

Fish Species Classification (Group 28)

Weijie Ju
A53311270

w3ju@eng.ucsd.edu

Xiao Lu
A53323673

xil078@eng.ucsd.edu

Ke Wang
A53300464

kew004@eng.ucsd.edu

Abstract

Fish image classification is a fined-grained image classification challenge that can be more challenging than other categories because of the low-quality and small-scale data. In this paper, we implemented several machine learning methods including Convolutional neural network, and improve the performance using Transfer learning with a 97% accuracy.

Keywords: Fine-grained image classification, CNN, Transfer learning

1. Introduction

Fine grained object categorization is a challenging problem. It aims to classify subclasses of a certain category and recognize objects that belong to the subclasses. Recent research mostly focuses on the classification among species such as cats and dogs, or well-distinguished categories, for instance, plane and flowers. The difficulty of this problem is mainly from the similarity of features among different breeds of the same species.

Fish resources are of vital importance to the marine ecosystem. However, the current illegal and disorderly fishing seriously threatens the marine ecological environment and the global sustainable supply of seafood. Researches on fish classification pave way for real-time fish image processing and will be of practical use in the development of underwater digital hardware equipment.

2. Literature Review

2.1. Fish population monitoring

The monitoring of fish population has many benefits. For example, it helps scientists to learn the habits of fish, their population dynamics and also in a higher view, provide vital importance in making strategies about species protection. Traditionally, fish population monitoring was to tag and track some of certain species manually. However, this method is costly and low-efficient because it requires the

observation of divers, underwater fish counting tools and echo sound equipment.

2.2. Machine learning in image classification

Considering the development of image classification research, it is a reasonable choice to use techniques such as SVM and CNN models, especially CNN models because it has shown its capability in image classification for many categories. A lot of CNN models such as vgg16, ResNet are proven to be robust in image classification.

A lot of research has focused on automatic fish monitoring and classification related to video and image processing. For example, in research [5], the scientists applied support vector machine(SVM), Convolutional neural network (CNN) models based on GoogleNet and achieved an over 80% accuracy.

2.3. Transfer learning

The training of a robust and well performed CNN model can be time and energy consuming and requires a large amount of high quality images. However, with the technology constraints, fish image data is usually low quality and small scale. Transfer learning makes use of pre-trained robust CNN models, extracting weights and features from the model, and with fine tuning we could obtain a model works for a similar dataset efficiently.

3. Dataset

The dataset we used is Fish Recognition Ground-Truth data, comes from university of Edinburgh, school of informatics[1]. This dataset is acquired from a live video that includes 27370 verified fish images. The whole dataset is divided into 23 clusters and each cluster is presented by a representative species, which is based on the synapomorphies characteristic from the extent that the taxon is monophyletic[2]. One problem of this dataset is that it is very imbalanced where the most frequent species is about 1000 times more than the least one. We dividing it into two parts: 90 % for training and 10 % for testing. Among the training data, 10 % is also used for validation.

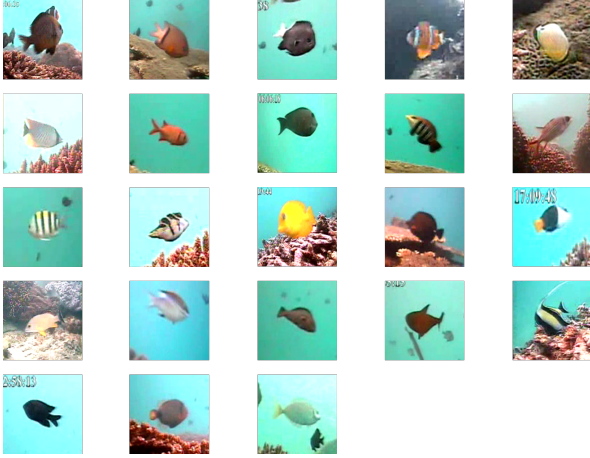


Figure 1. 23 fish species in Fish Recognition Ground-Truth dataset

3.1. Data Preprocessing

Next, let's focus on data preprocessing part. Considering that the dataset we have is imbalanced, and also for the purpose of improving the robustness and generality of our model, we use image augmentation from imgaug library. Augmentation techniques we used includes random rotation, affine transformation, superpixeling, blurring, sharpening, embossing, flipping, adding Gaussian noise and change of contrast to enlarge the dataset (figure 2). After that, we resized all images to $150 \times 150 \times 3$ for consistency. Then, we use different strategy for deep learning models and non-deep learning models. For non-deep learning ones, we spread all pixels into a single vector so that for each image it will have 67500 features. For deep learning models, since we use pre-trained models as a start. Data generator is used to extract features from pre-trained models. After features are extracted, we still need to spread them into a single vector for training purpose. The whole dataset is split that 90 % for training and the rest 10% for testing.

