Chicken or Dogs?

Team 3: YounHyuk Cho, Chang Liu, Yueqi Zhang, Wenhang Bao, Jiayi Wang

Goals

Our goal in this project is to be able to classify images of chickens and dogs with few criteria:

1. Good accuracy- We want to be able to classify the images with good accuracy. We would want the accuracy to be as high as possible without increasing the computational time too much which brings us to....

2.Decent computational time-We want to be able to find a good balance between good accuracy and good computational time. Even if accuracy can be increased, we don't want it at the cost of very high increase in computational time

3.Interpretability- We want our model to have good interpretability as well so if we have two classifers that have very similar accuracy and computational time, we will always choose a model that has better interpretability

Baseline Model and Feature: GBM+SIFT Feature

The feature that is included in our baseline model is feature extarcted by using SIFT(Scale-Invariant Feature Transform). This feature was provided in class. As our baseline model, we used Gradiant Boosting Model and tried to tune the parameter so that we could acheive "good" predictive performance while having a decent computational time. We fixed our shrinkge at 0.001 and tried to change values of number of trees and interaction depth to see how well the model performs and how much time it took to train the model. Below is the table summary of the results we got using baseline model:

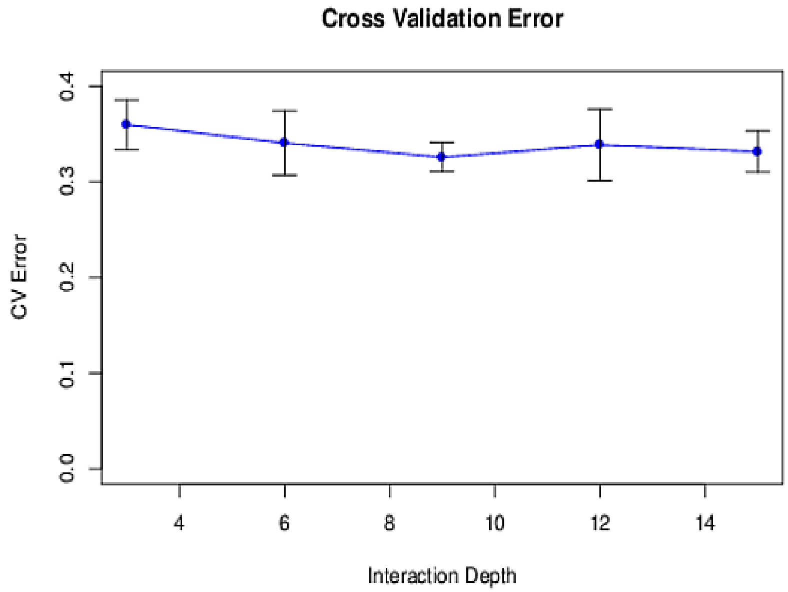
N_tree	CV_err	Accuracy	Best _par	Train_time (using best_par)
250	0.3255	67.45%	9	260.50s
300	0.3205	67.95%	15	461.84s
....	....	....	....	....
2000	0.2815	71.85%	15	2706.64s

Table

As you can see from above table, we tested few numbers for n_trees and used 5 fold cross-validation to tune the parameter interaction depth.

Considering both accuracy and training time, n_tree=250 seems to be the most reasonable choice. When we ran the code with n_tree=2000, although accuracy increased by about 5%, the computational time almost increased by 10 times

This is the visualization of the CV results using ntree=250



The best choice seems to be when interaction depth=9

Other Classifiers using only SIFT feature:

Using Cross-validation to tune parameters to avoid overfitting

Trees

For trees, we examined the cross-validated error results and selected the complexity parameter associated with minimum error which was cp=0.0134. We pruned back the tree to avoid overfitting the data. The complexity parameter is used to control the size of the decision tree and We selected a cp that minimized the cross-validated error.

SVM(Support Vector Machines)

For Support Vector Machines, we tuned cost of constraints violation and selected value of the cost of constraints that minimized the cross-validated error. The value of the cost that minimized the cross-validated error was cost=0.0001

eXtreme Gradient Boosting(Xgboost)

Fixing eta=1, nround=50, we tuned and selected a maximum depth of the tree that minimized the cross-validated error. The maximum depth of the tree was 4

Random Forest

The best parameters are: number of trees: 310; The minimum number of samples required to split an internal node: 2; The criteria to measure the quality of a split: Gini impurity.

Neural Network

The best parameters: 3 hidden layers, with (6,6,7) neurons in each layer;

