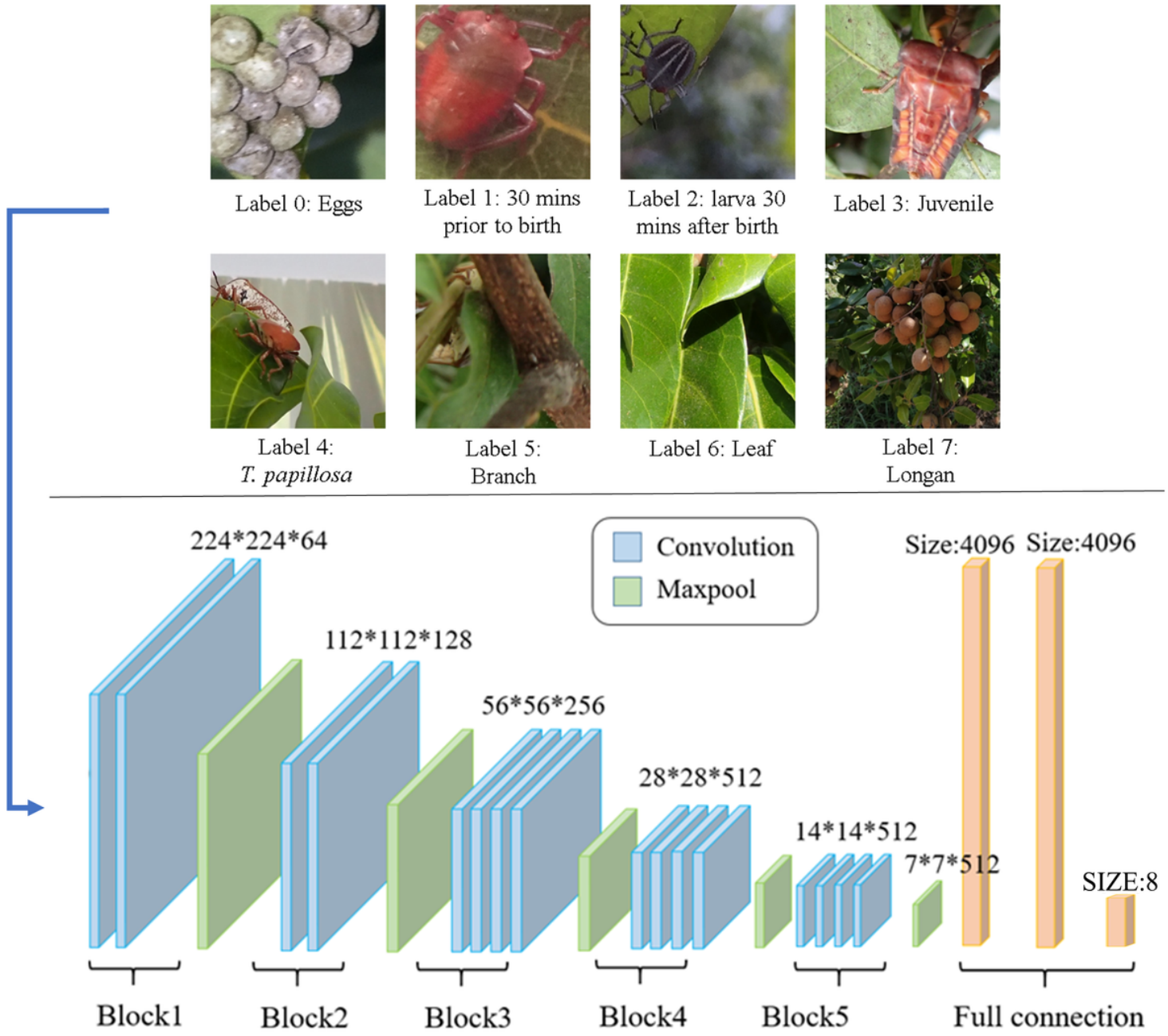
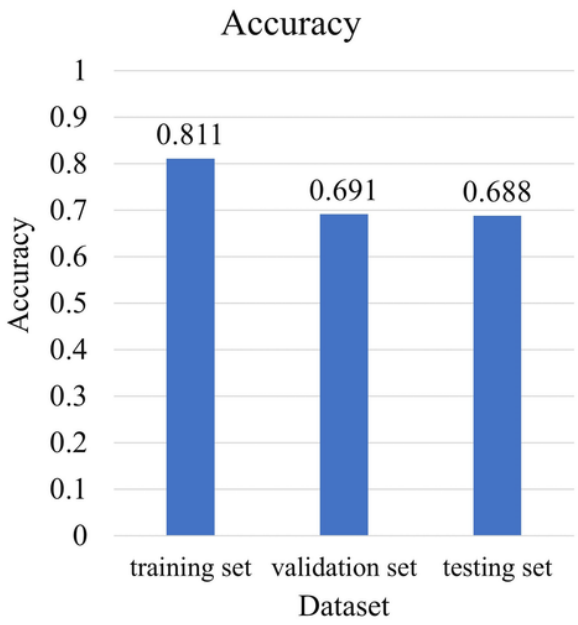


Figure 8

VGG16 model architecture



(a) The accuracy of the self-trained VGG16 model on the training set, validation set, and test set, respectively.

Predict Label	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**All**
0	17	0	0	0	1	0	0	2	20
1	0	3	0	15	1	0	1	0	20
2	0	1	1	9	5	2	0	2	20
3	0	0	0	14	6	0	0	0	20
4	0	0	0	2	17	1	0	0	20
5	0	0	0	0	0	20	0	0	20
6	0	0	0	0	4	14	2	0	20
7	0	0	0	0	3	0	0	17	20
All	17	4	1	40	37	37	3	21	160

(b) The confusion matrix was obtained using predicting the test set by the self-trained VGG16 model.

Figure 9

The accuracy and confusion matrix in the self-trained VGG16 model

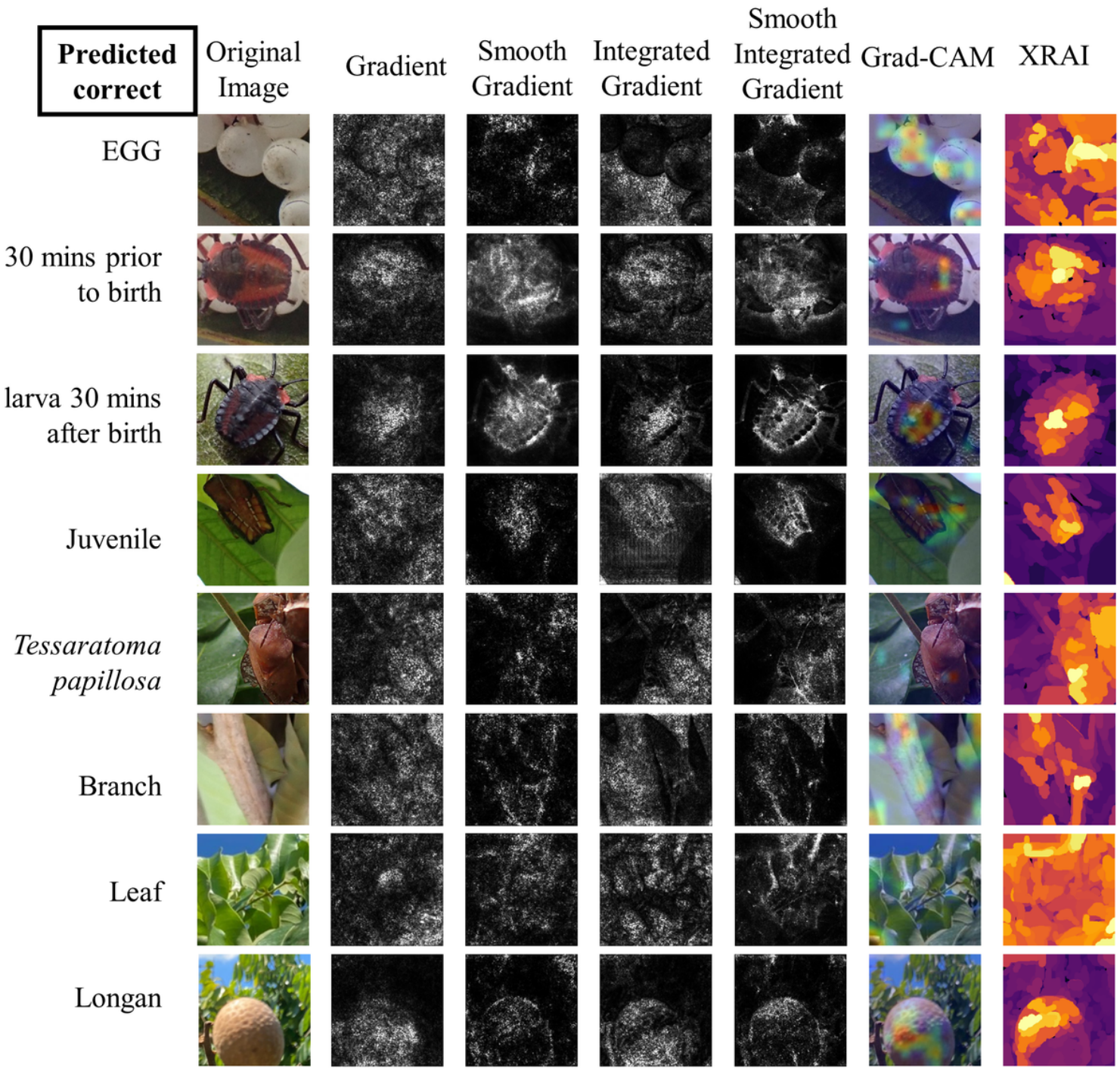


Figure 10

Using different visualization methods to output images with correct predictions for each category

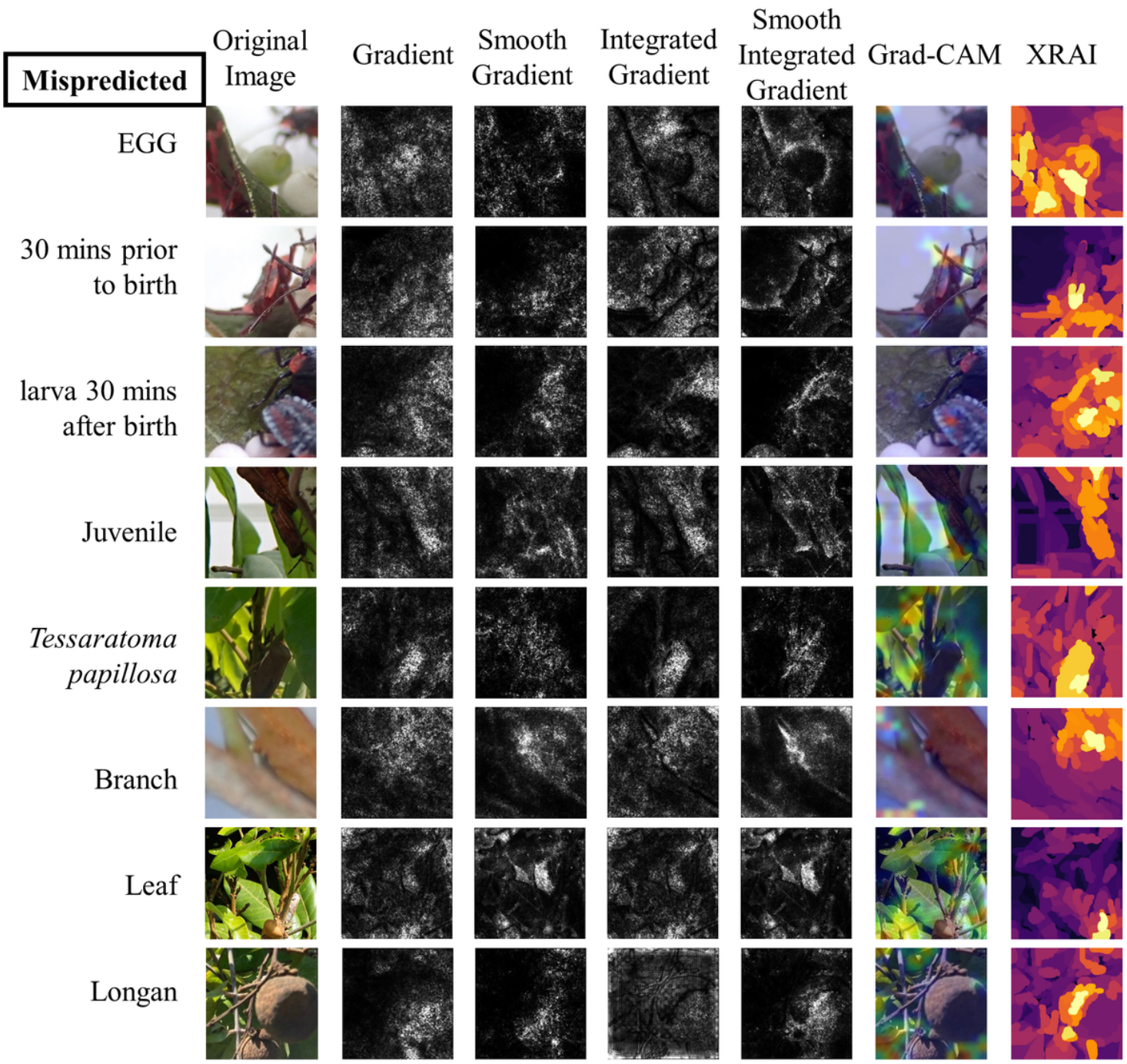


Figure 11

Using different visualization methods to represent images with incorrect predictions for each category

