Wildlife Classification With Deep Learning Methods

Yiheng Ye, Zhexu Li, Andy Liu,

Background and Motivation

Wildlife conservation is an essential part of environment protection, a diverse wildlife provides balance and stability to the ecosystem. Modern wildlife management requires in-depth understanding about the behavior and diversity of different animals in the habitat, usually through sensors and cameras. Traditionally, such insights were produced by human experts who manually label and study those images. But with more cameras installed, manually labeling that large amount of images generated everyday becomes infeasible. Therefore, it's important to automate the classification process.

According to previous studies, there are several experiments working on the classification of wildlife. Miao et al [4] used several CNN networks on 20 species with 111467 images and get 87.5% best accuracy. Larson [5] also reveals some important aspect in CNN training for wildlife images. Furthermore, Norozzade et al. [6] worked on simplify background of 3.2 million wildlife images and reach a 93.2% accuracy in species classification. In our case, we will work on a relatively smaller data set with more species categories and try to reach an optimal validation accuracy.

In this project, we aim to build an accurate wildlife classification model to solve that problem and free the experts from this tedious task. We plan to construct a deep neural network for the task, and train it on an animal image dataset from kaggle which contains 90 animal species. The resulting model will be helpful for automatic wildlife classification, and it could provide valuable information about characteristics of certain species.

Dataset: Animal Image Dataset

Our dataset is an animal image collection of 5400 images of 90 different animal species[3], the dataset turned out to be tricky to work with. Each category only has 60 images with irregular dimensions, some of them are only 100 * 100 while the others could be 1280 * 720. Originally, we took a naive approach which transforms the data to a regular size, and preprocess the data beforehand to pick the most suitable images. The preprocessed dataset turned out to hurt the performance of the model a lot. To achieve a better performance, we utilized with various augmentation techniques, including flipping, shifting, rotation, binarization, and blurring. We didn't use noising because many images in the dataset are already quite noisy. The augmented dataset contains 10378 images. The augmentation process expanded originally small dataset by a lot, and it will be helpful for us to build a more robust classifier.

One batch grid of images in the original dataset.

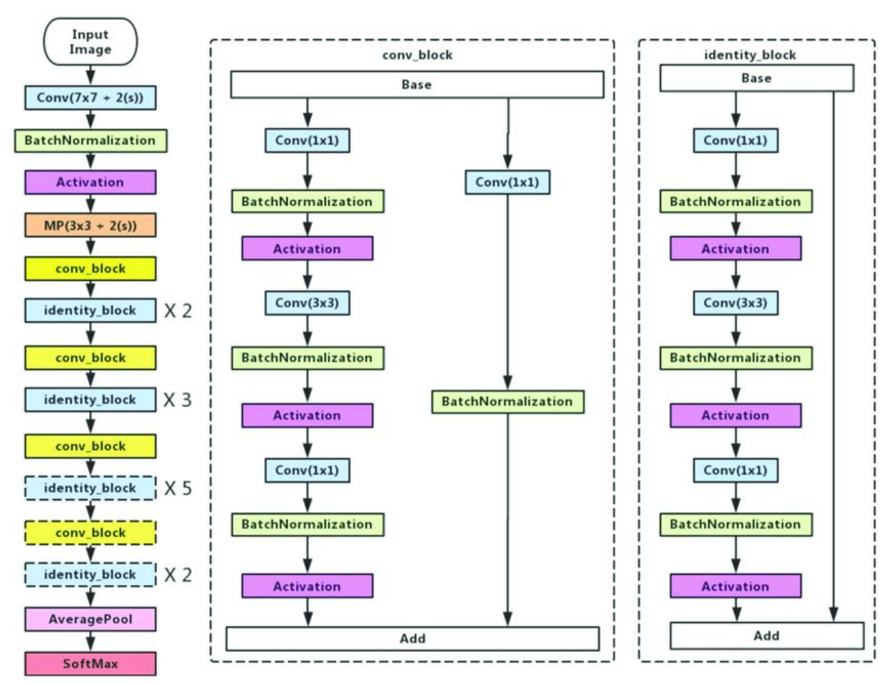
One batch grid of images in the augmented dataset.

Method Applied

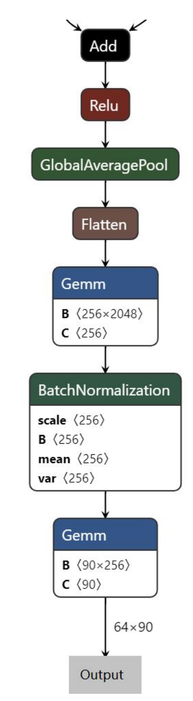
Before we make final decision on which model we are going to use on our data set, we have made different attempts using some common models. We tried naive MLP first, but it only reaches 29% training accuracy after 10 epochs, and then we used pre-trained VGG16, whose performance was not ideal as its validation accuracy only reaches 62%. After that, we tried Resnet50 [2], a 50 layers network which introduced several improvements over the VGG architecture.The most important improvement was the Residual Block, which is defined by:

$$y = F(x, \{W_i\}) + x \tag{1}$$

Where F(x) is approximated by the weight (Wi) layers we have in each block. In our model, one such block includes 2~3 convolutional layers and thus we will use this structure to perform shortcut connection within our neural network.



Resnet50 architecture [1]



Our Improved Architecture

The vanilla Resnet50 structure [1] is shown on the left image above. The Resnet50 model (with pretrained weights) achieved an impressive 85% validation accuracy. We then tried the DenseNet 121, which is a more recent and more complex 121 layers model. The DenseNet 121 model only achieved a validation accuracy of 77%, and the training process was extremely computationally expensive. We also observed some overfitting near the end of training, so we did not try any other more complex model, and decided to stick with ResNet50.

We experimented with various techniques in efforts to optimize the Resnet50 architecture. Since we observed some slight overfitting, we included extra regulations. In the end, the optimal modification we made was to **replace the fully connected layer with our "regulation block" which includes a linear layer to wrap up pretrained weights, a batch normalization and a dropout layer (20%) for better regulation, and a linear layer to generate classifications**. The batch normalization is defined by:

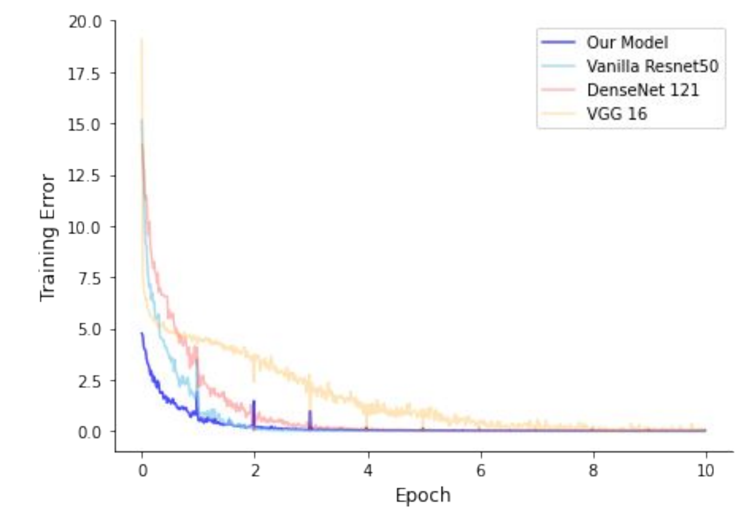
$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{VAR[x^{(k)}]}} \tag{2}$$

Which acts like a regulation term. And because we are doing a classification task, we used Cross-Entropy loss which is defined by:

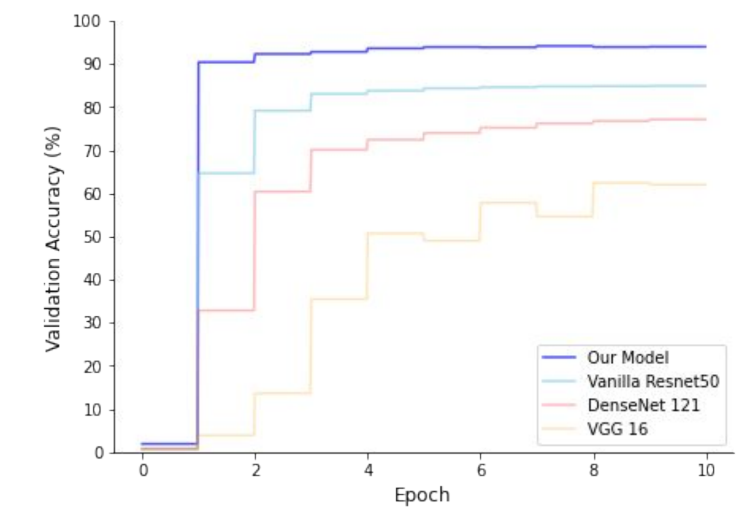
$$Loss = -\Sigma_{c=1}^{M} y_{o,c} log(p_{o,c}) \tag{3}$$

And based on our experiments, our modification **improved our model performance significantly with a little extra computational cost**.

Experiment Results



Training Error



Validation Accuracy

We conducted many experiments using the models mentioned before. All models were trained based on Cross Entropy Loss, using Adam optimizer with the best learning rate selected based on numerous trials, and batch size = 64, and trained for 10 epochs. The experiment results are shown in the plots above, results of MLP were not plotted because it's not meaningful. Based on the Training error vs. Epoch plot on the left we can see our improved model converged fastest among all 4 models. And based on the Validation Accuracy plot on the right, it's clear that our model consistently outperformed all other models by a considerable amount. In fact, our model achieved 93% validation accuracy at the end of training, which was about 8% higher than the vanilla Resnet50, 16% higher than the DenseNet 121, and 31% higher than the VGG 16. Considering our modifications only introduced one extra linear layer (one replaced the linear layer in original network), one batch normalization layer, and one dropout layer, the improvement in validation accuracy is quite impressive considering the very little extra computational cost The improvements could be results of better regulations we introduced, and our model may have striked a good balance between complexity and simplicity, while MLP and VGG was a bit too simple, requiring much more data, and DenseNet was too complex, resulting in serious overfitting.

Conclusion and Future Works

In this project, we experimented with various data augmentation techniques to improve the quality of our animal dataset which is small and contains images of irregular dimensions. We also tried a number of state-of-the-art deep learning architectures, including MLP, VGG, ResNet, and DenseNet, and experimented with different regulation terms to reduce overfitting. After a lot of trials and errors, we discovered the optimal architecture for our task, which was built upon the ResNet50 structure with additional regulation blocks. Our improved model achieved a validation accuracy of 93% after being trained for 10 epochs, which brought considerable improvements over other models, with a little extra computational costs. In the future, we will experiment with more augmentation techniques and regulation terms to try to improve the performance. We will also use techniques like class activation maps to visualize the prediction of our model to better understand its decision processes.

References

[1] Ji, Qingge & Huang, Jie & He, Wenjie & Sun, Yankui. (2019). Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images. Algorithms. 12. 51. 10.3390/a12030051

[2]] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition arXiv:1512.03385

[3] Sourav Banerjee, Animal Image Dataset (90 Different Animals), Kaggle

[4] Miao, Z., Gaynor, K.M., Wang, J. et al. Insights and approaches using deep learning to classify wildlife. Sci Rep 9, 8137 (2019)

[5] Larson, Julia, "Assessing Convolutional Neural Network Animal Classification Models for Practical Applications in Wildlife Conservation" (2021). Master's Theses. 5184

[6] Norouzzadeh, M.,Nguyen A., Kosmala, M. et al. "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning".