Image classification

Alejandro Pardo
Universidad de los Andes
la.pardo2014@uniandes.edu.co

Lina Barbosa
Universidad de los Andes
lm.barbosa1099@uniandes.edu.co

Abstract

We implemented an image classification method based In general terms, this method construct image classifiers using PHOW (dense sift), spatial histograms of visual words and a Chi2 Support Vector Machine (SVM), which will be used to assign specific labels to input images. The purpouse of this work is to get closer to the performance on the algorithm over the Caltech 101 database, approximately 60%

1. Introduction

Image Category Recognition is one the most relevant tasks in computer vision and over the past year there have been lots of related works trying to improve this techniques. However, despite of their actual use in commercial applications such as Photosynt, it is yet a real challenging problem, because no one has yet achieved the performance level of a human in this task [4] A remarkable thing in this area is that all of the created algorithms for image classification must be general. It means that it must be able to achieve a high performance not only over its own database. Based on that, this paper is focused on the implementation of the classification method included in VLFeat library which uses Pyramid Histogram of Visual Words to create appropriate classifiers to give an image a specific category. This algorithm obtain approximately an overall precision of 60 over the Caltech 101 database and we tried to get closer to this value implementing it in the Imagenet database.

2. Classification Method

We implemented the classification method included in VLfeat library [5] on the ImageNet database[3]. In general terms, this method construct image classifiers using PHOW (dense sift), spatial histograms of visual words and a Chi2 Support Vector Machine (SVM), which will be used to assign specific labels to input images.

Pyramid Histogram of Visual Words (PHOW) is a model in which image features are treated as words. The main idea consists in creating an histogram of the frequency of those words (image features) inside the image. Cause the spatial information is really important, the model works by dividing the image into increasingly fine sub-regions, which are called pyramids. The histogram of visual words is then computed in each local sub-region.[2]

At the first stage of the method, a feature dictionary is created using dense SIFT descriptors over the selected training images and then quantized using k-means clustering to form a visual vocabulary constructed of N words (clusters). Based on this, given an input test image, its PHOW features are extracted using dense SIFT descriptors at different image scales and then are quantized into visual words using the dictionary created and a KdTree model to assign it to the nearest word in the dictionary. After this, an spatial histogram is generated involving all the pyramids (image subregions) through the creation of spatial bin in x and y axis depending of the selected image partition. Based on the spatial histogram, a feature map is creating using an homogeneous chi squared kernel map. In the last stage of the algorithm, a nonlinear Chi2 SVM classifies each of the test images using its own features map, giving it a label which will be compared with its ground truth to finally generate a confusion matrix.

Due to the change of database it was necessary to adjust some parameters of the algorithm in order to get a good performance on the new one. It resulted relevant because one of the characteristics of the Caltech 101 database was that all of the objects appeared centered in the image and there was a few additional information inside it, that means that in some cases the object was not in context but just in a white background or that there was just one object per image. On the other hand, the Imagenet database showed all of the objects inside a real context making them closer to the real life. Based on this, the first adjusted parameter was the number of words or clusters in the construction of the PHOW features dictionary. Because the amount of information in the new dataset was higher, it resulted necessary to increase the number of clusters in which the features could be reorganized with the purpose of increase the probability of discriminate between images. On the other hand and based on the same reason, the spatial partitioning was

adjusted too. Because of the increase of the information in the images and of the background, more image partitions in the construction of features maps was necessary. Because of the increase of image information, the local analysis and extraction of features resulted relevant in order of eliminate the effect of the image background and object size in the classification process. Finally, because the images in the new database are presented in real context, the size of the objects could be not always the same, as in Caltech 101 (centered object and similar size). Because of this, it results convenient to analyze the effect of the scale in the features extraction with the objective of decrease possible detection error due to changes on objects size in the database.

3. Database

To evaluate the described method the Imagenet database was used[3] . This is a dataset of 199400 labeled images belonging to 996 different categories, based on the WordNet hierarchy[1]. The hole database was divided into train and test sets, both of 99600 images, including 100 images for each of the categories. All of the images were RGB JPEG format with size of 256x256 pixels.

