

Humpback Whale Identification Challenge

Changyu Li, Wenda Chen, Jiayu He and Huang Lin

Abstract—Humpback whale identification challenge project intends to predict the species of a humpback whale using only tail images. This project not only give people an insight of machine learning techniques, but also preserve endangered species. During the process of training validation and testing, we encountered several obstacles such as data mismatch and validation loss increasing. However, the team worked hard and attempted several ways to improve the results such as data augmentation and Grayscale image. Though the final result is far from ideal, Our results revealed the fact that common CNN methods may not be a good fit for this Humpback whale identification challenge.

Our group's Github link is:

<https://github.com/rogerlin123/ECE228Group24>

I. INTRODUCTION

Humpback whale, which takes its common name from the distinctive hump on its back, is a mammal living in oceans around the world. This surface-active species is a favorite of whale watchers as they jump out of the water and slap the surface with their pectoral fins or tails. Humpback whale tail fluke patterns, in combination with varying shapes and sizes of whales dorsal fin and/or prominent scars, are unique. This makes it possible for humpback whales species to be determined by its tail pattern. To aid whale conservation efforts, scientists use photo surveillance systems to monitor ocean activity. They use the shape of whales tails and unique markings found in footage to identify what species of whale theyre analyzing and meticulously log whale pod dynamics and movements. For the past 40 years, most of this work has been done manually by individual scientists, leaving a huge trove of data untapped and underutilized.

The data set used in this project contains thousands of images of humpback whale fluke. The task of this project is to help scientists identify humpback whale species by their tails' image. Convolutional Neural Network is applied in this project. The input to our algorithm is an image of humpback whale tail. Our Convolutional Neural Networks predict five most likely classes of humpback whale type. If one of the predicted classes matched the real humpback whale class, then we marked this prediction was successful.

Though CNN has been proven to be a powerful tool in image classification task, given the relatively imbalanced dataset in this project and more than 3000 classes of whales, it is a very big challenge for CNN model to achieve its high prediction accuracy when it is usually applied in other prediction problems.

We firstly compared the performance between popular model VGG16 and ResNet without doing data preprocessing. Then We optimized the data set using data augmentation

as well as RGB transfer before feeding them to the VGG16 model to see if it made any difference.

II. RELATED WORK

Whale identification falls into the category of image classification. In this section, we will discuss some popular methods used in classification.

In paper [1], the author mentioned that there are two major categories of methods used in image classification: statistical approaches and neural networks.

Principle Component Analysis (PCA) is one of the statistical approaches. Agarwal and some other researchers applied this method in the field of human face recognition[2]. PCA reduces the feature dimensions while extracting the features, thus improving the performance of the classifier.

Another statistical method that is used frequently in classification is SVM_KNN. This method is an improved version of NN classifier, which converts the distance matrix of K neighbors, and applies multi-classes SVM [3].

The methods mentioned above are some popular statistical approaches in machine learning. There are also two popular neural networks, VGG and ResNet. In our project, we used these two networks.

VGG is used by a team in the competition ILSVRC-2014[4]. It won the second price of the competition, showing an excellent performance in image classification task.

Residual Network, also known as ResNet, is introduced by K.He [5]. It is the first algorithm that reaches the human level accuracy in terms of image classification. Its original application is image denoising. It applied residual blocks to approximate a denoised image.

Although our task is image classification, the project's details are quite different from ordinary classification tasks. First, the characteristics of whales' tails are hard to differentiate. For example, it is easier to see the difference between cats and dogs than seeing the difference between whale's tails. Moreover, we need to handle the extremely unbalanced data(for example, some classes only have 1 training sample).

III. DATASET AND FEATURES

We have two datasets with different size. The small dataset has 9850 pictures of whales over 4251 classes. We split the dataset into training set, validation set and test set. Training set is 70%, validation set is 30% and test set has 15610 images.

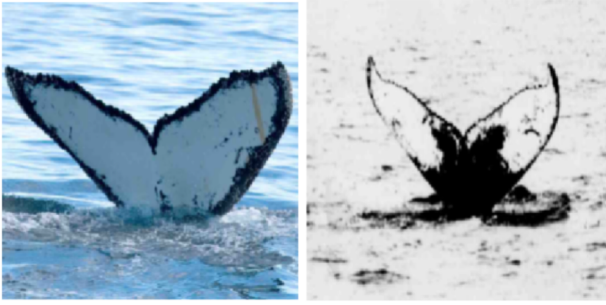


Fig. 1. original training set with different color

Since our training set has greyscaled image as well as RGB images, we converted all the images to greyscaled. These images also have different sizes, so we resized all images to $224 * 224$ to fit our VGG16 model.

