Developing Open Source code for Pyramidal Histogram Feature Sets

BTech Project Report
by
Subodh Misra
subodhm@iitk.ac.in
Y6481
Guide: Prof. Amitabha Mukerjee
Dept of Computer Science and Engineering
IIT Kanpur - 208 016
INDIA

April 12, 2010

CONTENTS

1 IMPORTANCE OF PYRAMIDAL HISTOGRAM FEATURE SETS

It is very easy for the human brain to recognize any object and classify those under different classes but if we want a computer system to do this job then it becomes an extremely difficult task. If we observe carefully, we do recognize that objects are classified largely due to change in shape and appearance. So if we are able to represent shape and appearance of images by some means and use them to train SVM classifier then we can classify images.
To achieve this purpose, Pyramidal Histogram Feature Sets become an utmost important utility in the process of image classification.

So I develop two types of descriptor and a representation of spatial layout which is applied to both types. The descriptors consist of visual words computed on a dense grid, which will be referred to as *appearance*; and image gradients again on a dense grid, which will be referred to as (local) *shape*.

These descriptors are frequently used nowadays in image classification algorithms but are not yet been implemented in OpenCV library. So it will be a great contribution to OpenCV library.

2 DATASETS

The Caltech-101 dataset compiled by Fei-Fei et al. (2004) is used for testing the codes for building the feature sets.

3 PRESENT IMAGE CLASSIFICATION MODELS

3.1 SERRE POGGIO CLASSIFIER

This is the biologically inspired model of object recognition. The way human brain works is that it builds hierarchical models invariant of size and position but has feature specificity. There are simple cells which select specific features and there are complex cells which perform the pooling operation to get size and position invaiance. In this way we are able to recognize objects according to its features not the size and position. In serre poggio classifier visual cortex mechanism is implemented by alternating layers of simple and complex cells.

Simple cells use convolution with local filters and compute higher order features.

Complex cells use pooling mechanism to get invariance.
Following layers are calculated to get the feature vector of an image.

- S1 layer : Gabor filter of different size is used to get S1 features from original image
- C1 layer : S1 features are convolved with 2D max filter to get C1 features to get spatial

invariance.
- S2 layer : Now there are d2 prototypes from training images. we perfrom template matching between each of d2 prototypes and C1 features. We get S2 features for each of d2 prototypes.
- C2 layer : Now we perform global MAX operation and we get array of d2 scalars.

Finally we perform all pairs linear SVM classification.

3.1.1 Saurabh Daptardar's thesis

Mtech student of IIT Kanpur Y7 batch had imlemented Serre poggio classifier as part of his thesis work "Explorations on a neurologically plausible model of image object classification".

3.2 Mutch and Lowe classifier

This is also the biologically inspired model of object recognition as was Serre Poggio model but there are some changes in this algorithm.

Following layers are calculated to get the feature vector of an image.
- Image layer : Image pyramid of 10 scales is created
- S1 layer : Gabor filter of size 11 X 11 is used to get S1 features from original pyramid
- C1 layer : S1 pyramid is convolved with 3D max filter of size 10 X 10 and depth 2 units to get C1 pyramid.
- S2 layer : Now there are d2 prototypes from training images. we perfrom template between each of d2 prototypes and C1 pyramid. We get S2 pyramid for each of d2 prototypes.
- C2 layer : Now we perform global MAX operation and we get array of d2 scalars.

Finally we perform all pairs linear SVM classification.
Sparse features are used in this mechanism and only the features that are highly weighted by SVM are considered.

Differences between Serre Poggio and Mutch n Lowe algorithm are:
- Different sized gabor filters are used in Mutch And Lowe for S1 feature calculation and pyramid approach is used to extract S1 features.
- Image height is always scaled to 140 in Serre Poggio while in Mutch n Lowe shorter edge is scaled to 140 while maintaining the aspect ratio.
- C1 samples do not overlap in serre poggio model.

3.3 Bosch and Zissermann technique of object classification

This technique estimates the Region of Interest(ROI) in an image in order to reduce background noise and hence better feature calculation of the object in the image. It improves the image classification by not just calculating the bag of visual words but also the orientation gradient to get the edge distribution in the object and better feature calculation. For classification of images, it uses Random forest and ferns classifier or the SVM classifier.

4 OBJECTIVE OF BTP

Theoretically Bosch and Zissermann technique proved to be more accurate than 'Serre Poggio' or 'Mutch and Lowe' and also it has not been implemented yet in C++, so it was decided to implement Bosch and Zissermann technique. The implemented code was decided to be contributed to OpenCV library.

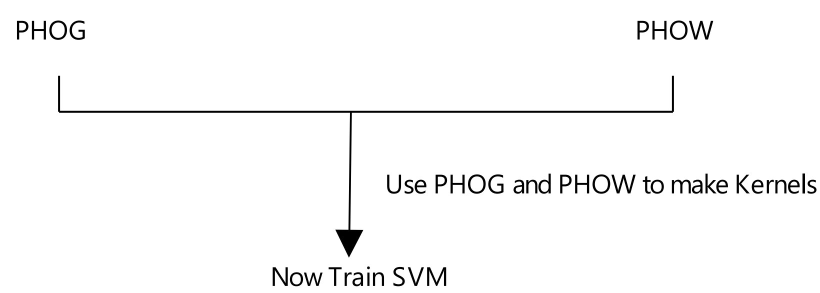
5 Work Done

Comparative study of various image classification algorithms in terms of accuracy and time taken. Dataset used was Caltech101.

Algorithm	No. of Training Images = 15		No. of Training Images = 30	
	Accuracy	Time Taken	Accuracy	Time Taken
Mutch and Lowe Algorithm	46.30%	~1day	54.40%	~1day
Serre Poggio Algorithm	36.20%	~4.5days	43.60%	~7days
Saurabh Daptardar's implementation of Serre Poggio Algorithm	28.60%	~1day	28.60%	~2.5days

Implementation of Bosch and Zissermann algorithm :

Given a set of images of different classes, we will calculate Pyramid Histogram of oriented Gradients(PHOG) and Pyramid Histogram of Visual Words(PHOW) for those images and then train SVM on those features.



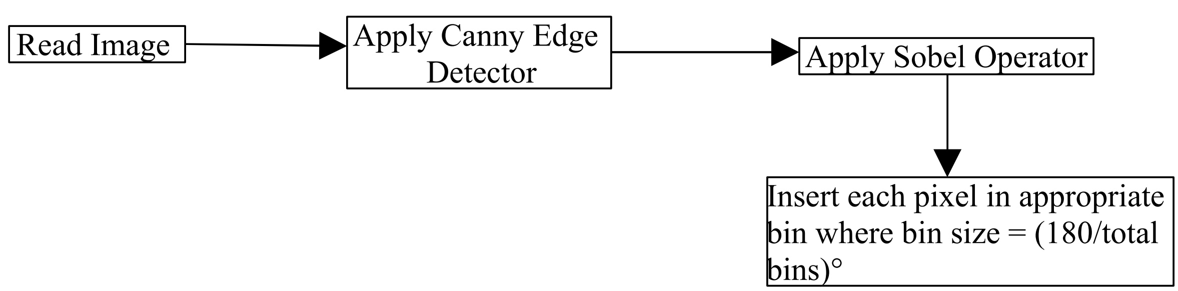
Objects can be classified according to their shape or appearance. PHOG provides a good estimate of the shape of an object and PHOW provides a good estimate of appearance of an object

Pyramid Histogram of Oriented Gradients(PHOG):

Estimating the shape of an object is nothing but estimating the edge orientations of the object. So if we calculate gradients of edges and put them in groups then we get HOG. For example, suppose we want gradients to be divided into groups 0-20 degrees, 20-40 degrees ….. 160-180 degrees., then we call each interval a 'bin' and put each pixel in a bin such that gradient of edge at that pixel lies in the interval denoted by that bin.
Now we calculate PHOG by dividing the image into 4 parts and doing the above steps for each part. In this way we get a pyramid of HOGs. This is necessary to get a good estimate of the shape of the object.

Canny edge detector is used to find the edges in an image and then Sobel operator is used to get gradient of the edges which can then be finally put into respective bins.

Implementation of PHOG:



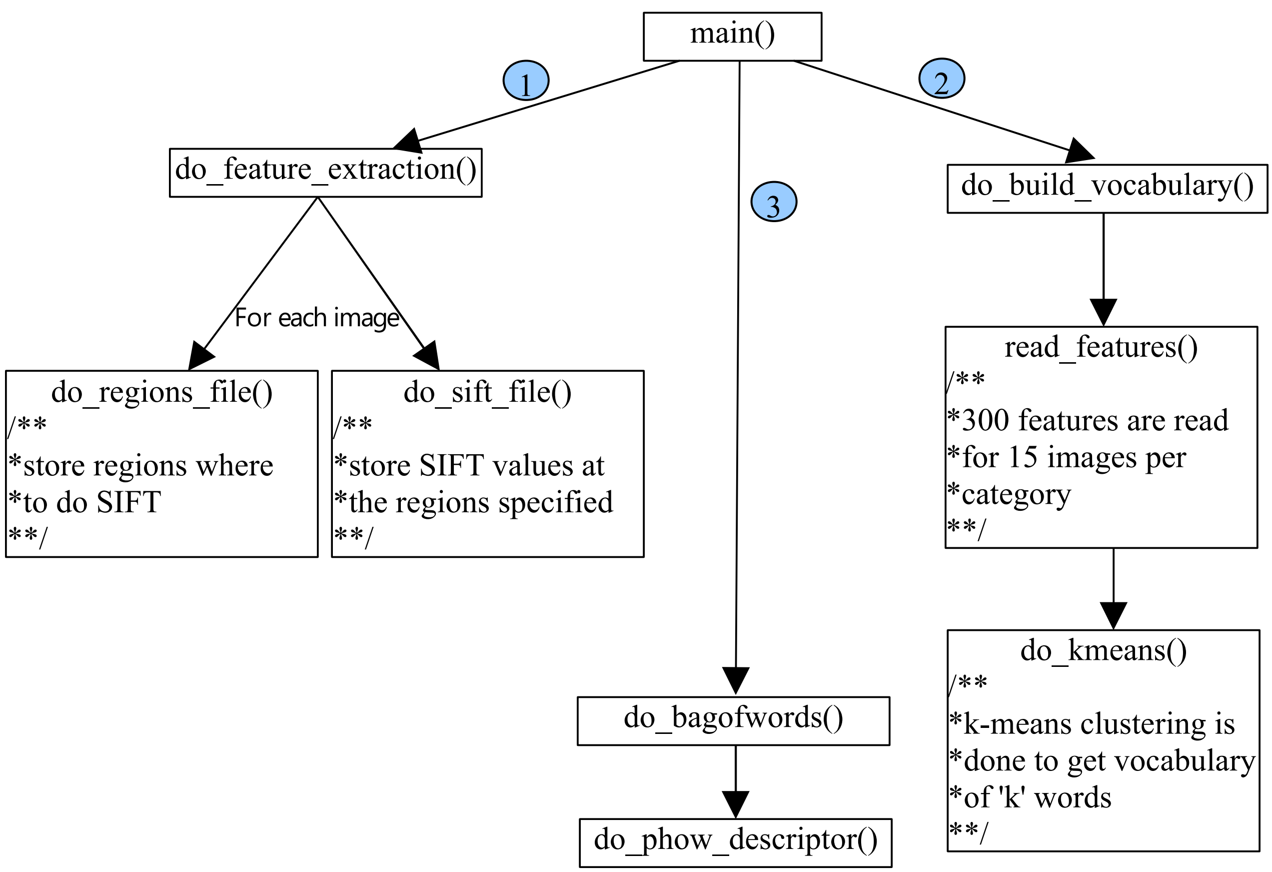
Pyramid Histogram of Visual Words(PHOW):

PHOW is used to get a good estimate of appearance of an object. We make meaningful sentences in a language by using words defined in the vocabulary of that language. Similarly objects are made by using key features(words) from a reference set which consist of features that define different classes of objects. So a face consist of features such as eye, nose, lips. A bicycle consists of chain, tyres, handle. Hence these individual features are called 'words' which collectively from a vocabulary.
Our aim is to build a vocabulary of words for a set of images and then define each object using words from that vocabulary. Objects from same class will be defined more or less by same set of words and hence we will be able to classify an object by looking at its bag of words.

We calculate features in an image by using SIFT feature calculator. Pick some fixed number of features from each image and then do k-means clustering in order to get 'k' words which will be our vocabulary. In order to get HOW we calculate distance of each feature of that image from all k words and put the feature into the ith bin such that distance of that feature was least from ith word. In this way we get a histogram denoting the number of features belonging to each

word.
So objects from same class must have similar histogram because they have similar key defining features. So we can classify an object by looking at its Histogram of visual words.

Implementation of PHOW:



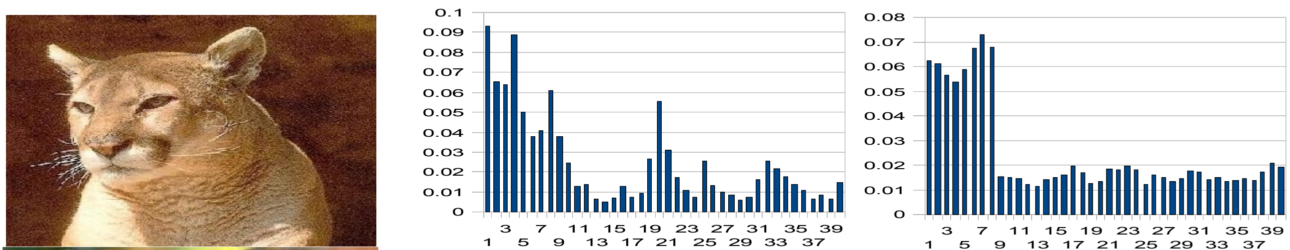
6 Experimental Validation of my code

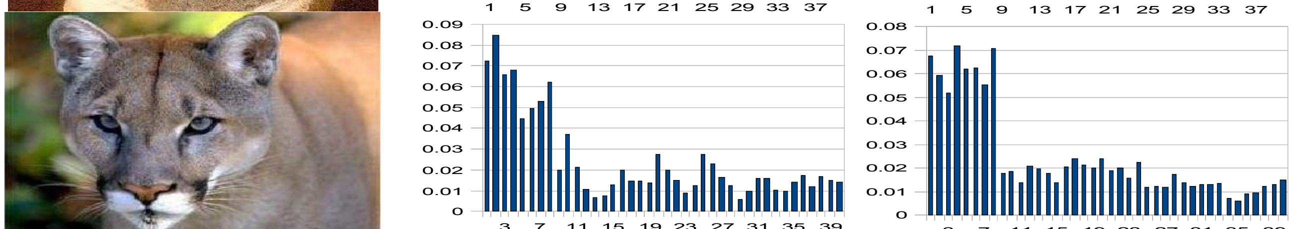
I checked the results of my code with that of the Matlab code written by Anna Bosch and found that the results are comparable and hence it was implemented correctly by me.

Experimental Validation of PHOG :

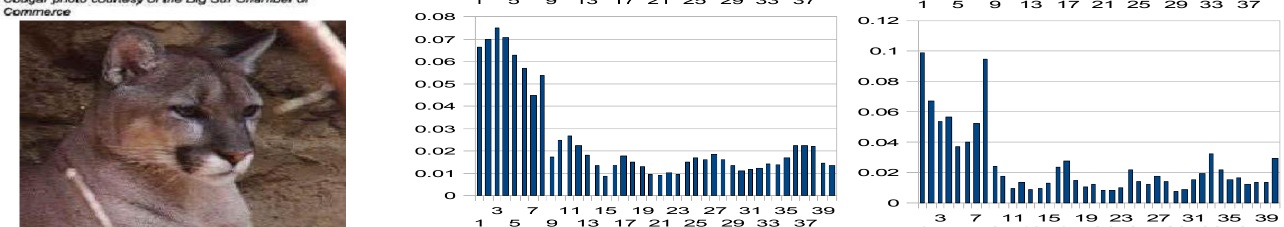
<u>My implementation</u> **<u>Anna Bosch code[9]</u>**

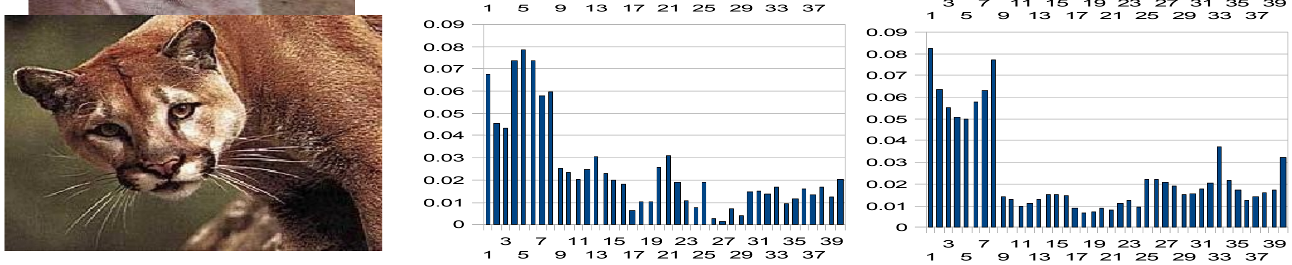
My Implementation **Anna Bosch code[9]**