4. Results

We performed a few experiments in order to find out the most relevant parameters of the algorithm. First, we made 18 experiments by changing the number of images per category used for training, the number of target categories and the number of divisions made to the image to compute the descriptors. We mix all of the parameters, and we obtain a total of 18 experiments, which results are shown in table 2 in annex. After the training, we evaluate the estimated model into the training database in order to find out how was the model adjusted to the data, this results are shown in table 1. Without surprises, the accuracy of the estimated model into the training database was 1 in all the cases. This experiments were made with the original parameters of the function.

In order to give an adjustment to the algorithm to the new database, we changes a few parameters as is mentioned in section 2. Then we changed the number of words from 300 to 500, number of scales from 1 to 2. We evaluated the different combinations of this parameters into a test database formed by 10, 50 and 80 categories. Results can be seen in annex, within table 4. Again, we performed a validation by classifying, with the estimated model, the images used for training, getting results showed in table 3 in annex.

Image 1 show us the confusion matrix of one of the experiments, Test images was 50 in all the cases and we can see that the max accuracy is 60% for only one of the categories, which show us that the method may not be very good on this data base.

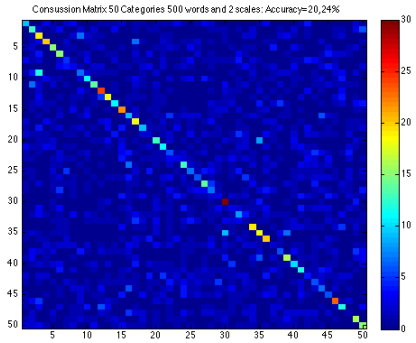

Figure 1. Example of result of the experiments

5. Discussion

As we can see in tables 2 and 4, the accuracy of the algorithm decrease rapidly with the number of categories included. We can observe in table 2, the decreasing of the performance between 10 and 50 categories, the mean accuracy decrease between these tow values is approximately 20%, which means that the performance is reduced about a 60% with the increasing of the number of categories. However, we also can see that differences between 50 and 80 categories are quite smaller than differences between 10 and 50. This could mean that the complexity of the problem depends in the scale of this, problems with same scales may have same complexity. The behavior of table 4 is very similar as the table 2, explained previously.

Again we will observe table 2. We can see that as the number of images increase, the accuracy increase, and that is the expected result.We already know that a good algorithm that wants to solve a difficult problem, needs to have a several amount of information in order to learn all the variations between features of the same object and also to have a good estimated model. We can see this reflected in table 2, with the better results obtained when the number of images for training by category increase to 80. One disadvantage of the increasing of the data, is the increasing with processing times. We can see in the table 5 that changes between number of categories and numbers of images are very significant, and depend on the applications this could be an important factor.

Now, between the number of partitions in X of the image, we observed that differences were not quite important. We can see in the best case, an increase of the accuracy in 0.3%. If an increase in computational cost would exist, option of increase this value would be discarded immediately. However, the increase in train time is only 2 seconds, as we can see in table 5. In conclusion, this parameter is not the more important, because it doesn't help with the performance of the algorithm and neither with the computational

cost.

Furthermore, we can see results shown in table 3 which show us that number of Words was a important parameter, with the increase of 200 words, table shows that for the algorithm seems to be easier to describe the images with more words. We can see increments of the accuracy in all the cases which the number of words go from 300 to 500. In all the cases, the increase of accuracy at leat 2% which means that this value plays an important role in this classification problem. Moreover, we increase the number of scales evaluated to compute the descriptor from 1 to 2 scales, and we observed that this parameter was also quite important to the improvement of the accuracy, this results can be seen in table IV. Again we can see improvements of 3% in every accuracy that increases its number of scale.

Another important thing to stand out is the performance of the classifier. We can see the different accuracy obtained for the Test images in tables II and IV and conclude that the features that are used to describe the images are not enough for the classifiers to make a good model. However, table I and III show us that the non- linear classifier is adapting perfectly to the data, so its behavior is pretty good. So we can conclude that the classifier used is the indicated to approach to the problem, but we have to choose better features for it.

