Object Recognition in Natural Images Using Multiscale Histograms of Visual Words

Jaime Enrique Cascante Vega
Universidad de los Andes
je.cascante10@uniandes.edu.co

Angela Castillo Aguirre
Universidad de los Andes
a.castillo13@uniandes.edu.co

Abstract

The classification of images according to the objects in it is a challenge that has been studied for a long time. Strategies such as Pyramid Histogram of Words (PHOW) and Scale Invariant Feature Transform (SIFT) have been implemented to determine the descriptors that will be extracted from the images. In this study, it was assessed the performance of the methodology to classify images, implemented by Caltech researches, in the Caltech-101 data set and in the ImageNet dataset. The results show that the performance of PHOW function in Caltech dataset is about 50% better that in the ImageNet dataset.

1. Introduction

One of the open challenges in computer vision is the classification of an image, according to the objects within it. As time passed, different strategies have been implemented and proposed in order to solve this problem in an automatic way. One outstanding used method to approach this, is the Pyramid Histogram of Words (PHOW). This method extracts the features of an image creating a representation of the image according to the objects place in the image.

The strategy of this method consists in the extraction of Scale Invariant Feature Transform (SIFT) descriptors, calculated as the points on a grid with spacing *M* pixels. On each point of the grid, the descriptors are computed in four different support patches, which lead to represent each point into four different SIFT descriptors. The variations of the scales will depend on the multiple descriptors taken from the image. After the extraction of the descriptors, the dense features (i.e. histogram representation of every single level) are classified using K-means clustering method to quantify the *V visual words* [1].
The SIFT and the PHOW methods are comparable in the sense that PHOW uses SIFT in order to be completed. So, the SIFT method is just a single part of the whole algorithm that PHOW uses. There is just one representation extraction grid that is used in SIFT, while PHOW is a complete pyramid which levels are composed by the SIFT method. Because of this, PHOW method follows the same characteristics of the SIFT method, so it is scale invariant because its performance depends on a scale invariant method.

The most relevant hyperparameters to take into account while using PHOW is the number of visual words to describe the object within the image, which is improved by the dense features. Also, the partition of grids that are in every level is a hyperparamater that affects the right performance of the method. Along with this, the amount of levels that the pyramid has, determines the quality of the descriptors that are going to be generated to have a good representation of the objects in the image. Another relevant parameters in the script was the number of the train and the test sets to assess the algorithm, because this is closely related to the amount of information that is going to be available so that the classifier would learn the multiclasses that there are in the database.

1.1. Datasets

1.1.1 Caltech101

The Caltech101 dataset is the first object recognition dataset. It have 101 classes and ost images have little or no clutter (background noise). The objects tend to be centered in each image. Most objects are presented in a stereotypical pose. Every class or categories have about 40 to 800 images, most 50 images. The size of each image is approximately 300×200 pixels. In figure below is shown the average image for each class, as seen most of the classes can be recognized even using the average over all of the class set (No huge variability indicator).

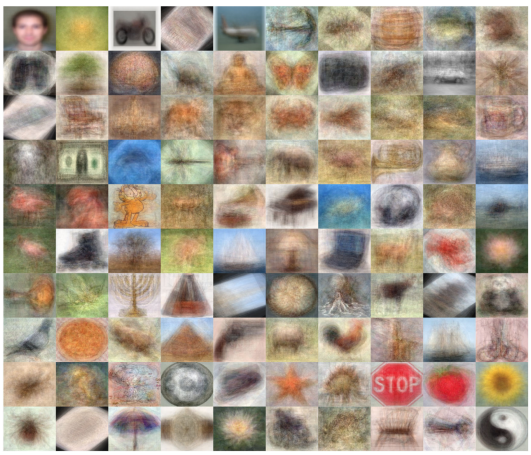

Figure 1: Average of the image for each category.

