Computer Vision Lab 07 -PHOW

Juan Carlos Leon Alcazar
Universidad de los Andes
jc.leon@uniandes.edu.co

1. Description of the datasets

There are two main image databases used for this laboratory:

Caltech 101: The caltech 101 dataset contains a total of 9145 images distributed in 101 categories. Each category contains between 40 to 800 color images, most of them are catalog images a contain a single instance of a single object. Most of the images are pictures of the object and a few are hand draw. There is some variability in the resolution of images, but most of them have a resolution of around 250x300 [3].

ImageNet: The subset of the image net contains 996 object categories each with 100 instances for a total of 199200 images. All images are in format JPEG contain color information, the image size is standard on the dataset at 250x250. Unlike caltech 101 The categories from imagenet are created according to the WordNet hierarchy.[2]

2. Recognition Method

The base recognition method for the laboratory is the "pyramid histogram of visual words" also know as PHOW[1]. In this method, images are represented by means of an spatial pyramid as proposed proposed by Lazebnik[4]. Such pyramid is built upon a regular grid of N partitions, where N increases(decreases) as the level of the pyramyd increases (decreases). This representation is thus an orderless collection of multi scale feature histograms for a single image.

The features calculated at each region of the grid are the SIFT[5] descriptors proposed by lowe. These features are scale and rotation invariant descriptors created from the gradient information in region of an image. The multi scale features are used to build a codebook or dictionary by using a clustering method (usually Means or Expectation Maximization). This codebook is then used to quantize the descriptor vector for every other image in the train set. Finally, the whole set of quantized vectors is used to train a classifier SVM,

For the classification step features are calculated and quantized on the test set under the same scheme described for the train set, Finally the trained SVM is used for classification.

2.1. Changes over the base recognition method

The code base for this laboratory is provided in http://www.vlfeat.org/applications/caltech-101-code.html As a matlab scripts which uses the VLFeat Library[6] to implement the classification process described above.

Before directly using the classifier in the imagenet database, we first try to optimize the hyperparameters in the smaller (and easier caltech) 101 set. The main goal of this procedure is to avoid the hyper parameter search in the imagenet dataset as there is not enough time of complete such task at that scale.

In the base PHOW code we identify 7 Hyperparameters for different stages of the problem [1]

- Number of words for dictionary (any integer number larger than 0)
- Spatial partition in X (any combination of N - $N > 0$ - integers larger than 1)
- Spatial partition in Y (any combination of N - $N > 0$ - integers larger than 1)
- Svm bias multiplier (Any real number)
- Clustering algorithm (Three different approximations to the EM algorithm are available)
- Max number of iterations for the clustering algorithm (Positive Integer)

[1] The upper limits for the number of iterations in the clustering process and the depth of the KDTree and the image resize operation are also parameters but are considered of far less importance than the mentioned above

- Type of kernel (Chi square, intersection kernel, Jensen-Shannon kernel)
- SVM C (Real number)
- SVM solver (sdca, sgd and linear are available)
- Svm Max Number of Iterations (positive integer)
- Epsilon (goodness of fit) of svm. (Real number)

The initial number of possible hyper parameters combinations is not even bounded. To better approach this problem (given the time and computing power limitations), we approach the hyper parameter search by optimizing a single hyper parameter at the time, find the best possible accuracy for it, and then proceed to optimize a second hyperparameter. The full set of hyper parameter is not used, as we select a subset of those who could produce larger improvements in the classification process:

This is the order for the parameter optimization, we briefly describe the reason to select the parameter:

1. Number of words (Changes the number of visual words in the codebook, which are then used to represent every word in the training and test set. This parameter essentially tunes the initial data representation)
2. Spatial partitions (Like the number of words, this parameters changes size of the visual words in the codebook,Again this parameter tunes the initial data representation)
3. SVM C (Controls the loss function of the svm, and thus the global optimization process, by assigning a penalty to the misclassified elements)
4. Kernel (specifies a N-dimensional space where the linear separation of the data is performed, a space where the data is approximately linearly separable will bring the best results)
5. Solver (The algorithm used to linearly separate the data of the SVM)
6. Clustering algorithm (create the codebook from the initial representation of the training images, like the number of words has a great influence in the initial data representation)

The initial parameter tuning was performed with a subset of from caltech (30 images - 15 test, 15 train-) per class with 30 classes.

The best set of hyper parameters is:

- Number of words: 1000
- Spatial partitions: [2,3,4]
- SVM C: 50
- Kernel: Intersection Kernel
- Solver: SDCA
- Clustering algorithm: ANN

Sampling	Accuracy
996 Categories, 10 images per class	12.80
100 Categories, 100 images per class	19.84
310 Categories, 31 images per class	14.86

Table 1. Accuracy over the full image net dataset

The performance of this set of parameters over the subset of 30 classes is 80.67%, The performance over the whole set of 100 classes and 9145 Images degrades significantly to is 68%. This figure is only slightly better than the initial configuration at 67.50%.

