Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories

Ana Maria Martinez Sidera, Department of Computer Science IIT

***Abstract*—In this paper the authors present a method for recognizing scene categories based on approximate global geometric correspondence. The main algorithm for this is made a partition in the image, decreasing in each step the region. After each reduction, compute the histogram of local features of each region in the image. They called all this sub-region compute as a ”spatial pyramid”. This is just a extension of a bag-feature image representation. I'm going to use this method to check the performance between using different classifiers to distinguish between different objects by modifying the number of these to also see how the algorithm behaves with different numbers of objects. In addition I will vary the number of images that I enter to the system to find the fluctuation of the algorithm.**

I. Introduction

In this paper, I am going to consider the problem of recognizing the semantic category of an image in order to improve the process of image classification. For example, if we want to classify between office or beach. In an image we can find telephones, tables, computers... objects characteristic of an office and in the other water or sand. If we tried to categorize the two whole images we could not come up with an interesting solution or performance. But if we reduce the image depending on the characteristics of each one, we can get to understand which objects are in each type of image and better the performance of our classifier.

The spatial pyramid goes one step and takes into account all the characteristics from different levels. If we were at level 0 of our image you can find more detailed images of a beach. But in the case of an office we need to capture the characteristics of a level 2 in order to have a better performance in our final classifier. This algorithm has been used on numerous occasions, have recently demonstrated impressive levels of performance.

It should be possible to develop image representations that use low-level features to directly infer high-level semantic information about the scene without going through the intermediate step. This philosophy is inspired by the evidence that people can recognize scenes by considering them as a total manner, while overlooking most of the details of the constituent objects.

For represented features based on aggregating statistics of local features over fixed sub-regions, it use a kernel-based recognition method that works by computing rough geometric correspondence on a global scale using the pyramid matching scheme. This algorithm involves repeatedly subdividing the image and computing histograms of local features at increasingly fine resolutions with each sublevel.

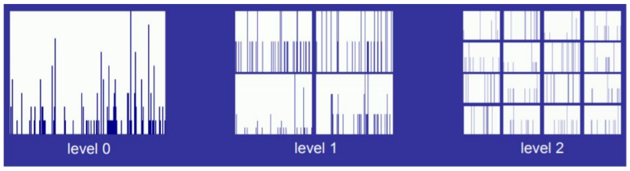

Fig. 1. A schematic illustration of the spatial pyramid representation.

The rest of this chapter is organized as follows. In Section II, I am going to review pyramid matching. In Section III, I am going to describe the feature extraction accomplished in this project. Section IV presents my original experimental results. Finally, in Section IV I will talk about the conclusion of my project.

II. Spatial Pyramid Matching

Pyramid matching works by placing a sequence of increasingly coarser grids over the feature space and taking a weighted sum of the number of matches that occur at each level of resolution. At any fixed resolution, two points are said to match if they fall into the same cell of the grid; matches found at finer

Manuscript received November 20, 2017.

resolutions are weighted more than matches found at coarser resolutions.

Grauman and Darrell present the pyramid match kernela new kernel function over unordered feature sets that allows them to be used effectively and efficiently in kernel-based learning methods. Each feature set is mapped to a multi-resolution histogram that preserves the individual features distinctness at the finest level. The histogram pyramids are then compared using a weighted histogram intersection computation, which they show defines an implicit correspondence based on the finest resolution histogram cell where a matched pair first appears.

In the following image you can see how to make a pyramid match kernel with a split image. In each region it is associated to a characteristic depending on the number that appear in each region.

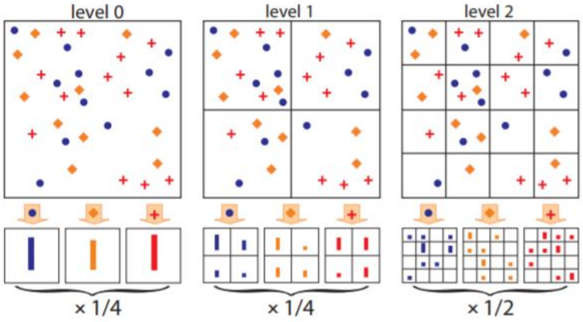

Fig. 2. Example of constructing a pyramid for L = 2.

This can change the results, and allows us to appreciate in greater detail the characteristics and how they appear throughout the image. In this way we can capture all the essence of the image with the different levels. To perform the pyramid matching we will first have to calculate the histogram of the level 0 of each one of the images that we have.

