PHOW Classification

Silvana Castillo
Universidad de los Andes
Cra 1 #18a-12
ls.castillo332@uniandes.edu.co

Laura Daza
Universidad de los Andes
Cra 1 #18a-12
la.daza10@uniandes.edu.co

Abstract

In this paper we adapt the code phow_caltech101 to work on a subset of the ImageNet dataset trying to solve a more complex recognition or classification problem by using dense-sift descriptors. We made numerous experiments, changing different parameters (number of categories, number of words, size of the dataset and spatial partition) with the purpose of finding the best results, evaluating the performance by the accuracy, execution time and confusion matrix. We found that increasing the number of categories decreases the performance, by increasing the number of words we found a plateau in the accuracy and the changes made in the spatial partitions don't affect much the performance.

1. Introduction

In this paper we use dense-sift descriptors for image recognition in a subset of the ImageNet dataset by adapting the code *phow_caltech101* and changing different parameters with the purpose of finding the combination that results in a better performance (evaluated by accuracy, execution time and confusion matrix).

2. Database

The Caltech 101 dataset consists of a total of 9,146 images, split between 101 different object categories (plus the background/clutter category). Each object category contains between 40 and 800 images where common and popular categories such as faces tend to have a larger number of images than others. Each image is about 300x200 pixels. Additionally, images of oriented objects such as airplanes and motorcycles were mirrored to be left to right aligned and vertically oriented structures such as buildings were rotated to be off axis.[2]

ImageNet is an image dataset organized according to the WordNet hierarchy. Each meaningful concept in WordNet, described by multiple words or word phrases, is called a "synonym set" or "synset". There are more than 100,000 synsets in WordNet, majority of them are nouns (80,000+). In ImageNet, they aimed to provide on average 1000 images to illustrate each synset where all the images of each concept are quality-controlled and human-annotated. It contains full resolution images with an average size of around 400×350 pixels.[1]

3. Description of the recognition method

We implemented the code *phow_caltech101* which is a dense-sift descriptors for image recognition (set by default in the Caltech 101 database) the uses the *vl_feat* library (open source library that implements popular computer vision algorithms specializing in image understanding and local features extraction and matching). This algorithm can be divided in 5 stages:

1. PHOW features that are dense multi-scale SIFT descriptors (coarse statistic of the gradients of the frame appearance. Due to canonization, the descriptors are invariant to translations, rotations and scalings of the image. Due to their statistical nature, they are also very robust to other and not modeled sources of noise).

2. k-means for fast visual word dictionary construction.

3. Spatial histograms as image descriptors.

4. A homogeneous kernel map to transform a Chi-square support vector machine (SVM) into a linear one

5. SVM classifiers to create a model (to predict results). [4]

The only adjustments we had to made to adapt the algorithm to the database was to change the paths (specifies a unique location in a file system) so it read instead the ImageNet dataset, as both datasets had each category labeled by the name of the file in which its images were, there wasn't a huge difference when reading the data. Also, we changed some values depending of the experiments the number of categories, words or spatial partitioning.

The algorithm by default trained and tested only in the dataset of training with an output of a model and the results of accuracy and confusion matrix. For this reason, we had to create a new function to predict the performance of the algorithm over dataset of testing with the results of accuracy and confusion matrix.

4. Training and test results

4.1. Results in Caltech 101 dataset

By default 15 training images are used, which result in about 64% performance, this is a good performance considering that only a single feature type is being used and there is a really small number of images.

The results of table 1 to 3 and figures 1 to 8, correspond to the changes of number of categories.

