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

«Цифровая обработка бинарных изображений»

по дисциплине «Технологии распознавания образов»

Выполнил студент группы
ПИЖ-б-о-21-1
Зиберов Александр
« » мая 2023 г.
Подпись студента_____
Работа защищена
« »_____20__г.
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:

Проработать примеры лабораторной работы в отдельном ноутбуке.

Задание 4.1. Изменить размер изображения.

In [19]:
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

In [20]:
```
def imgshow(inp, out):
    position = [121, 122]
    headers = ["Input", "Output"]
    img = [inp, out]
    for i in range(2):
        plt.subplot(position[i])
        plt.title(headers[i])
        plt.imshow(img[i])
```

In [21]:
```
image = cv2.imread('pictures/forest.jpg',0)
img = cv2.imread('pictures/forest.jpg',1)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

Первый способ изменения размера задается в процентах

In [22]:
```
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)
imgshow(img, resized);
```

```
Resized Dimensions :  (360, 576, 3)
```

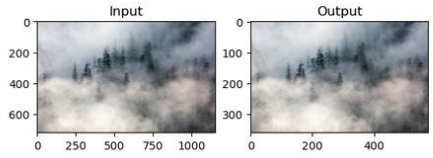
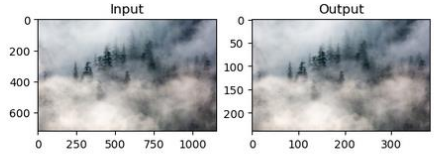


Рисунок 1 – Пример 1

Второй способ изменения размера задается вручную

In [23]:
```
print('Original Dimensions : ',img.shape)
width = 384
height = 240
dim1 = (width, height)

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

```
Original Dimensions :  (720, 1152, 3)
Resized Dimensions :  (240, 384, 3)
```



Третий способ: задается коэффициентом масштабирования

In [24]:
```
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)
imgshow(img, res);
```

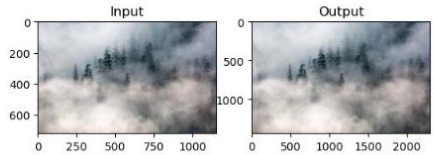
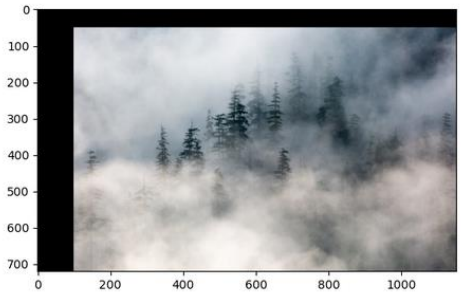


Рисунок 2 – Пример 2

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

In [25]:
```
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 [26]:
```
M = cv2.getRotationMatrix2D((cols/2,rows/2),90,1)
dst = cv2.warpAffine(img,M,(cols,rows))
plt.imshow(dst);
```

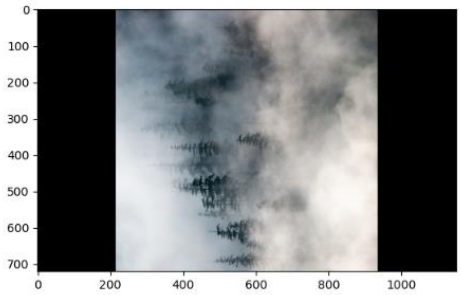
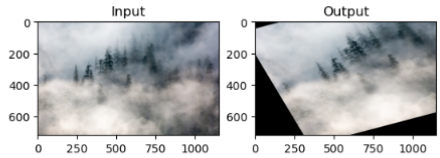


Рисунок 3 – Пример 3

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

In [27]:
```
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 [28]:
```
img = cv2.imread('pictures/bin.jpg', 0)

plt.subplot(121),plt.imshow(img), plt.title('Input')
plt.axis("off")

ret, thresh = cv2.threshold(img, 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.intp(box)
imp = cv2.drawContours(img, [box], 0, (0, 0, 255), 2)
imp = cv2.cvtColor(imp, cv2.COLOR_BGR2RGB)

plt.subplot(122),plt.imshow(imp), plt.title('Output')
plt.axis("off")
plt.show()
```

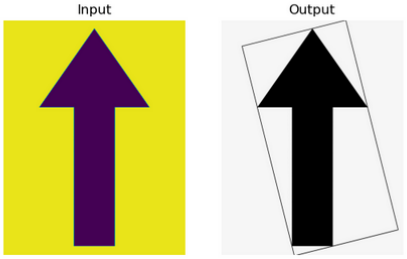
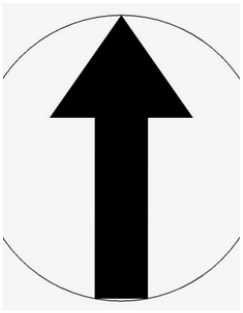


Рисунок 4 – Пример 4

Задание 4.6. Провести охват изображения в круг.

