DS200 Graduate Project

Visualizing the World

Junsheng Pei(id: 3034340729)
Zihao Yang(id: 3034339975)
Nuochen Lyu(id: 3034339416)

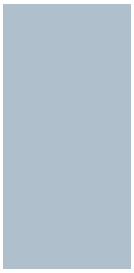
Dec 6th, 2018

Feature Extraction

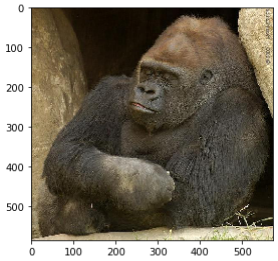
Effective features

Mean of RGB channels: The mean value of each color channels is a very effective feature, even though very simple. By computing the three mean values for the red, green, and blue channel, we can find the foremost color in the picture. For instance, the mean RGB color of a picture of dolphin is (172, 191, 205) which is close to light blue. And the mean RGB color of a picture of a gorilla is (87, 77, 58), which is close to brown. With the three features, we can distinct images by the dominant tone of the images.

Fig[dolphin]



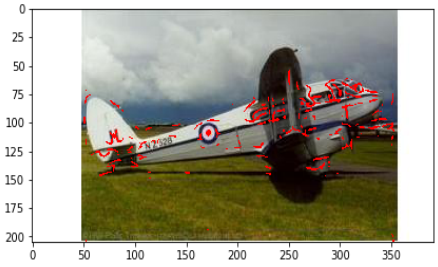
RGB(172, 191, 205)



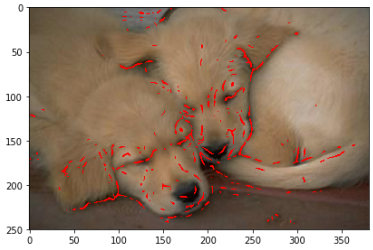
fig[gorilla]

RGB(87, 77, 58)

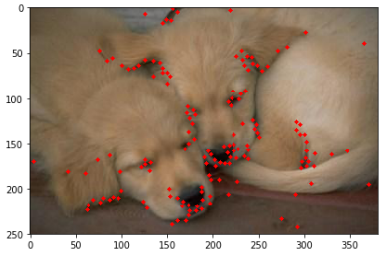
The entropy of the outliers of the images: After reading the documents of openCV, we find two interesting methods cv2.cornerHarris and cv2.goodFeaturesToTrack. These two methods help us find the corners in the images, and with the corners, we have an approximate outlier in each image. The by computing the entropy of the distribution of the images. We can get see how complex the images are. The method may be more useful if we build a matrix pixel by pixel. The entropy of the outliers is not so informative, but it is a cool feature and can improve our performance a little. Like the pictures below, we plot the edges. Different from cv2.cornerHarris, the method cv2.goodFeaturesToTrack can find the best n edges with a maximum distance.



Fig[edges in a plane] entropy: 10.035



Fig[edges in dogs] entropy: 11.46



Fig[150 edges with min distance]

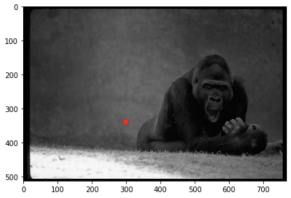
Hu's 1st & 2nd moment invariant: If we regard the grayscale values of an image as a 2D distribution of density, then we can use moments to describe the distribution. For pixel intensities, I(x, y), raw image moments of order (i + j) are calculated by

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y)$$

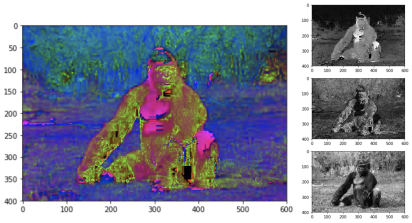
Hu's 7-moment invariants are derived from the normalized central moments, which could remain unchanged under rotation and scaling. The most effective ones are the first and second invariants since their orders are the lowest. Fig[Chimpanzee_centroid] shows the centroid of an image. The central moments are computed based on it.

