Pyramid of Histograms of Visual Words (PHOW) for image classification

Johana M. Ramirez Borda
Los Andes University
jm.ramirez11@uniandes.edu.co

Francisco A. Rozo Forero
Los Andes University
fa.rozo1843@uniandes.edu.co

Abstract

The SIFT method consists on extracting image features that are invariant to image scaling and rotation so they can be used for further matching and recognition of objects regardless of their position or orientation. One variationo of this algorithm is the dense-SIFT, which instead of use DOG to identify interest points, find features in the whole image to create the HOG (histograms of gradients). Likewise, PHOW is a method that implements the dense sift algorithm at an image for multiple window sizes for the Gaussian kernel. In this work, the Caltech101 and ImageNet datasets were used in order assess the PHOW method provided with the VL_Feat library found in [9]. First, the best combination of parameters for Caltech101 was found obtaining an ACA of 0,73. Then, when using this parameters in ImageNet, the ACA was far below being of 0,19. For this reason, the best combination of parameters for ImageNet was found obtaining an ACA of 0,2385.

1. Introduction

The SIFT method (Scale-Invariant Feature Transform) was first proposed by David Lowe in 1999 and published in 2004 [6]. It consists on extracting image features that are invariant to image scaling and rotation, and partially invariant to changes in illumination and 3D camera viewpoint, so they can be used for further matching and recognition of objects regardless of their position or orientation. Firstly, the algorithm uses difference of gaussian (DOG) pyramid of images to identify the key points where features don't change at multiple scales or orientations. Second, a model is fitted at each key point in order to determine location and scale. Then, local gradients of oriented directions are measured to assign one or more orientations to each key point and, at this point the data has been transformed relative to its orientation and scale and therefore invariance is provided. Finally, local gradients are measured in the region around each point and are transformed too. These compose the called bag of features, which is a list of characteristics for images[7].

The dense-SIFT method is similar to the previous explained but differs on the identification of the interest points. While SIFT uses DOG to identify those points and then build the histograms of gradients (HOG), dense-SIFT doesn't identify interest points, but divides all the image in overlapping regions and uses all of them to build HOG [3]. Therefore, it finds the features for the whole image (pixel windows every [x,y] step).

It has been found that when testing classification algorithms, the best results are obtained using dense-SIFT. The explanation is that with dense SIFT, there can be obtained a larger number of descriptors and therefore one gets more information than that obtained with sparser set of points [5].

With this in mind, PHOW is a method that implements the dense sift algorithm at an image for multiple window sizes for the Gaussian kernel [8]. Therefore, PHOW is scale invariant since it repeatedly iterates the dense sift code by varying the scales (σ) that are used when identifying the features.

This method uses as hyperparameters the different window sizes that will smooth the image when applying the gaussian filter before every dense sift algorithm, and the step size, which is every X and Y pixels that the features will be extracted. Also, it uses the "numSpatialX and Y" which is crucial in the orientation assignment. These parameters establish how many bins of the gradient histogram are going to be taken into account and then choose a region around those to remove the effects of scale and rotation. These bins and their closest regions then form another histogram. Other parameters are: number of words, which defines how many features are going to be extracted; C in SVM, which determines the error margin to be allowed during training; "size" of the cell within the window, which is crucial when and the number of train and test images.

2. Materials and methods

2.1. Dataset

In 2003, in the Institute of Technology in California, Fei-Fei et al. created and compiled the data ser of digital images called Caltech 101. Caltech 101 contains about 9146 images that are distributed between 101 categories such as faces, watches, ants, pianos, airplanes, motornbikes, etc. It also contain an additional background category. Within it, the images of oriented objects were mirrored to be left to right aligned. Each category contains about 40 to 800 images (the majority contains about 50). The size of each image is roughly $300x200$ pixels. The annotations were carefully clicked for each image and they are contained in the file "Annotations.tar". This file contains two pices of information: the general bounding box and a detailed outline specified by humans, enclosing the object (see Figure 2.). In their website, their recommend a popular number of train images of 1 to 30 and for test, from 20 to 30. These recommendations were taken into account when varying the parameters of the classifier. One of the advantages of this dataset is that almost all the images are uniform in image size and in the relative position of objects.Therefore, users usually don't have to resize, scale or crop images [2],[10]. Examples of this dataset are shown in Figure 1.