In [29]:
```
image = cv2.imread('pictures/bin.jpg',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 [30]:
```
image = cv2.imread('pictures/bin.jpg',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);
```

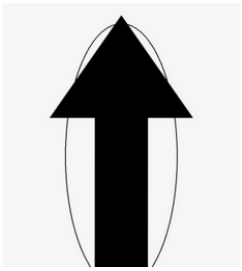


Рисунок 5 – Пример 5

Задание 4.8. Провести прямую линию вдоль оси симметрии изображения.

In [31]:
```
img = cv2.imread('pictures/bin.jpg', 0)
img_orig = img.copy()
ret, thresh = cv2.threshold(img, 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(
    img,
    (cols-1,righty),
    (0,lefty),
    (0,255,0)
    ,2
)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

imgshow(img_orig, img)
```

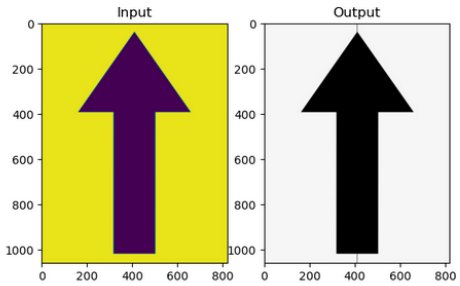


Рисунок 6 – Пример 6

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

In [32]:
```
image = cv2.imread('pictures/bin5.jpg')

original_image = image
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
edges = cv2.Canny(gray, 50,200)
contours, hierarbiny= 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);
```

Рисунок 7 – Пример 7

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

In [33]:
```
img = cv2.imread('pictures/lightning.jpg', 0)

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

imag = cv2.imread('pictures/lightning.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('pictures/lightning.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();
```

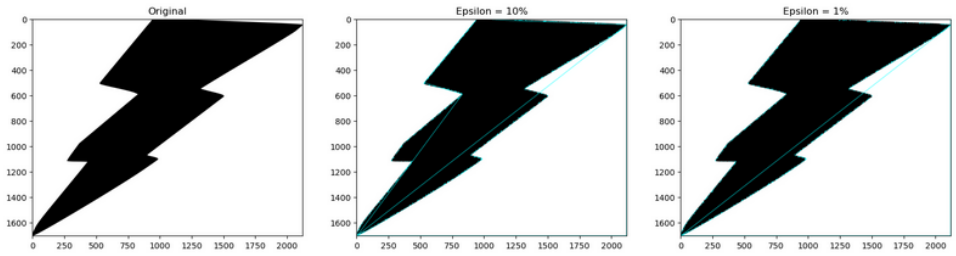
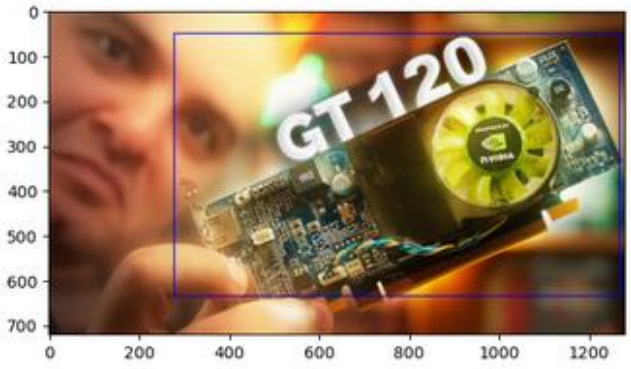


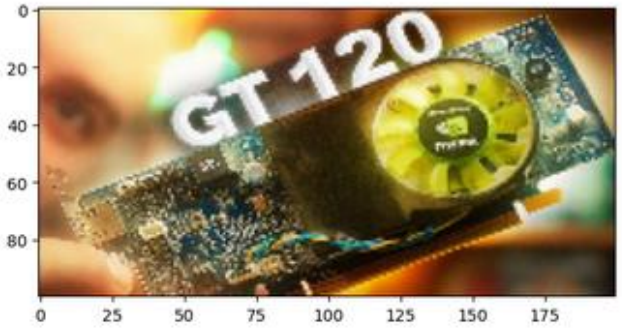
Рисунок 8 – Пример 8

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 [34]:
```
img = cv2.imread('pictures/maxresdefault.jpg', 1)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

image = cv2.rectangle(img, (279, 51), (1269, 635), (0, 0, 255), 2)

plt.imshow(image);
```



In [35]:
```
crop = img[51:635, 279:1269]
piece = cv2.resize(crop, (200,100), interpolation=cv2.INTER_LINEAR)