4.2. Results in subset of the ImageNet dataset

Table 1. Accuracy and execution time results in train set

# Categories	Accuracy (%)	Time (min)
5	20	1.164
30	32	3.031
55	25.20	8.149
80	12.40	11.733
105	9.26	15.556
250	3.04	42.454
400	0.00	76.185

Due to the results obtained in table 1, we decided to run the same between the number of categories 24 until 27, looking for the optimal quantity of categories for the best performance

Table 2. More Accuracy and execution time results in train set

# Categories	Accuracy (%)	Time (min)
24	31.08	2.150
25	30.88	2.489
26	30.47	2.531
27	32.24	2.625
28	31.63	2.851
29	29.49	3.018

Table 3. Accuracy and execution time results in test set

# Categories	Accuracy (%)	Time (min)
5	42.2	3.807
27	29.85	22.873
30	29.77	23.155
55	25.16	57.904
80	18.613	98.226
250	13.996	218.348
400	12.578	674,244

The execution time of 400 categories in table 3 gets really high, we think it could be caused because the virtual machine was saturated with jobs of other tasks.

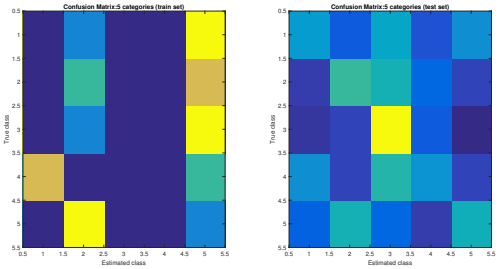

Figure 1. Confusion matrix for 5 categories

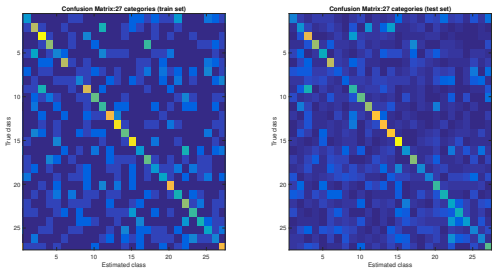

Figure 2. Confusion matrix for 30 categories

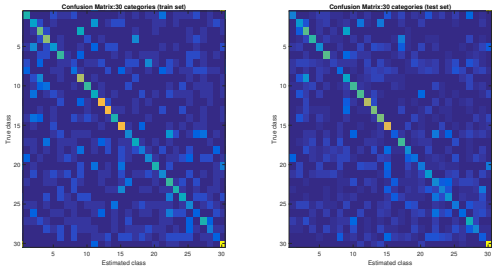

Figure 3. Confusion matrix for 30 categories

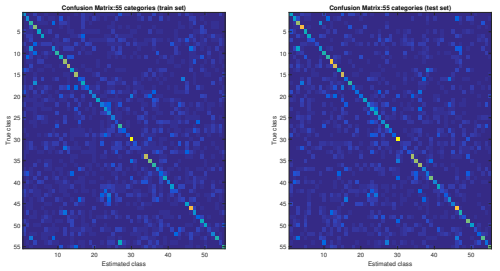

Figure 4. Confusion matrix for 55 categories

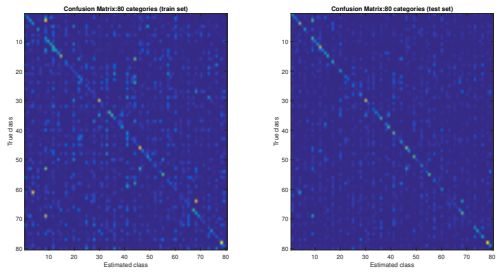

Figure 5. Confusion matrix for 80 categories

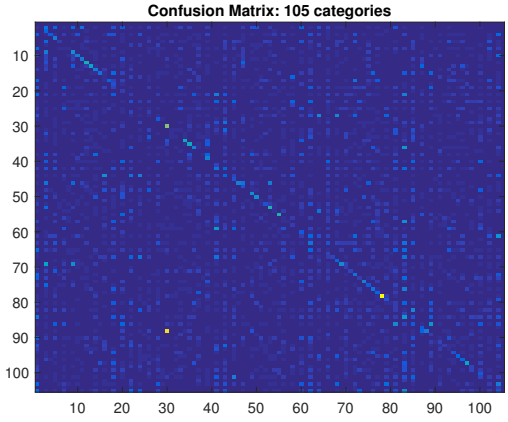

Figure 6. Confusion matrix for 105 categories in train set

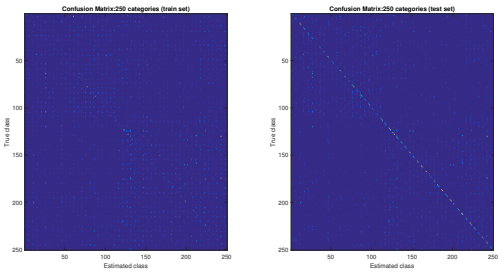

Figure 7. Confusion matrix for 250 categories

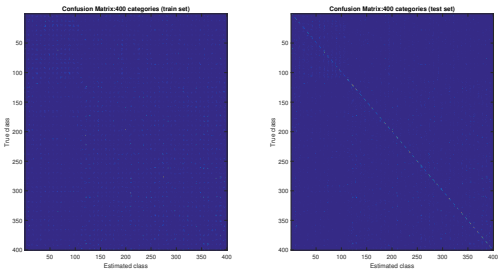

Figure 8. Confusion matrix for 400 categories

The results in table 4 and 5, corresponds to the changes of number of words:

Table 4. Accuracy and execution time results with 5 categories in the train dataset

# Words	Accuracy (%)	Time (min)
20	44	0.654
40	48	0.763
60	56	0.771
80	48	0.855
100	46	1.289
200	52	1.351
300	52	1.416
400	52	1.486
500	52	1.812

Table 5. Accuracy and execution time results with 5 categories in the train dataset

# Words	Accuracy (%)	Time (min)
20	41	4.932
40	39	9.859
60	41	15.694
80	41	21.598

The results in table 6, corresponds to the changes of spatial partition in the train dataset:

Table 6. Accuracy results with 5 categories changing X and Y partitions

numSpatialX	numSpatialY	Accuracy (%)
2	3	52
3	2	52
3	3	52
3	4	48

5. Discussion of the results

