Image semantic classification using PHOW descriptors

Felipe López
Universidad de los Andes
Bogotá, Colombia
jf.lopez718@uniandes.edu.co

Abstract

The classification of images in computer vision, by semantic recognition of their content, is a wide studied computer vision problem. In this paper, the PHOW solution, first presented by caltech, is presented and characterized using different dataset. This dataset, the IMAGENET datase, is know in the field to be much more difficult so the performance of the algorithm is expected to be reduced in this new set of images.

1. Introduction

The PHOW representation, Pyramid Histrogram Of visual Words, is a descriptor for appearance of the images. This method is an implementation of the SIFT descriptors in a multi-scale level. This algorithm, when applied to the Caltech-101 dataset, its performace is fairly good. In Caltech-101, the classifier achieves 65 percent average accuracy by using a single feature and 15 training images per class. However, this dataset seems to be an inaccurate representation of the real classification problem because the performance is fairly reduced when the dataset changes to a more sophisticated one.

1.1. ImageNet Database [1]

The imagenet database is a set of images organized according to the WordNet hierarchy. In WordNet 100,000 synsets are described, most of them (80,000) being nouns. ImageNet provides and average of 1000 images to illustrate each of the synsets in WordNet. There are more than 100,000 synsets in WordNet, majority of them are nouns (80,000+). In ImageNet, we aim to provide on average 1000 images to illustrate each synset [1]. Figure 1 and 2 illustrate two examples of images from ImageNet from the acorn category. It is of notice that this images are much more complicated, they show the object from different perspectives and illuminations. Also shows the object interacting with other objects (categories) as shown in Figure 1, which makes the classification a lot harder than databases where the object is centered and alone in the image, like the Caltech-101 datset. This results in ImageNet being a fair representation of all the categories and objects that one might want to classify in the computer vision environment. However, for this paper, the sub-set of ImageNet-tiny was used in which only 200 categories are represented.

Figure 1. Example image of the 'acorn' category in ImageNet

Figure 2. Example image of the 'acorn' category in ImageNet

2. Implementation

The classification algortithm using PHOW descriptors was obtained, along with the Caltech-101 dataset, was obtained from [2] . This algorithm was intended to work on Caltech-101 and some adjustments were made for it to work on the ImageNet database. The only modification the function needed to work on the new dataset was to change the path of the dataset from Caltech-101 to ImageNet, and to change the reading image format from *.jpg* to *.JPEG*. Another change was made but this change lacks importance in the algorithm performance, the scores images were separated for the training set and the test set in order to better see the difference between the scores of these two sets.

The Caltech-101 database is in its nature very different from the ImageNet database. Examples of im-

ages from the same category ('accordion') in Caltech-101 (Figures 3 and 4) and in ImageNet (Figures 5 and 6) are provided. Here it is seen that the Caltech-101 dataset could be labeled as 'easier' considering that the objects are centered, with little or no occlusion very little light and focus changes and little or no interaction with other objects. While the ImageNet database is comprised of images with very different light conditions, interactions, occlusions and scales.

Figure 3. Example image of the 'accordion' category in Caltech-101

Figure 4. Example image of the 'accordion' category in Caltech-101

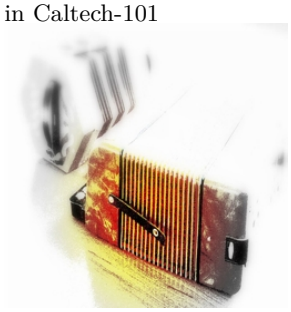

Figure 5. Example image of the 'accordion' category in ImageNet

Figure 6. Example image of the 'accordion' category in ImageNet

3. Results

3.1. Compared performance

The performance of this classification algorithm drops significantly when the database is changed. For instance, when the Caltech-101 dataset is evaluated using only 5 categories, the results scores for the train and test sets shown in Figure 7 are obtained. It can be seen that the performance if very accurate and the train and test scores fit very well, it can later be confirmed in the confusion matrix of this evaluation in Figure 8, where an accuracy of 99 percent shows a very pleasant classification with one simple error between the Faces and Faces_easy categories. However, the picture does not look so good for the ImageNet database. For only 5 categories (as previously done for Caltech-101) the performance of the same algorithm is pretty unsatisfying. The scores for this evaluation are show in Figure 9, showing very low scores for the test set images, and the confusion matrix in Figure 10 tells a similar story where a very low accuracy, for only 5 categories, of 65 percent. Only by chance, the classification had a probability of 20 percent of being right. In both evaluations, the 5 categories were randomly picked and 65 training images were used to train the algorithm to later test 20 images.

