An Efficient Insect Pest Classification Using Multiple Convolutional Neural Network Based Models

Hieu T. Ung · Huy Q. Ung · Binh T. Nguyen.

Abstract Accurate insect pest recognition is significant to protect the crop or take the early treatment on the infected yield, and it helps reduce the loss for the agriculture economy. Design an automatic pest recognition system is necessary because manual recognition is slow, time-consuming, and expensive. The Image-based pest classifier using the traditional computer vision method is not efficient due to the complexity. Insect pest classification is a difficult task because of various kinds, scales, shapes, complex backgrounds in the field, and high appearance similarity among insect species. With the rapid development of deep learning technology, the CNN-based method is the best way to develop a fast and accurate insect pest classifier. We present different convolutional neural network-based models in this work, including attention, feature pyramid, and fine-grained models. We evaluate our methods on two public datasets: the large-scale insect pest dataset, the IP102 benchmark dataset, and a smaller dataset, namely D0 in terms of the macro-average precision (MPre), the macro-average recall (MRec), the macro-average F1-score (MF1), the accuracy (Acc), and the geometric mean (GM). The experimental results show that combining these convolutional neural network-based models can better perform than the state-of-the-art methods on these two datasets. For instance, the highest accuracy we obtained on IP102 and D0 is 74.13% and 99.78%, respectively, bypassing the corresponding state-of-the-art accuracy:
67.1% (IP102) and 98.8% (D0). We also publish our codes for contributing to the current research related to the insect pest classification problem.

**Keywords** insect classification, CNN-based methods, ensemble methods

### 1 Introduction

Nowadays, agriculture plays an essential role in many developing countries worldwide, especially in Southeast Asia. It reckons for a substantial portion of each country’s GDP and employs a vital part of its workforce. Among those countries, Vietnam and Thailand become two of the world’s largest agricultural exporters. When rice is one of the most well-known agrarian products exported, other kinds of commodities, including coffee, cocoa, maize, fruits, and vegetables, contribute to the region’s GDP. For instance, palm oil is one of the leading agricultural products for both two ASEAN countries, Indonesia and Malaysia.

Insect pests are well-known to be the most threatened to crops and agricultural products. Crop areas such as rice and wheat are easily affected by insect pests, causing a heavy loss to the crop owner. For this reason, protecting crops and agricultural products from insect pests becomes a must action in different ASEAN countries for keeping and increasing the volume and the quality of yearly agricultural products exported. Among that, insect identification is needed for early pest forecasting to prevent further crop damaged. Manually identifying insect pests on a large farm by expert human resources is time-consuming and expensive. Nowadays, following the popularity of high-quality image capture devices and the achievements of machine learning in pattern recognition, an automated image-based insect pest recognition system is promising to reduce the labor cost and do this task more efficiently. There have been multiple difficulties in extracting useful features for insect classification problems using images. It is challenging to derive discriminative features from the insect image for classification since there are many pest species and variants of their size and shapes. Most recent works applied traditional machine learning methods using hand-crafted features such as GIST, HOG, SIFT, and SURF as proposed in \cite{15, 6, 11, 2} respectively. However, the hand-crafted features lack representation for the large-scale variant shapes of various objects.

Convolutional Neural Networks (CNN) based features are recently very adaptive to different specific computer vision tasks. With the success of the CNN-based features in many classification tasks, especially on the ImageNet Large Scale Visual Recognition Competition or ILSVRC\cite{18}, there have been several works using CNN-based features in the insect pest classification problem. Wu and colleagues \cite{25} had proven CNN-based features could be more efficient than hand-crafted features in this task. Secondly, metamorphosed insects transform by many distinct stages (e.g., egg, larva, pupa, and adult) in...
their life. Also, they may have significant extensive inter-species similarities. For each species, an effective method has to capture features for representing large-scale variant shapes. There is no previous research addressing this problem on insect recognition. In bird breed or car model classification, this problem is solved using a fine-grained image classification method (29, 3). Fine-grained image classification uses the discriminative features extracted from informative regions of the object and encoding them into vectors for classification.

This paper proposes several methods to handle the problems mentioned above in the insect classification problem.

