RDFIA - TME 1

Lucrezia Tosato & Marie Diez

Contents

1 1-ab: SIFT / Bag of Words

1.1 Section 1 : SIFT

1.1.1 Computing the gradient of an image

1. A matrix is separable if its rank is ≤ 1. To calculate the rank, it is necessary to calculate the determinant of the matrix using Sarrus' rule.

For the matrix

$$M_x = \frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

the determinant is:

$$det(M_x) = (\frac{1}{4})^3 \cdot [(-1 \cdot 0 \cdot 0) + (0 \cdot 2 \cdot (-1)) + (1 \cdot (-2) \cdot 0)] - [(1 \cdot 0 \cdot (-1)) + (0 \cdot (-2) \cdot 1) + ((-1) \cdot 2 \cdot 0)] = [0] - [0] = 0$$

Since the determinant is equal to 0, we move on to calculate a sub-matrix of order 2 by removing the third row and third column from the original matrix M_x:

$$M_x^{'} = \begin{bmatrix} -1 & 0 \\ -2 & 0 \end{bmatrix}$$

$$det(M_x^{'}) = (\frac{1}{4})^3 \cdot [(1 \cdot 0) - (0 \cdot (2))] = [0 - 0] = 0$$

Since the determinant of the sub-matrix is still equal to 0, it can be deduced that the matrix has the rank is 1.

We have the proof of correctness of the separability calculating $h_y h_x^T$ with $h_x = \frac{1}{2}\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$ and $h_y = \frac{1}{2}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$

$$h_y h_x^T = \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = M_x$$

Similar reasoning can be done for the matrix M_y to verify its separability.

2. It is useful to separate the convolution kernel because this way we must do 2 convolutions with 3 multiplications (six in total) instead of one with nine multiplications achieving the same results. With less multiplications computational complexity goes down and the network can run faster.

1.1.2 Computing the SIFT representation of a patch

3. The gaussian weighting enable to give more importance to the center pixel of patches and to avoid overlap between them, and that enable to remove noise and details.

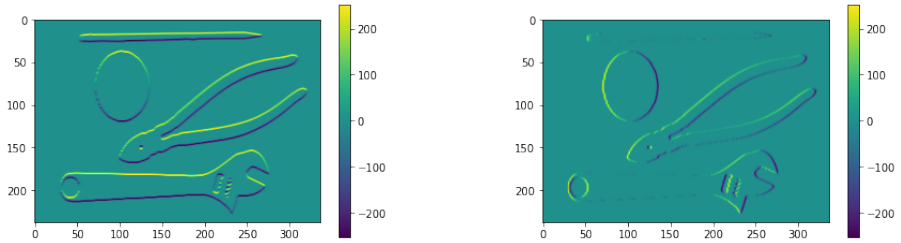
4. The discretization of the directions plays the role of achieving invariance to rotation. In this way, it will be possible to recognize an object even when in other images it will be rotated.

5. The two main steps for post processing are Normalizing and Thresholding: the vector is normalized to unit length to enhance invariance to affine changes in illumination. To suppress image features with low contrast, the interest points are usually also thresholded on t_max, finally the vector is again normalized, that enable to enhance invariance to non-affine changes in illumination.

6. The SIFT method is a useful method for describing an image area when doing image analysis because it encapsulates all the most important information about the area in a numeric vector, which can be compared with other numeric vectors to identify similar regions in multiple images. It is a method that is invariant of scale, rotation and illumination, therefore effective and widely usable, unlike other methods such as Harris which is not invariant to scale changes for example. Array comparison also allows comparison in a non-computationally complex way.

7. Interpretation of our results :

- Gradients - Ix Iy
  For that we need to inverse the filter before doing the convolution, we obtain :

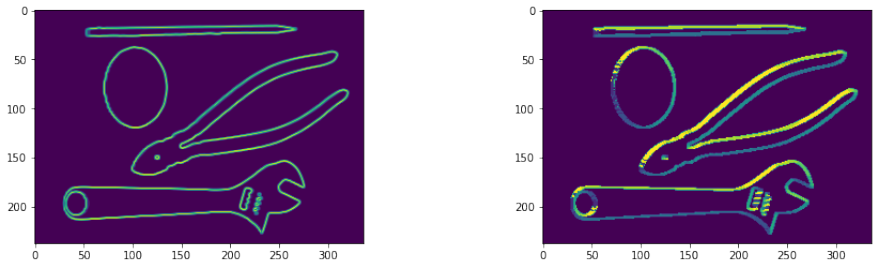


(a) Ix (b) Iy

Figure 1: Gradients

We can indeed observe that on the image Ix, horizontal edges are zeros. Similarly for Iy, verticale edges are zero too. At an egdes, the gradient crosses the contour from the darkest to the lightest intensities, what's explain why we can see positives and negatives values of gradients.

