

Министерство науки и высшего образования Российской Федерации  
Муромский институт (филиал)  
Федерального государственного бюджетного образовательного учреждения высшего  
образования  
«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет \_\_\_\_\_ ИТР \_\_\_\_\_

Кафедра \_\_\_\_\_ ПИН \_\_\_\_\_

# ЛАБОРАТОРНАЯ РАБОТА №1

По \_\_\_\_\_ Цифровая обработка информации \_\_\_\_\_

Тема \_\_\_\_\_ Бинаризация изображений. Работа с OpenCV \_\_\_\_\_

Руководитель

Белякова А.С.

\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Студент \_\_\_\_\_ ПИН - 121 \_\_\_\_\_  
(группа)

Ермилов М.В.

\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Муром 2024

## Лабораторная работа №1

Тема: бинаризация изображений. Работа с OpenCV.

Цели и задачи: получение навыков обработки изображений с помощью библиотеки OpenCV.

Ход работы: задание: напишите программу на Python, реализующую бинаризацию изображений из тестовых наборов данных.



Рисунок 1 – Пример изображения из тестового набора

Листинг кода 1 – программа для бинаризации изображений:

```
import cv2 as cv
from pathlib import Path

def create_dir(path: Path):
    if (path.exists()):
        print(f'dir \"{path}\" already exists')
        return
    print(f'creating dir \"{path}\"')
    path.mkdir()

def handler_image(border,src_img: Path, dst_img: Path):
    if dst_img.exists():
        print(f"img already exists {dst_img}")
        return
    img = cv.imread(src_img)
    img = cv.cvtColor(img,cv.COLOR_BGR2GRAY)
    ret, img = cv.threshold(img, border, 255, 0)
```

					МИВлГУ 09.03.04 - 0.007				
Изм.	Лист	№ докум.	Подпись	Дата					
Разраб.		Ермилов М.В.			Бинаризация изображений. Работа с OpenCV	Лит.	Лист	Листов	
Провер.		Белякова А.С.					2	4	
Реценз.						МИ ВлГУ ПИН-121			
Н. Контр.									
Утверд.									

```

print(f"creating {dst_img}")
cv.imwrite(dst_img, img)

def binarization(path: Path,min_border,max_border):
    for border in range(min_border,max_border+1):
        border_dir = Path.cwd()/f'{border}'
        create_dir(Path(border_dir))

        dest_dir = Path(border_dir/path.name)
        create_dir(Path(dest_dir))

        for img in path.iterdir():
            handler_image(border,img,Path(dest_dir/img.name))

training_set_A = Path('D:/ucheba/coi/A. Segmentation/1. Original Images/a. Training Set')
testing_set_B = Path('D:/ucheba/coi/A. Segmentation/1. Original Images/b. Testing Set')

def main():
    binarization(training_set_A,130,140)
    binarization(testing_set_B,130,140)

if __name__ == "__main__":
    main()

```

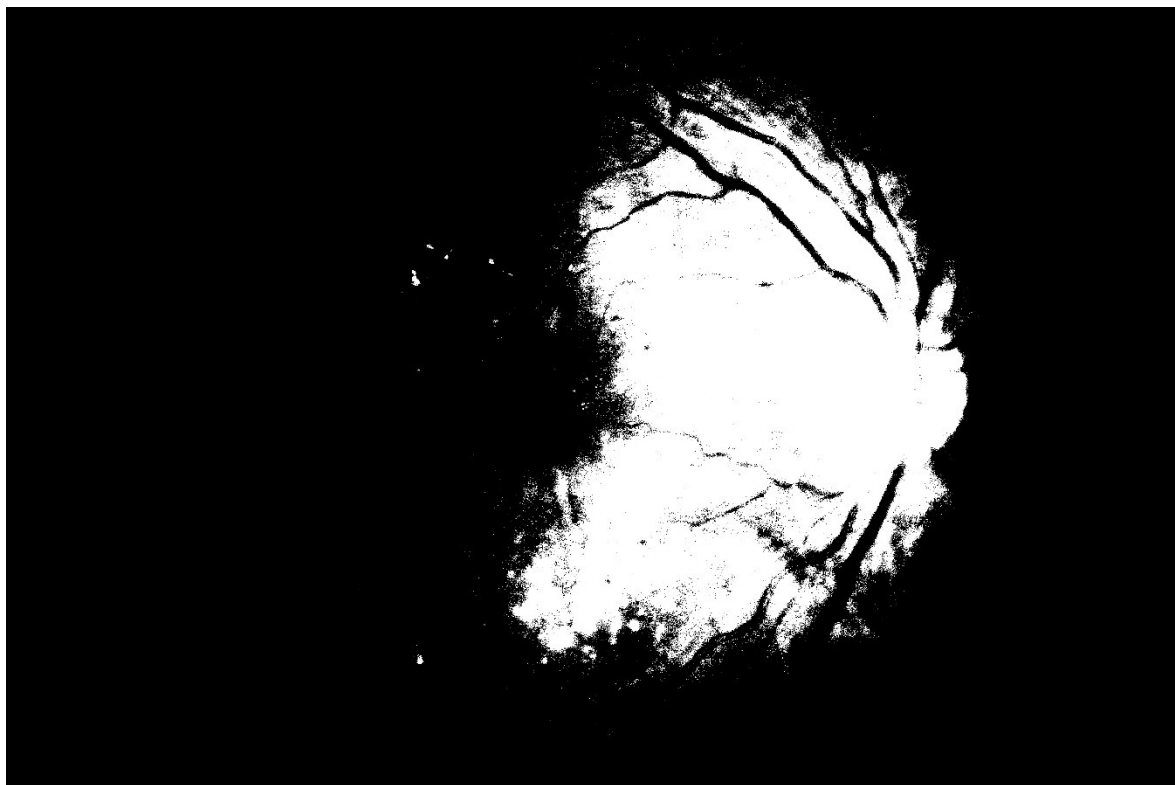


Рисунок 2 – Результат при бинаризации изображения с порогом 130

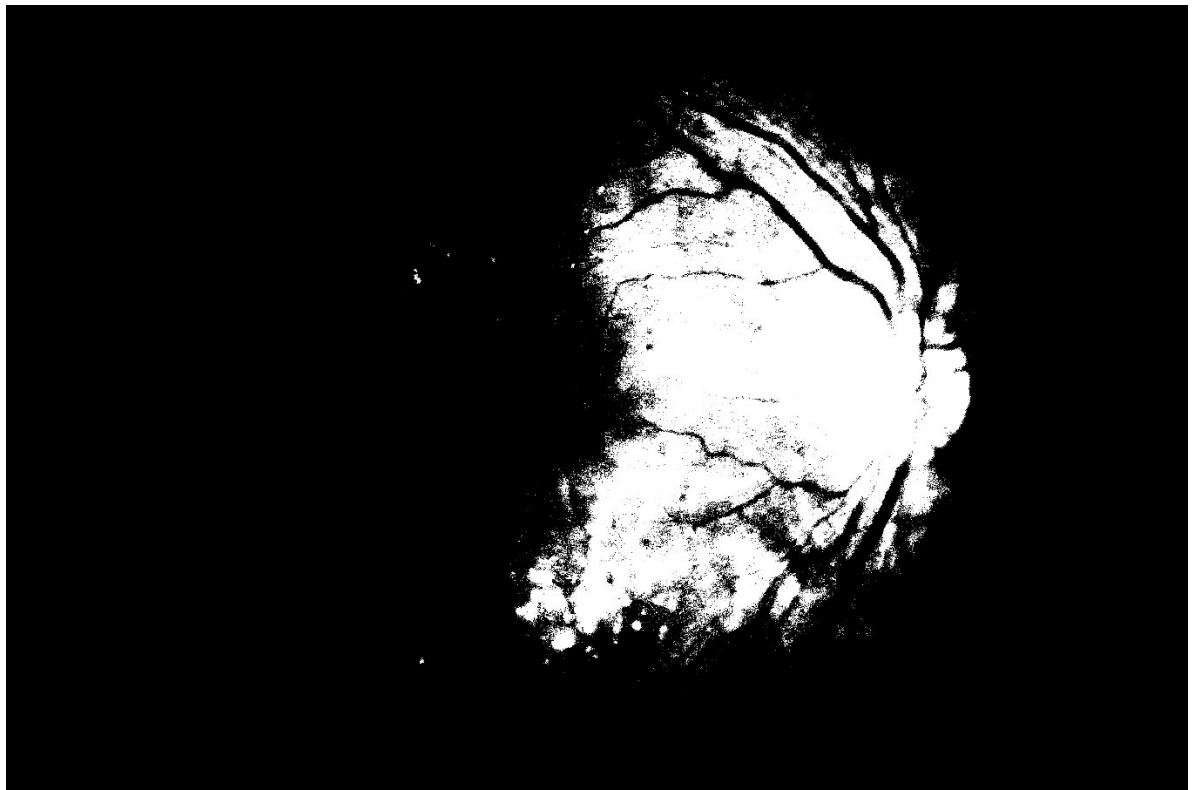


Рисунок 3 – Результат при бинаризации изображения с порогом 135

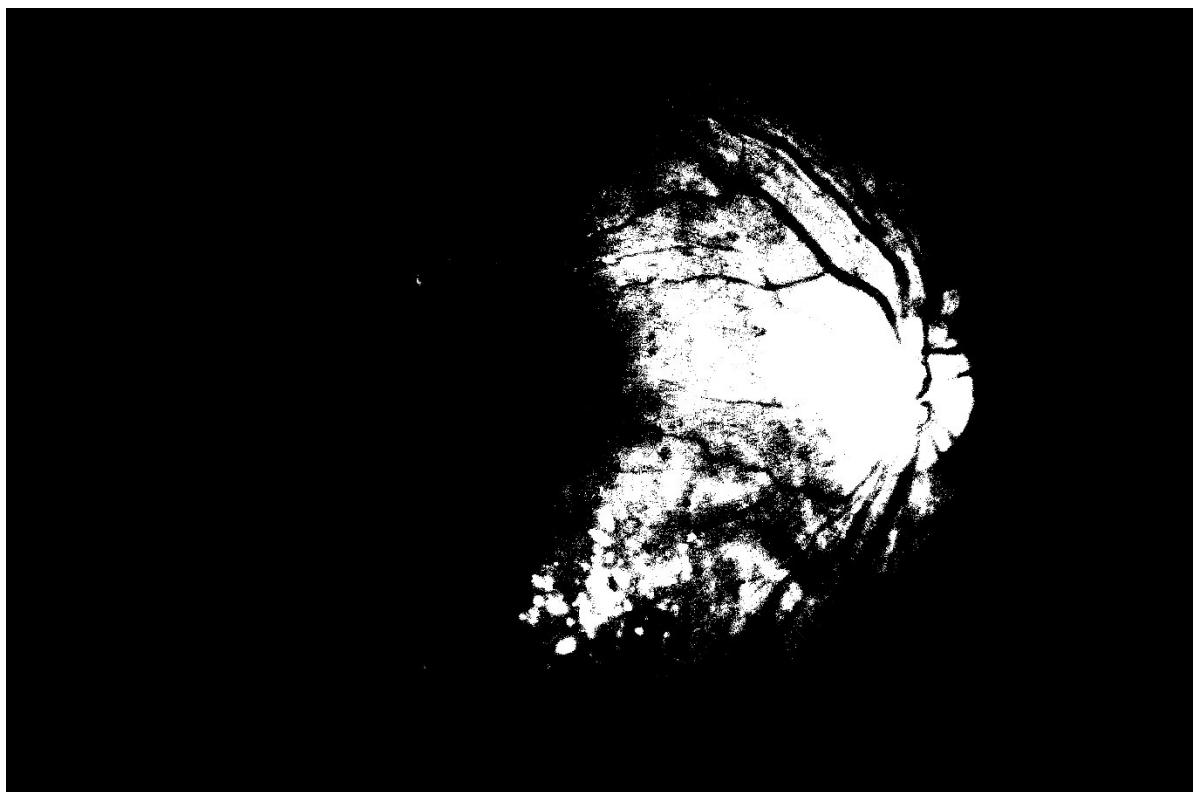


Рисунок 4 – Результат при бинаризации изображения с порогом 140

Вывод: в ходе работы получили навыки обработки изображений с помощью библиотеки OpenCV.

					МИВлГУ 09.03.04 – 0.007	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		