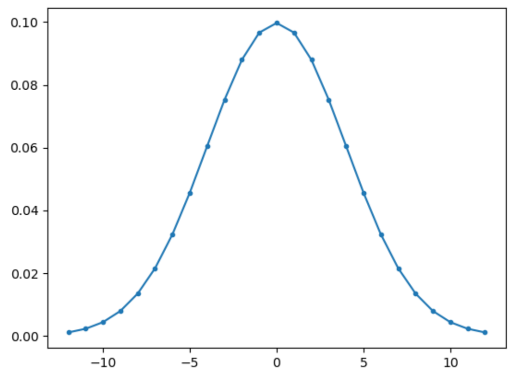
High-Level Computer Vision
Summer Semester 2019

Report 1: Image Filtering and Object Identification

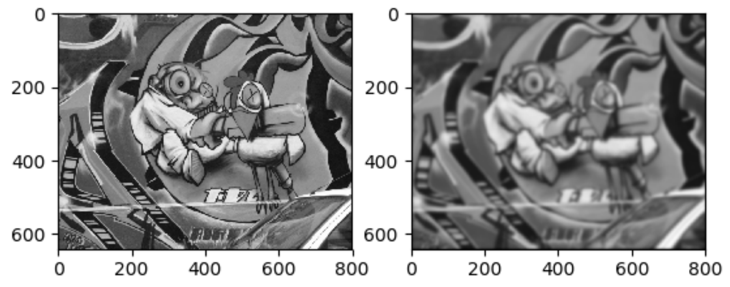
Group Members:
Chirag Bhuvaneshwara - 2571703 - chiragbhuvaneshwar@gmail.com
Jyotsna Singh - 2576744 - s8jysing@stud.uni-saarland.de
Jyothsna Shashikumar Sastry - 2571729 - s8jysast@stud.uni-saarland.de

Question 1: Image Filtering

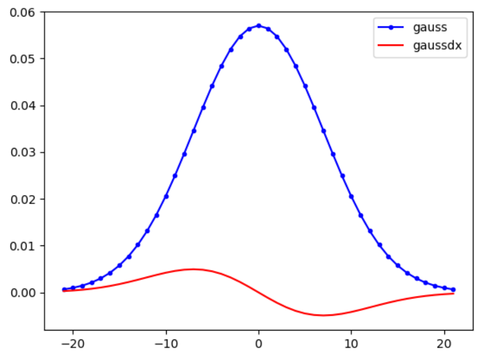
Gaussian distribution



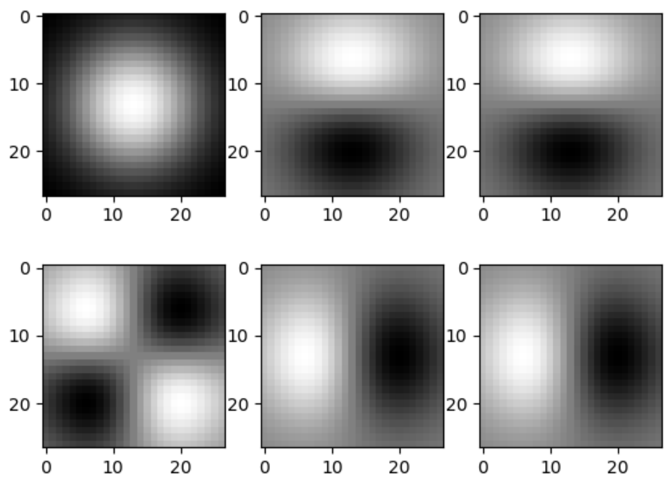
Gaussian Convolution



Gaussian Derivative



Filter Combinations



1c)

1. first G, then G'. => applying Gaussian convolution with Gaussian kernel of same variance twice.

2. first G, then D'. => applying Gaussian convolution twice and taking its derivative in x direction.

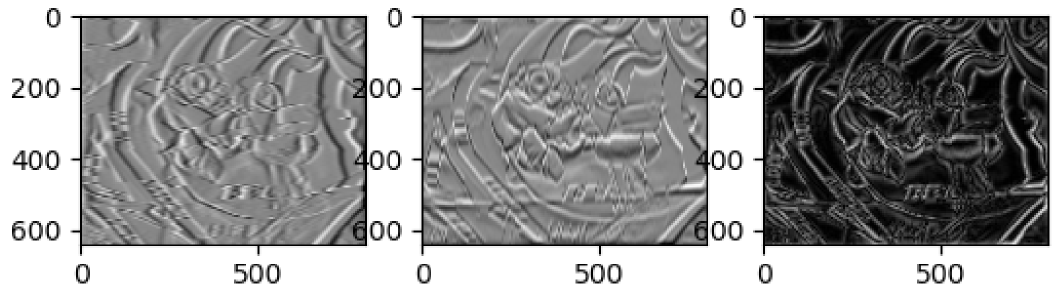
3. first D, then G'. => applying Gaussian convolution twice and taking its derivative in x direction.

4. first D', then D. => applying Gaussian convolution twice and taking its derivative in both x and y directions

5. first G', then D. => applying Gaussian convolution twice and taking its derivative in y direction

6. first D', then G. => applying Gaussian convolution twice and taking its derivative in y direction

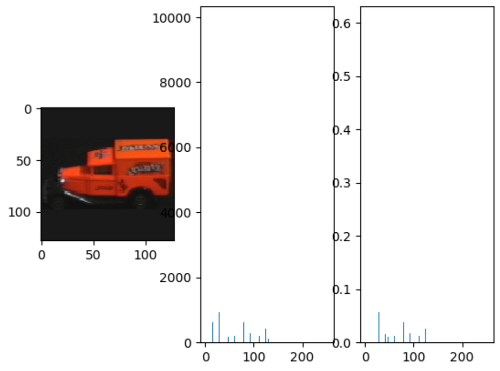
2D Gaussian derivatives in x and y directions

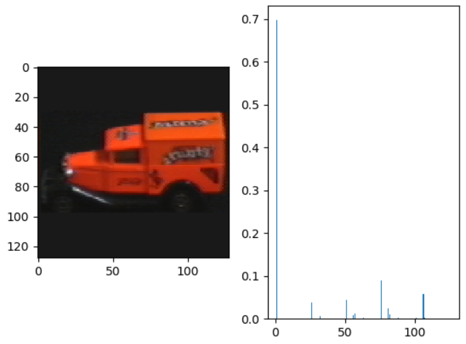


1d) The first plot above shows Gaussian derivative of the image in the X direction which only captures the difference in the gray values in the X direction of the input image.
Similarly, the second image shows the Gaussian derivative of the image in the Y direction. Lastly, the 3rd image shows the gradient magnitude of the image which captures both the X and Y derivative information and is therefore rotationally invariant.

Question 2: Image Representations, Histogram Distances

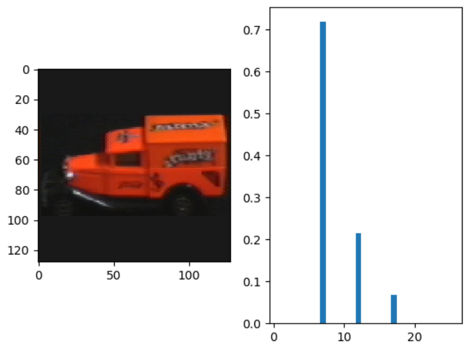
Normalized histogram **RGB histogram**

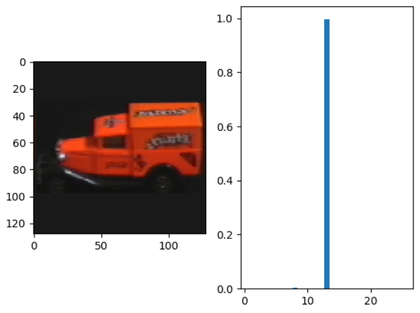




RG histogram

dxdy histogram





Question 3: Object Identification

3a)

The function **find_best _match** computes the **distance**(given by dist_type) between every pair of model and query image **histograms**(given by hist_type) in the **D** matrix.

The model image with minimum distance to the query images is returned as the **best_match** for the query image.

3b)

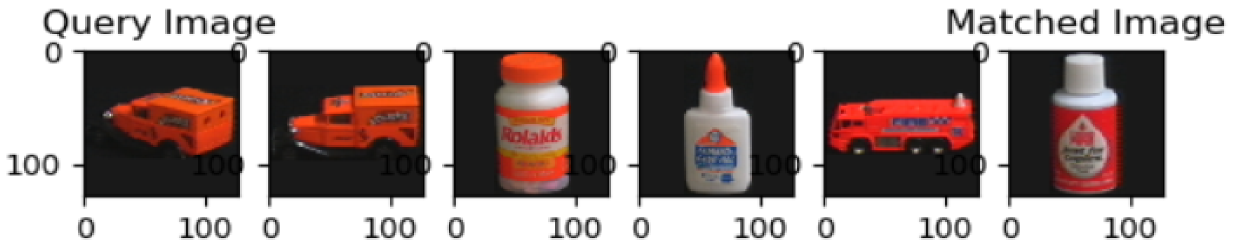
The **show_neighbours** function finds 5 model images with the smallest distance from the query image

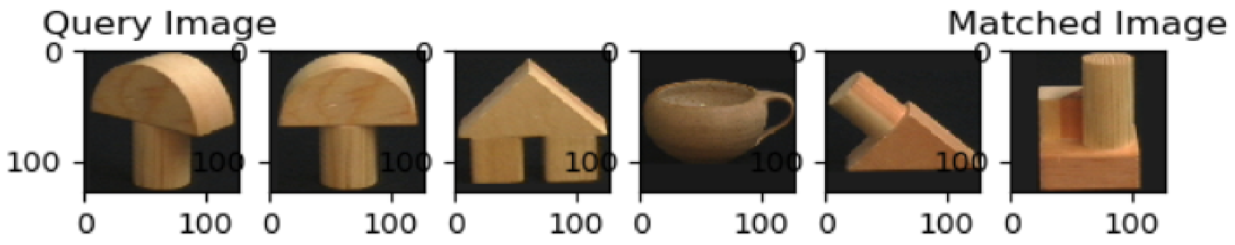
dist_type = 'intersect'

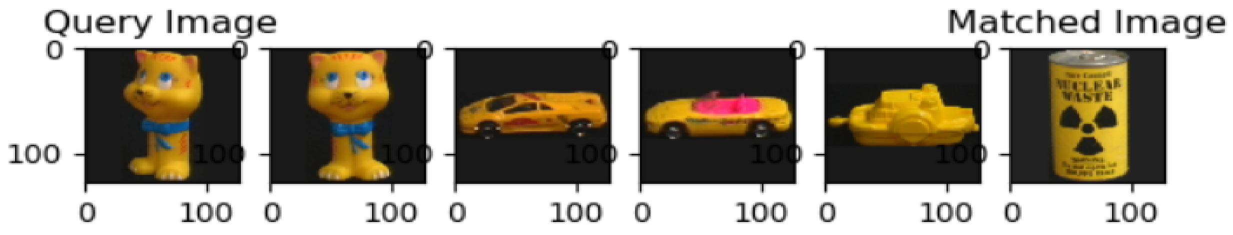
hist_type = 'rg'

num_bins = 30

The above combination of distance, histogram functions and bin size yields the following output







3c)
The **recognition_rate** and experiment results

chi2				
Bins	**grayvalue**	**rgb**	**rg**	**dxdy**
30	0.54	0.80	0.94	0.65
60	0.53	0.79	0.92	0.69
90	0.48	0.73	0.91	0.70

intersect				
Bins	**grayvalue**	**rgb**	**rg**	**dxdy**
30	0.50	0.82	0.88	0.61
60	0.51	0.78	0.89	0.65
90	0.44	0.71	0.84	0.68

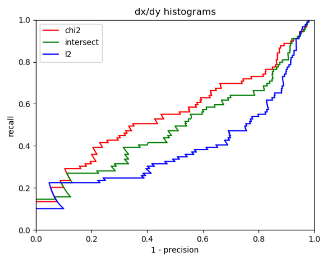
l2				
Bins	**grayvalue**	**rgb**	**rg**	**dxdy**
30	0.40	0.40	0.72	0.47
60	0.34	0.32	0.65	0.52
90	0.29	0.26	0.53	0.51

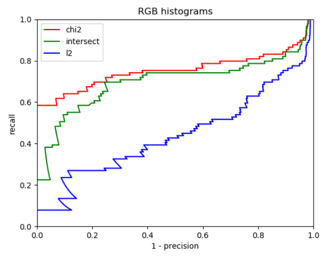
From the above tables, it is evident that the combination that yields the best results is: distance type chi2 for rg histograms with bin size in the range 30 - 60

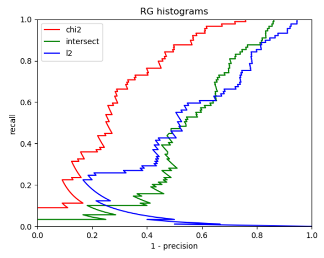
Question 4: Performance Evaluation

4b)

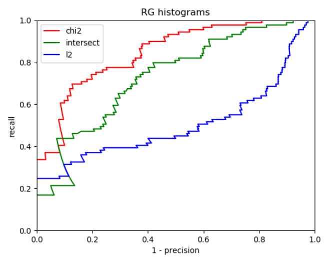
Bin Size: 20

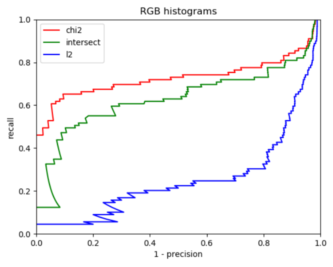


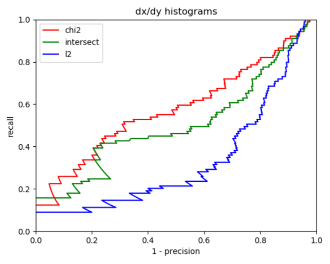




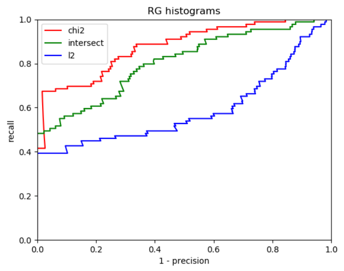
Bin Size: 40

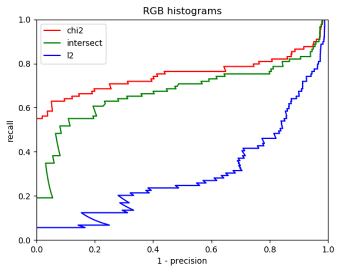


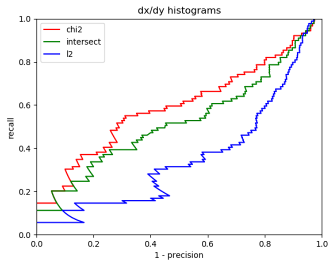




Bin Size: 60







For an ideal RPC curve - the area under it should be 1.0. We always chose the better curve as the one with more area under it. From the plots we can see that for Bin Size of 60, distance type chi2 and histogram type RG we get the best curve.

Out of histogram types - we see that dx/dy histograms in general provide poor results and RG works well with bigger bin sizes.

Distance type chi2 always outperforms intersect followed by l2 distance type in all cases.