Scene Segmentation and Interpretation: PASCAL Project

Savinien Bonheur and Albert Clerigues

***Abstract*— In this project we face a simplified version of the 2006 Pascal Challenge. We extended the given architecture to ease the testing of different strategies and parameter combinations. Two different approaches are shown in this report. The first one corresponds to the simple BoVW strategy suggested during the lectures, which soon revealed some serious limitations. A second more statistical approach aims to improve the previous issues through the use of Soft BoVW with fisher vector encoding.**

I. Architecture Overview

To ease with development we modified the provided architecture by making it modular and extended the caching and options system to be equally flexible. The modularity has been achieved by defining four basic blocks with a standard I/O, so that it will work regardless of the specifics of the algorithms used to implement each block:

1) Words: Responsible for extracting the low level feature representations of image patches.

2) Dictionary: dictionary, a.k.a. Bags of Words, creation, i.e. clustering, and operations involving dictionary specific operations.

3) Features: Takes care of building and preparing the descriptors for the classification phase.

4) Classification: final block in charge of training and using classifiers.

Additionally, the options system in this framework has been extended by adding new fields to the struct `VOCopts` that contain the options for each module of the framework. This provides an easy and convenient way of changing the global configuration.

Finally, the caching system was improved by adding parametric filenames depending on the specified options for each module. The system is incremental, meaning that the cached module filenames will also include the name and options of the previous modules.

II. Bag of Visual Words

A. Strategy overview

The first implementation was done according to a simple strategy that uses straightforward functions from the VLFeat library [5].

1) Words: DSIFT descriptors using `vl_dsift`.
2) Dictionary: Hierarchical K-Means, `vl_hikmeans`.
3) Features: Histogram of Words, `vl_hikmeanspush`.
4) Classifier: SVM, `svc` from PRTools v5.

Additionally, we added some other modifications to the strategy with the hope of increasing results. Firstly, we used the provided annotation system to extract the initial vocabulary only from the bounding box around the object we are interested in describing. This was done with the aim of increasing the vocabulary that describes each object, while keeping the space and time requirements low. The classification strategy should be changed accordingly, since the built dictionary doesn't consider background information. For that matter we added a classifier subwindowing feature to the framework, where one descriptor is obtained with the subset of words in each considered subwindow. Finally, all the different descriptors obtained from one image are classified and the biggest confidence level returned.

The implementation of this strategy helped us get comfortable with the given architecture and reinforced our understanding of the BOW strategies. Ultimately, by pushing the strategy to its maximum, the limitations of this approach were revealed.

B. Limitations

The classification results from this approach were progressively better as we increased the number of extracted SIFT descriptors and the number of Hierarchical K-Means clusters. However, they soon started converging towards the same results, shown in Figure 1, even as we further increased some of the parameters.

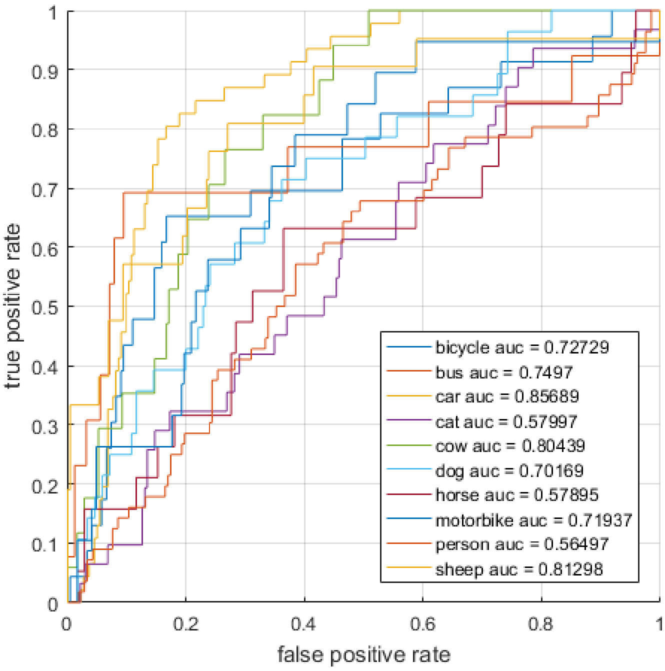

Fig. 1: Best results of the first approach, obtained with bounding box vocabulary, 60 HK-Means clusters and SVM.