Then we will divide the images into four squares, in half in each shape. Returning to perform the histogram of each of the parties. At no time do we forget the first histogram done and it will be part of our final algorithm that we will introduce in our classifier.

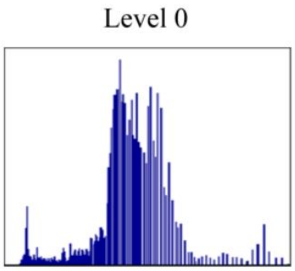

Fig. 3. Example of constructing a histogram for L = 0.

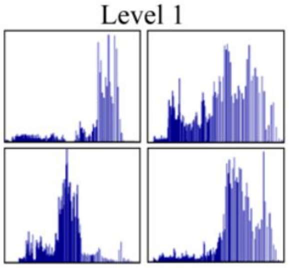

Fig. 4. Example of constructing a histogram for L = 1.

As we did in the previous step, in this we are going to divide each region of level 1 in half, obtaining in this case level 2. In each region we have achieved, we will perform the histogram and put it into our final algorithm.

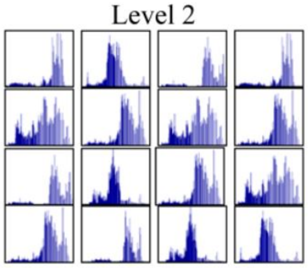

Fig. 5. Example of constructing a histogram for L = 2.

Having at the end in each of the parts each of the histograms that have been made to improve the performance of our classifier. With this we will not only be able to recognize small objects in a small room, but we will also be able to appreciate large objects such as beaches or cliffs in an image of nature.

III. Feature Extraction

In the paper they use two types of feature extraction but I'm just going to describe the one I have used SIFT extractor.

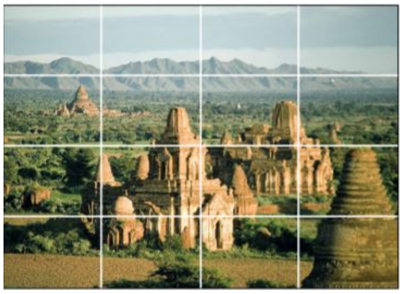

Fig. 6. Spatial pyramid histogram

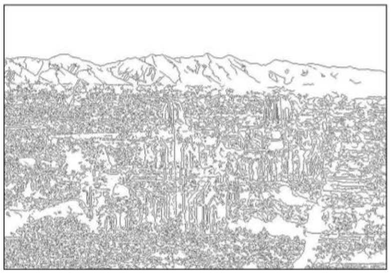

Fig. 7. Weak features

In 2004, D.Lowe, University of British Columbia, came up with a new algorithm, Scale Invariant Feature Transform (SIFT) in his paper, Distinctive Image Features from Scale-Invariant Keypoints, which extract keypoints and compute its descriptors. There are mainly four steps involved in SIFT algorithm. From the image above, it is obvious that we can't use the same window to detect keypoints with different scale. It is OK with small corner. But to detect larger corners we need larger windows. For this, scale-space filtering is used. In it, Laplacian of Gaussian is found for the image with various values. LoG acts as a blob detector which detects blobs in various sizes due to change in . In short, acts as a scaling parameter. For eg, in the above image, Gaussian kernel with low gives high value for small corner while Gaussian kernel with high fits well for larger corner. So, we can find the local maxima across the scale and space which gives us a list of (x,y,σ) values which means there is a potential key-point at (x,y) at scale.

But this LoG is a little costly, so SIFT algorithm uses Difference of Gaussian which is an approximation of LoG. Difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different , let it be and k. This process is done for different octaves of the image in Gaussian Pyramid.

The "weak features" is one of the extractors of characteristics that the authors used in their work and they comment that they have very small spatial support (to single pixel) and take on just a few possible discrete values. Edge points at 2 scales and 8 orientations (vocabulary size 16).

The strong features that are computed over larger image patches and quantized using a large vocabulary to capture more distinctive and complex patterns of local appearance, described above. It is represented in below image: SIFT descriptors of 16x16 patches sampled on a regular grid, quantized to form visual vocabulary (size 200, 400)

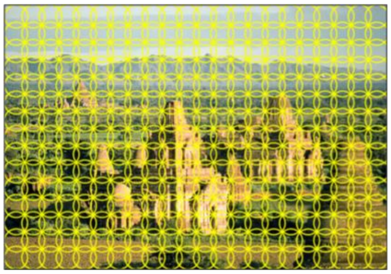

Fig. 8. Strong features: SIFT

As we can see in the two upper images, the amount of characteristics that the SIFT algorithm can reach is much greater than developing it with weak features. That is why I have not used it in my project, but it could be used to see how performance fluctuates depending on the characteristics that we give to our classifiers.

IV. Experiments

Having the algorithm explained previously and implemented in Python I checked with different classifiers the results obtained. The classifiers that can be used are: Linear Regression, Polynomial regression, Radial basis function, Logistic Regression and KNN.

Linear Regression: is a linear approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.

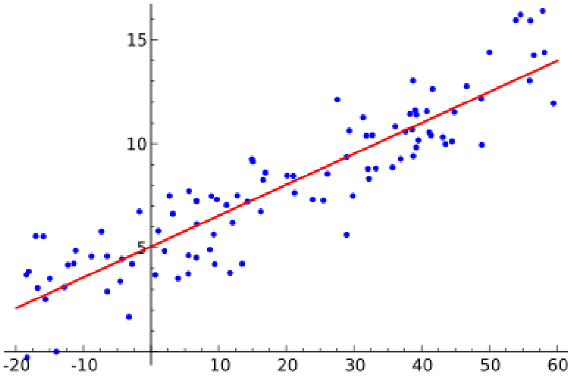

Fig. 9. Linear Regression

Polynomial regression is a form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modelled as an nth degree polynomial in x. The predictors resulting from the polynomial expansion of the ”baseline” predictors are known as interaction features.

A **radial basis function** (RBF) is a real-valued function whose value depends only on the distance from the origin or alternatively on the distance from some other point c, called a center. The norm is usually Euclidean distance, although other distance functions are also possible.

Logistic Regression is a regression model where the dependent variable (DV) is categorical. The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It allows one to say that the presence of a risk factor increases the odds of a given outcome by a specific factor.

The **k-nearest neighbors algorithm** (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space.

Another of the options that I have been modifying to see how the algorithm acted in different situations are the quantities of objects to be identified. The minimum is two objects to be classified and the maximum five objects that we can differentiate. They could be implemented more but I have decided not to overload the code due to the limitations of my computer.

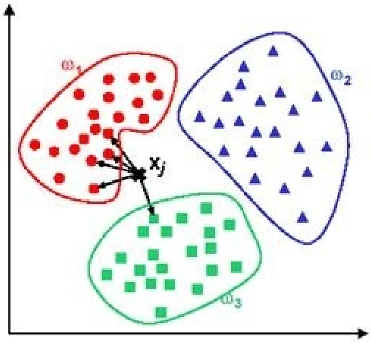

Fig. 10. KNN

Fig. 11. Airplane

Fig. 12. Motorbike

A. *Scene Category Recognition*

For this work I have used a dataset of images provided by the Stanford University about faces, airplanes and motorbikes.

V. How to use the program

To use the program you need to introduce two parameters to it. First of all we need to know the level with which you want to feed the classifiers. This level can fluctuate between 0 - 3 (included) and if none is entered by default between level 2. At that moment it begins to process all the characteristics of the images and their levels and saves them in some variables with which we feed the sorter. This classifier is the second parameter that we have to introduce into our program.

Acknowledgment

The author would like to thank my professor for the patient guidance, encouragement and advice he has provided throughout my time as his student. I would also like to thank the TA of the subject for helping me when I needed it and has guided me in the development of this work.

References

[1] 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, vol. 2. IEEE, 2006, pp. 21692178.

[2] Y. Rubner, C. Tomasi, and L. J. Guibas, The earth movers distance as a metric for image retrieval, International Journal of Computer Vision, vol. 40, no. 2, pp. 99121, 2000.

[3] P. Li, J. Ma, and S. Gao, Actions in still web images: Visualization, detec- tion and retrieval, in Web-Age Information Management. Springer, 2011, pp. 302313.

[4] http://vision.stanford.edu/Datasets/40actions.html

[5] https://github.com/CyrusChiu/Image-recognition

Ana Maria Martinez Sidera
Software and Electronic Engineering.

Double Master's degree which starts with one year at Polytechnic University of Madrid, and another year on IIT, majoring Computer Science.