1. Firstly, we apply a CNN model with an attention mechanism to make the feature extractor focusing on the insects in the input image. The attention mechanism is necessary since the captured images of insects in the crop often include a complex background containing leaf, dust, branch, etc.
2. Secondly, we utilize a multi-scale CNN model to capture the features of size-variant insects.
3. Thirdly, we apply a multi-scale learning method for fine-grained image classification to address the high inter-class similarity problem.
4. Finally, we use the soft-voting ensemble method to combine these models to improve the performance.

The rest of the paper can be organized as follows. We briefly present all related works in the insect classification problem in Section 2 and describe our proposed techniques in Section 3. We compare our methods with previous techniques and show the experimental results and our discussion in Sections 4 and 5. The paper ends with our conclusion and future works.

2 Related Work

Recent years have witnessed the notable performance of deep learning methods in different object recognition problems. One of the well-known competitions related to object recognition is the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where all participants competed with each other for object detection and image classification at a large scale. Among mixed approaches at ILSVRC, convolutional neural networks, including Inception (20) and ResNet (7), became the winners of the ILSVRC competition. Other works can be found at (14, 13, 22).

For the insect classification task, Cheng et al. (4) presented a new approach using deep residual networks to enhance the performance of the crop pest classification problem based on pest images with complex farmland background. The proposed technique could achieve a classification accuracy of 98.67% for ten classes of crop pest images, better than a plain deep neural network, AlexNet. Liu and colleagues (10) proposed a new approach for localizing pest insect objects in pest images using saliency detection algorithms and then used a deep convolutional neural network classifying agricultural
pest insects. The experimental results showed this technique could bypass the previous methods in terms of a mean Accuracy Precision (“mAP”), which was 0.951. Wang et al. [24] investigated an insect pest classification problem using deep convolutional neural networks based on crop pest images. They compared the performance of two selected deep neural networks, LeNet-5 and AlexNet, and measure the effects of both convolutional kernels and the number of chosen layers on the final classification accuracy in various experiments. Thenmozhi and co-workers [21] studied the crop pest classification problem using Convolutional Neural Networks and measured the performance of the proposed techniques and several pre-trained deep learning architectures, including AlexNet, ResNet, GoogLeNet, and VGGNet, on three different datasets (NBAIR, Xie1, and Xie2). The experimental results showed that the proposed technique could outperform other chosen pretrained methods.

Wu et al. [25] presented a newly large-scale benchmark dataset for insect pest recognition (IP102). This dataset has more than 75,000 images belonging to 102 categories, where there are about 19,000 annotated images with bounding boxes for object detection. The authors applied hand-crafted (GIST, SIFT, and SURF) and CNN-Based (extracted by AlexNet, VGGNet, GoogLeNet, and ResNet) features to measure the corresponding performance of these methods in the dataset IP102. Ren and colleagues [17] proposed the feature reuse residual network (FR-ResNet) for the insect pest recognition problem, combining features from the initial layer of a residual block with the residual signal and stack the feature reuse residual block to create the proposed network. The experimental results on the dataset IP102 illustrated the improved performance compared to the previous methods. Liu et al. [9] also designed a new residual-based block called deep multi-branch fusion residual network (DMF-ResNet) to learn multi-scale representation. It combines basic residual and the bottleneck residual architecture into the residual module with multiple branches. The outputs of these branches can be concatenated and fed into a new module to recalibrate response adaptively and then model the relationship between these branches. Deep multi-branch fusion residual networks had been created by stacking these blocks and applying them for insect pest classification. They measured the performance of the proposed method and compared it with other state-of-the-art approaches. The experimental results illustrated the enhancement of the proposed technique. Other works related can be found in more details at [1, 12].

3 Methodology

In this section, we present different approaches using retention attention networks (RANs), feature pyramid networks (FPNs), multi-branch and multi-scale attention networks (MMAL-Nets), and the ensemble technique (ET) for improving the performance of the insect pest classification problem. It is worth noting that these methods have specific advantages. For example, the RANs focus on the most crucial region, FPNs efficiently solve small-scale object prob-
lems. In addition, MMAL-Nets practically enhance the fine-grained classification problem for recognizing similar objects. Finally, the ensemble technique can help combine different weak classifiers to create more efficient algorithms having better performance.

3.1 Residual networks

Residual networks or ResNet \cite{he2015deep} was proposed by He et al. in 2015. The network could help avoid the vanishing gradient problem in training deep neural networks by presenting the skip connection technique among layers. Consequently, the gradient can easily flow back to the input, and the network’s weights can be updated. Recently, residual networks can be built by stacking multiple residual blocks (as depicted in Fig.1) to create hugely deep neural networks (maybe up to 1000 layers) depending on each problem.