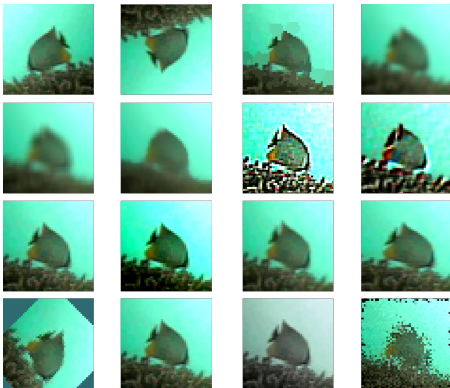


Figure 2. Demonstration of 16 types of data augmentation

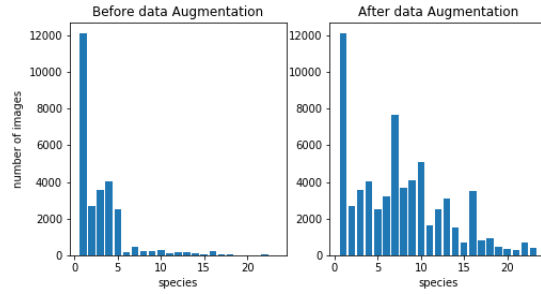


Figure 3. Number of images before/after data augmentation

4. Methods

We applied 6 non-deep learning methods: Naive Gaussian, Random Forest, MLP, KNN, Logistic Regression and SVM. The general process for image classification includes feature detection, feature extraction and feature recognition. After that, we use several pre-trained models like ResNet50 and VGG19 to do transfer learning.

4.1. Non-deep learning Models

4.1.1 Naive Bayesian

Naive Bayes utilizes the Bayes conditional probability model and applies the maximum a posteriori decision rule for class classification.

4.1.2 Random Forest

Random forest is an algorithm that generate small decision trees of randomly chosen subset of data [3]. Each decision tree will generate a classifier based on the subset of data that capture certain feature of the whole dataset and the majority vote is employed to classify a class.

4.1.3 KNN

The k-nearest neighbor classification assigns each object to the class based on the class of its k nearest neighbors [4]. The neighbors of a point is decided by the distance between them and each neighbor might carry different weight in its influence of the point.

4.1.4 Logistic Regression

Logistic regression uses logit function that take the features of dataset as input and output the predicted class [6].

4.1.5 SVM

Supporting vector machine is a technique that outputs a hyper plane that separates the dataset into predefined classes [9].

4.1.6 MLP

A multiple layer perceptron (MLP) consists of a group of feedforward neural network. It has three types of layers: an input layer, hidden layers, and an output layer. Each node is processed through a nonlinear activation function before sent into the next layer, and the weight of the node is updated by gradient descent. The only purpose for using this model is to make future comparisons with deep learning models.

4.2. CNN

Convolutional neural network has been widely applied to the research in fine grained image recognition. It contains an input and output layer, and multiple hidden layers. The hidden layers of convolutional neural network typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers. Although convolutional neural network has been proven to be effective in image classification, to achieve its best performance, a large dataset is required. For example, AlexNet uses 1.2 million high-resolution images for training, and the neural network has 60 million parameters and 650,000 neurons. Compared to the dataset used in AlexNet, the dataset for cat face classification is far too small to train a high-performance model. Moreover, it would take much time to train such a model even given a desired dataset. Hence, instead of extracting features from our model, we use pre-trained models such as VGG16 to extract features and classify images. In this project, for Convolutional Neural Network method, we applied three pre-trained models: VGG16, VGG19, ResNet50 and InceptionV3.

4.2.1 VGG16

VGG16 network architecture was first introduced by Simonyan and Zisserman[8]. This architecture uses small 3×3 convolution filters with increased depth. The number of weight layers in the network is 16. The model contains three fully connected layers, 13 convolutional layers with rectified linear activation unit(ReLU) non-linearity, 5 max pooling layers and the soft-max layer as the final layer. ReLU is a function that returns the value provided as input or 0 if the input is not positive. It is defined as follows:

$$F(x) = \max(x, 0)$$

The softmax function is defined as follows:

$$\sigma(z_i) = e^{z_i} / \sum(e^{z_j})$$

$$i = 1, \dots, k \quad z = (z_1, \dots, z_k)$$

It takes a vector of K real numbers as input and normalize it into probability distribution. As a result, each component

will be in the interval (0,1) and the components will sum up to 1. So the result can be interpreted as the probability of an image be classified into a certain class. In general, VGG16 model achieves a high performance in its accuracy of classification to be 92.7% and it was in the top 5 test accuracy in ImageNet, a dataset that contains 14 million images and 1000 target categories.

4.2.2 VGG19

The architecture of VGG19 is similar to VGG16. VGG19 consists of 19 layers of neural network while VGG16 contains 16.

4.2.3 ResNet

The ResNet architecture was introduced by He Kaiming et al. in 2015[7]. The characteristic of this network is the stack of a specially designed structure named residual block. By introducing residual representation and shortcut connections, it solved the degradation problem of deep networks, and achieved 3.57% error on the ImageNet test set, and was the champion on the ILSVRC 2015 classification task. ResNetV2 refined the structure of residual block to ensure the information can be transmitted to the next block, and got higher accuracy.