(h, w) = piece.shape[:2]
plt.imshow(piece);
```



In [36]:
```
center = (w / 2, h / 2)
M = cv2.getRotationMatrix2D(center, 90, 1)
rotated = cv2.warpAffine(piece, M, (150, 150))
plt.imshow(rotated);
```

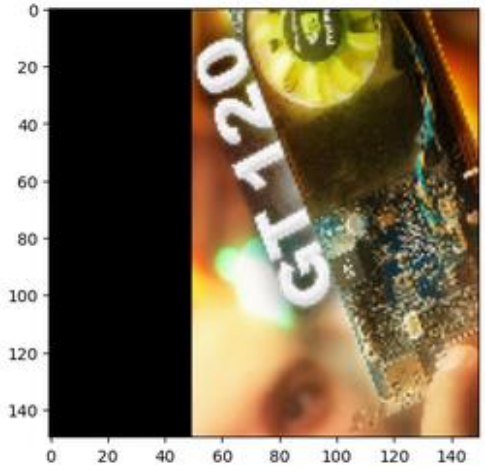


Рисунок 9 – Пример 9

Индивидуальное задание

Задание.

1. Повернуть изображение на 90 градусов при помощи аффинных преобразований
2. Вырезать область изображения (биллборд)
3. Произвести upscale изображения
4. Найти и отрисовать контуры изображения

In [15]:
```
import cv2
from matplotlib import pyplot as plt
%matplotlib inline

def imgshow(image):
    plt.imshow(image)
    plt.xticks([])
    plt.yticks([])
    plt.axis("off")
    plt.show();
```

Импортируем изображение:

In [16]:
```
# Импортируем исходную картинку
img = cv2.cvtColor(cv2.imread('pictures/billboard.jpg'), cv2.COLOR_BGR2RGB)
imgshow(img)
```

Рисунок 10 – Индивидуальное задание (1)

Перевернем картинку на 90 градусов при помощи функции cv.warpAffine():

In [17]:
```
# Находим разрешение изображения, его центр
(h, w) = img.shape[:2]
(cX, cY) = (w // 2, h // 2)

# Поворачиваем изображение на 90 градусов
M = cv2.getRotationMatrix2D((cX, cY), -90, 1.0)
rotated = cv2.warpAffine(img, M, (w, h))
imgshow(rotated)
```

Рисунок 11 – Индивидуальное задание (2)

Выделим область биллборда в прямоугольник:

In [18]:
```
img = rotated
selected = cv2.rectangle(img, (208, 288), (343, 531), (255, 0, 0), 2)
plt.axis('off');
imgshow(selected)
```

Вырежем область изображения:

In [19]:
```
img = rotated
# Задаем параметры
width, height = 130, 230
x, y = 210, 290
# Обрезаем картинку
crop_img = img[y:y+height, x:x+width]

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

Рисунок 12 – Индивидуальное задание (3)

Проведем апскейл изображения, для этого воспользуемся моделью FSRCNN_x4, считав ее при помощи функции cv2.dnn_superres.DnnSuperResImpl_create():

In [20]:
```
img = crop_img
# Создаем объект superresolution
sr = cv2.dnn_superres.DnnSuperResImpl_create()
path = "pictures/FSRCNN_x4.pb"

# Считываем модель
sr.readModel(path)
sr.setModel("fsrcnn", 4)
# Апскейлим изображение
upscaled = sr.upsample(img)

plt.figure(figsize=(12,8))
plt.subplot(1,3,1)
plt.imshow(img)
plt.subplot(1,3,2)
plt.imshow(upscaled)
plt.show()
```

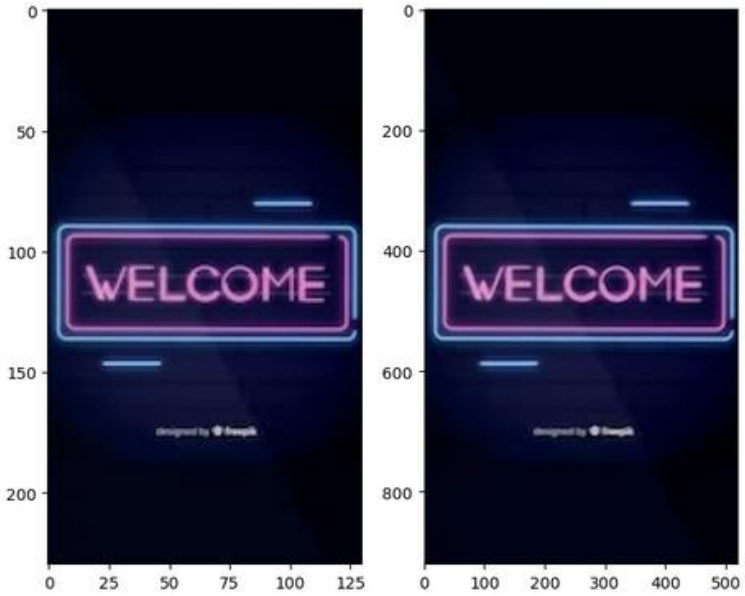


Рисунок 13 – Индивидуальное задание (4)

Отрисуем контуры изображения. Воспользуемся преобразованием изображения в градации серого, функцией cv2.Canny() находим края изображения, затем при помощи cv2.findContours() и drawContours() отрисуем:

In [21]:
```
img_copy = upscaled.copy()
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
edges = cv2.Canny(image=img_gray,threshold1=100, threshold2=200)
contours, hierarchy = cv2.findContours(image=edges, mode=cv2.RETR_TREE, method=cv2.CHAIN_APPROX_NONE)
cv2.drawContours(image=img, contours=contours, contourIdx=-1, color=(0, 255, 0), thickness=2, lineType=cv2.LINE_AA)
imgshow(img)
```

Рисунок 14 – Индивидуальное задание (5)

Вывод: В результате выполнения работы были изучены основные операции геометрических преобразований изображений, такие как изменение размера, сдвиг, вращение, аффинное преобразование и т. д.

1. Что такое аффинное преобразование?

Аффинное преобразование — это преобразование плоскости, которое сохраняет прямые и параллельность. Аффинное преобразование может выполнять повороты, масштабирование, сдвиги и отражения относительно прямых.

2. Что происходит при аффинной трансформации изображения?

При аффинном преобразовании все параллельные линии исходного изображения остаются параллельными и в выходном изображении.

3. С помощью какой функции можно совершить изменение размера изображения?

cv.resize (img, dim, interpolation=…)

Первый аргумент – матрица изображения, второй dim либо width, height – размер изображения, третий – метод интерполяции

4. Перечислите 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.

5. Как реализуется вращение изображения?

Поворот изображения на некоторый угол достигается с помощью матрицы:

$$M = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

У функции вращения cv.getRotationMatrix2D(,) первые два аргумента – координаты центра, третий аргумент – угол поворота.5. Чему равна площадь бинарного изображения?

6. С помощью какой функции можно осуществить вращение изображения?

cv2.getRotationMatrix2D(center, angle, scale)

center: Центр вращения

angle(θ): угол поворота.

scale: коэффициент масштабирования

7. Какие функции позволяют выполнить охват объекта?

Функция cv2.drawContours() возвращает структуру box, которая содержит 37 следующие аргументы: верхний левый угол (x, y), ширину, высоту, угол поворота. Чтобы нарисовать прямоугольник, нужны 4 угла прямоугольника, которые задаются функцией cv2.boxPoints ().

Окружность с минимальной площадью, охватывающей объект, можно нарисовать м с помощью функции cv2.minEnclosingCircle ().

Используя функцию cv2.ellipse(), можно вписать изображение в эллипс с минимальной площадью.

8. Какая функция позволяет аппроксимировать контур?

Функция cv2.approxPolyDP(cnt,epsilon,True). Первый аргумент cnt = contours [i] – массив с координатами 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 epsilon = 0.1*cv2.arcLength(cnt,True).

9. Как осуществить выделение интересующей области, создание для нее отдельного изображения?

Выделим на изображении интересующую нас область, заключив ее в прямоугольную рамку с помощью функции рисования cv2.rectangle. Фрагмент изображения, заключенный в рамке, выведем на экран. Используя функцию .shape, получим размер изображения и изменим его с помощью функции cv2.resize. Функция 42 cv2.getRotationMatrix2D предназначена для поворота изображения, а функция cv2.warpAffine – для аффинного преобразования.