A concatenated approach based on transfer learning and PCA for classifying bees and wasps

Jiahong Zhang1*, Hongxiang Guo2, Zekai Chen3#

1School of Mathematics and Statistics, Yunnan University, Kunming, Yunnan, 650500, China
2School of Information and Electronics Engineering, Beihang University, Beijing, 100000, China
3School of Automation, Beijing Institute of Technology, Beijing, 100000, China
#These authors contributed equally.
*Corresponding author’s e-mail: 20181910158@mail.ynu.edu.cn

Abstract. Convolutional neural networks (CNNs) are a type of deep model that can accomplish image classification tasks resulting in outstanding performances. From recognizing handwritten numbers to tachyarrhythmia, researchers and engineers have used CNNs for classification tasks in various domains. However, there lacks a comparison and analysis of the performances of different models in classifying insects. Previous methods are currently limited to using only one particular type of network in finishing one specific task. This approach fails to use the strengths of different models. In this paper, a concatenated model is developed, which is able to take advantage of the strengths of multiple CNNs. The performances of VGG, ResNet, XceptionNet, and EfficientNet in classifying bees and wasps are firstly compared. Then, outputs of the penultimate layers in the four networks are merged, which is similar to encoding. In addition, we use principal component analysis (PCA) to decrease computation cost and increase explanatory. Finally, to achieve the final classification, a deep neural network (DNN), which takes the concatenated outputs and four labels separately as input data and output, is structured. In comparison to the traditional classification methods, the proposed model achieves high-level performance and better explanation. Furthermore, it is worth remarking that this paper discovers the encoding characteristics of VGG, ResNet, XceptionNet, and EfficientNet.

1. Introduction

Convolutional Neural Networks have been used in the real-world environment in a variety of domains, including image classification, object detection, and semantic segmentation. When accomplishing image classification tasks, the network classifies an object by extracting the features of the image and learning them. However, learning the comprehensive features of input images is a highly challenging task because a single model is unable to focus on all details. Most of the current approaches utilize only one kind of network for a particular task. However, such approaches fail to maximize the potentials of CNNs because of the limited amounts of features for learning. In this paper, a concatenated model is developed in which the learning abilities of multiple networks are combined and analyzed. To begin the development, we first compare the performances of VGG [1], ResNet [2], XceptionNet [3], and EfficientNet [4] in classifying bees and wasps. Then we merge the penultimate outputs of VGG, ResNet, XceptionNet, and EfficientNet. In the following, PCA [5] technique is adopted to decrease computation
cost and extract valid information. Finally, a DNN model in which the merging outputs function as the input data is trained to classify bees and wasps. It should be explained that some significant but latent observations are found while developing the model, which leads us to analyze the encoding characteristics of VGG, ResNet, Xception, and EfficientNet.

The main contributions of this work can be summarized as follows:

- We evaluate the performances of VGG, ResNet, Xception, and EfficientNet in classifying bees and wasps according to some evaluation metrics and outcomes of Grad-CAM[6] and saliency maps [7].
- We compare the encoding features of VGG, ResNet, Xception, and EfficientNet, which can provide meaningful theoretical hints and enrich the relevant literature for better comprehending the differences among the above networks.
- We develop a high-performance concatenated model.
- We identify some possible future works can be done to improve the performance.

The rest of this paper is organized as follows: We evaluate the performance of VGG, ResNet, XceptionNet, and Efficient-Net on classifying bees and wasps using the visualization of the outputs of the penultimate layers (Grad-CAM and Salient map) in Sect.2. In Sect.3, the details of the concatenated approach, including PCA dimension reduction, autoencoding, and DNN training, are illustrated. Finally, Sect.4 concludes this paper.

2. Model formulation

In this study, we construct a conceptional end-to-end system that includes feature extraction, dimension reduction, and final classification. The developed model mainly consists of two parts:

1) Pretrained model (VGG, ResNet, XceptionNet, and EfficientNet) with transfer learning for feature extraction;
2) Encode boosting model in which PCA is adopted to decrease computation cost, and DNN is structured for final classification.

The overall framework for the proposed model is shown in figure 1.

![Figure 1. The overall framework of the proposed model.](image-url)
We elaborate the detailed formulation of the model in the following content.