3.2 Residual attention networks

Wang and colleagues presented residual attention networks \cite{wang2015residual} (RAN) for image classification problems by adding attention mechanism in CNNs to help the networks to decide which location in the image needs to be focused on. Furthermore, one could build the networks by stacking multiple attention modules.
that generate attention-aware features to guide the feature learning. As a result, the residual attention networks showed their efficiency at the early stage when surpassing all state-of-the-art image classification methods. In general, each attention module can be constructed by two branches: a trunk branch and a mask branch. The “trunk branch” performs features extraction, and it can be adapted to any network structure. In this work, we use the pre-activation residual block for the features processing branch. The “mask branch” is used for learning attention masks that softly weigh output features. This mask can be used as a feature selector during the feed-forward phase. Attention module (as depicted in Fig.2) output with residual-attention learning can be formulated by

$$H_{i,c}(x) = (1 + M_{i,c}(x)) \times F_{i,c}(x),$$  

(1)

where $x$ is the input, $i$ ranges over all spatial positions, and $c$ is the index of the channel ($c \in 1, ..., C$). Also, $M(x)$ is the mask branch output and $F(x)$ is the original feature by the trunk branch.

3.3 Feature pyramid networks

Recognizing objects with highly variant scales is challenging in computer vision. Similarly, in insect classification, the scale of the insect in a captured image is usually small. One way to address this problem is to change the size of a single input image with different scales for constructing a pyramid of multiple input images. However, it makes designed networks bigger and requires much memory and longer training time than using a single input image. Feature Pyramid networks[8] (FPN), is an alternative way to construct a pyramid of features with a small extra cost. FPN is a features extractor with architecture composed of a bottom-up and a top-down pathway. The bottom-up pathway
Insect Pest Classification Using Multiple CNN Based Models

is the usual feed-forward computation of the backbone Convolutional Neural Networks; ResNet is selected in our work. The top-down pathway constructs higher resolution features by up-sampling features maps from higher pyramid levels. These features are then added element-wise with features from the bottom-up pathway via lateral connections (as shown in Fig. 3 and Fig. 4). Finally, we apply a $3 \times 3$ convolution on each merged map to generate the final feature map. This filter reduces the aliasing effect of up-sampling. As a result, final feature maps have the exact spatial sizes and the same numbers of channels. In our classification model, after generated all pyramid features, we apply global average pooling on each feature map. Then we feed them into the classifier to conduct the final probability distribution result.

3.4 Multi-branch and multi-scale attention learning network

Fine-grained image classification is a sub-field of object classification where the classifier distinguishes between visually highly similar objects. The purpose is to make the model focusing on details from coarse level features to fine level features to discriminate similar objects. Many types of research on fine-grained classification methods reached the state-of-the-art on many fine-grained benchmark datasets. In this work, we apply the multi-branch and multi-scale attention learning networks (MMAL-Net) \cite{27} for fine-grained image classification on our pest classification task.

The key of the fine-grained classification is to identify informative regions in an image accurately. Usually, we need to localize the object and discriminate...
parts by drawing bounding boxes by hand. In MMAL-Net, we do not have to do extra annotations, object localization, and multiple discriminate part localization being done automatically by two modules with only the category labels: Attention object location module (AOLM) and attention part proposal (APPM) module. MMAL-Net has three branches in the training phase: a raw branch, an object branch, and a parts branch, all of them using the same ResNet-50 as the features extractor and dense layers as the classifier. In the raw branch, the networks mainly study the overall characteristics of the object. Then the AOLM needs to obtain the object’s bounding box information with the help of the features maps of the raw image from its branch (as visualized in Fig. 5). Thus, the accuracy of object localization is achieved by only using category labels. After obtaining the object’s bounding box, we crop the input image following by the bounding box’s coordinates to get the finer scale of the object image, and it can be used as the input of the object branch. Finally, the object branch can learn to obtain the final classification result with the input containing structural features and the fine-grained features of the object.

Additionally, with the feature maps from the object branch, APPM proposes several part regions of the object, which can be used as the input for the parts branch (as shown in Fig. 6). The part images cropped from the object image can train the networks to learn the fine-grained features of different parts in different scales. In the testing phase, the part branch can be disabled, and the final result can be obtained by ensembling the logits from the local branch. We combine the logits from the raw branch and local branch to obtain the final result in our work.