In ResNet, each residual block can be defined as:

$$y = F(x, W_i) + x$$

Here $F(x, W_i)$ is the residue function, which is fit by two ReLU layers. If the optimal case for the block is an identical mapping, it is easier to fit F than y by simply setting all weights W_i to zero. In this way, the output of the block can be identical to the input, and the deeper model should produce no higher training error than its shallow counterpart.

4.2.4 InceptionV3

The Inception architecture was introduced by Szegedy et al. in 2015[10]. With several improvement and refinement, the third version Inception V3 achieved an accuracy greater than 78.1% on the ImageNet dataset. The network has 42 layers, including convolutions, average pooling, max pooling, concatenates, dropouts and fully connected layers. Different from other models, it applies different sizes of filters on one layer, and the output is then concatenated and sent to the inception module.

5. Experiment

For CNN model, we construct a three-layer model with dropout to train the fish images for classification. The first layer is fully connected with 512 units, a ReLU activation function and dropout 0.5; the second layer is again fully

References

- [1] B. Boom, P. Huang, C. Beyan, C. Spampinato, S. Palazzo, J. He, E. Beauxis-Aussalet, S.-I. Lin, H.-M. Chou, G. Nadarajan, J. Chen-Burger, J. van Ossenbruggen, D. Giordano, L. Hardman, F.-P. Lin, and B. Fisher. Long-term underwater camera surveillance for monitoring and analysis of fish populations. In *2012 21st international conference on pattern recognition (ICPR 2012)*. Red Hook : Curran Associates, Inc., 2012.
- [2] B. J. Boom, P. X. Huang, J. He, and R. B. Fisher. Supporting ground-truth annotation of image datasets using clustering. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 1542–1545, 2012.
- [3] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.
- [4] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
- [5] M. M. M. Fouad, H. M. Zawbaa, N. El-Bendary, and A. E. Hassani. Automatic Nile tilapia fish classification approach using machine learning techniques. In *13th International Conference on Hybrid Intelligent Systems (HIS 2013)*, pages 173–178, 2013.
- [6] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:2000, 1998.
- [7] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *CoRR*, abs/1512.03385, 2015. [eprint: 1512.03385](#).
- [8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.
- [9] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Process. Lett.*, 9(3):293–300, June 1999.
- [10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions, 2014.

Individual Contributions

- Weijie Ju

Non-deep learning models, paper writing, peer review

- Xiao Lu

CNN models, paper writing, peer review

- Ke Wang

Literature review, Data preprocessing, paper writing, peer review, presentation video editing

Replies to critical reviews

Critical review from team 7

1. Since your dataset is imbalanced, why did you not attempt to use data augmentation techniques before running your experiments?

Response: We tried to find another dataset that is not that imbalanced. However, some is of bad quality that other irrelevant images mixed in and some is over classified that more than 1000 species in total. In final report, we implemented data augmentation so that the dataset is not that imbalanced anymore.

2. How did you select which deep learning models to use?

Response: We referenced from the rank of most successful deep learning models from github and selected several representative ones.

3. Why did you choose a 90:10 split for the training/validation to test? Could this affect your testing results since if we naively predict the most frequent fish then we'd have relatively high accuracy?

Response: Before we did data augmentation, we used this split rate so that enough data could be used for training to make our model more robust. It could affect the testing results but not that severe. After data augmentation we have 65,754 images and the largest species contains about 12,000 images, less than 20% of total data.

4. No conclusion derived

Response: Sorry for the missing part in the presentation. We did not reach the conclusion when we did the presentation. The conclusion part is now included in the final report

Critical review from team 11

1. What are some examples of models that have achieved high accuracy?

Response: As mentioned in the video, models include SVM, self-trained CNN. We did not explain in detail because of time constraints, but I do agree that we should put some illustrative graphs or results in the presentation.

2. First it is stated that data was split into 90% for training and 10% for testing; then it is stated 90% for training and 10% for testing.

Response: Thank you for pointing it out. There was a discrepancy (one team member mixed up validating and testing) and we have corrected it in the report.

3. The dataset was unbalanced but not addressed clearly.

Response: We did not implement data augmentation at first, but we have included it in the final report.

4. Justification for both non-deep learning and deep learning methods should be posted; some methods were not mentioned but the accuracy was posted.

Response: Thank you for pointing it out. We will add them in the report

5. The team should focus on addressing some issues of the dataset rather than testing many machine learning methods. Any created CNN should be reported alongside other models.

Response: The dataset issues have been resolved. As for the latter comment, we believe it is more important to show insightful results instead of piling up all our experiments to the presentation. We make many attempts and it is not realistic to present them all in the video.

6. More metrics should be employed to show the performance of the model.

Response: Interesting point. We might try to work on it in the future. But for now, we have fixed the unbalanced dataset issues.

Critical review from team 39

1. Why did you choose Adam as optimizer, did you try other optimizers like SGD?

Response: Yes we did. The final performance doesn't vary much after 200 epochs, but basically SGD took longer time to converge.

2. The structure on the ppt is kind of messy (separate text and plot).

Response: Sorry about that. Maybe we will do better with a larger page limit of slides next time.

3. Can you explain more about how you deal with unbalance of the dataset?

Response: Please refer question #1 by group 7 and #3 and #6 by group 11.