МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе № 3.10

по дисциплине «Технологии распознавания образов»

Выполнил студент группы ПИЖ-б-о-21-1
Трушева В. О.____.«__»_____2023г.
Подпись студента_____
Работа защищена « »_____20__г.
Проверила Воронкин Р.А. _____
(подпись)

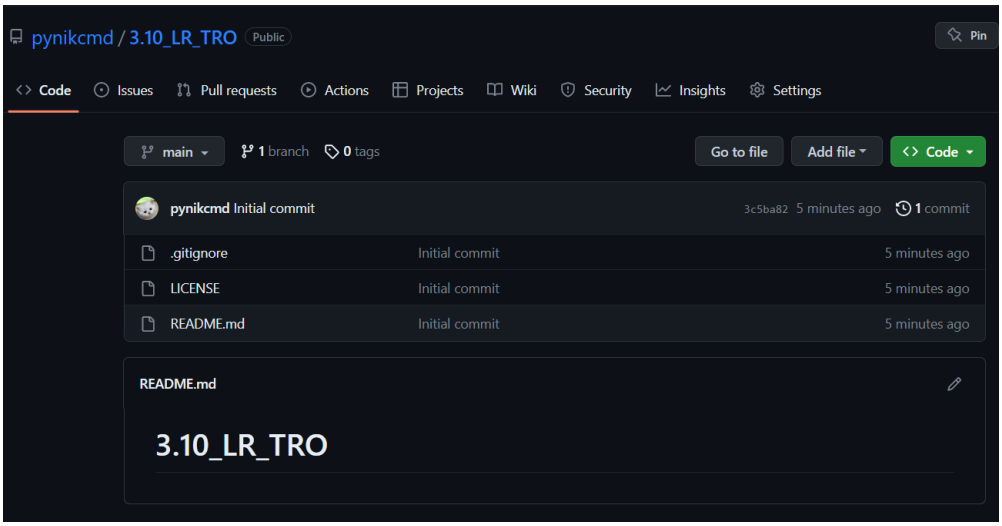
Ставрополь 2023

Цель работы: изучить основные операции геометрических преобразований изображений, такие как изменение размера, сдвиг, вращение, аффинное преобразование и т. д

Методика и порядок выполнения работы

1. Изучить теоретический материал работы.

2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования (выбор языка программирования будет доступен после установки флажка Add .gitignore).



3. Выполните клонирование созданного репозитория на рабочий компьютер.

```
D:\fgit>git clone https://github.com/pynikcmd/3.10_LR_TRO.git
Cloning into '3.10_LR_TRO'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

4. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.

```
D:\fgit\3.10_LR_TRO>git flow init

Which branch should be used for bringing forth production releases?
   - main
Branch name for production releases: [main] Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [D:/fgit/3.10_LR_TRO/.git/hooks]
```

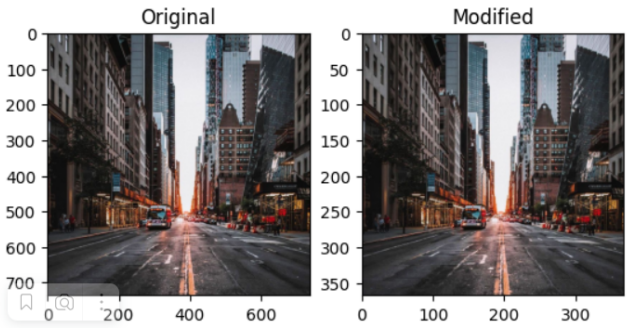
5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.

6. Проработать примеры лабораторной работы в отдельном ноутбуке.

Первый способ изменения размера задается в процентах

In [10]:
```
img = cv2.imread('city.jpg', 1)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
scale_percent = 50  # процент изменения
width = int(img.shape[1] * scale_percent / 100)
height = int(img.shape[0] * scale_percent / 100)
dim = (width, height)

resized = cv2.resize(img, dim, interpolation=cv2.INTER_AREA)
print('Resized Dimensions : ', resized.shape)
img_print(img, resized);
```

```
Resized Dimensions :  (368, 368, 3)
```

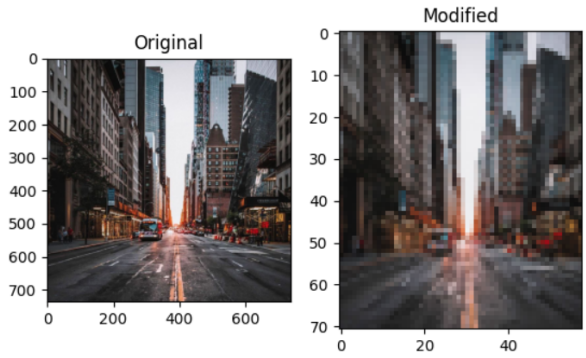


Пример 1.1

Второй способ изменения размера задается вручную

In [11]:
```
print('Original Dimensions : ',img.shape)
width = 58
height = 71
dim1 = (width, height)

# resize image
resized1 = cv2.resize(img, dim1, interpolation=cv2.INTER_AREA)
print('Resized Dimensions : ', resized1.shape)
img_print(img, resized1);
```

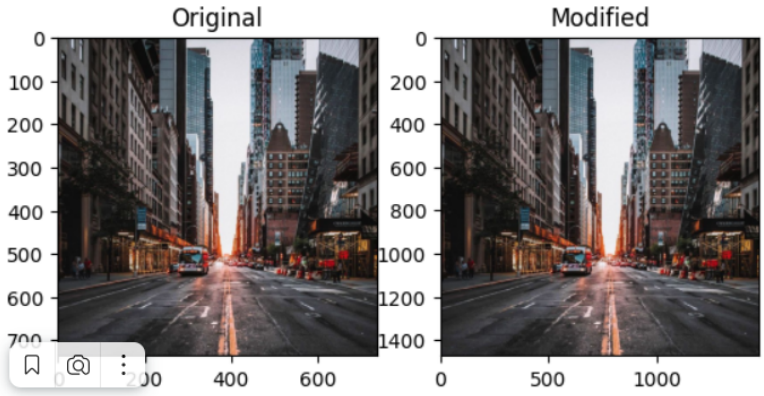
```
Original Dimensions :  (736, 736, 3)
Resized Dimensions :  (71, 58, 3)
```



Пример 1.2

Третий способ: задается коэффициентом масштабирования

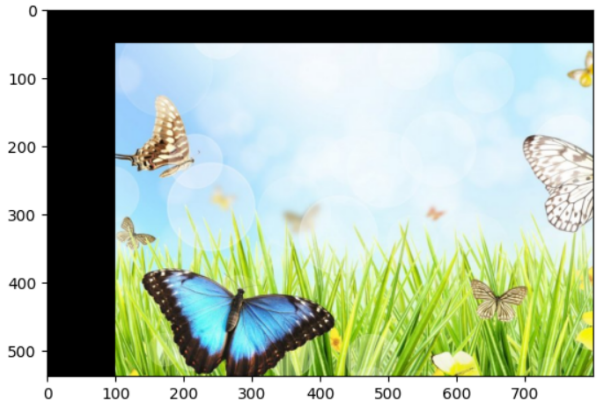
In [12]:
```
res = cv2.resize(img,None,fx=2, fy=2, interpolation = cv2.INTER_CUBIC)
img_print(img, res);
```



Пример 1.3

Задание 4.2. Определить размер изображения и сдвинуть изображение на 100 столбцов и 50 строк.

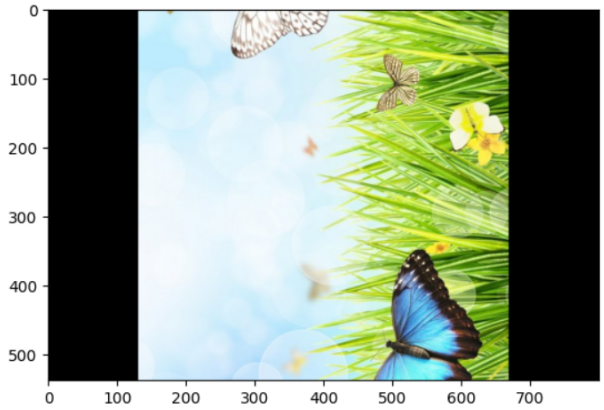
In [21]:
```
img2 = cv2.imread('pic.jpg',1)
img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)
rows,cols,colors = img2.shape
M = np.float32([[1,0,100],[0,1,50]])
dst = cv2.warpAffine(img2,M,(cols,rows))
plt.imshow(dst);
```



Пример 2

Задание 4.3. Определить размер изображения, его центр и повернуть его на 90 градусов.