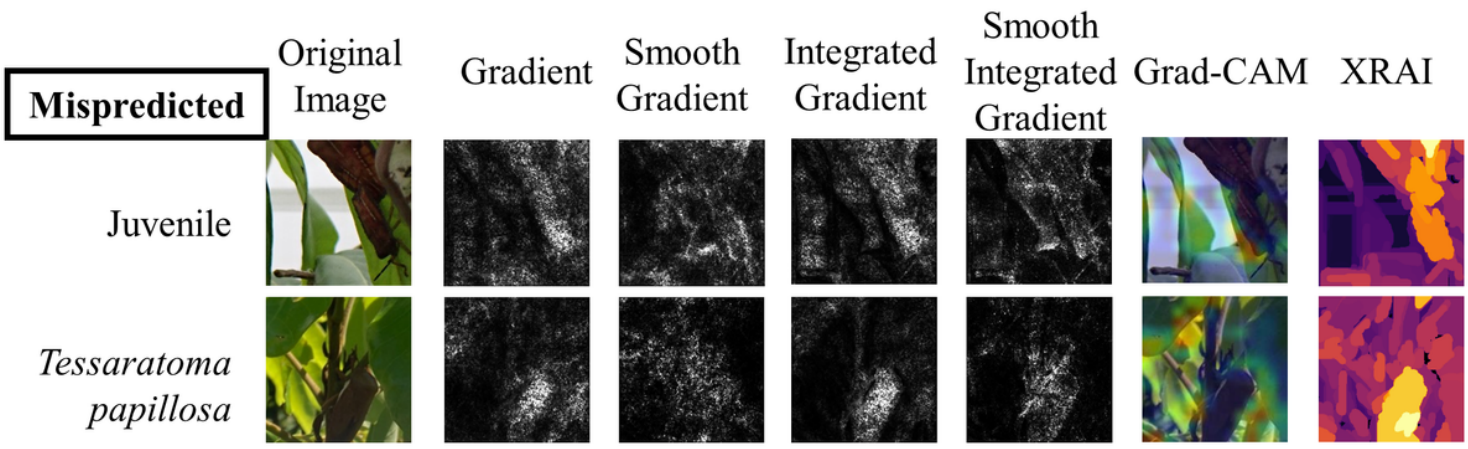


Figure 12

The pattern features of objects other than the target are also mistakenly adopted, leading to misprediction.