1.1.2 ImageNet

The ImageNet project is a large visual database designed for use in visual object recognition problems. Most of the images from ImageNet are from URLs and have been hand annotated to indicate the objects in the picture, roughly 14million images. ImageNet contains over 20 thousand ambiguous categories. And a typical category contains at least hundred of images. In the subset considered for this work only 200 classes are considered all images of size 256×256.

1.2. Introductory Questions

- How does the problem changes when the number of categories increases?

  As the number of categories increase the computation time increase a lot. Since, as will be discussed in next sections, one SVM is trained per class (One Vs all) and considering the additional computational cost of computing the PHOW features the time does increase exponential proportional to the number of categories. About the performance for example having initially K categories the chance of classifying an image correctly is $1/K$, hence as the number of categories K increase the chance decrease and does the performance (ACA).

- How does the problem changes when the size of the train set changes?

  Intuitively as the number of training images increase the robustness of the representation of the model increase too. More image means more representatively the dataset is of the problem hence the more representative is the model obtained. The performance is expected to grow as the number of image does. It also can be viewed from the statistical learning perspective, according to the Chernoff Bounds:

$$P\left[\frac{1}{m}\sum_{i=1}^{m} X_i - p \geq \epsilon\right] \leq \exp\left(-2\epsilon^2 m\right)$$

  Where p is the apriori-probability of each class, hence, as the number of training examples increase the confidence of the prediction also increase, hence more a accurately approximation of the empirical error is found and as the dataset is assumed to be more representative of the problem the performance (ACA) must increase.

- How does the result changes when the size of the test set changes?

  As there are more images are available for the test set, the results would be expected to get worse, if the number of images in the train set is kept constant. This is, the final result of the ACA should be less than when the sets are balanced or there are more images available for the train set. The above is due to having more images in the train set, the model will have more variability to classify an image in the test set. So, as the train set has fewer images, less information will be considered available for the model to be accurate.

- How does the result changes when the number of dictionary words changes?

  The number of words in the dictionary control directly the input space of the learning algorithm, hence as the number of dictionary words increase does the representation space (more words are considered for each images) and the performance must increase. However, increasing so much the number of words can make the models to overfit and one will have models with high bias, and small variance (Gauss-Markov Theorem).

- How does the result changes when SVM C parameter changes?

  As is discused in next section the parameter C control directly the penalization error and can be understood as the inverse of a regularization parameter, thus varying this constant to extremes lead to high variance/bias model. Small C (small penalization on classification errors) will results in models with high variance (underfit) but the optimal hyperplane will in

general have good margin, large C (high penalization on classification errors) will result in models with high bias (overfit), i.e the models will not care about the margin of the hyperplane but will care a lot on classification errors. Also, as the parameter C increase the computational time does increase, the optimization problem will be less convex, hence more time will take for solving each optimization problem.

- How does the problem changes when the spatial partitioning changes (conf.numSpatialX/Y)?

  The spatial partitioning variation affects the importance of the spatial information in the partitions. This means that as the partitions are increased within a same level in the pyramid, the relevance of the space occupied by the objects will change the description that exists in each representation given by the histogram of the partition. Basically the bag of words will have bigger object in scale of the dataset, however this parameters is in general so dependent of the scale one want to represent in the histogram of visual words.

2. Methodology

2.1. Feature Extraction: Pyramid histograms of visual words

Pyramid histograms of visual words (PHOW) features are a variant of dense SIFT descriptors, extracted at multiple scales [2].

2.2. Learning to discriminate

The class of hypothesis is given by linear decision boundaries or hyperplanes, but the product $\mathbf{w}^T\mathbf{x}$ is defined as the inner or dot product $\langle \mathbf{w}, \mathbf{x} \rangle$, so the hypothesis $h(\mathbf{x})$ is:

$$h(\mathbf{x}) = \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$$

The margin is the normalized distance of and instance from the hyperplane, so the optimal hyperplane can be constructed with the *worst* margins (the nearest instances to the hyperplane), additionally, as the data is assumed to be non-linear separable, looseness variables ζ_i is added to the examples that can not be classified correctly by the linear hypothesis. Thus, for an input instances $\mathbf{x_i}$ the hypothesis is given by:

$$y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) \geq 1 - \zeta_i, \quad \forall i \in [1, m] \tag{1}$$

This looseness variables are directly relating the *confidence* classification for an instance. If $\zeta_i = 0$ the instance $\mathbf{x_i}$ is classified correctly and with confidence, if $\zeta_i > 1$ the instance $\mathbf{x_i}$ is misclassified. So, for $0 < \zeta_i < 1$ the instance is classified correctly but without good margin. One might want to penalize the misclassified instances, but without compromising the margin, the margin by geometry is given by $M = 1/||\mathbf{w}||$ [3, 4]. This, can be write as optimization problem with restrictions:

$$\begin{aligned} \min \ & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\zeta_i \\ \text{subject to } & y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) - 1 + \zeta_i \geq 0 \\ & \zeta_i \geq 0 \end{aligned} \tag{2}$$

Where C can be understand as the penalization constant. This problem usually receive the name of soft margin hyperplane, as looseness variables allow error on the train set, making the optimal hyperplane *soft*.

2.3. Multicategory Classification: One vs All.

As multi-category classification is the goals of the present work. One vs all strategy is used in the training procedure for produce a combined resultant classifier that output the maximum output of each one of the classifier. Hence the label is given as:

$$y_i = \arg\max h_j(x_i) \quad \forall i \in \{1, \ldots, K\}$$

Where K is the number of classes.

2.4. Model Selection: Penalization of the error

Before talking about the model selection procedure I use I will introduce some basic ideas about statistical learning. The *VC* dimension of a model is defined as the maximum number of instances that the model can shatter in the given input space. For linear hypothesis it's known that the *VC*-dimension is equal to the dimensionality of the input space. Although if some assumptions are made over the hypothesis the *VC*-dimension is reduced [3]. Specifically it can be considered a class of linear models:

$$h(\mathbf{x}) = \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b), \quad |\mathbf{x}| < C_1, \quad |\mathbf{w}| < C_w \tag{3}$$

For some value of C_w. This set of linear hypothesis have bounded norm of the weights and of the input instances and the VC-dimension is reduced (can be less than the dimensionality of the input space) [3, 4]. This deduction is also a consequence of the optimization problem 2, for reasonable values of C the solution of the problem $\mathbf{w}^*$ will be bounded by the inverse of the margin of the support vectors. Described as this the SVM algorithm is directly controlling the VC-dimension thus it control the complexity of the model reducing the risk of committing overfitting. From a regularization view the constant C is like the inverse of a regularization constant, for large values

of C the problem will focus in minimizing $C\sum\zeta_i$ and the minimization objective of $||\mathbf{w}||^2$ will be ignores. This cause that the margin can be arbitrarily small increasing substantially the risk in committing overfitting, this can be interpreted in terms of the variance-bias trade off as C increase the model offer high bias, while increasing C the model offer high bias [5, 6].

One problem in using kernels is that the dimensionality of the instance grows boundlessly, for example for the Gaussian Kernel the dimension $\mathcal{H}$ have infinite dimension. Let the empirical error probability be defined as

$$\widehat{L_n}(h) = \frac{1}{m}\sum_{i=1}^{m} I\{h(x^{(i)}) \neq y^{(i)}\} \tag{4}$$

Where $h(x)$ are the linear hypothesis. More generally we can assume $\mathcal{C}$ as class of hypothesis $h(x) : \mathcal{R}^d \to \{-1, 1\}$, where d is the input dimension. The goal is to obtain the function $h \in \mathcal{C}$ with smallest error probability [7]. A well known result in statistical learning is that if the class of models $\mathcal{C}$ has finite VC-dimension then the error probability of the selected hypothesis is:

$$L_n(h^*) \leq \mathcal{O}\left(\sqrt{\frac{k\log m}{m}}\right) \tag{5}$$

Where k is the VC-dimension of the selected hypothesis $h \in \mathcal{C}$. The class of hypothesis defined as in equation 3 are linear classifiers with VC-dimension less than dimension of the input instances so it can be bounded as eq. 5. However, kernels are mapping our input space into a higher dimensional space $\mathcal{H}$ even infinite space, in the present problem as the *hand-crafted* features are assume to be representative of the problem *linear SVMs* are used for classification, hence the error is effectively bounded by the VC dimension that correspond to the number of words and to the number of training examples. The bound of the empirical error is hence given by:

- Caltech-101

$$L_n(h^*) \leq \mathcal{O}\left(\sqrt{\frac{101\log 30\cdot 101}{30\cdot 101}}\right) = 0.3406$$

- ImageNet

$$L_n(h^*) \leq \mathcal{O}\left(\sqrt{\frac{200\log 80\cdot 200}{80\cdot 200}}\right) = 0.2292$$

2.5. Model selection algorithm

For selecting the best model we do a grid search for $C \in \{1, 10, 100\}$ and number of words $NW \in \{600, 700, 800, 900, 1000\}$. For tunning these parameter we used on 80 images to train and 20 images to train in ImageNet and 30 images to train and 15 images to test in Caltech101.

Algorithm 1 Model Selection

for C in Grid of C **do**
  for NW = Search on number of words **do**
    Compute the PHOW features with NW number of words
    return X (Feature Vectors)
    for i = 1 to Number of Classes **do**
      Compute the optimal hyperplane of class i with cost C
      return model$_i$
    end for
    return models with NW number of words and C cost
    Obtain h_f from ▷ Train Classifier
    $h_f = \arg\max$ ACA
    if ACA$_f >$ ACA **then**
      return h_f
    end if
  end for
end for
  return Best set of models h_f

3. Results

3.1. Experiments on Caltech101

The result of the confusion matrix for the PHOW and SVM against the number of training images is shown in the figure 2.

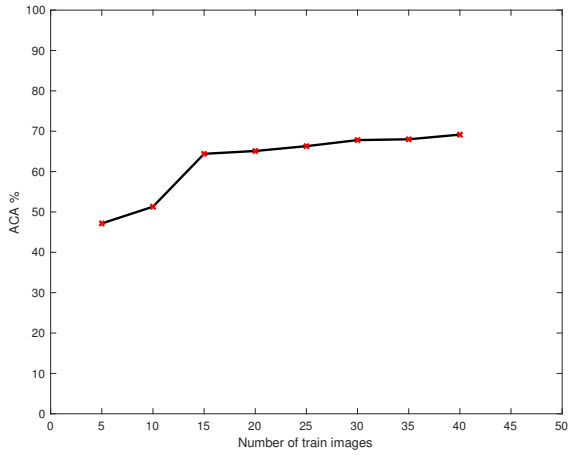

Figure 2: Performance of PHOW and SVM vs the number of training images.

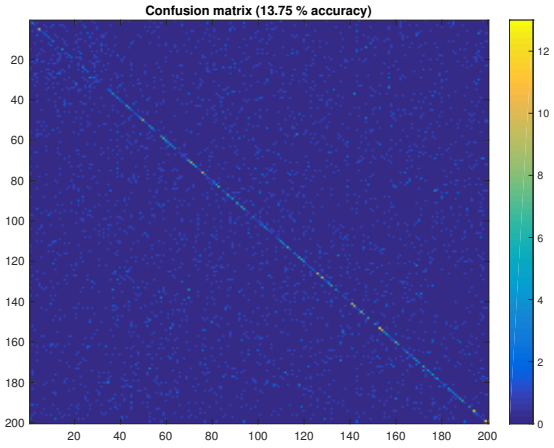

Figure 4: 600 number of words and $C = 100$

3.2. Experiments on ImageNet

The results of the performance in the variation in the parameter C are shown in the figures 3 and 4. As it is possible to see, the accuracy of the method gets worse in $0,33\%$ when the parameter C is increased in a 1000%. Besides, it was also varied the amount of visual words, and the result of the performance is shown in figure 5. Compared to the obtained performance in the increase of the C parameter, the increase of the *Number of Words* parameter shows a better performance of the method (Fig. 5).

3.2.1 Examples of trainval set

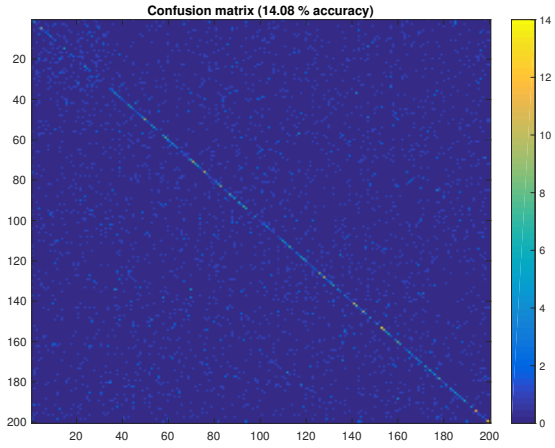

Figure 3: 600 number of words and $C = 10$

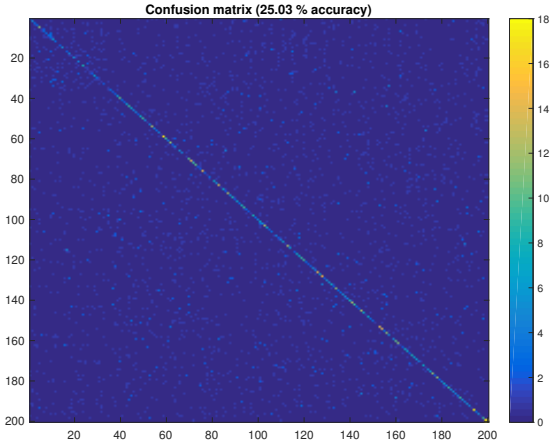

Figure 5: 1000 number of words and $C = 10$

3.2.2 Results of test set

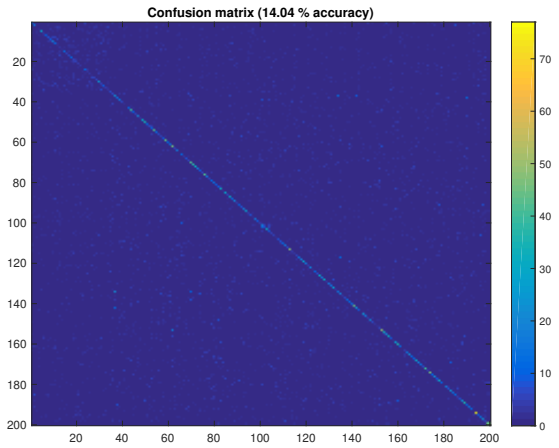

Figure 6: Best result on the test set.

4. Discussion

The best results obtained for each dataset is shown in table below:

	Caltech101	ImageNet
ACA	69.15%	19.03%

Table 1: Best results in dataset used. Best parameters of Caltech101: $C = 10$, $NW = 1000$ **RESULT on TRAIN-VAL**. Best parameters on ImageNet: $C = 100$, $NW = 1000$ **RESULT on TEST**

The chance of classifying correctly one image in Caltech101 is $1/101 = 0.0099$ while on ImageNet is $1/200 = 0.005$. Furthermore, the images of Caltech101 have limited variability, as mentioned before most of the images are centred have no noise in the background, have similar illumination and pose, however in ImageNet the variability of the image is huge, first the illumination is no constant on all dataset, the background is not constant, i.e cluttering on the background is present, there also more variability intra class. This means that for examples in the white wolf class, there's variability in scale, pose, setting in which the wolf is present.

The classes which have performance less than 2% were: *'Airedale', 'Italian_greyhound', 'barrow', 'chain_saw', 'reel', 'soft-coated_wheat...' 'titi', 'weasel'*. and less than 1% (as 100 images in test were tested it means that NO image were classified correctly) were *'Airedale', 'chain_saw', 'weasel'*. The images from these categories are specially images in which the colour is one of the best descriptors (Airedale are dogs with a striking colour, or chain saw have a lot of variability in colour in the images are colours from red, green, yellow, etc.)

In the images below are two samples of these two worst categories. As is seen in the *Chain Saw* category image personally I first recognize a person rather than a chain saw, similarly in the *weasel* category image there basically an image of teeth not weasels.

Figure 7: A sample image of the Chain Saw category.

Figure 8: A sample image of the Weasel category.

Similarly the best performance were obtained in the *web_site* class with an maximum accuracy of 77%. Looking a bit in this class one can see that the variability of the images is in general less, mostly all images are for white web page with some text. Hence, it make sense that the performance on this class is the best.

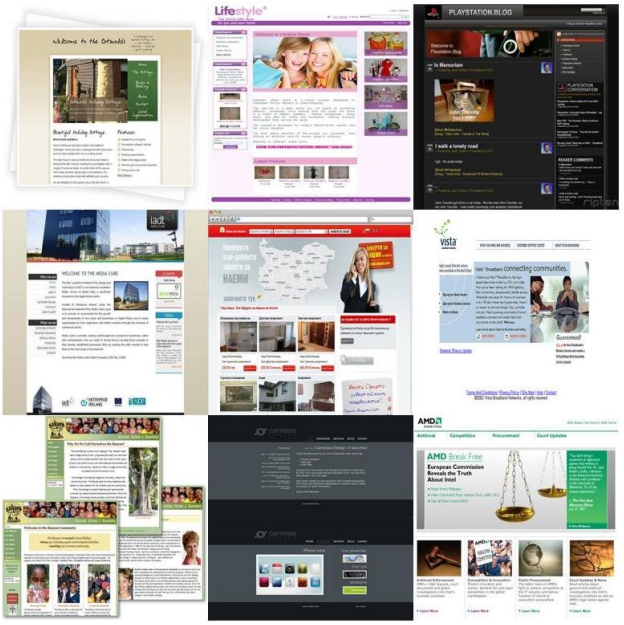

Figure 9: A sample image of the Weasel category.

A clear problem in using PHOW features for the recognition task in ImageNet is that the spatial partitioning as the variability of scales in ImageNet is huge this parameters will affect a lot the performance. Even pyramid representation is used for assess scale invariance in image in where object have small scale compared to the image size most of the visual words extracted will depend on the background that is mostly clutter in same variability as the images having then a bad performance.

5. Conclusion

In conclusion most of the difference of the Caltech101 and ImageNet dataset is the variability of the images. In Caltech101 basically the recognition task is a limited task as the dataset is of images with similar condition, contrary in ImageNet the variability of the images is so huge that even in some cases, as the one presented before, the category does not exactly correspond to the recognition made *naturally*. The best performance were naturally obtained in classes in where images have no significant variability. Hence in conclusion variability of the dataset in many contexts (scale, colour, illumination, resolution, pose, etc) is the most determining factor that affect the performance of recognition algorithms.

Additionally PHOW is mostly a descriptor of appearance, hence for increasing the performance of the algorithm we suggest to add shape information (HOG features) and in some categories the colour seems to be a good descriptor, and we think that adding the representation as histogram of an image or using color-PHOW might as proposed in [1] might increase the performance of the recognition algorithm.

References

[1] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," *2007 IEEE 11th International Conference on Computer Vision*, 2007.

[2] A. Bosch, A. Zisserman, and X. Muñoz, "Image classification using random forests and ferns," in *ICCV*, pp. 1–8, IEEE, 2007.

[3] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, pp. 273–297, 1995.

[4] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[5] M. Kearns, Y. Mansour, A. Y. Ng, and D. Ron, "An experimental and theoretical comparison of model selection methods," *Machine Learning*, vol. 27, pp. 7–50, Apr 1997.

[6] A. K. Jain, R. P. Duin, and J. Mao, "Statistical pattern recognition: A review," 1999.

[7] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, vol. 31 of *Applications of Mathematics*. Springer, corrected 2nd ed., 1997. missing.