Overall Summary Table

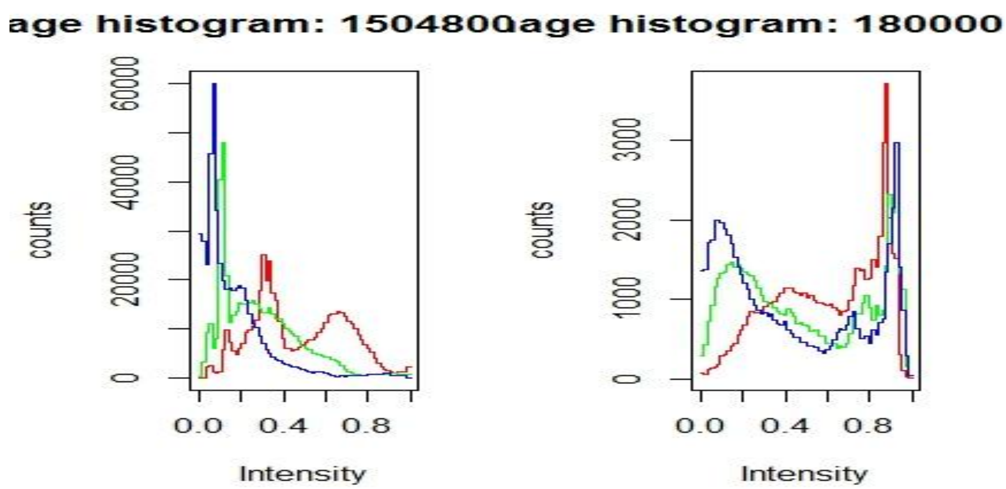
Classifier(best_parameter)	Accuracy	CV_error	Train_time(including tuning)
GBM(depth=9)	67.4%	0.326	5573.34s
SVM(cost=0.0001)	63.59%	0.292	950s
Tree(cp=0.0134)	62.8%	0.372	30.5s
XGBoost(depth=3, eta=1)	66%	0.34	20.70s
Random Forest(ntree=310)	63.5%	0.365	1465s
Neural Network(3 hidden layers with (6,6,7) neurons each layer)	75.5%	0.245	536.24s

Table

Other features considered

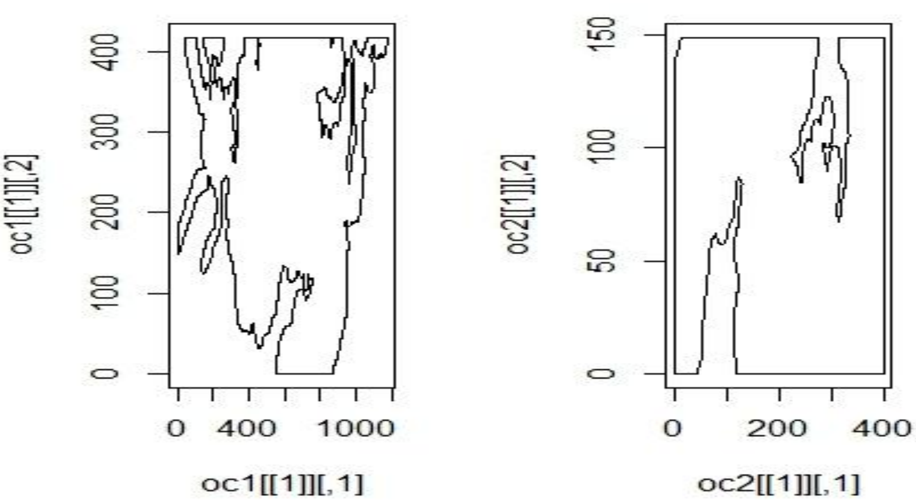
Color Histogram (RGB)

Color feature seemed to be the most basic yet important feature that we could consider to classify images since looking at imgaes, we saw that color of the chicken and dogs were different for many images. We will found out soon that color is the most important feature to be considered for this project.



Outline (contour)

Below is the contour plot of an image. From thie contour plot, we would not be able to tell what each of the two pictures represent. At first we considered outline feature to be included in our model but after looking at the plots of few images, we decided not to use outline feature because information we could get from contour plot seemed very limited.



Deep Features

We will be able to see from the tables below that including color features in the model, improves both the accuracy and computational time but we continued to have some doubts in using color feature as our primary feature because color feature seemed to be very unstable compared to some of the more advanced features, such as deep features. We extracted deep features using python but using deep features had few limitations. Because the size of the data set was very large, it was computationally slow and difficult to run classifiers using deep features in R. Instead, for deep features, we used python to train the model and was able to get good results

Color

Summary Table:

Classifier(Best _parameter)	Accuracy	CV_error	Train_time
GBM(depth=12)	91.6%	0.084	845.41s
SVM((cost=0.0001)	81.5%	0.185	108s
Tree(cp=0.01)	87%	0.13	1.4s
XGBoost(depth=3)	92.7%	0.073	35.81s
Random Forest	90.45%	0.0955	135.8s
Neural network(3 hidden layers with (6,6,6) neurons each layer)	86.5%	0.135	124.93s

Color+SIFT

Summary Table:

Classifier(Best _parameter)	Accuracy	CV_error	Train_time
GBM(depth=15)	91.25%	0.0875	6997.45s
SVM((cost=0.0001)	83.0%	0.17	1096s
Tree(cp=0.)	92%	0.0805	15.27s
XGBoost(depth=2)	93.0%	0.07	171.11s
Random Forest	88.05%	0.195	1418.7s
Neural network(3 hidden layers with (6,6,6) neurons each layer)	87.35%	0.1265	124.93s

Deep Feature

Using deep learning method(Convolutional Neural Network)+deep features

Summary Table:

Classifier	Accuracy	CV_err	Train_time
Deep Learning	94.5%	0.055	237s

The Final Model

The model using color feature gives very good accuracy but as final model we use deep learning method that uses deep features because we believe color is a good feature, it is still very unstable. When test images are set of dogs of similar color to the chicken the model built on the color feature could fail dramatically. Using deep learning method with deep features, we were free from the possible problems of instability that might arise when we use color features and we got better accuracy while maintaining a good computational time.