- Gradient norm and Gradient Orientation



(a) Gradient norm (b) Gradient orientation

Figure 2: Gradients norm and orientation

The gradient norm $G_n = \sqrt{I_x^2 + I_y^2}$ correspond to the edges, null value of gradient correspond to flat region, and positive values to the edges. Differents direction of can be seen in the gradient orientation image.

- Patches, gradient magnitude, gradient orientation and SIFT

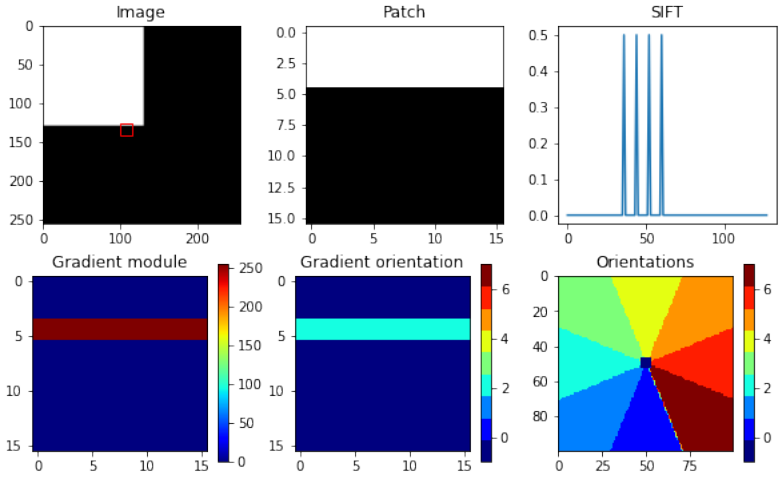


Figure 3: Patch 1

On this first result we can see a strong and constant modulus of the gradient, this translates into 4 peaks of the same value on the SIFT representation. Moreover we can see that the first 4 regions as well as the last 16 have a null value, indeed they correspond to the flat regions for which the norm of the gradient is 0, indeed the derivation on a constant region provides a null result.

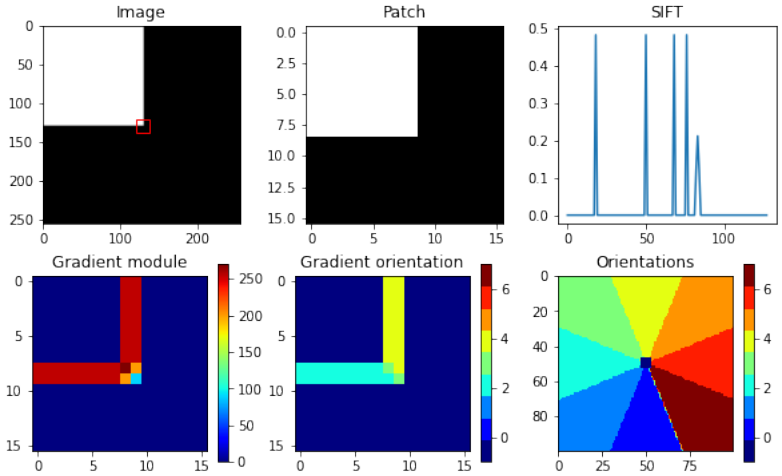


Figure 4: Patch 2

On this second result we can see 2 different orientations of the gradient, which corresponds well to our expectations for the presented patch. Moreover we can see that the gradient moduli are not constant, we can notably see a particularly low value in the lower right corner, which translates into a lower peak on the sift representation. Indeed the value of the bins is calculated with the sum of the gradient weighted norms for a given direction.

1.2 Section 2 : Visual Dictionnary

8. Each SIFT descriptor corresponds to a region/patch with a certain pattern, this patch is a local descriptor of the image. All similar patterns from all of differents images can be grouped together using a clustering algorithm (in SIFT domain). The cluster centers correspond to the visual dictionary and represent the most representative local descriptors among all descriptors for a cluster. This dictionary will then be used to create a global Bow image descriptor from the locals descriptors. By giving this new descriptor as an input to an SVM, we will be able to classify the images of our dataset into different categories which should correspond to the expected categories. To summarize, the need for a visual dictionary for our image recognition goal is justified by the fact that what we want to achieve is a quick method of comparing a model image with all the objects detected in a new image in order to classify them correctly.

9. To demonstrate that the cluster's center that minimize the dispersion is the barycenter ($c = \frac{1}{n}\sum_{i=1}^{n} x_i$) of the points x_i, we need to start from the formula: $min_c \sum_{i=1}^{n} ||x_i - c||_2^2$, since we want to minimize this summation we have to derive it with respect to c:

$$\frac{\partial \sum_{i=1}^{n} ||x_i - c||_2^2}{\partial c} = 0$$

It's easily provable that $||x_i - c||_2^2$, that is equal to $\sum_{i=1}^{n}(x_i - c)^2$ during the derivation, create another summation that is not fundamental for the purpose of the demonstration, since at the end gives rise to two constants which are simplified, not affecting the demonstration.

For simplicity this second summation is not considered in this demonstration.

$$\sum_{i=1}^{n}(-2)(x_i - c) = 0 \rightarrow (-2)\sum_{i=1}^{n}(x_i - c) = 0 \rightarrow \sum_{i=1}^{n}(x_i - c) = 0 \rightarrow \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} c = 0 \rightarrow \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} c \rightarrow \sum_{i=1}^{n} x_i = nc$$

From this we can find c:

$$c = \frac{\sum_{i=1}^{n} x_i}{n}$$

that is exactly the barycenter.

10. To choose the "optimal" number of clusters there are two classes of methods :

- Direct methods
  For example the Elbow method consists in making a curve by running the k-means algorithm with different values of K and computing the variance of the different clusters. The optimal number of clusters is the point representing the elbow.
- Statistical Testing methods
  Consists of comparing evidence against null hypothesis. An example is the gap statistic

The choice of K is not obvious and depends on the problem.

11. We create a visual dictionary from the SIFTs and not directly on the patches of raw image pixels because we want to work on the elements that best represent the images. Indeed SIFT correspond to a vector containing the most significant informations of an image. It is useless to work with elements that bring weak information, it add a very important and unnecessary processing time. SIFT is invariant to scale, rotations and changes in brightness, if we took patches of raw pixels, they would depend on the RGB values of the image, the illumination, and the position of the object in the image taken into consideration. For these reasons, comparing numerical values of raw image pixels would not serve the purpose of recognition.

12. Interpretation of our results :

- $max\_images\_per\_class = 10$

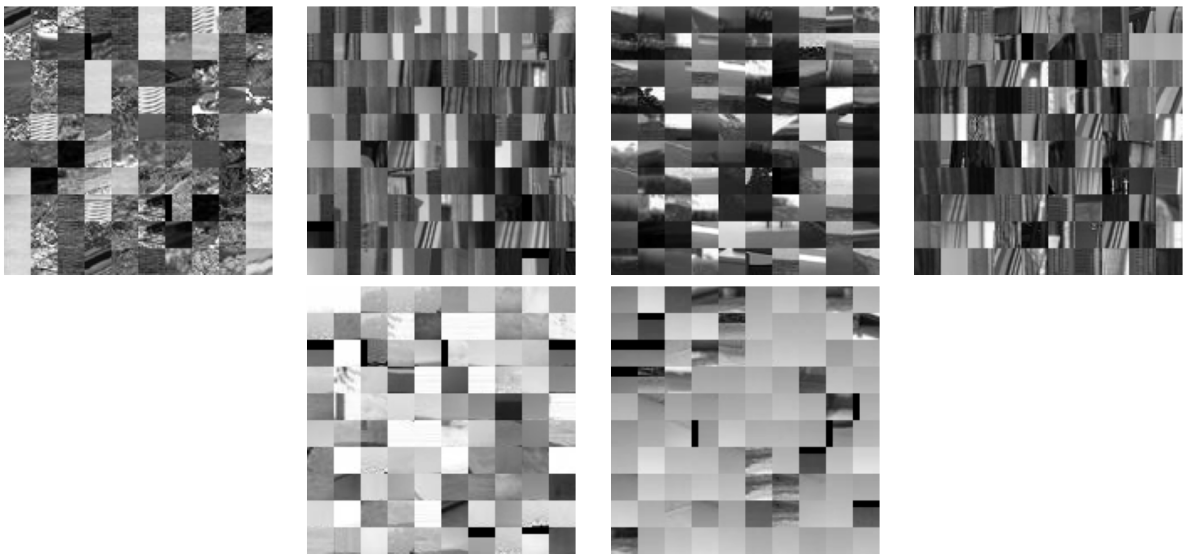


Figure 5: Top 100 regions the most similar in sift space to different cluster center - respectively center 0 - 4 - 9 - 10 - 13 - 16

- $max\_images\_per\_class = 1000$

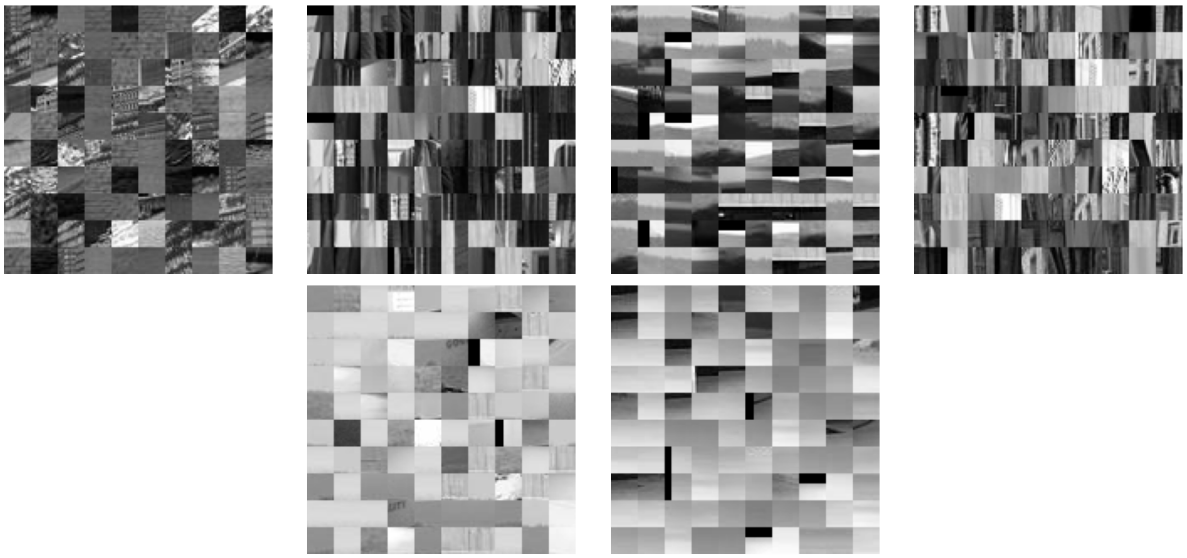


Figure 6: Top 100 regions the most similar in sift space to different cluster center - respectively center 0 - 4 - 9 - 10 - 13 - 16

- $max\_images\_per\_class = None$

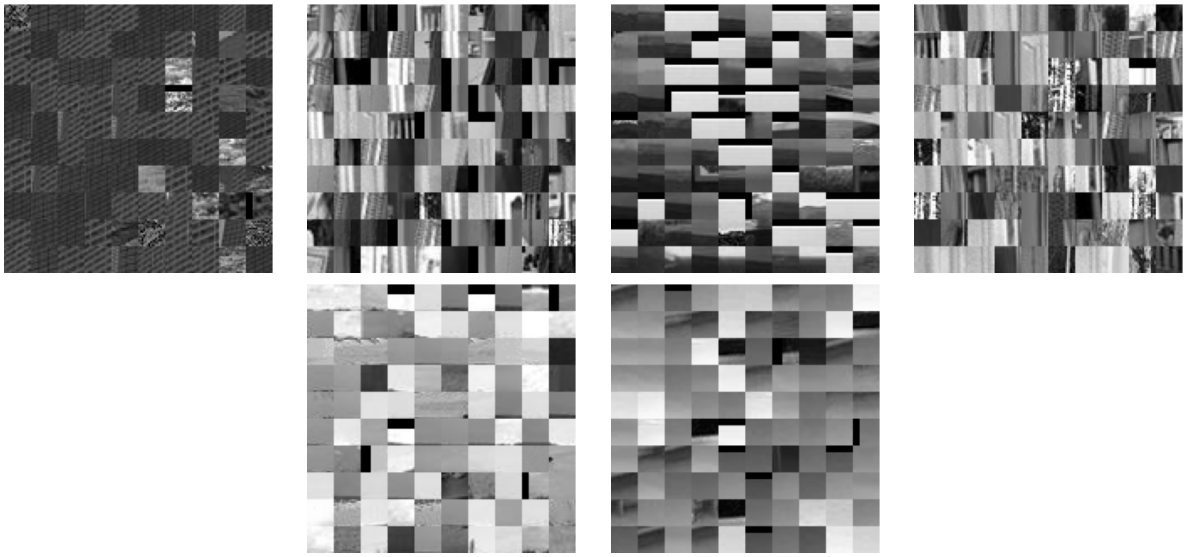


Figure 7: Top 100 regions the most similar in sift space to different cluster center - respectively center 0 - 4 - 9 - 10 - 13 - 16

We can see on these results that the similar patches are grouped together, in terms of SIFT descriptor. Indeed by choosing only the 100 closest patches to a center (visual word) in the SIFT space we obtain the 100 closest regions to the center in question.

Now, to deepen understanding our vision of the dictionary we can perform some experiments. If we want to look at the most similar region (in the sift space) of the center 16 :

```
center = 16
center_vect = vdict[center]
dist = ((sifts - center_vect)**2).sum(axis=1)
top50 = dist.argsort()[:50]
top50_regions = regions[top50]
display_images(top50_regions)
```

We obtain :

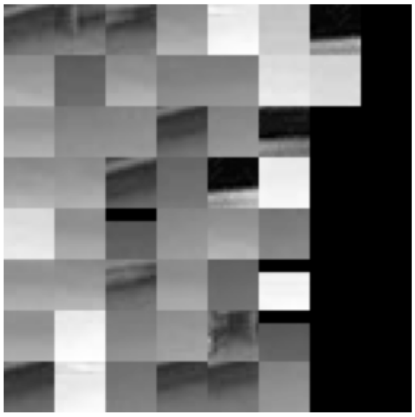


Figure 8: Top 50 regions the most similar in sift space to the cluster 16

We can see that the results look very similar, we can deduce that the visual word corresponding to this center is similar to this regions.

If we want the most similar region to the center 0 :

```
center = 0
center_vect = vdict[center]
dist = ((sifts - center_vect)**2).sum(axis=1)
top50 = dist.argsort()[:1]
top50_regions = regions[top50]
print(top50_regions.shape)
display_images(top50_regions)
```

We obtain :

Figure 9: The most similar in sift space to the cluster 0

If we want the less similar region to the center 0 :

```
center = 0
center_vect = vdict[center]
dist = ((sifts - center_vect)**2).sum(axis=1)
top50 = dist.argsort()[-1:]
top50_regions = regions[top50]
print(top50_regions.shape)
display_images(top50_regions)
```

We obtain :

Figure 10: The less similar in sift space to the cluster 0

That make sens because with previous result we can imagin that the visual word correponding to the center 0 is very textured as the most similar result, and the opposit of textured region is indeed flat region like in the less similar.

1.3 Section 3 - Bag of Words (BoW)

13. The z vector is the new global descriptor of an image and represent the similarity between the patches of the image and "mean" descriptors of the visual dictionary. Concretely, z tells us what is the importance of each visuals word in the image, which are the most present, the least present, so it is the number of patchs that have a specific bag of words features as their closest features.

14.

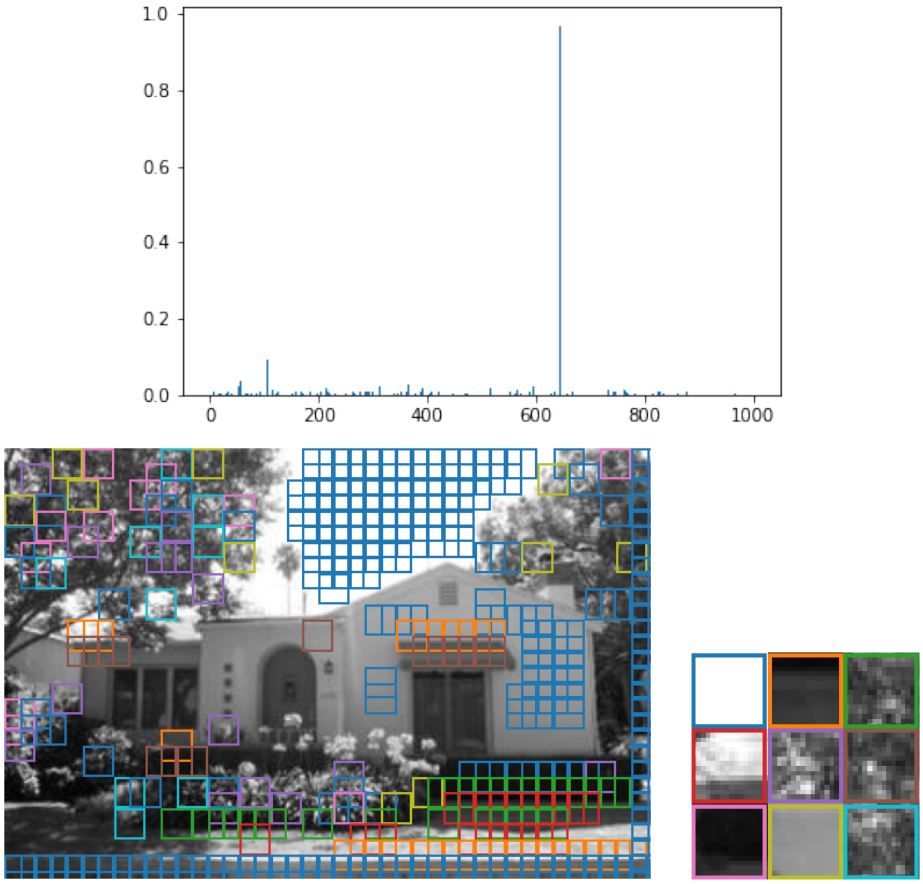


Figure 11: BoW - Image - Patches

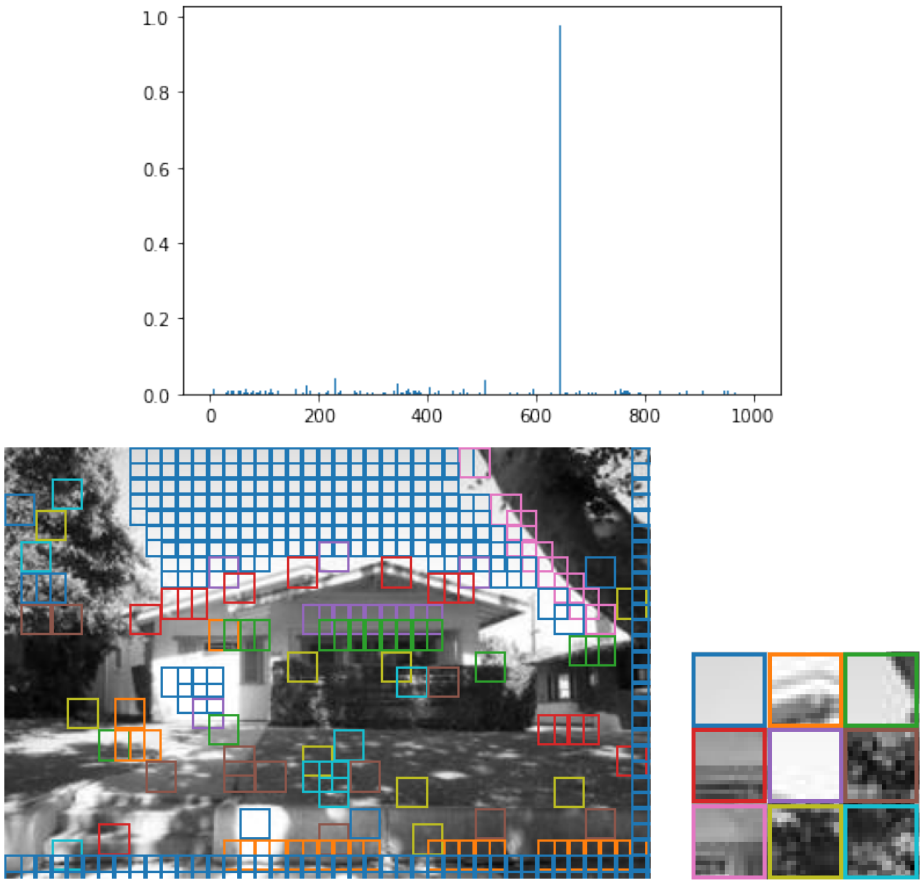


Figure 12: BoW - Image - Patches

We can find 3 results.
The first one corresponds to the BoW of the image, we can see that among all the SIFT descriptors of the image 1 many are close to the same visual word. The image has many similar regions in terms of SIFT descriptor and therefore belonging to the same cluster associated with the visual word, this visual word is very representative of the image.
The last image corresponds to the 9 most used descriptors of the visual dictionary. The most present visual word corresponds to the "flattest" region of the image here represented in white (which is equivalent to any other uniform region in terms of SIFT descriptor). We can find the regions of the image closest to this descriptor (blue rectangle). And this for the 9 descriptors from the most present to the least present in the image.
The final image represents the regions associated with the 9 visual words most present in the image with their locations. The association of a region with a visual word corresponds to the visual word closest to the SIFT descriptor of the region in question.

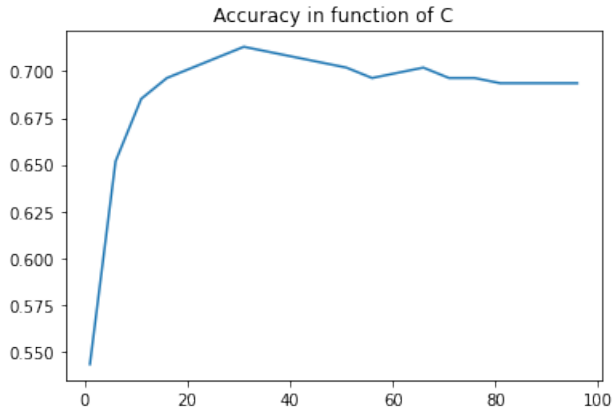
15. The purpose of nearest-neighbors encoding is to associate a descriptors with the class that is closest to it, and to be able to grouping all similar descriptors. Another method that can be used is to consider a larger number of classes each with a certain probability.

16. The interest in sum pooling is to aggregate and quantify which are the most frequent descriptors in an image. In particular, this type of pooling takes the sum of inputs, so we can have up to 1000 descriptors for an image, each with a different weight according to the frequency with which it appears in the image. Another type of pooling that we could use is the max pooling, in this case we would save only the maximum value, so an image would be represented only by a descriptor, this method is dangerous because we are going to lose a lot of information, which in many cases can make the difference to differentiate images.

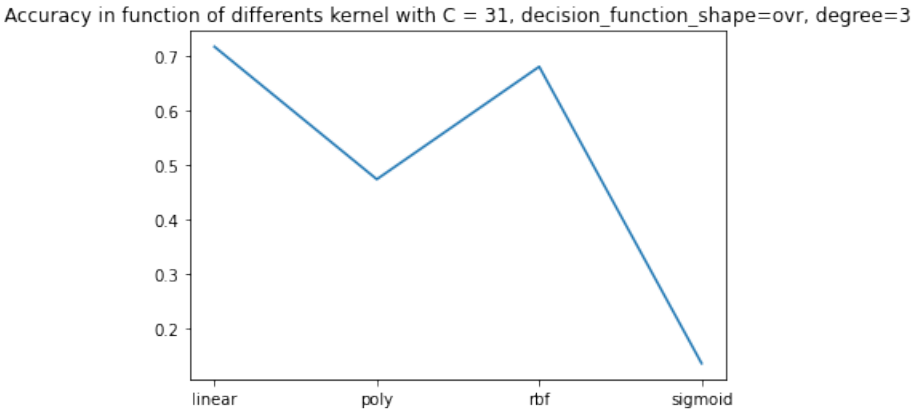
17. The interest in using L2-normalisation is to be able to compare SIFT with different values. In particular, depending on the size of the image we can find 10 clusters as well as 1000 and we want to be able to compare them. Instead of using the L2 normalization we could use the L1 normalization, the difference between the two is that the L2 normalization tends to exaggerate the distance between the values, while with the L1 normalization the distance between the values is always equal.

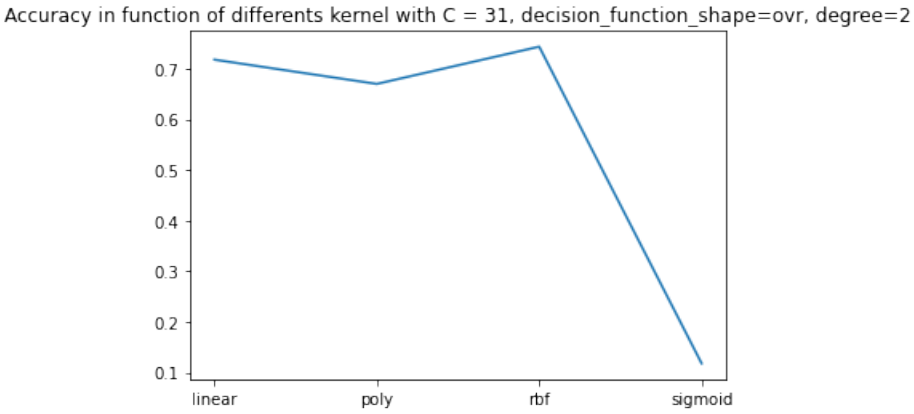
2 1-c: SVM

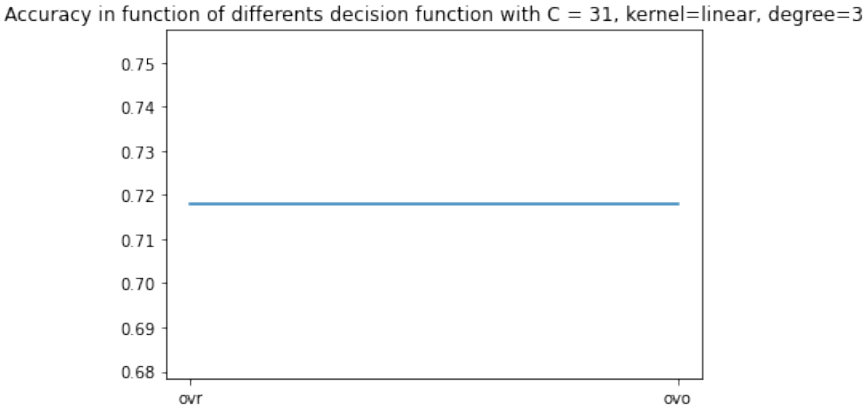
1. Here are the results of the accuracy curves according to different values for different hyperparameters, the C parameter is ranging from 0 to 100 by steps of 5. This is the resulting plot :



We can see that a high value of C increase the accuracy of our model until a C of 31. From now on, we'll test sequentially various hyperparameters while keeping the best value we got beforehand, our C value will be fixed for the next test, etc. We'll now test different degree value for the polynomial kernel, our result are the following:







We can see that the decision function do not seem to impact the results that mean that ovo or ovr approach are similar in this case, in term of results. While the value of C and the kernels have more impact on the accuracy. Indeed, C controls the trade-off between a smooth decision boundary and correct classification of the training set, it is the degree of punishment we wish to inflict on our model for each misclassified point. If C is large then we punish our model a lot, we risk overfitting and we have a low resistance to noise, because the margins are smaller. On the contrary, if C is small, we punish our model less, we have larger margins and we are therefore more resistant to noise, but we can fall into underfitting if we do not punish our model enough. So we have to find a compromise, we can find the value by looking at the performances on a validation set.

The 2 best kernels seems to be linear and rbf. The poylnomial kernel with a small degree provided also a good values, indeed a poylnomial kernel with degree of 1 is equivalent to a linear kernel.

The results of these plots are however biased, as there is an interdependence of the hyperparameters. We therefore implemented a grid search approach to be more precise and to obtain the best hyperparameters, we refined the search intervals thanks to the results of these first plots.

Here are the results of the gridsearch :

```
# Avec grid search
from sklearn.model_selection import GridSearchCV

parameters = {'kernel':('linear', 'rbf', 'poly', 'sigmoid'), 'C':np.arange(26,36,1), 'decision_function_shape':('ovr','ovo'), 'degree':(1,2,3)}
svc = SVC()
clf = GridSearchCV(svc, parameters)
clf.fit(X_train, y_train)
```

```
GridSearchCV(cv=None, error_score=nan,
             estimator=SVC(C=1.0, break_ties=False, cache_size=200,
                           class_weight=None, coef0=0.0,
                           decision_function_shape='ovr', degree=3,
                           gamma='scale', kernel='rbf', max_iter=-1,
                           probability=False, random_state=None, shrinking=True,
                           tol=0.001, verbose=False),
             iid='deprecated', n_jobs=None,
             param_grid={'C': array([26, 27, 28, 29, 30, 31, 32, 33, 34, 35]),
                         'decision_function_shape': ('ovr', 'ovo'),
                         'degree': (1, 2, 3),
                         'kernel': ('linear', 'rbf', 'poly', 'sigmoid')},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=0)
```

```
[8] clf.best_params_
```

```
{'C': 35, 'decision_function_shape': 'ovr', 'degree': 1, 'kernel': 'rbf'}
```

Figure 13: GridSearch results

With best score during mean cross-validated of the best hyperparameters found during the gridsearch :

```
clf.best_score_
```

```
0.7376916984664124
```

Figure 14: Best score

2.

- C
  C is regularization parameter for SVM, it controls the power of the regularization, how much you want to punish your model for each misclassified point. We explain in the question 1 the impact of high or low value and C.
- kernel
  Corresponds to a projection that allows the projection of points in another space in where the observations are more easily linearly separable after this transformation. That enable to have more complex boundaries.
- degree
  Correspond to the degree of the polynomial kernel function ('poly') and then is ignored by all other kernels.
- decision_function_shape
  Ovr correspond to one-vs-rest approach and Ovo to one-vs-one. With ovr with obtained N classifier while with ovo we obtained N(N-1)/2 classifier. These are 2 differents approach that can be use to make a multi-classifier from binary classifier.

3. The validation set is used to tune the hyperparameters of a model, once we set them we use a test-set to compute the effectiveness of the model without learning bias because in effect the test dataset is the subset of the actual dataset, which has not yet been used to train the model. The model has no knowledge of this data set, so we evaluate it with the test dataset.