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  Ермаков Павел Андреевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  Проверил:  Гетманская А.А  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Нижний Новгород  
2021 г.

Содержание

[Задача 2](#_Toc85474545)

[Работа 3](#_Toc85474546)

[Результаты и сравнения 3](#_Toc85474547)

[Код main.py 5](#_Toc85474548)

[Вывод 6](#_Toc85474549)

# Задача

Выделить на изображении здоровую часть листа и повреждения листа. Использовать watershed и фильтры, уменьшающие шум. Написать какие 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 медианного фильтра с размером матрицы 3\*3, 5\*5, 7\*7, фильтра Гаусса 3\*3, 5\*5, 7\*7, и Erosion Filter (Фильтр сужения) с размером ядра 3\*3, 5\*5, 7\*7.

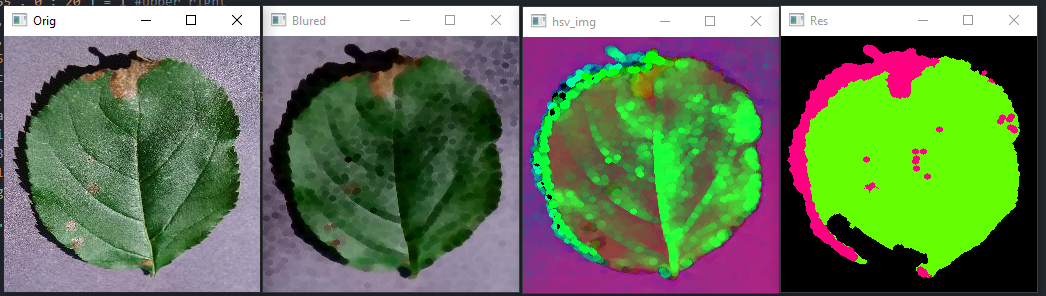
При использовании медианного фильтра со всеми вышеприведенными размерами матрицы результат был нормальным. Но на некоторых изображениях не были подсвечены некоторые повреждений листа и иногда съедалась здоровая часть листа.

При использовании Erosion Filterсо всеми вышеприведенными размерами ядра результат был более благоприятен. Но было довольно много изображений на которых целые части листа были отмечены, как поврежденные.

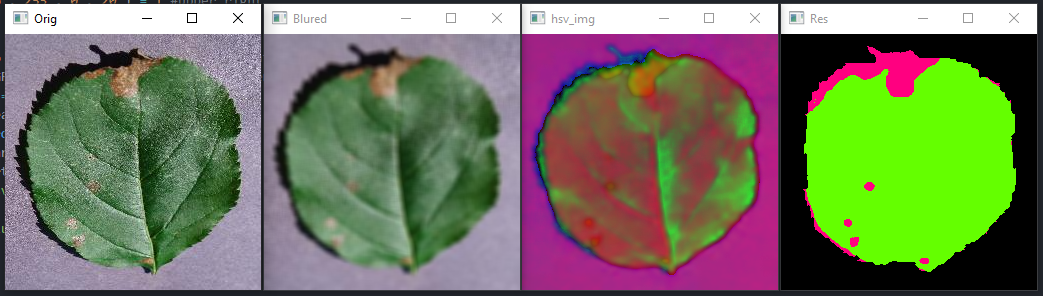
При использовании фильтра Гаусса с размером матрицы 7\*7 был получен наиболее благоприятный результат. Целые части листа не отмечались как поврежденные в сравнении Erosion Filter и практически все поврежденные части листа были отмечены верно.

# Результаты и сравнения

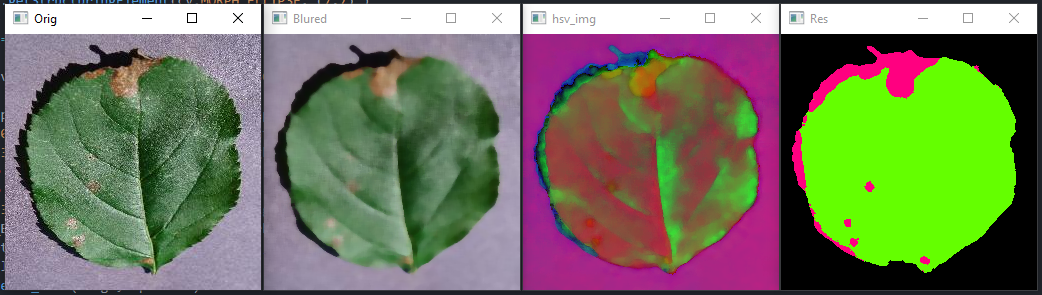
На данном изображении был применен Erosion Filter с размером ядра 7\*7. Мы можем видеть, что очень много точек отмечены как поврежденные. Но если мы посмотрим на оригинал, то увидим, что центр листа цел



Вот пример работы фильтра Гаусса с матрицей 7\*7 с этим же изображением. Отмечины все поврежденные части листа и не затронуты целые.



На данном изображении представлен пример работы медианного фильтра 7\*7. С данным изображением листа все в порядке, но можем видеть отличия от фильтра Гаусса. Менее точное определение поврежденной части и небольшой съезд правого края листа.



# Код main.py

import numpy as np

import cv2 as cv

import sys

def CalcOfDamageAndNonDamage (image) :

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

    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 # center

    markers [ 236 : 255 , 0 : 20 ] = 1 #upper right

    markers [ 0 : 20 , 0 : 20 ] = 1 #upper left

    markers [ 0 : 20 , 236 : 255 ] = 1 #low left

    markers [ 236 : 255 , 236 : 255 ] = 1 #low right

    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] = ( 0, 255, 100)

    mask [ill\_part > 1 ] = (128, 0, 255)

    cv.imshow("hsv\_img", hsv\_img)

    cv.imshow("Blured", image\_blur)

    return mask

def main():

    img = cv.imread('9.jpg')

    if img is None:

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

    cv.imshow("Orig", img)

    resimg = CalcOfDamageAndNonDamage(img)

    cv.imshow("Res", resimg)

    k = cv.waitKey(0)

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

    main()

# Вывод

По результатам проведенной работы я с делал вывод, что фильтр Гаусса 7\*7 работает лучше