Using this and other obtained results, we started drawing some conclusions about the fundamental problems of this approach. The main issue was disclosed when we looked at the classification performance for each class.

Class	**AUC**
Person	0.56
Horse	0.58
Cat	0.58
Dog	0.7
Motorbike	0.72
Bycicle	0.73
Bus	0.75
Cow	0.8
Sheep	0.81
Car	0.86

TABLE I: Sorted AUC results from first approach.

The trend drawn from Table I is that the best results were obtained for objects with low word intra-class variability. Broadly speaking, the visual words built for rigid objects, such as car class, will be found in other images much more consistently. On the other hand, non-rigid objects visual words are much less frequently repeated in the same proportions throughout the different samples.

The second proposed approach takes into consideration the found limitations and uses statistical encoding to provide more flexible descriptors that can capture the intra-class variability.

III. Soft Bag of Visual Words

A. Fisher vectors

The Fisher vector encoding represents a vector as a log likelihood gradient mean and variance from this vector to each word, represented as Gaussian distributions, from a dictionary built as a Gaussian Mixture Model [1].

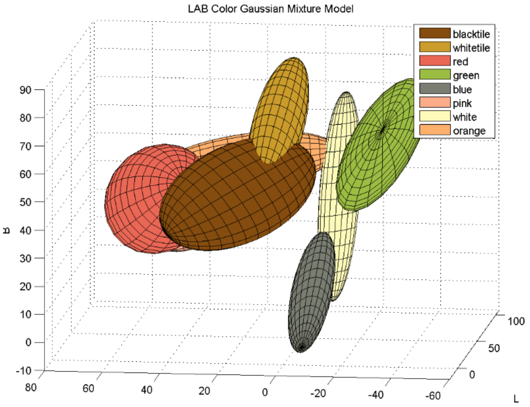

Fig. 2: Example of a 3 dimensional GMM.

When using a Gaussian Mixture Model for dictionary creation, the clustering is applied to the sift vectors extracted from a class of images and represent the words, or clusters, as high dimensional Gaussian distributions. The Fisher vectors are thus, for each dimension, the log likelihood gradient mean and variance between the descriptor vector and each GMM cluster. The obtained vector is weakly sparse.

B. Fisher Vector Properties

Compared with BoVW, Fisher vectors and their use of GMM create a bigger visual vocabulary, $2DN_c$ against DN_c for a K-Means made vocabulary, where D is the descriptors dimensionality and N_c the number of GMM clusters. However since the size of the dictionary is strongly linked with the computational cost, the Fisher vectors encoding is faster to execute [3] .

Although Fisher vectors mostly discard non-class related information [3], it is sensible to the area of the background compared with the object area, and thus, is not scale invariant [3]. This sensibility can be compensated by applying a L2-normalization [4] to the Fisher vectors.

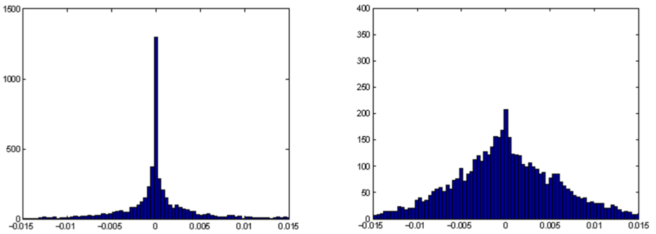

Fig. 3: Distribution of the values in the first dimension of the Fisher Vector obtained with 256 Gaussians (a) with no power normalization. (d) with a = 0.5 power normalization. Both histograms have been estimated on the 5,011 training images of the PASCAL VOC 2007 dataset.