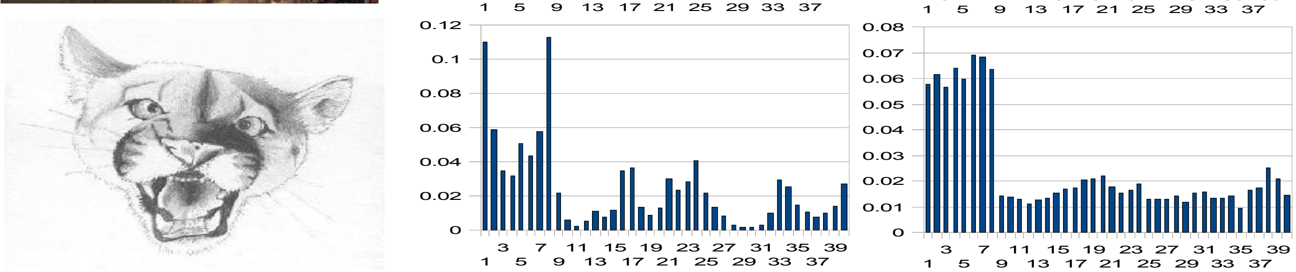




Cougar photo courtesy of the Big Sur Chamber of Commerce

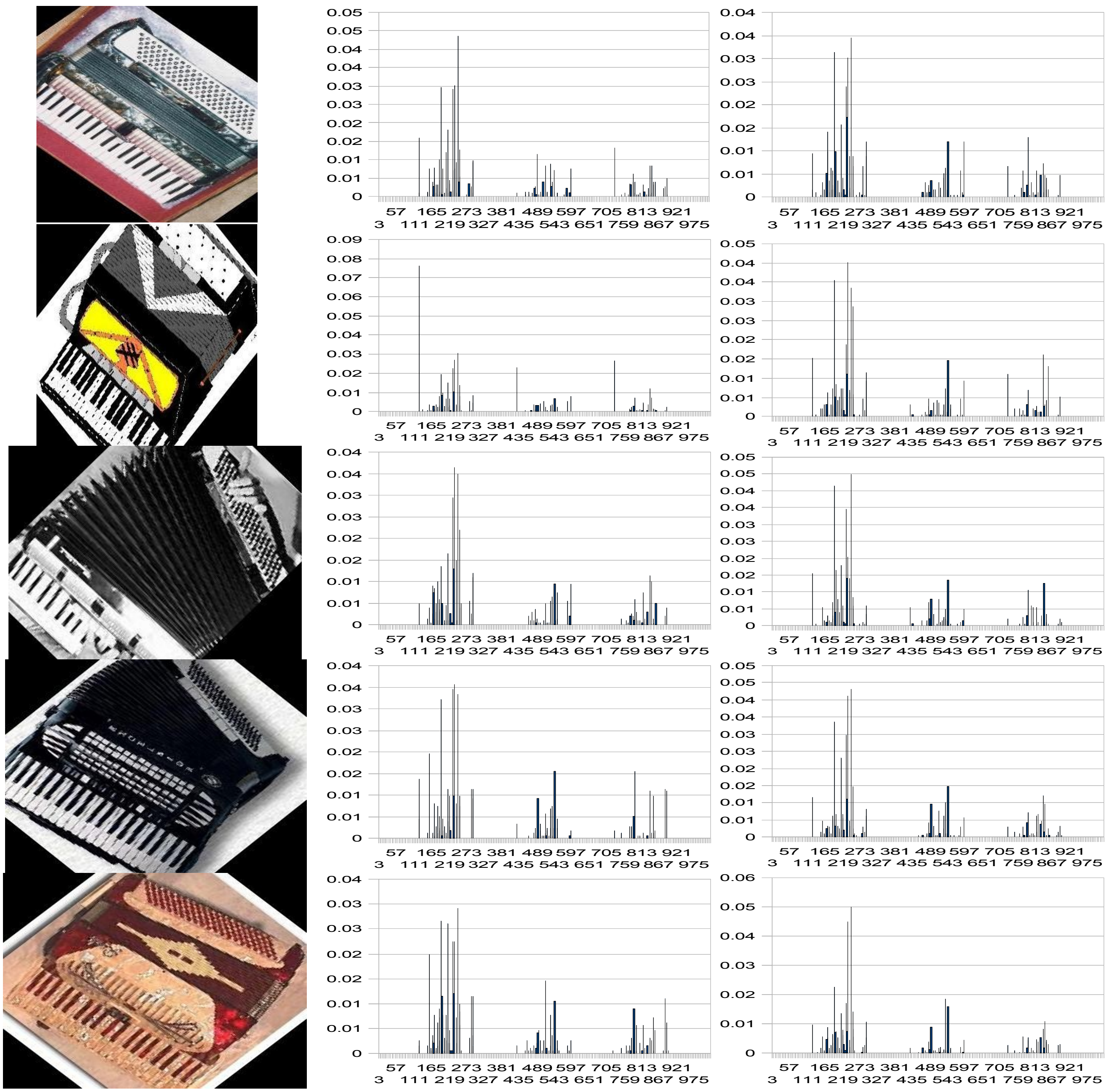




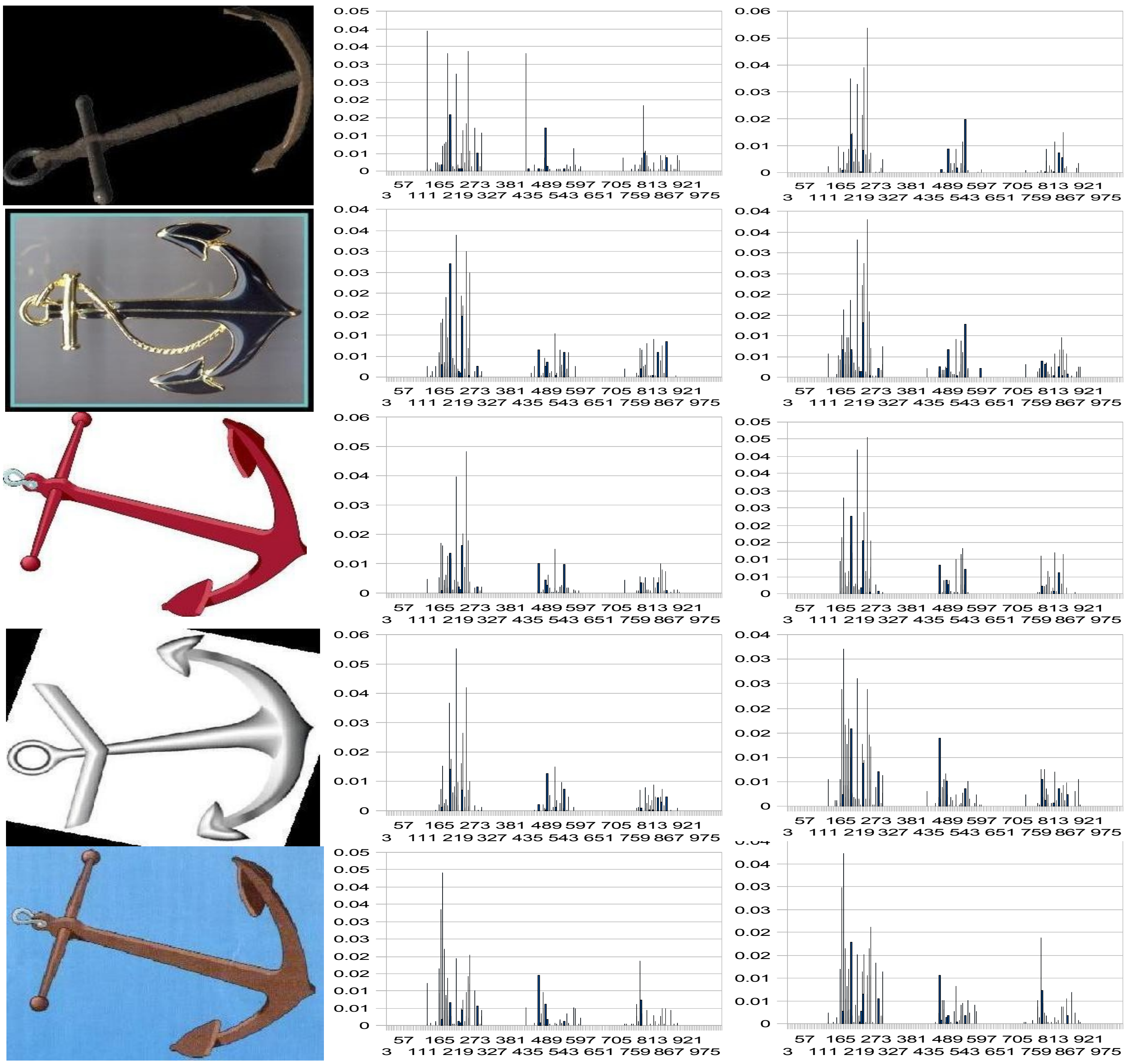


Experimental Validation of PHOW :

My implementation **Anna Bosch code[9]**



My implementation **Anna Bosch code[9]**



Theoretical Validation Of Above Results

Above results can be validated by measuring the relative closeness of PHOG and PHOW features of two different images. This can be measured by calculating Bhattacharyya coefficient for the features of two images. If two images belong to same class then Bhattacharyya coefficient will be more than that for images of different classes.
The Bhattacharyya coefficient is an approximate measurement of the amount of overlap between two statistical samples. The coefficient can be used to determine the relative closeness of the two samples being considered.
Hence intra-class Bhattacharyya coefficient should be more than inter-class Bhattacharyya coefficient.

Calculation of intra-class Bhattacharyya coefficient
I have calculated this by measuring Bhattacharyya coefficient of two PHOG(or PHOW) features of same class and averaging over all pairs of images in same class. Total images per class was 30

Calculation of inter-class Bhattacharyya coefficient
I have calculated this by measuring Bhattacharyya coefficient of two PHOG(or PHOW) features both of different classes and averaging over all such pairs of two classes. Total images per class was 30

Bhattacharyya coefficients for PHOG features

Class	Chair	Cougar_Face	Cup	Faces	Laptop
Chair	0.998444	0.768454	0.878444	0.768555	0.898414
Cougar_Face	X	0.998432	0.698334	0.901111	0.797666
Cup	X	X	0.996666	0.701114	0.858434
Faces	X	X	X	0.997563	0.807411
Laptop	X	X	X	X	0.995454

Diagonal cells indicate Bhattacharyya coefficient for PHOG features of intra-class images (calculated as described in 'Calculation of intra-class Bhattacharyya coefficient').

Other cells indicate Bhattacharyya coefficient for PHOG features of inter-class images (calculated as described in 'Calculation of inter-class Bhattacharyya coefficient').

Bhattacharyya coefficients for PHOW features

Class	Chair	Cougar_Face	Cup	Faces	Laptop
Chair	0.994876	0.563234	0.778441	0.576854	0.798314
Cougar_Face	X	0.987777	0.697734	0.953711	0.746776
Cup	X	X	0.994566	0.755114	0.688434
Faces	X	X	X	0.996663	0.707411
Laptop	X	X	X	X	0.979545

Diagonal cells indicate Bhattacharyya coefficient for PHOG features of intra-class images (calculated as described in 'Calculation of intra-class Bhattacharyya coefficient').

Other cells indicate Bhattacharyya coefficient for PHOG features of inter-class images (calculated as described in 'Calculation of inter-class Bhattacharyya coefficient').

As we can see that intra-class variabilty is very less(values close to 1) and inter-class variability is high so my feature calculations are correct.

References

[1] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. 2007.

[2] ”Saurabh Daptartdar”. ”*Explorations on a neurologically plausible model of image object classification*”. PhD thesis, 2009.

[3] Christoph Lampert. Detecting objects in large image collection and videos by efficient subimage retrieval. 2009.

[4] Jim Mutch and David Lowe. Object recognition with sparse, localized features. 2006.

[5] H. Jhuang T. Serre L. Wolf T. Poggio. Biologically inspired system for action recognition. 2007.

[6] Poggio” ”Riesenhuber. ”hierrachical model of object recognition in cortex”. 1999.

[7] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 411–426, 2007.

[8] http://cbcl.mit.edu/software-datasets/

[9] http://www.robots.ox.ac.uk/~vgg/index.html