Image Processing Homework

Fabrizio Casadei - 1952529

March-April 2021

Exercise 1

Data for the exercise:
Image:

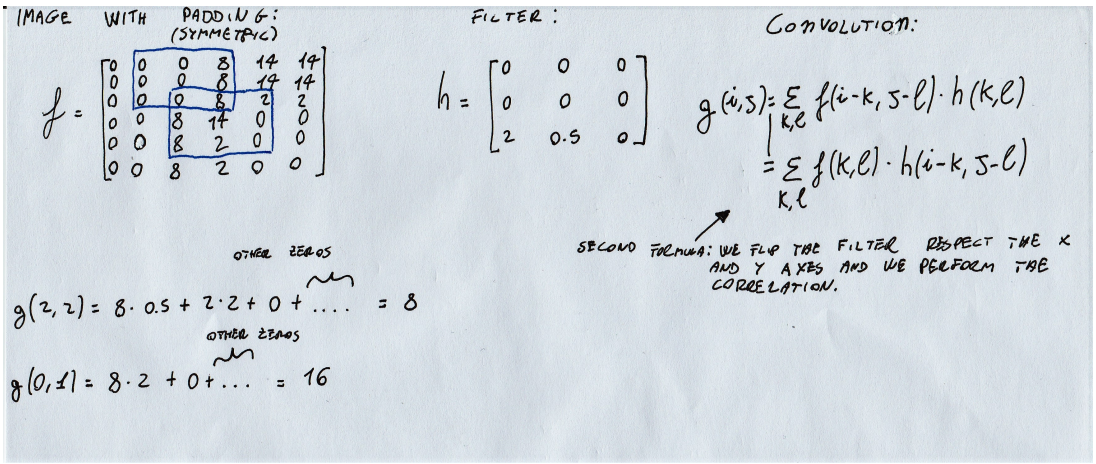
0	0	8	14
0	0	8	2
0	8	14	0
0	8	2	0

Kernel:

0	0	0
0	0	0
2	0.5	0

Paddings: symmetric padding

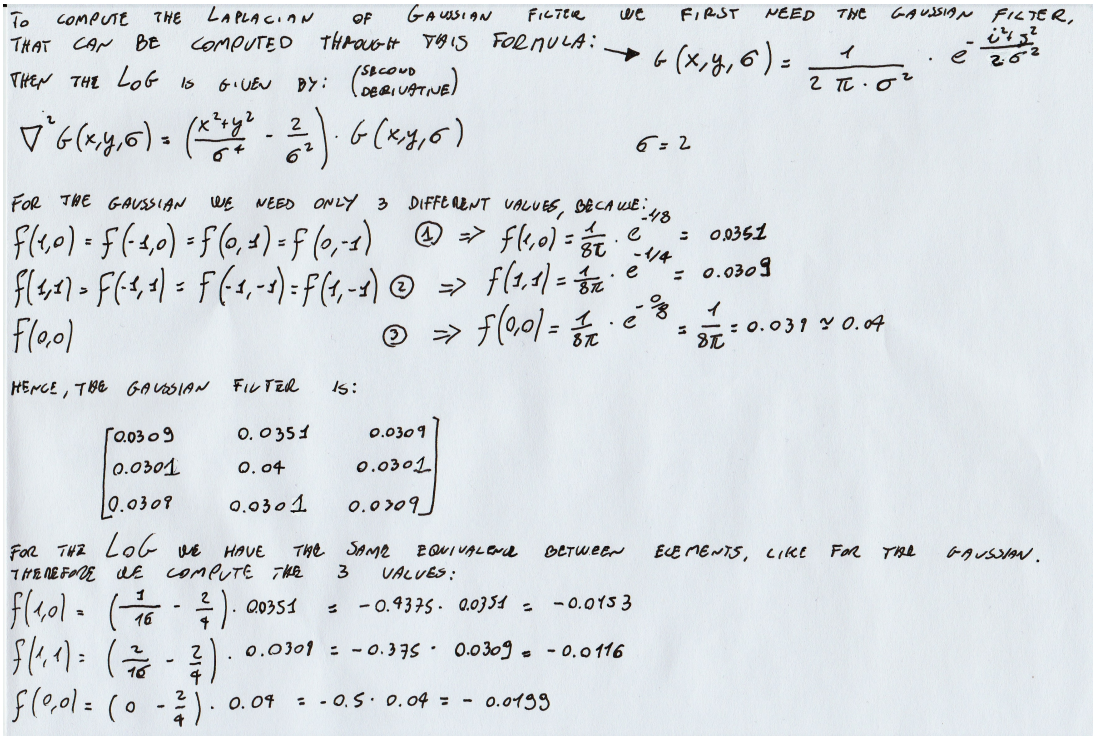
First Coords: 2 2
Second Coords: 0 1

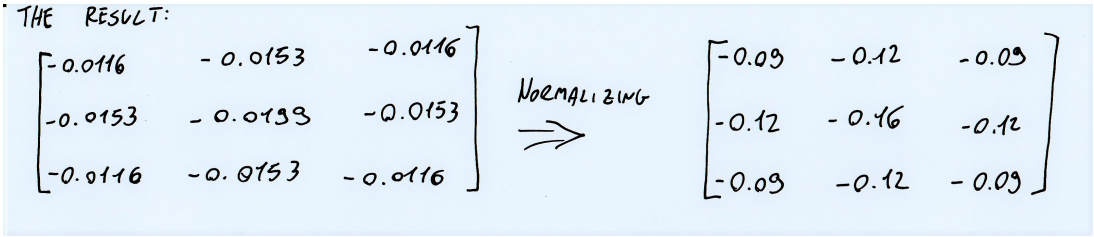


Beyond the same original image shape to the result, the symmetric padding allows adding information directly from the image using its data distribution mirroring the values. This is is both an advantage and a drawback, depending on where the interesting information are located and how much noise is present.

Exercise 2

Data for the exercise:
LOG filter dimension: 3×3, $\sigma = 2$





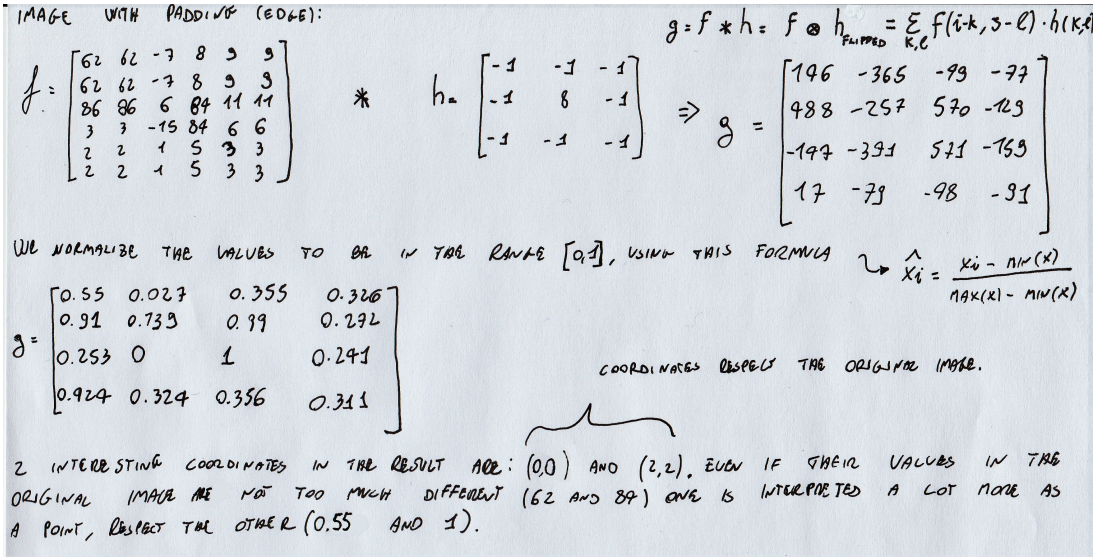
Exercise 3

Data for the exercise:
Image:

62	-7	8	9
86	6	84	11
3	-15	84	6
2	1	5	3

Kernel:

-1	-1	-1
-1	8	-1
-1	-1	-1



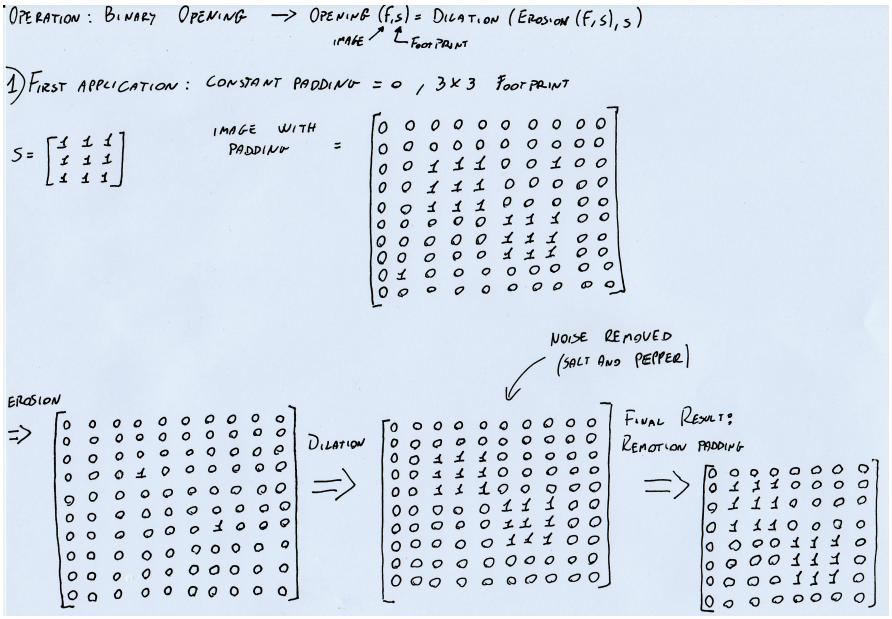
In this exercise has been used the "edge" padding to have the same size as the original image after the convolution and keep the data information from the image. The filter assigned is a *Laplacian edge detector filter*, more in the specific is a point filter which highlights points in the middle and the around points. Given the summation of the filter's elements equal to zero, we know that is a second-order derivative high pass filter (very sensitive to noise).

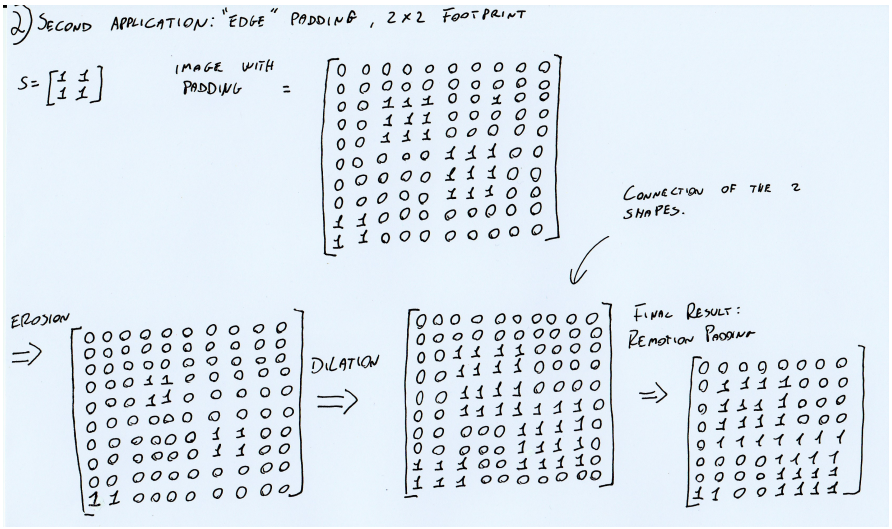
Exercise 4

Data for the exercise:
Image:

0	0	0	0	0	0	0	0
0	1	1	1	0	0	1	0
0	1	1	1	0	0	0	0
0	1	1	1	0	0	0	0
0	0	0	0	1	1	1	0
0	0	0	0	1	1	1	0
0	0	0	0	1	1	1	0
1	0	0	0	0	0	0	0

Operation: Binary Opening





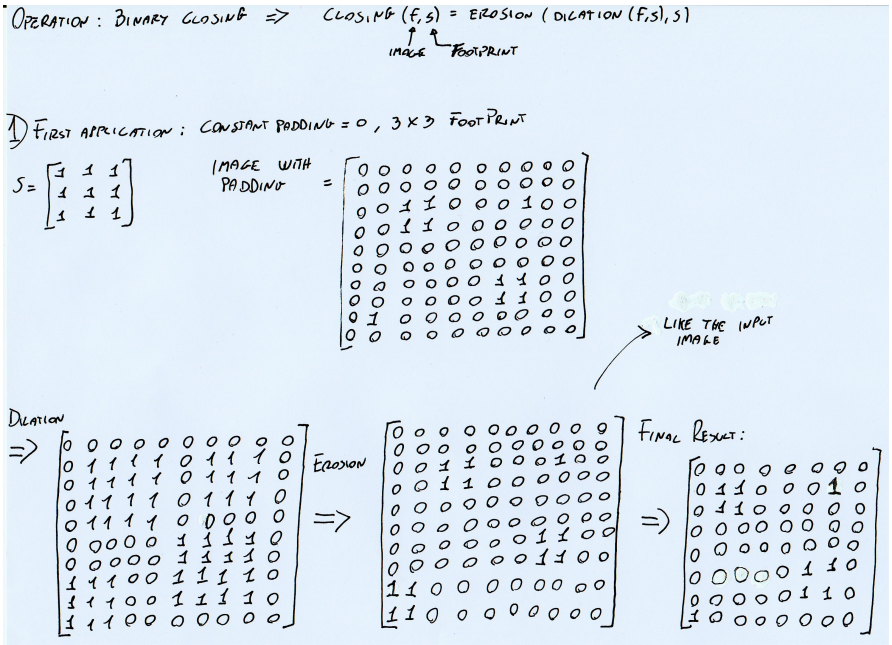
In this exercise has been chosen 2 kinds of padding (constant and edge) to have the same shape in output. The binary opening has been used to remove the so-called "salt and pepper noise" highlighted in the above capture, which has been erased in the output image. Another possible usage of opening is the extraction of large features from the image, removing not only noise but also small details of the image. in the example image, the 2 squares made up of ones can be seen as a large feature considering the structuring element used.

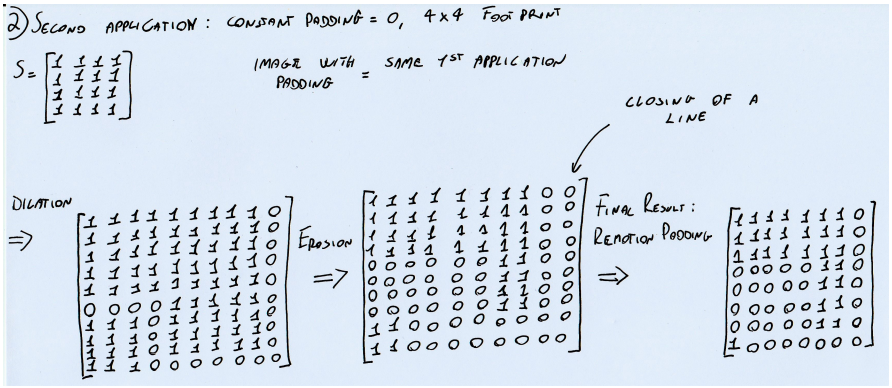
Exercise 5

Data for the exercise:
Image:

0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	0
0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0

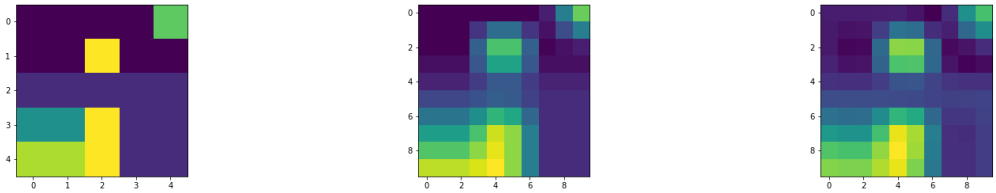
Operation: Binary Closing





Like in the previous exercise, to maintain consistency with the sizes of the input and the output we choose the constant padding and zero as a constant for both the applications. In the first after having applied the binary closure operator, we have obtained the same image as input for the process. One possible result after using closing operator can be the connection of edges in the image, and we see this in the second application using a 4×4 footprint. Moreover can be used also to fill small holes.

Exercise 6



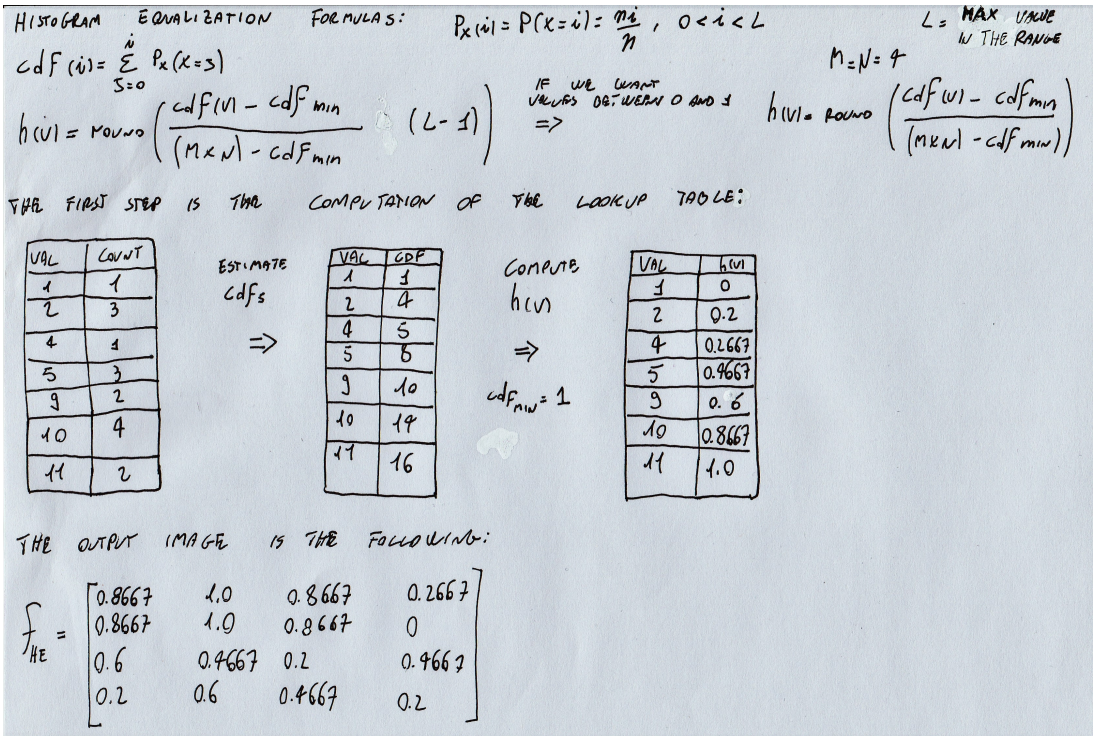
from left to right: the original image, the *Bilinear* interpolation and the *Bicubic* interpolation.

The key aspect for these 2 methods is how much a new pixel in the new resolution is influenced by each one in the original image. This is estimated using a distance function. The *bilinear interpolation* takes 4 pixels (2×2) into account to estimate new pixels while the *bicubic interpolation* considers 16 pixels (4×4). *Bicubic* interpolation solution is smoother, and in general it introduces less artifacts. An important drawback of the *Bicubic interpolation*, and so an advantage of the *bilinear*, is the computational speed which could be also one order of magnitude bigger.

Exercise 7

Data for the exercise:
image:

10	11	10	4
10	11	10	1
9	5	2	5
2	9	5	2



The *Histogram equalization* is a technique used to spread the pixel intensity into the whole range of possible values, which means having a greater contrast in the image leading to a sharping effect. It's possible to appreciate this in the above result matrix. The values have been normalized to be in the range of [0,1].

Exercise 8

Facing this kind of task we can assume that we want to segment these different types of objects guaranteeing a correct separation in the case of connected shapes. A manner to act is to use a threshold based on the intensity value of the pixel, then use morphological operations (erosion for foreground objects and dilation for background objects) to define markers and labels, that are used to separate the most important objects from the other elements. After these pre-processes, it's possible to use techniques like the *Watershed*, that exploit shapes thanks to the morphological nature of the elements. Applying the *Watershed* method all the connected shapes are correctly separated, but the quality of the results is very dependent on the definition of the markers/labels.

Another possible way to improve the algorithm is the usage of the *Entropy*, a measure of chaos and uncertainty which deal with spatial constraints (2 in the case of images) and the intensity constraints. To compute a correct segmentation task using Entropy, one crucial aspect it's the choice of the entropy disk used as a circular kernel for a convolution (based also on the level of the image's noise). In this case could be very useful to create a mask using the entropy, and use it with the original image to correctly separate any kind of shape. To conclude, entropy it's very useful for segmentation tasks because can highlight the curvature of objects used to define features and groups.

Exercise 9

If we want to extract frequencies of an image, it's possible to use the *Fourier Transform* or the *Fast Fourier Transform*, to have a representation in the frequency domain of an image. Centring the frequencies and multiplying them with a specific filter we can extract certain information useful for several operations. More in specific, for the ones asked in this question:

- The **High pass filter**, extract the information with high frequency in the image and these information can be used to make edge detection and images enhancement making them sharper. This filter completely removes the low frequencies data in the image. Could be crucial to apply denoising before applying it, otherwise the noise it's amplified (unlike the low pass filter).
- The **high-boost filter**, is a second-order derivative filter used to sharp the image, therefore, it's used to emphasize the high frequency components that represent the image details. This filter doesn't remove the low frequency components which are taken in the basic form of the signal, while the high frequencies are enhanced summing the result from *High pass filter*.
- The **homomorphic filter**, is a generalized technique used in image processing, involving a non linear mapping to a different domain, in which linear filter are applied, following mapping back to the original domain. and it's used for image enhancement, doing at the same time the normalization of the brightness across the image (intensity range compression) and increases the contrast (reflection). Can be also used to remove multiplicative noise like the variation of illumination in the scene. For doing this can be used any high pass filter.

Images analyzed with the *Fourier Transform* have different amplitudes and phases associated with the same universal set of Fourier components. Doing the convolution in frequency manipulates both amplitude and phases, as a function of 2D spatial frequency and orientation.

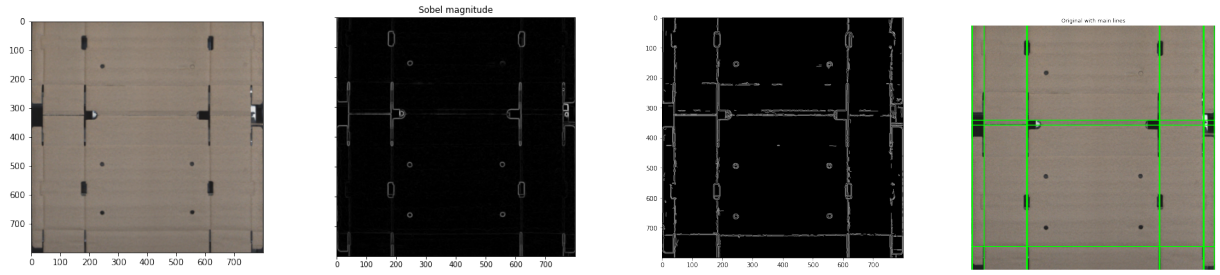
The shape and the orientation of an object can be inferred using the *Fourier Transform*. Changes in the orientation of the image in the spatial domain are visible also in the frequency domain. While regarding the shapes, images with different spatial frequencies lead to a different representation in frequency. For example, Information about the shape of an image can be inferred, applying the *Fourier Transform*, centralize the spectrum of the image, then multiplying with an *High-Pass filter* like the *Gaussian High-Pass Filter*, and return to the spatial domain: decentralizing the spectrum and doing the inverse of the Fourier transformation. By doing this operation it's possible to extract information about the edges and contours of the image.

Exercise 10

In order to extract the informative edges, has been computed several operations. First of all, all the images have been loaded, converted from BGR (because has been used the CV2 library) to grayscale, resized and performed a convolution with a 3×3 *Guassian filter* to denoise the images. The next process consists in the application of morphological operations to highlight all the edges in the images, especially the cardboard folding lines. More in the specific has been used the opening operation (erosion and then dilation), which produces a sharp effect on the folding lines. Then using the horizontal and vertical *sobel filter* has been computed the gradient magnitude as the hypothenuse of the 2 outcomes. This magnitude allows the detection of all contours in the image. The next task is the implementation of a method to detect the main lines from the found extracted edges. To do this, the first step has been the usage of the *Canny edge extractor* which detects edges using derivatives, non-maximum suppression and the hysteresis step (thresholding and linking). Hence, specific parameters has been defined for each image, regarding:

- **sigma**, the standard deviation of the Gaussian filter applied
- **The low threshold**, used in the hysteresis step to identify non-relevant pixels
- **The low threshold** used in the hysteresis step to identify the strong pixels.

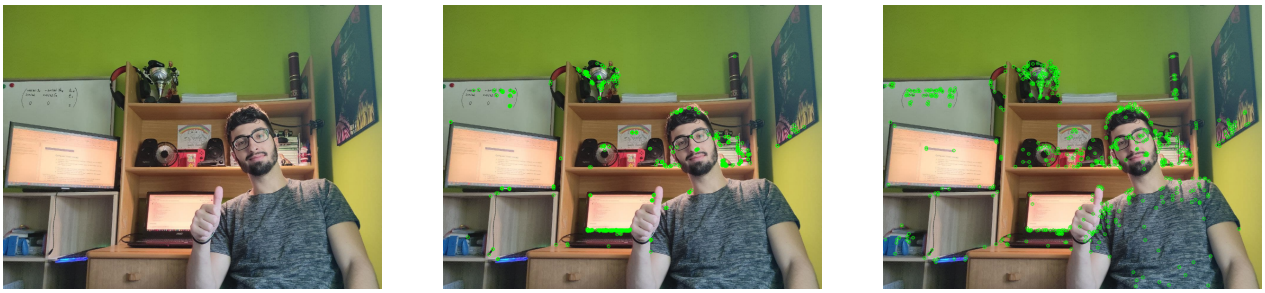
Then has been used one more time a morphological operator, which is the *dilation*, to make the edges thicker. Finally has been used the *Hough transform* to detect the lines and draw them into the original images. New parameters have been added for each image, regarding the angle of lines and the threshold for the minimum vote for a line. The results are now shown:



From left to right: an original example image, the application of the *Sobel* magnitude, the *Canny* outcome and the final result using *Hough transform*

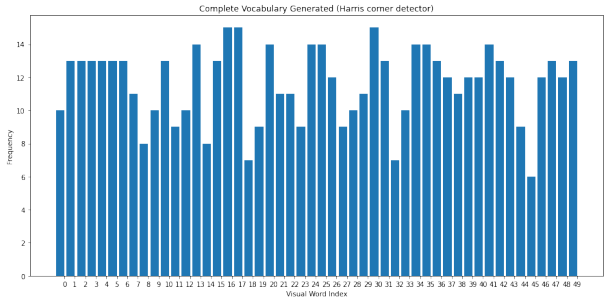
Exercise 11

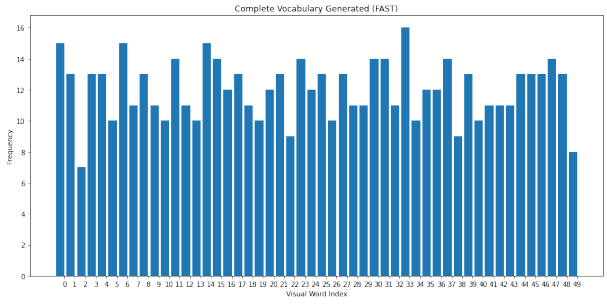
For the first part of the exercise, has been exploited 2 main techniques to extract features in images: *Harris corner detector* and *FAST*. Both are corner detectors, but *FAST* is used in real-time application thanks an optimization that work in this way: considering circle patches of an image and the intensity values inside this, the main clue is under which pixels inside the patch should be analyzed first, and if it's suitable to going in depth with the other or passing to another patch. fixing threshold values for both the methods and using *SIFT* descriptors has been found these results:

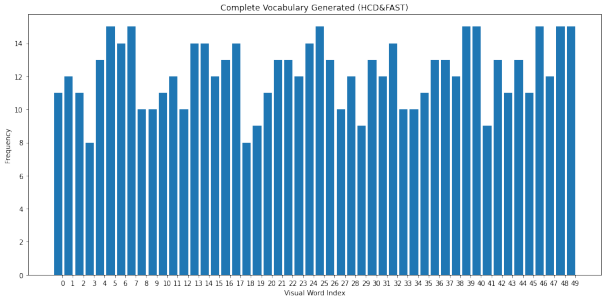


From left to right: the original image which represents the working environment, the features extracted using Harris (514 features) and the ones detected using FAST (648 features)

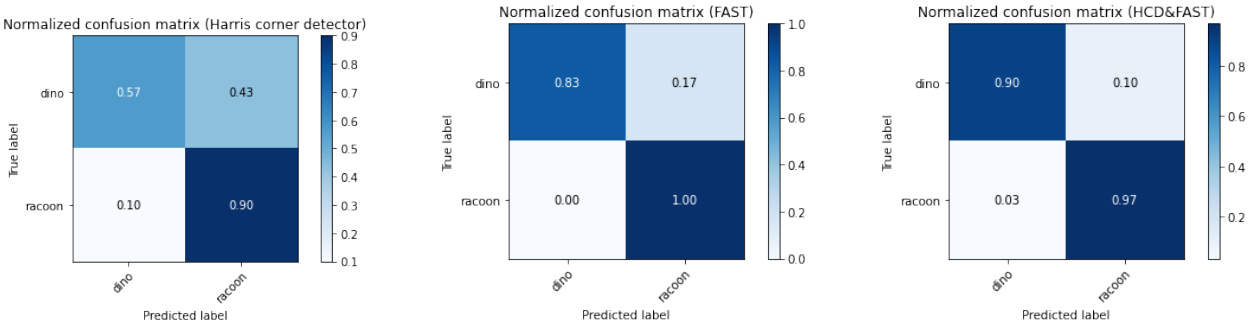
In the second part of this exercise, has been built a dataset with 2 classes: Dinosaurs and Racoons (for each class, 30 instances for the training and 30 for testing) and used in *Bag of words model*. The local image features have been extracted in 3 ways: using only Harris or only FAST and combining the features found by them. After the extraction of features, the data has been clustered defining a *Kmeans* model, then a normalization step has been performed to present the histrogram of words frequencies. As a classification model has been used the *SVM*, which exploits the maximum margin separator, a decision boundary with the largest possible distance from the data instances. Histrograms results are the following:







After having trained the *SVM* models, these are the results in terms of confusion matrix:



The time spent for feature extraction and accuracy results (can be derived from the diagonal of these matrices) are:

- *Harris corner detector* accuracy = 73% , time for features detection and building of descriptors: 18.10 [s]
- *FAST* accuracy = 91.7%, time for features detection and building of descriptors: 6.43 [s]
- *HCD + FAST* accuracy = 93,3%, time for features detection and building of descriptors: 13.1 [s]

Exercise 12

has been computed several pre-processing operations on the image, in detail: using CV2 the images have been converted from BGR to RGB, then the template image has been cropped cutting off the background information in order of having a better face matching. Then has been applied the *Histogram of oriented gradients* on both the template and the target image. Then we have worked directly on the image returned by *HOG* which describes the information relative to the direction and magnitude of gradients. Then we have used the *Normalized Cross Correlation* which is used to evaluate and find a correspondence between the target image and the template. from this operation, we fix a certain threshold used to choose up to which values of correspondence we can establish a consistent matching. Using the "Oscar night" target has been chosen a low threshold equal to 0.40, which returns a lot of correct face recognition boxes, and just 3 wrong matches that have been identified with a specific check. The *NCC* values of these boxes are between the threshold (0.40) and a maximum value equal to 0.43. While using the "Big Bang Theory" target, has been chosen a greater threshold equal to 0.605, given the higher level of correspondence, that returns also a wrong matching which has been localized and coloured differently (correct match is green, otherwise is red). In this case the NCC values are 0.22 (the wrong match), 0.6070, 0.6072 and 0.615. To verify if a bounding box contains a correct recognition has been applied one more time the *NCC* on the most correct box which has been returned from the *NCC* applied on *HOG* images and the others one, to filter the matches. The results obtained are the following:

From left to right:the target images, the templates and the results achieved

Exercise 13

To accomplish this task has been used several techniques studied in this course. to extract the main axis of the provided images, we have worked with 2 kinds of images: grayscale, and binary. As the first step, has been applied derivative filters like horizontal Sobel, vertical Sobel and the Laplacian. The type of the input image and the filter used has been empirically chosen. Then for each outcome has been performed the *Fourier Transform* and each frequency has been centred, then filtered with a *Gaussian High Pass filter* in order of having information about contours. Then the signal has been flipped to have higher values on the direction of the main axis and given the symmetry, has been taken the 2 points at maximum intensity to draw the axis. A correction about the vertical displacement has been performed to better fit the input images. The results obtained using this process are the following: