Advanced Machine Learning – Assignment #1

Alessio Sampieri 1765522
Michele Meo 1599032
Data Science, "Sapienza" University of Rome

Lorenzo Ceccomancini 1667798
Paolo Mandica 1898788
Due Date: October 30, 2020

Question 1

1.d

All following comments will relate to Fig.1, in which we specify the numbers of the figures also used in the following argument.

In the first figure, we obtain a 2D gaussian filter applying separately a 1D gaussian filter on the x-axis and the y-axis. It is possible because the gaussian filter is separable, it means that a 2D gaussian filter can be written as the product of two 1D gaussian filters along the axis, and because the convolution is a linear operation.

The separability of the gaussian filter allows us to reduce the computational complexity with respect to directly applying a 2D gaussian filter: this because the application of a 2D gaussian filter with size N on a picture of pXp pixels has a p^2N^2 computational complexity, instead the application of two 1D gaussian filters on the same image has a linear complexity with respect to the size of the filter, $2p^2N$.

In the second figure, we apply a 1D gaussian filter to smooth the signal along the x-axis and, then, we apply a derived gaussian filter along the y-axis to emphasize the horizontal edges. Regarding the third figure, we know that convolution is a linear operation, which means that for the commutative property the order in which we apply the filters does not matter: for this reason the third figure is identical to the second one.

For the fifth and sixth images the reasoning is the same as for the pair of images 2 and 3, the difference is in the application of the Gaussian filter and the derived gaussian filter, respectively, along the y axis and the x axis. The application of this filter allows us to emphasize the vertical edges.

The fourth figure is the result of the application of two Gaussian filters derived along the vertical and horizontal axes, it provides us a gradient of the picture, useful for identifying edges with different inclinations.

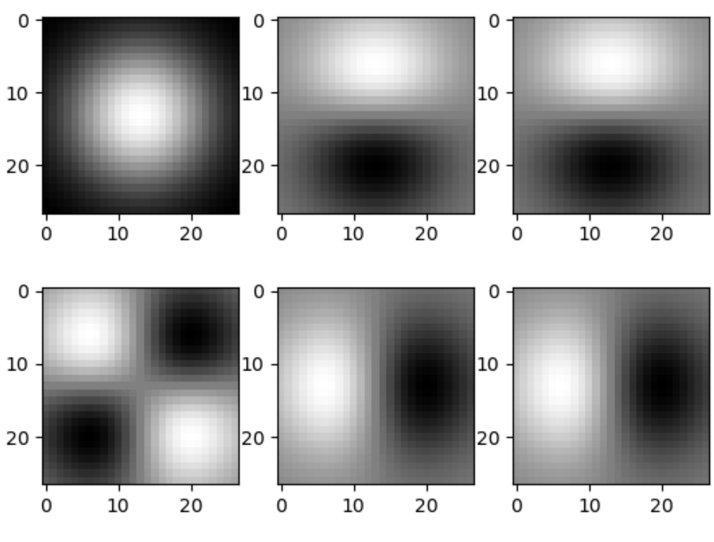

Figure 1: Impulse responses of 6 different filters. From the first row and from left to the right, we have: (1) first Gx, then Gx^T; (2) first Gx, then Dx^T; (3) first Dx^T, then Gx; (4) first Dx, then Dx^T; (5) first Dx, then Gx^T; (6) first Gx^T, then Dx.

1.e

In Fig.2 and Fig.3 we show, in order, the application of a Gaussian filter derived along the horizontal direction, a Gaussian filter derived along the vertical direction and, finally, the magnitude of the gradient, given by the norm l2 of the two filtered images previously obtained. Therefore we can see how, in the same order, each filter identifies respectively the vertical edges, the horizontal edges and, finally, all the edges within the image.

It is foundamental to smooth an image, in this case through a Gaussian filter, before applying the derivative operation as this allows us to cut all the high frequencies characterizing the image, for example frequencies due to noise or tiny edges that do not constitute the boundaries of objects in the image, such as surface textures.

Furthermore, we note how, thanks to the linearity of the convolution, it is possible to derive the Gaussian filter and then directly apply this derived Gaussian filter to the image instead of applying the two filters separately: this allows us, once again, to reduce the computational complexity of the operation.

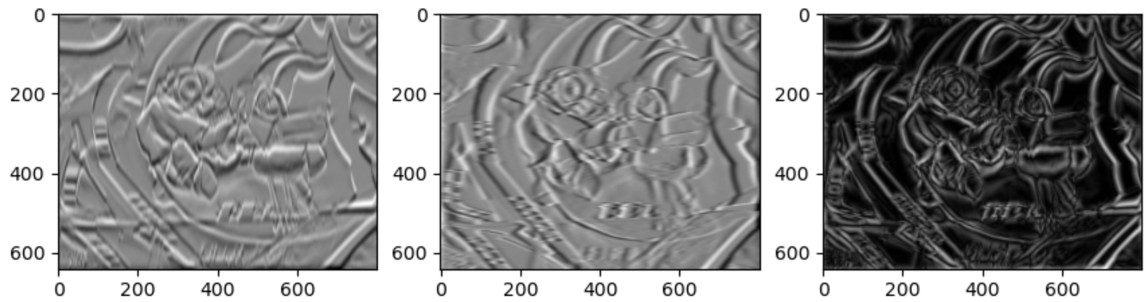

Figure 2: In order, filtered picture obtained applying Dx, Dx^T and the magnitude of the gradient.

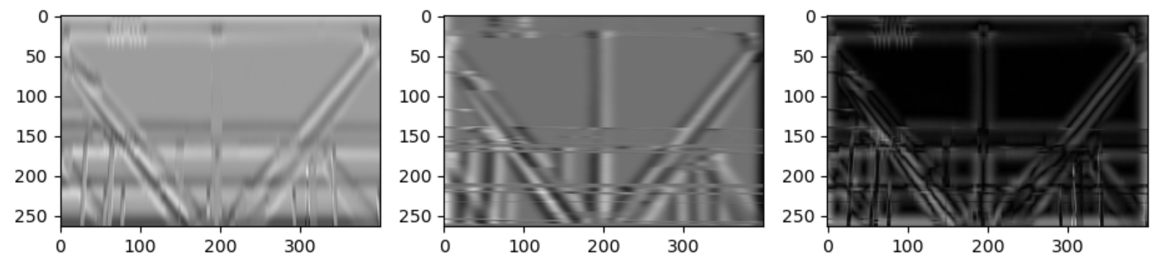

Figure 3: In order, filtered picture obtained applying Dx, Dx^T and the magnitude of the gradient.

Question 3

3.c

The following table shows the ability that different types of distances, histograms and number of bins have in recognizing and comparing images.

Hist. type	***Dist. type***	***# Bins***	***# correct over 100***
RGB	Intersect	*30*	*72*
RGB	Intersect	*10*	*70*
RGB	Intersect	*50*	*70*
RGB	Intersect	*5*	*69*
RG	Intersect	*30*	*65*
RG	Intersect	*50*	*65*
RG	Intersect	*5*	*63*
RG	Intersect	*10*	*62*
RGB	Chi2	5	*61*
RGB	Chi2	*10*	*60*
RG	Chi2	*5*	*58*
DxDy	Intersect	*10*	*58*
DxDy	Intersect	*30*	*58*
DxDy	Intersect	*50*	*58*
RGB	L2	*5*	*57*
RG	L2	*5*	*55*
RGB	L2	*10*	*54*
RG	Chi2	*10*	*53*
RG	L2	*10*	*52*
DxDy	Intersect	*5*	*50*
RG	Chi2	*30*	*42*
DxDy	Chi2	*30*	*41*
DxDy	Chi2	*10*	*40*
DxDy	L2	*30*	*40*
DxDy	Chi2	*5*	*39*
DxDy	L2	*5*	*39*
DxDy	L2	*10*	*39*
RG	L2	*30*	*39*
RGB	Chi2	*30*	*38*
DxDy	Chi2	*50*	*38*
DxDy	L2	*50*	*38*
RG	Chi2	*50*	*35*
RGB	L2	*30*	*34*
RG	L2	*50*	*31*
RGB	Chi2	*50*	*30*
RGB	L2	*50*	*30*

Table 1

It is possible to immediately notice how the first eight positions are occupied by the intersect distance. Subsequently RGB with intersect gives higher values for each number of bins, followed by RG and DxDy. However, considering the intersect distance, it can be seen that there is no substantial variation in correctness, in fact RGB with intersect remains between 69% and 72% correctness for each bin size analyzed. Typically, the lowest performance is achieved by increasing the bins to 50, often combined with L_2 distance. It is important to note that DxDy histograms never exceed 58% of correctness, therefore the other types of histograms are preferable.

Question 4

4.b

From the figures below it is possible to observe the different plots as the number of bins and the type of histogram vary. From these graphs it is clear how the best performances are obtained with the RGB histogram, as we have also been able to verify in point *3c*. Also, the distance which gives us the best performance is the intersect.
So, in general, it is possible to say that the distances L_2 and χ^2 are less performing than the intersect distance, as verified in *Table1* where their performance is actually lower.
On the other hand, it is possible to note that in the plots concerning the *Dxdy histogram* there is no prevalence of distance measurement. In fact, the three measures start at the bottom and then grow very slowly all together.

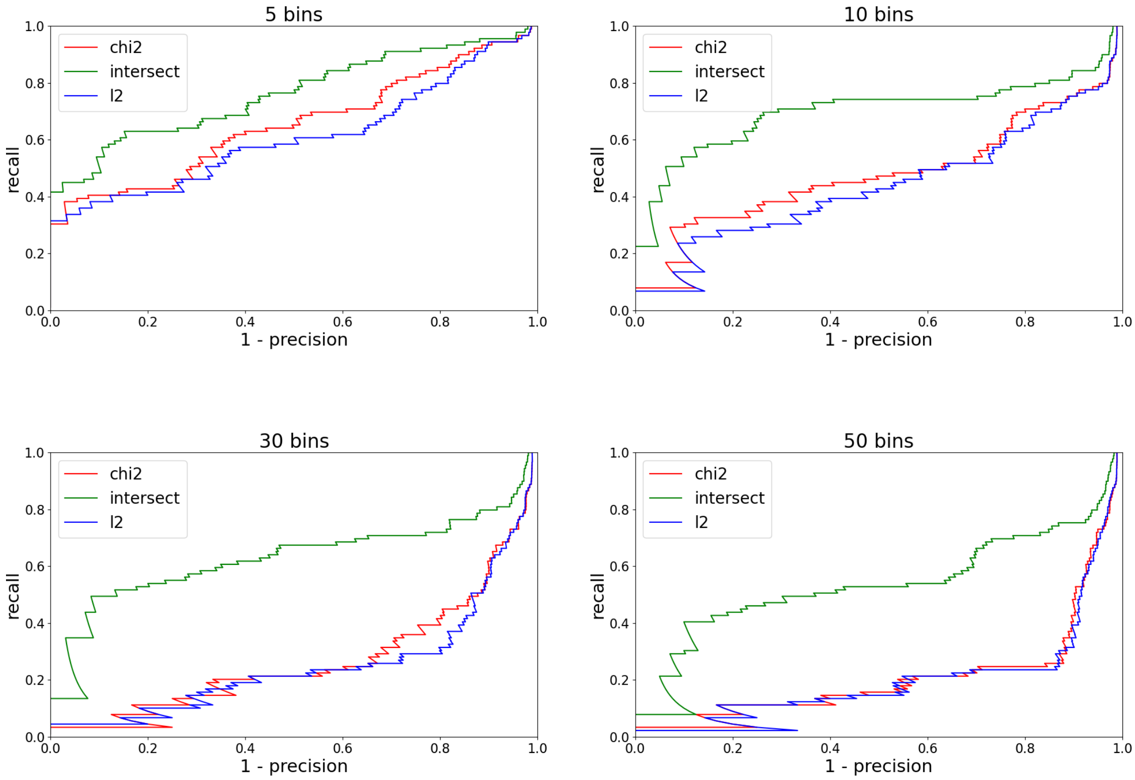

Figure 4: RPC for RGB Histograms

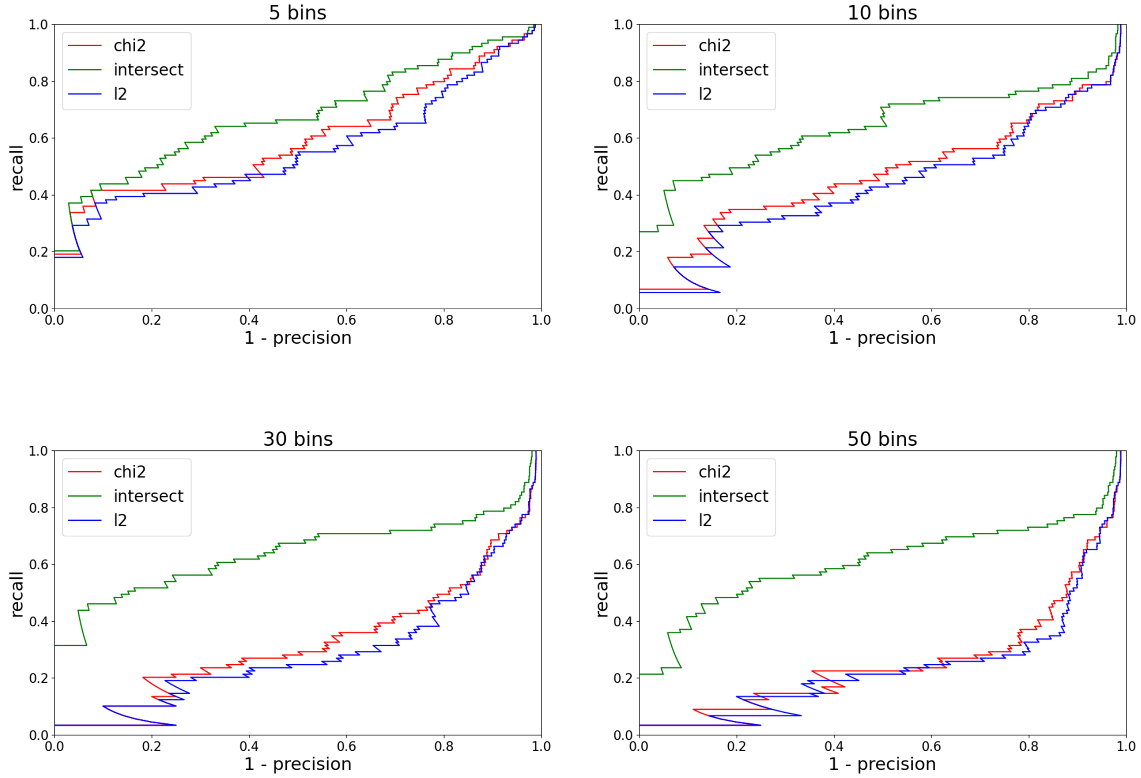

Figure 5: RPC for RG Histograms

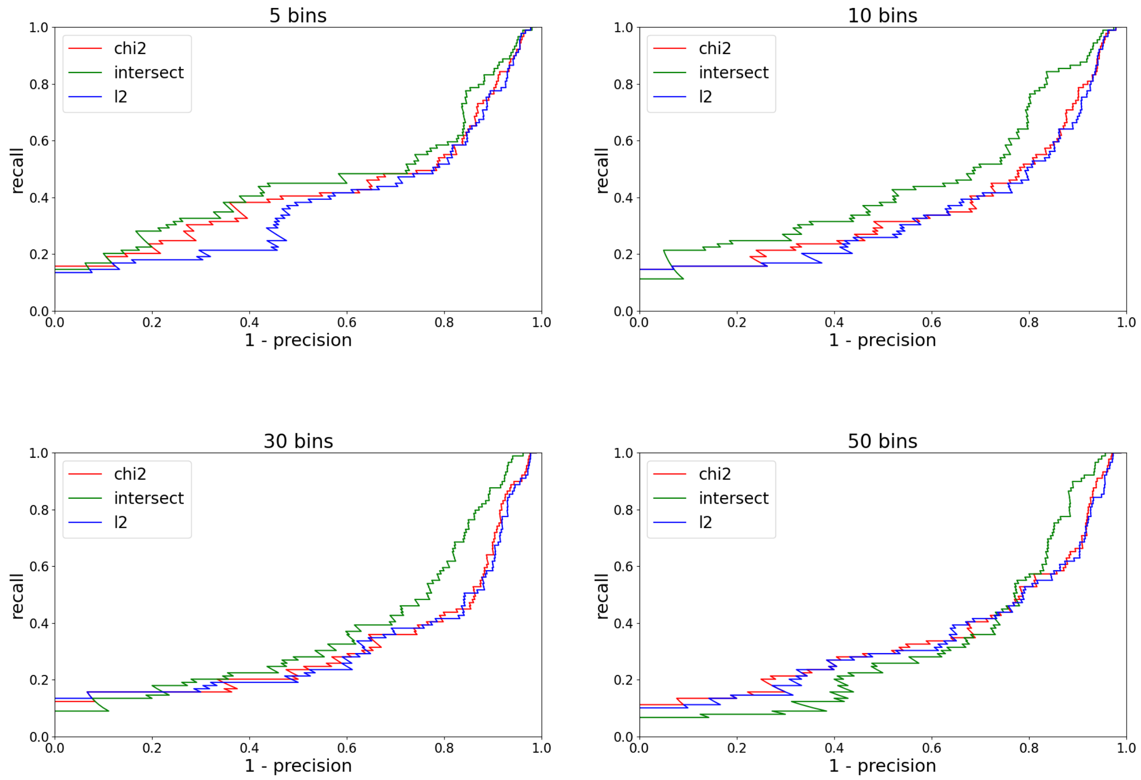

Figure 6: RPC for DxDy Histograms