It is also notable that Fisher encoding is weakly sparse, with up to half of non-zero values. It has been shown that applying power normalisation with a coefficient of 0.5,compensates this sparsity and so reduces the weight of the most often appearing features compared to the others. Finally, the projection of the features in the Fisher higher dimensional space allow the use of linear classifiers and hence reduce the computational weight of the binary classifier which label the pictures.

C. Final Strategy Overview

The blocks used for this approach are:

1) Words: DSIFT descriptors using `vl_dsift`.
2) Dictionary: Gaussian Mixture Model, `vl_gmm`.
3) Features: Fisher Vectors, `vl_fisher`.
4) Classifier: SVM, `svc` from PRTools v5.

In the final implementation we used dense sift to represent the initial vocabulary, since it is a robust descriptor and VLFeat provides a fast and accurate implementation of it.

We will then build the dictionary using the required GMM clustering, creating the words in a Soft Bag of Words fashion, to be able to use Fisher Vectors for decriptor computation. The GMM implementation used is also from the VLFeat library, which runs very fast thanks to CPU multi-core support.

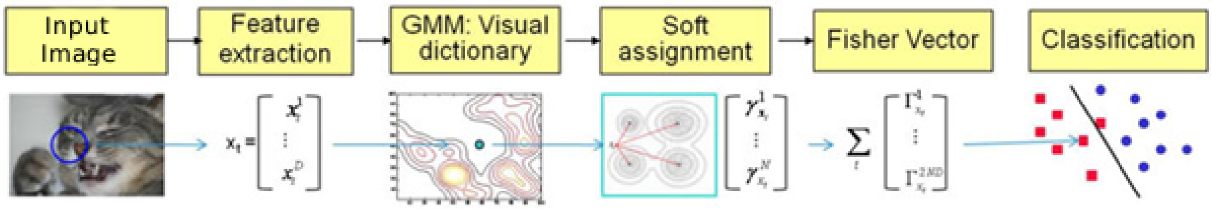

Fig. 4: Summarized pipeline of the final approach.

Finally, Fisher Vector encoding is used to project the dense sift descriptors through soft words assignment in a fast and efficient way. The obtained FV can be directly used as the final descriptor to train a classifier. VLFeat also provides an implementation of Fisher encoding, which has already built in the power normalization and the L2_normalization previously discussed. We do not consider the zero-order coefficients of the log likelihood gradient as it does not add significant discriminant information for classifying [2], but the main reason being the VLFeat implementation doesn't offer any option to compute them.

We then train a linear SVM classifier, whose performance is nearly optimal with the current configuration, as discussed in Section III-B. We empirically confirmed that a linear SVM already performs optimally by trying different polynomial and radial kernels with no significant improvements.

IV. Results

A. Final Parameters

The final results are obtained by using whole images to extract the initial vocabulary, given that the background irrelevant information is mostly discarded by Fisher encoding, where the dense sift descriptors have been extracted at step of 4 pixels, producing 330 MB of data for each class.

To compute the Fisher Vectors, a GMM dictionary is built with 128 clusters, which will produce descriptors of $2 \cdot 128 \cdot 128 = 32768$ elements. Then, the vocabulary extracted from each image is encoded into a Fisher Vector with the 'Improved' option specified which already provides power and L2 normalization.

Finally, the SVM classifier is trained with the obtained descriptors and the testing is performed accordingly.

B. Results Overview

In Figure 5 the resulting ROC curves and AUC, area under curve, from the testing dataset classification are shown. The results are significantly better than the first approach, with 19% improvement on the average AUC.

The rigid-body effect described in Section II-B is successfully reduced thanks to the use of FVs, but it can still be seen that classes representing more rigid objects are slightly more accurately classified. The overall results for all classes have proportionally improved with soft word assignment, except for the person class, which has a significantly lower ROC compared to the other classes. Although the new approach has solved most of the problems related with inter-class variability, the person class has significantly more than other already variable classes, i.e. cat. The ranges of clothes, poses and backgrounds for the person class are significantly more diverse than for other classes and require much smarter strategies to deal with it.

