Billè Matteo [Mat. IN2000125] Ingegneria Informatica-Applicazioni Informatiche

IMPLEMENTION OF A CLASSIFICATOR BASED ON BAG-OF-WORDS APPROACH

Project 2

Description of project

The project purpose is to solve the classification problem of 15 class of images. We use the bag-of-words approach with the analysis of feature histograms for each image.

The classifier is developed on Matlab, with the use of self-made procedures and other functions already existing (i.e. kmeans, knnsearch, svg, …)

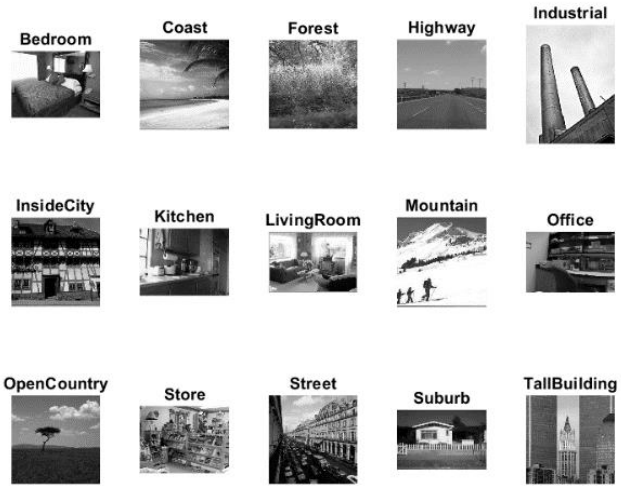

Figure 1: Sample for each class.

We use the dataset (from [Lazebnik et al., 2006]) that contains 15 categories and a total of 4485 images subdivided in Training set and Test set. The training set contains 1500 figures, 100 for each class; instead, the Test set contains 2985 images not equally distributed in the classes. Example of images in Figure 1.

Training

In the training phase all the 1500 images are resized to 256x256 pixels and subsequently 128-values features are extracted with Speed-up Robust Feature (SURF) descriptor algorithm from the points detected by SURF detector. We choose SURF because it is faster and it has similar result compared to Scale Invariant Feature Transform (SIFT). We use SURF point detector and SURF descriptor algorithms that are already developed in Matlab. We use `detectSURFfeatures` to extract SURF point and we notice that with default value of attribute `MetricThreshold` (set to 1000), that selects only SURF point with a metric stronger than defined threshold, some images in the category 'Coast' have no features, probably because they are very flat. Therefore, the value of `MetricThreshold` is set to 100: it is very low, but in this way all the images have enough features and we experimentally notice that this change improves a little bit the accuracy at the cost of high number of total features and, consequently, the lengthening of time of training. We try also to apply a regular sampling in space scale instead of Surf detector, but the result is worst.

Once all the images data are computed, kmeans algorithm is applied on extracted data in order to find k cluster centroids. The kmeans algorithm is limited at ten iteration: this choice does not

guarantee the convergence of the algorithm, but it experimentally is a good compromise between time of execution and a good result.

After the kmeans has found the cluster centroids, two histograms are created for each image and both are taken from [Van Gemert et al., 2008]:

$$CB(w) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases}1 & if\ w = \underset{v\in V}{\operatorname{argmin}}((D(v,r_i)) \\ 0 & otherwise\end{cases} \quad (1)$$

$$PLA(w) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases}K_\sigma(D(w,r_i)) & if\ w = \underset{v\in V}{\operatorname{argmin}}((D(v,r_i)) \\ 0 & otherwise\end{cases} \quad (2)$$

where

$$K_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2}\frac{x^2}{\sigma^2}\right).$$

Where *n* is the total number of features of an image, r_i is the i-th feature of the image, *w* is a "word" of vocabulary *V* and $D(w,r_i)$ is the distance between images. In the report, the "standard" histograms are those reported in the formula number (1), instead the "plausible" histograms are those reported in the formula number (2).

All the features previously found are associated to bins on the base of (1) and (2).

In Figure 2 you can see the two histograms of one random image with 20 bins.

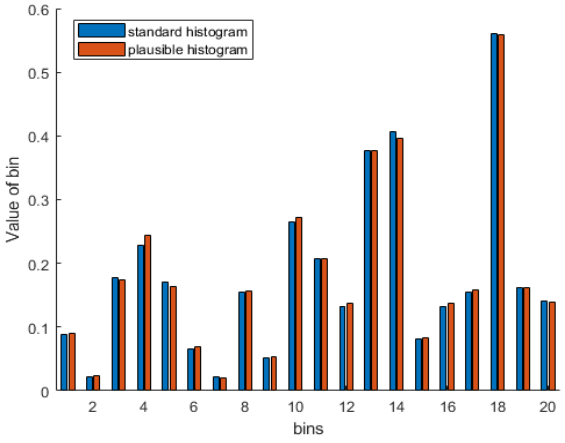

Figure 2: Normalized distribution of features per bins.

