

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Систем автоматического управления

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Техническое зрение»
Тема: «Пороговое преобразование»

Студент гр. 1492

Ляховский А.С.

Преподаватель

Федоркова А.О.

Санкт-Петербург

2024

Цель работы.

Научиться применять пороговые преобразования для разделения пикселей на группы по признаку яркости

Ход работы.

Задание 1.

Используя пороговые преобразования, добиться того, чтобы птицы попали в одну группу, а фон – в другую. Какая функция для этого подошла лучше всего? Ручной поиск, адаптивный или автоматический?

Код программы:

```
import cv2
import numpy as np

path = 'Laba_2\\2-0.jpg'
window_name = 'figure_1'
window_name_2 = 'figure_2'
img=cv2.imread(path,flags=cv2.IMREAD_REDUCED_GRAYSCALE_4) # Прочитать в ЧБ
1:4
# Адаптивное пороговое преобразование
img_n = cv2.adaptiveThreshold(img,255,
cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,47,55)
cv2.namedWindow(window_name,cv2.WINDOW_GUI_EXPANDED) # Создать окно
масштабируемое
cv2.namedWindow(window_name_2,cv2.WINDOW_GUI_EXPANDED)
cv2.imshow(window_name,img) # Показать изображение в окне window_name
cv2.imshow(window_name_2,img_n )
if cv2.waitKey(0)==121: # При нажатии на Y сохранить img_n
    cv2.imwrite('task_1.jpg',img_n)
cv2.destroyAllWindows() # Удалить все окна
```

Результат работы программы:



Рисунок 1 Исходное изображение



Рисунок 2 Обработанное изображение

Задание 2.

Используя пороговые преобразования, добиться того, чтобы фигура девушки попала в одну группу, а фон – в другую. Какая функция для этого подошла лучше всего? Ручной поиск, адаптивный или автоматический?

Код программы:

```
import cv2
```

```

import numpy as np

path = 'Laba_2\\2-1.jpg'
window_name = 'figure_1'
window_name_2 = 'figure_2'
img=cv2.imread(path,flags=cv2.IMREAD_REDUCED_GRAYSCALE_2) # Прочитать фото в
ЧБ 1:2
# Ручной поиск: Обнуление яркости части пикселей изображения
_,img_n = cv2.threshold(img,80,255,cv2.THRESH_TOZERO)
cv2.namedWindow(window_name,cv2.WINDOW_AUTOSIZE) # Создать окно window_name
cv2.namedWindow(window_name_2,cv2.WINDOW_AUTOSIZE)
# Показать изображения
cv2.imshow(window_name,img)
cv2.imshow(window_name_2,img_n )
# При нажатии на Y сохранить img_n
if cv2.waitKey(0)==121:
    cv2.imwrite('task_2.jpg',img_n)
cv2.destroyAllWindows() # Удалить все окна

```

Результат работы программы:

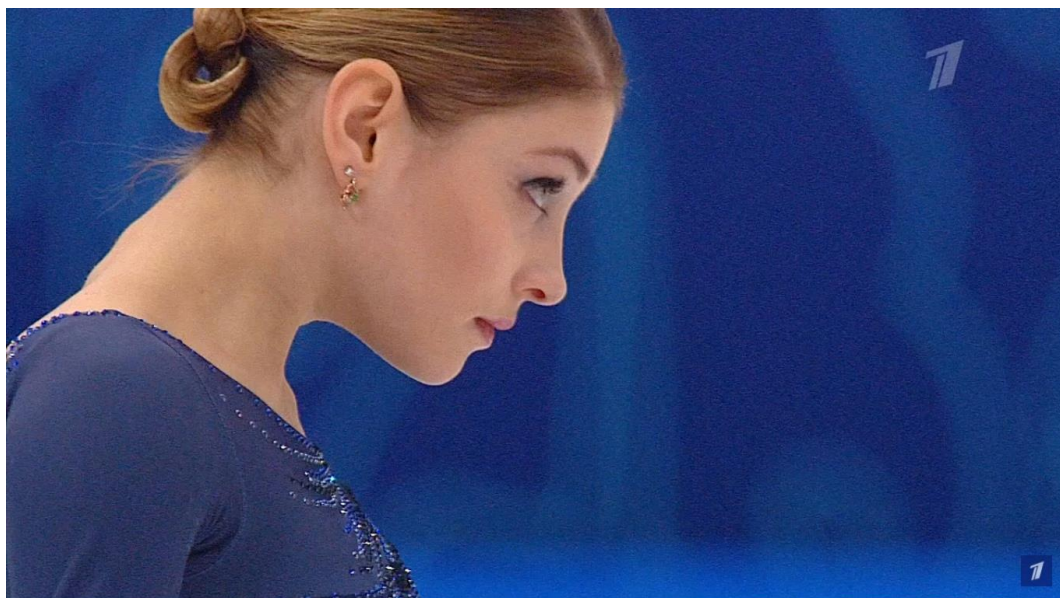


Рисунок 3 Исходное изображение



Рисунок 4 Обработанное изображение

Задание 3.