2.1. Transfer learning for four pretrained models
The first part of the concatenated model is composed of 4 pretrained networks with transfer learning, which can be functioned as a feature extractor. To be more specific, the composition of this part is VGG, ResNet, XceptionNet, and EfficientNet. The reason for choosing the above four CNNs is mainly that each model can be characterized by different factors such as linearity and complexity. Moreover, we note that the attention mechanisms of the four models are very different in the process of extracting input image features. Below are brief introductions of the four original networks.

• VGG network was proposed by Simonyan and Zisserman in 2014 [1]. Compared to the prior-art configurations, VGG shows a significant improvement by increasing the depth of the network using small convolution filters (3 × 3). With the assistance of the network, Simonyan and Zisserman secured the first and second positions in ImageNet Challenge 2014, respectively.

• As the network goes deeper, it is difficult to train attributes to gradient vanishing or gradient exploding. To alleviate the complexity of training [2], Microsoft research team developed the Residual Network. By adopting the formation of the residual block, ResNet solves the exposed degradation problem, which refers to the phenomenon that the accuracy degrades rapidly as the depth of the network increases.

• Xception model was developed on the basis of an Inception network, also known as GoogLeNet [3]. The intuition behind Xception model is to replace the convolutions in inception modules with depthwise separable convolutions. The replacement resulted in a significant improvement on large image classification problems compared to IncetiponV3. Since the number of parameters in InceptionV3 and Xception share the same order of magnitude, the improvement is more convincible.

• Tan proposed the EfficientNet after discovering that balancing network depth, width, and resolution can lead to better performance [4]. By scaling all the dimensions of depth, width, and resolution using a simple but highly effective compound coefficient, EfficientNet outperforms previous CNNs in a series of classification tasks.

In order to implement transfer learning to the four models, we remove the fully connected layers and manipulate the dimension of the outputs. Furthermore, the number of neurons of the penultimate layer in VGG, ResNet, XceptionNet, and EfficientNet is rescaled to 500 for the purpose of comparing the results under the same standard.

2.2. Encode boosting model
The second part is the encode boosting model, whose functions are re-extracting valid information, reducing dimension, decreasing computation cost, and completing final classification. The experiment is conducted to verify the utility of encode boosting approach, during which we compare the encoding results of VGG, ResNet, XceptionNet, and EfficientNet. First, we manipulate the dimension of outputs for the four networks from 500 to 2. Subsequently, the outcomes of the manipulation are visualized in a 2D coordinate through which we discover the dissimilar coding features of the four networks. The observation leads to the development of encode boosting model.

The first stage of encoding boosting model is to extract valid information. For each of the four networks, we reduce the dimensions of the outputs using PCA technique. Then we sort the characteristic vectors in the order of the highest explanatory to the lowest. After that, starting from the vector with the highest explanatory, we successively get the characteristic vectors from the sorted sequence until the accumulated explanation rate reaches 95%.

The second stage is merging the processed output. The stage results are taken as the input data by the DNN model for classifying bees and wasps. When merging the outputs, this paper utilizes the concatenate method, which takes the outputs evenly and mixes them with identical weight. The structure of the deep neural network is shown in figure 2.
In order to increase the robustness of the DNN, Dropout regularization [8] and Batch-Normalization [9] are adopted. Additionally, we use the L1 and L2 integrated regularization [10] to avoid overfitting problems.

3. Experiments
This paper utilizes a dataset of bees and wasps from Kaggle to evaluate the proposed model. Meanwhile, experiments are carried out on the same data to compare the performance among five models (four original models and one concatenated model). All experiments are conducted on colab platform, which normally contains a Tesla T4 GPU with CUDA 10.1, and two Intel Xeon 2.20GHz processors under a Linux virtual machine. The program codes of data preprocessing and modeling are written in Python 3.6 and Tensorflow 2.4.0, which can be made available on Github.

3.1. Dataset
The Kaggle dataset consists of hand-curated, closed-up photos of 4 categories: bees, wasps, others, and insects (apart from bees and wasps). The dataset can be accessed from the link Bee vs. Wasp. The proportion of training set to the validation set to test set is 7:2:1. Since the challenge issued by Kaggle is primarily to distinguish bees from wasps, the distribution of the original dataset is disproportional. We recreate the dataset based on the original one by increasing the numbers of data for the categories other than bees and wasps. This lead to an even distribution for all categories. The structure of the re-created dataset is shown in table 1.

| Categories | Bee | Insect | Other | Wasp |
|------------|-----|--------|-------|------|
| Train      | 2229| 1708   | 3658  | 3461 |
| Validation | 636 | 487    | 1045  | 988  |
| Test       | 318 | 244    | 522   | 494  |

3.2. Evaluation metrics
The evaluation metrics we used are accuracy, precision, specificity, and AUC [11].

\[
Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)
\]

\[
Precision = \frac{TP}{TP + FP} \quad (2)
\]

\[
Specificity = \frac{TN}{FP + TN} \quad (3)
\]

\[AUC = Area\ Under\ ROC\ Curve\]
Accuracy, precision, specificity, and AUC are adopted for evaluation purposes. All the metrics are calculated using true positive (TP) predictions, true negative (TN) predictions, false positive (FP), and false negative (FN) predictions. The accuracy is the rate of true predictions to all predictions made. Precision is regarded as the ratio of the true positive predictions to all the positive predictions. The ratio of true negative predictions to the sum of false positive and true negative predictions is defined as specificity. AUC refers to the area under the receiver operating characteristic (ROC) curve, which measures the quality of a binary classification model.

3.3. Experimental results and analysis

3.3.1 Transfer Learning Model

We first evaluate the performance of four original models, the outcomes of the experiments are shown in table 2, 3, and 4.

Table 2. Precisions of original models on different categories.

| Model     | Average | Bee    | Insect  | Other   | Wasp   |
|-----------|---------|--------|---------|---------|--------|
| VGG       | 0.9448  | 0.9393 | 0.9495  | 0.9811  | 0.9093 |
| EfficientNetB5 | 0.9661  | 0.9623 | 0.9538  | 0.9867  | 0.9615 |
| ResNet50  | 0.9426  | 0.9419 | 0.9224  | 0.9775  | 0.9284 |

Table 3. Specificities of original models on different categories.

| Model     | Average | Bee    | Insect  | Other   | Wasp   |
|-----------|---------|--------|---------|---------|--------|
| VGG       | 0.9810  | 0.9549 | 0.9918  | 0.9905  | 0.9566 |
| EfficientNetB5 | 0.9895  | 0.9905 | 0.9918  | 0.9934  | 0.9825 |
| ResNet50  | 0.9819  | 0.9857 | 0.9865  | 0.9886  | 0.9668 |

Table 4. Total performances of original models.

| Model     | ACC    | AUC    | Precisions\* | Specificity\* |
|-----------|--------|--------|--------------|---------------|
| VGG       | 0.9448 | 0.9393 | 0.9495       | 0.9811        |
| EfficientNetB5 | 0.9661  | 0.9623 | 0.9538       | 0.9867        |
| ResNet50  | 0.9426 | 0.9419 | 0.9224       | 0.9775        |

\* Precision and specificity there were calculated using micro average among categories.

Predicting results are visualized to analyze the performances and characteristics of every model. We use Saliency map and Grad-CAM to scrutinize the unique quality of each pixel and to see where the model concentrates on. The outcomes are shown in figure 3. The heatmaps are shown in 225×225. From the top row to the bottom row are heatmaps of EfficientNetB5, ResNet50, VGG, and XceptionNet.
Figure 3. Concentration of models, each image group is showed in order original image, saliency map and Grad-CAM.

From which we are able to notice that:

- EfficientNetB5 captures more details than other models, given the fact that the model has multiple concentrations on the object.
- ResNet50 outperforms other models in identifying the overall details. To elaborate, the Grad-CAM of ResNet50 has the largest area of the red zone, which indicates the scale of details the model concentrates.
- VGG has the most accurate concentration. For example, when dealing with an image of a bee, the Grad-CAM of VGG suggests that the model concentrates precisely on the bee’s body. Moreover, when detecting the “other” category, VGG concentrates precisely on the mountain.
- Grad-CAM might not be useful for evaluating the XceptionNet because although the outcomes of Grad-CAM are ambiguous, XceptionNet performs the best in accomplishing all classification tasks.
- XceptionNet classifies objects referring to some local details and environment.