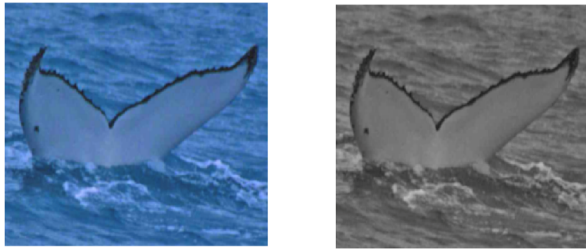


Fig. 2. training set converted to greyscale

By examining the distribution of our dataset, we find that the class labels are very unbalanced. Class with most samples has 810 samples while there are about 3000 classes which only has 1 sample.

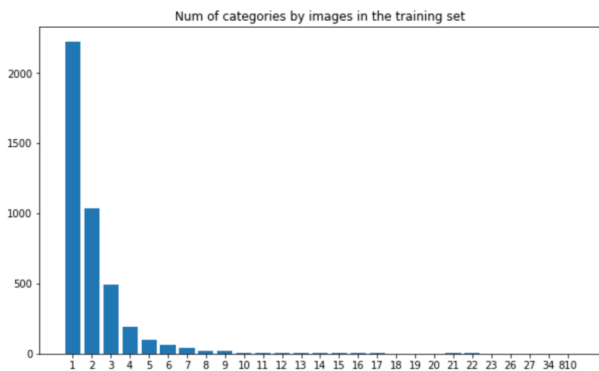


Fig. 3. training set converted to greyscaled

Therefore, we decided to do the augmentation for the training set for optimization. We used crop, flip and rotate method to generate more samples for the minority classes.



Fig. 4. training set images were rotated to generate more images

IV. METHOD

We used Convolutional Neural Network(CNN) for our identification task. Convolutional Neural Network is a special type of Artificial Neural Network(ANN) that is usually used with image identification problems whose inputs are raw images. These inputs images are processed through a series of layers that perform a convolution followed by a set of non-linear operations. Similar to the visual cortex in humans, CNN utilizes the spatial locality from input image. Each neuron in one layer is connected to all neurons in the next layer, therefore CNN can create layers of discrete filters to learn spatial shift invariant features of the image[9].

To balance our training time and performance, we decided to do the transfer learning. The model we chose were VGG16 and Resnet18. We replaced the last several fully connected layers as our own to retrain the parameters. CNN consists of series of modules. We had the convolution module as a sandwich structure: first layer is a convolutional layer, then followed by an activation function layer, finally a max pooling layer. After several convolutional modules, we connected it with several fully connected layers. When an image is imported into a CNN, the convolutional layers will extract its high dimensional features and use it as the input to next layers. Activation functions can simulate non-linear function and use pooling layers to extend the reception field. The fully connected layers are the place where most of the parameters exist. It will finally do the classification and calculate the percentage for each class. Finally we will get a highest percentage label as our predictions class.

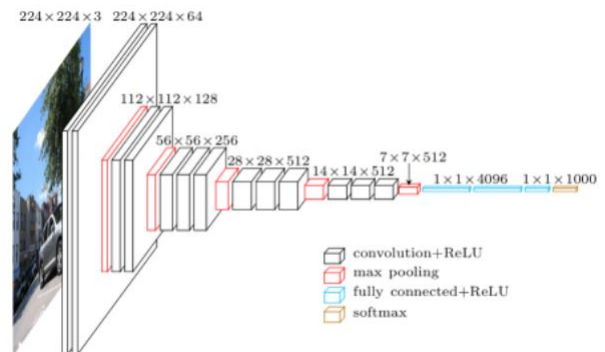


Fig. 5. VGG16 Architecture[1]

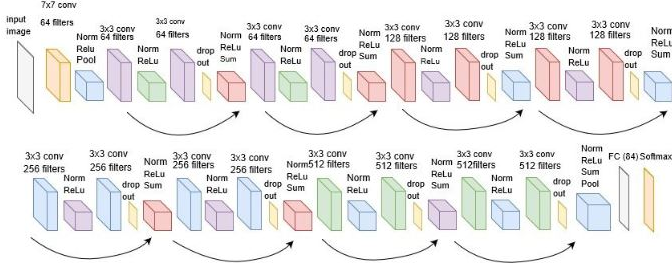


Fig. 6. Resnet18 Architecture[2]