3. Results on Imagenet

Producing a result for the whole dataset is a challenging task. It is not possible to train with the whole 99600 set of training images, it was empirically established that the RAM memory on the server (100GB) can handle the whole classification process with up to 10000 images. This means that either a subset of the images must be selected or a subset of the categories must be selected, we use both approaches

- Select all 996 categories and 10 images for each class (10000 images total)
- Select only 100 categories (random) and use all 100 images for each class (10000 images total)
- A third 'hybrid' strategy with 310 categories and 31 images for each class (9610 images total)

Results are summarized in table 1

3.1. Sensitivity to Parameters

The initial configuration of hyper parameters (see section 2.1) is used as a base configuration to analyse the sensitivity of the method to changes in a subset of hyperparameters in the imagenet dataset, The set of hyper parameters explored is:

Number of Categories set of the parameters to explore: 50,100,200,500,1000

Size of training set set of the parameters to explore: 10,20,50,100

Classes	Accuracy
10	44
50	21.20
100	13
200	13
500	10.08

Table 2. Accuracy according to the number of classes used for training and test. No test with the complete dataset are performed as the kaget test (500) classes already consumes over 100GB of RAM memory.

Number of Images	Accuracy
5	25.10
10	36.00
20	45.00
50	52.20
100	56.00

Table 3. Accuracy according to the number of images per class used for training and test.

Spatial Partition	Accuracy
2 4	25.10
2 3 4	36.00
2 3 4 5	45.00
2 4 8	52.20
2 4 8 16	56.00

Table 4. Accuracy according to the spatial partition (same partition sued for both axes) per image used for training and test.

Number of spatial partitions set of the parameters to explore: [2 4] [2 3 4] [2 3 4 5] [2 4 8] [2 4 16] .

results are shown in tables 4, 3 and 2.

4. Results Analysis & Limitations

While the method can reach an accuracy of 64% on the caltech 101 databse, these results clearly can not be reproduced on the much larger and much complex imagenet dataset, the overall classification accuracy is allways under 20% for the larger possible experiments that can be executed with the available hardware. Nevertheless this figures are significantly better than a random guess over the dataset (which would produce about 0.01% accuracy) this suggest that the method could be a good starting point for the recognition of images in large scale datasets.

Regarding the parameter exploration. It can be stated that the method is insensitive to the spatial partitions as the initially suggested pyramyd of 2 and 4 partitions has almost the same accuracy of deeper pyramids. This suggest that the most relevant information can be captured at these first two scales, additional data of other scales is likely noisy data discarded by either the classifier or the clustering process. This is propably related with the object size in the images, which is ussualy about half or a quarter of the image size.

It is clear that the classification improves the larger the training set is. This could be attributed to the larger amount of different samples from each class which contributes to build a better model of the class.

It is also interesting to notice that the smaller the amount of classes in the classification process the better the results. This suggest that either there is not enough capacity in the SVM to capture the complete variability of a large number of classes (996), or the base descriptor (SIFT) is not capable of capturing the relevant information when the classification problem grows large.

Finally it is also worth noticing that the procedure requires a very large amount of computer resources, the subset used for the laboratory is an order of magnitude smaller than the complete imagenet dataset. Applying this method to the full dataset will probably require require a very large computer or cluster with Terabytes of RAM memory available. Even with today's hardware such requirements are not easy to meet.

4.1. Improvements

A first step to improve the results would be to perform a proper search over the full set of identified hyperparameters, as it is likely that the proposed approach (while time efficient) could lead to a local minima.

Even a proper parameter search would not address the 2 most clear limitations found for this approach: Model capacity Base descriptors limitations.

To address the first issue it would be necessary to change the kernel for the SVM, this process is, very difficult as there is not an straightforward method to optimize the representation space, this process is rather (to the authors best knowledge) a trial and error approach. Given the large size of the problem optimizing the kernel could be a very long task, whose results are not granted. Probably the best improvement for this issue would be to change the classifier for one that can better handle the large variability in the dataset, it is possible that a very large random forest could perform better in this scenario.

Regarding the second limitation, the SIFT descriptors works at the local level, thus, ignoring regional and global patterns in the image that might be critical for the process of classification. A better approach might also use regional information (like the deformable part models[**?**]) or global information obtained from second order statistics of the image.

Finally an strategy that could address both the descriptor and classification effectiveness would be a deep neural network, which explicitly builds a better model representation over the whole image dataset.

References

[1] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[2] 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.

[3] 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.

[4] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[5] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[6] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1469–1472. ACM, 2010.