In conclusion, these two parameters would be the indicated to perform more experiments and improve significantly the results in the data base. These can be explained trough the complexity of the descriptor used, in both cases descriptors was designed for Caltech-101 that was a simple data base in contrast with Imagenet. When we increase the number of words we increase the possibilities for the descriptor to differentiated between objects that with 300 words was similar. Also, scale is always an important factor in vision, and in the most of the problems this parameters increase the performances of the algorithms.

6. Limitations and improvements

The principal limitation of this method is the data base for which was designed. Caltech-101 was a database with uniform images, with only one object centered and easily differentiated. With this characteristic is really evident that the performance in other data base like Imagenet will be smaller. Imagenet images are not uniform, have clutter and many other features that we already mentioned, that make the problem quite more complex. This could be improved by the adjustment of the parameters in the classifier as it was explained previously.

Another limitation of the algorithm is centered on the computing resources needed to train the classifier. In this case, where only a part of the training set was used, the time needed to do this was not so long. Nevertheless, it was possible to observe in table VI that when the number of training images per category and the amount of categories trained increase the time needed to train the classifier was higher. Based on this, we can conclude that the computational cost required for the train stage will be really high in the case that all the train set was used. In this case, the method must be improved by optimization algorithms that could try to distribute tasks in pro of reducing the operation time. Nevertheless, in real time applications this limitation could be reduced if the training model is created previously. As is shown in table VI, the test time is really lower than the train time, so if the information required to classify the images is already obtained, the test process could be faster and adequate to real time classification.

Finally, the most important improvement to this and the other methods of classification is to use information of low and high level, both of them. Maybe using a technique bottom-up or top-bottom to contemplate other kind of information that is not associated to local regions of the image, but it is information that can be extracted with relations and interactions between the pixels, regions and even objects of the image. This can be done, by mixing methods of segmentation as first approach and then some kind of recognition and finally classification.

References

[1] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[2] S. Khaligh-Razavi. What you need to know about the state-of-the-art computational models of object-vision: A tour through the models. *CoRR*, abs/1407.2776, 2014.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[4] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[5] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

Annex

Table I. Accuracy Categories vs Divisions vs # of training- Train Data Base

	Divisions	2			4		
	# of training	10	50	80	10	50	80
Categories	10	1	1	1	1	1	1
	50	1	1	1	1	1	1
	80	1	1	1	1	1	1

Table II. Accuracy Categories vs Divisions vs # of training- Test Data Base

	Divisions	2			4		
	# of training	10	50	80	10	50	80
Categories	10	0,306	0,356	0,42	0,292	0,344	0,424
	50	0,1188	0,1332	0,1644	0,112	0,1332	0,1752
	80	0,08875	0,0965	0,123	0,09425	0,10225	0,13325

Table III. Accuracy Categories vs Scales vs Number of words- Training Data Base

	Escala	1		2	
	# of words	300	500	300	500
Categories	10	1	1	1	1
	50	1	1	1	1
	80	1	1	1	1

Table IV. Accuracy Categories vs Scales vs Number of words- Test Data Base

	Escala	1		2	
	# de palabras	300	500	300	500
Categories	10	0,448	0,452	0,462	0,484
	50	0,186	0,2024	0,2164	0,2248
	80	0,13775	0,15325	0,1635	0,1795

Table V. Times for training

	Divisiones	**2**			**4**		
	# de training	**10**	**50**	**80**	**10**	**50**	**80**
Categories	**10**	20,74	20,51	22,70	19,68	22,56	23,33
	50	51,10	57,22	78,14	50,30	57,28	81,19
	80	80,35	84,20	123,88	83,87	87,26	135,20

Numbers given in seconds.

Table VI. Times for Test

	Divisiones	**2**			**4**		
	# de training	**10**	**50**	**80**	**10**	**50**	**80**
Categorie	**10**	0,02	0,01	0,01	0,01	0,02	0,02
	50	0,04	0,03	0,05	0,05	0,05	0,07
	80	0,06	0,09	0,07	0,10	0,12	0,13

Numbers given in seconds.