Object recognition by Pyramid histograms of visual words

Julio Nicolás Reyes Torres
Universidad de los Andes
jn.reyes10@uniandes.edu.co

Juan David Triana
Universidad de los Andes
jd.triana@uniandes.edu.co

Abstract

This article presents the evaluation of the method "Pyramid Histograms of Words" (PHOW) for object detection. The algorithm is implemented in 2 widely known databases: caltech 101 and Image Net, it seeks to analyze the effectiveness of detection for each of them. The " VLFeat " library is used and adjusted to implement the algorithm in the Matlab language. Certain parameters that change the operating conditions in the algorithm related to the form of program execution are also evaluated. Finally, the effectiveness of the method in the 2 databases is compared and the classes with the greatest difficulty in detection are analyzed.

1. Introduction

Different algorithms have been used for the detection of objects in Computer Vision, over time different methods have been developed that have greatly improved the detection of local and global characteristics in the images. The algorithm used in this article is called PHOW (Pyramid histograms of Words), it is an extension that improves the characteristics of the bag-of-words algorithm (BoW). The BoW algorithm is based on identifying the characteristics of the objects as words creating a "visual vocabulary", it is able to obtain the frequency of repetition of the characteristics, but it does not have the capacity to recognize the spatial information, that is, it identifies the characteristics of objects but can not say where they are. [2, 4, 1]

PHOW method solves the spatial problem of BoW, consists in dividing an image into sub-regions creating a pyramid with spacial characteristic of the image and therefore the characteristic histogram of the visual words is obtained locally in each sub-region.

PHOW sub-regions are formed by extracting the characteristics of the method called "SIFT" [6], created by David Lowe in 1999. It is a robust algorithm that extracts the distinctive characteristics of each image based in histogram-of-gradients (HoG) that identify the features which are invariant to image scaling and rotation, and invariant to change in illumination and 3D camera viewpoint, in addition they are well located in the domains of space and frequency and avoid the probability of disruption by occlusion, clutter, or noise. [5, 2]

2. Methods

2.1. Data description

In this practice the resources are supplied by "VLFeat.org" [7], which is an open source cross-platform that supplies different kind of vision algorithms in "Matlab", in our case the code for implementing the PHOW algorithm. Also, it is presented the information related with the 2 databases used to run the algorithm.

2.1.1 VLFeat Library

It is the library for Octave, Matlab and C, that supplies some segmentation, clustering and classification functions code. Specifically in our case, and example of implementing the PHOW algorithm to classify Caltech-101 images. The classification code implements:

Classifier:

- PHOW features (dense SIFT)
- Spatial histograms of visual words
- Chi2 SVM.

Speedup computation:

- VLFeat fast dense SIFT
- kd-trees
- Homogeneous kernel map.

2.1.2 Caltech 101 Database

It is an images database create by the Computational Vision Laboratory at Caltech. The set of images contains 101 categories. [3]

- 40 to 800 images per category
- Image size = 300 x 200 pixels

Caltech 101 sample.

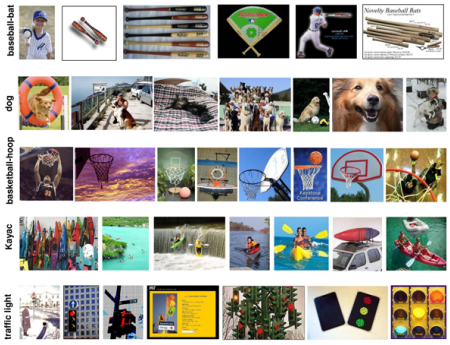


Figure 1: **Database sample:** In order Top-Down: baseball bat, dog, basketball Hoop, Kayac, traffic light.

2.1.3 ImageNet Database

It is an image database organized by hierarchy, each node of the hierarchy is depicted by hundreds and thousands of images. This database has an average of over five hundred images per node.

- More than 14 million images
- More than 20.000 categories.

Example of hierarchical organization for sports class:

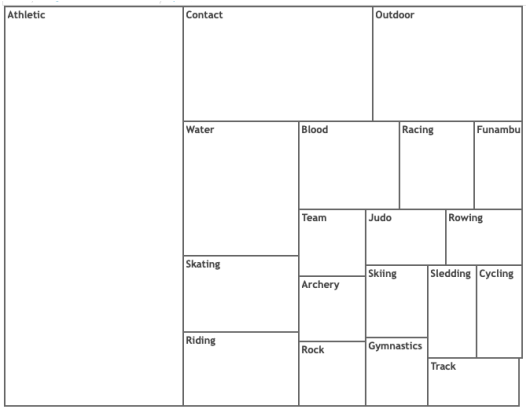


Figure 2: Tree map visualization for sports class.

2.2. Pyramid histograms of Words (PHOW)

As previously stated, PHOW method solves the the spatial problem of BoW (algorithm based on identifying characteristics of objects as words creating a "visual vocabulary"), where visual words are obtained locally by each sub-region which belong to a hierarchical tree. A short description about its algorithm is presented following:

1. PHOW sub-regions formed by partitioning the whole image hierarchically into a quad-tree of multilevel sub-blocks. At this point, the spatial information is defined by the hierarchical partition
2. Feature histograms are extracted from all different levels of the hierarchical sub-regions
3. Extract SIFT features from its surrounding area for each sub-region
4. K-means++ is implemented to cluster the feature points to obtain the final cluster centers of each sub-region as visual words
5. Finally, a visual vocabulary is composed of visual words.

It is mandatory to highlight the SIFT method implemented in many Computer Vision fields, because it is a robust algorithm that uses the called "histogram-of-gradients (HoG)" to find image features, it is partially or totally invariant with many common image deformations, as scaling and rotation, and changes in illumination. PHOW features uses the same components that SIFT but instead implements a structure with multiple scales building a hierarchical pyramid of sub-regions, in each one, SIFT features are computed and matched into a small codebook. [2, 1]

Although PHOW is implemented with a gradient histogram, it should not be confused with textons (basic unit of texture), because textons are built with a bank of lineal filters that represents many local patterns as edges, scales and orientations. Instead, PHOW calculates the histogram of gradient which takes a group of nearby pixels, and rather than describing them by their individual values, describes them in terms of their relationship to each other to find the image feature, that is, each gradient obtained in each pixel is weighted and represented in each bin. The most important hyperparameters for the PHOW strategy are:

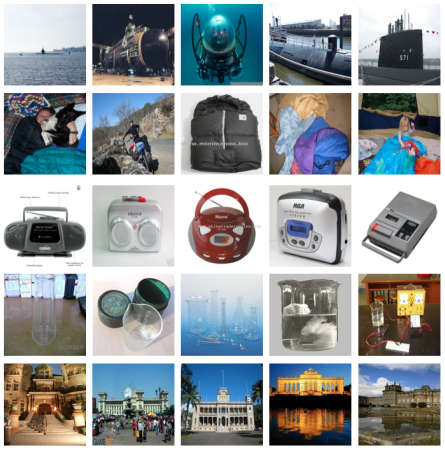
- **Window size:** Which is defined to attach the features depending of its size.
- **C:** It is an inherent parameter of SVM to adjust the error admitted of the support vector to the maximun distance from the discriminative hyperplane.

- **Step:** It is the jump distance of the window when extracting gradients of the image.
- **Number of words:** Number of centers in each cluster.

2.3. Database Challenge

Caltech 101 and ImageNet databases have different classes, some of them are typical kind of classes as animals (Gorillas, birds, zebras, dogs, deer, etc), flowers, Volcanoes, that in general represent normal things with known structures, colors and textures. But also, there are other classes that avoid those "normal" characteristics because, they can have a lot of different representations. In those classes, the objects may have extreme changes in the forms, colors, size, textures, points of view, in general a lot of possibilities. In the figure (3) is presented a set of "easy" and "hard" classes.

(a) Easy classes



(b) Difficult classes

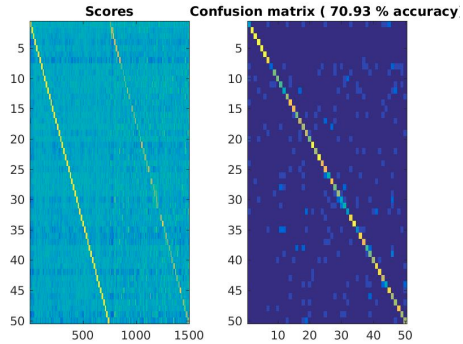
Figure 3: **Challenge in classes.** Image (a) shows a set of catalog easy classes, because the images represent objects with know structures, colors and textures. Instead, image (b) shows really very difficult classes, because the objects have different forms, colors, size, structure. A lot of possibilities in the detection.

3. Results

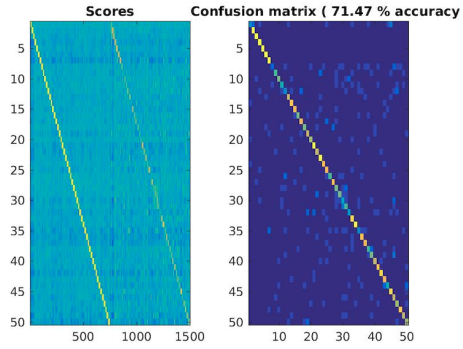
Note: Results are presented only with Caltech 101 database in the first phase of testing the hyperparameters effects.

Next, the different results are presented modifying the considered most important hyperparameters, to find the best method precision. The results are presented into a confusion matrix, the validation of its functionality is obtained by the accuracy of the matrix using always 50 different classes and 500 train images. This number of classes does not chain because is the parameter to reference the validation for each hyperparameter.

1) In figure (4) is shown the variation of the typical C parameter for the Support Vector Machine (SVM), the goal is to identify if a greater range of error acceptance, improves the precision in the classification.



(a) $C = 10$. Accuracy of $70,93\%$

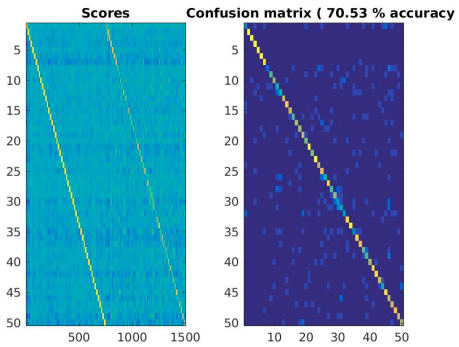


(b) $C = 30$. Accuracy of $71,47\%$

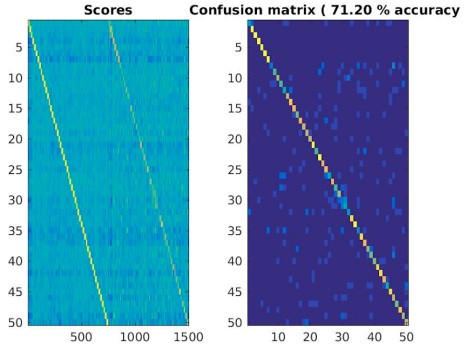
Figure 4: **Analysis of "C" parameter.** Image (a) (C=10). **Left**, Scores for train and test data. **Right**, represents the confusion matrix with its respectively percentage of accuracy. Image (b), the same analysis as image (a) but now using C=30.

2) Another important hyperparameter is the step in which the window will jump between the pixels, in figure (5) is

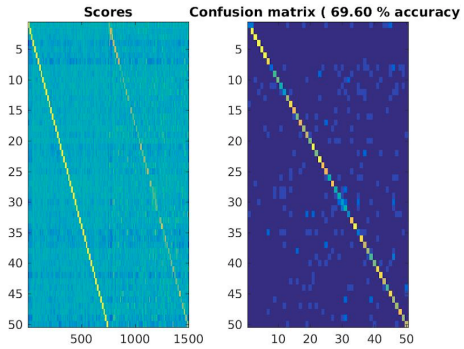
shown this variation. Step parameter changes to test the better value to move the window. The window size for this experiment was 2x4 pixels.



(a) $Step = 1\ pixel$. Accuracy of 70.53%



(b) $Step = 4\ pixels$. Accuracy of 71.20%



(c) $Step = 6\ pixels$. Accuracy of 69.60%

Figure 5: **Analysis of "Step" parameter.** Image (a) step = 1 pixel, Image (b) step = 4 pixels, Image (c) step = 6 pixels.

After enhancing the algorithm with the best hyperparameter values as it was shown before, we proceeded to apply an evaluation of the Image Net dataset. The following figure shows the accuracy and the confusion matrix:

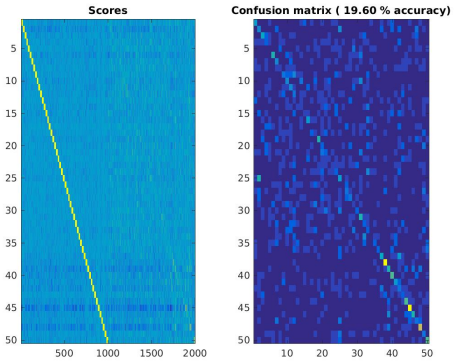


Figure 6: Recognition evaluation for the entire Image Net test set using the best optimization of the algorithm

4. Discussion

It was proved different cases where some variables were modified. The results presented before show how much change the accuracy with some specific variables values. The variation of different hyperparameters give us the accuracy which is the variable that indicates the best condition for Caltech 101 and ImageNet.

One of the most important parameters that shows a significantly variation of the accuracy is shown in figure (4). This corresponds only to the variation of C, fundamental parameter of "SVM" Classifier. Figure (4).(a) represents the result for the predefined value (C=10) and figure (4).(b), the best result after vary C among different values, it was C=30 with an accuracy equal to 71,47 %. The final result indicates that in general a greater value of C is better because admit more bad classify points and adjust them to correct them, but this parameter has a limit, with a very high value the SVM this is going to take a lot of points including the well classify ones.