Then, the histograms are normalized to have norm equal to 1. Also, a total histogram relative to the number of images for each "word" is created. The Inverse Document Frequency (IDF) array is calculated on this total histogram and it gives a weight at each visual word: we experimentally show that the using of IDF technique improves accuracy. We try to create a stop word list with a threshold to block most common "words", but there is not an improve in accuracy.

15 one-vs-rest SVM both linear and gaussian (based on Euclidean distance) are implemented in Support Vector Machine (SVM) training: the positive class has labelled as "1" and the negative class as "-1". An attempt with gaussian based on Chi2 Kernel and Earth Mover's Distance (EMD) Kernel is done, but for Chi2 Kernel there is problem with scale and so all the images are classified in the same class; instead, for EMD kernel the execution is too long for my pc to resolve the algorithm in acceptable time.

Testing

In the testing phase all the images of the test dataset are evaluated. Like in training phase, in this phase a resize is done and 128-values SURF features are extracted from points detected by SURF detector. The histograms are built on the base of the centroids found in training phase. Two techniques are employed to classify images: nearest neighbour and SVM. Both methods are evaluated with different numbers of bins for each histogram, in particular numbers of bins are 15, 20, 50, 100, 250 ,500, 750, 900.

The first method is nearest neighbour in which the histogram of the testing image is compared to histograms of training images. The n training images that have histogram with smaller Euclidian distance from the testing image histogram vote for the class. We try different numbers of voting images (1, 3, 5, 10, 15, 20, 50) to find the best number of voters.

The second way uses SVM: we try both linear and gaussian SVM to evaluate which is better of the two. The best scale of kernel is chosen automatically.

Results

Each method is evaluated with different values of bins to find the best number of bins that maximize accuracy (15, 20, 50, 100, 250 ,500, 750, 900 bins). To have comparable performance we use same histograms to train SVM and to evaluate nearest neighbour method.

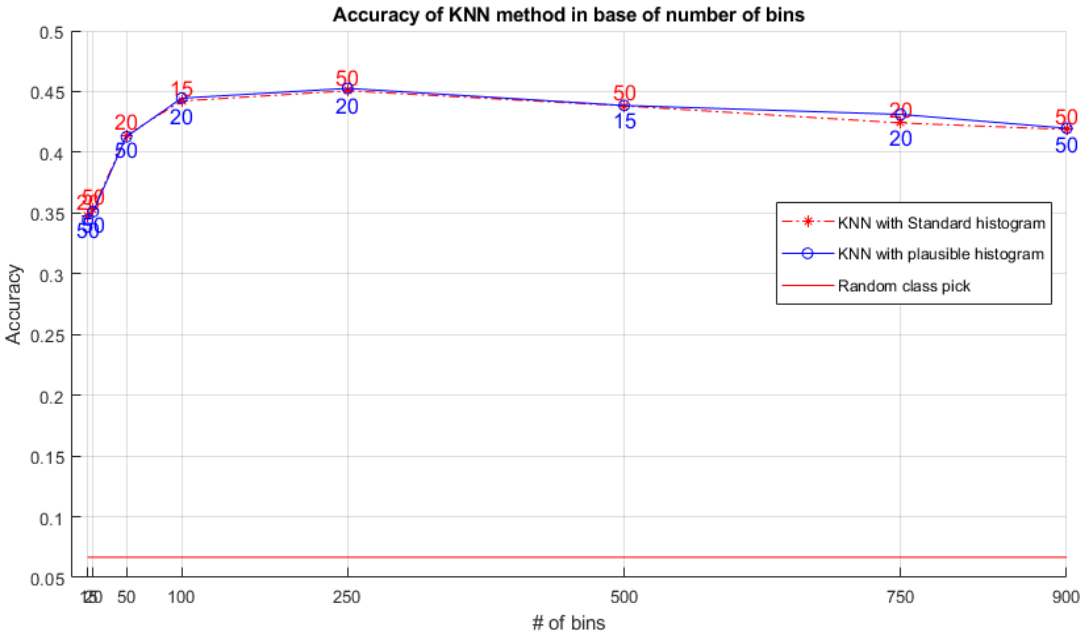

Figure 3: Accuracy of KNN method.

