Using Color for Object Recognition

Mohammad Moghimi
Computer Science and Engineering Department
University of California, San Diego
mmoghimi@cs.ucsd.edu

Abstract

Object Recognition is an important task in many Computer Vision problems. The problem is usually defined as classifying new images given a set of training images. One of the most challenging problems is Category-level Object Recognition which has got attention in the past decade. In this project the work on Fine-grained Category Object Recognition is presented. Despite General Object Recognition, images in this problem have a lot in common. But they still have differences that we can use for the classification. In this research, we use color feature for classification in many different ways. We are going to try to use color features which has not given much attention by the computer vision community to see if we can use it to improve the classification of 200 bird species (CUB200)[7]. The goal is to improve the performance of existing papers [5] [4] and also find more general way to use color in other similar problems.

1. Introduction

Object Recognition is important in many Computer Vision problems. The problem is defined as classifying new images given a set of training images. The output of the classification can be a label for the whole image, a bounding box for the object, fine segmented boundary or the object with all of its parts. There are two different types of Object Recognition in terms of labels assigned to images. Some Object Recognition systems recognize specific objects and some others detect the object's category. For example, the output can be whether the image is an specific car or if it has any car in it. In this project we focus on fine-grained visual object categories in which the categories are somewhat similar to each other and focus is on the little differences between the categories.

In [7] Welinder *et al*. collected a database of 200 bird species called CUB200 with more than 6000 images and later extended it to CUB200-a, a more challenging dataset of birds in unconstrained pose and environment. There are different features that might be useful bird classification. For example shape of the bird's beak, or belly can be a distinguishing feature for some categories. Usually most of object recognition methods work on gray level images and extract information about corners and edges but in this project we did a series of experiments to see the effect of using color as a feature for recognition.

We are going to work on CUB200 and CUB200a bird species datasets. There are two general different ways to extract features: global and local. Global features are extracted from the whole image rather than local patches. One simple and usual way to get global information is to create color histograms for the whole image. There are many color spaces such as RGB, YCbCr and HSV that might be useful for us. We will show the result of doing different color spaces and their respective performances which is classification accuracy in this problem. Subsets of these color channels is another option that have been tried to improve the performance. The results will be discussed in section 3.5.

In addition, color constancy is an issue that should be considered when color features are being processed [3]. The effect of different natural lightings should be eliminated. There are several source codes available on the Internet which have been tested.

In the experiments effect of different number of train/test images, different distance function and classification will be discussed too. Distance functions are an important part in a classification system. χ^2, Earth Mover's Distance (EMD), linear kernel or even simpler distance functions are possible distance functions to use.

The next section will explain the classification method that have designed for the classification. Section 3 discusses the results of the experiments and the last section concludes the project and points out the possible directions for the future works.

2. Methodology

Like many other computer vision algorithm in object recognition, first we need to extract some features from the

input image. To do this, it might be necessary to do preprocessing to make the system more robust. As mentioned before, there are two general ways of extracting features which are global feature extraction and local feature extraction. In both cases the output of feature extraction would be a vector representation of the image. To find such a representation histograms are used extensively in computer vision. The output of a histogram with a specified number of bins can be used features for the next stage which is the classification.

In this project both global and local features are used for recognition. Since the focus of this work is using color for recognition,we only used color to build the histograms. Therefore, for the global features, a histogram of the region of interest is created. Specifically, marginal histograms of different color channels of a set of color spaces are used to extract color features. In addition to global features, local features have got an extensive attention from object recognition community in the last years.

They are usually called local invariant features. The goal of these features is to be invariant to many changes like pose, intra-category variation, size, orientation and illumination. The most important one for color features is illumination invariance. The extracted color features should be as invariant as possible to illumination to make a robust classifier. However there are quite many methods for automated recognition, we are relying on the part location meta data available for CUB200a.

Classification is the next step after extracting features. There are many possible choices for the classifier. Some typical methods are Support Vector Machines (SVM), k-Nearest Neighbor(kNN) or decision trees. KNN is chosen for this project just because of simplicity. Although, other methods should be tested to make a reasonable decision.

KNN basically calculates the distance between a sample image and all training images in the dataset to find K best matches. Then the final decision is done after doing a voting procedure. The performance of the object recognition system is highly dependent on the distance calculated between features extracted from all pairs of images. χ^2 distance has been chosen for this purpose. After calculating the whole distance matrix which includes distances of all pairs of images, images with minimum distance are chosen as the best matches for the given input. Voting on the labels of K best matches results in the final decision. For a multi class problem the voting procedure needs some hacks to work because it is possible that all K best matches are from different categories.

3. Experiments and Results

We ran a series of experiments to see the effect of different changes to the performance of the classification. As mentioned before, all experiments are done on two similar datasets CUB200 and CUB200a which are explained in more details in section 3.1. To show the performance of a multi class classifier, confusion matrices are used which will be discussed in section 3.2. Section 3.3 color constancy and then different regions of interest in section 3.4 and different color spaces in section 3.5 are explained.

3.1. Dataset

In this project, all experiments are done on two datasets namely CUB200 and CUB200a. They are both datasets of the same 200 bird species mostly from North America. CUB200 is a dataset of 6033 bird images and the other dataset has 9665 images. There is not a significant difference between these two dataset other than the meta data than comes with them. CUB200 has the segmented annotation of bird with brush strokes as well as bounding boxes as opposed to part locations in CUB200a. They both has some set of attributes for each image. For most of the experiments CUB200 is used which is publicly available on the Internet. But for the experiments using part location CUB200a is used.

Fig. 3.1 shows six sample bird images from CUB200. As you can see in the figure, there are some bird species that are easily recognizable just using their color (top row) and there are some categories for which the color is not a good feature (bottom row).

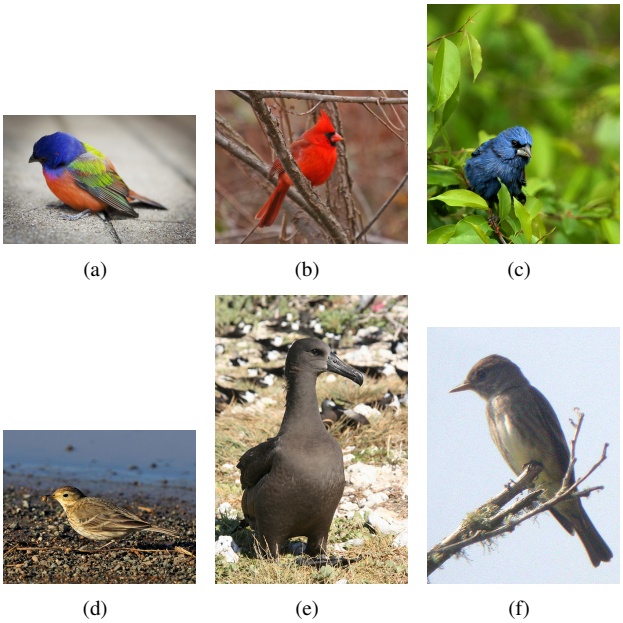


Figure 1. Sample images from CUB200 dataset. Painted Bunting (a), Cardinal (b) and Indigo Bunting (c) are shown in the first row and American Pipit(d) , Black footed Albatross (e) and Olive sided Flycatcher (f) are in the second row. Top-view of sphere and its projection onto an image plane.

3.2. Confusion Matrix

The output of a multi-class classifier is a matrix namely confusion matrix which shows the number of instances from class A that are assigned to class B for all pairs of classes A and B. The numbers on the diagonal show the correct classification as opposed to non diagonal elements which show the miss-classifications. Fig. 3.2 shows a sample confusion matrix for the classification of CUB200. It is 200 by 200 matrix which is shown using color mapped values to make it more understandable. The ideal case would be having dark red points on the diagonal and dark blue everywhere else. One could calculate the accuracy percentage of a classifier by summing up the diagonal elements and dividing it by the sum of the whole matrix.

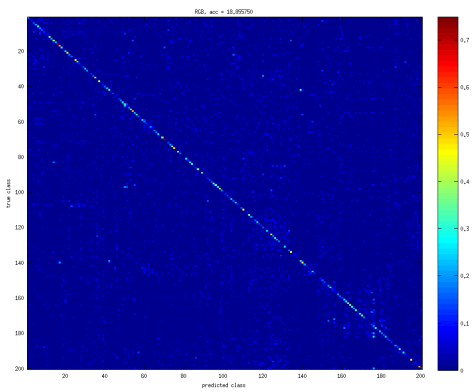
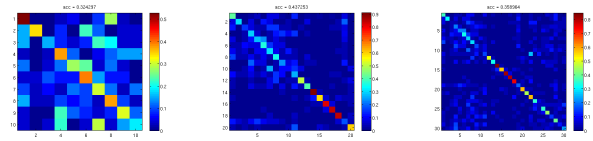


Figure 2. This figure shows a sample confusion matrix for CUB200. It is a 200 by 200 matrix which is shown using color mapped values. As the color bar shows, the hotter the color, the higher the value.

The number of classes are really important for the performance of a multi class problem. For example when there are 10 classes, there 10% of the elements in the confusion matrix are considered to be the desired location which are ones on the diagonal. When there are 100 classes, the percentage drops to 1%. These numbers are also the same as the random classifier accuracies too since the random classifier assign labels randomly and put the data in a random element in the confusion matrix. It is expected that the accuracy of the classifier drops when the same method is tested one an dataset with more number of classes. But this is not always true. For example in Fig. 3.2, the accuracy of NN method on first 10 categories of CUB200 is better then the accuracy of the same method on the first 20 categories.

3.3. Color Constancy

Since color perceived by the camera is dependent on both object and light source, the color in the image is not just the bird's color. For example, the color on a white surface is totally different when a tungsten lamp or a florescent lamp is turned on. There are some source codes available on the Internet which try to correct the colors to make them independent to the light source based on some training data that they have use. Fig. 3.3 shows an image of a bird and the corrected image of it. As it can be seen in the figure, the color of the bird didn't change very much and since all of the images are taken outdoors, it seems that color constancy will not help much. There is not much of a difference either on the monitor or on a printed paper. But definitely without testing the effect of this change, we cannot be sure whether using color constancy is beneficial or not.



(a) (b) (c)

Figure 3. The performance of NN method on CUB200 for the first 10 categories, 20 categories and 30 categories respectively. The accuracies are 32.4, 43.7 and 35.9 percents.

3.4. Region of Interest

One of the most important factors in creating color histograms is the region of interest. Region of interest or ROI is the area that we are extracting the color features from. The region of interest could be manually fed to the algorithm and automatically chosen by the algorithm itself. The brush strokes annotation are available for the CUB200 dataset. Depending on the size of the brush, it may need to be dilated or eroded the segmented bird region. Fig 3.4 shows the brush annotation for a sample from the dataset. We will show that the erosion helps to improve the accuracy of the classification.

Another kind of RIO selection is segmentation. Usually some ground truth points are needed for segmentation, otherwise the problem will be very hard and also ill-defined. Although, sometimes the problem seems very easy, our experiments show doing an automated segmentation on the whole dataset even with some ground truth points is not an easy problem. Fig. 3.4 shows the image and the output of grab-cut segmentation algorithm [2].

Table. 3.4 shows the results of different ROI selection methods on different color spaces. It seems that in all color spaces eroding the brush annotation with a disk of 15 pixels radius gets the best performance because we neither want to allow any non-bird part in the region nor want miss any parts of the bird. The automated segmentation didn't get good results too. The result for automated segmentation are similar to do the segmentation on the whole image.

Another option is to have different regions for different parts of the bird's body. For example, In some cases the

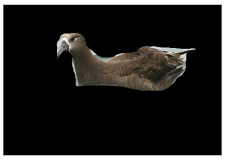
(a)

(b)

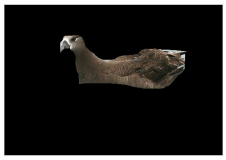
Figure 4. (a) is the original bird's image and (b) is the color corrected image. They look completely similar either on the computer's monitor or on a printed paper [1].

(a)

(b)



(c)

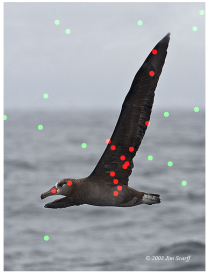


(d)

Figure 5. (a) shows a sample bird image. (b) is the brush annotation done by a human user. (c), (d) are brush annotated regions eroded by a disk of size 10 and 15 pixels respectively.

beak has a different color than other parts of the bird. This kind of information may be important when we are extracting color histogram. Fig. 3.4 shows a sample bird with its parts marked on it. It happens that some birds don't have some of the 12 typical bird parts. The reason could be either inherent to the type of the bird or due to occlusion. This information is available for CUB200a dataset and the results cannot be compared with the other ones on CUB200. Note that the classification accuracy of using different parts is 20.5%.

(a)



(b)

(c)

Figure 6. (a) is the original image. (b) shows the image with the ground truth points marked with green and red circles and (c) is the output of the grabcut-segmentation.

Another way of finding good regions to extract features from are salient regions. Fig. 3.4 shows an example bird image with its salient regions extracted using Dirk Walther's MATLAB code [6]. The basic idea behind salient regions is that these regions have a high response to some filters. The use of salient regions is set for the future works. Although, they don't seem to be quite accurate.

	YIQ	YCbCr	RGB
whole image	-	13.1514	9.3084
brush (erosion = 15px)	**18.9334**	17.2170	10.3465
brush (no erosion)	17.2209	17.0656	-
brush (erosion = 10px)	18.0811	17.4211	10.3606
automatic segmentation	12.2661	12.2744	7.9154

Table 1. Comparison of different ROI selection methods. The numbers are the accuracy of classification on CUB200.

(a)

(b)

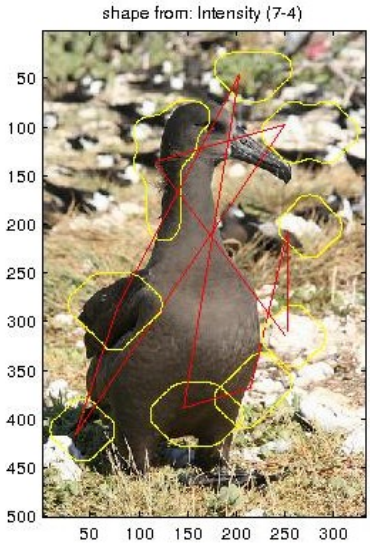
Figure 7. (a) is a sample image from CUB200a dataset and (b) is the same image with the its part marked on.

3.5. Color Space

The representation of color is a factor that should be investigated too. The way that experiments are done is first find the distance between each color channel's output. Then combine these distance measure using a weighted sum. For example for RGB, one distance matrix which is of size of the number of test image by the number of training images is created for χ^2 distance between marginal R histograms. Another one is created for G and so one for B. In the case of CUB200 they could be as big as 6000×6000 which needs significant amount of time and storage to deal with. Table. 3.5 shows the result of using 7 different color spaces. All weights are set to 1 in this experiment. According to this result, there is a quite large difference between different color spaces. For example, some color spaces like YIQ, YCbCr and HSV got better results possibly because of their orthogonality characteristics.

To optimize the accuracy of the classification, one may want to search for the best set of weights to calculate a better between two set of histograms of different color channels. In this project, this is done by using a simulated annealing approach. Simulated annealing is one of hill climbing type methods that decreases the learning factor (step size) over time. Fig. 3.5 shows the result of changing weight vs.

(a)



(b)

Figure 8. (b) is a sample bird image and (a) is the same image with its salient images marked by yellow boundary lines.

Color Space	Classification Accuracy
YIQ	**18.9334**
YCbCr	17.2170
RGB	10.3465
HSV	18.4551
XYZ	7.8032
Lab	8.5005
Luv	17.9914

Table 2. This table shows the accuracy classification using seven different color spaces.

the classification accuracy. It always saves the best set of weights which have been seen so far. Therefore the graph is always ascending. Using this optimization method, the classification could be optimized up to 22%.

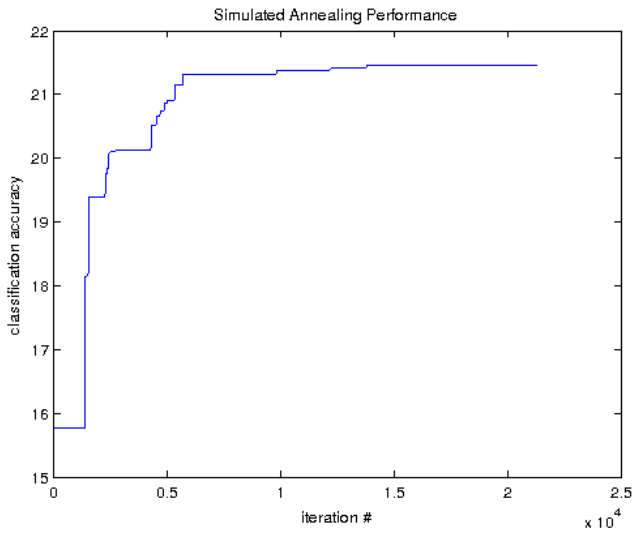


Figure 9. The output of simulated annealing method. The diagram is the classification accuracy vs. number of iterations.

4. Conclusion and Future Work

In this project, we have done a series of experiments to find the best way to use color. These experiments are done on different aspects of the problem which include where and how to extract the color features from. The ROI selection experiments show that when there is segmentation available, we can get better results. Depending on the amount information such as complete segmentation, localized parts, or just random point on the object, the classifier performs differently. Therefore basically the more information, the better.

Also, different color spaces have different characteristics. Some emphasize the change in some directions more than the others. For example, if the classification of different brown color is wanted, an appropriate color space is needed to map the color of each point to the dimensions which truly show the differences in the brown colors. In this projects, different color spaces and their combinations were tested to see which one results in the best classification accuracy. Our experiments show that YIQ has the best performance compared to the other color spaces.

More thorough experiments should be done using the part annotation data available for CUB200a dataset to see how much we can get from part locations data. Some experiments like automated segmentation or cropped bounding box should also be tested to able to compare the result of localized parts information. Also, a combination of segmentation and the brush strokes annotation could be a good option too.

After building a color based classifier for using color feature, the next problem to study is the way to combine it with the output of other classification methods. Multiple Kernel Learning (MKL) and simple voting are possible options for this matter. Also, better ways of combining different classifiers should be investigated.

References

[1] K. H. A. Chakrabarti and T. Zickler. Computational color constancy with spatial correlations. Technical report, Harvard Sch. of Engg. and App. Sc., 2010. Tech. Rep. TR-09-10.

[2] A. B. C. Rother, V. Kolmogorov. Grabcut - interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

[3] M. Ebner. *Color Constancy*. John Wiley & Sons, England, 2007.

[4] C. W. S. Branson and S. Belongie. Interactive labeling and online learning of part and pose models. In *Submitted to International of Comptuer Vision, 2011*, 2011.

[5] C. W. S. Branson and S. Belongie. Interactive localization and recognition of fine-grained visual categories. In *Submitted to International of Comptuer Vision, 2011*, 2011.

[6] D. Walther and C. Koch. Modeling attention to salient proto-objects. *Neural Networks 19, 1395-1407*, 2006.

[7] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.