Figure 1. Example of Caltech101 images.

Figure 2. Example of Caltech101 images.

On the other hand, in 2009, in the Conference on Computer Vision and Pattern Recognition (CVPR), the ImageNet database was presented by researches of the Science department at Princeton University. This database was built based on the hierarchical structure provided by WordNet. In this sense, the categories are shown in relation to other categories, for example, within the WorldNet, the word "dog" would be nested under "canine" and this one under "mammal". See Figure 2. Therefore, ImageNet provides 12 subtress consisting of a total of 3.2 million image with annnotations (see Figure 4.) and spreaded over 5247 categories. There are about 600 images in each category. This database is commonly preffered since it was constructed with the goal that all the obbjects in the images should have variable positions, view points, poses, background etc [1].

Figure 3. Example of ImageNet images and their structure based on WorldNet [1].

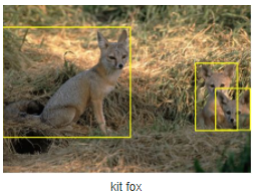

Figure 4. Example of ImageNet annotations [4].

These datasets were used in order to classify them and evaluate the PHOW algorithm trough ACA index when varying the following parameters: size of train set, size of test set, number of categories of images to classify, number of dictionary words and the parameter C in the SVM classifier. Then, when the combination of parameters that gave the best value ACA −calculated from de confusion matrix− was found, this combination was used to run the code in the other data set $(Imagenet)$. The code that was run in both cases is in the function $phow\_caltech101$ that was provided and can be found in [10]. This function downloads the database, it allows the change of the parameters mentioned above, then it train the vocabulary, compute spatial histograms, compute the feature map ontained, then train the classifier (SVM) and finally it test this classifier and evaluate, giving as result the confusion matrix. Using this code, we varied the parameters mentioned and calculated the ACA in each case in order to compare and select the parameters that improved the classification. Finally, when the combination of parameters that improved the performance (ACA index) of the Tiny datset was found, it was used in the ImageNet dataset. Once obtained this result, other trials were performed in order to find the best combination of parameters for ImageNet. The combination of chosen parameters was obtained experimentally by observing the ACA value obtained for each combination. All the results of performance were illustrated with the ACA index that is shown in the tables in the results section.

3. Results and discussion

3.1. Variation of parameters in Caltech101

In the first place, we tried to find the best combination of parameters that improved the algorithms performance.

To do so, the first parameter to evaluate was the number of train images. Values of 5, 15 and 40 were tested.

Table 1. Variation in the number of train images. As this was the first variation in parameters, the rest of them are the default parameters: test=15,categories=5,words=300 and C from SVM=10.

Caltech		
Variation of train set	**ACA**	**Time**
5	0,8551	3:11
15	0,8776	2:47
40	0,9339	3:09

As seen in *Table 1*, the best value of ACA was obtained with a train set of 40 images. This may be because, the more images the algorithm uses to learn, the more objects or patterns it will recognize when evaluating the test images for the first time. However, it is importat to mention that a value extremely high wont be good either because the algorithm will detect much noise intead of objects of interest.

Second, using $train = 40$, the number of images in the test set was varied.

Table 2. Variation in the number of test images.As the best ACA was obtained with train=40, this value was kept and the rest of them are the default parameters: categories = 5,words = 300 and C from SVM = 10.

