PHOW Classification

Carlos Andres Reyes Rivera
Universidad de los Andes
ca.reyes1787@uniandes.edu.co

Lina Maria Fierro Zambrano
Universidad de los Andes
lm.fierro1340@uniandes.edu.co

Abstract

The present lab is about the use of VLFeat in the classification of the Image-Net database. This database is conformed by 995 categories organized according to the WordNet hierarchy. The principal objective of this lab is achieve the best accuracy through the modification of the phow_caltech101 script parameters

1. Database

During this lab we used two database, the caltech101 and the Image-Net. The first is the original database of the phow_caltech101 script and we used to understand and prove the phow algorithm. The second is the databased on which we want to work and we used to modified the script and test the classification.

1.1. caltech101

This database is conformed by 101 categories collected in September 2003 by Fei-Fei Li, Marco Andreetto, and Marc'aurelio Ranzato. The most of the categories have about 50 images and each image is roughly 300 x 200 pixels. The other categories have about 40 to 800 images and the same size. The most of the images are focused on the object and they have the same scale [1].

1.2. Image-Net

This database is conformed by 995 categories organized according to the WordNet hierarchy (noun). On each node the database has an average of over five hundred images per node. At the end of each node the database has about 100 images per categories (synsets). All the images are quality-controlled and human-annotated. ImageNet compiles an accurate list of web images for each synse and all have a large-scale they aren't focused on the object [2].

2. Classifiers

2.1. Pyramid histograms of visual words

Pyramid histograms of visual words (PHOW) is an important method to classify image, it is a variant of SIFT that classify based into gradient orientations. In Caltech 101 database the images were not divided into train and test but ImageNet was divided. To classify images we used "phow_caltech101" script of "vl_feat" library so we need modify the script to take into account the train images in train stage and test images in the corresponding step so we have two different variable that contained the train images and test images but the classes of two subset data was the same.

3. Results

In order to find the best "phow_caltech101" script parameters in the Image-Net database we change some parameters which we consider necessaries. Accord with our experience the number of categories, the number of training images, and the spatial partitioning could be significant for the result. In this way, first we change the parameters in the original database and observed the behaviour of the algorithm. Then we make the same changes in the Image-Net data base because this data was divided in train and test, so it was necessary modify the script to do train stage with train images and test stage with test images.

3.1. Caltech 101 database

Initially we decide to change the number of categories. This parameter was important because this is the most significant difference between both databases. As we want to see a significant difference we chose the 5, 15, 50, 101 number of categories and obtain the follow results.

Table 1: Results Caltech 101 accord categories number.

Caltech 101				
Num Cat	**Nun Im**	**Sp Part**	**Accuracy**	**Time [S]**
5	15	[2,4]	97.33%	72.84
15	15	[2,4]	83.11%	143.13
50	15	[2,4]	70.93%	405.87
101	15	[2,4]	68.10%	841.95

Accord with Caltech university [1] the most popular number of training images are 1, 3, 5, 10, 15, 20 and 30. In order to got the best results we decided to use this amount of image mix with the best result of the last parameter. The follow are the results obtained with this configuration.

Table 2: Results Caltech 101 accord the image trainning number.

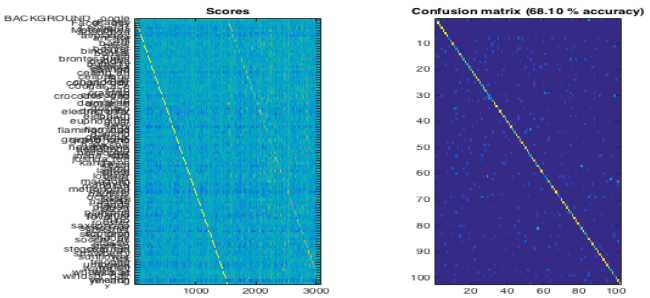
Caltech 101				
Num Cat	**Nun Im**	**Sp Part**	**Accuracy**	**Time [S]**
5	1	[2,4]	22.67%	49.2
5	3	[2,4]	76%	55.16
5	5	[2,4]	86.77%	60.77
5	10	[2,4]	92%	67.98
5	15	[2,4]	97.33%	72.845833
5	20	[2,4]	92.00%	87.20
5	30	[2,4]	96.00%	99.229688

Finally we proved the spatial partitioning. In this case we decided to use the tiny version of the "phow_caltech101" script and change the "phow_caltech101" scrip variable. In this case we chose 2,8,10,20 for the value of this parameter. Additionally we chose the number of categories and the number of training which have the best results. The following is the results obtained with this configuration