The variance of Hue and Saturation: Color moments are measures that can uniquely characterize color distribution in an image. We generate the first three moments(mean, variance and skewness) over the Hue,

Saturation, and Value of an image. The coolest ones are the variance of Hue and Saturation. Fig[Chimpanzee_HSV] is the visualization of the HSV form(and separately) of an image.

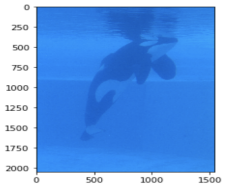


Fig[Chimpanzee_centroid]

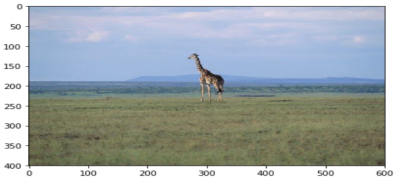


Fig[Chimpanzee_HSV]

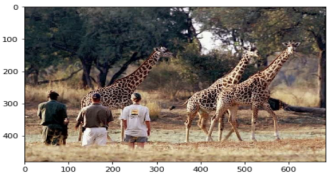
Insignificant and bad features



Fig[whale]

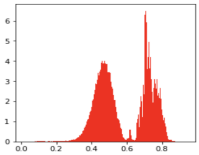


Fig[Giraffe_simple]

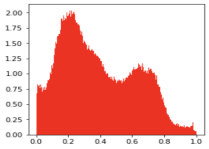


Fig[Giraffe_complex]

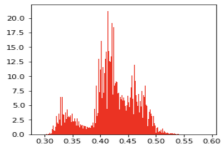
Histogram Bin: The color distribution histogram does vary a lot between different type of animals such as whale(mostly blue) and giraffe(more even color). However, the histogram also differs in single species. In giraffe simple, the color is evenly distributed. However, in giraffe_complex, we have trees, humans and multiple giraffes. The histogram whale is mostly focused in dark blue and light blue so it only has two peaks. While the giraffe picture does have wider spread histogram but the shape is very different as shown below. This feature decreases accuracy.



Fig[histogram_whale]



Fig[histogram_giraffe_simple]

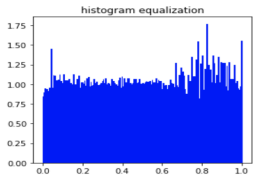


Fig[histogram_giraffe_complex]

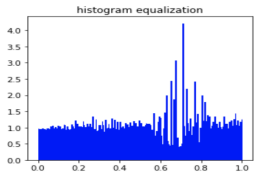
Histogram Entropy & Standard Deviation: The histogram entropy shows the contrast and order of the color. For the whale picture, the colors are obscure and mixed. While the giraffe pictures are bright and trenchant. Thus the entropy for whale is high due to its blurry while the giraffe has a relatively low entropy. So it does work for classifying some animals. But it depends on the camera quality and the location of the animals. Although it is reasonable to assume that comes animals are in the water while some are in the desert so histogram entropy would be helpful. However, there exist bad pictures that destroy the model. For instance, the giraffe entropy varies between grassland and wood.

	Whale	Giraffe simple	Giraffe complex
his_entropy	12.04	4.99	5.39
his_entropy_with_eq	5.32	5.50	5.57
his_std	0.0166	0.0056	0.0020
his_std_with_eq	0.0027	0.0014	0.0001

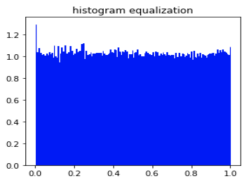
Histogram Equalization: Histogram equalization is a methodology to spread out and normalize the histogram distribution. For some picture that is too dark or foggy. It would help to recover the color so we could better compare the picture to regular pictures. However, here this feature is not effective because it wipes out the information such as the range of the color and the height of the histogram which counterparts the original model.



Fig[histogram_whale_eq]



Fig[histogram_giraffe_simple_eq]



Fig[histogram_giraffe_complex_eq]

Mean of Hue, Saturation, and Value: Since the RGB model and HSV model can be transformed from each other, the mean of H, S, and V is to some degree redundant, due to the existence of the mean of red, green and blue. Below is the test accuracy for each single color moment.