3.5 Ensemble method

Ensemble learning is a machine learning technique that combines accuracy-low models to obtain more accurate predictions. There are many ways to combine models; in this work, we combine the multiple model’s predictions. Combining
the models’ prediction results can reduce its variance and the generalization error. Specifically, soft voting is one of simple, fast, and reliable ensemble method. The soft-voting method is that we take the sum of all the member models’ prediction results on each sample, and then we divide by the number of ensemble members. The final result is the class with the highest probability. Suppose there are $m$ members model and a classification task with $n$ labels. $P_{ij}$ is the predicted probability of model $i = 1, ..., m$ for label $j = 1, ..., n$. One can calculate the ensemble result as follows:

$$P_j = \frac{\sum_{i=1}^{m} P_{ij}}{m},$$  

where $P_j$ is the predicted probability of class $j$.

4 Experiments

This section presents our experiments and compares the proposed approach with the state-of-the-art methods related to the main problem. We evaluated our proposed models on two datasets, i.e., IP102 and D0, and measure the
corresponding performance with the previous methods using standard evaluation metrics, including precision, recall, F1-score, accuracy, and geometric mean score.

4.1 Datasets

We evaluated our proposed method on two datasets. The first dataset is IP102, which is a large-scale benchmark dataset presented in [25]. It contains 75,222 images of 102 insect pest species. This dataset has some challenges practically. Firstly, several classes have highly intra-class variances, as shown in Fig. 7(a). Secondly, there are images captured of the damaged crop as shown in Fig. 7(b). Thirdly, there are images including small-scale insects on the noisy backgrounds as shown in Fig. 7(c). Finally, there is a significant imbalance among the number of samples in classes.

The second dataset is D0, presented in [26]. It contains 4,508 images belonging to 40 insect pest species captured in the natural environment. Some examples are shown in Fig. 8.

4.2 Experiment settings

According to [25], IP102 is partitioned into three subsets: a training set of 45,095 images, a validation set of 7,508 images, and a testing set of 22,619 images. We applied this setting in our experiments. For D0, we arbitrarily partitioned it into three subsets, i.e., a training set, a validation set, and a testing set, with a ratio of 7 : 1 : 2.

We applied pre-processing steps on the input image of the size $h \times w$, where $h$ and $w$ are its height and width, respectively. Firstly, we resized the input image into $h' \times w'$ in which the aspect ratio of the original image is kept. First, the smaller value between $h$ and $w$ is resized to 256. Then, the larger value is assigned by multiplying it with the ratio of the bigger value and the smaller value. Secondly, we applied the random crop data augmentation with the window size of 256 $\times$ 256 on the training phase to address the over-fitting problem. Finally, we applied the center crop method with the same window size as the training phase in the testing phase.

The settings of all RAN, FPN, and MMAL-Net are set at Table 1 according to [23, 8, 27], respectively. We used a pre-trained ResNet-50 on ImageNet to initialize trainable weights in FPN and MMLA-Net since they utilize ResNet-50 as the feature extractor. For RAN, we initialized the weights arbitrarily.

In the training phase, we used categorical cross-entropy as the cost function. We utilized the Adam optimizer with the initial learning rate of $10^{-4}$, and the $\beta_1$ and $\beta_2$ coefficients are 0.9 and 0.999, respectively. We used exponential decay for scheduling the learning rate with a decay rate of 0.96. The mini-batch size and the maximum number of training epochs are set as 64 and 100, respectively. The training phase is stopped when the performance on
Insect Pest Classification Using Multiple CNN Based Models

Fig. 7: Examples for challenges in IP102. Column (a) presents three different worm species, but they are hard to be distinguished. Column (b) shows examples of low-scale insects. Column (c) gives examples of damaged crop fields.

The validation set doesn’t improve after ten epochs. We applied the dropout technique with the drop rate of 0.5 to address the over-fitting problem. To use the ensemble method on RAN, FPN, and MMAL-Net, we trained them in the same training set and tested by voting their predictions.