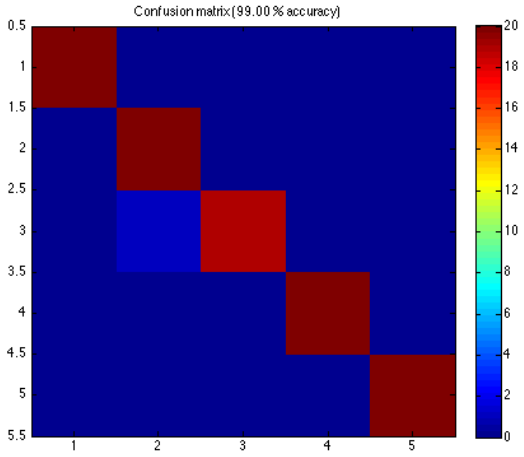

Figure 8. Confusion matrix for 5 categories on the Caltech-101 database

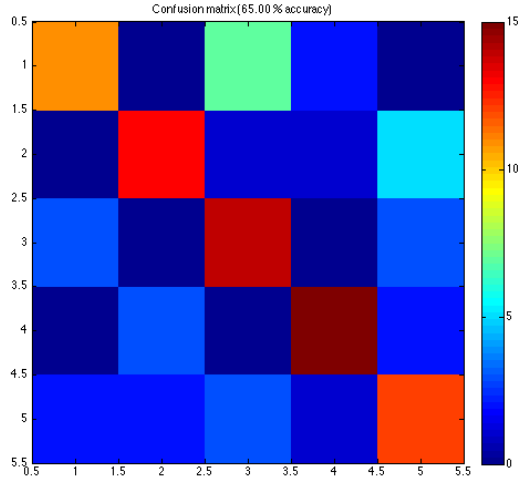

Figure 10. Confusion matrix for 5 categories on the ImageNet database

3.2. Changing the number of categories

When the number of categories increases, the probability of being right by chance decreases. So the real performance of an algorithm can be tested by measuring the accuracy for different number of categories to classify the images. This was implemented

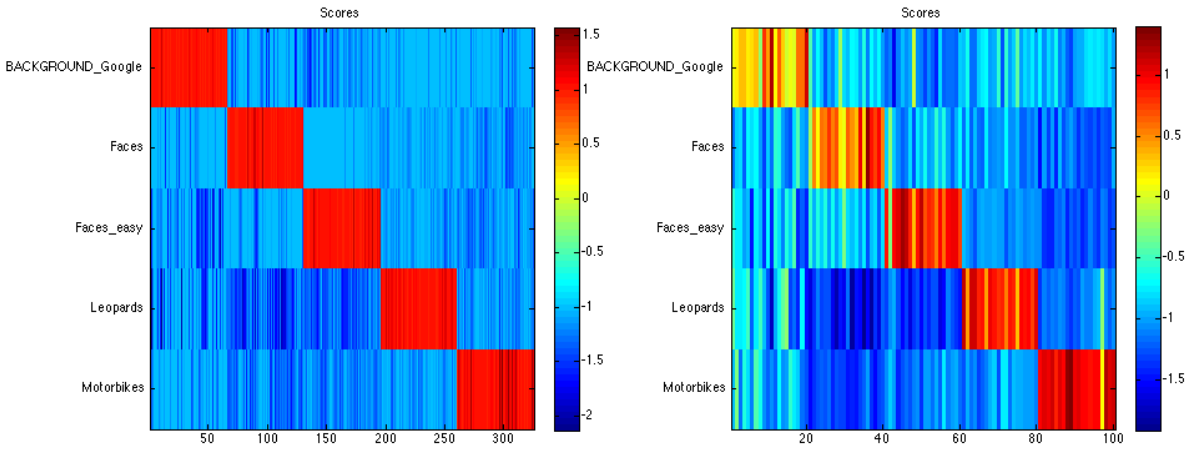

Figure 7. Scores for training and test sets for 5 categories on the Caltech-101 database

