МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

«Цифровая обработка бинарных изображений»

Отчет по лабораторной работе № 10 (4)

по дисциплине «Технологии распознавания образов»

Выполнил студент группы ПИЖ-б-о-21-1

Кучеренко С. Ю. « » 2023г.

Подпись студента_____

Работа защищена « »_____2023г.

Проверил 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.

Выполнение работы:

Примеры лабораторной работы

Задание 4.1.

Изменить размер изображения.

In [3]:
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

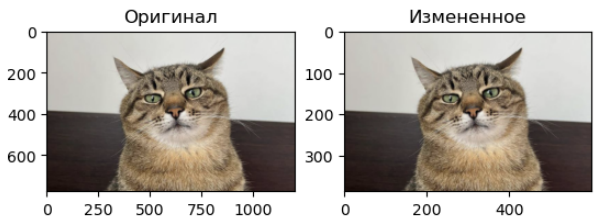
In [4]:
```
def img_print(orig, res):
    pose = [121, 122]
    signature = ["Оригинал", "Измененное"]
    img = [orig, res]
    i = 0
    while i < 2:
        plt.subplot(pose[i])
        plt.title(signature[i])
        plt.imshow(img[i])
        i += 1
    return 0
```

In [5]:
```
image = cv2.imread('image/cat.jpg',0)
img = cv2.imread('image/cat.jpg',1)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

Первый способ изменения размера задается в процентах

In [6]:
```
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);
```

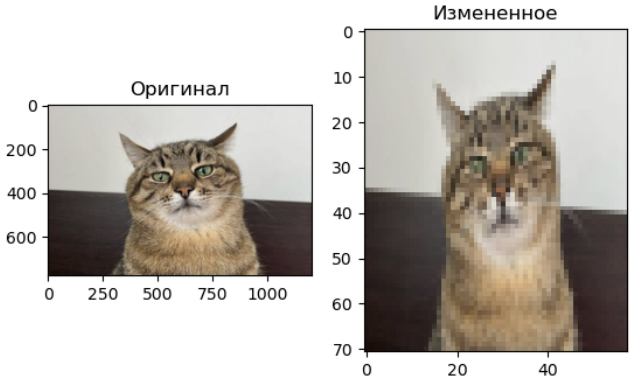
```
Resized Dimensions :  (387, 600, 3)
```



Второй способ изменения размера задается вручную

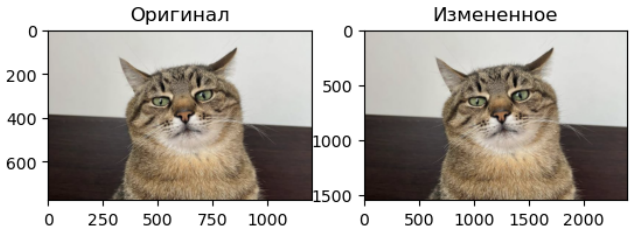
In [7]:
```
print('Original Dimensions : ',img.shape)
width = 58
height = 71
dim1 = (width, height)

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

```
Original Dimensions :  (775, 1200, 3)
Resized Dimensions :  (71, 58, 3)
```



Третий способ: задается коэффициентом масштабирования

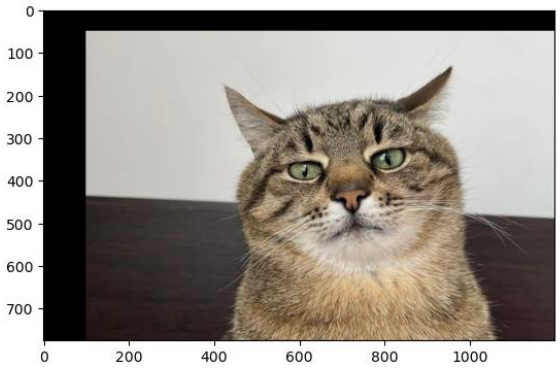
In [8]:
```
res = cv2.resize(img,None,fx=2, fy=2, interpolation = cv2.INTER_CUBIC)

#OR
#height, width = img.shape[:2]
#res = cv2.resize(img,(2*width, 2*height), interpolation = cv2.INTER_CUBIC)
img_print(img, res);
```



Задание 4.2.

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

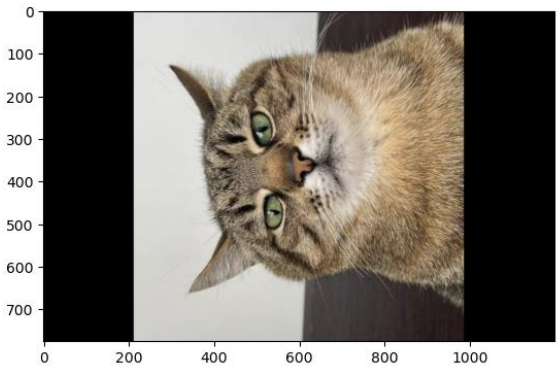
In [9]:
```
rows,cols,colors = img.shape
M = np.float32([[1,0,100],[0,1,50]])
dst = cv2.warpAffine(img,M,(cols,rows))
plt.imshow(dst);
```



Задание 4.3.

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

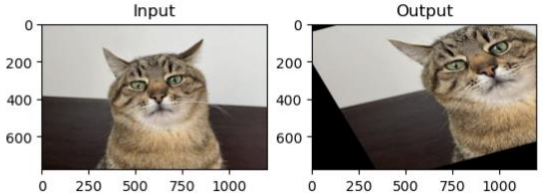
In [10]:
```
M = cv2.getRotationMatrix2D((cols/2,rows/2),90,1)
dst = cv2.warpAffine(img,M,(cols,rows))
plt.imshow(dst);
```



Задание 4.4.

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

In [11]:
```
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(img,M,(cols,rows))

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

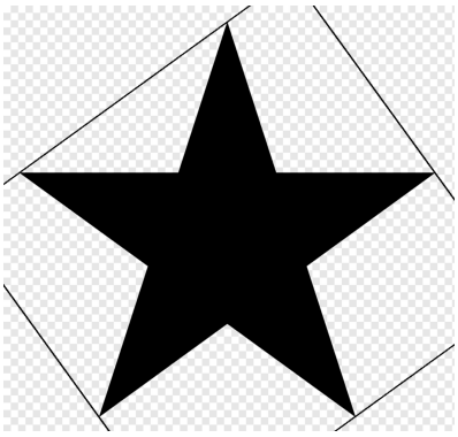


Задание 4.5.

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

In [14]:
```
image = cv2.imread('image/star.png',0)
plt.axis('off')
plt.imshow(image);
```

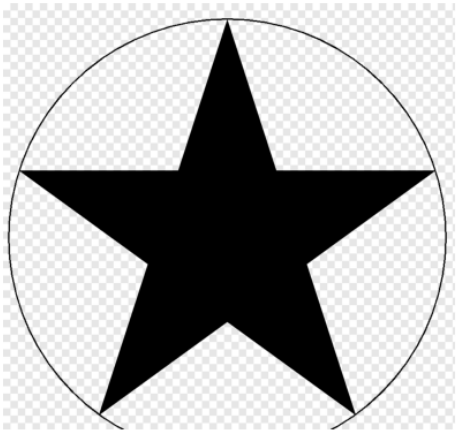
In [15]:
```
ret, thresh = cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)
contours, hierarchy = cv2.findContours(thresh, 1, 1)

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

box = cv2.boxPoints(rect)
box = np.int0(box)

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

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



Задание 4.6.

Провести охват изображения в круг.

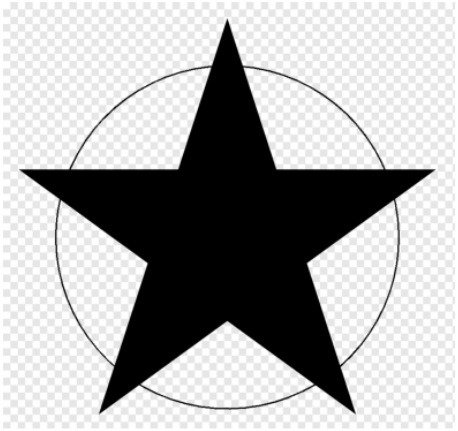
In [16]:
```
image = cv2.imread('image/star.png',0)

(x,y),radius = cv2.minEnclosingCircle(cnt)
center = (int(x),int(y))
radius = int(radius)

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

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



Задание 4.7.

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

In [18]:
```
image = cv2.imread('image/star.png',0)

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

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



Задание 4.8.

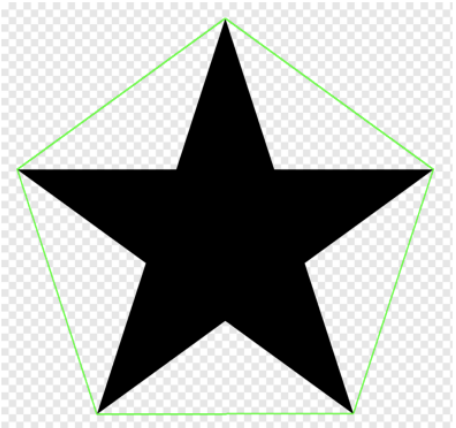
Провести прямую линию вдоль оси симметрии изображения.

In [19]:
```
image = cv2.imread('image/star.png',0)

ret, thresh = cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)
inv = cv2.bitwise_not(thresh)

contours, hierarchy = cv2.findContours(inv, 1, 1)
cnt = contours[0]
rows,cols = inv.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(image,(cols-1,righty),(0,lefty),(0,255,0),2)
img = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.axis('off')
plt.imshow(image);
```

Задание 4.9.

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

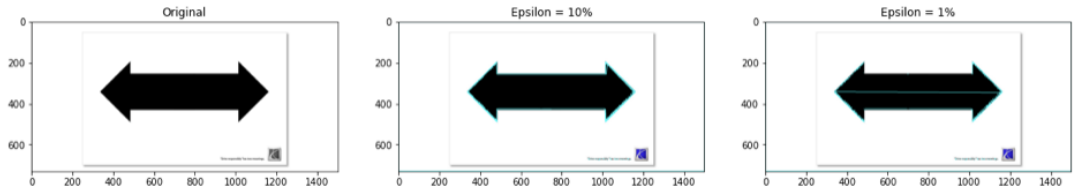
In [20]:
```
image = cv2.imread('image/star.png')

original_image = image
gray = cv2.cvtColor(image,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(image, [hull],0,(0,255,0),2)
plt.axis('off')
plt.imshow(image);
```



Задание 4.10.

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

In [60]:
```
img = cv2.imread('image/ch_b.jpg', 0)

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

imag = cv2.imread('image/ch_b.jpg')

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

plt.subplot(1,3,1)
plt.title('Original')
plt.imshow(img,'gray')

plt.subplot(1,3,2)
plt.title('Epsilon = 10%')
for i in range(np.size(contours)):
    cnt = contours [i]
    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(1,3,3)
plt.title('Epsilon = 1%')
imAg = cv2.imread('image/ch_b.jpg')
for i in range(np.size(contours)):
    cnt = contours [i]
    epsilon = 0.1 * cv2.arcLength(cnt, True)
    approx = cv2.approxPolyDP(cnt,epsilon,True)
    cv2.drawContours(imAg,[approx],-1,(0,255,255),2)
plt.imshow(imAg)
plt.show();
```



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.

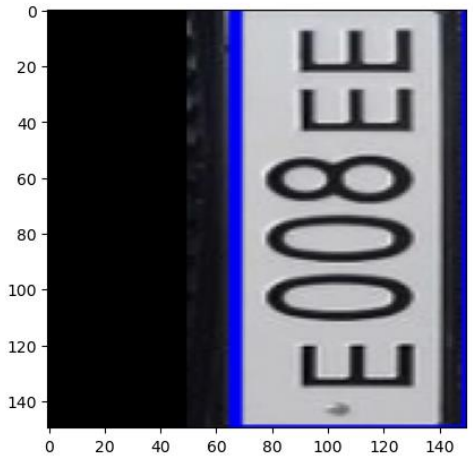
In [37]:
```
img = cv2.imread('image/avto.jpeg', 1)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

image = cv2.rectangle(img, (390, 400), (600, 450), (0, 0, 255), 2)

plt.imshow(image);
```

In [41]:
```
crop = img[390:450, 390:610]
piece = cv2.resize(crop, (200,100), interpolation=cv2.INTER_LINEAR)

(h, w) = piece.shape[:2]
print(w,h)

plt.imshow(piece);
```

```
200 100
```

In [42]:
```
center = (w / 2, h / 2)
M = cv2.getRotationMatrix2D(center, 90, 1)
rotated = cv2.warpAffine(piece, M, (150, 150))

plt.imshow(rotated)
```

Out[42]:
```
<matplotlib.image.AxesImage at 0x167f28640>
```



Самостоятельное задание

Считать изображение и выполнить:

1. Изменить размер изображения.
2. Определить размер изображения и сдвинуть изображение на 200 столбцов и 100 строк.
3. Определить размер изображения, его центр и повернуть его на 180 градусов.
4. Провести охват изображения в прямоугольник, повернутый так, чтобы площадь этого прямоугольника была минимальной
5. Нарисовать прямоугольник в месте, где нужно вырезать фрагмент, вывести на экран фрагмент, ограниченный прямоугольником, увеличив этот фрагмент.

In [1]:
```
import numpy as np
import cv2
import matplotlib.pyplot as plt
```

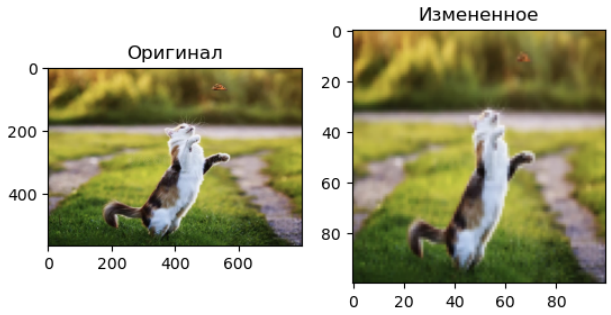
In [2]:
```
def img_print(orig, res):
    pose = [121, 122]
    signature = ["Оригинал", "Измененное"]
    img = [orig, res]
    i = 0
    while i < 2:
        plt.subplot(pose[i])
        plt.title(signature[i])
        plt.imshow(img[i])
        i += 1
    return 0
```

Изменение размера изображения.

In [5]:
```
img = cv2.imread('image/ind.jpeg', 1)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

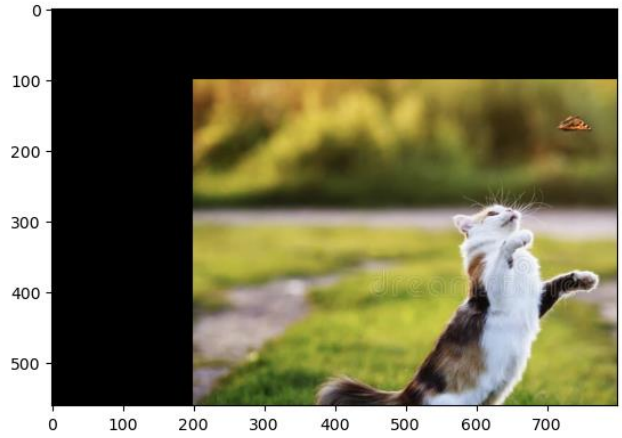
In [9]:
```
print('Original Dimensions : ',img.shape)
width = 100
height = 100
dim1 = (width, height)

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

```
Original Dimensions :  (561, 800, 3)
Resized Dimensions :  (100, 100, 3)
```



Определение размера изображения и сдвиг изображения на 200 столбцов и 100 строк.

In [10]:
```
rows,cols,colors = img.shape
M = np.float32([[1,0,200],[0,1,100]])
dst = cv2.warpAffine(img,M,(cols,rows))
plt.imshow(dst);
```



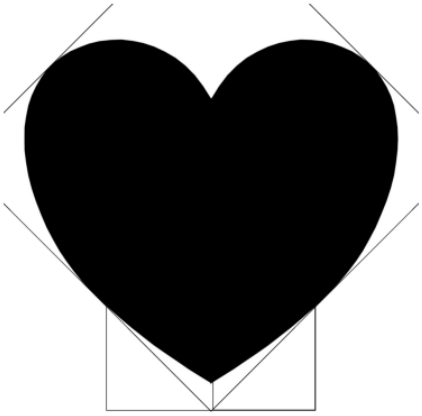
Определение размера изображения, его центра и поверот его на 180 градусов.

In [13]:
```
M = cv2.getRotationMatrix2D((cols/2,rows/2),180,1)
dst = cv2.warpAffine(img,M,(cols,rows))
plt.imshow(dst);
```

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

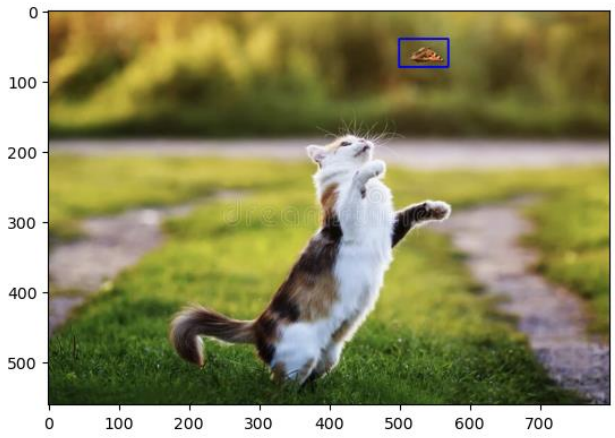
In [18]:
```
image = cv2.imread('image/ind2.jpeg',0)
plt.axis('off')
plt.imshow(image);
```

In [21]:
```
ret, thresh = cv2.threshold(image, 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.int0(box)

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

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



Нарисовать прямоугольник в месте, где нужно вырезать фрагмент, вывести на экран фрагмент, ограниченный прямоугольником, увеличив этот фрагмент.

In [42]:
```
img = cv2.imread('image/ind.jpeg', 1)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

image = cv2.rectangle(img, (500, 40), (570, 80), (0, 0, 255), 2)

plt.imshow(image);
```



In [45]:
```
crop = img[40:80, 500:570]
piece = cv2.resize(crop, (200,100), interpolation=cv2.INTER_LINEAR)

(h, w) = piece.shape[:2]
print(w,h)

plt.imshow(piece);
```

```
200 100
```