4.3 Evaluation metrics

We evaluated our proposed models with several suitable metrics for the imbalance among classes in IP102 and D0. The metrics consist of the macro-average precision (MPre), the macro-average recall (MRec), the macro-average F1-score (MF1), the accuracy (Acc), and the geometric mean (GM). To treat the
Table 1: General training settings for each network

| Model/ Hyper-Parameters | ResNet50 | RAN | FPN | MMAL-Net |
|-------------------------|----------|-----|-----|-----------|
| Learning rate           | 0.0001   | 0.1 | 0.0001 | 0.001 |
| Batch size              | 64       | 32  | 32  | 6         |
| Optimizer               | Adam     | SGD | Adam | SGD       |
| betas                   | (0.9, 0.999) | (0.9, 0.999) | (0.9, 0.999) | (0.9, 0.999) |
| momentum                | tum = 0.9 | tum = 0.9 | tum = 0.9 | tum = 0.9 |
| Scheduler               | Exponential | MultiStep | Exponential | MultiStep |
| decay rate              | 0.96     | 0.1 | 0.96 | 0.1 |
| Weight decay            | 0.00001  | 0   | 0.00001 | 0.00001 |
| Dropout                 | 0.3      | 0   | 0   | 0         |
| Maximum epochs          | 100      | 100 | 100 | 120       |
| Input Size              | 224 × 224 | 224 × 224 | 224 × 224 | 448 × 448 |

classes equally important, we computed the recall for each category, then took an average of them to obtain MRec as follows:

\[ \text{Rec}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \]  \hspace{1cm} (3)

\[ \text{MRec} = \frac{\sum_{c=1}^C \text{Rec}_c}{C} \]  \hspace{1cm} (4)

where \( C \) is the number of classes. \( \text{TP}_c \) and \( \text{FN}_c \) stand for the true positive and the false negative of the \( c \)-th class respectively. Similarly, we computed \( \text{Pre}_c \) and \( \text{MPre} \) as follows:

\[ \text{Pre}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \]  \hspace{1cm} (5)

\[ \text{MPre} = \frac{\sum_{c=1}^C \text{Pre}_c}{C} \]  \hspace{1cm} (6)

where \( \text{FP}_c \) stands for the false positive of the \( c \)-th class. MF1 is the harmonic mean of MRec and MPre as follows:

\[ \text{MF1} = 2 \frac{\text{MPre} \cdot \text{MRec}}{\text{MPre} + \text{MRec}} \]  \hspace{1cm} (7)

Acc is computed by the true positive value among all classes as follows:
Table 2: The comparison among different proposed models on IP102.

| Model/Metric   | Acc   | MPre  | MRec  | MF1  | GM   |
|---------------|-------|-------|-------|------|------|
| ResNet-50     | 70.79 | 62.89 | 65.71 | 63.89| 59.70|
| RAN           | 62.82 | 55.46 | 57.38 | 56.09| 51.95|
| FPN           | 70.42 | 62.52 | 64.74 | 63.27| 59.59|
| MMAL-Net      | 72.15 | 62.63 | 69.13 | 64.53| 58.43|
| Ensemble model| **74.13** | **65.72** | **70.74** | **67.65** | **62.52** |

\[ \text{Acc} = \frac{TP}{N} \]  
(8)

where \( N \) is the number of samples. GM is calculated based on the sensitivity of each class (denoted as \( S_c \)). \( S_c \) and GM are as follows:

\[ S_c = \frac{TP_c}{TP_c + FN_c} \]  
(9)

\[ \text{GM} = \prod_{c=1}^{C} \sqrt[\text{C}]{S_c} \]  
(10)

GM equals 0 if and only if one of \( S_c \) equals 0. To avoid this problem, we replaced \( S_c \) of 0 by 0.001.

4.4 Results

We conducted experiments to compare our proposed models and ResNet-50 as a baseline. Table 2 presents the results of those models on IP102. Among those single models, MMAL-Net achieves the best performance on Acc, MPre, MRec, and MF1, which is better at 1.36 percentage points of Acc than ResNet-50. However, the GM score of MMAL-Net is slightly lower than those of FPN and ResNet-50. It implies that the predictions of MMAL-Net on the minor classes are less accurate than those on the major classes. Besides, FPN and ResNet-50 yield comparable results. RAN achieves the lowest results. Combining RAN, FPN, MMAL-Net, and ResNet-50 by the ensemble method performs the best, better 1.98 percentage points than MMAL-Net.