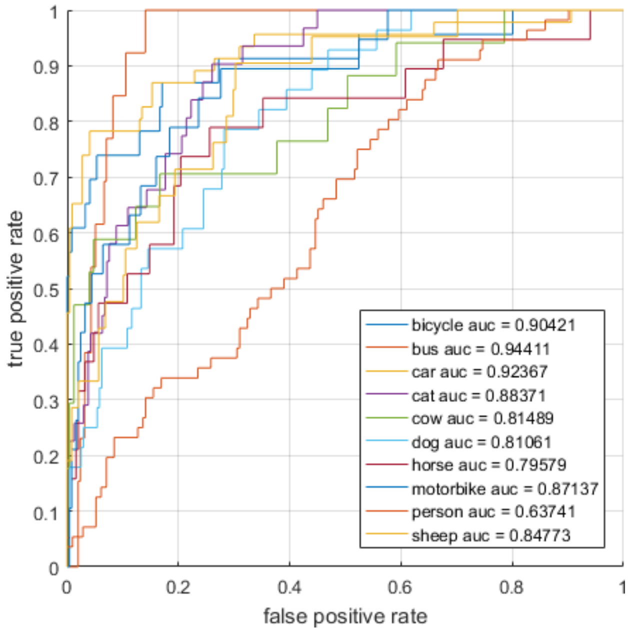

Fig. 5: Final results obtained with the SBoW approach.

Preliminary tests with bigger vocabulary and more GMM clusters show very little improvement, suggesting that this approach is reaching a saturation point like the first one. The capabilities of dense sift and fisher encoding to meaningfully summarize the information in each image are not perfect, and beyond a certain point even with a super dense vocabulary and thousands of GMM clusters we would tend to the same results.

V. Conclusions

In this project we have implemented and experimented with the Bag of Words strategy for object detection in images. Although BOW is now obsolete, with deep learning as the state of the art strategy, the same concepts and problems underlie in both for object detection. Any strategy needs a robust way to summarize the information contained in one image, such as intensities, colors, illumination, spatial relations, etc. in a compact and unique enough way so that the classifier can discriminate between the possible classes. This also applies to the more recent approaches, since information extraction, summarization and classification are implicitly done in the different layers of any deep learning architecture.

The observed results converging for both approaches hints that, in a pipeline, any stage can only perform up to a certain point, limited by the quality of the output of previous stages. If the information extracted from the images is of low quality we will obtain bad results regardless of the number of clusters or classifier used.

The architecture and strategy used finally is the same for all classes, being a balanced strategy that doesn't take advantage of specific objects properties consistent across images. For example, since the color of sheeps in the database is mostly white, we could have benefited from encoding the color information in the extracted sift descriptors. It would not be the case for person since they normally wear different colored clothing. This idea of optimising feature extraction for each class can be generalised for any specific algorithm or parameter to achieve optimal results. Future developments should optimise the pipeline for each class and, not very surprisingly, deep learning already features this behaviour implicitly.

References

[1] Jorge Sanchez, Florent Perronnin, Thomas Mensink, Jakob Verbeek. Image Classification with the Fisher Vector: Theory and Practice. International Journal of Computer Vision, Springer Verlag, 2013, 105 (3), pp.222-245.

[2] Z. Li,Lec 08 Feature Aggregation II: Fisher Vector, Super Vector and AKULA. Image Analysis & Retrv. Spring 2017

[3] G. Csurka and F. Perronnin, Fisher vectors: Beyond bag-of-visual-words image representations, in Computer Vision, Imaging and Computer Graphics. Theory and Applications, ser. Communications in Computer and Information Science, P. Richardand J. Braz, Eds. Springer Berlin, 2011, vol. 229, pp. 2842.

[4] Feng J, Ni B, Tian Q, Yan S (2011) Geometric lp-norm feature pooling for image classification. In: CVPRer Vision, Imaging and Computer Graphics

[5] A. Vedaldi and B. Fulkerson, VLFeat: An Open and Portable Library of Computer Vision Algorithms, 2008, http://www.vlfeat.org/