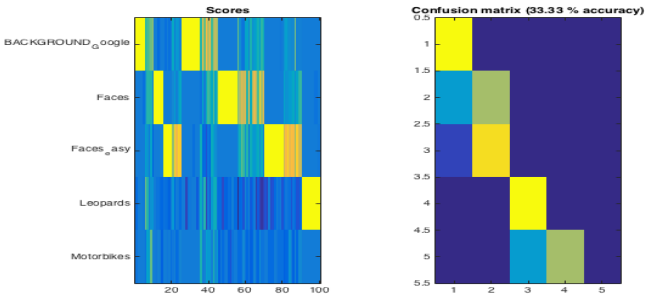
Table 3: Results Caltech 101 accord the spatial partitioning.

Caltech 101				
Num Cat	**Nun Im**	**Sp Part**	**Accuracy**	**Time [S]**
5	15	2	90.67%	73.85
5	15	8	94.67%	73.87
5	15	10	96%	74.60
5	15	20	94.67%	94.67

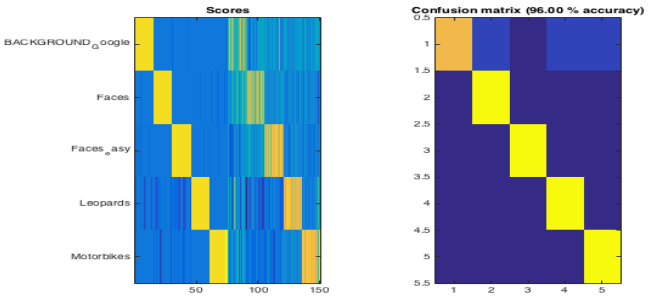
It is evident that the number of experiments, the spatial partitioning and the number of categories have a significant change in the results. In the follow images we can see the improves in the accuracy with the changes of the parameters.



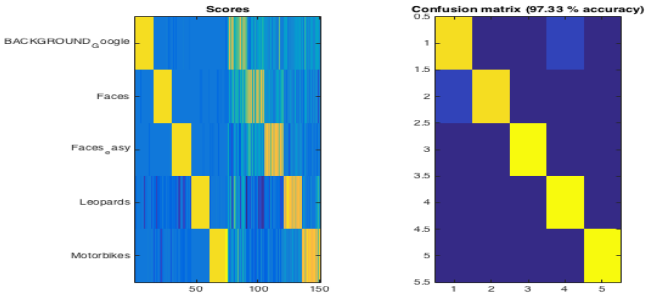
(a) Original Parameters



(b) Changes in number of categories



(c) Changes in number of training images



(d) Changes in spatial partitioning

Figure 1: Confusion Matrix

This changes can give us an idea to the possible results in the Image-Net database. We can see the number of cat-

egories has a negative change in the results, ergo when we have more categories the result turn worst. In the other hand the number of training images have a positive change in the results. In other words, if we have a few categories and many images we should to obtain a better result. Finally, with the space partition change we have both effects. If we have a middle number of this, we can obtain the best result.

3.2. Image-Net database

We did different experiments with this database as in Caltech 101 database changing some parameters to compare both performance and problems. Initially was changed the number of categories. We started with the original script that have 15 train images, 15 test images and 5 categories or synsets obtaining an accuracy of 55.67%. This result can be observed in figure 1.

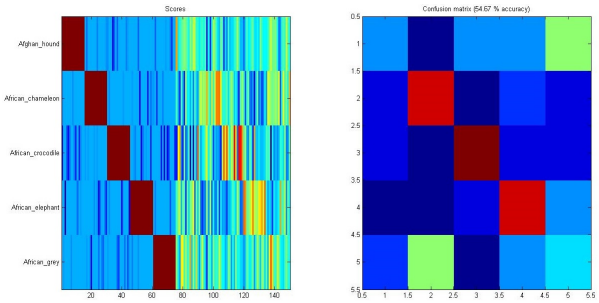


Figure 2: Confusion Matrix 15 train image, 15 test image, 5 categories

The number of categories change from 5 to 995 to compare accuracy and processing time. The results of these variations are show in table 4.

Table 4: Results ImageNet with variations in number of categories

Image Net Database			
Cat	**Accuracy**	**TrainTime [sec]**	**TestTime [sec]**
5	54,67%	34,39	0,002
15	32%	98,54	0,01
50	14,27%	288,43	0,04
101	7,99%	505,98	0,26
995	3,58%	6105,54	9,48

It is evident that the number of the categories affect the accuracy, as greater the number of categories less is the accuracy. In figure 2 can be observed the performance of algorithm with 101 categories that belongs to the ImageNet database. In this case the accuracy was 3,58%. Additionally the processing time increased as the categories were increased.

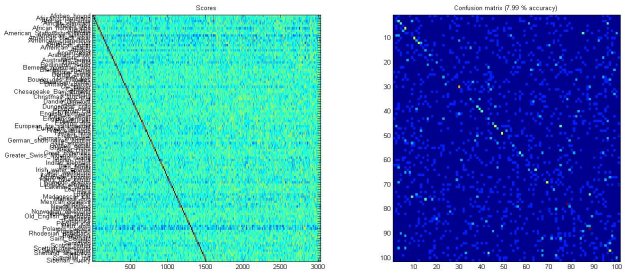


Figure 3: Confusion Matrix 15 train image, 15 test image, 101 categories

Given the above results the best accuracy is obtained with 5 categories, so we continued with variations in the number of train images from 1 image to all images available per category (100). The results of these experiments are showed in table 4.

Table 5: Results with variations in number of train image. 5 Categories, 15 test image

ImageNet database			
NumTrain	**Accuracy**	**TrainTime[sec]**	**TestTime[sec]**
1	40%	12,36	0,0014
3	40%	19,15	0,0019
5	49,33%	25,6	0,002
10	53,33%	36,97	0,0018
20	61,33%	37,82	0,0022
30	58,67%	47,54	0,004
60	72%	64,53	0,003
70	70,67%	71,34	0,004
80	72%	77,6	0,007
90	58,67%	75,44	0,005
100	61,33%	93,66	0,006

Unlike the increased number of categories, in the case of the variation in number or train image, we can observed that the accuracy is better where the number of images in larger but there is a point when the performance again starts to decrease. The best number of train images is 60 because the accuracy is 72%. Nevertheless with 80 images the performance is the same but the computer resource are greater, so for this reason the best is 60 to this parameter. Some examples about these results are presented below:

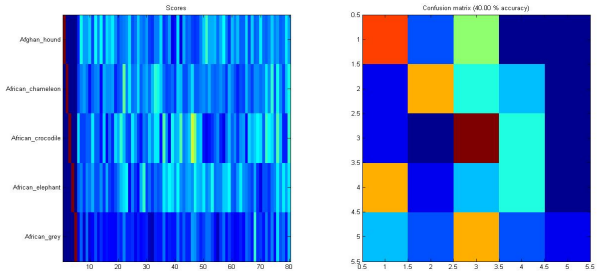


Figure 4: Confusion Matrix 1 train image, 15 test image, 5 categories

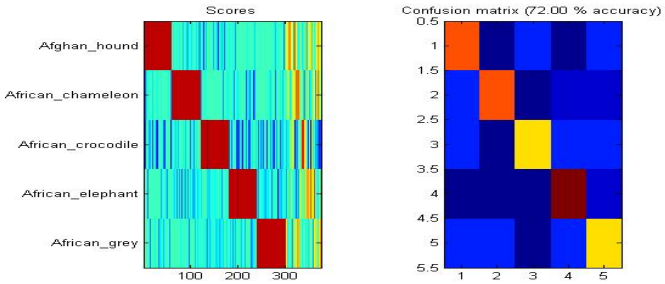


Figure 5: Confusion Matrix 60 train image, 15 test image, 5 categories

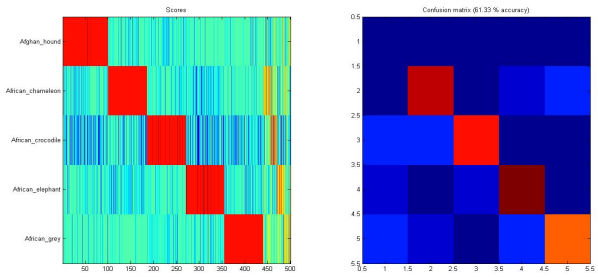


Figure 6: Confusion Matrix 100 train image, 15 test image, 5 categories

In figure 4,5 and 6 we can observe different results examples when we changes the number of train images. If we have few images the model does not classify properly but if we have more images the model is better because it have more possibilities to learn each categories.

Finally we change spatial paritioning in the range from 2 (original of script) to 100. The reason for this is we want see the effect that this parameter have in the algorithm performance. These changes made with the best parameters obtaining in previous experiments. The table 6 show these results.