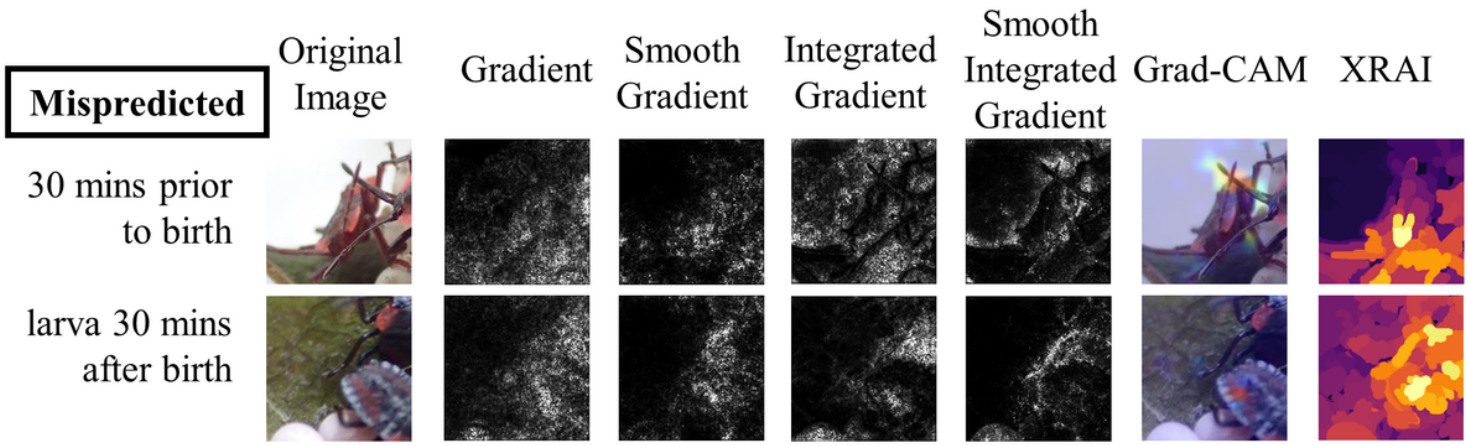


Figure 13

A target pattern feature with high importance but classified into the wrong category

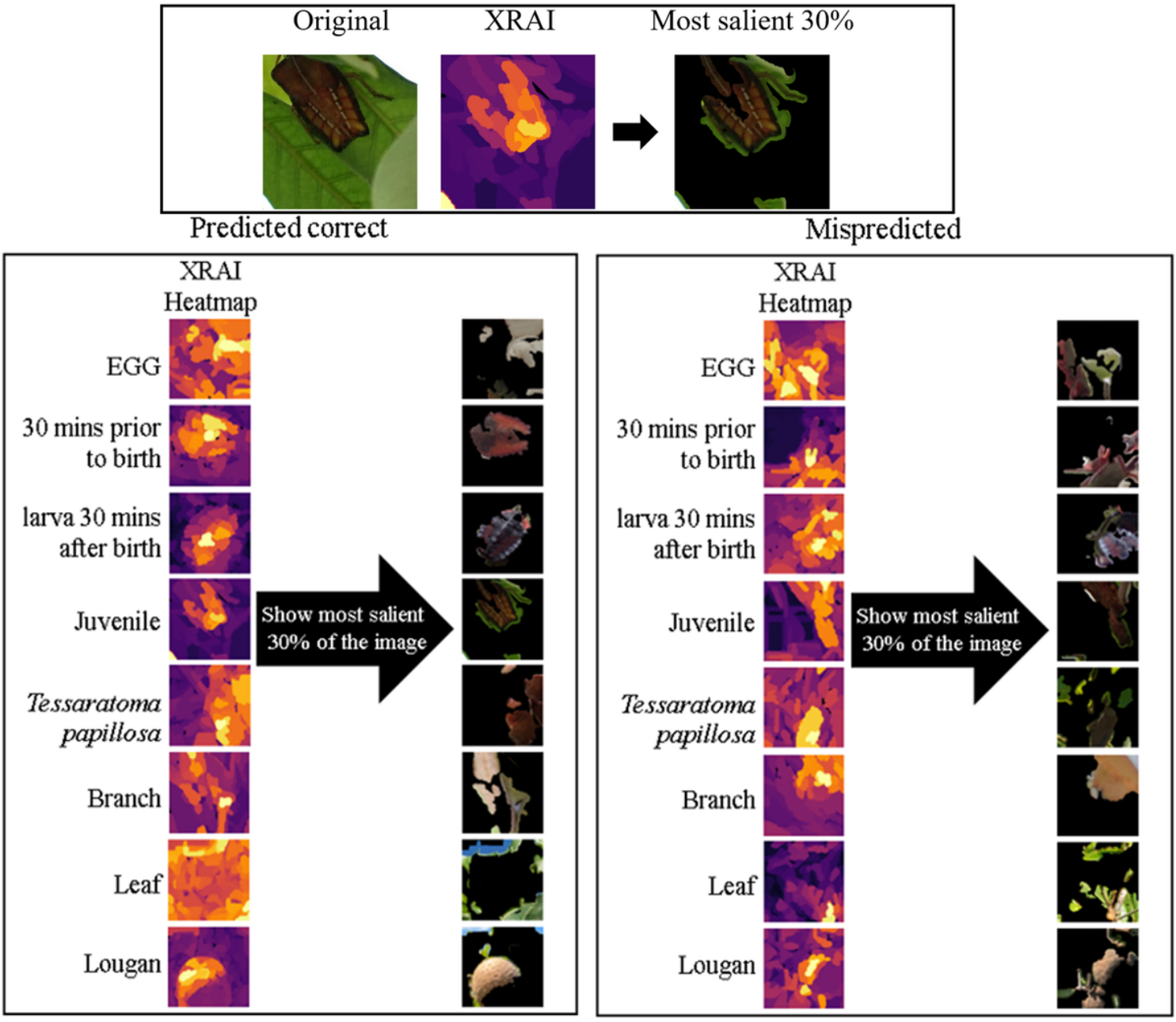
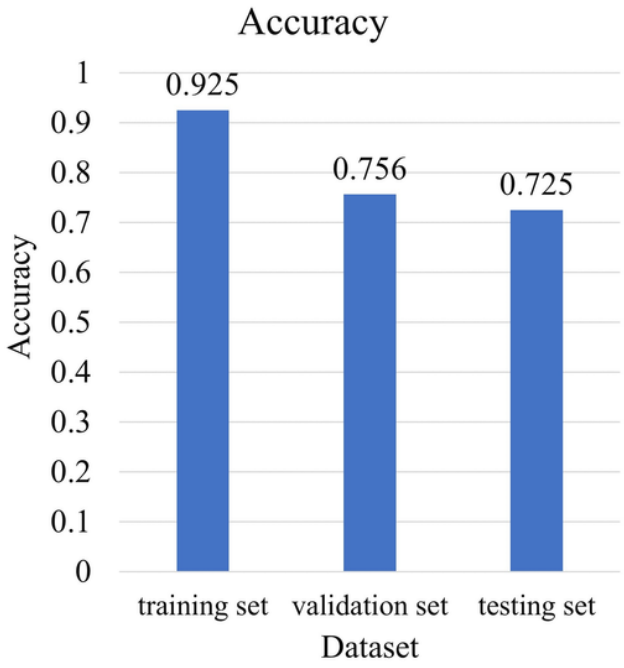


Figure 14

The most important 30% of the XRAI saliency map used to judge the importance of the pattern features



(a) The accuracy of the improved VGG16 model in the training set, validation set, and test set.

Predict Label	0	1	2	3	4	5	6	7	All
0	19	0	0	0	0	1	0	0	20
1	0	6	7	1	6	0	0	0	20
2	2	0	12	2	0	1	2	1	20
3	0	0	0	11	8	1	0	0	20
4	0	0	0	0	18	2	0	0	20
5	0	0	0	0	0	19	1	0	20
6	1	0	1	0	2	2	14	0	20
7	0	0	0	0	1	0	2	17	20
All	22	6	20	14	35	26	19	18	160

(b) The confusion matrix obtained by the enhanced VGG16 model predicting the test set.

