

2. Трансформації (Transformations)

1. Making your own linear filter (Створення власних лінійних фільтрів!)

код:

```
import cv2
import numpy as np
img=cv2.imread("/home/rodion/yulilia0/aboba/cv/trans/dog.jpg")
cv2.imshow ("original", img)

ker = np.ones((3, 3), dtype=np.float32)
ker /= (3 * 3)
dst = cv2.filter2D (img, -1, ker)
cv2.imshow ("ker 3", dst)

ker = np.ones((7, 7), dtype=np.float32)
ker /= (7 * 7)
dst = cv2.filter2D (img, -1, ker)
cv2.imshow ("ker 7", dst)

ker = np.ones((11, 11), dtype=np.float32)
ker /= (11 * 11)
dst = cv2.filter2D (img, -1, ker)
cv2.imshow ("ker 11", dst)

cv2.waitKey (0)
```

результат:



2. Додавання рамок до зображень

код:

```
import cv2
```

```


cv2.imshow ("original", img)

bottom = top = int(0.05 * img.shape[0])
right = left = int(0.05 * img.shape[1])
dst = cv2.copyMakeBorder(img, top, bottom, left,
right, cv2.BORDER_CONSTANT, None, [150,150,10])
cv2.imshow("ker 11", dst)

dst = cv2.copyMakeBorder(img, top, bottom, left,
right, cv2.BORDER_REPLICATE, None, [100,150,150])
cv2.imshow("ker 12", dst)

cv2.waitKey(0)

```

результат:



(не влізало зображення і аби було видно всю рамку кинула так)

3. Похідні Sobel (Sobel Derivatives)

КОД:

```
import cv2

img=cv2.imread("/home/rodion/yuliiia0/aboba/cv/trans/my.jpg")

img =cv2.GaussianBlur(img, (3, 3),0)
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
x = cv2.Sobel(img, cv2.CV_8U, 1, 0, 3, 0, cv2.BORDER_REPLICATE)
y = cv2.Sobel(img, cv2.CV_8U, 0, 1, 3, 0, cv2.BORDER_REPLICATE)
grad = cv2.addWeighted(x, 0.5, y, 0.5, 0)
cv2.imshow("2", grad)
cv2.waitKey(0)
```

результат:



4. Оператор Лапласа (Laplace Operator)

Код:

```
import cv2

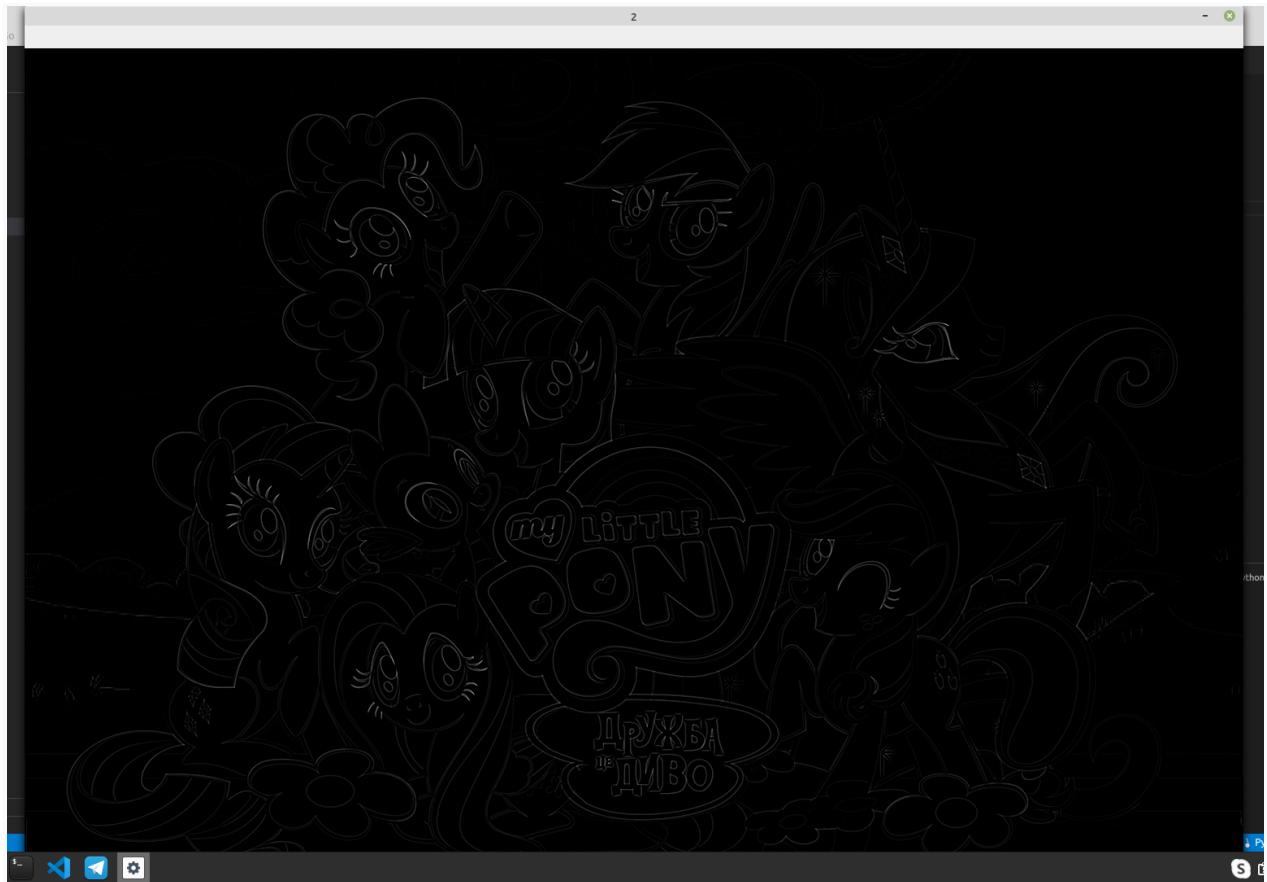
img=cv2.imread("/home/rodion/yuliiia0/aboba/cv/trans/my.jpg")

img =cv2.GaussianBlur(img, (3, 3),0)
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

dst = cv2.Laplacian(img, cv2.CV_8U, 3)

dst = cv2.convertScaleAbs(dst)
cv2.imshow("2", dst)
cv2.waitKey(0)
```

Результат:



5. Canny Edge Detector

Код:

```
import cv2
img=cv2.imread("/home/rodion/yulilia0/aboba/cv/trans/my.jpg")
img =cv2.GaussianBlur(img, (3, 3),0)
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
det = cv2.Canny(img, 5, 5*3, 3)

cv2.imshow("2", det)
cv2.waitKey(0)
```

Результат:



6. Трансформація лінії Хафа (Hough Line Transform)

КОД:

```
import cv2
import numpy as np
import math

img=cv2.imread("/home/rodion/yuliia0/aboba/cv/trans/my.jpg")
cv2.imshow("original", img)
img = cv2.GaussianBlur(img, (3, 3),0)
img=g = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img = cv2.Canny(img, 50, 200, 3)

lines = cv2.HoughLines(img, 1, np.pi / 180, 150, None)

for i in range(0, len(lines)):
    a = math.cos(lines[i][0][1])
    b = math.sin(lines[i][0][1])
    x0 = a * lines[i][0][0]
    y0 = b * lines[i][0][1]
    pt1 = (int(x0 + 1000*(-b)), int(y0 + 1000*(a)))
    pt2 = (int(x0 - 1000*(-b)), int(y0 - 1000*(a)))

    cv2.line (g, pt1, pt2, (0,0,255), 3, cv2.LINE_AA)
linesP = cv2.HoughLinesP(img, 1, np.pi / 180, 50, None, 50, 10)
for i in range(0, len(linesP)):
    l = linesP[i][0]
    cv2.line (g, (l[0], l[1]), (l[2], l[3]), (0,0,0), 3, cv2.LINE_AA)
    cv2.imshow("Standard Hough Line Transform", g)
    cv2.imshow("Probabilistic Line Transform", g)

cv2.waitKey(0)
```

результат:



7. Перетворення кола Хафа (Hough Circle Transform)

Код:

```
import cv2

img=g=cv2.imread("/home/rodion/yuliia0/aboba/cv/trans/tr.jpg")
cv2.imshow("original", img)

img = cv2.GaussianBlur(img, (3, 3),0)
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img = cv2.resize(img, (1280, 720))
g = cv2.resize(g, (1280, 720))
rows = img.shape[0]
circles = cv2.HoughCircles(img, cv2.HOUGH_GRADIENT, 1, rows / 8,
param1=100, param2=30, minRadius=1, maxRadius=30000)

for i in circles[0, :]:
    cv2.circle(g, (int(i[0]), int(i[1])), int(i[2]), (0, 0, 255), 3)

cv2.imshow("2", g)
cv2.waitKey(0)
```

Результат:

