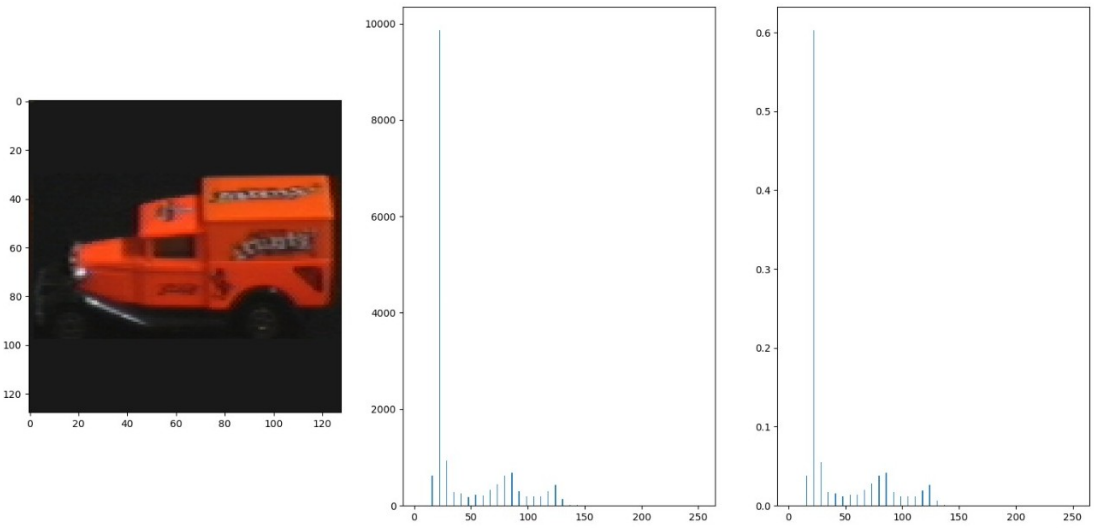
Report_Q2_Q3

November 5, 2020

1 Question 2

1.1 Question 2.a

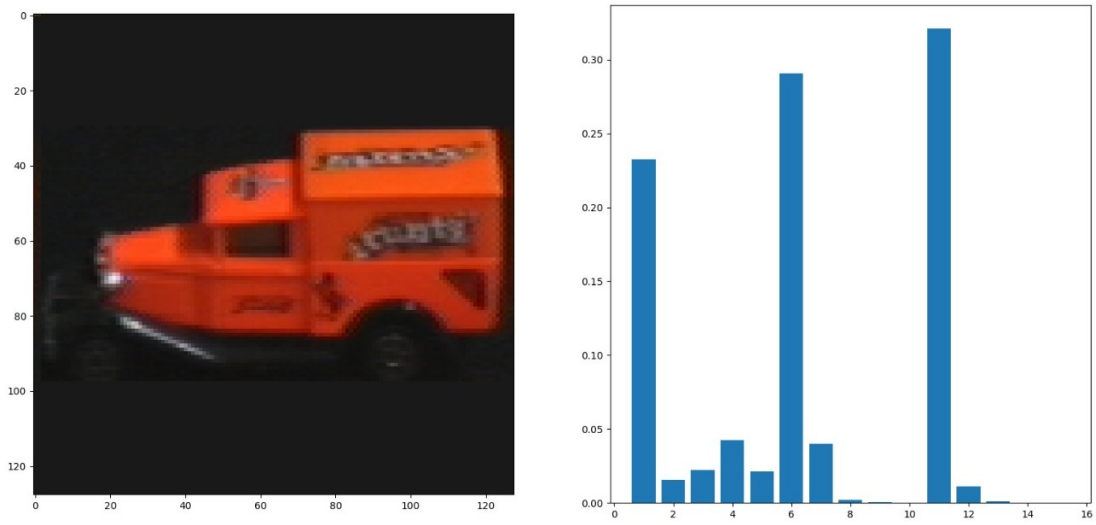
1.1.1 Gray-value histograms



Comment The first histogram is generated by the buit-in np.histogram function while the second one has been generated with our normalized_hist function. The greatest difference between the two plots is the scale, since the second is normalized while the first one is not. Overall, we noticed that the two histrograms are very similar. Also, in both histograms the highest peak is reached around 25, which is consistent with the fact that the black color in the picture is encoded by the triplet (25,25,25).

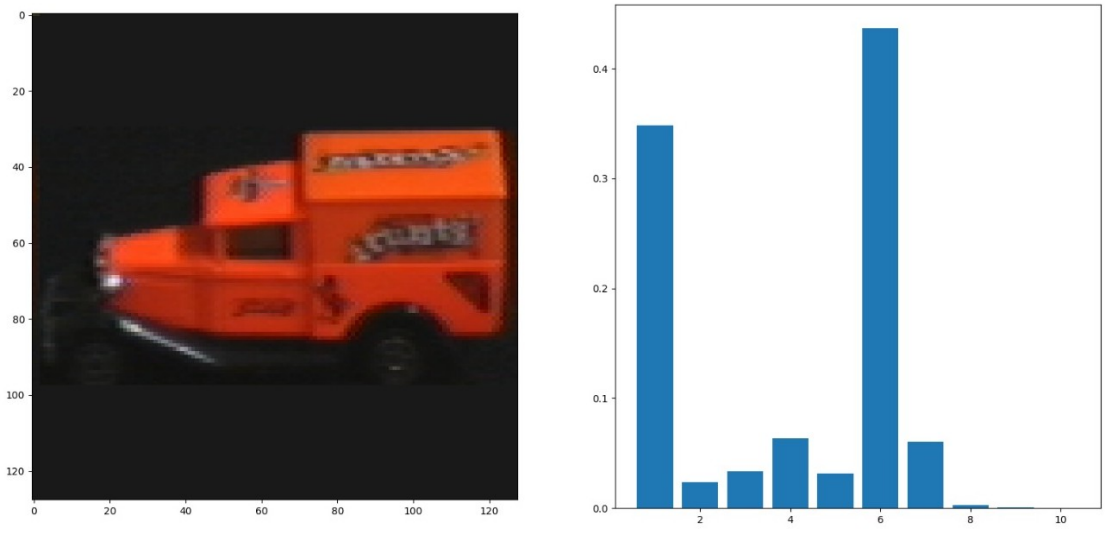
1.2 Question 2.b

1.2.1 RGB histograms



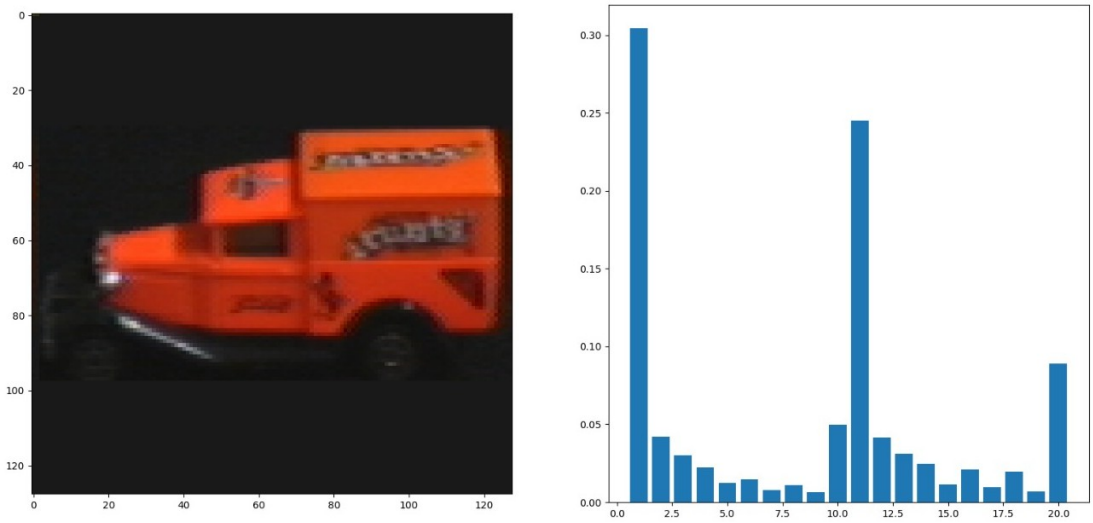
Comment In this histrogram the first 5 bins represent the R channel, the second group of 5 represent the G channel and the last group represent the B channel. / Also in this case the highest peaks are reached for the bins that include value 25. We can appreciate that the last 3 bins for each color are close to 0 for G and B channels while they are above 0.1 for the R channel, as one would expect by looking at the image.

1.2.2 RG histograms



Comment In this histrogram the first group of 5 bins represent the R channel and the last one represent the G channel. We can consider this plot as a subplot of the previous one, where the B channel has been ignored.

1.2.3 dx dy histograms

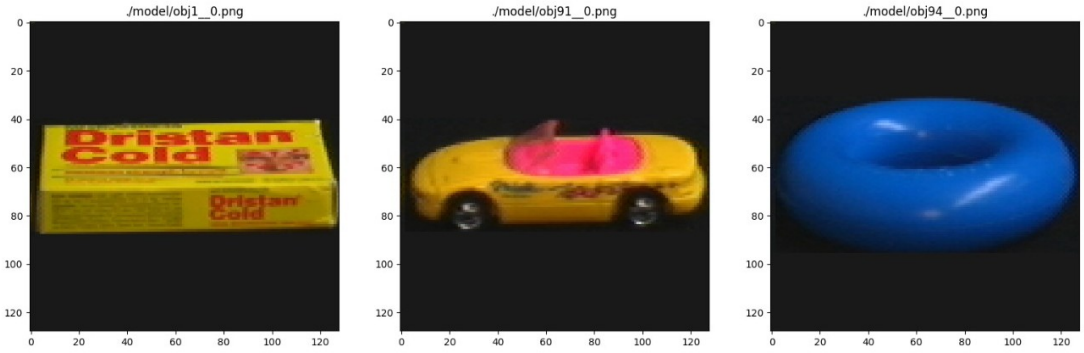


Comment In this histogram the first 10 bins represent the image produced by the derivative along the x axis and the last 10 bins respresent the image produced by the derivative along th y axis. From the fact that the two histograms are very similargraph, we can infer that neither vertical or horizontal edges preveal significantly. Moreover, we can see that most of the edges for both dx and dy are represented in the first bin, which suggests that most of the edges are smoothed.

1.3 Question 2.c

1.3.1 Distance functions

Distances between images 1 and 2 and between images 1 and 3



	grayvalue	rgb	rg	dxdy
l2	0.003322	0.004034	0.004571	0.000715
intersect	0.865906	0.848267	0.835602	0.933848
chi2	0.043809	0.049665	0.058704	0.005386

	grayvalue	rgb	rg	dxdy
l2	0.017315	0.018910	0.025211	0.004272
intersect	0.636292	0.658040	0.678467	0.866942
chi2	0.237930	0.264189	0.295165	0.025352

Comment The distances between the first and the second image and between the first and the third image have been calculated by considering the four types of historams and the three types of distances. From what we can see in the tables all three distance values indicate that the first two images are more similar then the the first and the last one. However not all histrograms have been able to detect the strong difference between the first and the last image, as we can see especially in the dxdy values.

1.4 Question 3.b

1.4.1 Visualize nearest neighbors

Comment Two of the three query images have identified theirselves as the best match. In the case of the Obj_100_40.png we can also notice that it also returned similar objects to itself.
Obj12_40.png did not find a good match with itself coherently with what we expected from the similarity index (0.726) that is lower than those for the other two images.
For the Obj17_40.png we have a correct best match and also the other objects that have been returned are similar among therselves.

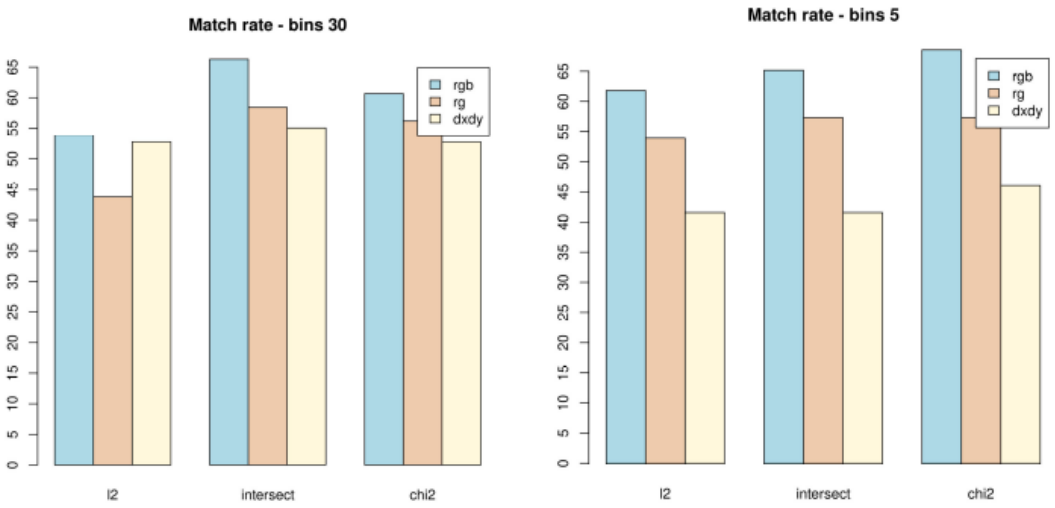
1.5 Question 3.c

1.5.1 Comparison amond distance types and histogram types

We iterate through the distances list and type of histograms list in order to obtain the recognition rate and we test this results with different numbers of bins.

hist_types : rgb, rg and dxdy
distance_types : l2, intersect and chi2
num_bins : bins50 , bins40 , bins30 , bins20 , bins10 , bins5 and bins2

[21]:
```
def test(num_bins):
    results=[]
    for dist in distance_types:
        correct=[]
        for hist in hist_types:
            [best_match, D] = match_module.find_best_match(model_images,␣
↪query_images, dist, hist, num_bins)
            num_correct=0
            for i in range(len(query_images)):
                    if best_match[i]==i:
                            num_correct+=1
            correct.append(100* num_correct / len(query_images))
        results.append(correct)
    return results
```



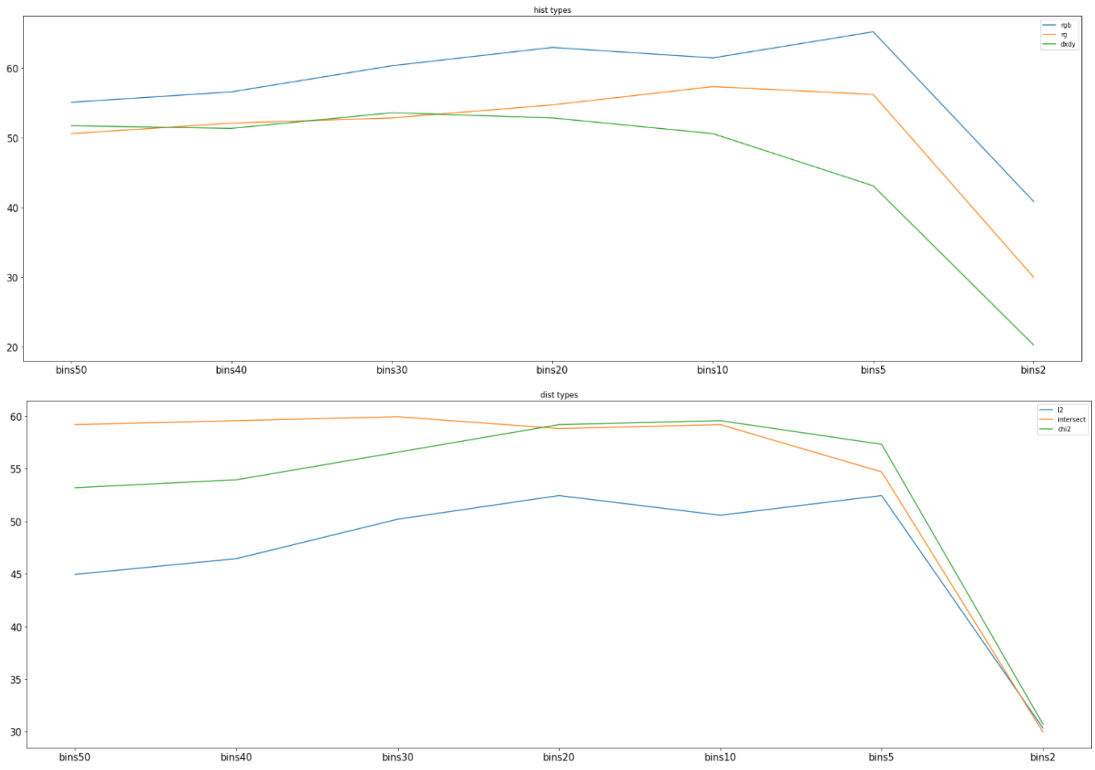
Comment In the figures above we can appreciate the recognition rate for each combination of histogram and distance types with number of bins equal to 30 (the one we used in question 3b) and number of bins equal to 5 (used in question 2b).

As we can notice, the combination intersect-RGB and intersect-RG are the most robust since they do not vary much with the variation in the number of bins. On the other hand, chi2 and l2 distances result to be much more sensitive to the number of bins parameter.

In the histogram representing num_bins 30 the worst result is obtained with the l2-RG combination, while the best one with the intersect-RGB combination.

In the histogram representing num_bins 5 we see that the the three methods to calculate distances agree on the hierachy of precision of the three types of histograms, with RGB being the most precise and dxdy being the less precise. In this case we see that the best result is obtained for the combination chi2-RGB.

```
[ ]: for l in bins:
         hist=[]
         dist=[]
         for i in range(3):
             hist.append(mean([l[0][i],l[1][i],l[2][i]]))
             dist.append(mean(l[i]))
         hists.append(hist)
         dists.append(dist)
```



Comment In order to evaluate the behaviours of single histogram and distance types, we computed the mean values of distances for each histogram type and mean values of histograms for each distance type respectively. Consequently we plot the obtained results whith descreasing values of bins.

In the first plot we can appreciate how the rate of recognition varies with decreasing number of bins for the three types of histogram. In particular, the rgb and rg histogram reach their peak when number of bins is equal to 5. The dxdy histogram, on the other hand, remains stable up to bins10 and then decreases.

In the second plot we can appreciate how the rate of recognition varies with decreasing number of bins for the three types of distances. We can see how the intersect distance remains stable when bins is greater than 10; instead performances of both distances l2 and chi_squared increase with smaller values of bins, reaching their maximum at bins_5.

To conclude, our analysis would suggest that the intersect distance perform better with high number of bins (>20), while the chi square distance performs better for small number of bins. For all types of distances the RGB histogram is the one that produces the best results.