

강좌_1



강좌_2

img_show.py



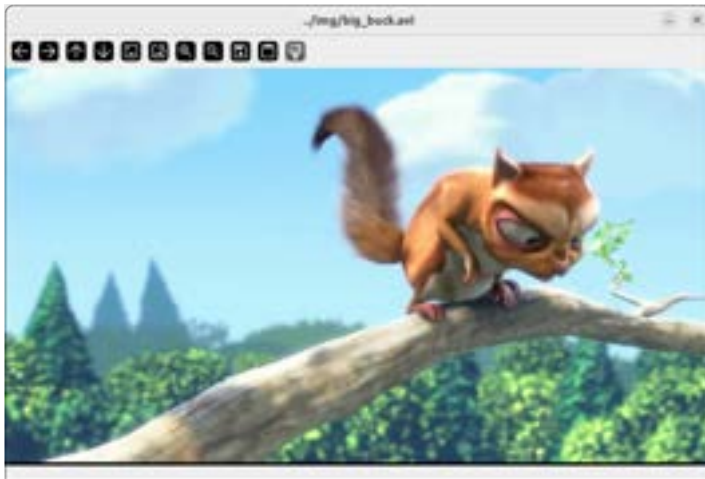
img_show_gray.py



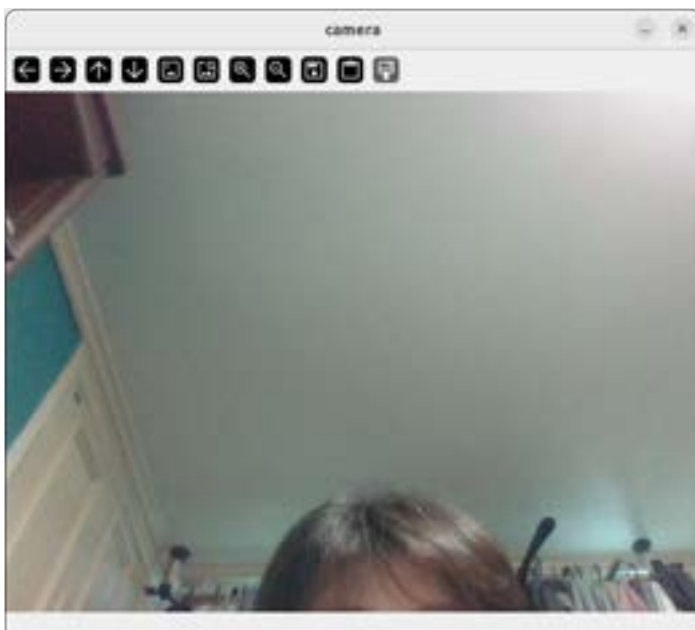
img_write.py



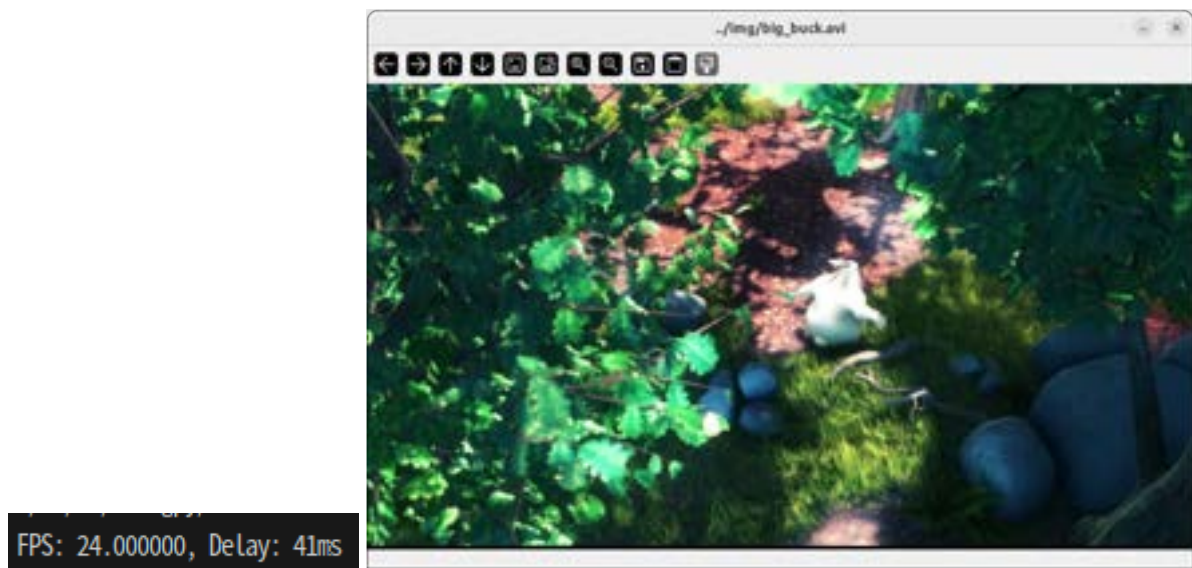
video_play.py



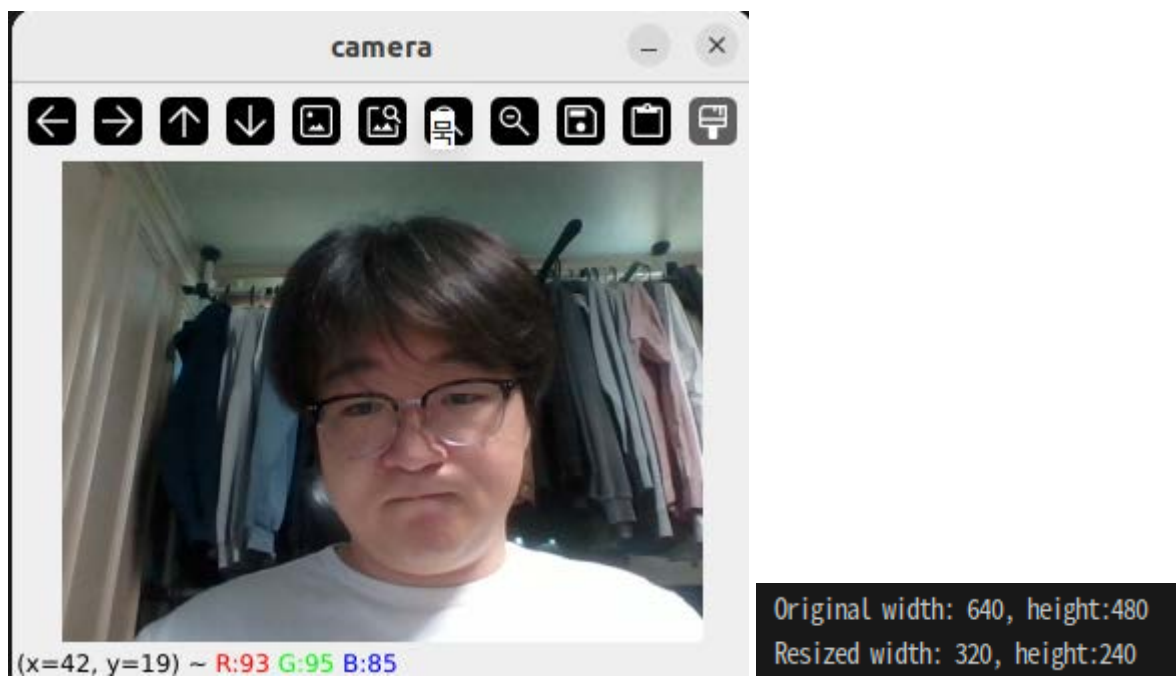
video_cam.py



video_play_fps.py



video_cam_resize.py



video_cam_take_pic.py



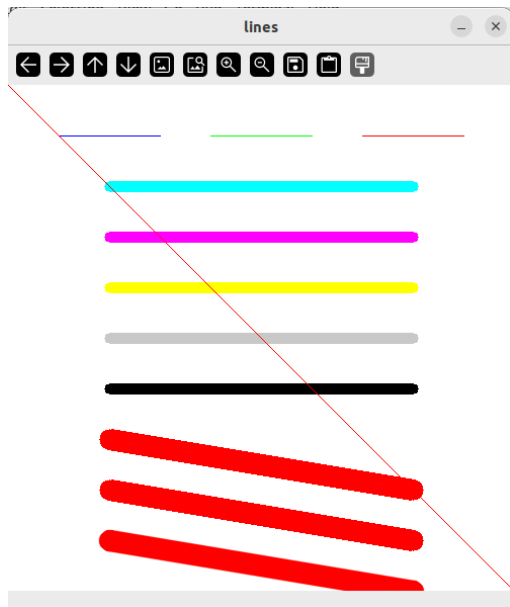
video_cam_rec.py



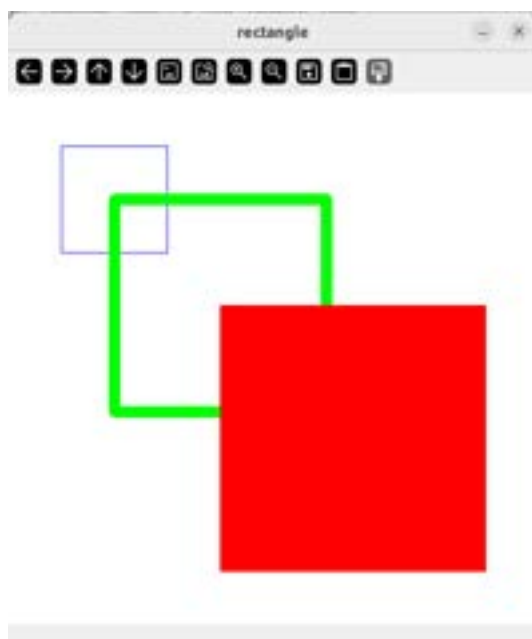
blank_img.py

blank_500.
jpg

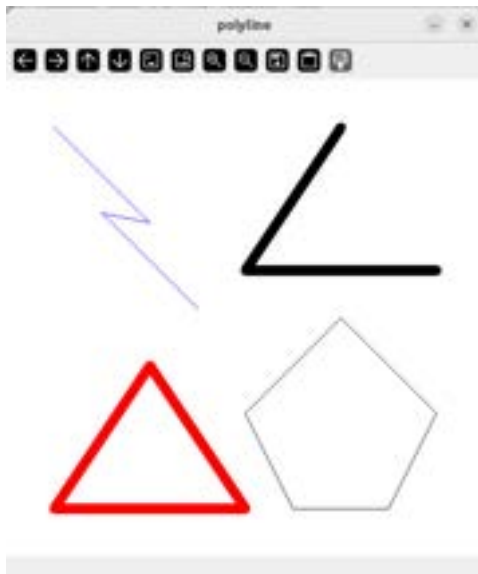
line.py



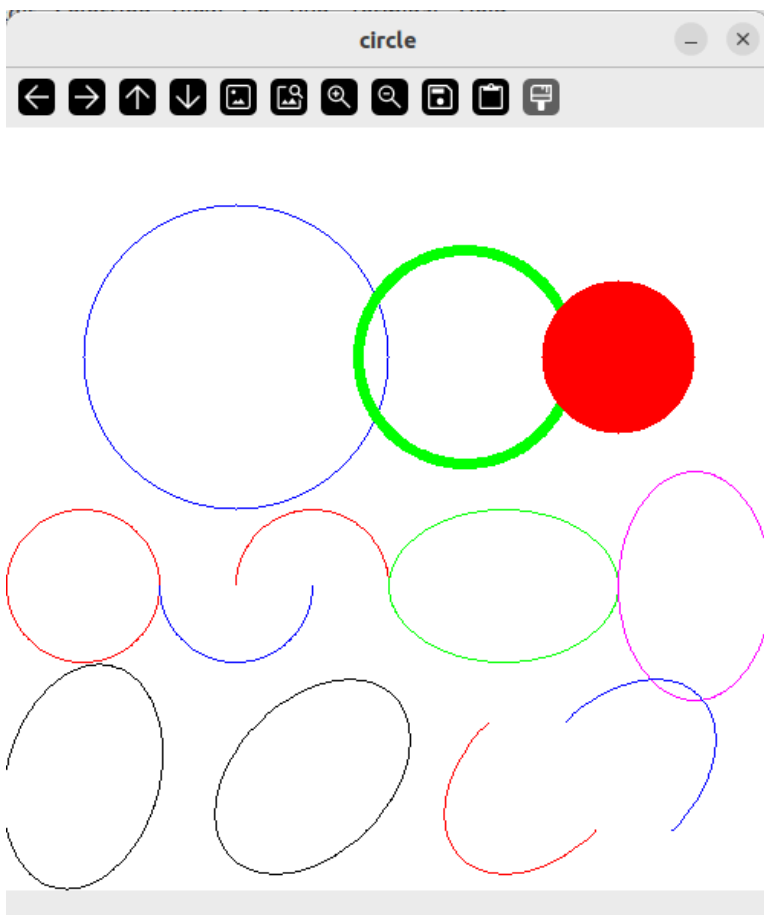
draw_rect.py



draw_poly.py



draw_circle.py



draw_text.py



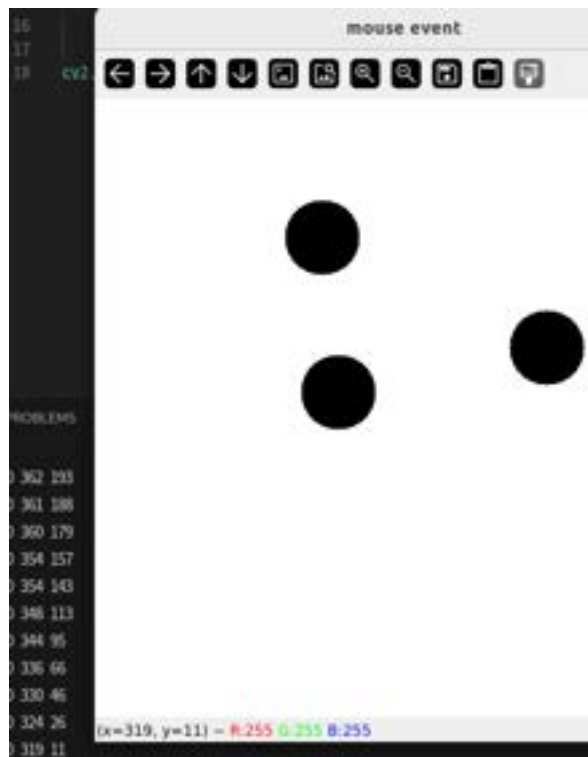
win.py



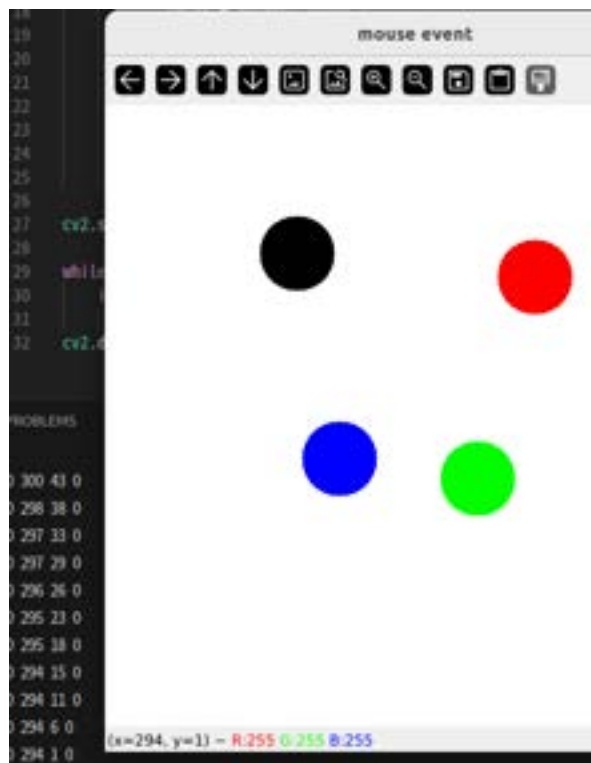
event_key.py



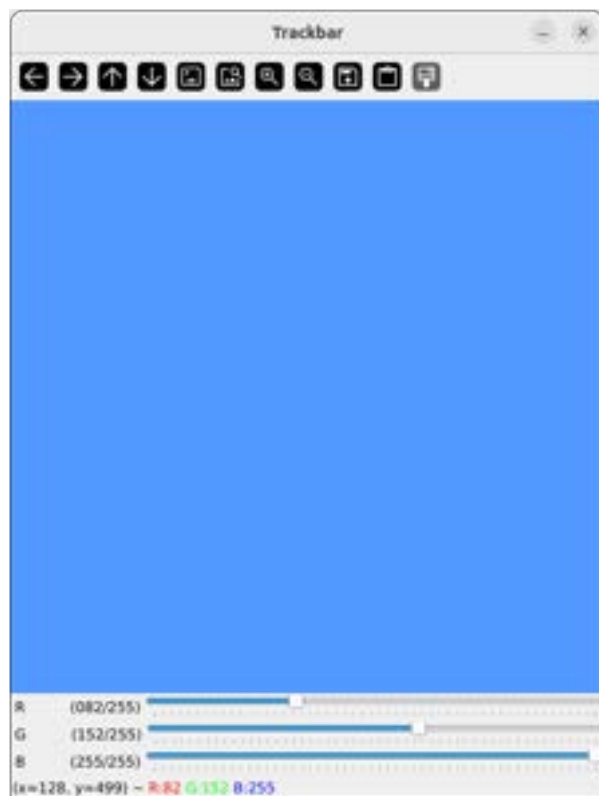
event_mouse_circle.py



event_mouse_circle_flag.py



event_trackbar.py



강좌_3

np_img.py

```
<class 'numpy.ndarray'>  
3  
(500, 500, 3)  
750000  
uint8  
1
```

np_create_array.py

```
[1 2 3 4] int64 (4,)  
[[1 2 3 4]  
 [5 6 7 8]] int64 (2, 4)  
[1.  2.  3.14 4. ] float64 (4,)  
[1. 2. 3. 4.] float64 (4,)
```

np_create_size.py

```
[[6.13267757e-310 6.13267757e-310 0.00000000e+000]  
 [0.00000000e+000 0.00000000e+000 0.00000000e+000]] float64 (2, 3)  
[[-21232 -31647 28900]  
 [ 0 -21232 -31647]] int16 (2, 3)  
[[0. 0. 0.]  
 [0. 0. 0.]] float64 (2, 3)  
[[1. 1. 1.]  
 [1. 1. 1.]] float32 (2, 3)  
[[[255 255 255 255]  
 [255 255 255 255]  
 [255 255 255 255]]  
  
 [[255 255 255 255]  
 [255 255 255 255]  
 [255 255 255 255]]] uint8 (2, 3, 4)
```

np_create_like.py

```
[[202 181 190]
 [202 181 190]
 [201 180 189]
 ...
 [199 178 187]
 [200 179 188]
 [200 179 188]]

[[201 180 189]
 [202 181 190]
 [201 180 189]
 ...
 [201 180 189]
 [202 181 190]
 [200 179 188]]] (293, 406, 3) uint8
```

np_create_arange.py

```
(5,)
int64
(3,)
float64
```

np_dtype.py

```
[0 1 2 3 4 5 6 7 8 9] int64
[0. 1. 2. 3. 4. 5. 6. 7. 8. 9.] float32
[0 1 2 3 4 5 6 7 8 9] uint8
[0. 1. 2. 3. 4. 5. 6. 7. 8. 9.] float64
```

np_reshape.py

```
[45 46 47 48 49]
[50 51 52 53 54]
[55 56 57 58 59]
[60 61 62 63 64]
[65 66 67 68 69]
[70 71 72 73 74]
[75 76 77 78 79]
[80 81 82 83 84]
[85 86 87 88 89]
[90 91 92 93 94]
[95 96 97 98 99]] (20, 5)
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23] (24,)
[[0 5]
 [1 6]
 [2 7]
 [3 8]
 [4 9]]
```

np_broadcast.py

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
34
35 import numpy as np
36 a = np.arange(10)
37 print(a)
38 b= a+1
39 print(b)
```

PROBLEMS OUTPUT DEBUG CONSOLE

```
h@h:~/insightbook.opencv_project_...
./debugpy/Launcher 33611 -- /home/h...
[0 1 2 3 4 5 6 7 8 9]
[ 1  2  3  4  5  6  7  8  9 10]
```

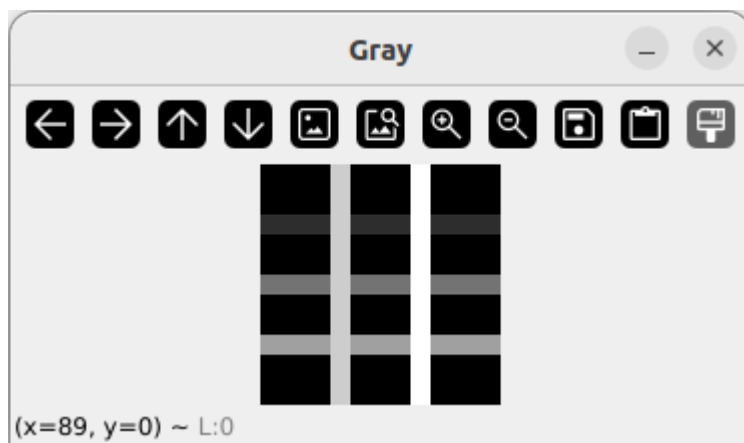
np_scalar_operation.py

```
[0 1 2 3 4]
[5 6 7 8 9]
[-2 -1  0  1  2]
[0 2 4 6 8]
[0.  0.5 1.  1.5 2. ]
[25 36 49 64 81]
[ True  True  True  True  True]
```

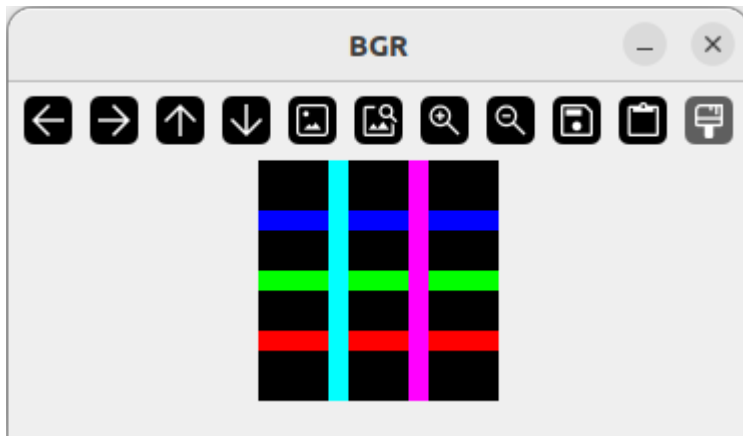
np_index.py

```
5
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

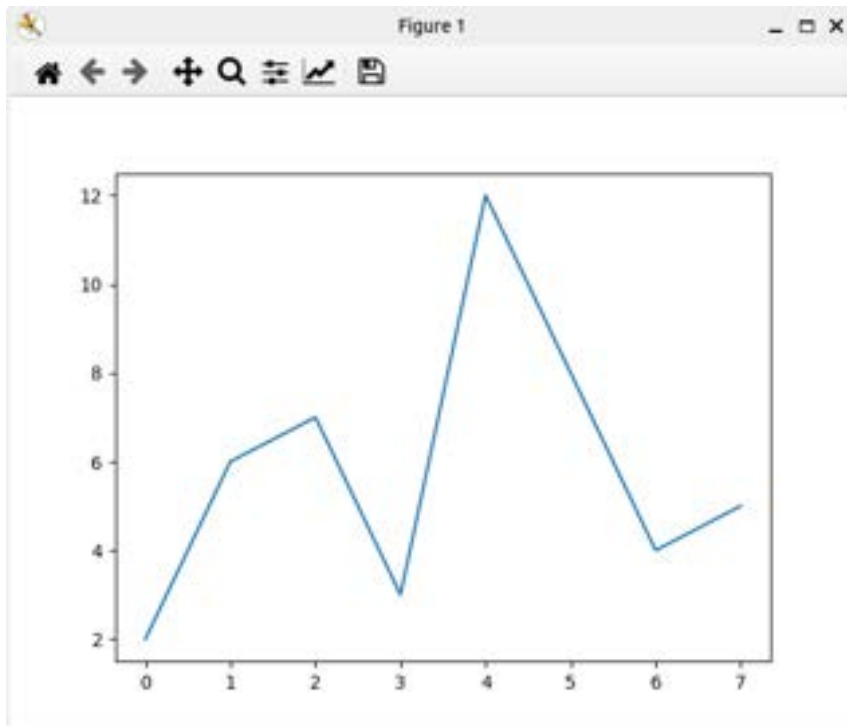
np_gray.py



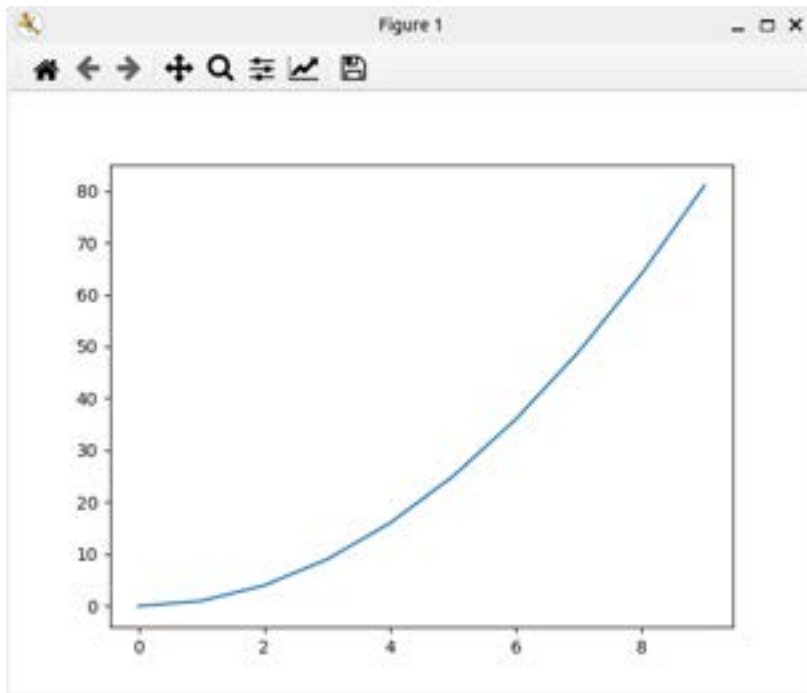
np_bgr.py



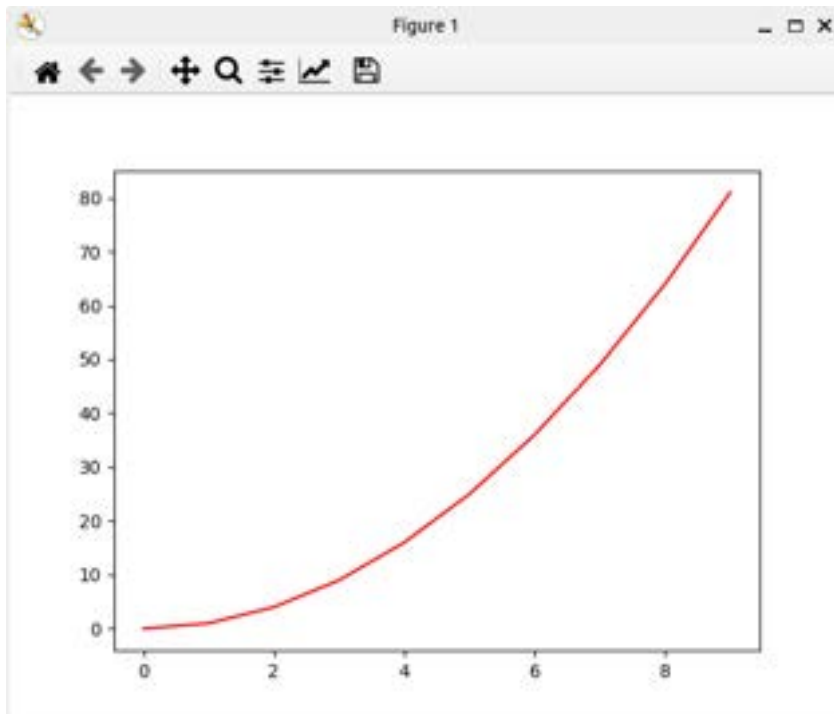
plt_plot.py



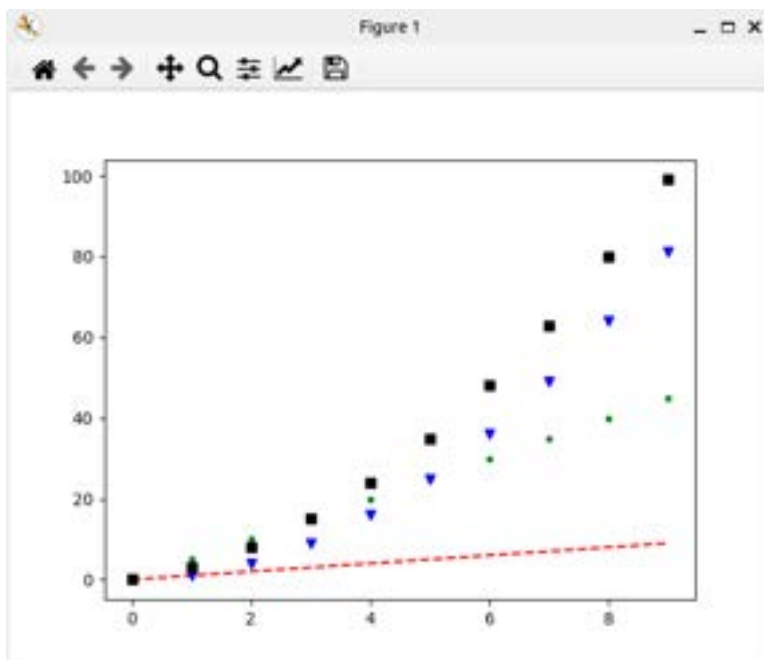
plt_simple.py



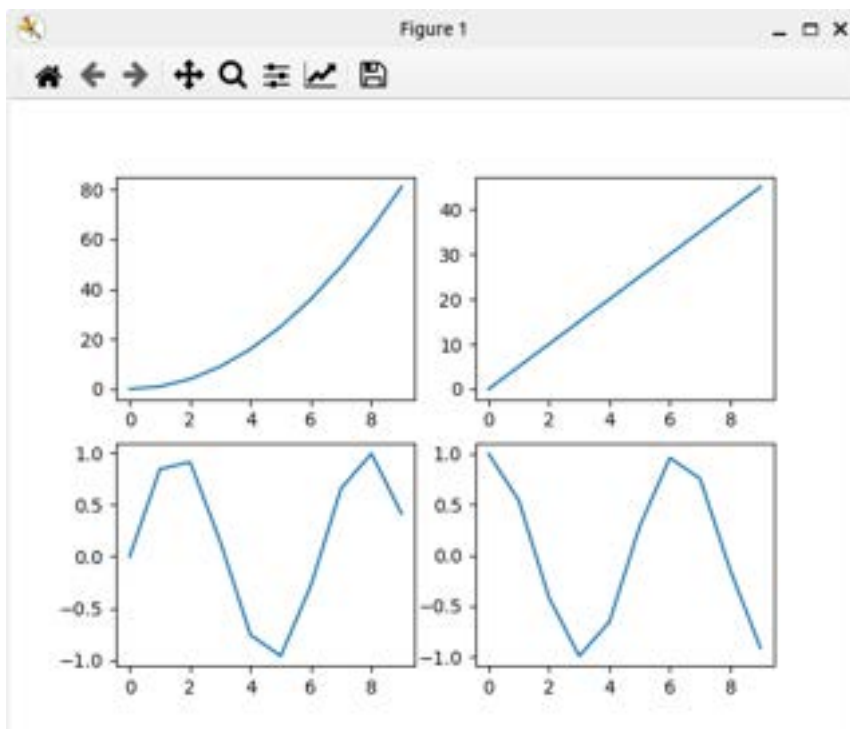
plt_color.py



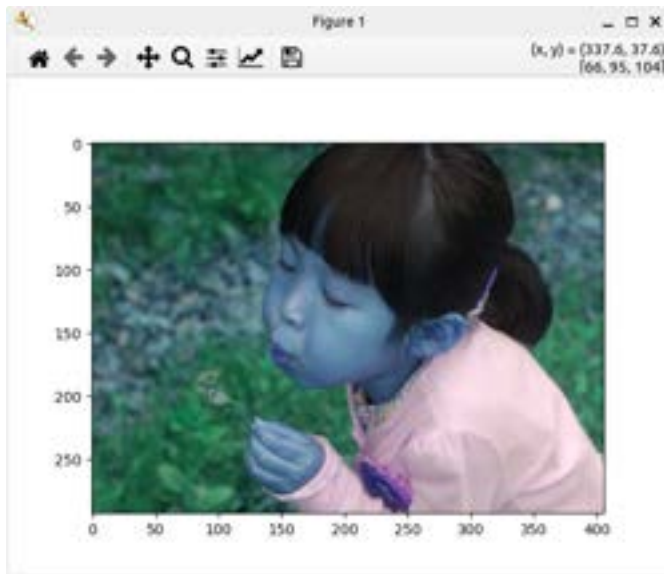
plt_style.py



plt_subplot.py



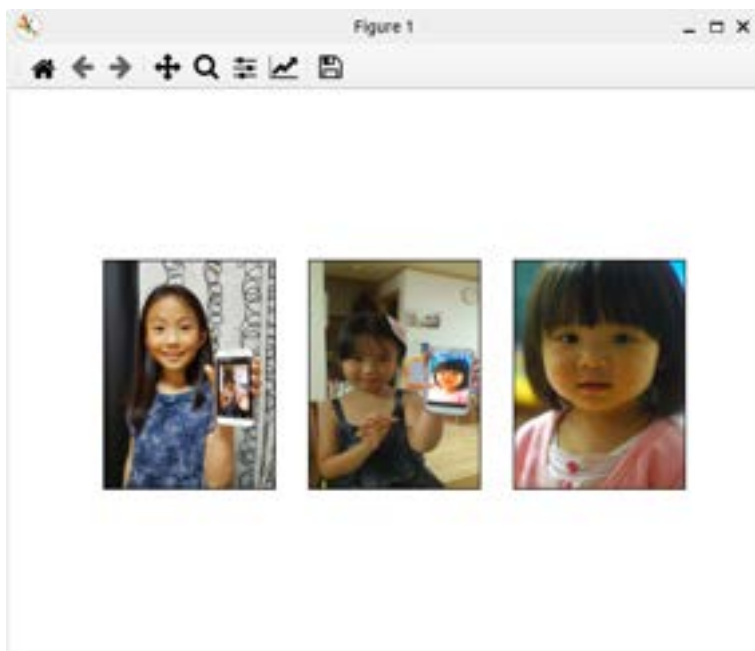
plt_imshow.py



plt_imgshow_rgb.py



plt_imshow_subplot.py

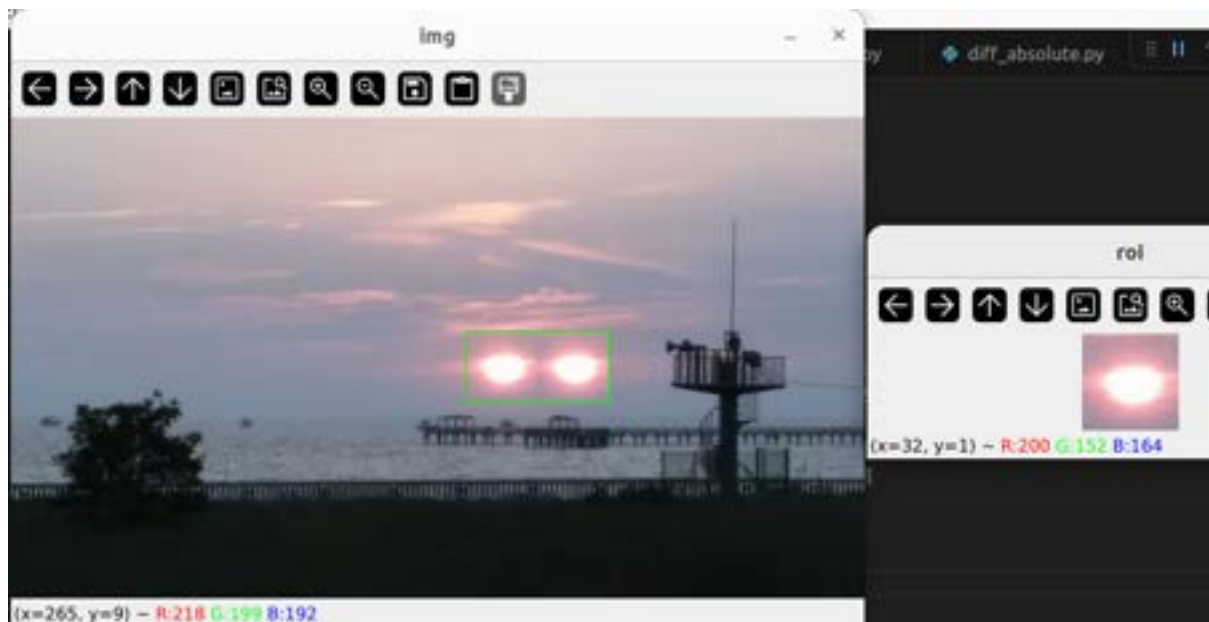


강좌_4

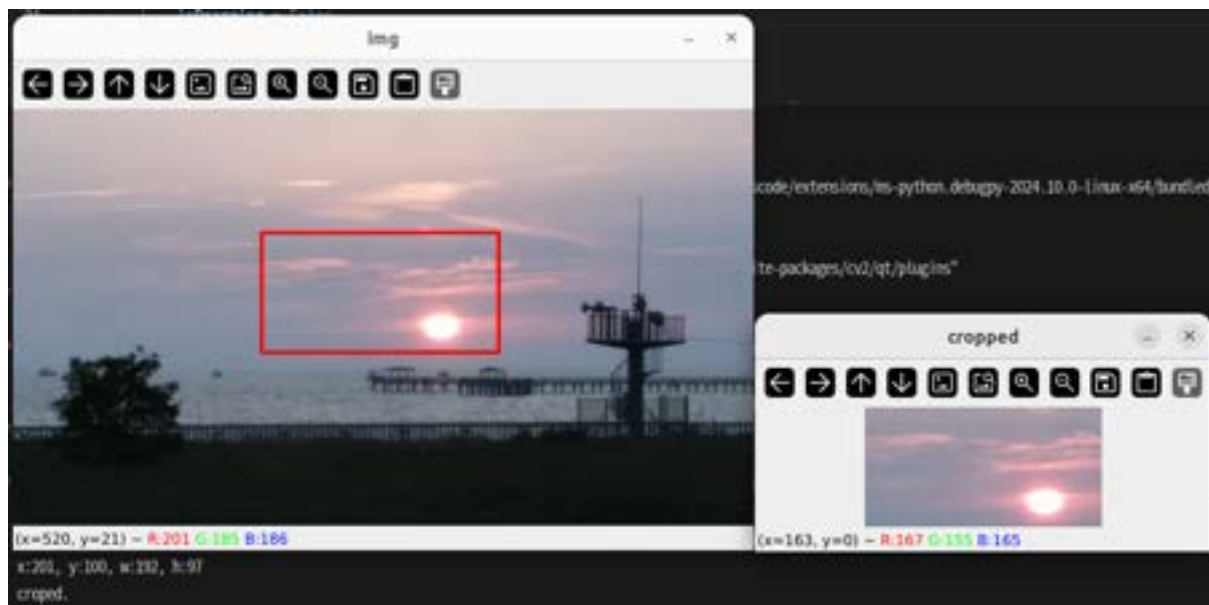
roi.py



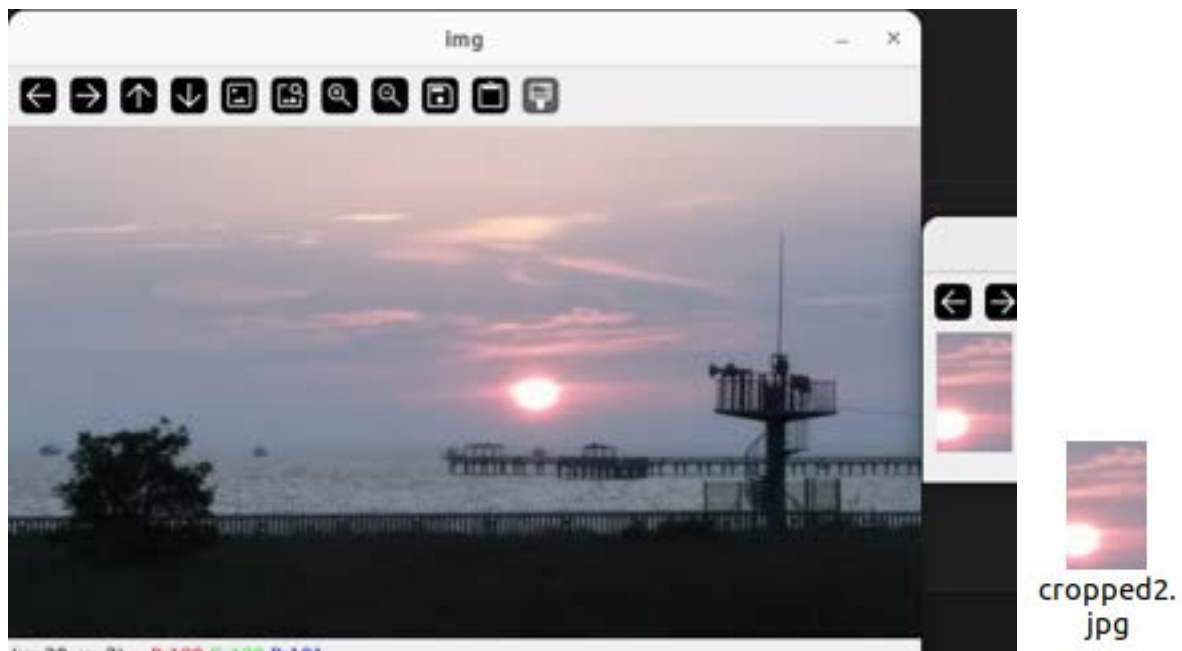
roi_copy.py



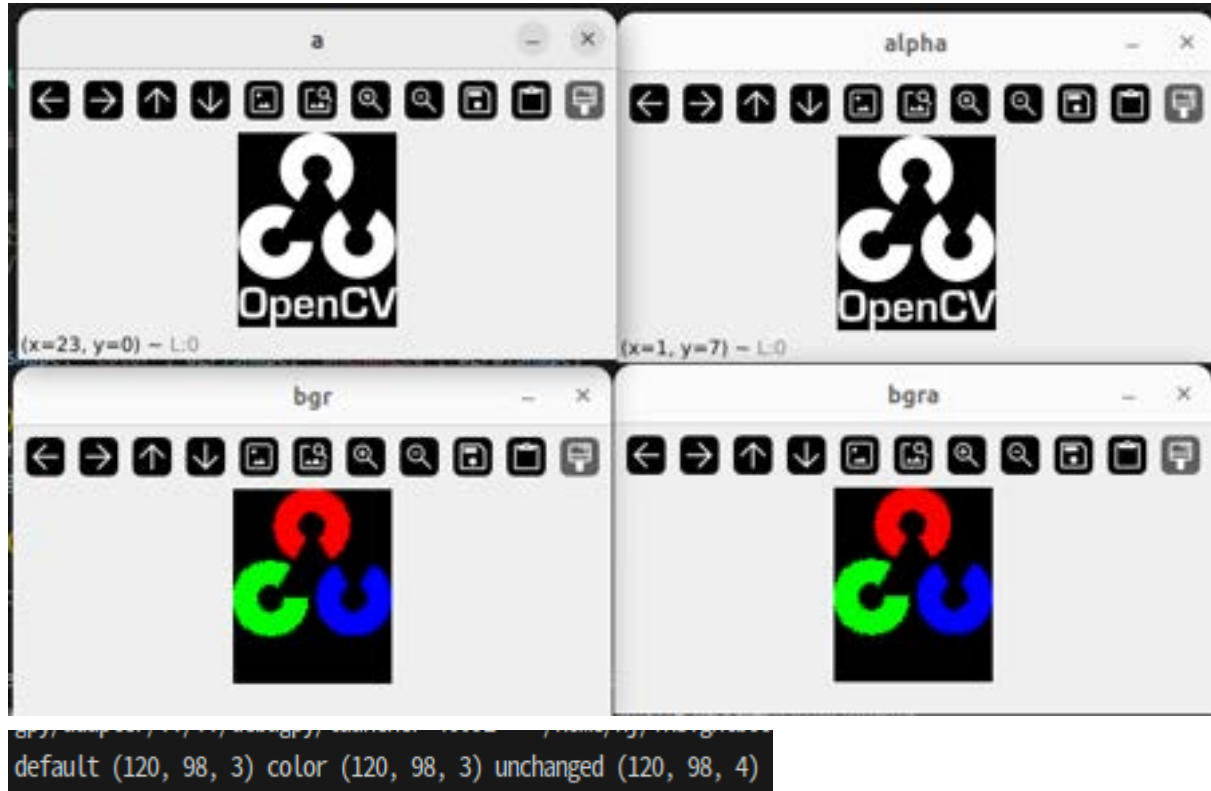
roi_crop_mouse.py



roi_select_img.py



rgba.py



bgr2gray.py



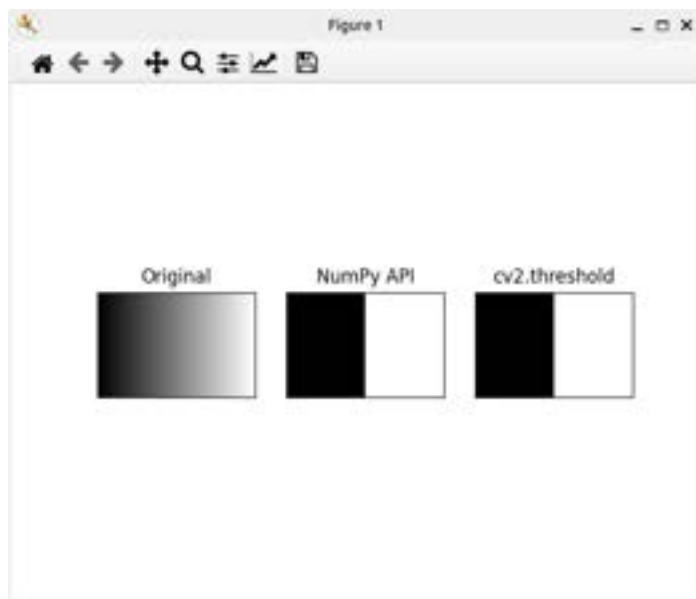
bgr2hsv.py

```
red: [[[ 0 255 255]]]  
green: [[[ 60 255 255]]]  
blue: [[[120 255 255]]]  
yellow: [[[ 30 255 255]]]
```

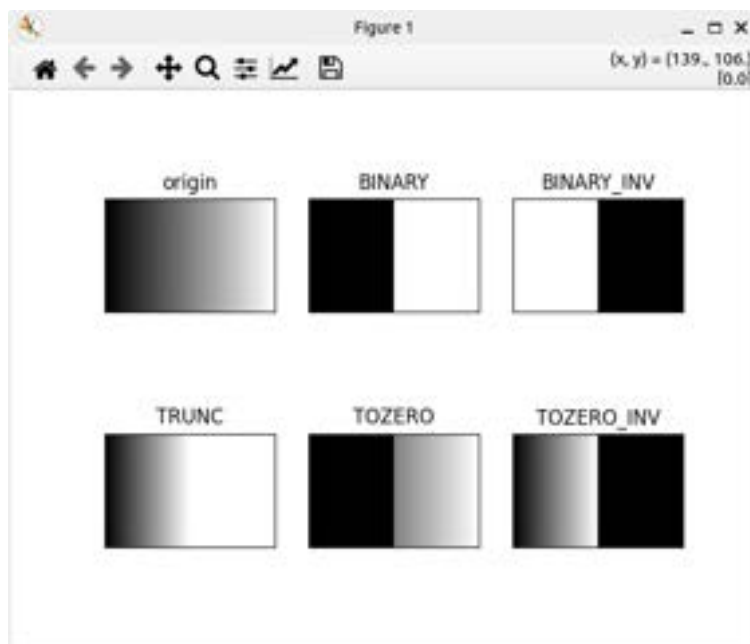
bgr2yuv.py

```
dark: [[[ 0 128 128]]]  
middle: [[[127 128 128]]]  
bright: [[[255 128 128]]]
```

threshold.py



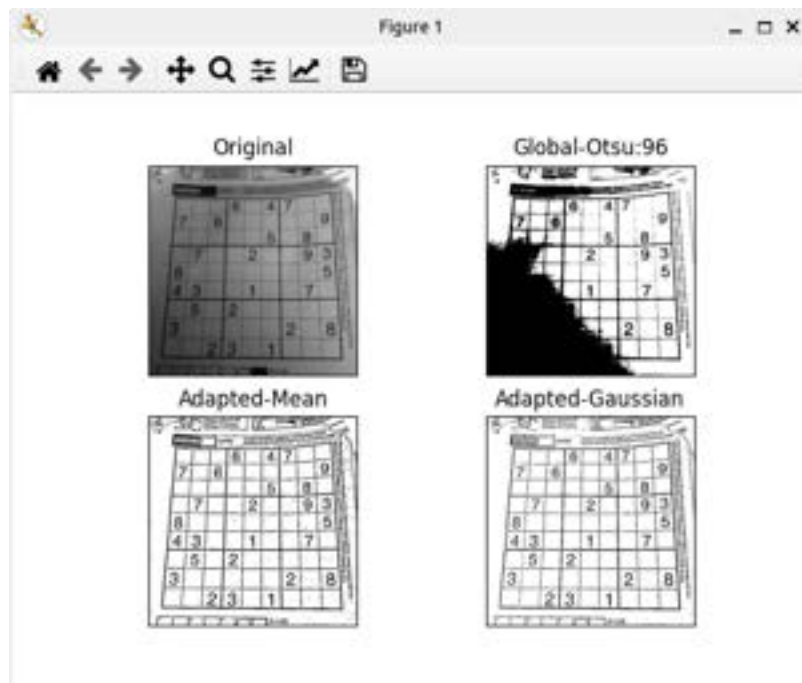
threshold_flag.py



threshold_otsu.py



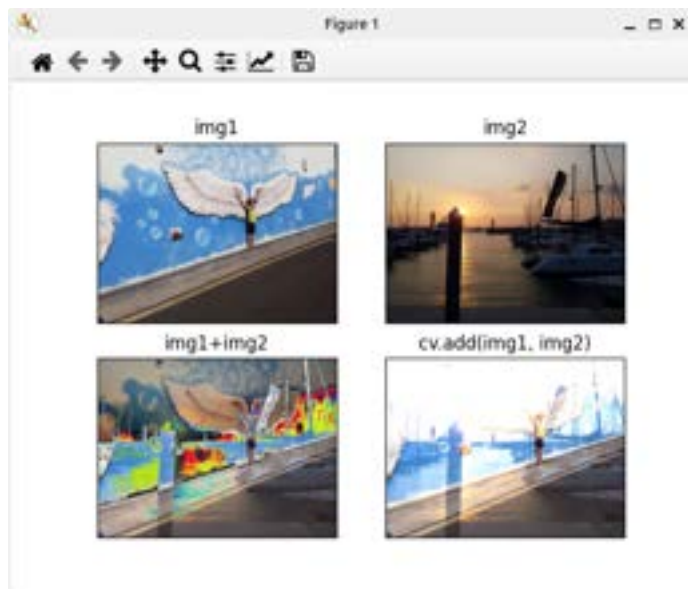
threshold_adpted.py



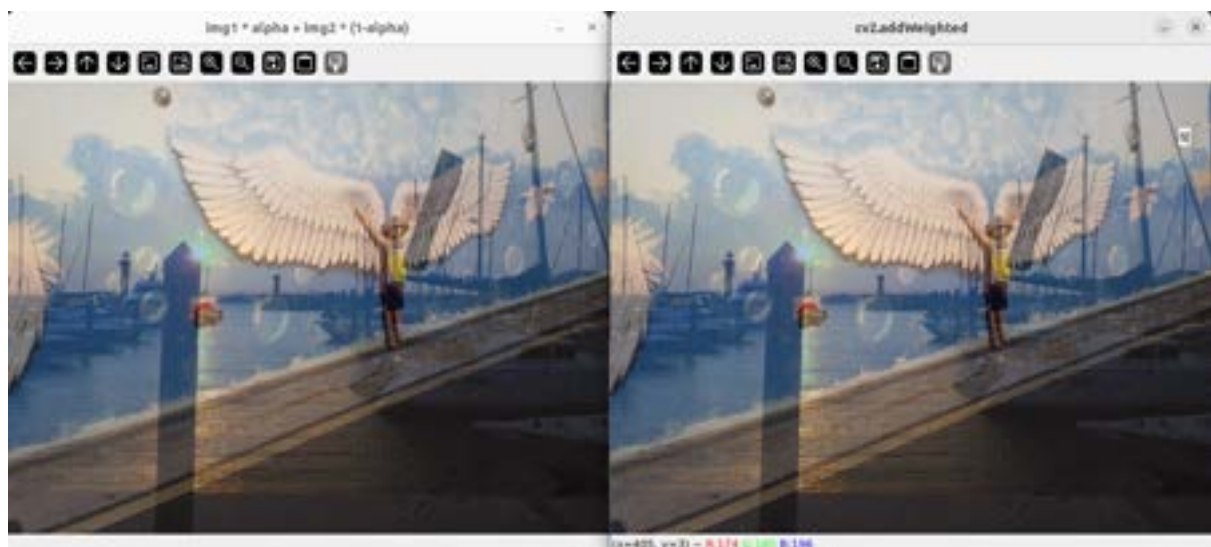
arithmetic.py

```
[[ 44 150]] [[255 150]]  
[[100 206]] [[100  0]]  
[[144 100]] [[255 100]]  
[[66.66666667 16.66666667]] [[67 17]]
```

blending_simple.py



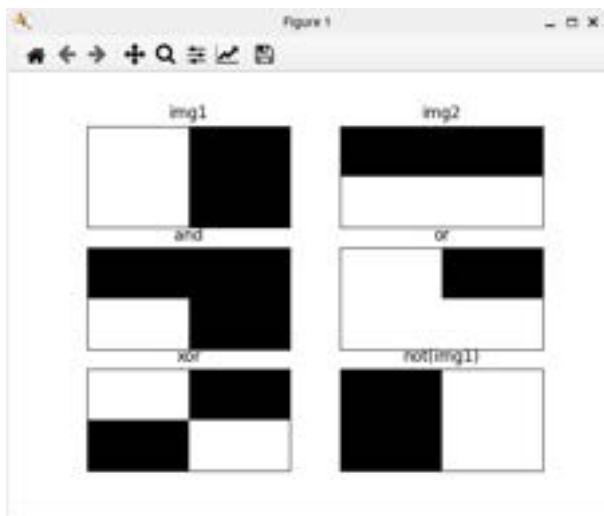
blending_alpha.py



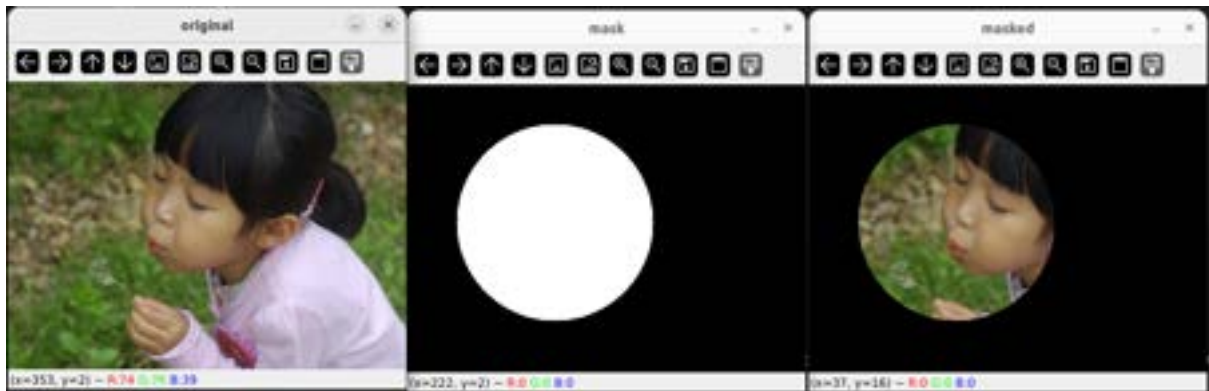
blending_alpha_trackbar.py



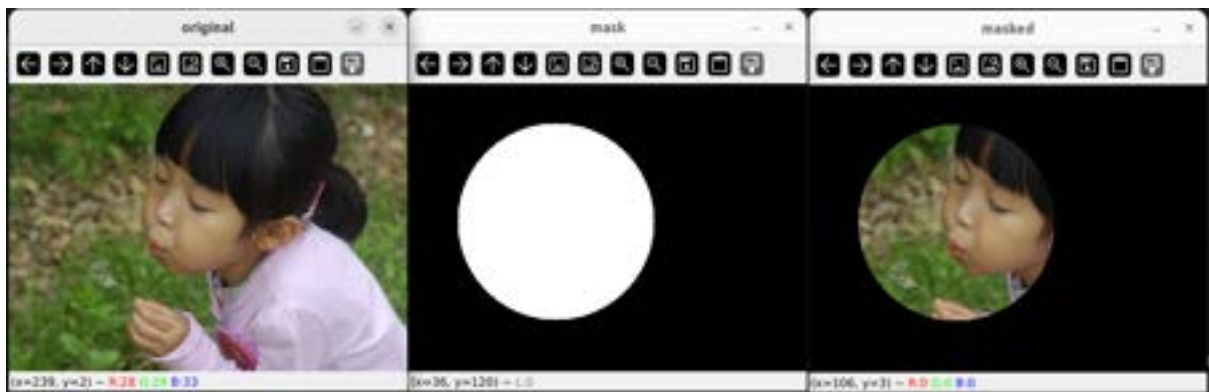
bitwise.py



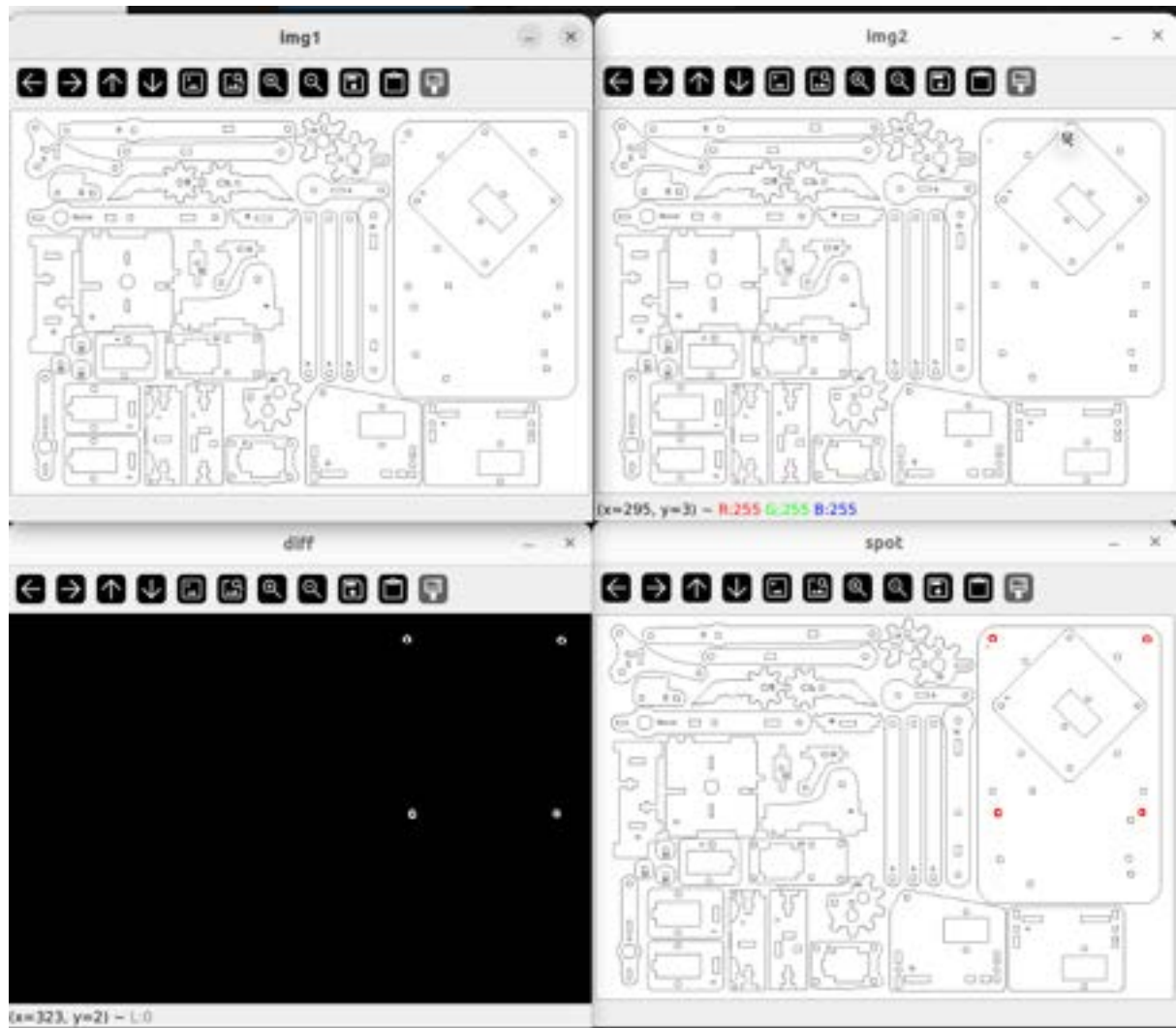
bitwise_masking.py



bitwise_masking2.py



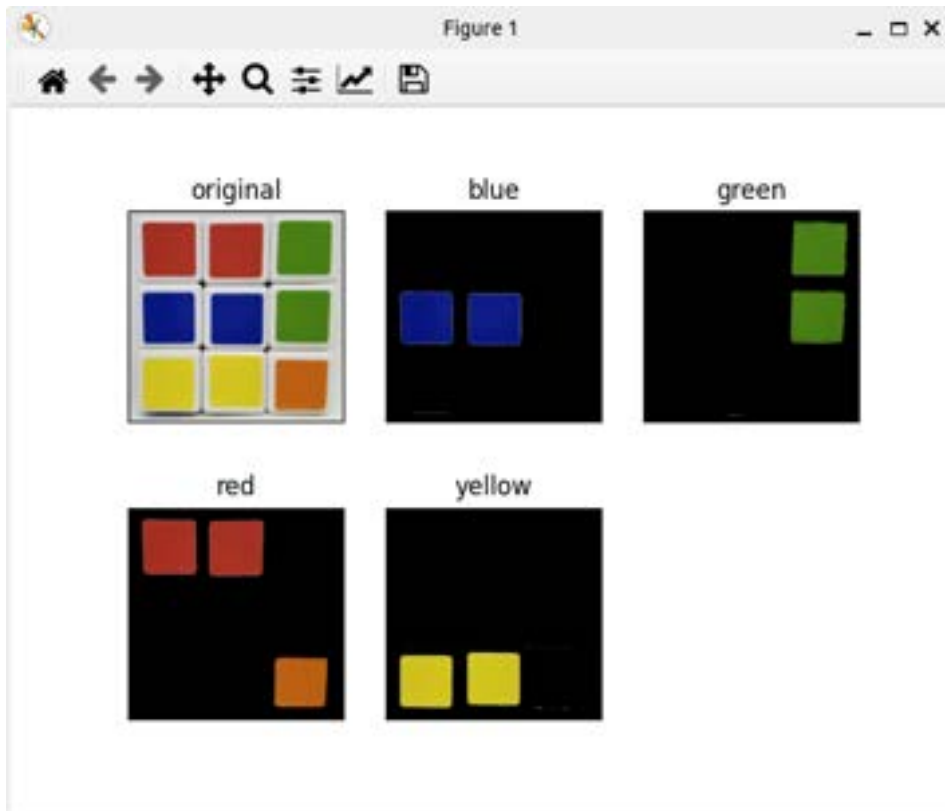
diff_absolute.py



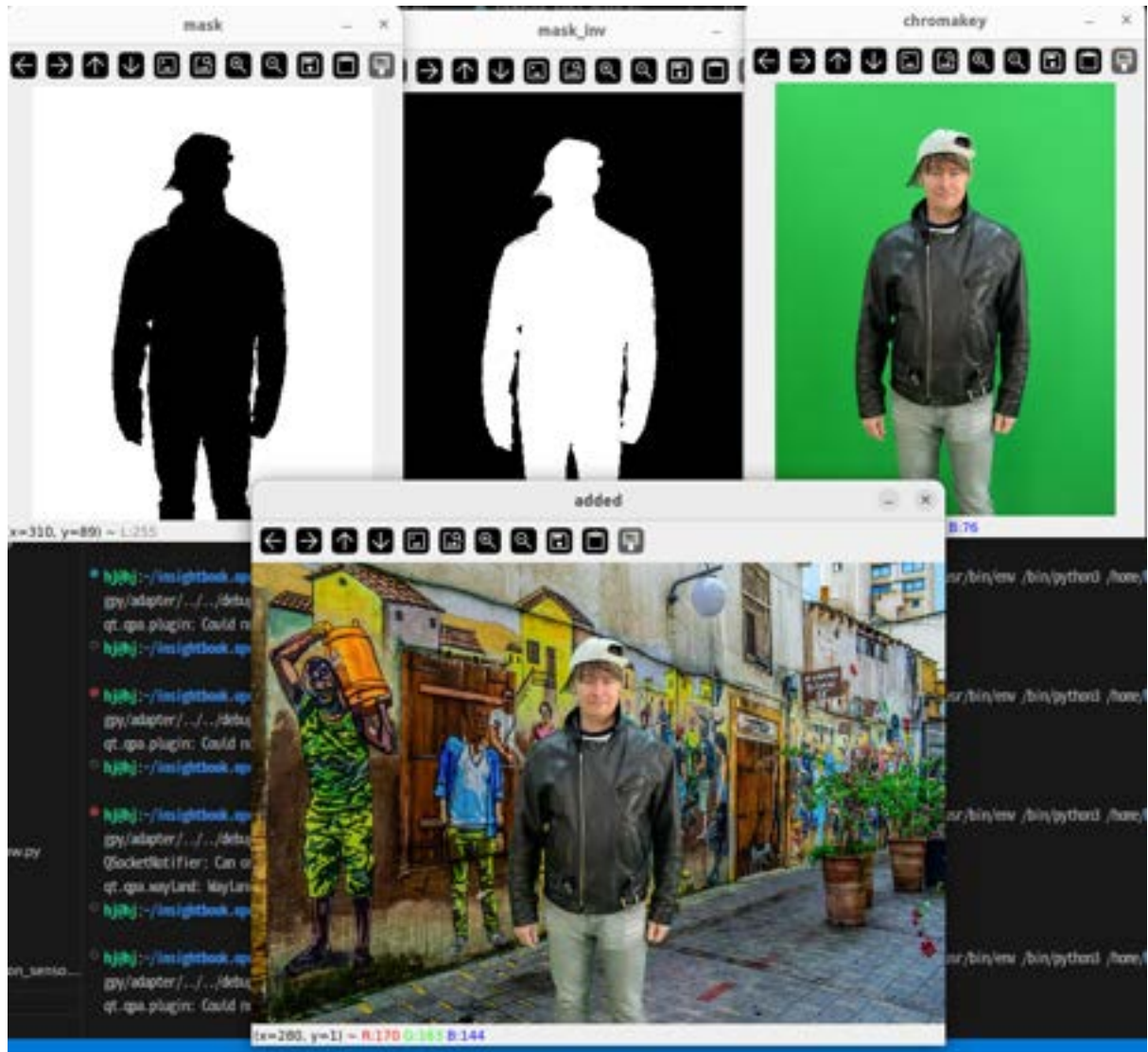
addition_rgba_mask.py



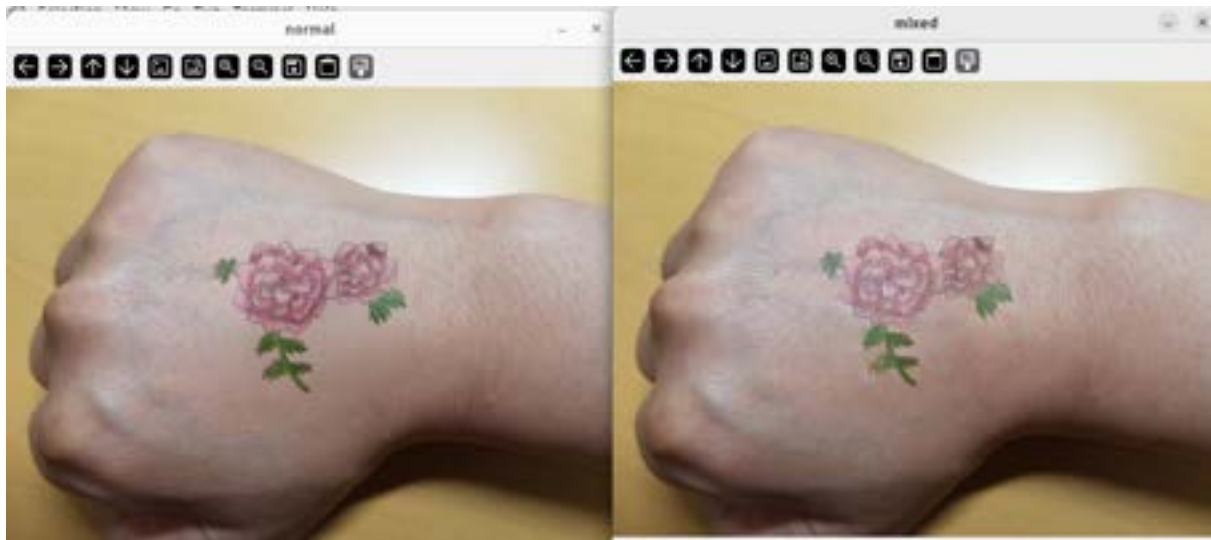
hsv_color_mask.py



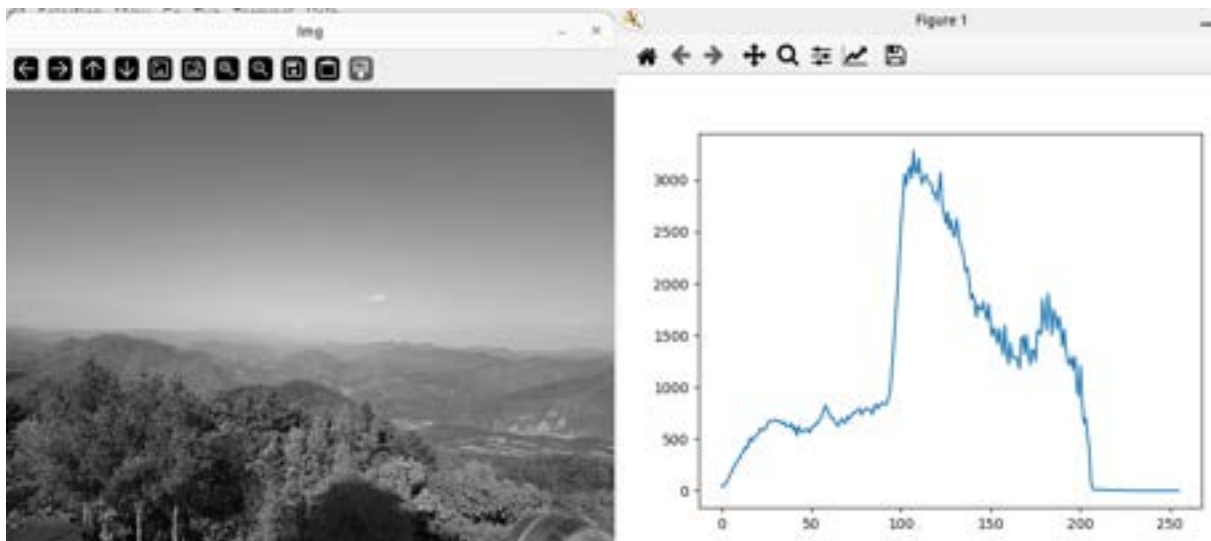
chromakey.py



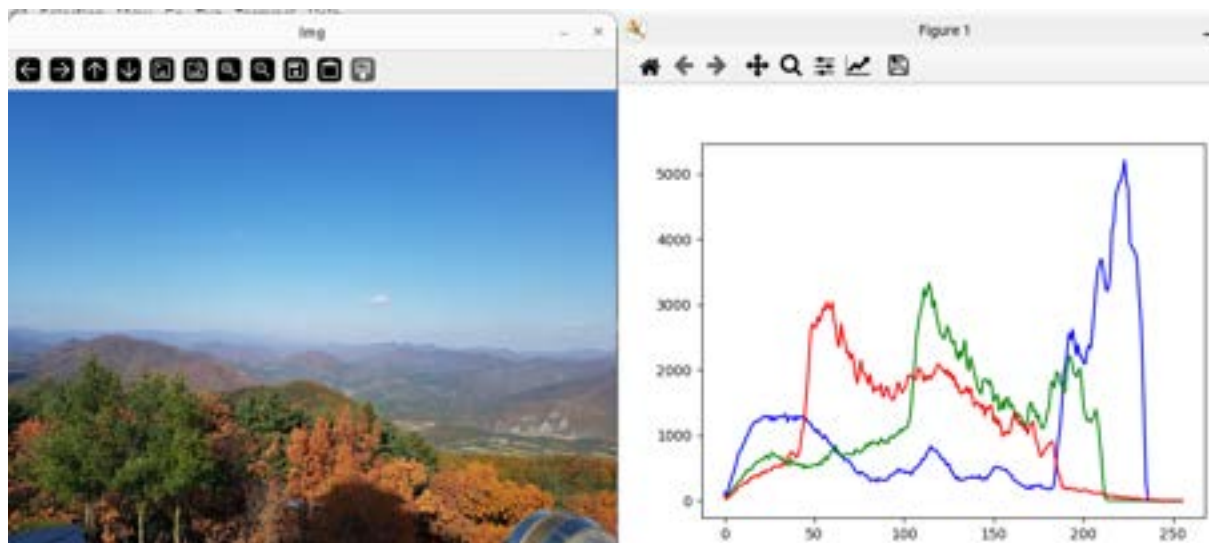
seamlessclone.py



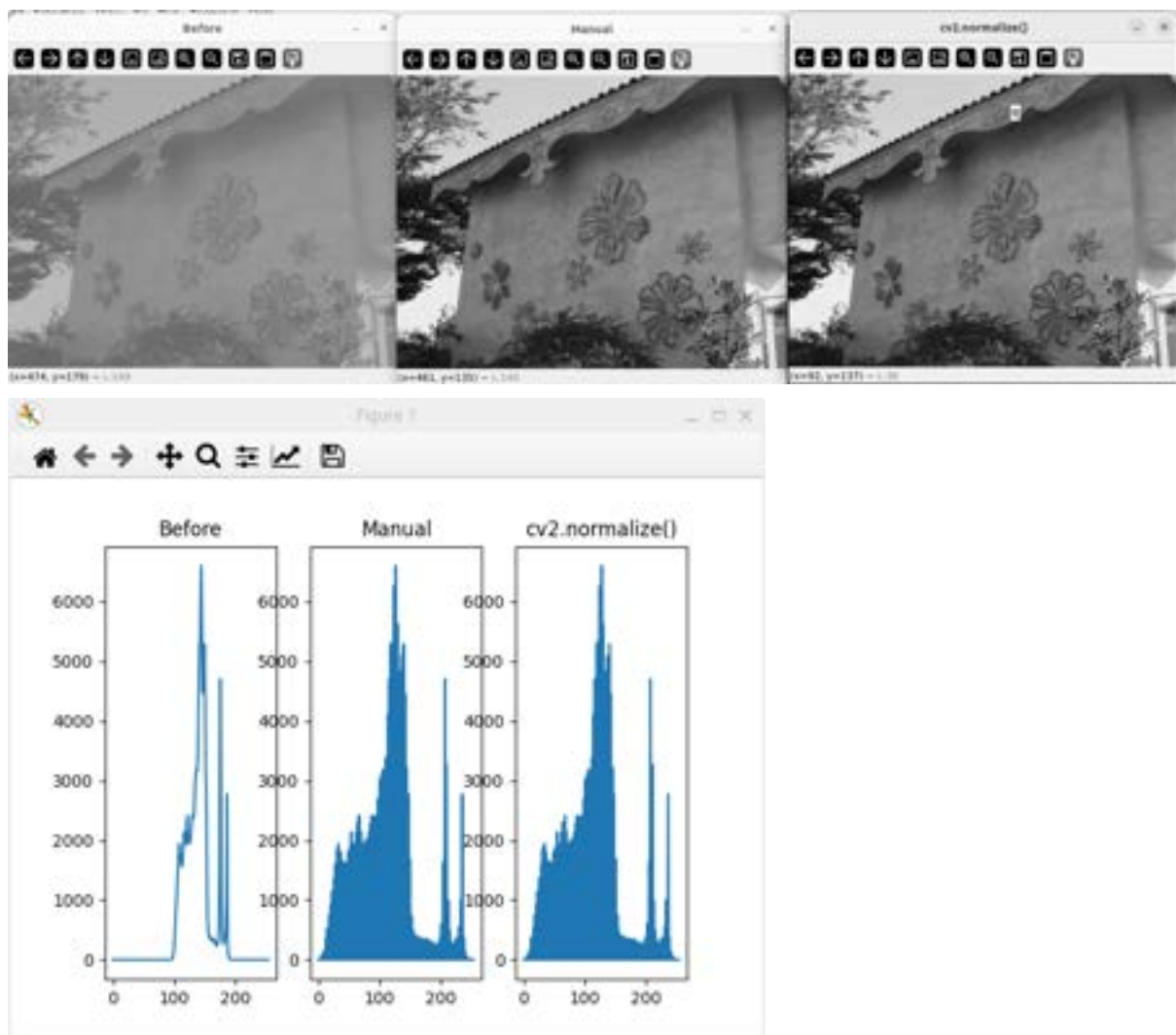
histo_gray.py



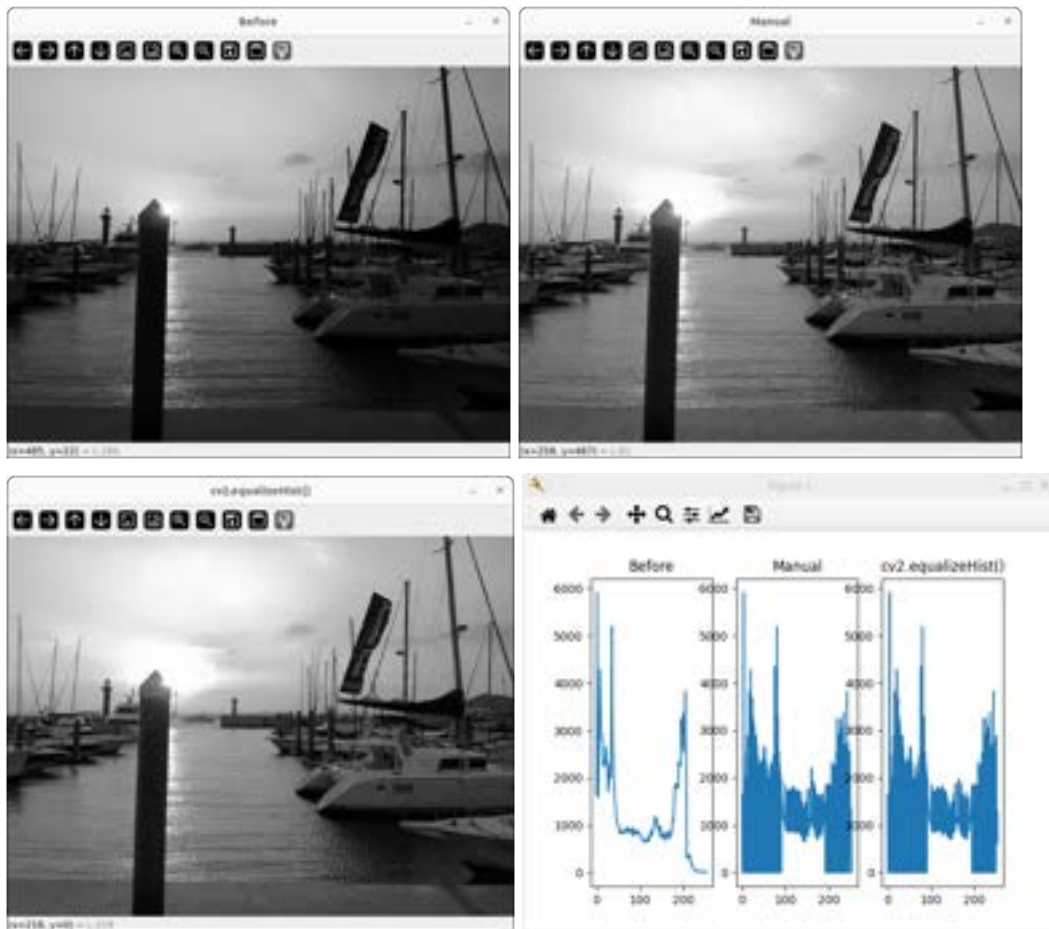
histo_rgb.py



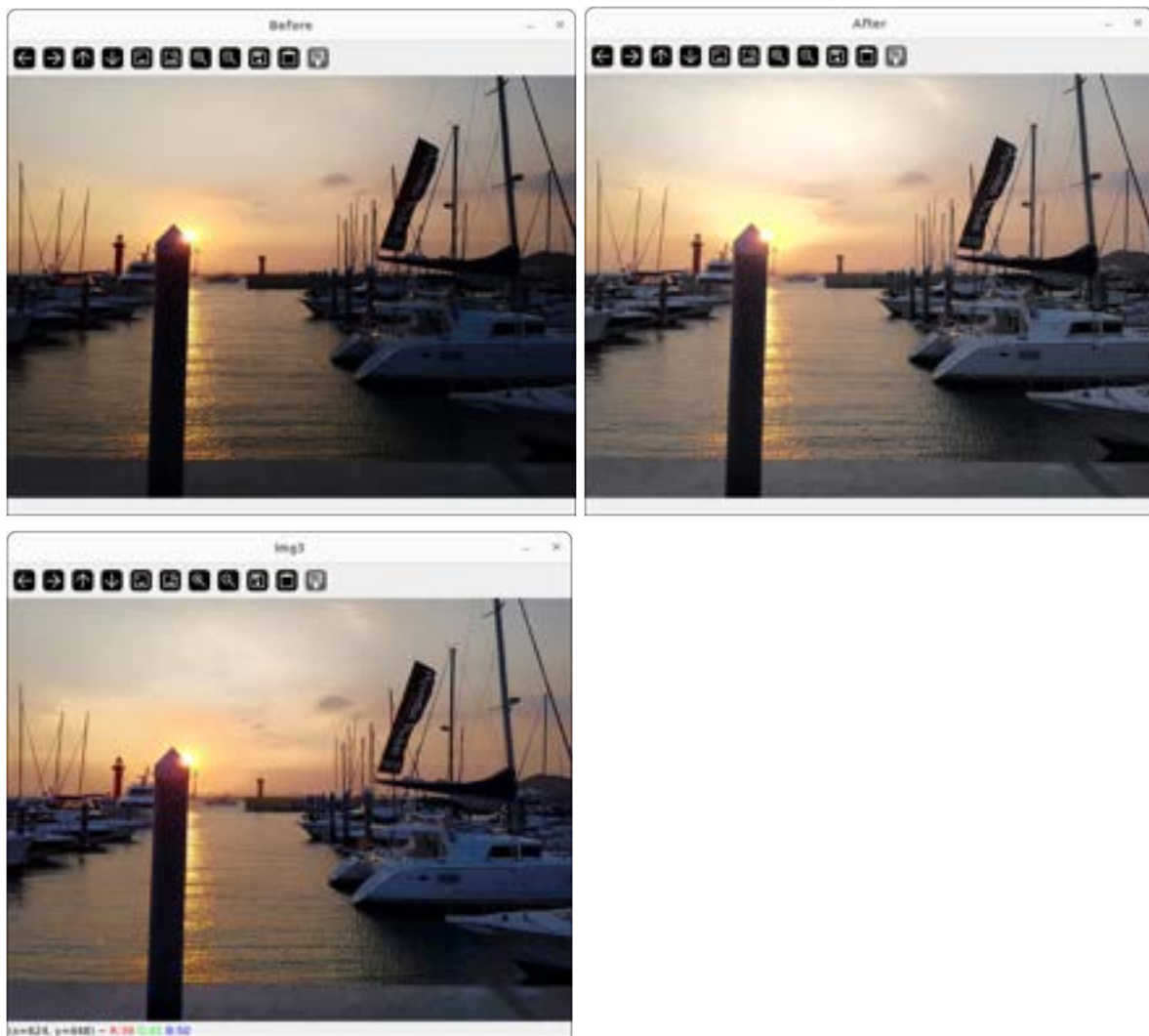
histo_normalize.py



histo_equalize.py



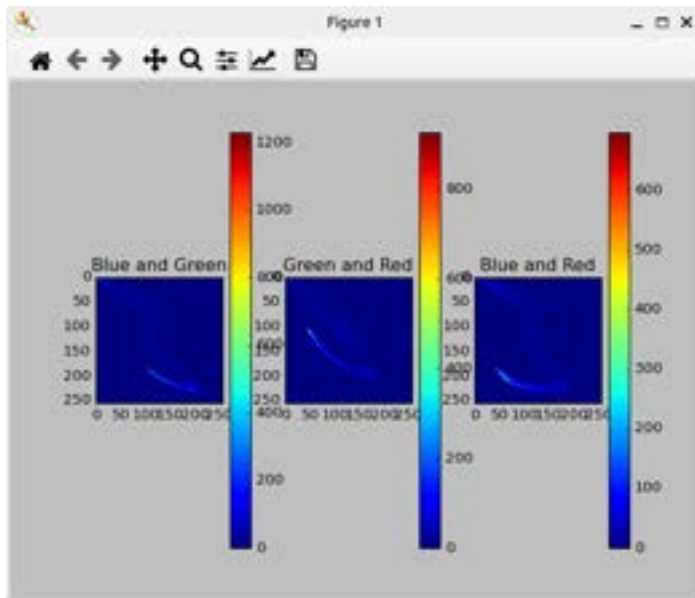
histo_equalize_yuv.py



histo_clahe.py



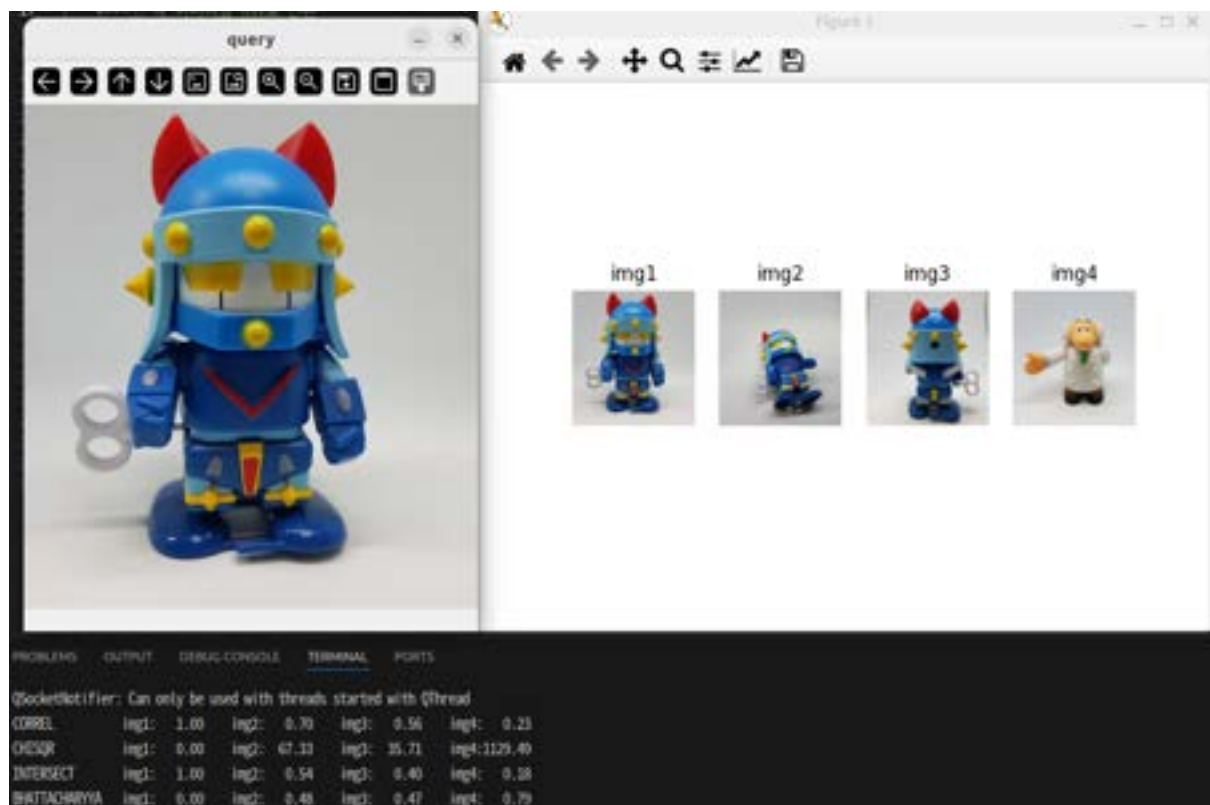
histo_2d.py



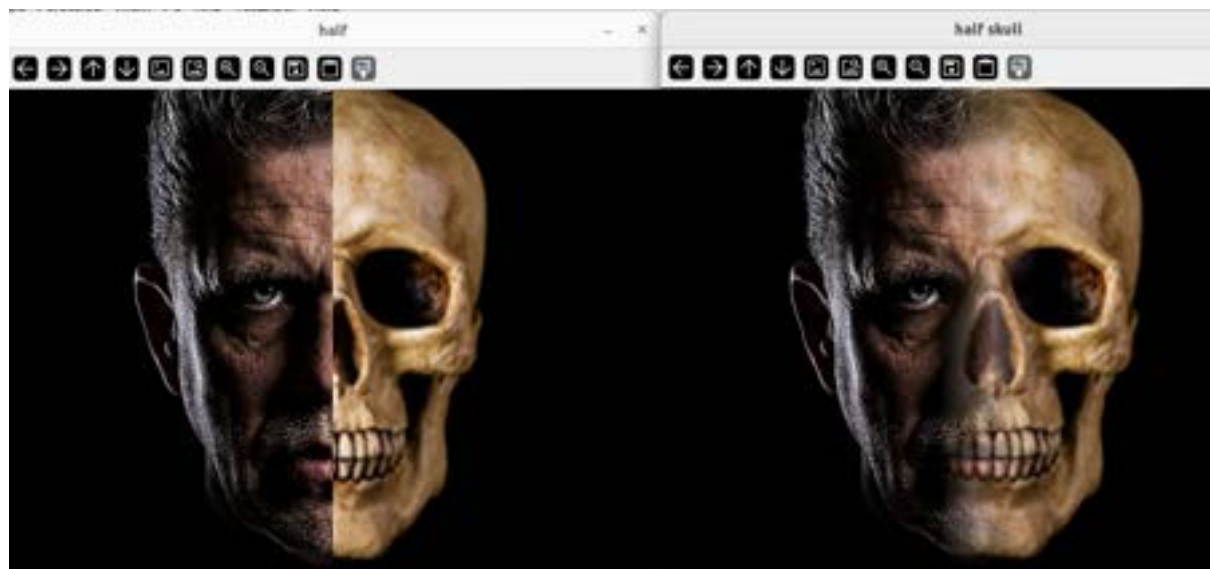
histo_backproject.py



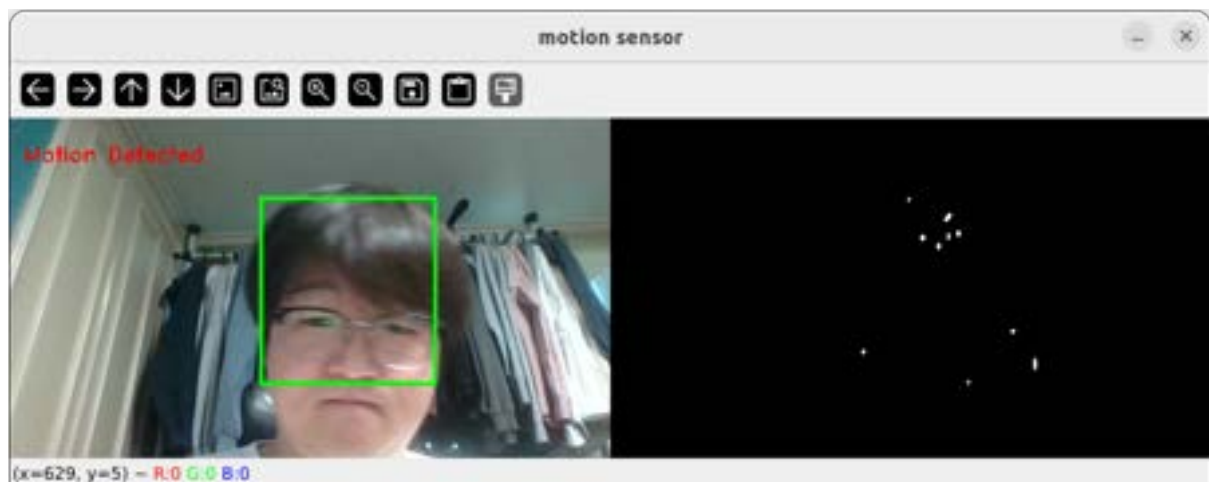
histo_compare.py



workshop_two_face.py



workshop_cctv_motion_sensor.py



threshold_flag_cvshow.py

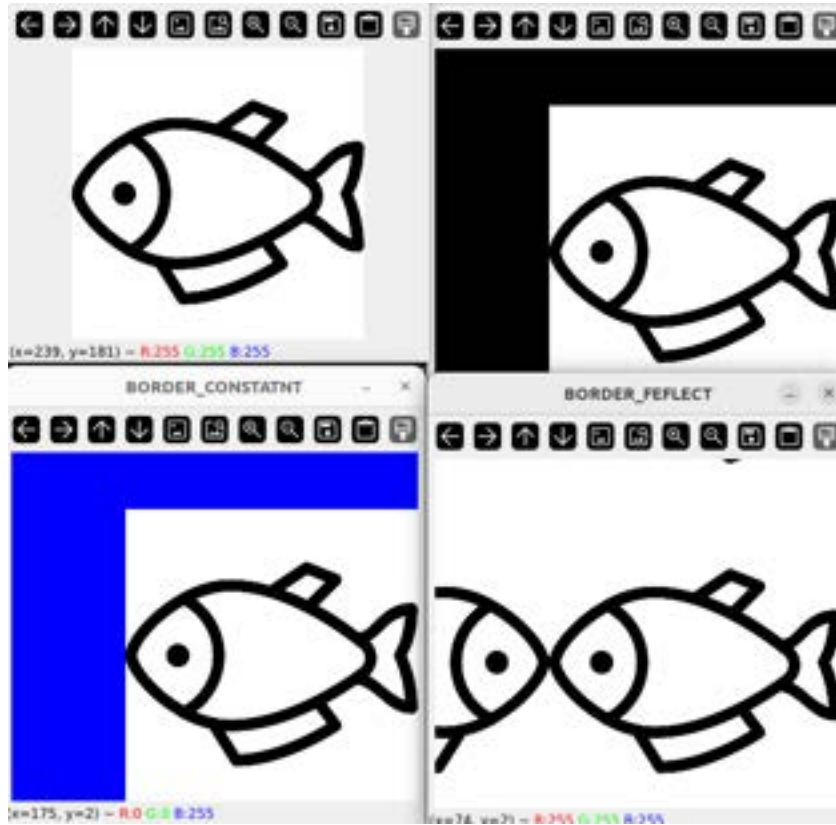


thresholds.py

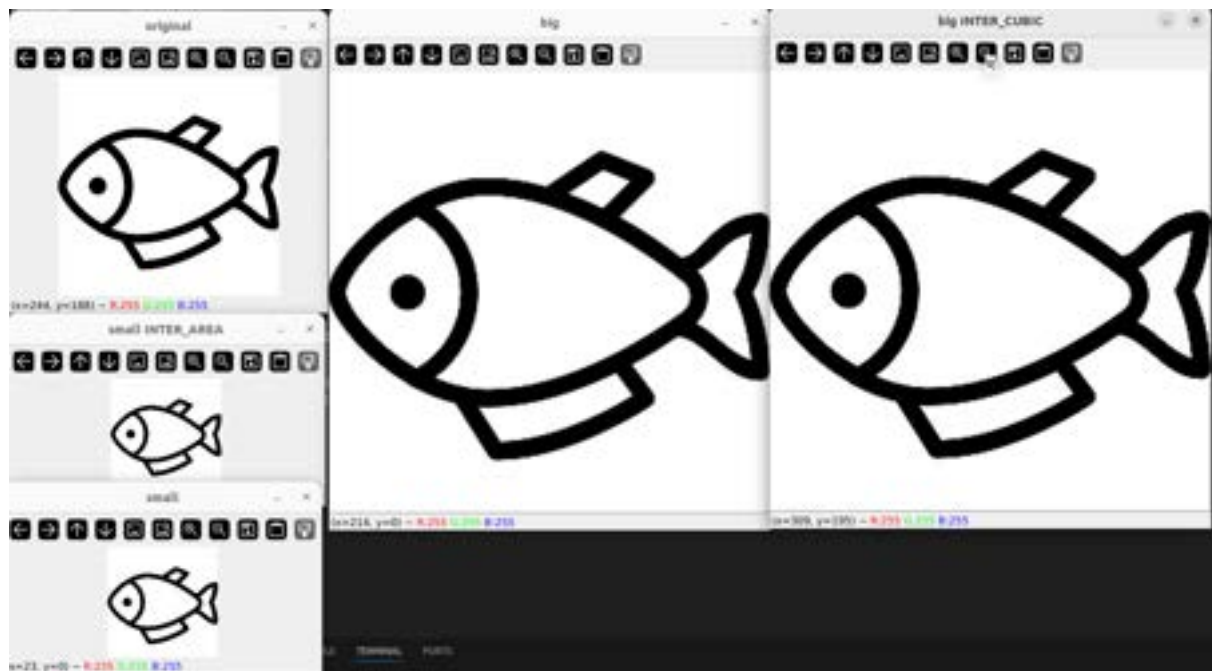


강좌_5

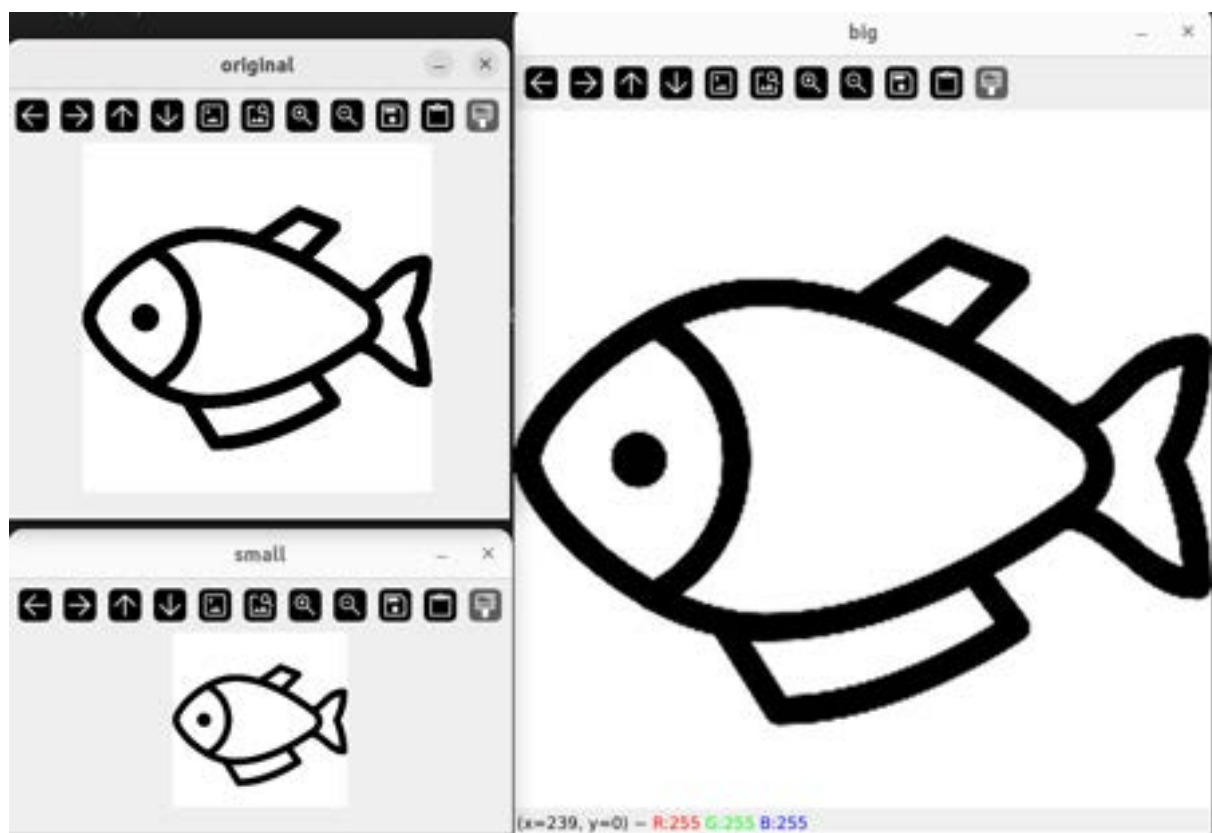
translate.py



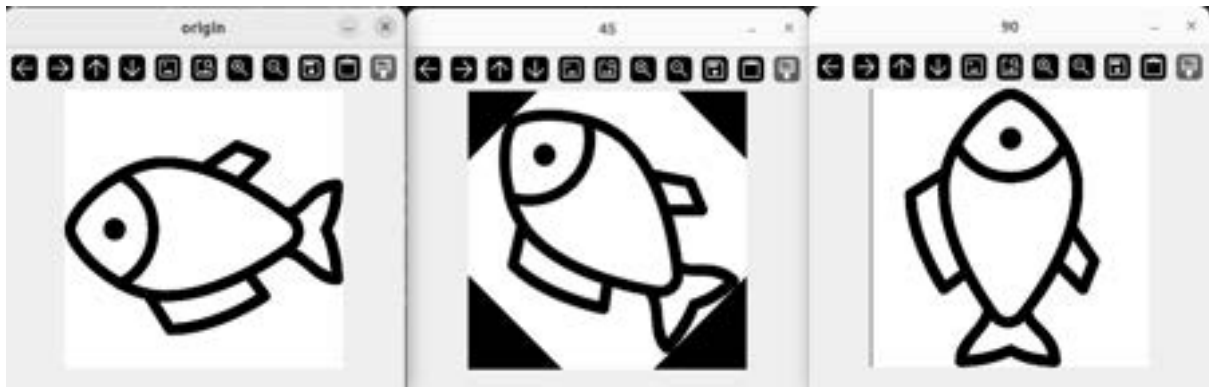
scale_matrix.py



scale_resize.py



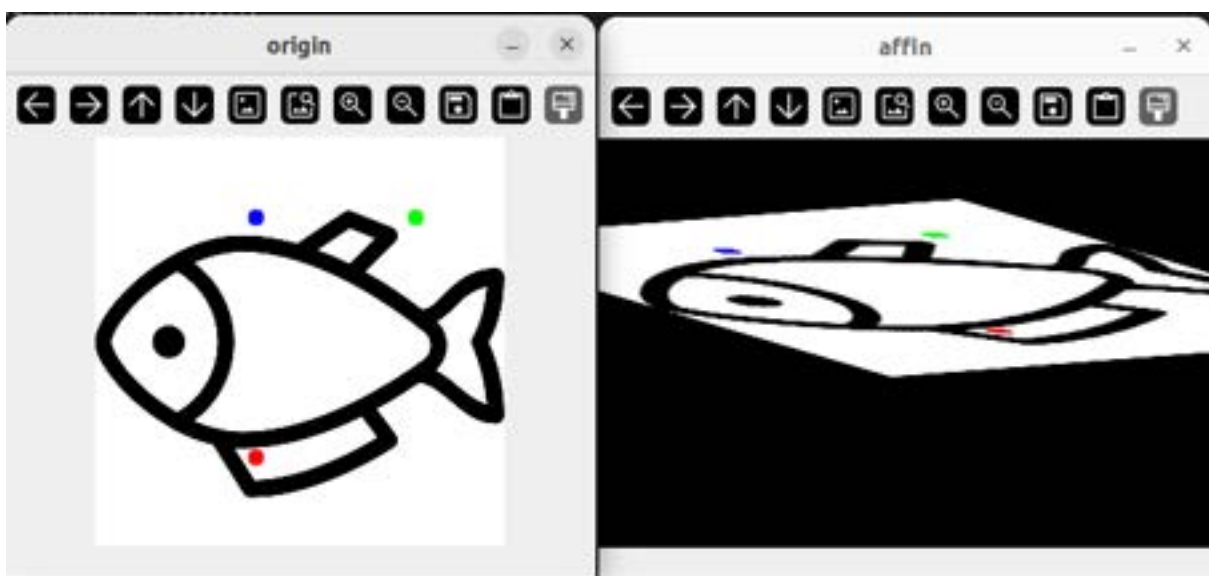
rotate_matrix.py



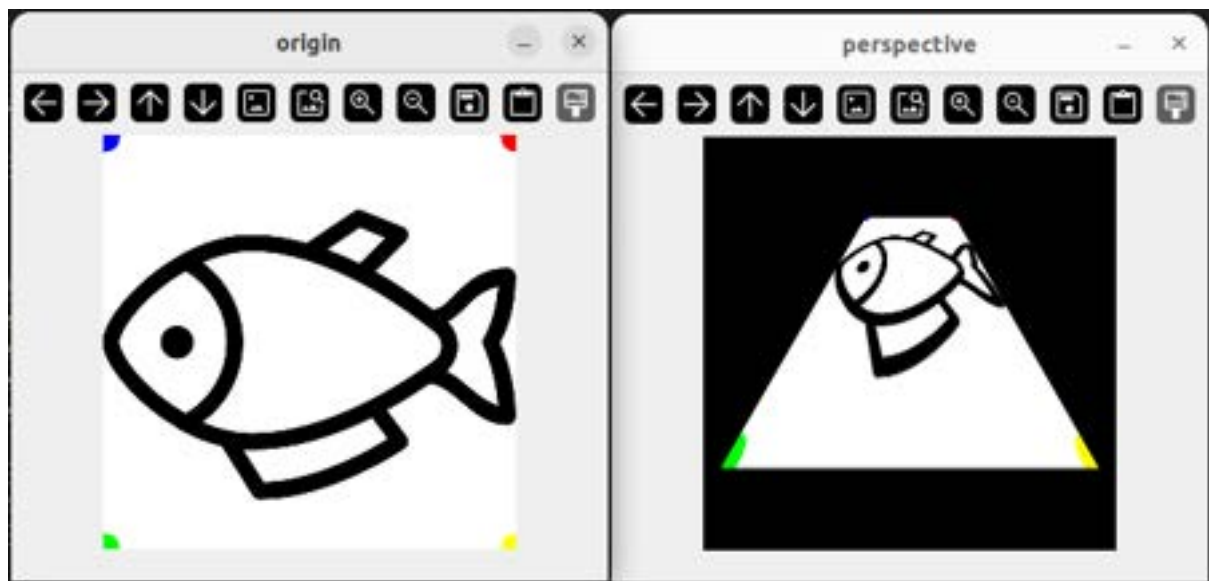
rotate_getmatrix.py



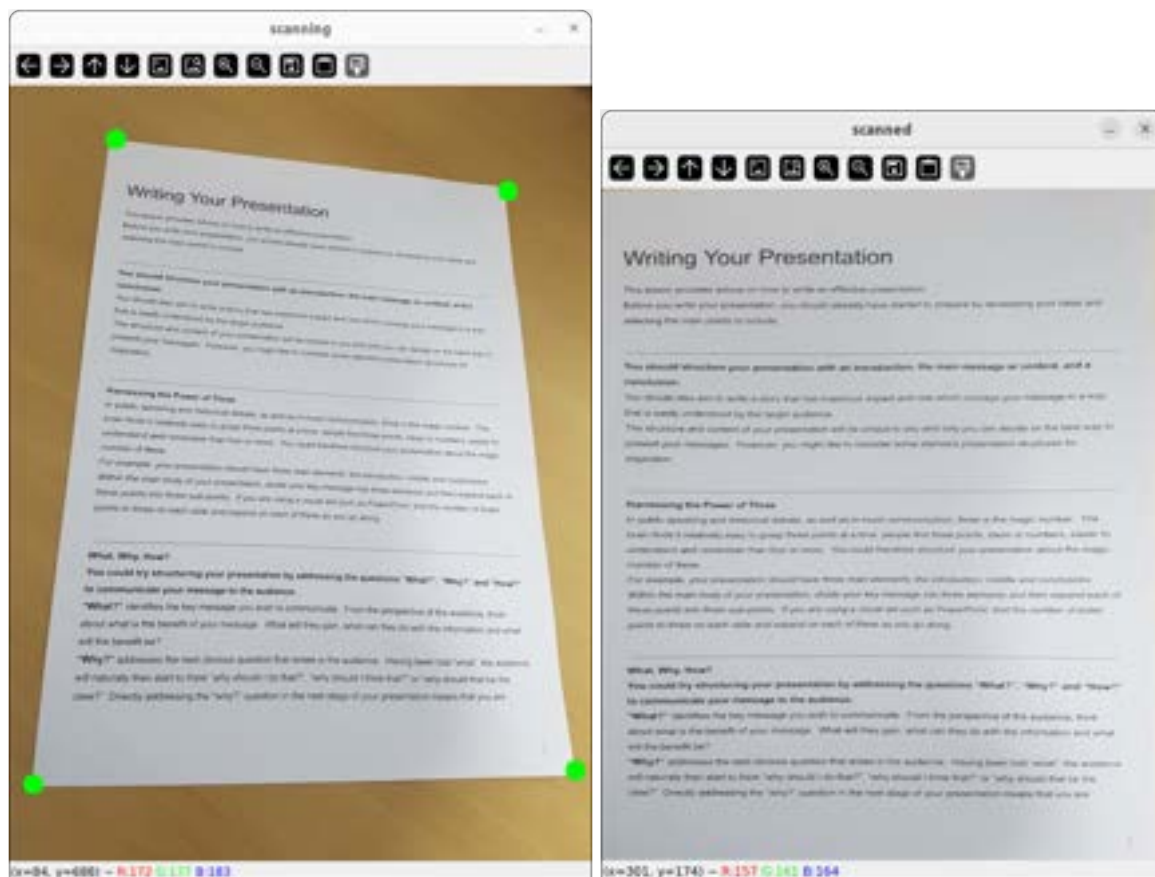
getAffine.py



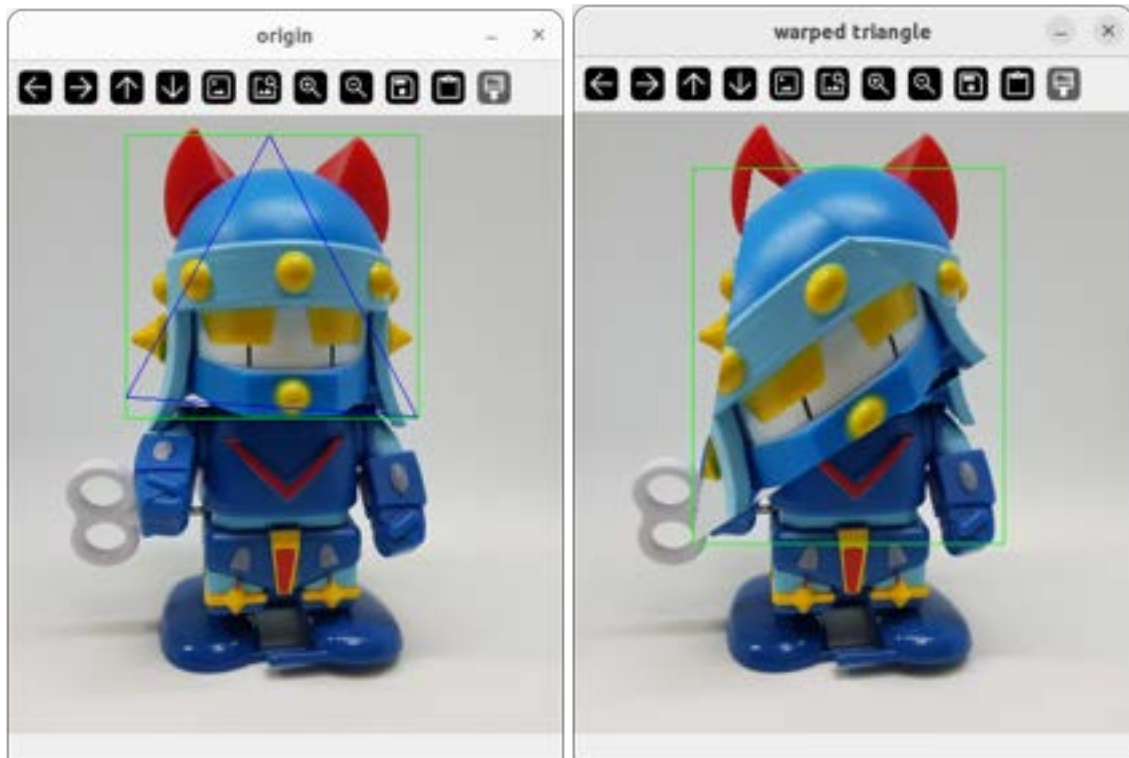
perspective.py



perspective_scan.py



triangle_affine.py



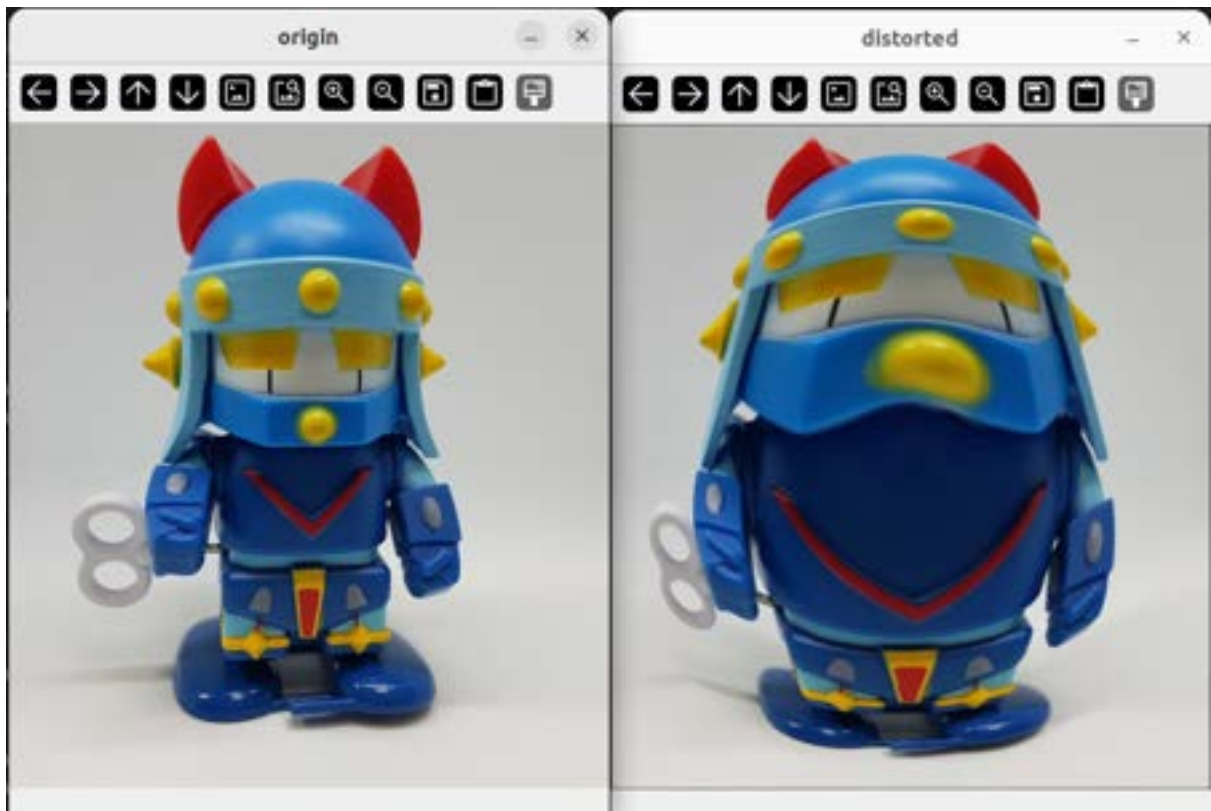
remap_flip.py



remap_sin_cos.py



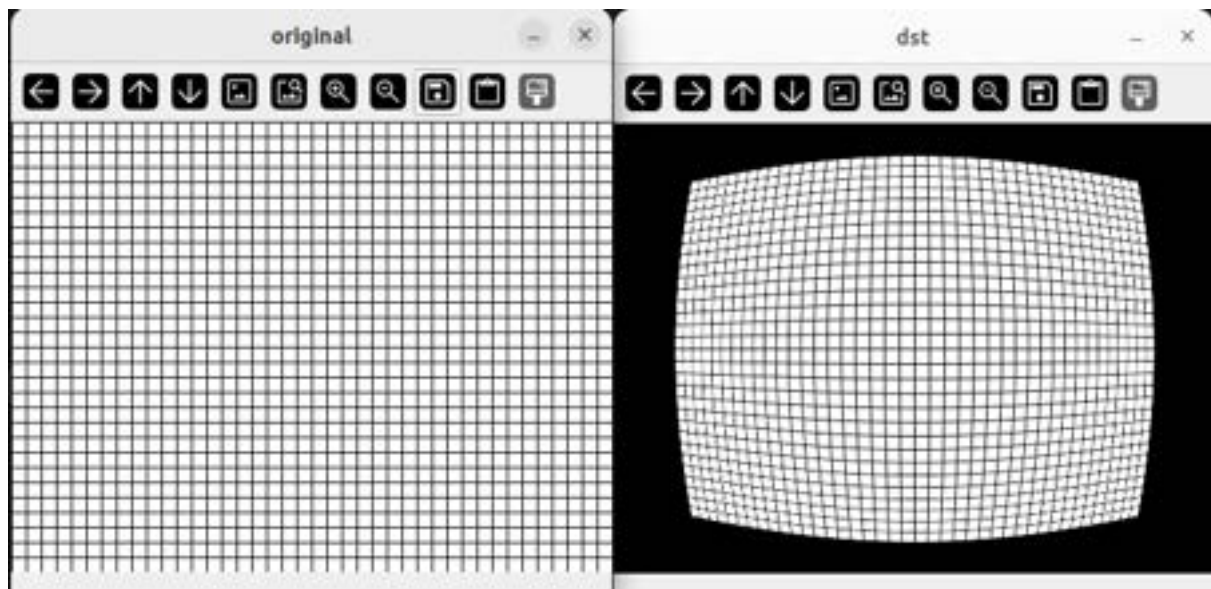
remap_lens.py



remap_barrel.py



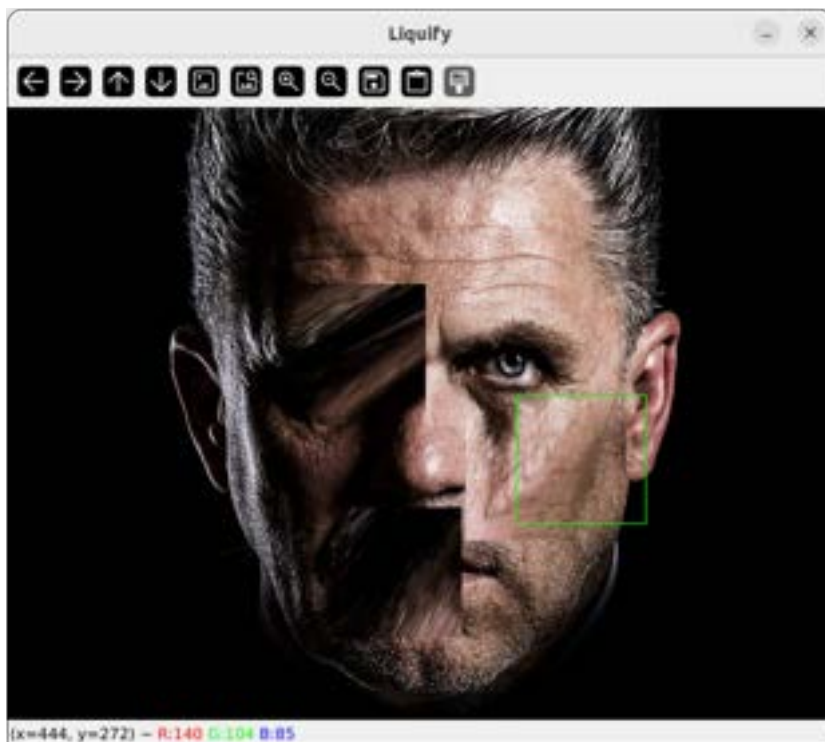
undistort_barrel.py



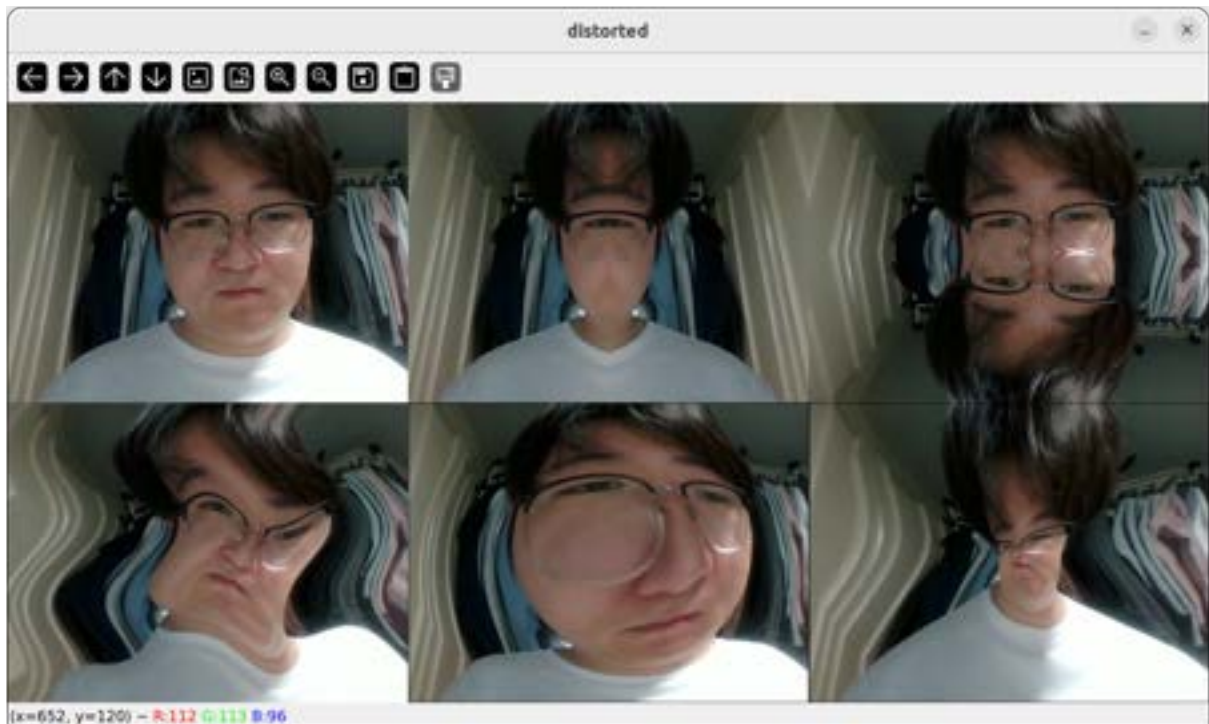
workshop_mosaic.py



workshop_liquify_tool.py



workhop_distotion_camera.py



강좌_6

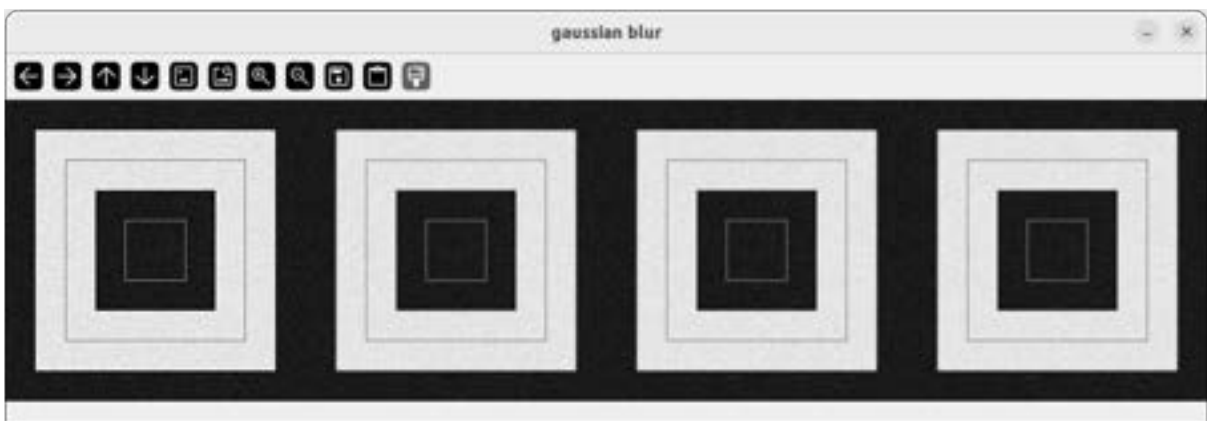
blur_avg_kernel.py



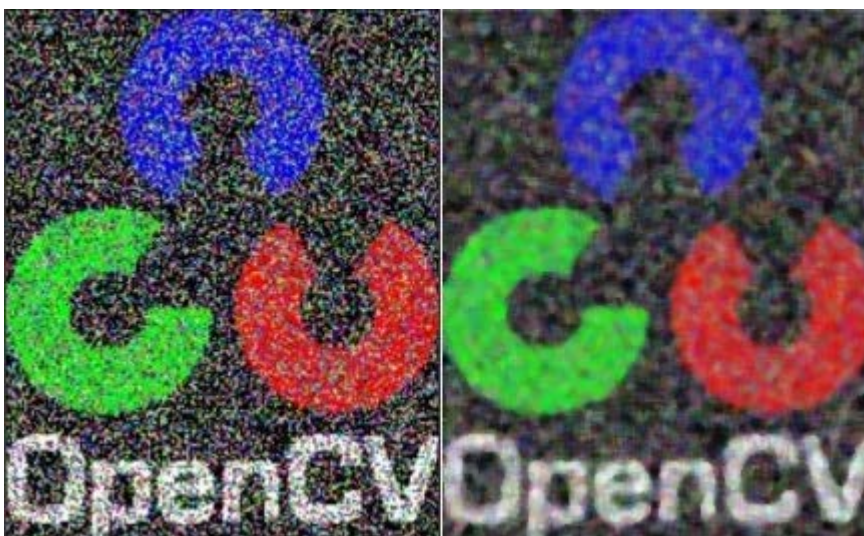
blur_avg_api.py



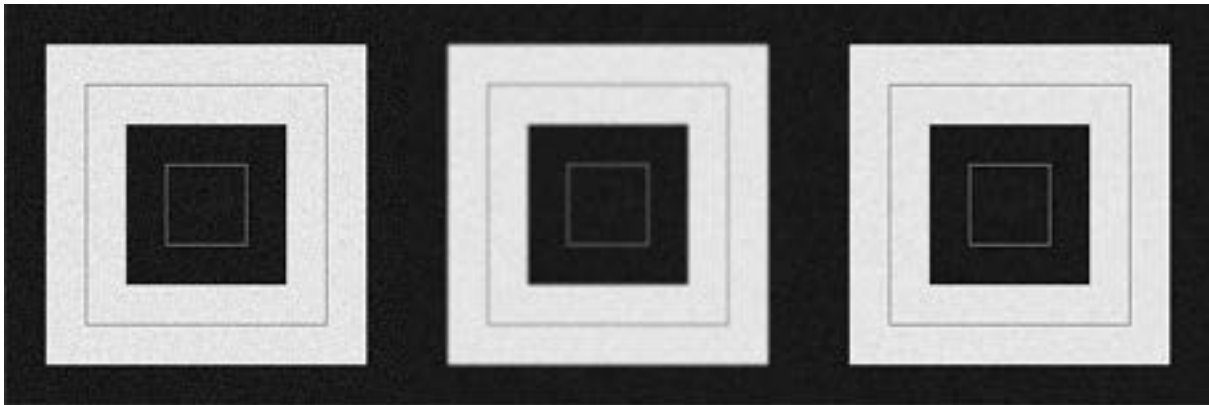
blur_gaussian.py



blur_median.py



blur_bilateral.py



edge_differential.py



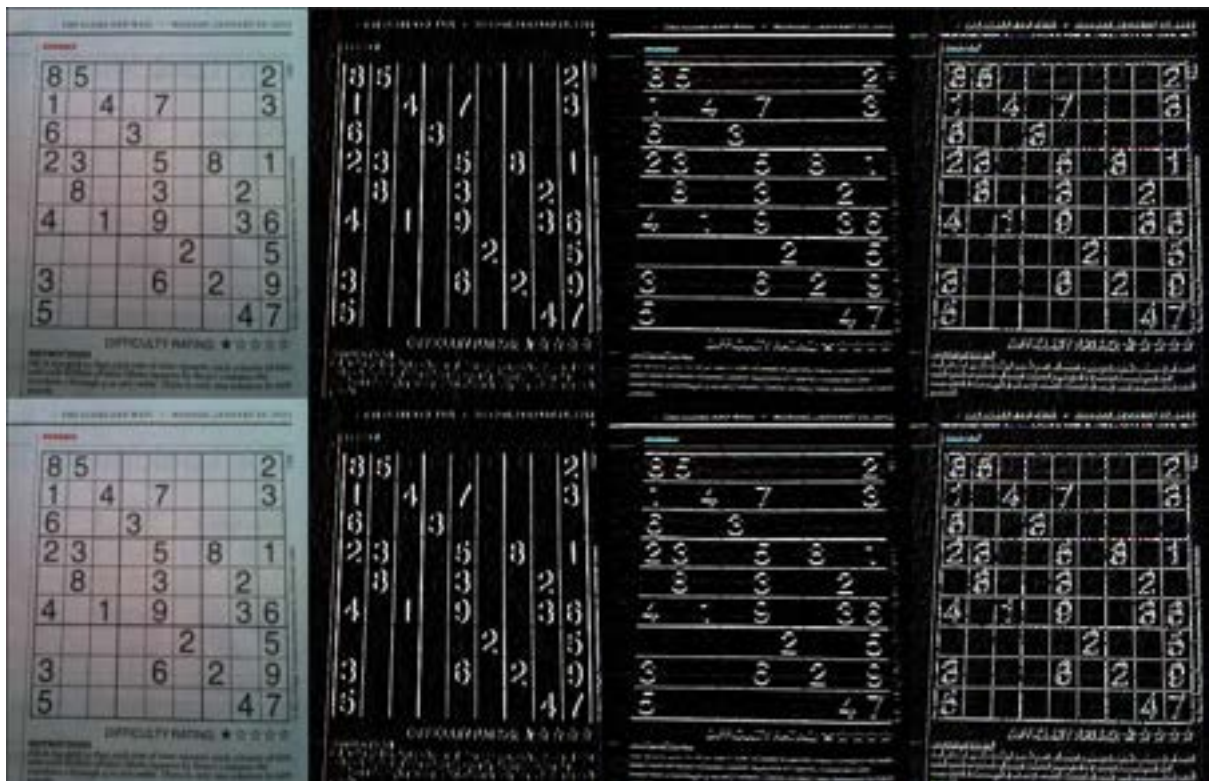
edge_roberts.py



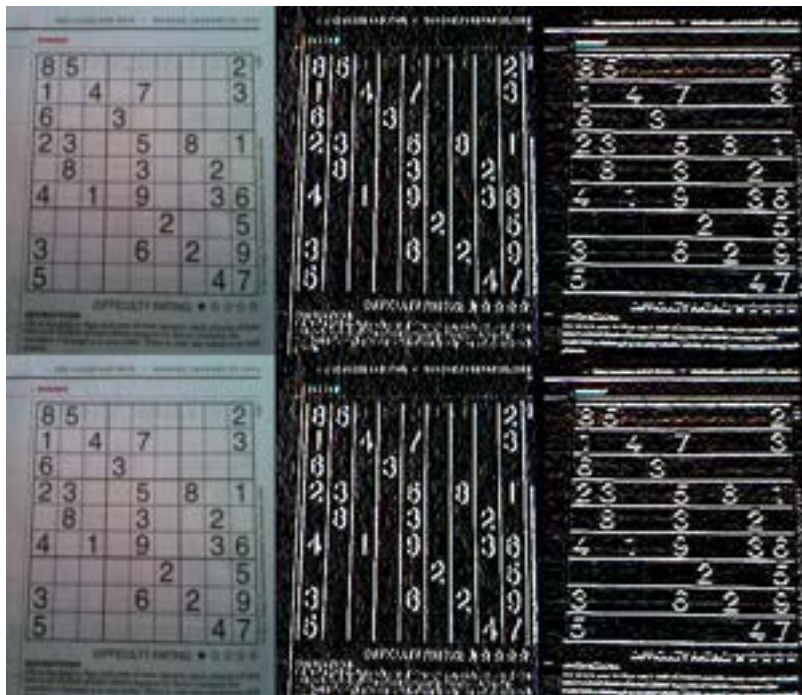
edge_prewitt.py



edge_sobel.py



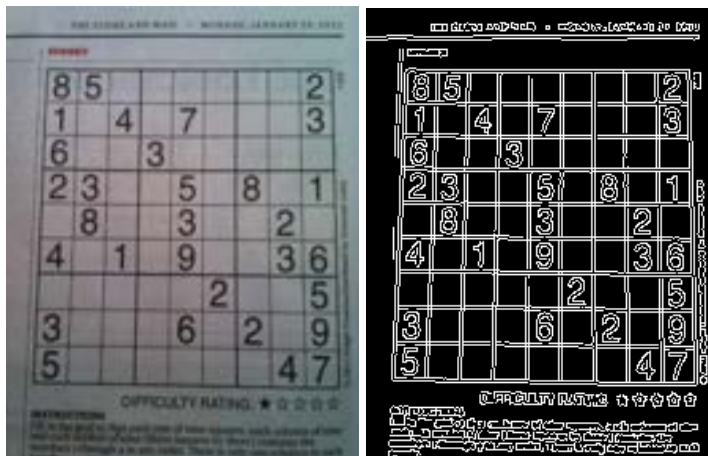
edge_scharr.py



edge_laplacian.py



edge_canny.py



morph_erode.py



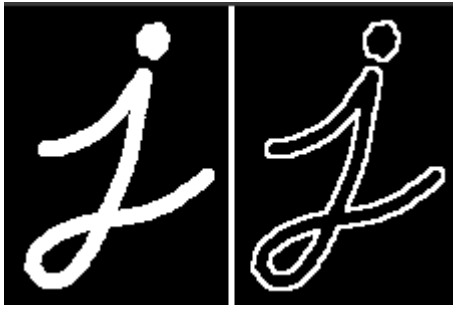
morph_dilate.py



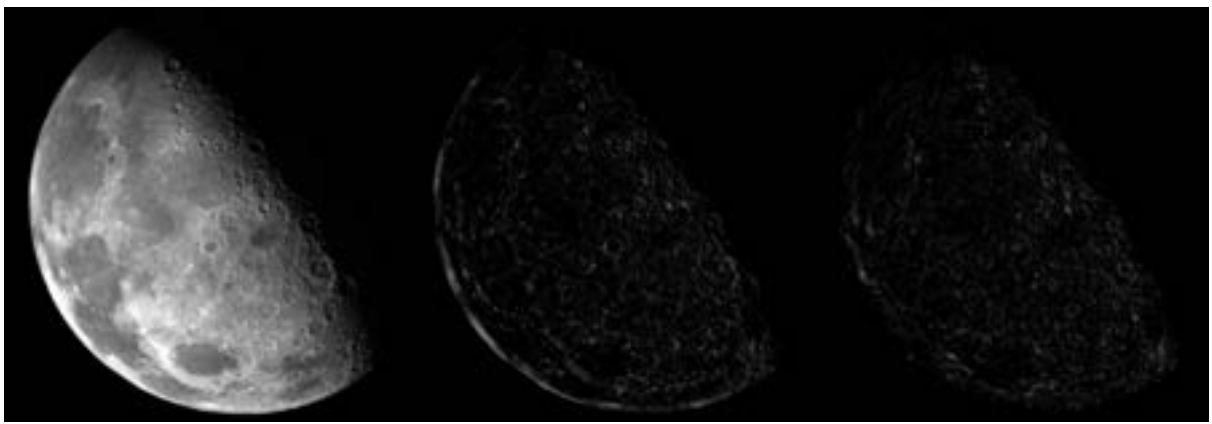
morph_open_close.py



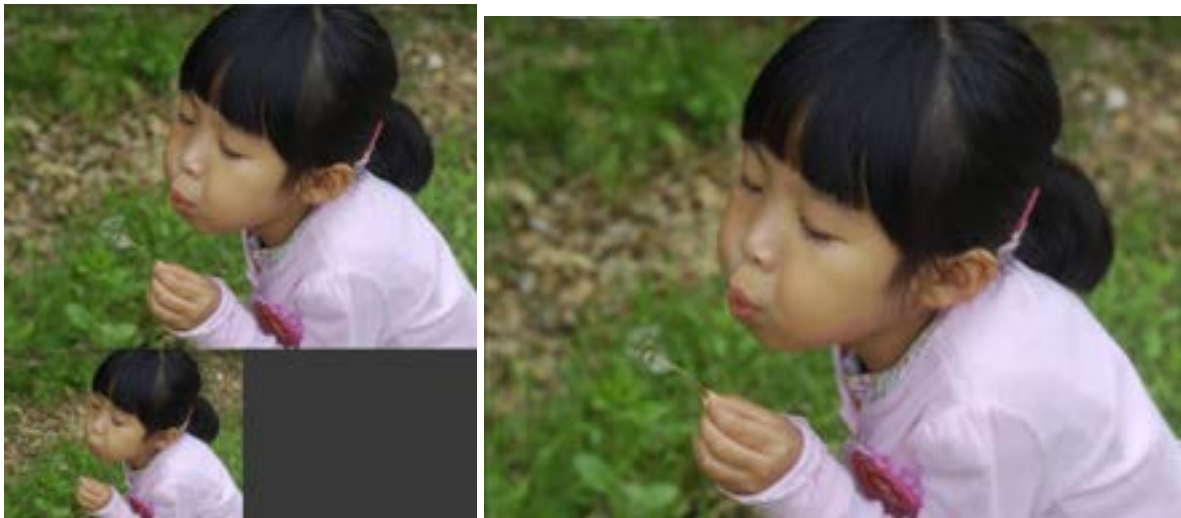
morph_gradient.py



morph_hat.py



pyramid_gaussian.py



pyramid_laplacian.py



workshop_mosaic2.py



workshop_painting_cam.py



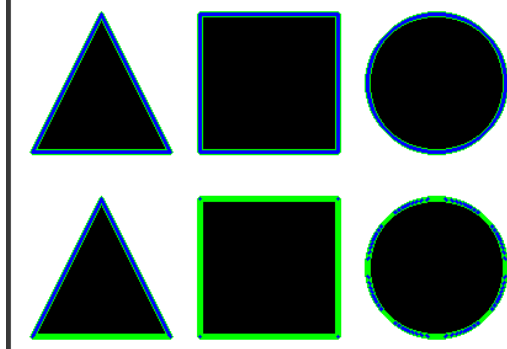
workshop_painting_cam_add_debugging.py



강좌_7

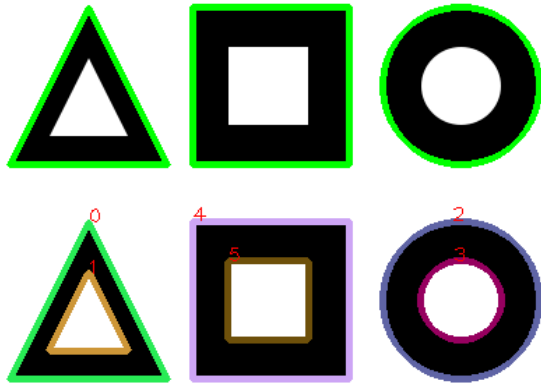
cntr_find.py

```
contour[0] shape: (316, 1, 2)
contour[1] shape: (448, 1, 2)
contour[2] shape: (336, 1, 2)
contour2[0] shape: (152, 1, 2)
contour2[1] shape: (4, 1, 2)
contour2[2] shape: (225, 1, 2)
도형의 갯수: 3(3)
```



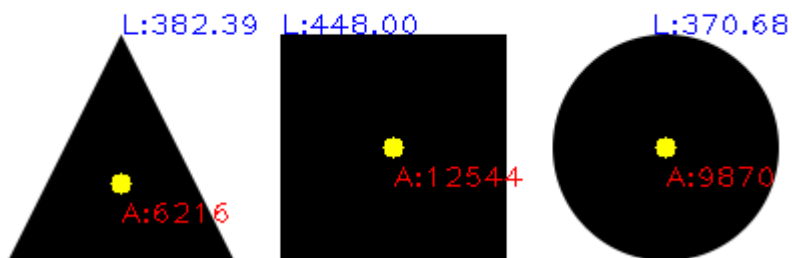
cntr_hierarchy.py

```
3 [[[ 1 -1 -1 -1]
   [ 2  0 -1 -1]
   [-1  1 -1 -1]]]
(1, 3, 4)
(1, 6, 4)
6 [[[ 2 -1  1 -1]
   [-1 -1 -1  0]
   [ 4  0  3 -1]
   [-1 -1 -1  2]
   [-1  2  5 -1]
   [-1 -1 -1  4]]]
```



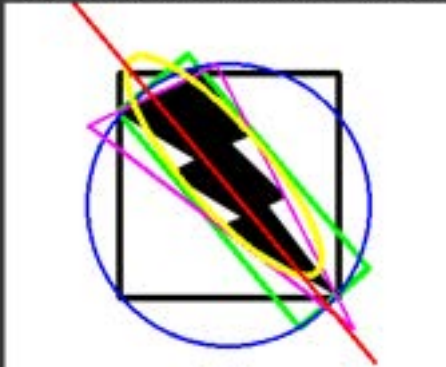
cntr_moment.py

```
area:9870.00
area:12544.00
area:6216.00
```

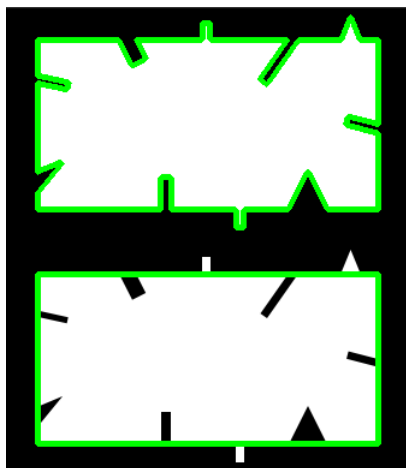


cntr_bound_fit.py

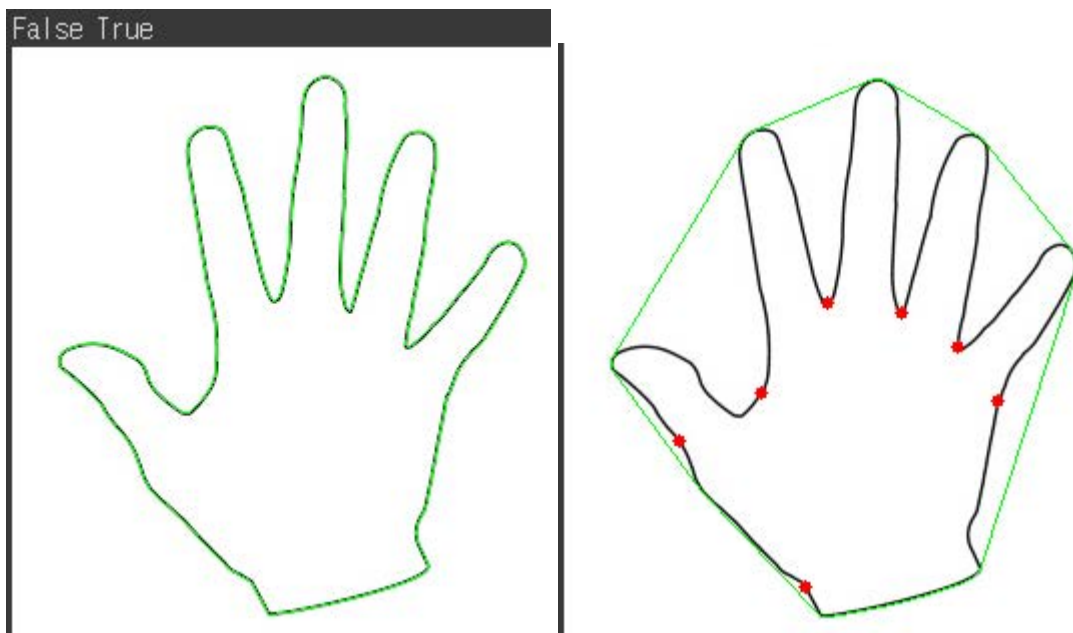
```
<class 'numpy.ndarray'> (332, 1, 2) int32
<(python-input-25-f3c50d7b6dc):22: DeprecationWarning: 'np.int0' is a deprecated alias for 'np.intp'. (Deprecated NumPy 1.24)
    box = np.int0(box) # 정수로 변환
<(python-input-25-f3c50d7b6dc):40: DeprecationWarning: Conversion of an array with ndim > 0 to a scalar is deprecated, and will
    cv2.line(img, (0, int(0-x*(vy/vx) + y)), (cols-1, int((cols-x)*(vy/vx) + y)), #
```



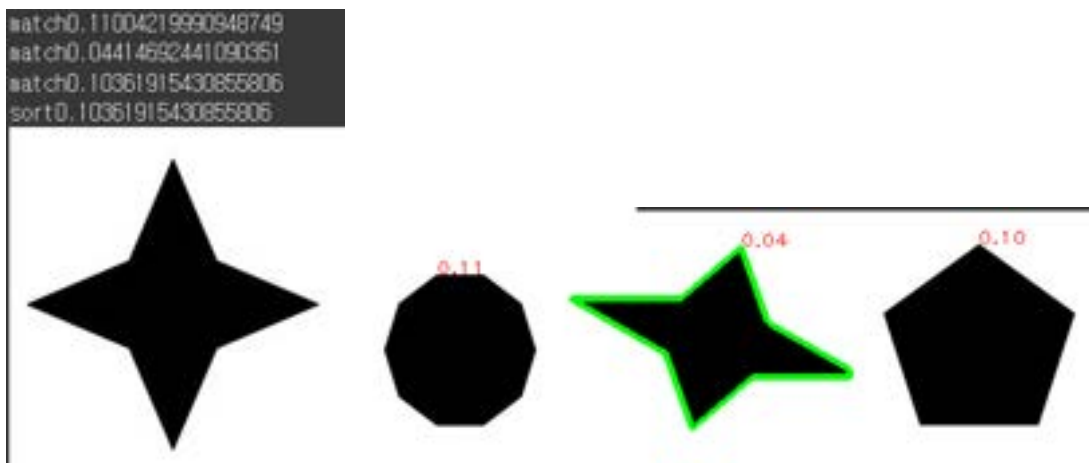
cntr_approximate.py



cntr_convexhull.py



cntr_matchShape.py



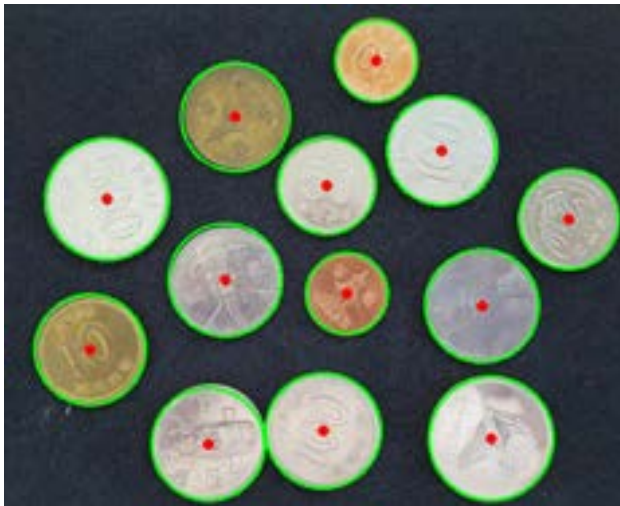
hough_line.py



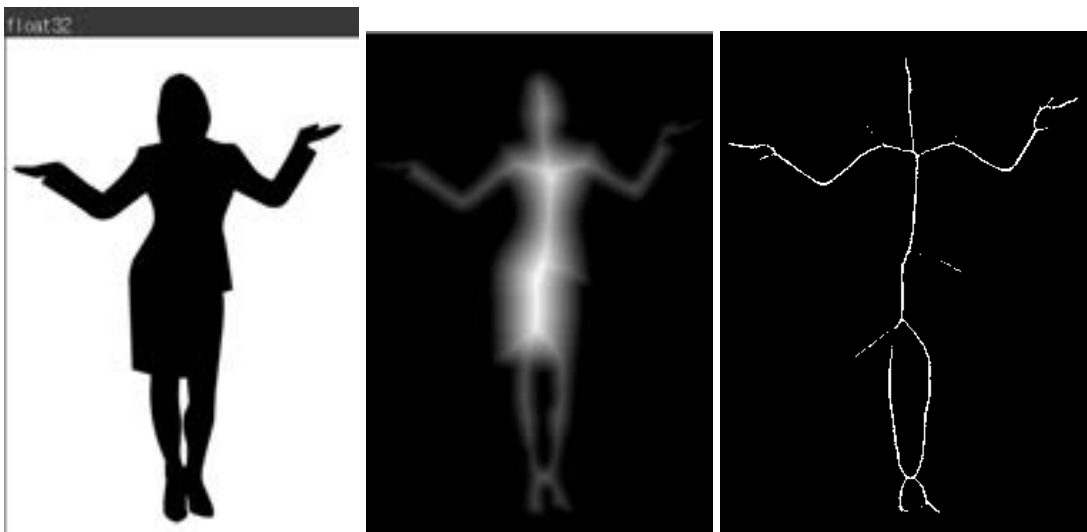
hough_lineP.py



hough_circle.py



distanceTrans.py



connected_label.py



flood_fill.py



watershed.py



grabcut.py

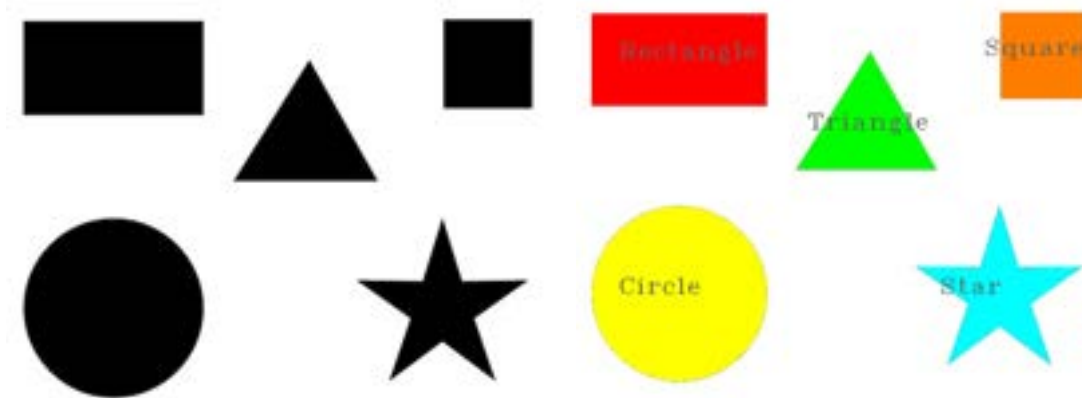


mean_shift.py

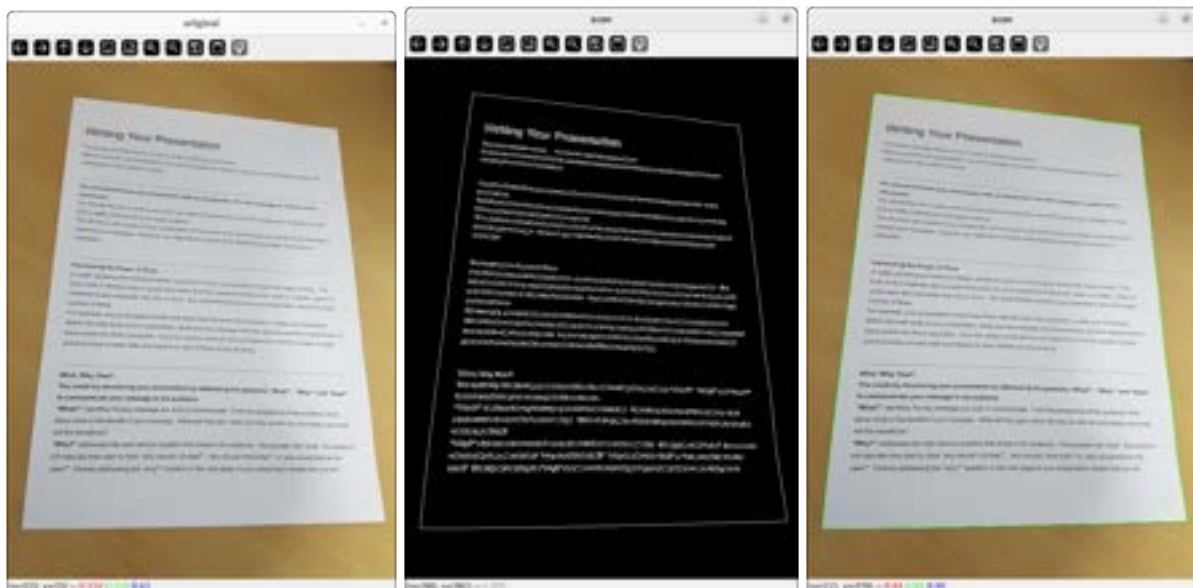


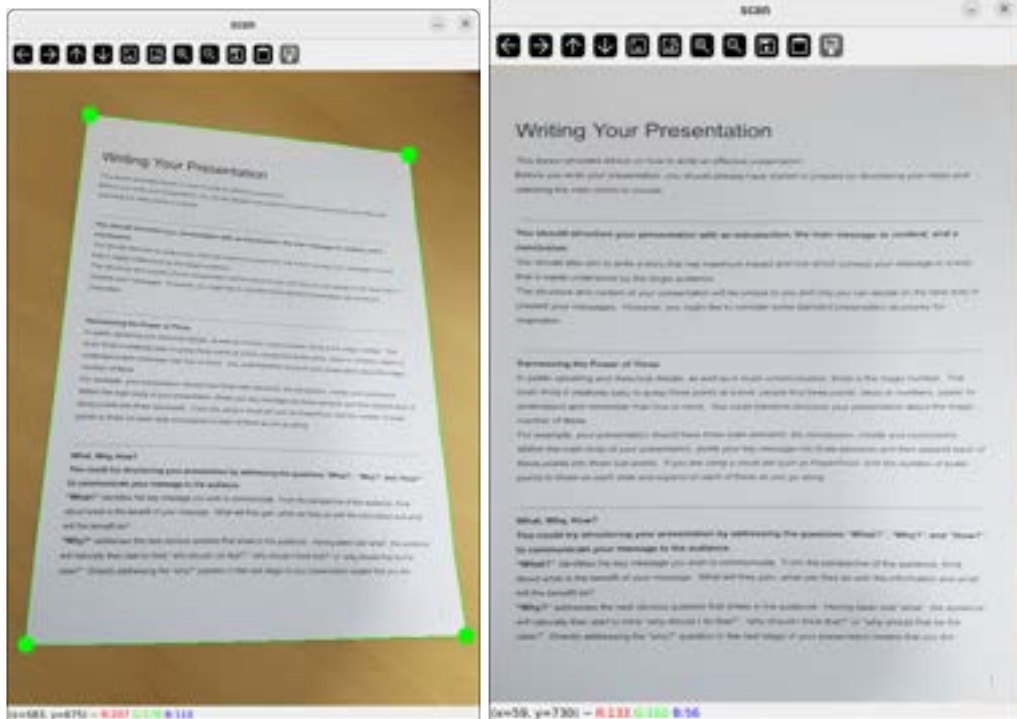
workshop_shape.py

```
vertices: 10  
vertices: 16  
vertices: 3  
vertices: 4  
vertices: 4
```

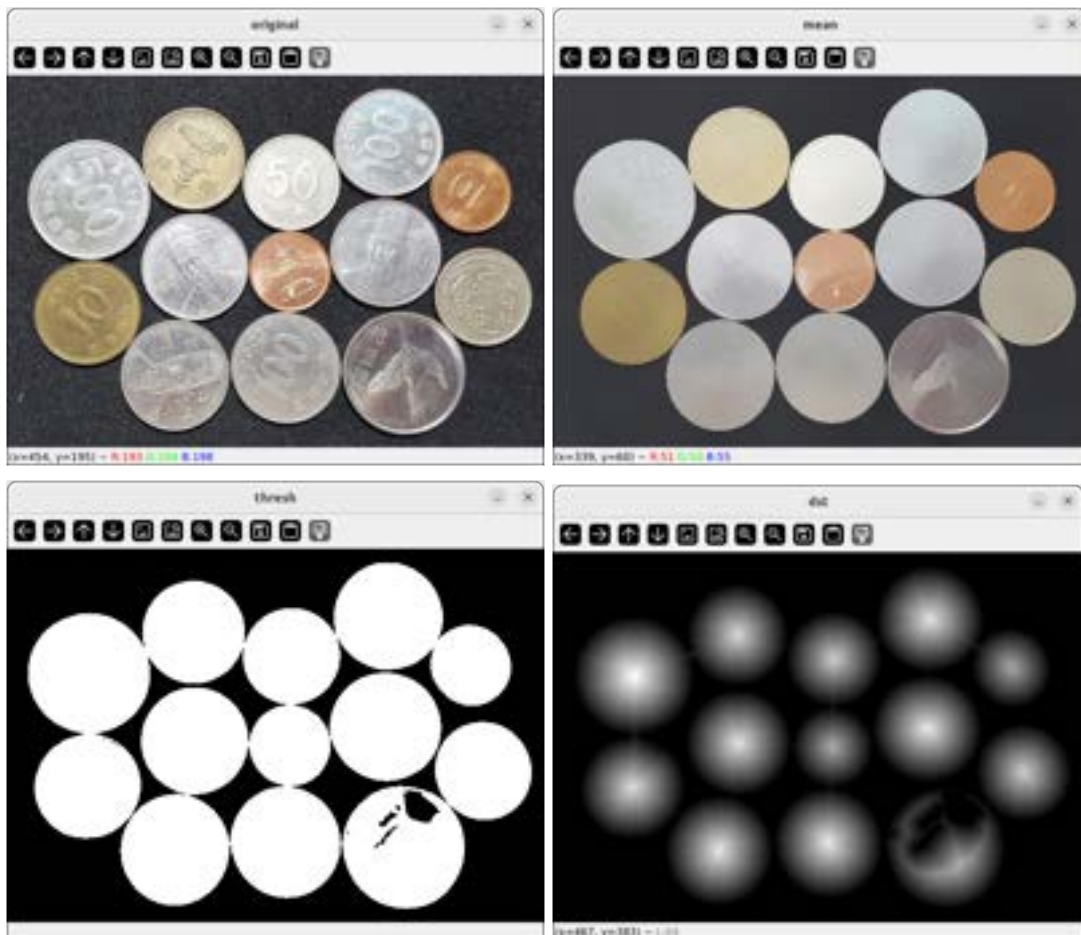


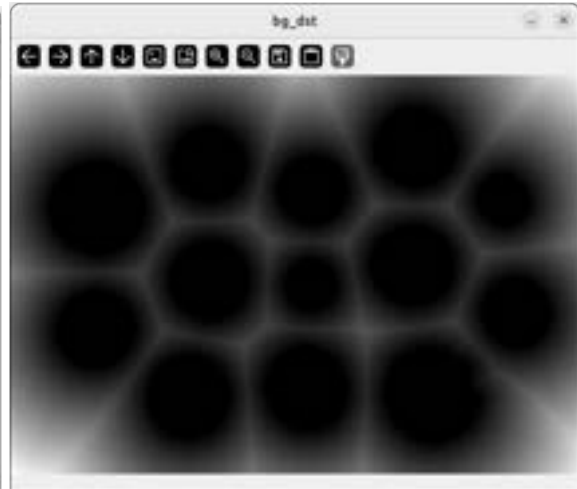
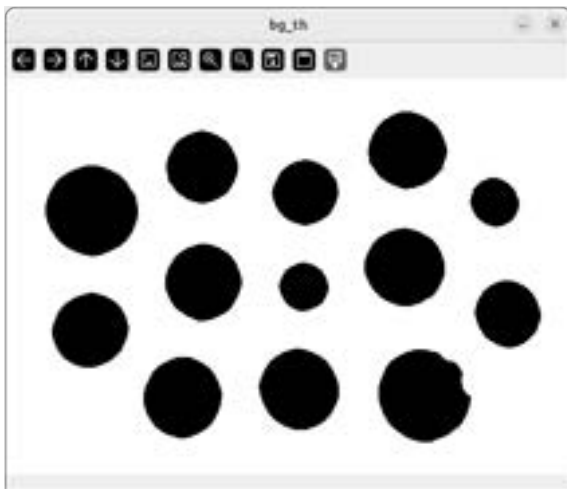
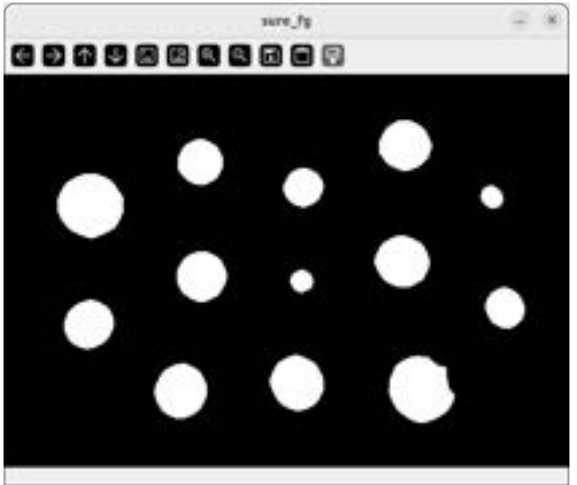
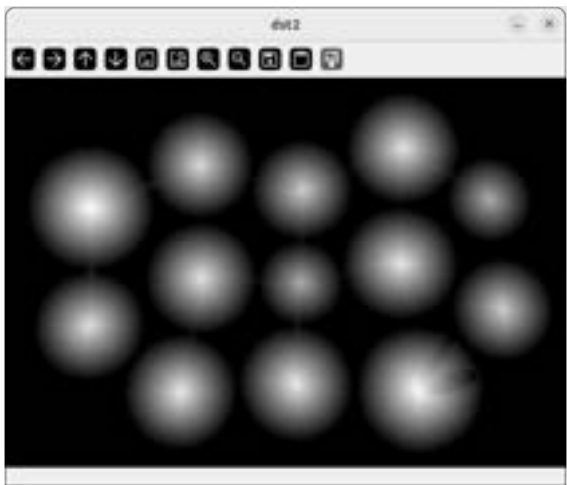
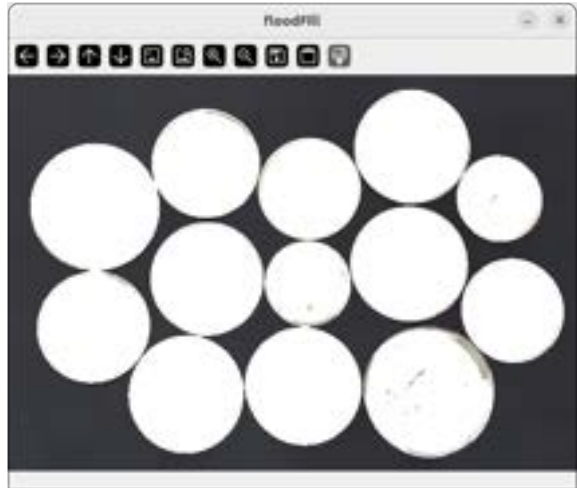
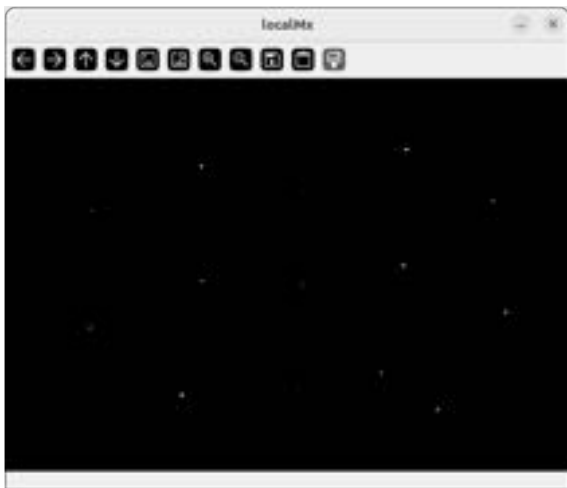
workshop_paper_scan.py

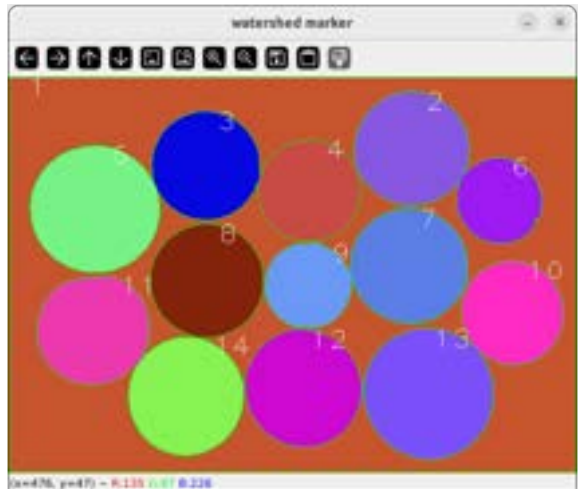
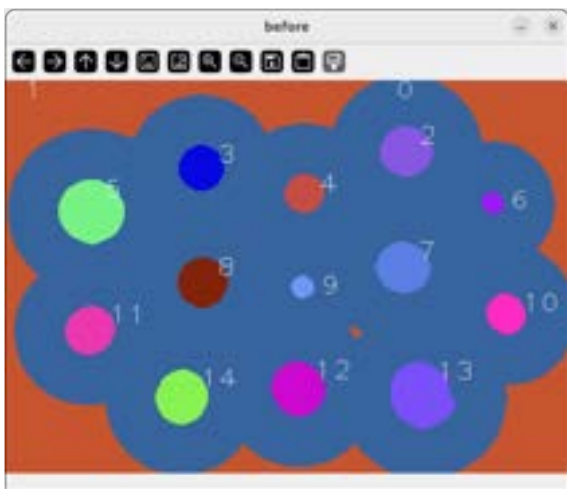
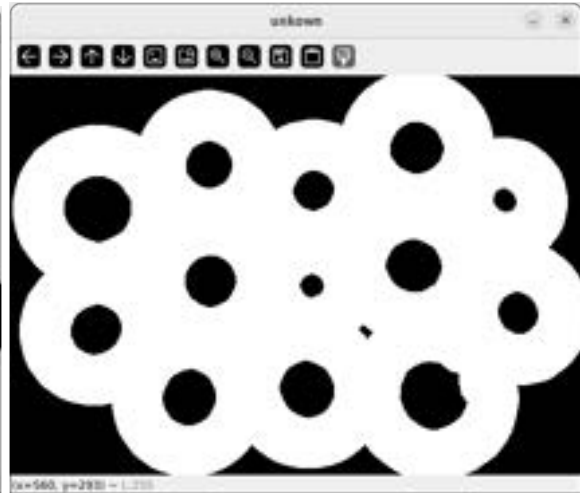
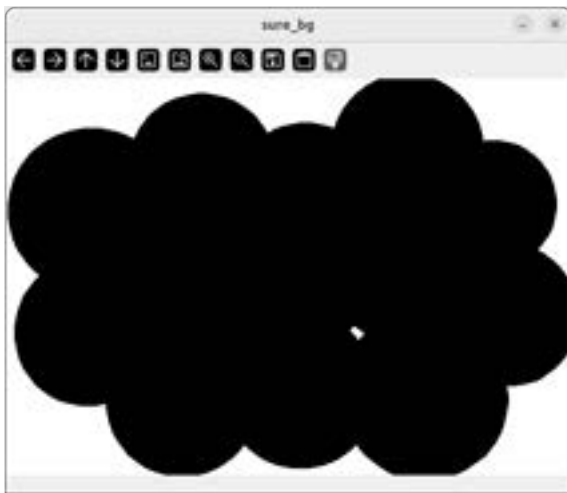


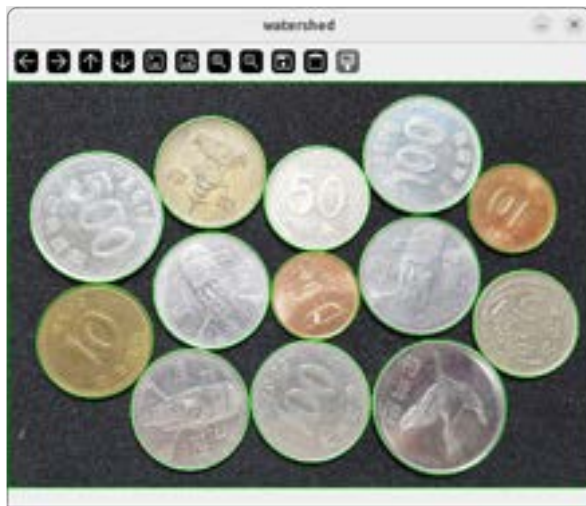


workshop_coin_count.py









강좌_8

avg_hash.py

```
gray.shape : (266, 410), dtype : uint8
avg.shape : (), dtype : float64
type<class 'builtin_function_or_method'>
bin.shape : (16, 16), dtype : int64
157.1796875
[[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 1 0 0 1 1 1 1 1 1]
 [1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1]
 [1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 1]
 [1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1]
 [1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 1]
 [1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
111111111111111111
100000000000000000
100000000000000000
100000000000000000
100000000000000000
1000000100111111
1100000111111111
1100000111111111
1100000001111111
1100001111111111
1100011111111111
1100011111111111
1000011111111111
1000011111111111
1000011111111111
1000011111111111
1000011111111111
1100011111111111
dhash : ffff8000800080008000813fc1ffc07fc3ffc7ffc7ff87ff87ff87ffc7ff
```



리스트 내포

```
[12] a = [1, 2, 3, 4]
     result = []
     for num in a:
         result.append(num + 3)
     print(result)

[13] a = [1, 2, 3, 4]
     result = [num + 3 for num in a]
     print(result)

[14] a = [1, 2, 3, 4]
     result = []
     for num in a:
         if num % 2 == 0:
             result.append(num + 3)
     print(result)

[15] a = [1, 2, 3, 4]
     result = [num + 3 for num in a if num % 2 == 0]
     print(result)
```

join

```
✓ join

[16] a = '.'
      print(a.join('abcd'))

⇌ a,b,c,d

[17] print('a'.join('abcd'))

⇌ aabacad

[18] print(''.join('abcd'))

⇌ abcd
```

tolist

```
tolist

import numpy as np
bin = np.array([0, 0, 1, 1, 0, 0, 1])
bin = bin.tolist()
print(bin)
print(bin)

[[ 0  0  1  1  0  0  1]]
[0, 0, 1, 1, 0, 0, 1]

''.join(str(i) for i in bin)

'0011001'

# 2진수 문자열을 10진수 문자열로 변환 --->
bin = np.array(['0000111', 1, '1110000', 1, '1111111', 1, '0000000', 11])
dash = []
for row in bin.tolist():
    # bin의 shape 배열일 때, bin.tolist()는 배열 배열형(2D)을 가진 Python 리스트로 변환
    s = ''.join(str(i) for i in row)
    print(s)
    dash.append(s+'\n')
dash = ''.join(dash)
print('dash : %s'%dash)
```

```
0000111
1110000
1111111
0000000
dash : 0000111
1110000
1111111
0000000
```

avg_hash_matching.py



template_matching.py





corner_harris.py



corner_goodFeature.py



kpt_gfft.py



kpt_fast.py



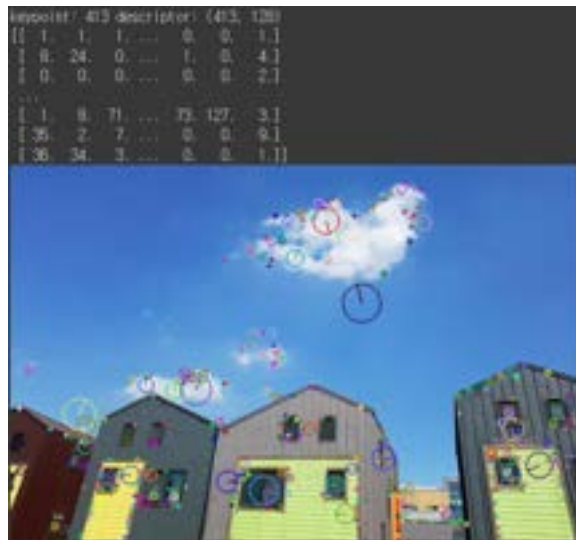
kpt_blob.py



kpt_blob_param.py



desc_sift.py



desc_surf.py

error

desc_orb.py



match_bf_sift.py



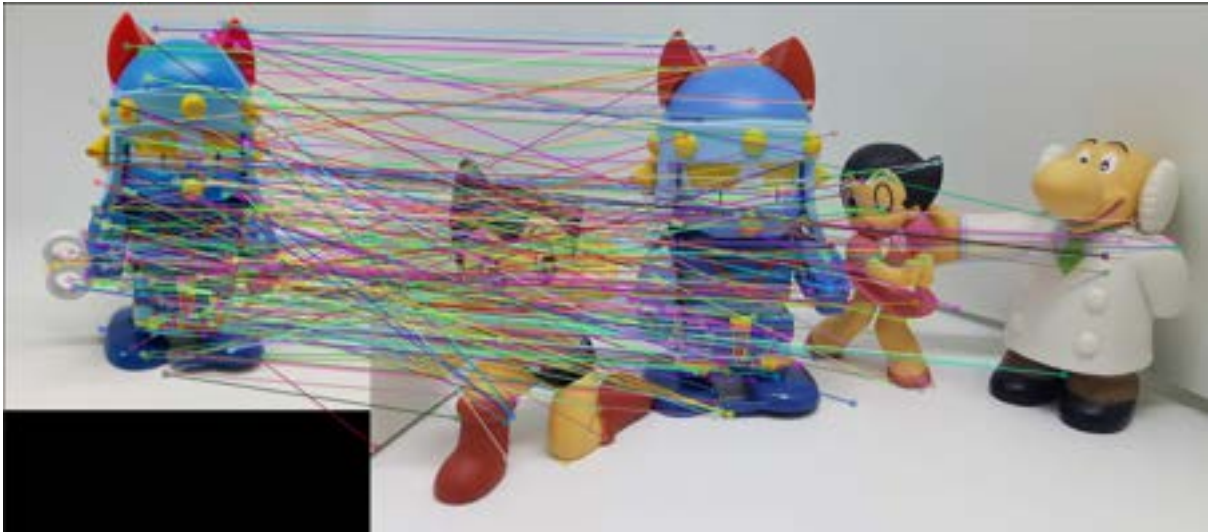
match_bf_surf.py

error

match_bf_orb.py



match_flann_sift.py



match_flann_surf.py

error

match_flann_orb.py



match_good.py



match_good_knn.py

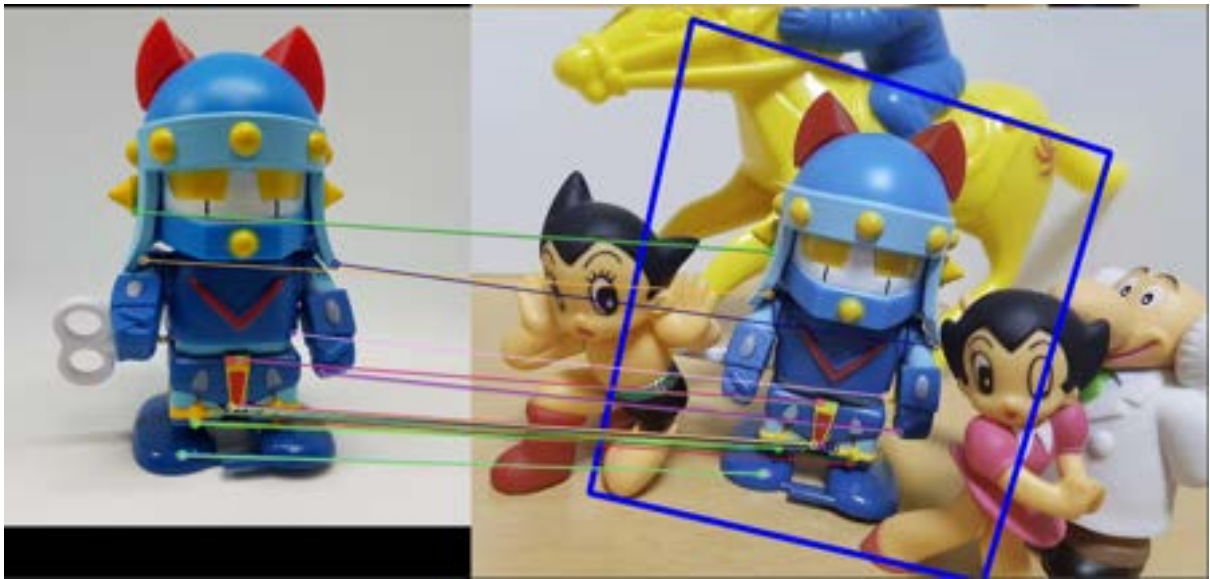


match_homography.py

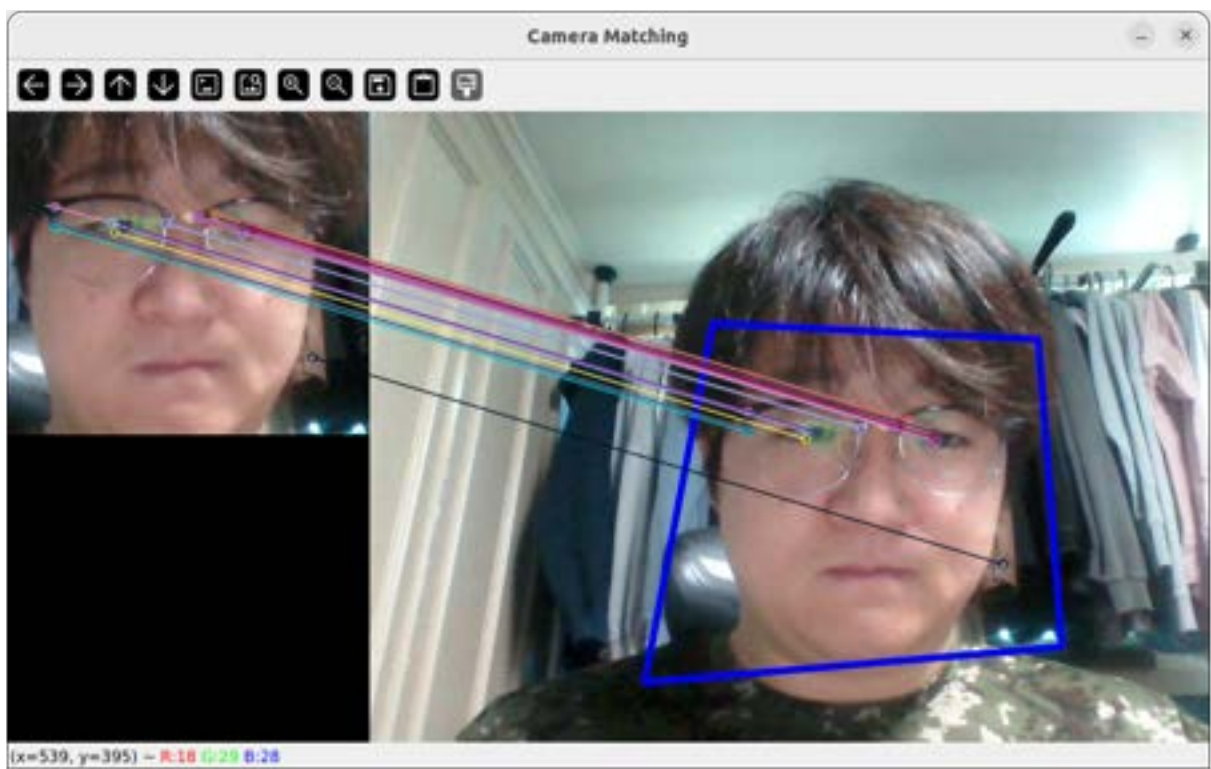


match_homography_accuracy.py

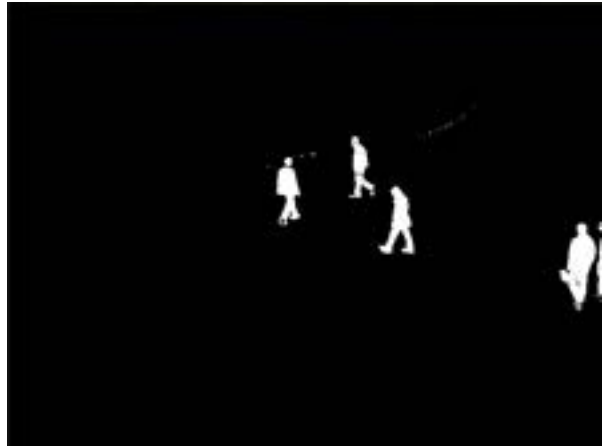




match_camera.py



track_bgsub_mog.py



track_bgsub_mog2.py



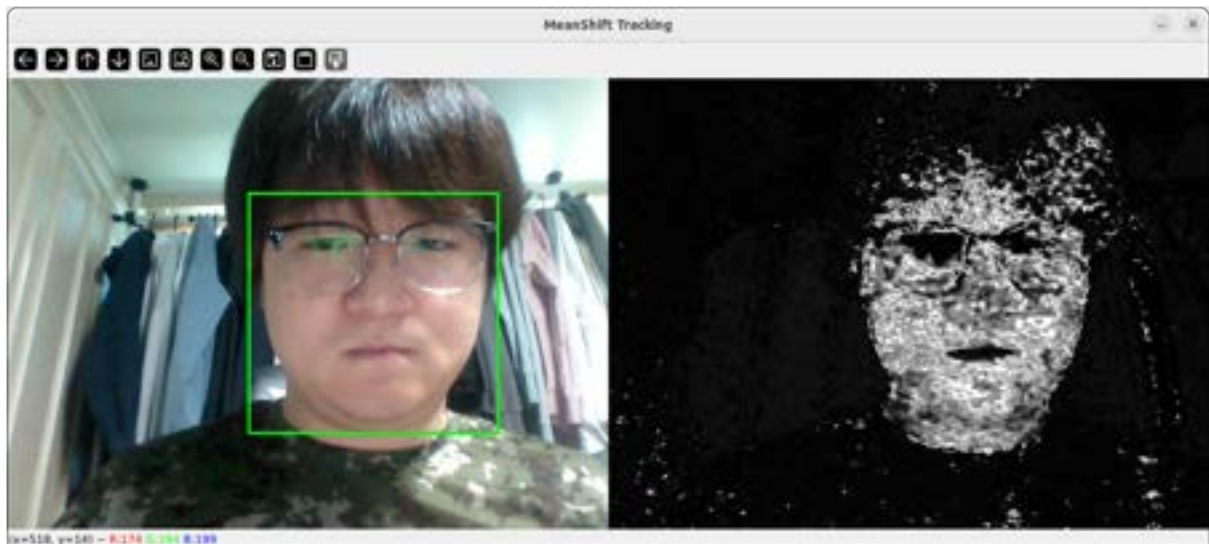
track_opticalLK.py



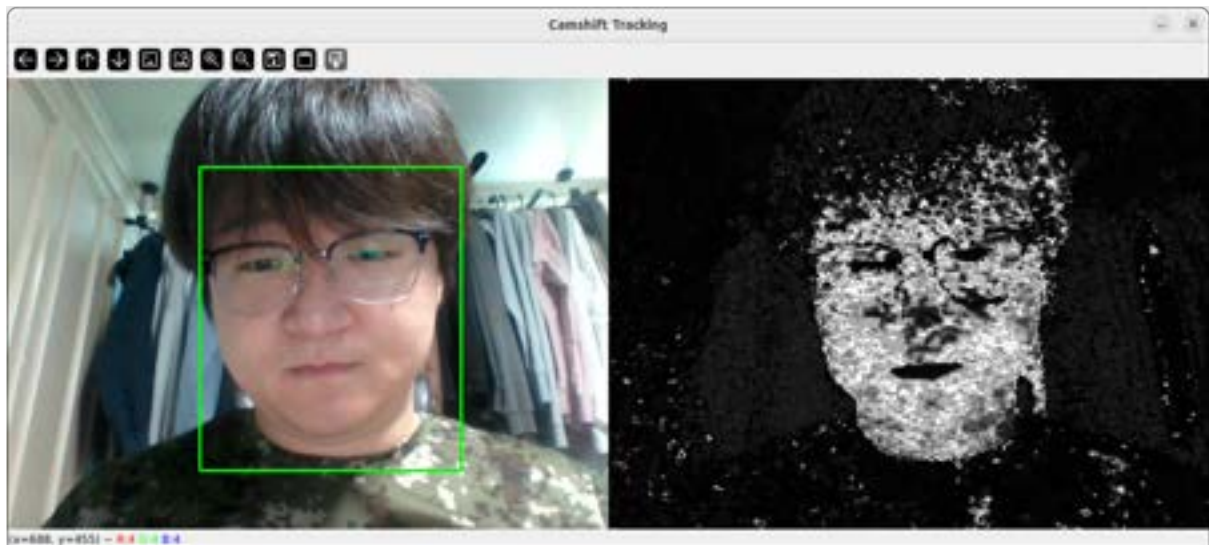
track_optical_farneback.py



track_meanshift_cam.py



track_camsifht_cam.py



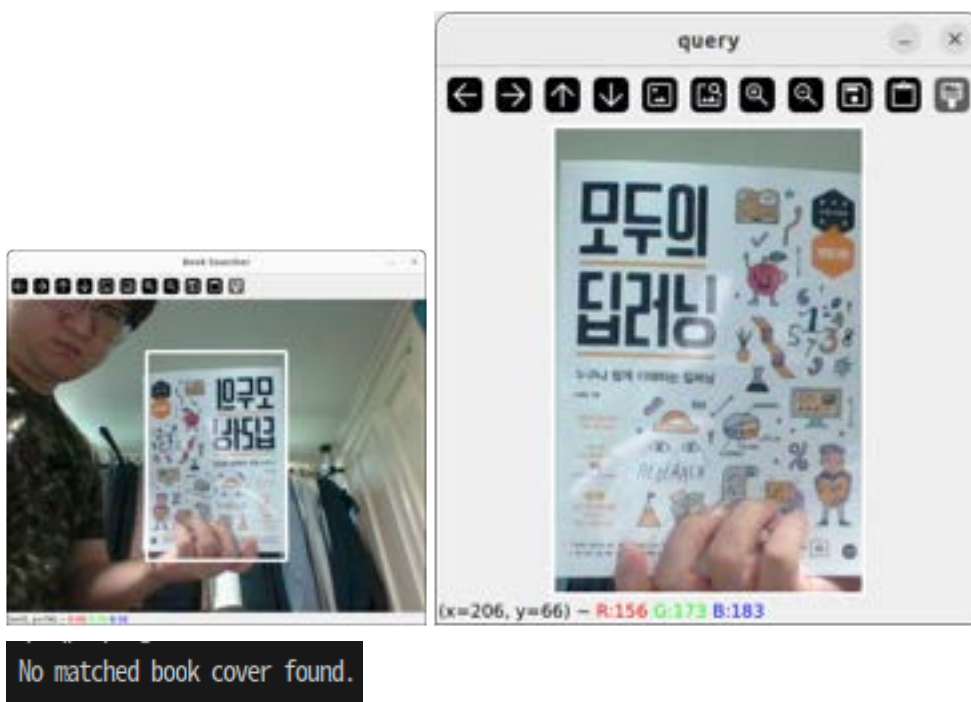
track_trackingAPI.py



workshop_panorama.py

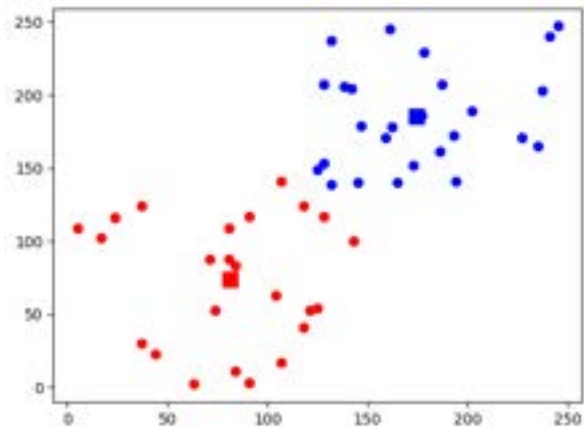


workshop_booksearcher.py



강좌_9

k-means_random.py

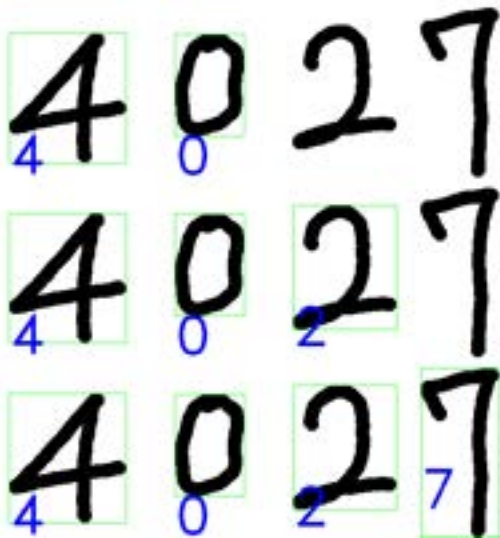


k-means_color.py

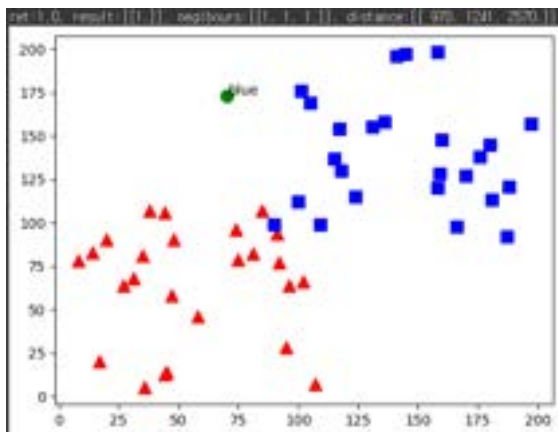
```
[[ 18  17 158]
 [ 95 137 178]
 [ 63  65 182]
 [107 108  97]
 [198 163 108]
 [196 126  43]
 [ 98  51  22]
 [ 28  24  97]
 [146 120  72]
 [151 152 150]
 [189 191 190]
 [135  77  27]
 [209 212 213]
 [ 56  24  10]
 [ 14 153 189]
 [ 12 109 137]]
```



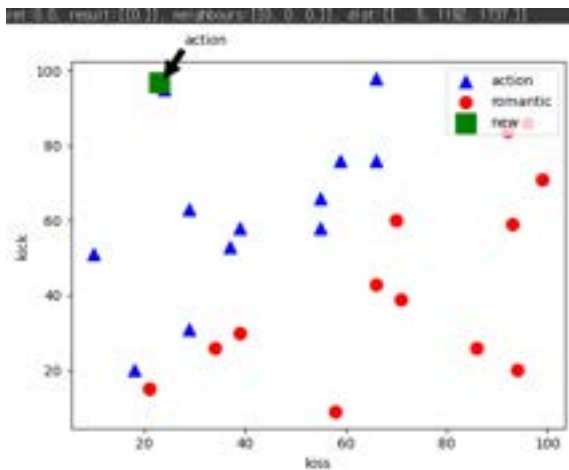
k-means_handwritten.py



kNN_random.py



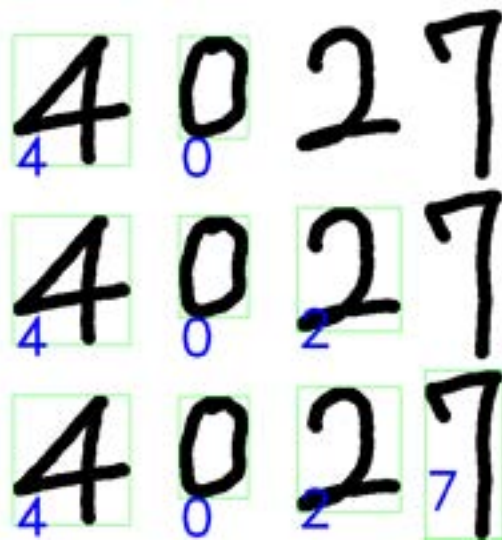
kNN_movie.py



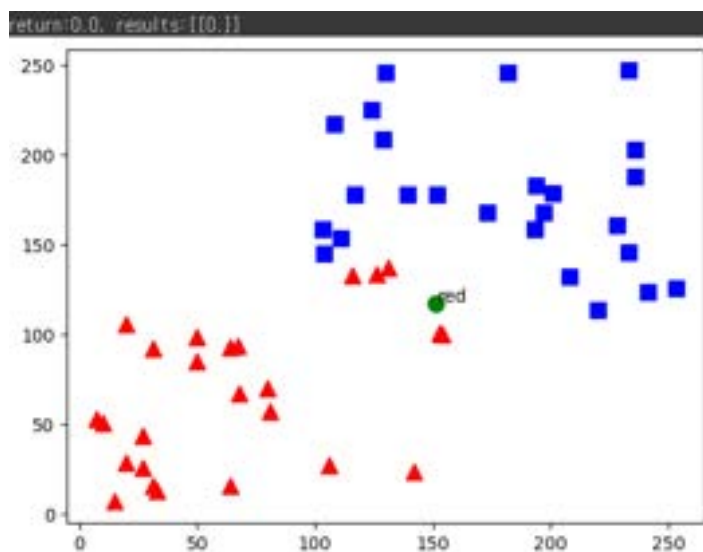
kNN_mnist.py

```
K:1, Accuracy :95.40%(477/500)
K:2, Accuracy :94.40%(472/500)
K:3, Accuracy :95.00%(475/500)
K:4, Accuracy :94.40%(472/500)
K:5, Accuracy :94.20%(471/500)
K:6, Accuracy :94.20%(471/500)
K:7, Accuracy :94.40%(472/500)
K:8, Accuracy :93.60%(468/500)
K:9, Accuracy :93.40%(467/500)
K:10, Accuracy :93.00%(465/500)
```

kNN_handwritten.py



svm_random.py

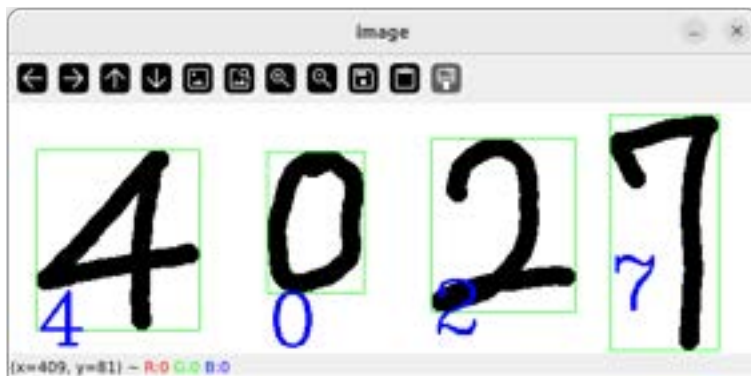


svm_random.xml

svm_mnist_hog_train.py

```
SVM training started...train data: (50, 90, 324)
SVM training complete. 17.75 Min
Accuracy: 98.80%
```

svm_handwritten.py



svm_hog_pedestrian.py



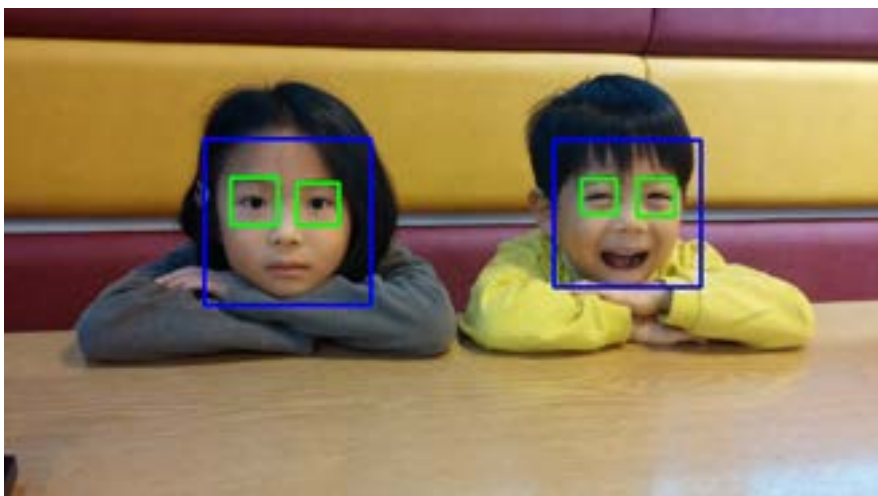
bow_plane_bike_train.py

```
Adding descriptor to BOWTrainer...
  airplanes: 800/800(100.00%)
  Motorbikes: 798/798(100.00%)
Adding descriptor completed...
Starting Dictionary clustering(50)...           It will take several time...
Dictionary Clustering completed...dictionary shape: (50, 128)
Compute histograms training set...(100.00%)
ova items: 1598/50
ova training...
ova training completed.
Training Elapsed: 00:07:20
Accuracy(Self)
  airplanes: 95.12 %
  Motorbikes: 95.74 %
```

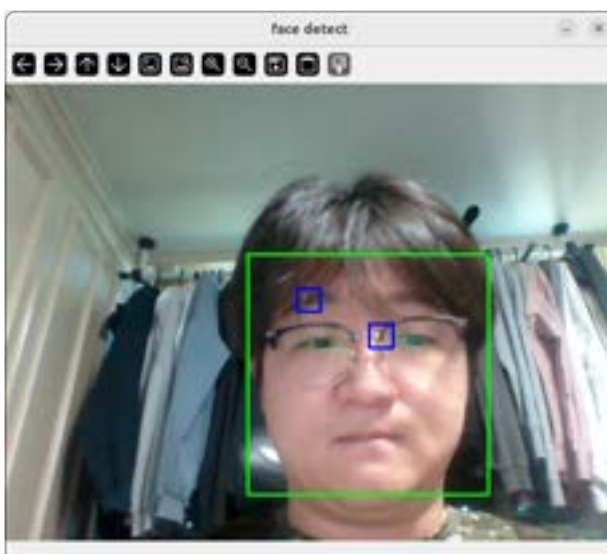
bow_plane_bike_test.py



haar_face.py



haar_face_cam.py



lbp_face1_collect.py



lbp_face2_train.py

```
Collecting train data set:  
    path:../faces/a_1, 400files  
Starting LBP Model training...  
Model trained successfully!
```

lbp_face3_recognize.py



workshop_face_mosaic.py



workshop_hannibal_mask.py



workshop_face_distortion_camera.py