Similarly, we conducted experiments to compare between those models on D0. Table 3 presents the experiment results. Overall, the results on D0 are significantly better than those on IP102. Among the single models, MMAL-Net again yields the best results. FPN and RAN have slightly lower performance than ResNet-50. RAN performs the worst. The ensemble model of RAN, FPN, and MMAL-Net achieves the highest performance. RAN performed significantly worse than other models on both IP102 and D0. It is probably because the pre-trained ResNet-50 did not initialize trainable weights of RAN on the training phase.
Table 3: The comparison among different proposed models on D0.

| Model/Metric   | Acc  | MPre | MRec | MF1  | GM   |
|----------------|------|------|------|------|------|
| ResNet-50      | 99.34| 99.12| 99.20| 99.14| 99.09|
| RAN            | 93.27| 92.93| 93.57| 93.06| 92.65|
| FPN            | 99.23| 99.08| 99.09| 99.07| 99.06|
| MMAL-Net       | 99.56| 99.48| 99.50| 99.48| 99.46|
| Ensemble model | 99.78| 99.66| 99.71| 99.68| 99.65|

Table 4: The comparison between our proposed models and the previous works. (EM: ensemble method)

| Dataset Name  | Name     | Method            | Acc  | MF1  | GM   |
|---------------|----------|-------------------|------|------|------|
| IP102         | Wu et al. [25] | ResNet-50       | 49.4 | 40.1 | 31.5 |
|               | Ren et al. [17] | FR-ResNet        | 55.2 | 54.1 | -    |
|               | Liu et al. [9] | DMF-ResNet       | 59.1 | 58.1 | -    |
|               | Ours      | MMAL-Net         | 72.2 | 64.6 | 58.4 |
|               | Nanni et al. [12] | + CNNs + EM      | 61.4 | -    | -    |
|               | Ayan et al. [1] | CNNs + EM        | 67.1 | 65.8 | -    |
|               | Ours      | RAN + FPN + MMAL-Net + ResNet50 | 74.1 | 67.7 | 62.5 |
| D0            | Xie et al. [26] | MLLF + MKB      | 89.3 | -    | -    |
|               | Ours      | MMAL-Net + ResNet50 | 99.6 | 99.5 | 99.5 |
|               | Thenmozhi and Reddy [21] | CNNs    | 96.0 | -    | -    |
|               | Ayan et al. [1] | CNNs + EM       | 98.8 | 98.8 | -    |
|               | Ours      | RAN + FPN + MMAL-Net + ResNet50 | 99.8 | 99.7 | 99.7 |

We compared our proposed method with the previous methods as shown in Table 4. For IP102, we compared with ResNet-50 implemented in [25], and some variants of ResNet, i.e. FR-ResNet and DMF-ResNet, proposed in [17] and [9], respectively. The results show that our MMAL-Net outperforms those models. In addition, our ensemble models of RAN, FPN, and MMAL-Net are significantly better than the ensemble methods proposed in [12] and [1]. For D0, our proposed models are better than those proposed in [26] and [1].

One can see that our implemented ResNet-50 is significantly better than the one implemented in [25]. The main difference between the two models is that we applied the random crop augmentation technique and the Adam optimizer while [25] did not utilize the augmentation technique and use the Stochastic Gradient Descent optimizer.

Tables 5 and 6 show the list of top 10 classes having the lowest accuracy using ResNet-50 on the dataset IP102 and D0, respectively. As visualized in Figs. 9, 10, and 11, the performance of MMAL-Net and our ensemble method is much better than the corresponding performance using ResNet-50 for all top 1, 3, and 5 accuracy in these two datasets. Between MMAL-Net and the proposed ensemble method, our ensemble model could achieve a better performance in these 10 classes in the dataset IP102. One can see similar behaviors in the dataset D0, as depicted in Figs. 12, 13, and 14.
Table 5: The top 10 classes having the lowest accuracy using ResNet-50 on the dataset IP102.

| Number | Species                                      | Accuracy(%) from ResNet-50 |
|--------|----------------------------------------------|-----------------------------|
| 1      | Therioaphis maculata Buckton                 | 16.67                       |
| 2      | Beet fly                                     | 25.0                        |
| 3      | Green bug                                    | 25.51                       |
| 4      | Polyphagotars onemus latus                   | 26.92                       |
| 5      | Large cutworm                                | 27.03                       |
| 6      | Rice shell pest                              | 27.64                       |
| 7      | English grain aphid                          | 32.49                       |
| 8      | white margined moth                          | 33.33                       |
| 9      | Bird cherry-oataphid                         | 34.27                       |
| 10     | Mango flat beak leafhopper                   | 35.71                       |

Fig. 9: The performance of ResNet-50 with the top 10 classes mentioned in Table 5 in the dataset IP102.

4.5 Visualization with Grad-CAM

This section presents visualization for our proposed models to show where those models focus on the input image to make the predictions. We utilized the Gradient-weighted Class Activation Mapping (Grad-CAM) proposed in [19]. In object classification, Grad-CAM commonly uses the computed gradient of a given target class flowing through the final convolutional layer of the feature extractor part to produce class activation maps (CAMs). Our paper visualized CAMs using the gradient flowing through the last feature extractor block of each proposed model.
Fig. 10: The performance of MMAL-Net with the top 10 classes mentioned in Table 5 in the dataset IP102.

Fig. 11: The performance of our proposed ensemble method with the top 10 classes mentioned in Table 5 in the dataset IP102.

Fig. 15 shows Grad-CAMs produced by ResNet-50, RAN, FPN, and MMAL-Net using the input images in IP102 and their correct classes. They show that MMAL-Net performs the best on focusing the insects in the input images even those insects are small such as in image 2. ResNet-50 and FPN seem to per-
Table 6: The top 10 classes having the lowest accuracy using ResNet-50 on the dataset D0.

| Number | Species                                      | Accuracy(%) from ResNet50 |
|--------|----------------------------------------------|---------------------------|
| 1      | Nilaparvata lugens Stl                       | 84.62                     |
| 2      | Pieris rapae Linnaeus                       | 85.71                     |
| 3      | Dolycoris baccarum Linnaeus                  | 88.24                     |
| 4      | Dryocosmus Kuriphilus Yasumatsu             | 90.00                     |
| 5      | Halyomorpha halys Stl                       | 90.00                     |
| 6      | Luperomorpha suturalis Chen                 | 90.00                     |
| 7      | Riptortus pedestris Fabricius               | 90.91                     |
| 8      | Laodelphax striatellus Fall                 | 91.67                     |
| 9      | Chauliops fallax Scott                      | 92.31                     |
| 10     | Plutella xylostella Linnaeus                | 92.31                     |

Fig. 12: The performance of ResNet-50 with the top 10 classes mentioned in Table 6 in the dataset D0.

form the same, which correctly focus on the region containing the insects. On the other hand, RAN seems to focus on large and less accurate areas to make predictions.

5 Conclusion and Future Works

In this paper, we have investigated different CNN-based methods, i.e., Residual-Attention Network (RAN), Feature Pyramid Network (FPN), and multi-branch and multi-scale attention network (MMAL-Net) for recognizing insect pests. Among these methods, MMAL-Net can achieve the best accuracy of 72.15%
Fig. 13: The performance of MMAL-Net with the top 10 classes mentioned in Table 6 in the dataset D0.

Fig. 14: The performance of our proposed ensemble method with the top 10 classes mentioned in Table 6 in the dataset D0.

and 99.56% on two datasets IP102 and D0, respectively. Furthermore, we visually validated that our models focused on the correct region, even on the input images of low-scale insects or a noisy background. With the combination of chosen models by the ensemble technique, we can obtain the better
accuracy of 74.13% and 99.78% on IP102 and D0, respectively, and bypass the state-of-the-art methods related to the insect pest classification problem on these two datasets. For contributing to the research community, we publish all source codes associated with this work at https://github.com/hieuung/Improving-Insect-Pest-Recognition-by-Ensembling-Multiple-Convolutional-Neural-Network-based-Models.

We aim to consider utilizing variants of CNNs to solve challenges in insect pest classification and applying efficient data augmentation methods in future work.