Figure 3 shows the accuracy of nearest neighbour technique. The accuracy is found as $acc = \frac{trace(confusion\ matrix)}{\#total\ test\ images}$. We can notice that the using of plausible histograms or standard

histograms does not change a lot the accuracy; however, plausible histogram is better of the two. The accuracy has a maximum of 0,4526, equal to 45,26% of probability to choose right class, using 250 bins. In every case all the results are better than a random classifier $acc_{rand} = \frac{1}{\#numclass} = \frac{1}{15} = 0{,}06667$ (6,67%). Different numbers of voting images are chosen to find the best configuration of k nearest neighbour (KNN): we use values 1, 3, 5, 10, 15, 20, 50 as values of k. For each numbers of bins in figure 3, indeed, we can see the numbers of k (voting images) that achieve the best accuracy.

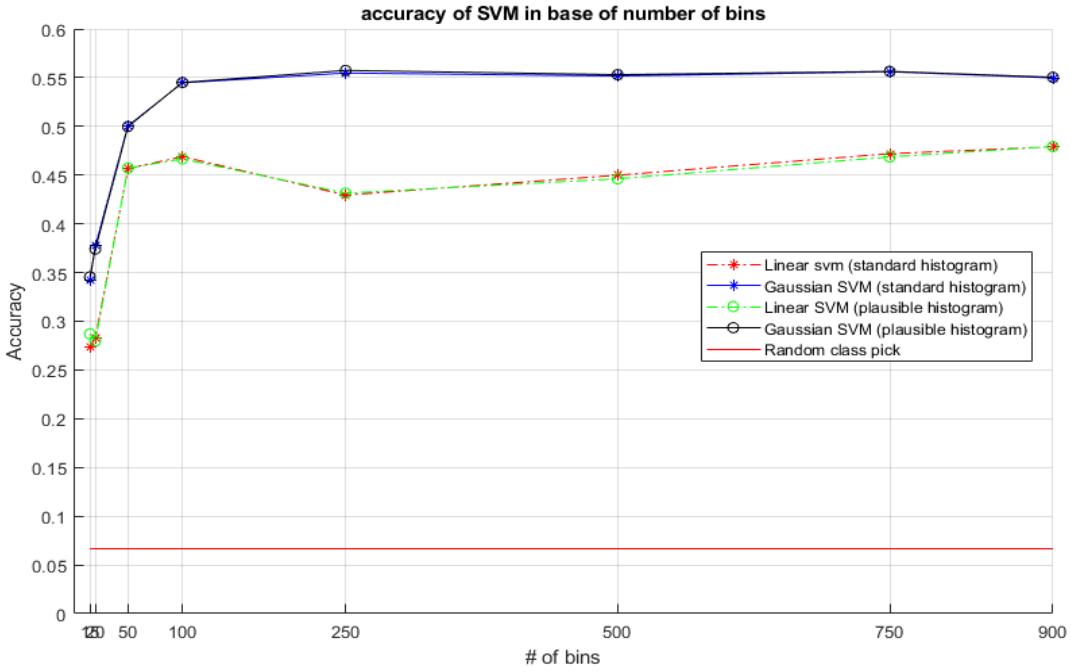

Figure 4: Accuracy of SVM method.

Figure 4 shows the accuracy of SVM technique found with the same formula as before. We can notice that linear SVM is always worse than gaussian SVM. Like in nearest neighbour the difference of accuracy between standard and plausible histograms is very small and the using of plausible histograms is barely better. We have the best accuracy which is 0,5575 (55,75%) with 250 bins.

We can observe that accuracy in KNN and in gaussian SVM decreases when numbers of bins become too big: this is probably because the features are arranged very randomly in bins and a good classification cannot be done.

Conclusion

In conclusion, like figure 5 proves, gaussian SVM classifier is always better than KNN, either with plausible histograms or with standard histograms. The result is not so high, because we did some implementation choices to save time at the cost of better accuracy.

