Advanced Machine Learning
Image Filtering and Object Identification

Katsiaryna Zavadskaya (1847985)   Theodoros Sofianos (1867968)
Mousaalreza Dastmard (1852433)   Hamed Asadollahi (1717717)

March 2020

1 Image Filtering

The aim of this section is to discover the effect of applying different combinations of filters to a test image. The list of the filters to be examined is:

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

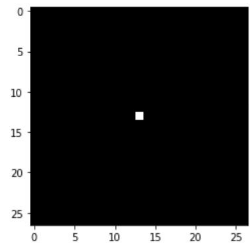

Figure 1: Test image

Where Gx is 1D Gaussian filter in x direction, Dx – Gaussian 1D derivative filter, and Gx^T and Dx^T are transposed Gx and Dx vectors, respectively. While, in fact, $Gx^T = Gy$ and $Dx^T = Dy$.

The result of applying these combinations of filters can be seen on Figure 2.

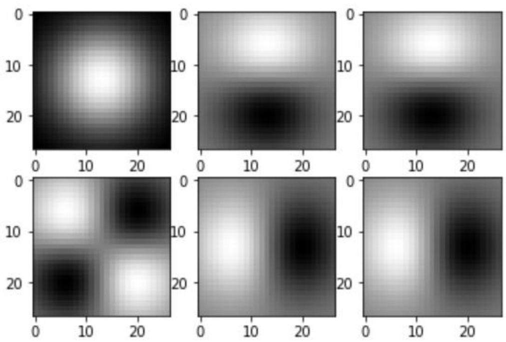

Figure 2: The result of filters' application

Since, convolution is commutative and associative operation, the order of applying filters to the test image is not important.

In detail, the 1st image of figure 2 is showing sequential convolution of the test image firstly with Gx – horizontally and secondly, convolving the result with Gy – vertically. The process is performed by replacing each pixel of the original image by a linear combination of its neighbors. The result is smoothed test image along both x and y directions.

The 2nd and 3rd images are showing again that the order of operations is not playing any role. The aim of applying Dx^T – Gaussian derivative along the y axis is to find an edge which is perpendicular to gradient direction. That is why we can see a horizontal smoothed edge on both of the images.

The 4th image shows application of derivative filter in both x and y directions, transforming central non-zero pixel of the test image into two edges. The differentiation inherits associative property from convolution, which means that also the order of Dx and Dx^T is not changing the result.

In the 5th and 6th images we see that the result is very similar to 2nd and 3rd. The only change is the direction of the derivative, which in this case is computed along the x axis – the edge found is vertical and perpendicular to x.

2 Object Identification

The aim of this section is to experiment with different distance and histogram functions to find out which combination is giving the best performance for our data in terms of the recognition rate. Where recognition rate is a ratio between number of correct matches and total number of query images. The results of our grid search is presented in the table 2. Because of computational resources available, the total number of query images is equal to 30, since we used only a sample of 30 images instead of full set of query images. In order to set up experiment we used next variations:

- distance = [intersect, l2, chi2]
- histogram = [grayvalue, rg, rgb, dxdy]
- number of bins = [5, 15, 30]

Our main performance evaluation measure is recognition rate (RR). When it comes to distance functions, we can see that **intersect** outperforms **l2** and **chi2** significantly with the best achieved RR = 0.90 versus 0.66 and 0.0, respectively. Since, **chi2** with all the combinations of histograms gives us RR = 0.0, we are going to asses efficiency of different histogram types by comparing them in relation with **intersect** and **l2** distances.

In the table 1 the best recognition rates are presented. The results were generalized for the number of bins, because we did not find any remarkable dependence between bins' number and RR.

Histogram	RR intersect	RR l2	Average RR
grayvalue	0.533333	0.400000	0.4666665
rg	0.866667	0.666667	0.766667
rgb	0.900000	0.666667	0.7833335
dxdy	0.633333	0.466667	0.55

Table 1: Evaluation of histograms

In the table 1 we added column with average recognition rate. Based on it, we declare that the two best histograms for our data are **rgb** and **rg** with only small difference in performance. While **dxdy** and **grayvalue** perform poorly.

Our best configuration is:

- distance = intersect
- histogram = rgb
- number of bins = 15.

3 Performance evaluation

In the last section we needed to evaluate the performance of our models based on Recall-Precision Curves. In order to do that we plotted three different distance functions: **chi2**, **intersect** and **l2** with respect to histogram types: **RG**, **RGB** and **dxdy**. The plots can be seen on the figure 3.

We are going to assess predictive performance of our models by looking at the Area Under each Curve. We expect AUC to be close to $\frac{1}{2}$ for a random classifier. As we can see on all three subfigures of figure 3, **chi2** is showing similar to random classifier behaviour.

The combination of **dxdy** histogram with **intersect** and **l2** distances is showing that these models' prediction abilities are even worse than a random classifier, which just means that **dxdy** is not a suitable image histogram for our type of data. Since **dxdy** is more useful for query/model images, which maintain the same shape, while our data consists of the front view of the object and its rotated view. We expect that for this type of data color histograms work better.

On the plots for **RG** and **RGB** histograms we see that combination of color histograms together with **intersect** and **l2** distances gives more correct predictions. Looking at AUC it seems that for both **RG** and **RGB** histograms the **intersect** is showing the best predictive ability, while **l2** is less accurate.

Finally, it is seen that the combination of **RGB** histogram and **intersect** distance has the best performance out of all presented systems. The same result was obtained in the second section of the report by evaluating system performance by computing recognition rate.

Distance	Histogram	# Bins	# Correct matches	Recognition rate
intersect	grayvalue	5	10	0.333333
		15	11	0.366667
		30	16	0.533333
	rg	5	22	0.733333
		15	26	0.866667
		30	21	0.700000
	rgb	5	26	0.866667
		15	27	0.900000
		30	24	0.800000
	dxdy	5	13	0.433333
		15	19	0.633333
		30	17	0.566667
l2	grayvalue	5	10	0.333333
		15	12	0.400000
		30	8	0.266667
	rg	5	20	0.666667
		15	18	0.600000
		30	10	0.333333
	rgb	5	20	0.666667
		15	17	0.566667
		30	8	0.266667
	dxdy	5	11	0.366667
		15	14	0.466667
		30	12	0.400000
chi2	grayvalue	5	0	0.000000
		15	0	0.000000
		30	0	0.000000
	rg	5	0	0.000000
		15	0	0.000000
		30	0	0.000000
	rgb	5	0	0.000000
		15	0	0.000000
		30	0	0.000000
	dxdy	5	0	0.000000
		15	0	0.000000
		30	0	0.000000

Table 2: The recognition rate for the combinations of distance and histogram functions

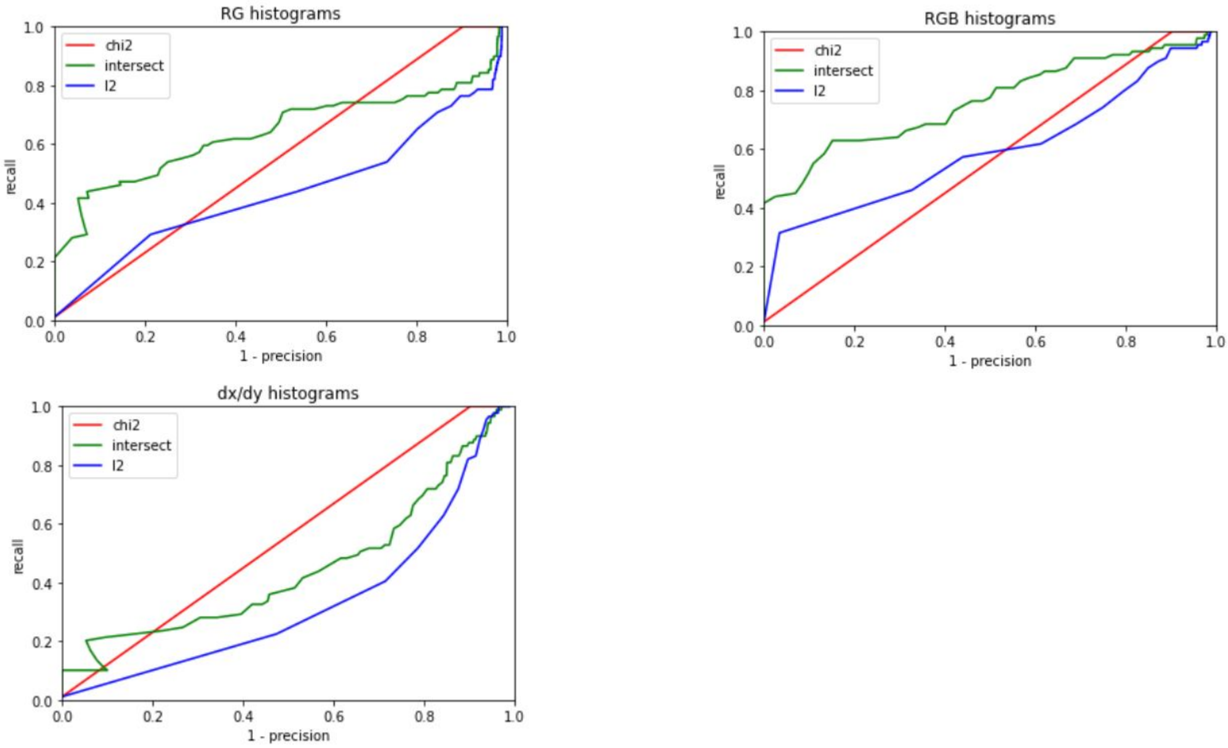

Figure 3: RPC curves