C++ Project

Image processing using SVM

Anthony Clerc

October 2016

1 Influences of the number of bins

The Figure 1 shows the evolution of the accuracy depending of the number of histogram bins. Generally speaking, a number of bins too low (under 30) does not describe enough well the image and the accuracy is weak. At the opposite, a number too high is also not optimal and leads to a decrease in the accuracy, as can be seen clearly with the blue curve. In fact, more than 100 bins gives poor results. It can be explain by the fact that the model overfit, which means that the model become to specific and not enough general. The graph indicates 30 bins to obtain the optimal accuracy.

The office curve (Grey) is specific has it reaches 100% of accuracy regardless the number of bins. This interesting result can be explained by looking at an office image (Figure 2). We can see a lot of different colours with a majority of blue, white and black. The colour diversity is better for this classes than for the toilet or the corridor, this explain why the classifier detect easily this class.

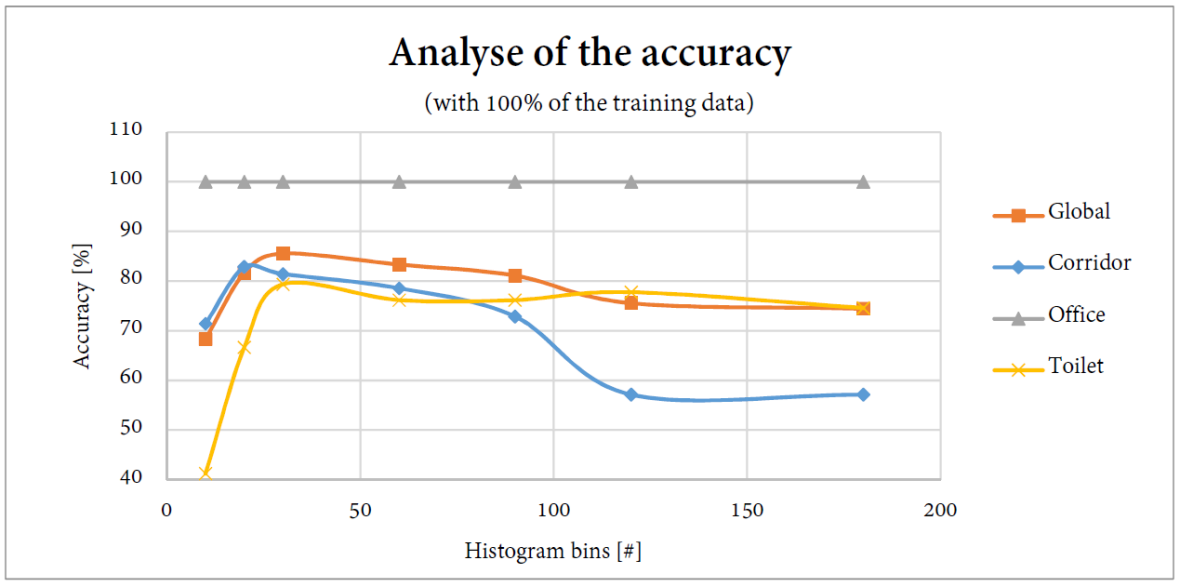

Figure 1: Analyse of the accuracy for different number of histogram bins

2 Influences of the training data size

The second graphic (Figure 3 describes the accuracy for different size of training data. It really interesting to see that increase of the number of data does not always lead to an increase of the

Figure 2: Office image

accuracy. Actually, 30% of the data give a result as good as 100%. And the figure shows that 50% of the data give worst result. This is due to the dataset, in fact, if there is a small dataset but with different image (modification of the light, the angle, zoom, etc.) the classifier is going to be better and more general. Conversely, a huge dataset made of identic pictures, will not bring more information to the classifier. The fluctuation is due to the randomise choice of the picture.

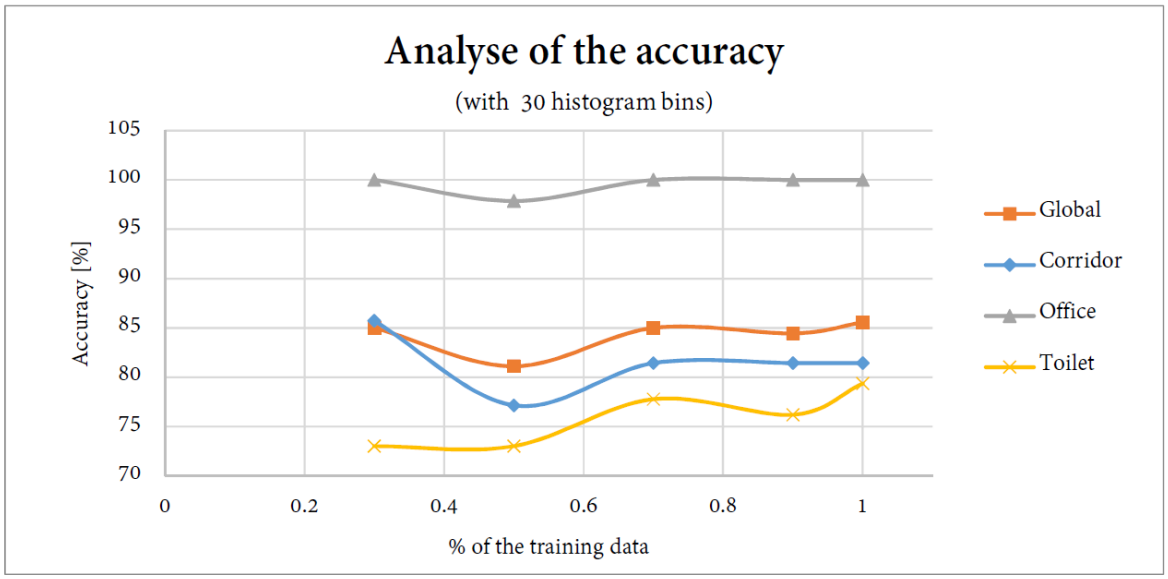

Figure 3: Analyse of the accuracy for different size of the training data

3 Improve of the Dataset class

It is quite difficult to make the class dataset completely generic, as the analyse of an image is in all respects different with an analyse of pure sensor value. However, it is possible to modify the program to read different data type, then, the analyse behind has to be specific. In order to make this class more generic, two things have to be modify. First the class should recognise what type of data are in the folders. And if this data are files (images, sounds, etc), the class return the path as it does currently. In the other case, if the data are just a ".txt" file with raw data, the class data set return an array with the raw data in it. Of course, the Dataset class as to inform the rest of the program which kind of data are returned. To be fore flexible with the data type, the class can be define as "Template Class", which allows to accept and use several data type.

4 Conclusion

To sum up, the choice of the classifier's parameter is not straightforward and require some analyse. Also, it is wrong to think that a high number of feature will leads to a good result, the features have to be decided depending of the type of data and their complexity. Finally, a huge training data set is as explained before, not always necessary.