In our method, the image will go through the preprocessing steps and become a 224×224 image with 3 channel. The matrix will then be imported into the first layer of our CNN model. For each convolution layer, input feature map will do convolution with the convolution kernel and generate new feature map, then new feature map will pass reLu activation function, whose function is

$$g(a) = \max(a, 0) \quad (1)$$

where a is the input feature map. The result will pass max-pooling layer, which will take the largest value in $n * n$ area. For fully connected layers, each layer has a weight W and bias b , and the result of the passing the FC and activation $g(x)$ layers with input feature map h is

$$h = g(Wx + b) \quad (2)$$

At the end of the network, we used softmax function to generate the most possible label as our prediction. The function for softmax is

$$\text{softmax}(a)_k = \frac{\exp(a_k)}{\sum_{l=1}^K \exp(a_l)} \quad (3)$$

, where K is the number of classes and a is the input feature map.

To update our parameters in the model, we will use gradient descent to find the updated parameters. This is accompanied by back propagation based on chain rule, which help to find the gradient direction the loss function is going to reduce and make a step towards that direction. The function is as following:

$$w = w + \alpha(d - y)x \quad (4)$$

where α is learning rate, w is weight, d is prediction, y is label, x is input

We chose VGG16 model because it is one of the best models which can extract the CNN features from images. It has a small convolution kernel which can significantly reduced the calculation by increasing the number of feature maps. These also helps increasing the conception field, make the model extract the feature more easily. For Resnet18, it has a special shortcut from previous layers. These structure make the information passed in the network to be as complete as possible. It solve the problem of gradient vanish and gradient explode and make it possible to train deep neural network.

While we use VGG16 as a typical architecture model for CNN as our method, we also try Resnet18 as another option since Resnets residual structure may let it act as a depth scalable architecture, and it may provide different performance on our identification task.

V. EXPERIMENTS, RESULTS AND DISCUSSION

In this section, because our task was not a binary classification task and we had more than 3000 classes, we decided not to present our confusion matrix in the report(the confusion matrix will be 3000 by 3000). Instead, the discussion will be based on cross entropy loss and accuracy.

Our experiment used VGG-16 and ResNet for the classification task. We applied transfer-learning using the pre-trained VGG and ResNet from the pytorch library. We picked the learning rate to be 10^{-3} . The batch size is 16. The baseline model trains for 20 epochs, and the model with data augmentation trains for 60 epochs. We picked these parameters to balance the computational time and performance. Using a learning rate of 10^{-4} would earn us a better result(not by much), however, it would take a much longer time to finish the training. The reason of training baseline for 20 epochs is to eliminate the fluctuation in accuracy. After data augmentation, some of the samples were picked for multiple times, so we train for more epochs. In terms of batch size, a too large batch makes the model less generalized, because the batch contains the characteristic of the training set. As a result, a reasonable batch size is between 16 and 32.

This experiment passed unaugmented RGB images to a pre-trained VGG16 classifier

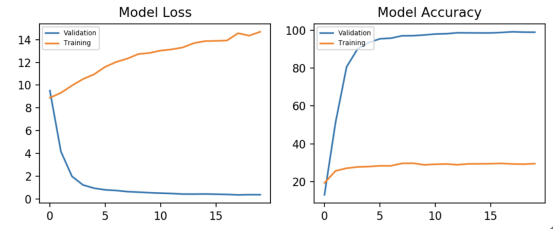


Fig. 7. VGG16 with RGB images, loss: 14.68, accuracy: 29.52%

This experiment passed unaugmented Grayscaled images to a pre-trained VGG16 classifier

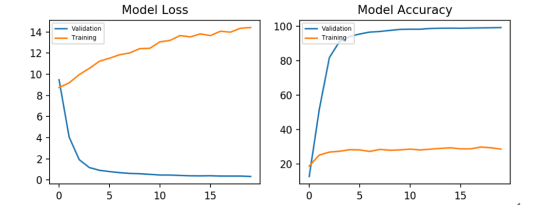


Fig. 8. VGG16 with Grayscaled images, loss: 14.41, accuracy: 28.55%

Our experiment applied a couple of settings. In the first option shown in the graph, we converted the images to RGB

image. In the second option, we converted the graph to a grey scale image. The experimental result shows that the RGB images give a better result(Fig. 7 vs Fig. 8). The reason is that some of the features is related to color. Eliminating the color of the images may make a negative difference in the features. We also tried to apply data augmentation to the method. We picked some transformation from the following list: rotation(+/- 25 degrees), 10% translation along the horizontal axis, 15% translation along the vertical axis, color jitter and Gaussian noise with signal-to-noise-ratio of 10. The experiment showed that affine operation did not help much in terms of improving the experimental result. The reason was that the shape of the whales tail is an important feature. If we apply a translation or a random cropping, then sometimes part of the tail would be moved out of the image edges. As the result shown in (Fig.7) and (Fig.9), we noticed that when using vgg16, an augmentation did not improve the classification accuracy. Moreover, it increased the loss significantly.