Another hyperparameter which change noticeably was the "Step", some variations are shown is figure (5), In that, Step varied between 1, 4 and 6, with a window size of [2x4]. The best result is presented in figure (5).(b) where the accuracy was of 71,20 %. The main observation is the best case was for Step=4 with a window of size [2x4], that is, the best results are obtained when the step has the same value to the width of the window. This means that there is no overlap when screening the feature extraction window. Using Step = 6 decreased the accuracy significantly. This is due to information loss during the screening since it doesn't take the complete amount of pixels.

After performing the algorithm's setup to optimal and running it on the Image Net 200 test set, we acknowledged

a strong decrease on the main accuracy of the evaluation as it is shown on figure 6. The different images and its variations regarding this complex data-set seem to decrease profoundly the accuracy.

(a) Caltech 101

(b) Image Net 200

Figure 7: **Image analysis of different classes onto the datasets** Image (a) image of a rhino from caltech101 dataset, Image (b) image of a zebra from the Image Net dataset

Figure 7 demonstrates the variation that the Image Net dataset contains. More explicitly a mere part of a zebra is shown on the Image Net set, which can vary completely how the recognition algorithms recognizes an object as a zebra. Moreover, Caltech 101 images contain a slim amount of variation in its images, since all of them tend to have full size representations of its classes. Other images contain occlusions and rather ”harder” representations.

In general both Caltech 101 and ImagesNet presents a lot of classes, some one easier than others but always there are some specific images with a higher challenge. In the figure (3).(a) are shown some samples that represent easier images, on the left there are 4 images for each class considered as easy to identify, and on the right a sample in each class that represent a more difficult identification. For example, in elephant images, on the right show an elephant far away blocked by nature, and in another a crocodile behind a window while people are watching it. In figure (3).(b), are represented a set of the hardest classes to identify. The images in those classes have a lot of different representations, because the multiple shapes that an object can have. As is shown, different shape for a submarine, a music case, sleeping bags. All those images represents a semantic notion of objects, but they change widely that even for human is difficult to identify.

One of the biggest challenges of the algorithm is the feature extraction as a whole bin set. This is because there are images that can contain multiple objects of the same type which could be recognized as a separate class. There is no actual resize implementation which can be useful to detect not only a texture representation of the image but also a vast and more complete interpretation of the entire object. Moreover, the Gaussian kernel is used as a feature extractor on the algorithm.

Finally, the fine tuning of method could be obtained using the best results for each hyperparameter, to improve specific features and do an exhaustive explorations on details. Moreover the algorithm, we could include a dense scale variant feature transform (DSIFT), which implements a more broad representation per bin of images. This means that there is a more general representation rather than just local of it. This means that there is a trade-off regarding small objects that could better be represented more locally. So a good change that can be applied to the algorithm is using the normal SIFT parameters. This evidently increases time execution, but would improve the results in terms of accuracy. Another change would be the use of random forests instead of SVMs to train the algorithm. Such a varied dataset (Image Net 200) could be better expressed using decision trees and could improve timing of execution as well.

5. Conclusions

- One of the most important parameter is the “Step”, this parameter is related with the variation in the jump of the window between the pixels. According with the results presented in the image (5), it was demonstrated that the best result is obtained when the step has the same value to the width of the window.

- The classifier SVM (support vector machine) has an

important influence in the final accuracy, specifically the C parameter is essential to define how many errors associate with the bad classify points are accepted to adjust. It could observe that in general a greater value of C is better because admit more bad classify points to adjust and to correct, but this parameter has a limit, a very high value of C is not good in the SVM because this classifier is going to take a lot of points including the well classify ones.

- The databases used in this work have a lot of challenge with the classes, there are typically easy images to identify but also there are a lot of images that represent difficult classes and images, this is because an object may have a lot of shapes, structures, size, colors, textures, in general there is a high challenge for the algorithm classification (Figure (3), represents the easiest and hardest samples). So, along the development of this work it could be determined the hyperparameters and their better values that presents the better results in accuracy.

- The dense sift algorithm implements a less local representation of the images when extracting its features, in order to reduce time execution. Evidently it reduces the accuracy and effectiveness of the algorithm since it is extracting less information per bin of representation.

References

[1] E. F. Can and R. Manmatha. Histogram of Flow and Pyramid Histogram of Visual Words for Action Recognition. Technical report.

[2] A. J. Chavez. IMAGE CLASSIFICATION WITH DENSE SIFT SAMPLING: AN EXPLORATION OF OPTIMAL PARAMETERS. Technical report, 2005.

[3] Computational Vision at Caltech. Caltech101.

[4] H. Gao, W. Chen, and L. Dou. Image classification based on support vector machine and the fusion of complementary features. Technical report.

[5] S.-M. Khaligh-Razavi. What you need to know about the state-of-the-art computational models of object-vision: A tour through the models. Technical report.

[6] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. Technical report, 2004.

[7] VLFeat. VLFeat.