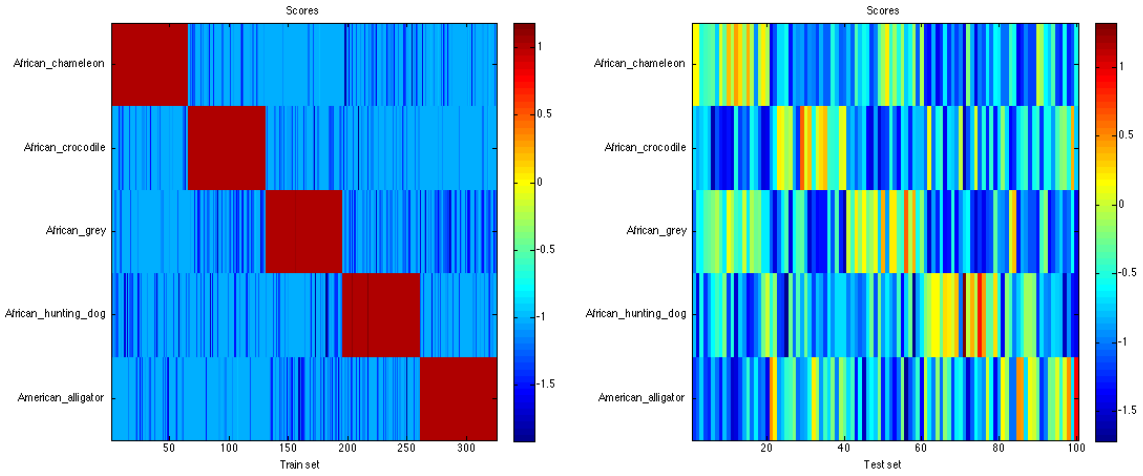

Figure 9. Scores for training and test sets for 5 categories on the ImageNet database

for 2,5,7,9,10,12,15,20,25,30,50,60,75,90 number of categories, the training set was held constant in 65 images and the test set in 20 images. The results of the accuracy are shown in Figure 11. Also the computational cost was addressed by measuring the running time of the algorithm for each number of categories, the results were summarized in Figure 12. The accuracy value has almost an exponential form when seen as a function of number of classes. For only two classes, the accuracy of only 95 percent shows how unpleasant this algorithm works for this dataset. From 5 categories and further the algorithm performance decreases significantly reaching below 35 percent for 90 categories. This indicates that the algorithm is useless when a task requires a classification in many categories. For the computational cost show in Figure 12, it was expected to have a linear behavior considering more classes lead to more images to be trained and tested. This indicates that as the number of classes increases, the algorithm confuses more between categories and the aid of chance is reduced, rendering the algorithm unsatisfying.

3.3. Changing the size of the training set

As the size of the training set increases, the algorithm gets more precise because it has more information to base its decision on. However, if the training set is too big the algorithm might fall into over-fitting resulting in a reduced performance on the test set. It is then of importance to measure the adequate size of the training set for the classifier to work best. This was implemented for training sets of 5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100 images and the accuracy of this implementation is shown in Figure 13. Here is shown the over-fitting