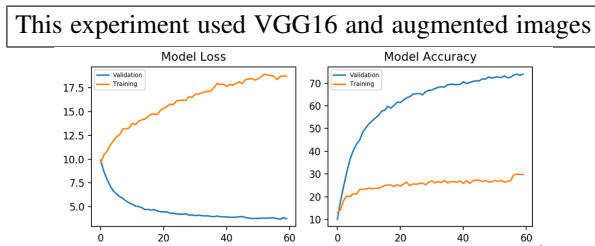


Fig. 9. VGG16 with augmented images, loss: 18, accuracy: 29.77%

When we compare the performance between ResNet(Fig. 10) and VGG16(Fig. 7), we can see that the change of model did not make a huge difference in the experimental result. The reason is that our project may not be suitable for this task. Both VGG-16 and ResNet are deep neural networks, which require a lot of training samples to be utilized. In our dataset, our training set contains around 7000 images, but we need to classify more than 3000 classes. In this case, the traditional machine learning techniques may give a better result compared to the deep network.

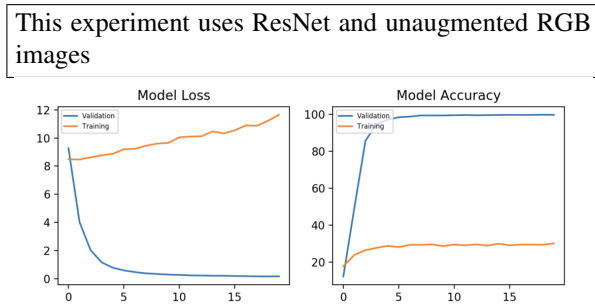


Fig. 10. ResNet with RGB images, loss:11.56 accuracy: 29.87%

VI. CONCLUSION/FUTURE WORK

As we can see from our experiment, two deep neural network didn't perform well on our classification task. The

data preprocessing including grey scale, data augmentation also didn't have a better performance. The most important reason is our method isn't suitable for this training set. Neural network needs large amount of data to train its parameter but we have only 1 samples in several classes. This extremely imbalanced data may not be an ideal training set for a deep neural network task. Some data augmentation process also destroy some of the features of original images. We should improve our data augmentation methods. For example, reduce the rotation angle, or increase the signal to noise ratio to prevent the noise diminishing the features.

For the future work, we may attempt to use traditional methods such as K Nearest Neighbour or Support Vector Machine methods, which is more suitable for classification task with small amount of data. We can also try using PCA or SVM to do the preprocessing and use its high dimensional features as our input to deep neural network.

VII. CONTRIBUTION

Wenda Chen: Helped implement the data preprocessing techniques and adjusted for different hyperparameters.

Changyu LI: Helped building the training process including data loading, training process and evaluation.

Jiayu He: Implemented the baseline training network, implemented the data augmentation and evaluation.

Huang Lin: Implemented the ResNet model, Implemented Grayscale image process for model optimization as well as try different hyper parameters.

REFERENCES

- [1] Jain, Anil K., Robert P. W. Duin, and Jianchang Mao. "Statistical pattern recognition: A review." *IEEE Transactions on pattern analysis and machine intelligence* 22, no. 1 (2000): 4-37.
- [2] J Agarwal, M., Agrawal, H., Jain, N. and Kumar, M., 2010, February. Face recognition using principle component analysis, eigenface and neural network. In 2010 International conference on signal acquisition and processing (pp. 310-314). IEEE.
- [3] Zhang, H., Berg, A.C., Maire, M. and Malik, J., 2006. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) (Vol. 2, pp. 2126-2136). IEEE.
- [4] Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- [5] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- [6] E. Oja, "Principal Components, Minor Components, and Linear Neural Networks", *Neural Networks*, vol. 5, no. 6, pp. 927-936. 1992.
- [7] Davi Frossard, VGG in TensorFlow, 2016, <https://www.cs.toronto.edu/~frossard/post/vgg16/>
- [8] Muhammad Abul Hasan, Fig.2. ResearchGate, https://www.researchgate.net/figure/Proposed-Modified-ResNet-18-architecture-for-Bangla-HCR-In-the-diagram-conv-stands-for_fig1_323063171.
- [9] Convolutional neural network - Wikipedia https://en.wikipedia.org/wiki/Convolutional_neural_network