To study the encoding features of VGG, ResNet50, XceptionNet, and EfficientNetB5, visualization and dimension reduction are combined. To elaborate, we obtain the outputs of the penultimate layers of VGG, ResNet, XceptionNet, and EfficientNet at first. Then the dimensions of those outputs are reduced using t-SNE[12], PCA, and ISOMAP [13]. The results of the visualization are shown in figure 4.
Figure 4. The dimension reduction results in different data sets, which arranged in four models, EfficientNet, ResNet, VGG and Xception, and three dimensional reduction methods, in the order of T-SNE, Isomap and PCA. And different colours indicate different classes of samples.

From the visualizations and comparisons, several observations are noticed:

- VGG possesses a sparse encoding feature.
- The origins of features extracted by all the models are likely from the Euclidean space rather than Manifold Space.
- The results after PCA manipulation indicate that certain categories have strong tendency towards some particular directions, which implies the dominate direction for classification of autoencoding process.

3.3.2 Encode boosting model
In this section, we compare the concatenated model with four original networks in classifying bees and wasps. Below are several charts that show the performances of VGG, ResNet, XceptionNet, EfficientNet, and Concatenated Model. Note that the best performance is shown in boldface and the second best performance is shown in italics.

| Model | Average | Bee | Insect | Other | Wasp |
|-------|---------|-----|--------|-------|------|
| VGG   | 0.9448  | 0.9393 | 0.9495 | 0.9811 | 0.9093 |
From the chart, certain conclusions can be drawn:

- EfficientNetB5 outperforms other models in classifying most of the categories.
- Encoder boosting model surpasses VGG, ResNet, and XceptionNet in ACC, Precision, and specificity.
- Encoder boosting model outperforms other models in classifying objects that belong to the “other” category, which indicates that the proposed model has great potential in classifying objects with complicated features.

4. Conclusion

This paper develops a concatenated model for classifying bees and wasps. The key components of the model are summarized as follow:

- The concatenation of the penultimate outputs of VGG, ResNet, XceptionNet, and EfficientNet.
- The dimension reduction process using PCA which distills the major features from encodings and make the encode boosting model more efficient.

Extensive experiments show that the proposed model achieves promising performance. Furthermore, some observations are found while developing the model. Due to the discoveries, we analyze the autoencoding features for the four separate models. Following the main idea of this work, future researches can be expanded in certain aspects:

- In the transfer learning part, each model can be assigned to focus on different parts of a classification task.
- For concatenation, extracted features can be selected according to the performances of models in the previous transfer learning part.
- Normalization can be adopted to the encoding model to reduce the impact of the encoding sparsity of VGG in an effective manner.

References

[1] Karen S. and Andrew Z.. (2014) Very Deep Convolutional Networks for Large-scale Image Recognition., arXiv preprint arXiv:1409.1556.
[2] Kaiming H. et al. (2016) Deep Residual Learning for Image Recognition., Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

[3] François C.. (2017) Xception: Deep learning with Depthwise Separable Convolutions, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition., pp. 1251–1258.

[4] Mingxing T. and Quoc L.. (2019) Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks, International Conference on Machine Learning. PMLR., pp. 6105–6114.

[5] Svante W., Kim E., and Paul G.. (1987) Principal Component Analysis, Chemometrics and Intelligent Laboratory Systems 2.1-3, pp. 37–52

[6] Ramprasaath R S. et al. (2017) Grad-cam: Visual Explanations from Deep Networks via Gradient-based Localization, Proceedings of the IEEE International Conference on Computer Vision., pp. 618–626

[7] Karen S., Andrea V.i, and Andrew Z.. (2013) Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, arXiv preprint arXiv:1312.6034

[8] Nitish S. et al. (2014) Dropout: a Simple Way to Prevent Neural Networks from Overfitting, The Journal of Machine Learning Research 15.1, pp. 1929–1958

[9] Sergey I. and Christian S..(2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, International Conference on Machine Learning. PMLR., pp. 448–456

[10] Kwangmoo K., Seung-Jean K., and Stephen Boyd. (2007) An Interior-point Method for Large-scale L1-regularized Logistic Regression, Journal of Machine Learning Research 8.Jul, pp. 1519–1555.

[11] Andrew P B.. (1997) The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms, Pattern Recognition 30.7, pp. 1145–1159

[12] Laurens Van der Maaten and Geoffrey H.. (2008) Visualizing Data Using T-SNE., Journal of Machine Learning Research 9.11

[13] Joshua B T., Vin De Silva, and John C L.. (2000) A Global Geometric Framework for Nonlinear Dimensionality Reduction, Science 290.5500, pp. 2319–2323.