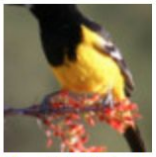
Introduction to Computer Vision
Final Project Report
Gleb Promokhov
Gregory Goh

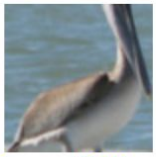
BirdNet

Experiments in classifying birds


Orchard Oriole


Scott Oriole


Ovenbird


Brown Pelican


Whip poor Will


Horned Puffin


Common Raven


White necked Raven

Abstract

Our projects explores how convolutional neural nets can be used for image classification tasks. From a public dataset of 11,788 images of birds belonging to 200 species classes, we built and trained a CNN classifier with comporable classificaiton accuracy. We trained several identical models using the raw images, images with the birds segmented out, and images with just the bird's head to examine how reducing the input data dimensions affect classification accuracy.

Dataset

The Caltech-UCSD Bird-200-2011 dataset [1] includes labeled images of 200 bird species with 11,788 images total. The dataset also includes binary image segmentations for each raw image, cropping the bird itself from the background. Each image is also annotated by crowdsourced participants with coordinates of several points, such as throat and tail, on each image as well as a table of color attributes of each bird. Images were evenly distributed between classes with no imbalance issues. This well put-together dataset has done the majority work of data collecting and cleaning for us.

We trained a CNN model using the raw images and images with binary segmentations applied. To reduce the image even further we realized that as humans we are able to distinguish the bird's species by just looking at their heads. We built a third dataset from the segmented images that cropped out each bird's head by plotting a line between the 'throat' and 'nape', and then setting all of the pixels on the same side as the 'tail' from this line to 0.

Each of these datasets were also augmented by taking each input image and generating several more with various degrees of scaling, zoom, shear transform and horizontal and vertical

flips. This is to help against overfitting by introducing more variation to the training data and capture more of the possible variation in the testing data.

Classifier

There are no formulas or algorithms for building an optimal neural net for each task, leaving us to have to try various combinations of layers and examine the results. We settled on a variation of the popular LeNet CNN [2] that was originally designed for MNIST digit recognition. It consists of an input layer following by 3 repetitions of convolutional, activation and max pooling layers. Each of these convolution layers use an increasingly complex set of kernels to capture more complex features of each image. This is followed by a flatten and dense layer, keeping only the 64 most activated features. The next dropout layer randomly ignores 50% of it's input to avoid overfitting. The final dense layer maps to a 200 length vector of class activations. A full figure of our model is included in the appendix.

Results

Model	Testing Accuracy%	Training Time
Baseline	0.50	N/A
Segmented images	18.75	2:52:53
Segmented, heads only	7.85	2:22:12

We trained our model using these segmented and segmented heads datasets and measured there testing accuracies. We used a 50/25/25% train/test/validation split, normalized each image to 300 by 300 pixels, and train for 100 epochs. The unsegmented dataset training took far too long to complete and did not include those results. Unfortunately the segmented heads dataset saw a much worse accuracy than the segmented images with minimal improvement to training time. More training epochs could boost the segmented heads accuracy, but it is quite likely it will plateau before the segmented model does. Using our segmented dataset, we were able to achieve a 1.45 percent increase in accuracy over one of the baselines in the original Caltech-UCSD study [1].

Discussion and Further Work

We were glad to see these improved results, but there were several questions left about these models we trained. It is difficult to ask questions and query the model about it's properties. Because of the neural network's complicated implementation and structure it was difficult to deduce if the (x, y) position of each pixel was used as a classification feature. This is undesirable since the pixel positions are a property of the photograph, not the bird itself. It is difficult to gain a intuitive understanding of the model as well. A natural question to ask could include "What part of the network learned what Albatross feet look like?". Surely there must be

some collection of weights that represent this neighbourhood of pixels, but neural nets do not organize themselves the way the brain does. While the convolution aspect of our model considers regions of pixels like we would, they are regions convolved with a predefined set of kernels across every pixel of the image. The convolutions that captured the Albatross' foot are going to be spread throughout the network as completely unrecognizable patterns that only the network 'understands'. The brain does a better job of segmenting what we see into logical units such as 'foot', 'beak', and 'wing', which our model has no way to knowing without even more detailed labeling of regions on each bird image. This lack of intuitive direction makes it difficult to query the model and learn in which way to tune parameters, leaving us to an ineffective 'try them all' approach.

Another problem we noticed is how by plotting activations across various layers we observed how many activated pixels were along the 'nape-throat' edge from the segmented heads dataset. This is undesirable since again, the exact location of that edge is a property of the image, not of the bird and shouldn't be used for classification. With the given model accuracies and training times it seems that segmenting just the heads of the birds removes too much information without a corresponding decrease in training time or better accuracy for it to be worth it.

Our model also suffered from the 'vanishing gradient' problem, where the effects felt by backpropagation vanish from the output to the input nodes of the network. Towards the end of training, the segmented heads model saw the average change in weights in the first convolution layer was in the hundreds place. This suggests that the inputted images carried to little information to be distributed as weights for every node, or that the network has to many nodes for the given problem. Resolving this issue requires tuning of the kernel sizes, image input sizes, exact layers used, etc. for more even distribution of weights.

Future work for this project includes an attribute classifier that would be able to classify the birds purely based on its labeled attributes instead of the image itself. As mentioned above, this dataset included crowdsourced labels alongside the images; participants were to label the bird in an image with as many of the 312 attributes as applicable, and how certain they are of their labels. With this data, we hope to create another neural network model that can classify the bird species based on their attributes. Once we have this attribute classifier, we also hope to create an ensemble model that combines the image classifier and attribute classifier to further improve our results.

All of our project code is available at: https://github.com/glebpro/computervisionproject2018

References

[1] Wah C., Branson S., Welinder P., Perona P., Belongie S. “The Caltech-UCSD Birds-200-2011 Dataset.” Computation & Neural Systems Technical Report, CNS-TR-2011-001. http://www.vision.caltech.edu/visipedia/CUB-200-2011.html

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, november 1998.

Appendix