Figure 15

The accuracy and confusion matrix of the improved VGG16 model

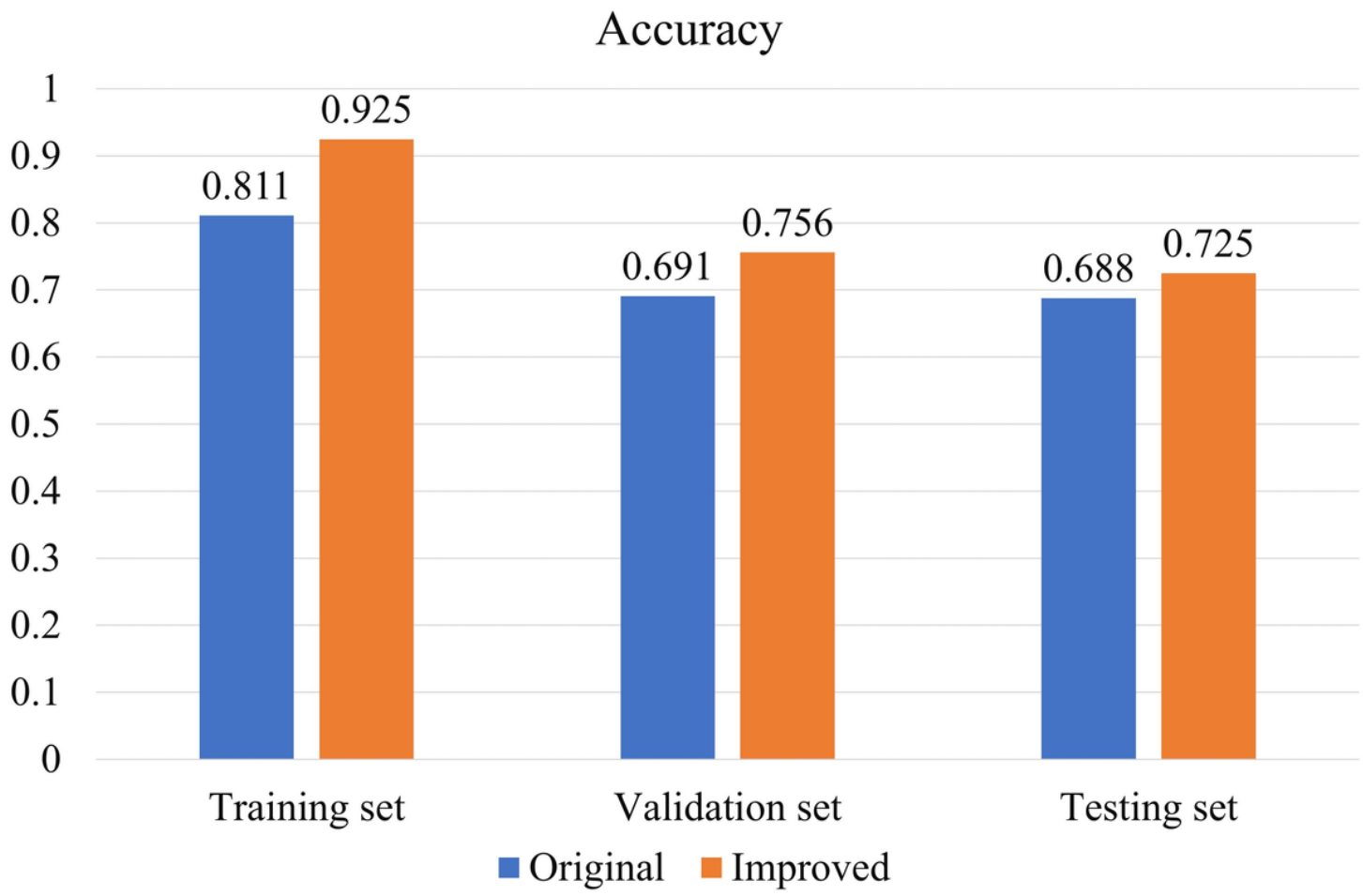


Figure 16

Comparison of the accuracy of the original and improved VGG16 model