Accuracy is not a reliable metric for the real performance of a classifier, because it will yield misleading results if the data set is unbalanced (if the number of samples in different classes vary greatly). For example, if there were 95 cats and only 5 dogs in the data set, the classifier could easily be biased into classifying all the samples as cats. The overall accuracy would be 95%, but in practice the classifier would have a 100% recognition rate for the cat class but a 0% recognition rate for the dog class. For this reason we compare not only by measuring the accuracy but also using the confusion matrix of every experiment (test with different conditions), This allows more detailed analysis than accuracy (proportion of correct guesses). [3]

5.1. Differences between datasets

The biggest difference between Caltech 101 and Imagenet is the quantity of categories, the Caltech 101 data set represents only a small fraction of possible object categories and the most simple categories, while ImageNet is

bigger and has categories that are much more complex. [1], [2].

Other difference is that images in Caltech 101 are very uniform in presentation, aligned from left to right, and usually not occluded. As a result, the images are not always representative of practical inputs that the algorithm might later expect to see. Under practical conditions, images are more cluttered, occluded and display greater variance in relative position and orientation of interest objects, as they are in ImageNet dataset. [1], [2].

Based on the previews analysis, it was expected that the results in the ImageNet dataset where going to be less satisfactory that the ones in caltech101, the former dataset makes a more complicated problem.

5.2. Number of categories

In regards of execution time, the results of table 1, 2 and 3 show that as the number of categories increases the execution time gets higher and higher, this is because the number of images that the algorithm has to process is increasing. Then, the number of categories and execution time are values directly proportional.

Originally this algorithm always run the model over the 101 categories (Caltech 101 dataset) and they always add more images but the categories never change. However in this case, we increase the number of images by adding more categories. Therefor, the performance of the algorithm is good with few categories and decreases while they increase because the algorithm has more and more categories with which it can get confuse, the problem gets more complicated with more categories and the results obtain less accuracy. Until, at around 400 categories the accuracy decreases almost to 0 (to many categories to complicated to classified).

In regards of the confusion matrix, we obtained an unusual result: the performance of number of categories from 5 to 400 where always better in the test set as seen in figures 1 to 8 (It can be more distinguish the diagonal line). This is unusual because as we train the algorithm (SVM) with the train data, it learns to predict with examples identical to the ones tested, then it should get a higher score.

However, there is a consistency in the best classes (the ones that get less confuse - more yellow in the figures) thought out all the experiments, as it is shown in 1 to 8, while the number of categories is increasing the best classes that got a god result with less categories also obtains a good result (sometime it decreases). That means that there are some classes that are "easy" to classify (don't get confused so much with other categories), for example categories 3, 15, 30 and 78 (academic_gown, African_hunting_dog, American_coot and bassinet respectively).

5.3. Size of training set

As we said before, the only way in which we are increasing the training set is by increasing the number of categories. Nevertheless, in this case the total number of images is getting higher but the number of images that represent each category never changes (100 images per category). then while the total of images is increasing, the percentage of images that represents a category is getting smaller and this cause the results of accuracy shown in table 1 and 3.

For this reason and base on the results of table 1, we looked for the optimal number of categories that with just the 100 images per category could obtain a good accuracy result (and be greater than 5 categories). As shown in table 2, we obtained that 27 categories was the optimal number, with an accuracy of 32.49%

5.4. Number of words

The results in table 4 show that increasing the number of words increases as well the accuracy of the algorithm, duplicating the initial quantity of words we get better accuracy, with a optimal number of words equal to 60 (accuracy of 56% in train and 41% in test). However, after 200 words in the train and over 80 words in test, we found a plateau in the improvement of accuracy, this means that quantity of words are more than enough to describe all the categories (5) and classify them, if we add more the results will not improve by much. Additionally, if we increase the number of categories then probably the quantity of words must increase as well (not in the same proportion) to obtain a good performance.

5.5. Number of spatial partitions

The changes we made in the number of the spatial partitions in 5 categories, shown in table 6 weren't very significant, because the accuracy stayed in 52% in almost all the experiments and it decreased when the partition got greater in the last experiment (48%), which can be due to the small size of every window resulting in less significant histograms than the previous ones, but still this change was minimal an we can conclude that the spatial partition doesn't affect much the accuracy of the algorithm (at least in the ranges used for the experiment). Nevertheless, we think is necessary make additional experiments, changing the partition with values more radical (greater) to be sure of the previews conclusion.

6. Limitations of the method

The biggest limitation of the method is that it needs a huge dataset (more than 100 images per category) to obtain good results, at least in ImageNet (as we said before this dataset provides a more complicated problem). However, at

the same time if the number of images increases too much, the execution time will increase to the point of being too expensive computationally. Also, to run the code it is necessary to have a huge dataset and also a library not included in Matlab, meaning that it takes a great amount of time to acquire them and to be capable of running the code for the first time. Additionally, the output results depend highly on the different input parameters, making it necessary to perform several tests, changing each one of the parameters, to find an appropriate combination and obtain the best result possible.

7. Possible improvements

One possible way to improve the results is by increasing the number of images that represents each category and with some experiments we can find the proportion of images per category that has to be increased and would permit and optimal performance with a fixed number of categories without causing over-fitting. Or instead, the optimal number of categories for each augmentation of images per category.

References

[1] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) *ImageNet Large Scale Visual Recognition Challenge*. IJCV, 2015.

[2] Fei-Fei Li, Marco Andreetto, and Marc 'Aurelio Ranzato. *Caltech 101 dataset*. COMPUTATIONAL VISION AT CALTECH. `http://www.vision.caltech.edu/Image_Datasets/Caltech101/` Collected in September 2003.

[3] Recovered from `http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html`. Department of Computer Science, University of Regina.

[4] VLFeat *VLFeat open source library, phow_caltech101() code and additional information*. Recovered April 7, 2016from: `http://www.vlfeat.org/`