CS583A: Course Project

Zhipeng Lin

December 2, 2019

1 Summary

Problem descriptions: I pariticipe an inactive with late submission competition of classifying ocean species based on photos. Methodology: I try many architectures The final model I choose is ResNet50, a deep convolutional neural network architecture, which takes 64×64 grey images as input and outputs the class labels. Implementation: I implement the convolutional neural network using Keras and run the code on Colab with Google Compute Engine Backend (GPU). Evaluation metric: Performance is evaluated on the multi-class logarithmic loss. Score and ranking: In the public leaderboard, our score is 4.41216; we rank 746 among the 1049 teams. In the private leaderboard, our score is 4.41558; I rank 778 among the 1049 teams.

2 Problem Description

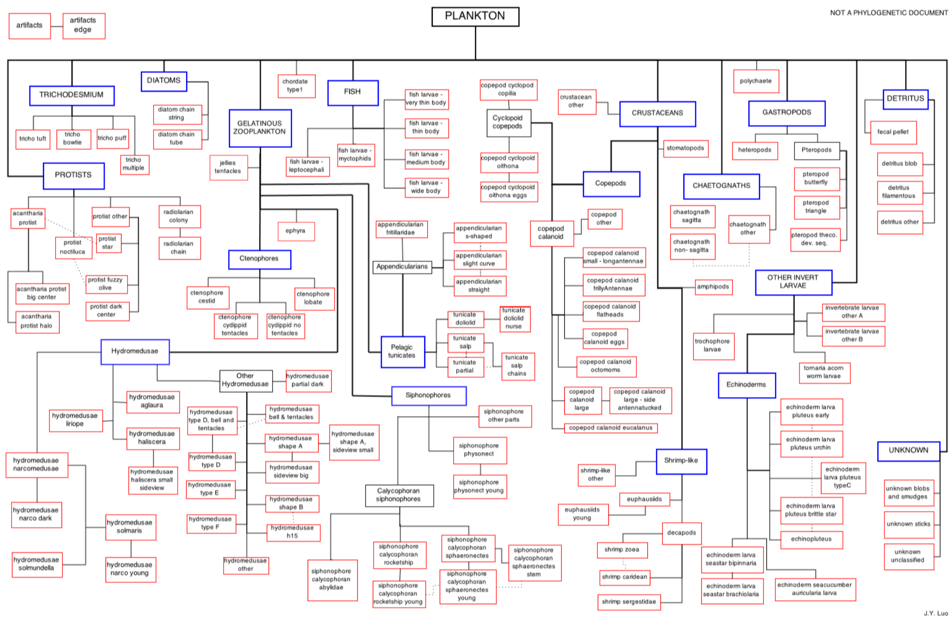
Problem. The problem is to classify species in ocean based on photos. This is a multi classification and image recognition problem. The competition is at `https://www.kaggle.com/c/datasciencebowl/`.

Data. The data are JPG gray-scale images in different size based on different species. The number of training samples is $n = 30336$. The number of testing samples is $n = 130400$. The number of classes is 121. One of the classification is 'Unknown' class. The training set is not balanced.

Challenges. There are many different species, from large fish to small plankton. Some of species are similar, some of species are still unknown. Image cannot represent 3 dimension structure of objects. The same species looks unlike in different direction. The training set is small.The number of common species are larger than rare ones.

3 Solution

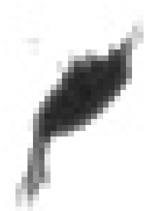
Model. I use and implement 5-layer CNN, ResNet18, ResNet50 . The model I finally choose is the ResNet50 [1], a standard deep convolutional neural network. A description of ResNet is online: `https://en.wikipedia.org/wiki/Residential_network`. The input of standard ResNet50 is 3 channels images. I changed it to 1 channel to match the input of my grayscale image data. In the model output, I changed the number of hidden points in the fully connected layer to 121.



(a) The classification of dataset.

Figure 1: The classifications.

(a)

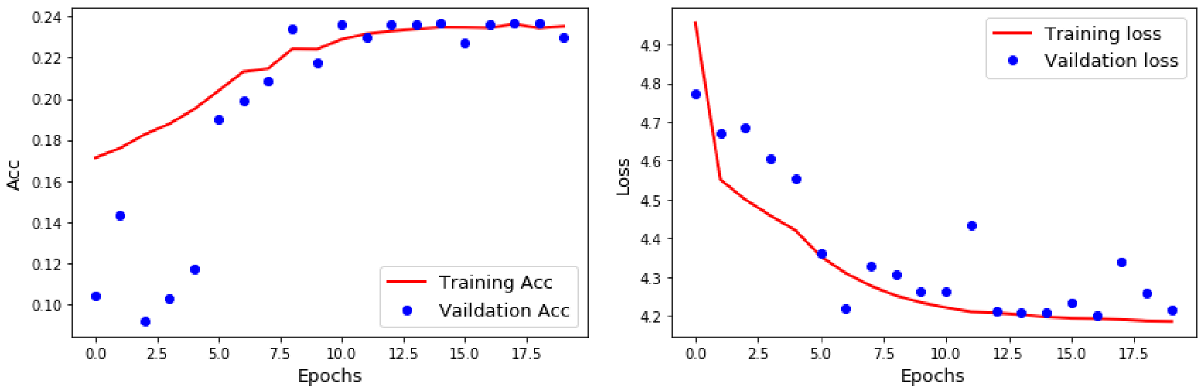


(b)

(c)

(d)

Figure 2: Some examples



(a) The classification accuracy on the training set and validation set.

(b) The loss on the training set and validation set.

Figure 3: The convergence curves.

Implementation. I implement the 5-layer CNN, ResNet18, ResNet50 model using Keras with TensorFlow as the backend. My code is available at `https://github.com/Linzp721/CS583/blob/master/Project/CS583_CourseProject.ipynb`. I run the code on Colab with Google Compute Engine Backend (GPU).

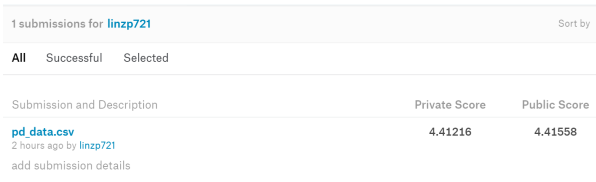
Settings. The loss function is categorical cross-entropy. The optimizer is RMSprop. The learning rate is $1E-3$ The regularization is L2-regularization, rate is $1E-4$ The epochs is 20 The batchsize is 128

Advanced tricks.

- Data augmentation. Feature and line in images are thin and light. First, I threshold the image on its mean value. Then, I dilate the the image. Operation effect is shown in the figures.

- Over-sampling and Under-sampling. Because the data is imbalanced (The number of samples in different categories varies widely. Some categories examples less than 100, others more than thousands), I tried over-sampling and under-sampling so that the number of samples in each category is about 300.

Validation. I partition the training data to 80%-20% for hyperparameter tuning. Figure 3 plots the the convergence curves on 80% training data and 20% validation data. Due to too many classifications, the accuracy becomes very low. So I used top-5 accuracy. From the convergence curves, accuracy increases and loss decreases with epochs. After 10 epochs, the curve gradient starts to decrease.

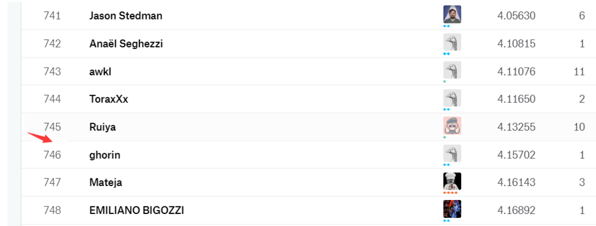


1 submissions for linzp721 — Sort by

All | Successful | Selected

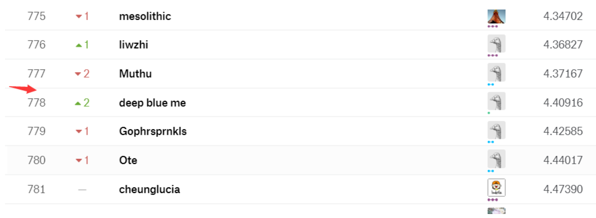
Submission and Description	Private Score	Public Score
pd_data.csv 2 hours ago by linzp721 add submission details	4.41216	4.41558

(a) Public and Private Score.



741	Jason Stedman	4.05630	6
742	Anaël Seghezzi	4.10815	1
743	awkl	4.11076	11
744	ToraxXx	4.11650	2
745	Ruiya	4.13255	10
746	ghorin	4.15702	1
747	Mateja	4.16143	3
748	EMILIANO BIGOZZI	4.16892	1

(b) Public leaderboard.



775	▾1	mesolithic	4.34702
776	▴1	liwzhi	4.36827
777	▾2	Muthu	4.37167
778	▴2	deep blue me	4.40916
779	▾1	Gophrsprnkls	4.42585
780	▾1	Ote	4.44017
781	—	cheunglucia	4.47390

(c) Private leaderboard.

Figure 4: Public and Private Score, Leaderboard

4 Compared Methods

ResNet50. I use the ResNet50 model using Keras withTensorFlow as the backend. Its parameters are pretrained on the ImageNet dataset. The training and validation accuracies are respective 23.52% and 22.97%.The multi-class logarithmic loss value is 4.2044.

ResNet18. I implemented Resnet18 before implementing Resnet50. The residual block uses the Building block structure, while 50 uses the Bottleneck structure The training and validation accuracies are respective 23.48% and 22.04%. The multi-class logarithmic loss value is 4.3041.

5-layer CNN. I implemented a 5-layer convolutional neural network with 2 Dense layers.I apply maxpooling and dropout in this model. The training and validation accuracies are respective 5.16% and 4.39%.The multi-class logarithmic loss value is 4.7826.

Random guess. I use random guess strategy to calucate the baseline. Generate a 121-dimensional onehot vector randomly, and then use multi classification logloss to get the result The multi-class logarithmic loss value is 34.2371.

5 Outcome

I participated in an inactive with late submission competition. In the public leaderboard, our score is 4.41216; I rank 746 among the 1049 teams. In the private leaderboard, our score is 4.41558; I rank 778 among the 1049 teams. My ranking is not displayed directly on leaderborad because of late submission. So I got the ranking based on the historical leaderboard. The screenshots are in Figure 4.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.