Accuracy	Hue	Saturation	Value
Mean	0.0797	0.0831	0.0764
Variance	0.1163	0.0997	0.0897
Skewness	0.1229	0.0864	0.0764

Hu's skewness invariant: this feature is of 3rd order, therefore, too much information about the "density" of an image is missed. In other words, the information is mainly contained in the low-order moments, as follows.

invariant_1	invariant_1	invariant_1	invariant_1	invariant_1	invariant_1	skew
0.122923588	0.1262	0.0897	0.1063	0.0997	0.1130	0.0864

Model selection

Logistic Regression

The logistic Regression does not produce good accuracy for our project. The logistic regression categorizes dichotomous data. It classifies the data by placing a hyperplane. Thus it requires good and clear training data in order to a have good classify. However, in our project, the data quality is not good. Animals with the same label have various colors. And the location of the animal in the picture, the environment of the animals and the image quality would confuse the classifier.

Fandom Forest

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. It adds additional randomness to the model while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. Since the model only needs to traverse along the path from the root to the leaf, it could deal with data with massive features and make predictions with high accuracies in a very short time.

SVM

SVM model is entering huge trouble here in our project. For a multi-classify problem, SVM is not a good candidate. In our project, it takes almost infinite time to fit the training data. As the class number grow, the computation time grows exponentially.

K-Nearest Neighbors

In the model of K-Nearest Neighbors, we select the k entries in our database which are closest to the new sample and find the most common classification of these entries as our prediction. In fact, KNN doesn't build a learning model, it just stores the entire training dataset. KNN is one of the simplest machine learning. The number of neighbors is the most important parameter in KNN. When k is small, the training accuracy is large, however, the test accuracy is small. When increasing k, the test error increases first and then decreases. Finally, we get the best trade-off when k is around 5.

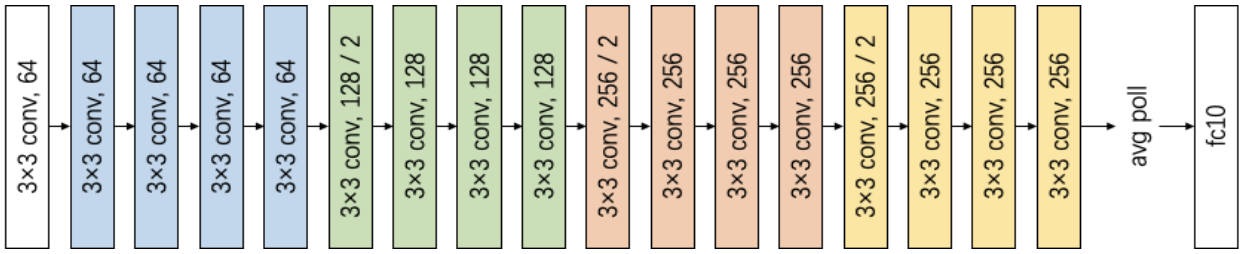
Comparison of the four models:

	Testing Accuracy	Training Accuracy	Computing time
Logistic Regression	22.26%	23.92%	0.089s
KNN	19.27%	41.00%	0.009s
Random Forest	**43.19%**	**99.83%**	**0.705s**
SVM	12.6%	100.00%	Very long

The feature, advantages, and disadvantages have the four model have been covered in the previous section. From the above table, **we could find that random forest works the best for our data with a 43.2% accuracy in the testing data.** While the other model gives poor performance, with logistic regression does slightly better than the other two. SVM takes a long time for training.

PyTorch(Tensorflow)

We find it very different to increase the accuracy with only 15-20 scalar features because the images are more complex than the ordinary dataset. And in the last part of the project, we used state-of-the-art neural architectures to have better classification results. We ran a Residual Neural Network model called ResNet18 in PyTorch (instead of Tensorfloow). A residual neural network is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections or short-cuts to jump over some layers. And we use the annealed learning rate which should return the previous learning rate multiplied by a decay factor in our model. The basic structure of ResNet-18 is as below:



The structure if ResNet-18

When we ran the ResNet-18 in the CIFAR-10 dataset, we can get a validation accuracy of 57.32% with only one epoch and more than 90% accuracy with several epochs. But in our own dataset, this model didn't perform very well. We set the training batch size as 128 and finally get a validation accuracy of 37.2% with 20 epochs. This might be because the dataset is not large enough or we need better image processing.