Caltech		
Variation of test set	**ACA**	**Time**
5	0,96	2:51
15	0,88	2:47
40	0,89	3:01

As can be seen in Table 2. the best value of ACA was obtained when the test set was very small (5). However, we decided that it was best to use a larger set $(test = 15)$ since it contains more variety of images which can give a better reason to be to the results. Additionally, it is understood that when having more images to evaluate, it is more likely that the classification is incorrect in some cases. This is a price that is paid (tradeoff) in exchange for having more robust results (which can be obtained by evaluating more images).

Then, using $train = 40$ and $test = 15$, the parameter of number of categories to calssify was varied.

Table 3. Variation in the number of categories. Train=40 was kept because of Table1. and as the ACA was very similar varying test, test=15 was selected. The rest of them are the default parameters: words = 300 and C from SVM = 10.

Caltech		
Variation in number of categories	**ACA**	**Time**
5	0,97	5:00
30	0,72	6:33
65	0,66	6:58
101	0,61	8:10

As can be seen in Table 3. the highest value of ACA was obtained with the default parameter $categories = 30$. This may be because when the number of categories increase, objects are more likely to have characteristics or features to each one of those. Hence, the probability of belonging to the correct category decreases.

Then, using the parameters mentioned above, the number of dictionary words was varied.

Table 4. Variation in the number of number of words. Train=40 and test=15 were kept because of Table 1 and Table 2. The rest of them are the default parameters: C from SVM = 10.

Caltech		
Variation in number of words	**ACA**	**Time**
200	0,66	5:00
400	0,97	5:48
600	0,74	6:04

Here, the number of words influence it as well; if there is a large number of categories, few words won't be enough to have a good representation that can differentiate the images from each other. Likewise, having too many words over-representate the images, reducing the specificity of their features, making them more likely to share characteristics with other classes and hence be misclassified (see *table 3* and *table 4*).

Finally, the parameter C in the SVN classifier was varied.

Table 5. Variation in the parameter C of SVM. Train=40, test=15 and words=300 were kept because of Table 1 and Table 2 and Table 3.

Caltech		
Variation in parameter C-SVM	**ACA**	**Time**
5	0,72	8:24
10	0,73	5:00
20	0,71	4:50

In the Table 5. the highest value of ACA was also obtained with the default value. The parameter *C* in SVM's tells how much misclassification will be allowed while training. If the parameter is to low, the weights will have more significant changes during the training session because it tries to find the longest distance that separates the two classes at a cost of misclassification. In contrast, a large *C* can end in a bad classifier because it allows more misclassification in the training. In consequence, more images will be misclassified.

Lastly, observing the "Time" columns in all the tables, it was noted that the processing time is proportional to the number of categories, size of train set, size of test set and the number of dictionary words, because the more data is asked to process, the longer it takes too. Moreover, when the algorithm is run for the first time, the time can be longer than the second time if the parameters stay the same or differ a little. This happens because the second time, most of the data is already held in the RAM memory. So, the computer won't have to search it into the ROM memory, which happens the first time the code runs (see *table 1* and *table 2*). It was observed when running the codes because on the first run, the memory used was longer (ie. 516.8 MB when varying the train set with train=40) than later processes (ie. 385MB when varying test set).

In short, using *train/test* = 40%/15%, 30 categories, 400 words and $C = 10$ the **best** ACA obtained was: 0,73.

3.2. Variation of parameters in ImageNet200

Firstly, using the best combination of parameters of Caltech101 the ImageNet dataset was classified and the obtained result was:

ACA ImageNet = 0,1969

As it can be seen, this result is too low in comparison to the caltech101. In consequence, the parameters were tested experimentally again for this dataset. First, the train/test proportion of images was increased from 40/15 to 70/30 because, as the imageNet dataset contains twice the number of classes of the caltech101, more base images could bring more reliable results. Moreover, in previous work, as well as in related literature, we found that this proportion gave the best results. Hence, a number of words of 100,300 and 400 was run for this proportion of images, having the best result for 400 words with an ACA of $0,2378$. (See Table 6.)

Table 6. Variation in number of words using 70%/30% proportion.

ImageNet	
Variation in number of words using 70/30	**ACA**
100	0,175
300	0,215
400	0,238

Unlike Caltech101, in this dataset the number of word was proportional to the ACA index for all the testes values. This may be due to the fact that as the previous dataset is smaller, a very large number of words is not striclty neessary. Moreover, ir can cause overfitting. Also, the time spent varied: when using the proportion 4/15 the mean time was 38 minutes, when using this proportion (70/30) the minimum time was 50 minutes and the maximum was 70 minutes. This is understandable since a larger dataset and a larger amount of images in each set are being used.

Once obtained our best number of words, we proceeded to vary the parameter *C* of the SVM's classifier, giving it values of 5,10 and 20; being 20 the best one with an ACA of 0,2385 (see Table 7).

Table 7. Variation in parameter C, using 70/30 proportion and 400 words.

ImageNet	
Variation in parameter C SVM	**ACA**
5	0,2352
10	0,2378
20	0,2385

The transition between the caltech101 dataset and the imageNet dataset harshly decreased the performance (from 0,73 to 0,24). In Table 3 can be seen how increasing the number of categories reduce the performance of the algorithm. Therefore, it was expected that increasing the categories from 102 to 200 was going to reduce the performance. It is worth to mention, that this only happens while keeping the same parameters, as there is an optimum set of parameters for each number of classes.

Figure 6 shows the confusion matrix obtained for the best combination of parameters in Imagenet.

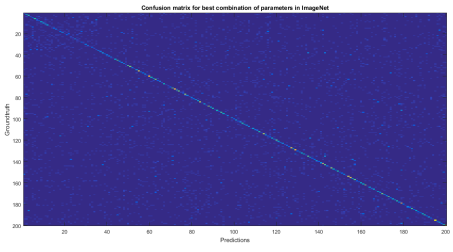

Figure 5. Confusion matrix for the best combination of parameters in ImageNet.

As the **best** ACA value obtained in Imagenet was of 0.2385 (see Table 7), most of the images were misclassified and very few categories had an outstanding classification out of the 200 available. According with Fgure 6, the best results, with $70\% \geq$ true positives for their own classes, in descending order were for: whippet, zebra, eggnog, police van and German short-haired pointer. Also, three categories had no true positives on their own, being binder, Labrador Retriever and malamute. These images, all have an speceific object that makes them part of an specific category. However, they also include background information that is shared between many other classes. Also, the orientation or degrees of rotation of the objects might make some classes susceptible to look alike others.

3.3. Further Work

Modest and strong boosting methods, that can improve the performance of the classifier by combining multiple classifying algorithms such as bagging and logistic regression. Moreover, experimentally, by testing several values for the number of words, an average standard value for each number of classes might provide a starting point for this parameter for further applications. Additionally, textons information can be used too, as objects in images like backgrounds behave similarly to textures.

4. Conclusions

PHOW algorithm uses features to represent characteristic of objects by using the dense sift algorithm. The process is not fast because of all the classes and the amount of words that the classifier needs. Also, the SVM's training requires a lot of time because is training a classifier for each category (1 vs all for all categories). However, once the features are extracted, the classification is fast a robust differentiation of the classes based on their features. The best combination of parameters obtained expertimentally was: train set = 40, test set = 15, categories = 30, number of words = 300, and C SVM = 10 giving an ACA of 0.73 for Caltech101. Nevertheless, the results were good only when the number of categories was low. Once the algorithm was implemented to the imagenet database with 200 classes, the performance dropped drastically (0.24). This meant that the algorithm requires a more complex or alternate processing.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[2] 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. *Computer vision and Image understanding*, 106(1):59–70, 2007.

[3] R. Gate. What is the difference between sift and dense sift ?, 2015.

[4] ImageNet. Download the object bounding boxes.

[5] T. Lindeberg. Scale invariant feature transform. *Scholarpedia*, 7(5):10491, 2012.

[6] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[8] V. org. Dense sift as a faster sift.

[9] V. org. Dense sift as a faster sift, 2018.

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