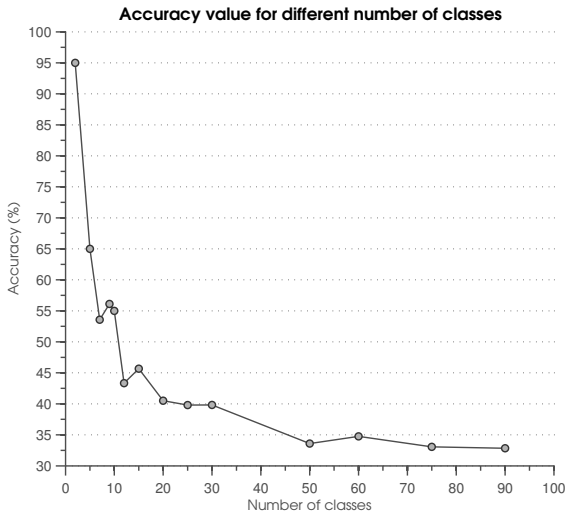

Figure 11. Accuracy for different numbers of categories

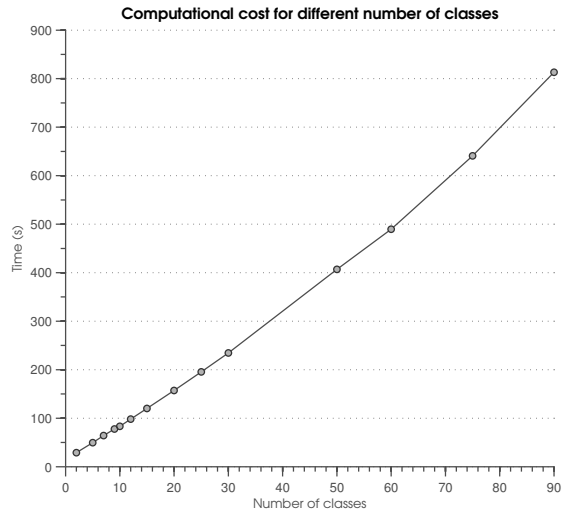

Figure 12. Computational time for different numbers of categories, done in an 8GB RAM Macintosh computer with a 2.7 i7 Intel processor

example where past the set of 90 training images the accuracy is reduced significantly due to the model being too adjusted to the training set. Under-fitting is also seen when the training set is too small, as with the 5 images set. This implementation was done by holding constant to 15 classes and 20 training images, so it should be taken into account that only by chance the probability of being right is of 6 percent. One could say, then, that the best training set consist of 75 images due to the accuracy it shows and the computational efficiency compared to the set of 90 images. This computational efficiency is shown in Figure 14 as the time it took the algorithm to run for each set. In this curve it is of interest that is linear as in the previous example but it reaches a plateau after the 75 images set.

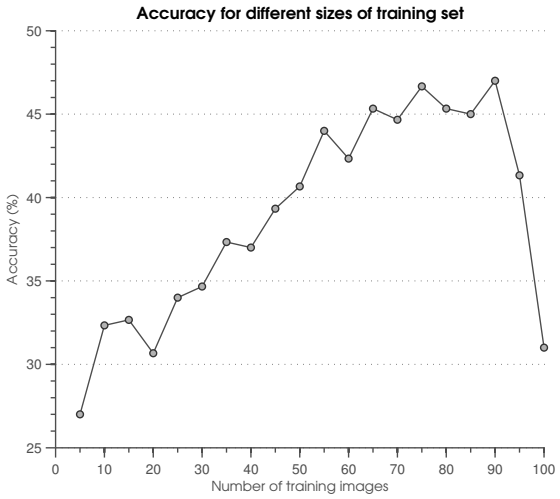

Figure 13. Accuracy for different sizes of training images sets

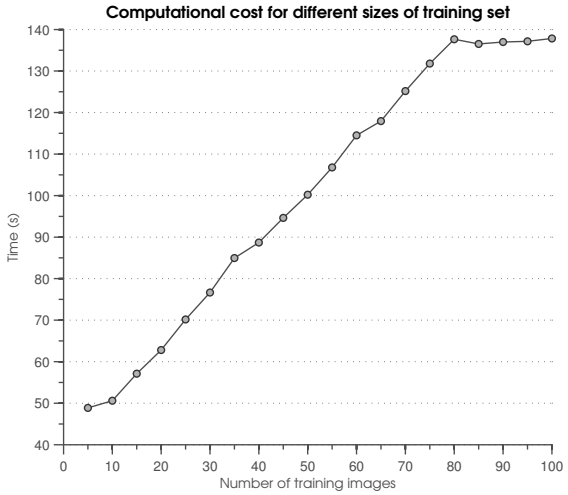

Figure 14. Computational time for different sizes of training images sets, done in an 8GB RAM Macintosh computer with a 2.7 i7 Intel processor

3.4. IMplementation in the entire dataset

To evaluate the overall performance in the entire dataset, the algotithm was implemented with parameters chosen from the previous implementations by accuracy and computational cost. The parameters chosen were 75 training images and 20 test images, as it was going to be evaluated in the entire database the number of classes had to be 200. The scores of this implementation are show in Figure 15 and the overall

accuracy of 25 percent, which is considered very low, is shown in Figure . The computational time of this implementation was 6,8883 seconds. This shows that this entire database, considered to be a fair representation of the categories one might want to classify, is too complicated for this algorithm, and in order to be improved the algorithm needs to take into account the variability of the objects to be classified.

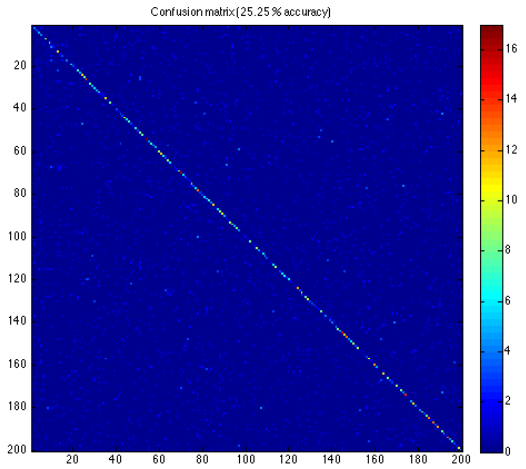

Figure 16. Confusion Matrix for the performance on the entire database

4. Discussion and Conclusions

Considering the results shown, it is straightforward to say that the Caltech-101 database, although it provided a lot of imporovement in this area, is not an adequate representation of this problem considering that the performance of the same algorithms is significantly reduced when evaluated on more complicated datasets. In this paper, it was also shown, the importance of parameters such as the size of the training set and the number of categories evaluated on the overall accuracy of the algorithm. One needs to adjust this parameters correctly in order to obtain the best performance of the algorithm, considering the best performance is still a very poor performance in this database. Improvement needs to be made in order to make this algorithm useful in a real problem and not only in a fairly easy database, for instance the algorithm needs to be able to account for object variability and interaction between other objects to improve the accuracy.

References

[1] Image-net.org, "Imagenet," 2015.

[2] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008.

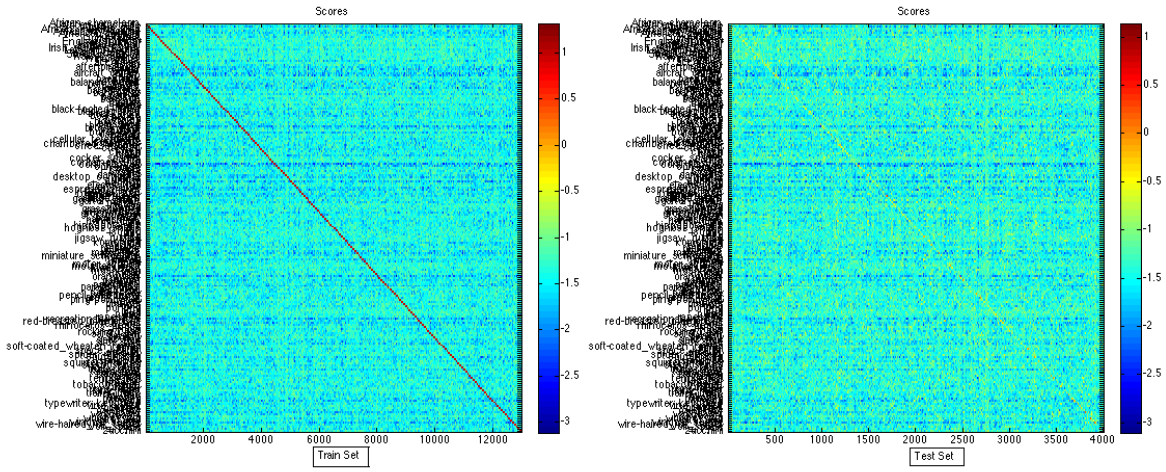

Figure 15. Scores of training and test sets for the performance on the entire database