References

1. Ayan E, Erbay H, Varçın F (2020) Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks. Computers and Electronics in Agriculture 179:105809
2. Bay H, Tuytelaars T, Van Gool L (2006) Surf: Speeded up robust features. In: European conference on computer vision, Springer, pp 404–417
3. Chen Y, Bai Y, Zhang W, Mei T (2019) Destruction and construction learning for fine-grained image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
4. Cheng X, Zhang Y, Chen Y, Wu Y, Yue Y (2017) Pest identification via deep residual learning in complex background. Computers and Electronics in Agriculture 141:351–356
5. Cortes C, Vapnik V (1995) Support vector machine. Machine learning 20(3):273–297
6. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05), IEEE, vol 1, pp 886–893
7. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. CoRR abs/1603.05027, URL http://arxiv.org/abs/1603.05027
8. Lin T, Dollár P, Girshick RB, He K, Hariharan B, Belongie SJ (2016) Feature pyramid networks for object detection. CoRR abs/1612.03144, URL http://arxiv.org/abs/1612.03144
9. Liu W, Wu G, Ren F (2020) Deep multi-branch fusion residual network for insect pest recognition. IEEE Transactions on Cognitive and Developmental Systems
10. Liu Z, Gao J, Yang G, Zhang H, He Y (2016) Localization and classification of paddy field pests using a saliency map and deep convolutional neural network. Scientific reports 6:20410
11. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2):91–110
12. Nanni L, Magnolo G, Pancino F (2020) Insect pest image detection and recognition based on bio-inspired methods. Ecological Informatics p 101089
13. Nguyen BT, Dang-Nguyen D, Tien DX, Phat TV, Gurrin C (2018) A deep learning based food recognition system for lifelog images. In: Marsico MD, di Baja GS, Fred ALN (eds) Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2018, Funchal, Madeira - Portugal, January 16-18, 2018, SciTePress, pp 657–664, DOI 10.5220/0006749006570664, URL https://doi.org/10.5220/0006749006570664
14. Nguyen DMH, Vu HT, Ung HQ, Nguyen BT (2017) 3d-brain segmentation using deep neural network and gaussian mixture model. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp 815–824, DOI 10.1109/WACV.2017.96
15. Oliva A, Torralba A (2001) Modeling the shape of the scene: A holistic representation of the spatial envelope. International journal of computer vision 42(3):145–175
16. Peterson LE (2009) K-nearest neighbor. Scholarpedia 4(2):1883
17. Ren F, Liu W, Wu G (2019) Feature reuse residual networks for insect pest recognition. IEEE Access 7:122758–122768
18. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, et al. (2015) Imagenet large scale visual recognition challenge. International journal of computer vision 115(3):211–252

19. Selvaraju RR, Das A, Vedantam R, Cogswell M, Parikh D, Batra D (2016) Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. CoRR abs/1610.02391, URL http://arxiv.org/abs/1610.02391

20. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

21. Thenmozhi K, Reddy US (2019) Crop pest classification based on deep convolutional neural network and transfer learning. Computers and Electronics in Agriculture 164:104906

22. Ung HT, Tien DX, Phat TV, Nguyen TT, Nguyen BT (2020) Vietnamese food recognition system using convolutional neural networks based features. In: Nguyen NT, Hoang BH, Huynh C, Hwang D, Trawinski B, Vossen G (eds) Computational Collective Intelligence - 12th International Conference, ICCCII 2020, Da Nang, Vietnam, November 30 - December 3, 2020, Proceedings, Springer, Lecture Notes in Computer Science, vol 12496, pp 479–490, DOI 10.1007/978-3-030-63007-2_37, URL https://doi.org/10.1007/978-3-030-63007-2_37

23. Wang F, Jiang M, Qian C, Yang S, Li C, Zhang H, Wang X, Tang X (2017) Residual attention network for image classification. CoRR abs/1704.06904, URL http://arxiv.org/abs/1704.06904

24. Wang R, Zhang J, Dong W, Yu J, Xie CJ, Li R, Chen T, Chen H (2017) A crop pests image classification algorithm based on deep convolutional neural network. Telkomnika 15(3)

25. Wu X, Zhan C, Lai YK, Cheng MM, Yang J (2019) Ip102: A large-scale benchmark dataset for insect pest recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

26. Xie C, Wang R, Zhang J, Chen P, Dong W, Li R, Chen T, Chen H (2018) Multi-level learning features for automatic classification of field crop pests. Computers and Electronics in Agriculture 152:233–241

27. Zhang F, Zhai G, Li M, Liu Y (2020) Three-branch and mutil-scale learning for fine-grained image recognition (tbmsl-net). arXiv preprint arXiv:200309150

28. Zhang J, Ji R, Yuan X, Li H, Qi L, et al. (2011) Recognition of pest damage for cotton leaf based on rbf-svm algorithm. Nongye Jixie Xuebao=Transactions of the Chinese Society for Agricultural Machinery 42(8):178–183

29. Zheng H, Fu J, Zha Z, Luo J (2019) Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. CoRR abs/1903.06150, URL http://arxiv.org/abs/1903