Table 6: Results with spatial paritioning changes. Train images: 15, test images:15, categories:5

ImageNet database			
SpatialX	**Accuracy**	**TrainTime[sec]**	**TestTime[sec]**
2	72%	64,53	0,003
8	62,27%	69,12	0,0015
10	68%	67,26	0,0019
20	56%	71,71	0,044
50	54,67%	84,3	0,12
100	54,67%	84,37	0,3

It is evident that changes in spatial partitioning affect the performance of algorithm and the processing time; when this parameter is larger the accuracy decrease but the time is greater. So with these results we can say that the best spatial partitioning is 2 because the accuracy to this number is larger (72%). The figures 7 and 8 show some examples of results when changed spatial partitioning and the performance. We can observe that this parameter can affect in 20% the performance, so is very important at the moment of parameters choice.

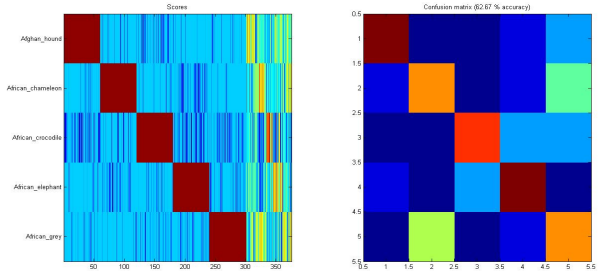


Figure 7: Confusion Matrix 60 train image, 15 test image, 5 categories and SpatialX=8

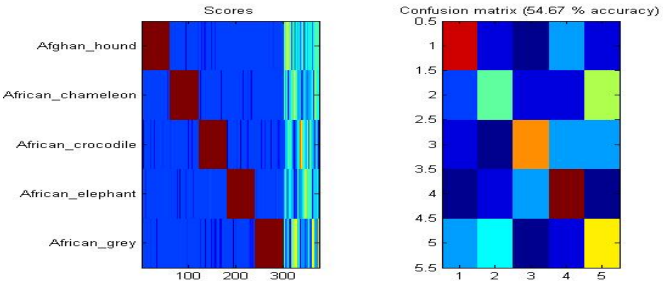


Figure 8: Confusion Matrix 60 train image, 15 test image, 5 categories and SpatialX=50

4. Discussion

PHOW represent one of the most important method to classify images. This method is basically the generalization of the sift but in this case we use more scale for the keypoint. In theory this method is one of the best matching method. However in this lab we realized that the method loses effectiveness as we increases the number of categories.

As we said in the results the best average in the Calthec 101 database is obtained with a few categories, many training images and a middle spatial partition. However this data base is really focus in the object and is not to complicated find a descriptor which achieved a good result. Additionally as we can see in the results section the differences between the accuracies isn't really significant and depend of the categories.

In the ImageNet, the number of categories also affect the algorithm performance. The reason for this is if we have more categories so the model have more possibility to wrong classify. Additionally we can observe that the results of Caltech 101 data base are better than ImageNet; this is because the last database have more variability in images. For this data base the best values of parameters are: train image=60, number of categories= 5 and spatial partitioning=2. In Caltech the best accuracy was 97.33% instead in the other data set was 72%. However in two cases the values of the parameters is different, so we can conclude that the values depend of the database and with this classify method is necessary change the parameters for each database. In other hand the processing time was greater if the size of data was very big, so is important choose correctly the values of parameters to obtain the best results and hold computer resource that can used in other works.

5. Limitations

The most remarkable limitation of this methods is the dependence of the accuracy with the number of categories. Additionally we can see this descriptor only take into account the grey image so the difference between color spaces is not take into account and this feature is important to classify because contain many information about the objects.

6. Improvements

The more intuitive improvement for this algorithm is the use of PHOW color in order to have more dimension and achieve a better results. However the introduce to more dimension couldn't be enough to get the best result. In this case we proposed to train a neural network with these descriptors.

References

[1] L. Fei-Fei, R. Fergus and P. Perona. . *Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories.*. IEEE. CVPR 2004, Workshop on Generative-Model Based Vision. 2004

[2] Olga Russakovsky and Jia Deng and Hao Su and Jonathan Krause and Sanjeev Satheesh and Sean Ma and Zhiheng Huang and Andrej Karpathy and Aditya Khosla and Michael Bernstein and Alexander C. Berg and Li Fei-Fei, *I*mageNet Large Scale Visual Recognition Challenge, (2015) International Journal of Computer Vision (IJCV),volume 115 number 3 pages 211-252

[3] P.Arbelaez. *Lecture 13: Recognition 02*. Computer Vision, Universidad de los Andes.