```
M = cv2.getRotationMatrix2D((cols/2,rows/2),90,1)
dst = cv2.warpAffine(img2,M,(cols,rows))
plt.imshow(dst);
```



Пример 3

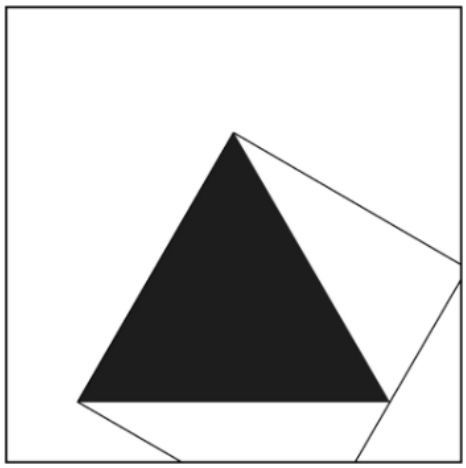
Задание 4.4. Определить размер изображения, задать 3 точки, изменить их координаты и провести аффинное преобразование всего изображения по этим точкам.

```
img3 = cv2.imread('flower.jpeg',1)
img3 = cv2.cvtColor(img3, cv2.COLOR_BGR2RGB)

pts1 = np.float32([[50,50],[200,50],[50,200]])
pts2 = np.float32([[10,100],[200,50],[100,250]])

M = cv2.getAffineTransform(pts1,pts2)
dst = cv2.warpAffine(img3,M,(cols,rows))

plt.subplot(121),plt.imshow(img3),plt.title('Input')
plt.subplot(122),plt.imshow(dst),plt.title('Output')
plt.show()
```

Пример 4

Задание 4.5. Провести охват изображения в прямоугольник, повернутый так, чтобы площадь этого прямоугольника была минимальной

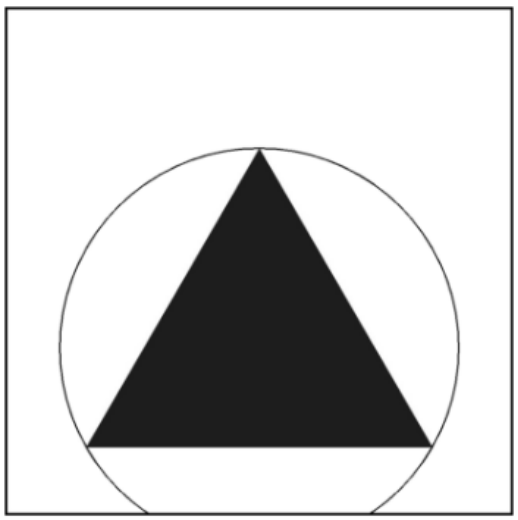
```
img4 = cv2.imread('geo.jpeg',0)
ret, thresh = cv2.threshold(img4, 127, 255, cv2.THRESH_BINARY)
contours, hierarchy = cv2.findContours(thresh, 1, 1)

cnt = contours[0]
rect = cv2.minAreaRect(cnt)

box = cv2.boxPoints(rect)
box = np.int_(box)

imp = cv2.drawContours(img4, [box], 0, (0, 0, 255), 2)
imp = cv2.cvtColor(imp, cv2.COLOR_BGR2RGB)

plt.axis('off')
plt.imshow(imp);
```



Пример 5

Задание 4.6. Провести охват изображения в круг

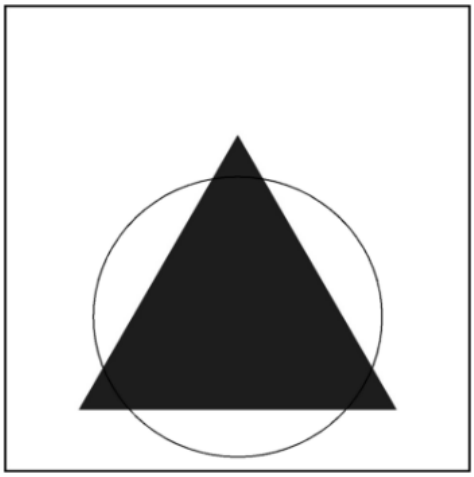
```
image = cv2.imread('geo.jpeg',0)

ret, thresh = cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)
contours, hierarchy = cv2.findContours(thresh, 1, 1)
cnt = contours[0]
rect = cv2.minAreaRect(cnt)
(x,y),radius = cv2.minEnclosingCircle(cnt)

center = (int(x),int(y))
radius = int(radius)


image = cv2.circle(image,center,radius,(0,255,0),2)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

plt.axis('off')
plt.imshow(image);
```



Пример 6

Задание 4.7. Провести охват изображения в эллипс, повернутый так, чтобы площадь этого эллипса была минимальной.

```
img5 = cv2.imread('geo.jpeg',0)

ellipse = cv2.fitEllipse(cnt)
imag = cv2.ellipse(img5,ellipse,(0,255,0),2)

imag = cv2.cvtColor(img5, cv2.COLOR_BGR2RGB)
plt.axis('off')
plt.imshow(imag);
```



Пример 7

Задание 4.8. Провести прямую линию вдоль оси симметрии изображения.

```
img6 = cv2.imread('geo2.png',0)

ret, thresh = cv2.threshold(img6, 127, 255, cv2.THRESH_BINARY)
contours, hierarchy = cv2.findContours(thresh, 1, 1)

cnt = contours[0]
rows,cols = img6.shape[:2]

[vx,vy,x,y] = cv2.fitLine(cnt, cv2.DIST_L2,0,0.01,0.01)

lefty = int((-x*vy/vx) + y)
righty = int(((cols-x)*vy/vx)+y)

img = cv2.line(img6,(cols-1,righty),(0,lefty),(100,255,100),2)
img = cv2.cvtColor(img6, cv2.COLOR_BGR2RGB)
plt.axis('off')
plt.imshow(img);
```

Пример 8

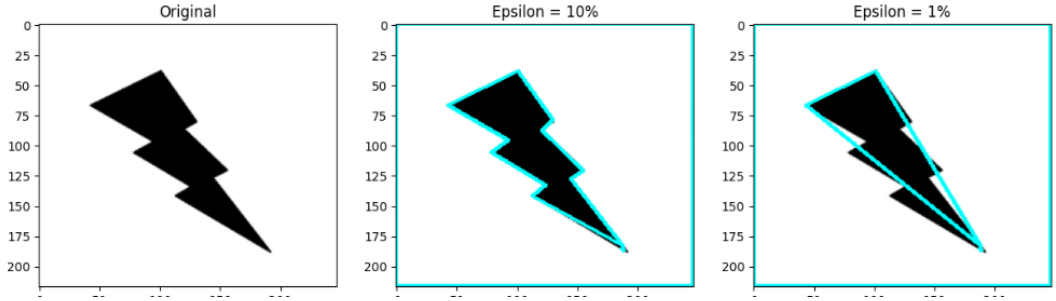
Задание 4.9. Нарисовать контур, охватывающий изображение, толщиной 2, вывести полученное изображение на экран.

```
img7 = cv2.imread('mol.png',1)

gray = cv2.cvtColor(img7,cv2.COLOR_BGR2GRAY)
edges = cv2.Canny(gray, 50,200)
contours, hierarchy= cv2.findContours(edges.copy(),
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
for cnt in contours:
    hull= cv2.convexHull(cnt)
    cv2.drawContours(img7, [hull],0,(100,255,255),4)
plt.axis('off')
plt.imshow(img7);
```

Пример 9

Задание 4.10. Выполнить аппроксимацию контура, полагая epsilon =1%, epsilon=5% и epsilon=10%.

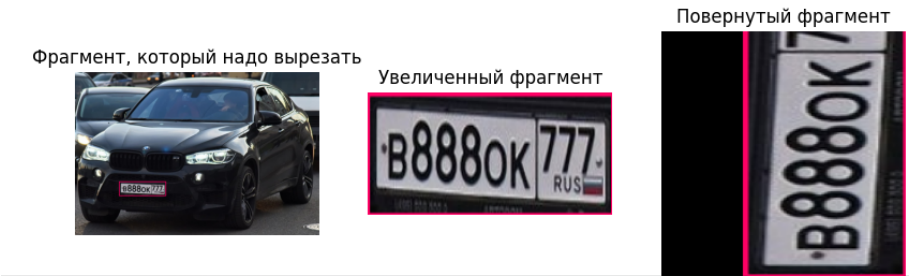
```
img = cv2.imread('mol1.png', 0)

ret, thresh = cv2.threshold(img, 0, 255, 0)
contours, hierarchy = cv2.findContours(thresh, 2, 3)

imag = cv2.imread('mol1.png')

f = plt.figure(figsize=(20, 20))

plt.subplot(1, 4, 1)
plt.title('Original')
plt.imshow(img, 'gray')

plt.subplot(1, 4, 2)
plt.title('Epsilon = 10%')
for cnt in contours:
    epsilon = 0.01 * cv2.arcLength(cnt, True)
    approx = cv2.approxPolyDP(cnt, epsilon, True)
    cv2.drawContours(imag, [approx], -1, (0, 255, 255), 2)
plt.imshow(imag)

plt.subplot(1, 4, 3)
plt.title('Epsilon = 1%')
imAg = cv2.imread('mol1.png')
for cnt in contours:
    epsilon = 0.1 * cv2.arcLength(cnt, True)
    approx = cv2.approxPolyDP(cnt, epsilon, True)
    cv2.drawContours(imAg, [approx], 0, (0, 255, 255), 2)
plt.imshow(imAg);

plt.show();
```



Пример 10

Задание 4.11. Нарисовать прямоугольник в месте, где нужно вырезать фрагмент (см. рис. 3), вывести на экран фрагмент, ограниченный прямоугольником, увеличив этот фрагмент. Определить размер изображения, его центр и повернуть его на 90 градусов.

```
img8 = cv2.imread('car.jpg', 1)
img8 = cv2.cvtColor(img8, cv2.COLOR_BGR2RGB)

plt.figure(figsize=(10, 10))
plt.subplot(1, 3, 1)
# Рисуем прямоугольник в месте,где собираемся вырезать фрагмент.
image = cv2.rectangle(img8, (190, 470), (390, 530), (255, 0, 100), 2)
plt.axis('off')
plt.imshow(image)
plt.title('Фрагмент, который надо вырезать')

plt.subplot(1, 3, 2)
# Выведем на экран участок изображения
crop = img8[470:530, 190:390]
# Увеличим этот фрагмент до 200x100
piece = cv2.resize(crop, (200,100), interpolation=cv2.INTER_LINEAR)
# Получим длину и ширину изображения как w и h
(h, w) = piece.shape[:2]
plt.axis('off')
plt.imshow(piece)
plt.title('Увеличенный фрагмент')

plt.subplot(1, 3, 3)
center = (w / 2, h / 2) # Получим центр изображения
# Повернем центр на 90 градусов с коэффициентом масштаба 1.0
M = cv2.getRotationMatrix2D(center, 90, 1.0)
rotated = cv2.warpAffine(piece, M, (150, 150))
plt.axis('off')
plt.imshow(rotated)
plt.title('Повернутый фрагмент')

plt.show();
```



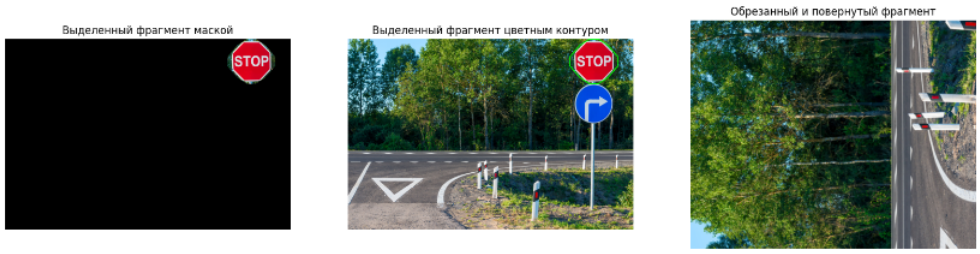
Пример 11

7. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.).

Условие.

Нарисовать многоугольник в месте, где нужно выделить фрагмент. Вывести изображение с выделенным фрагментом. И вывести изображение повернутое на 90 градусов и приближенное в центр.

```
: img = cv2.imread('image.jpg')

  # Определение вершин многоугольника
  pts = np.array([[90, 115], [120, 80], [150, 60], [180, 60], [210, 80], [240, 115], [240, 145],
                  [210, 180], [180, 200], [150, 200], [120, 180], [90, 145]])
  pts[:,0] += 610 # Смещение по горизонтали
  pts[:, 1] -= 60 # Смещение по вертикали

  # Применение маски к изображению
  mask = np.zeros(img.shape[:2], dtype=np.uint8)
  cv2.fillPoly(mask, [pts], 255)
  masked_img = cv2.bitwise_and(img, img, mask=mask)

  # Поворот и обрезка изображения
  rows,cols,colors = img.shape
  M = cv2.getRotationMatrix2D((cols/2,rows/2),90,1)
  rotated_img = cv2.warpAffine(img,M,(cols,rows))
  crop = rotated_img[150:550, 150:650]

  # Вывод изображений на экран
  plt.figure(figsize=(20, 20))
  plt.subplot(1, 3, 1)
  plt.imshow(cv2.cvtColor(masked_img, cv2.COLOR_BGR2RGB))
  plt.axis('off')
  plt.title('Выделенный фрагмент маской')

  plt.subplot(1, 3, 2)
  # Отрисовка многоугольника на изображении
  img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
  plt.imshow(cv2.polylines(img, [pts], True, (0, 255, 0), thickness=2))
  plt.axis('off')
  plt.title('Выделенный фрагмент цветным контуром')

  plt.subplot(1, 3, 3)
  plt.imshow(cv2.cvtColor(crop, cv2.COLOR_BGR2RGB))
  plt.axis('off')
  plt.title('Обрезанный и повернутый фрагмент')

  plt.show()
```



Результат работы индивидуального задания

8. Зафиксируйте сделанные изменения в репозитории.

9. Выполните слияние ветки для разработки с веткой main(master).

10. Отправьте сделанные изменения на сервер GitHub.

Контрольные вопросы

1. С помощью какой функции можно изменить размера изображения в OpenCV?

cv.resize (img, dim, interpolation=…). Первый аргумент – матрица изображения, второй dim либо width, height – размер изображения, третий – метод интерполяции.

2. Что такое аффинное преобразование?

Аффинное преобразование — это математическая операция, которая сопоставляет одно координатное пространство с другим. Другими словами, он сопоставляет один набор точек с другим набором точек. Аффинные преобразования имеют некоторые функции, которые делают их полезными в компьютерной графике.

3. Какие основные методы интерполяции?

cv.INTER_AREA – для сжатия,

cv.INTER_CUBIC и cv.INTER_LINEAR – для масштабирования.

По умолчанию используется метод интерполяции cv.INTER_LINEAR.

4. Сдвиг, смещение местоположения объекта.

Матрица смещения в плоскости (x, y) имеет вид: $$M = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{pmatrix},$$

где (tx, ty) – вектор смещения. В программе сначала определяем количество столбцов (rows) – это ширина, затем количество строк (cols) – это высота. С помощью функции cv.warpAffine() изображение сдвигается в направлении (tx, ty).

5. Какие аргументы у функции cv.getRotationMatrix2D(,)?

Первые два аргумента – координаты центра, третий аргумент – угол поворота.

6. Что такое функция cv2.drawContours()?

Она возвращает структуру box, которая содержит следующие аргументы: верхний левый угол (x, y), ширину, высоту, угол поворота.

7. Что делает функция cv2.minEnclosingCircle ()?

Окружность с минимальной площадью, охватывающей объект, рисуется с помощью функции cv2.minEnclosingCircle().

8. С помощью какой функции можно вписать изображение в эллипс с минимальной площадью?

cv2.ellipse()

9. Функция cv2.approxPolyDP(cnt,epsilon,True).

Она позволяет аппроксимировать контур. Первый аргумент cnt = contours [i] – массив с ко ординатами пикселей контура, аргумент epsilon задается в процентах, с уменьшением epsilon максимальное расстояние между ломаной прямой, аппроксимирующей контур, и самим контуром также уменьшается. Значение этого аргумента вычисляется функцией epsi lon = 0.1*cv2.arcLength(cnt,True).

10. Что делает функция cv2.rectangle?

Выделим на изображении интересующую нас область, заключив ее в прямоугольную рамку с помощью данной функции рисования.