МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Нижегородский государственный университет им. Н.И. Лобачевского»**

**Национальный исследовательский университет**

**Институт информационных технологий, математики и механики**

**Кафедра: Программной инженерии**

Направление подготовки: «Программная инженерия»

«Обработка изображений»

**Отчёт по лабораторной работе**

|  |  |
| --- | --- |
|  | Выполнил:  студент группы 381908-3  Имя  Тревогин Кирилл Александрович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  Проверил:  Гетманская А.А  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Нижний Новгород  
2021 г.

Содержание

[Задача 3](#_Toc86656071)

[Работа 3](#_Toc86656072)

[Результаты и сравнения 4](#_Toc86656073)

[Код 5](#_Toc86656074)

[Вывод 7](#_Toc86656075)

# Задача

Выделить на изображении здоровую часть листа и повреждения листа. Использовать watershed и фильтры, уменьшающие шум. Написать методы, с какими параметрами использовали, какие оказались наилучшими.

# Работа

В данной работе я смог выделить на изображении поврежденные части листа при помощи watershed и фильтра Гаусса с размером матрицы 7 на 7. В ходе выполнения работы были испробованы варианты с применением cv.blur с размером матрицы 7\*7, фильтра Гаусса 7\*7, и Erosion с размером ядра 7\*7.

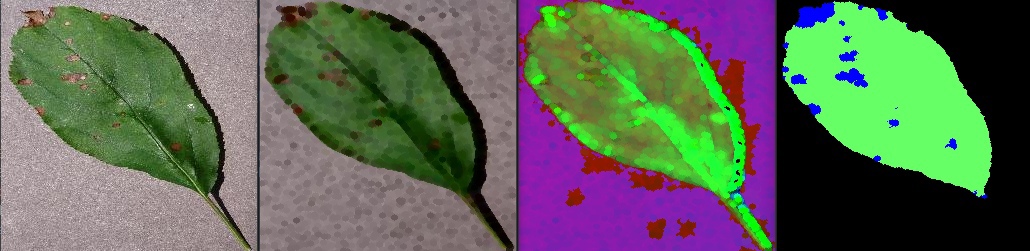
При использовании cv.blur результат был хорошим. Но на некоторых изображениях частично не были выявлены повреждения листа и иногда захватывала здоровую часть листа.

При использовании Erosion результат был удовлетворительным. Множество изображений на которых целая часть выделялась как поврежденная, также присутствовала потеря здорового листа.

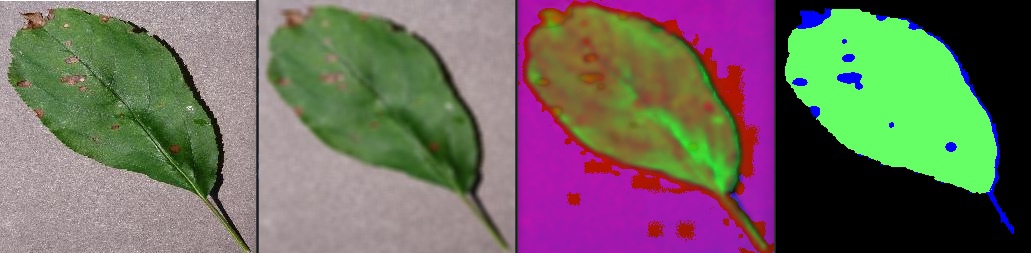
При использовании фильтра Гаусса был получен наиболее благоприятный результат.

# Результаты и сравнения

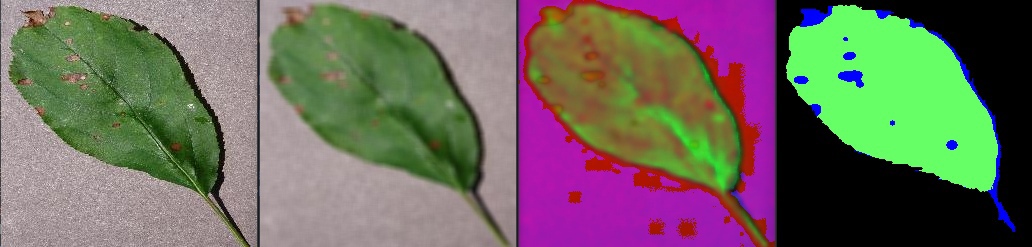
На данном изображении был применен Erosion Filter с размером ядра 7\*7. Мы видеть, что часть лита потерялось, в частности стебель, и часть точек ложные



Пример работы фильтра Гаусса с матрицей 7\*7 с этим же изображением. Все поврежденные части были отмечены и не затронуты целые.



На данном изображении представлен пример работы cv.blur 7\*7. Данное изображение похоже на гаусовское но при этом границы листа немного не.



# Код

import numpy as np

import cv2 as cv

import sys

def gnil (image) :

    image\_blur = cv.GaussianBlur(image, (7,7), cv.BORDER\_DEFAULT)

    # kernel = cv.getStructuringElement(cv.MORPH\_ELLIPSE, (7,7))

    # image\_blur = cv.erode(image,kernel)

    # image\_blur = cv.blur(image, (7,7))

    norm\_img1 = cv.normalize(image\_blur, None, alpha=0, beta=1, norm\_type=cv.NORM\_MINMAX, dtype=cv.CV\_32F)

    norm\_img1 = (255\*norm\_img1).astype(np.uint8)

    rgb\_planes = cv.split(image\_blur)

    result\_planes = []

    result\_norm\_planes = []

    for plane in rgb\_planes:

        dilated\_img = cv.dilate(plane, np.ones((7,7), np.uint8))

        bg\_img = cv.medianBlur(dilated\_img, 21)

        diff\_img = 255 - cv.absdiff(plane, bg\_img)

        norm\_img = cv.normalize(diff\_img,None, alpha=0, beta=255, norm\_type=cv.NORM\_MINMAX, dtype=cv.CV\_8UC1)

        result\_planes.append(diff\_img)

        result\_norm\_planes.append(norm\_img)

    result = cv.merge(result\_planes)

    result\_norm = cv.merge(result\_norm\_planes)

    hsv\_img = cv.cvtColor( image\_blur , cv.COLOR\_BGR2HSV)

    markers = np.zeros((image.shape[0], image.shape[1]) , dtype = "int32" )

    markers [ 100 : 140 , 100 : 140 ] = 255

    markers [ 236 : 255 , 0 : 20 ] = 1

    markers [ 0 : 20 , 0 : 20 ] = 1

    markers [ 0 : 20 , 236 : 255 ] = 1

    markers [ 236 : 255 , 236 : 255 ] = 1

    leafs\_area\_BGR = cv.watershed(image\_blur, markers)

    healthy\_part = cv.inRange(hsv\_img, (36,25,25), (86,255,255))

    ill\_part = leafs\_area\_BGR - healthy\_part

    mask = np.zeros\_like(image, np.uint8)

    mask [leafs\_area\_BGR > 1] = ( 102, 255, 102)

    mask [ill\_part > 1 ] = (250, 0, 0)

    cv.imshow("hsv\_img", hsv\_img)

    cv.imshow("Blured", image\_blur)

    return mask

def main():

    img = cv.imread('12.jpg')

    if img is None:

     sys.exit("Could not read the image.")

    cv.imshow("Origin", img)

    resimg = gnil(img)

    cv.imshow("end", resimg)

    k = cv.waitKey(0)

if \_\_name\_\_ == '\_\_main\_\_':

    main()

# Вывод

По результатам проведенной работы можно сделать вывод, что фильтр Гаусса 7\*7 является лучшим вариантом для определения повреждённой части листа.