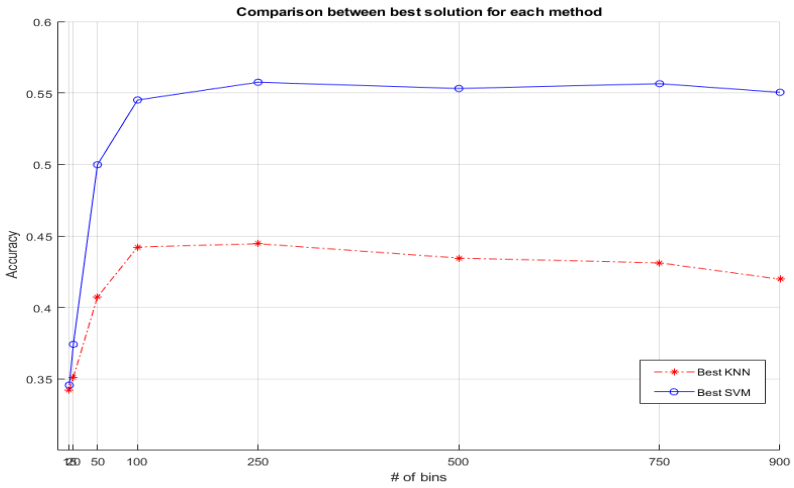

Figure 5: Comparison between SVM and KNN

In confusion matrices, figures 6 and 7, we can notice that the trend of SVM is similar to KNN, but with a better accuracy. We can observe obvious mismatching: for example Bedroom-Kitchen-Livingroom-Office are often confused and also Forest-Mountain-OpenCountry are mismatched. Another observation is that some classes, like Forest, have great accuracy in their testing class but they generate high wrong classification in other classes: i.e. in KNN 38% of OpenCountry images are classified as Forest and only 24% of images are classified correctly. This trend is also present in SVM but it is less significant than KNN.

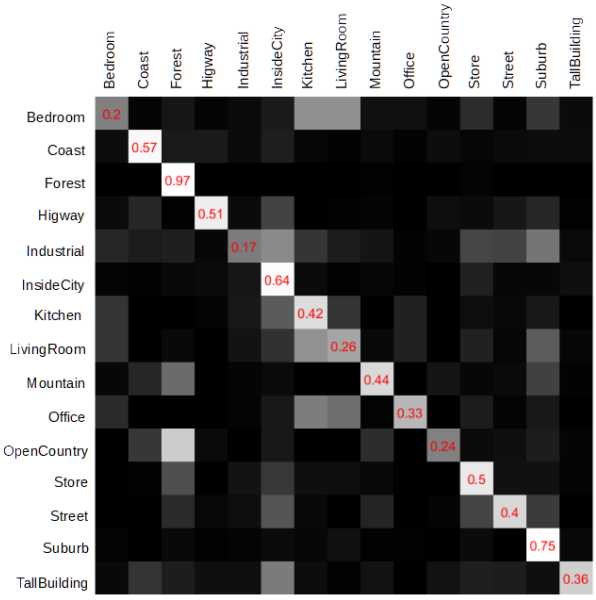

Figure 6: confusion matrix of best KNN.

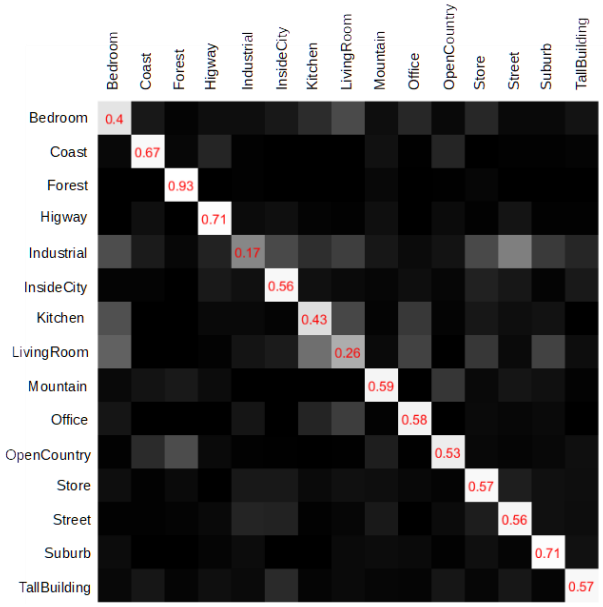

Figure 7: confusion matrix of best SVM.

References

[Van Gemert et al., 2008] Van Gemert, J. C., Geusebroek, J.-M., Veenman, C. J., and Smeulders, A. W. (2008). Kernel codebooks for scene categorization. In European conference on computer vision, pages 696–709. Springer

Felice Andrea Pellegrino, Slides of “Computer vision and pattern recognition”, Units A.Y. 2020-2021

[Falchi et al.,2010] Falchi F., Armato G., (2010). kNN based image classification relying on local feature similarity.

[Jinho k. et al.] Jinho k., Byung-soo k., Salvarese S. Comparing Image Classification Methods: K-Nearest-Neighbor and Support-Vector-Machines

[Frederic Jurie and Bill Triggs] Creating Efficient Codebooks for Visual Recognition