Используя пороговые преобразования, добиться того, чтобы текст попал в одну группу, а фон – в другую. Какая функция для этого подошла лучше всего? Ручной поиск, адаптивный или автоматический?

Код программы:

```
import cv2
import numpy as np

path = 'Laba_2\\2-3.png'
path_1 = 'Laba_2\\2-4.png'
window_name = 'figure_1'
```

```

window_name_2 = 'figure_2'
window_name_3 = 'figure_3'
window_name_4 = 'figure_4'
img=cv2.imread(path,flags=cv2.IMREAD_REDUCED_GRAYSCALE_2) # Считать фото в
ЧБ 1:2
img1=cv2.imread(path_1,cv2.IMREAD_REDUCED_GRAYSCALE_2)
# Адаптивное преобразования
img_n = cv2.adaptiveThreshold(img,255,
cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,7,13)
img_n1 = cv2.adaptiveThreshold(img1,255,
cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,11,28)
# _,img_n1 = cv2.threshold(img1,100,255,cv2.THRESH_BINARY)
# Создание окон
cv2.namedWindow(window_name,cv2.WINDOW_AUTOSIZE)
cv2.namedWindow(window_name_2,cv2.WINDOW_AUTOSIZE)
cv2.namedWindow(window_name_3,cv2.WINDOW_AUTOSIZE)
cv2.namedWindow(window_name_4,cv2.WINDOW_AUTOSIZE)
# Показать изображения
cv2.imshow(window_name,img)
cv2.imshow(window_name_2,img_n )
cv2.imshow(window_name_3,img1 )
cv2.imshow(window_name_4,img_n1 )
# Нажали Y - сохранить img_n и img_n1
if cv2.waitKey(0)==121:
    cv2.imwrite('task_3_1.jpg',img_n)
    cv2.imwrite('task_3_2.jpg',img_n1)
cv2.destroyAllWindows() # Удалить окна

```

Результат работы программы:

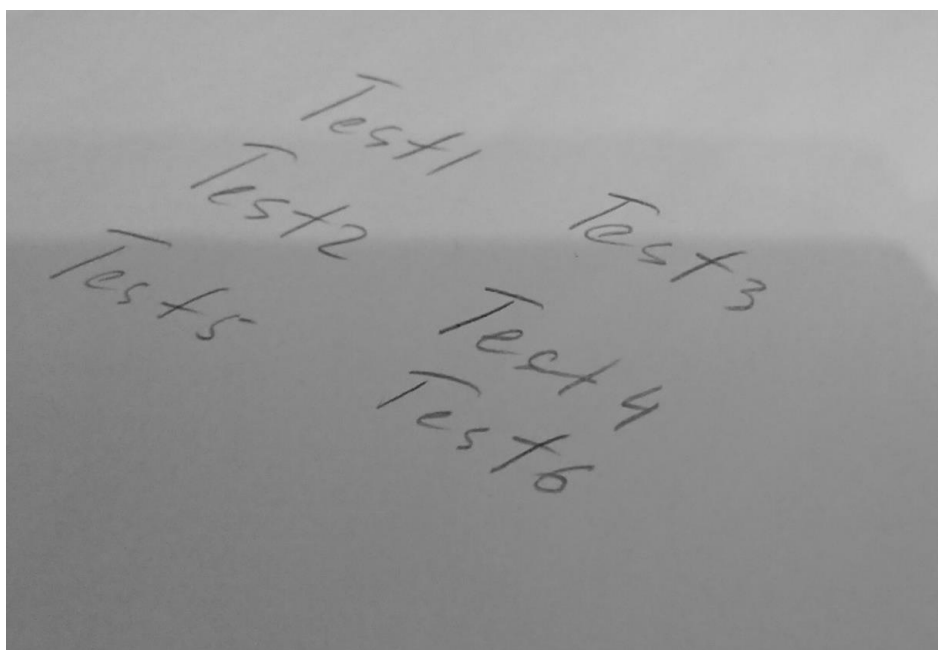


Рисунок 5 Исходное изображение

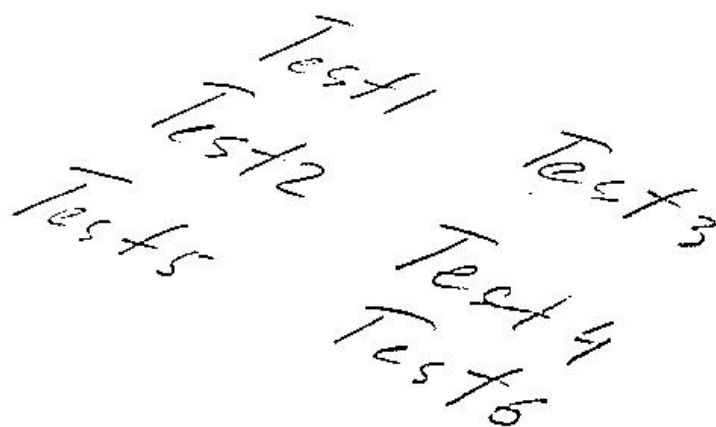


Рисунок 6 Преобразованное изображение

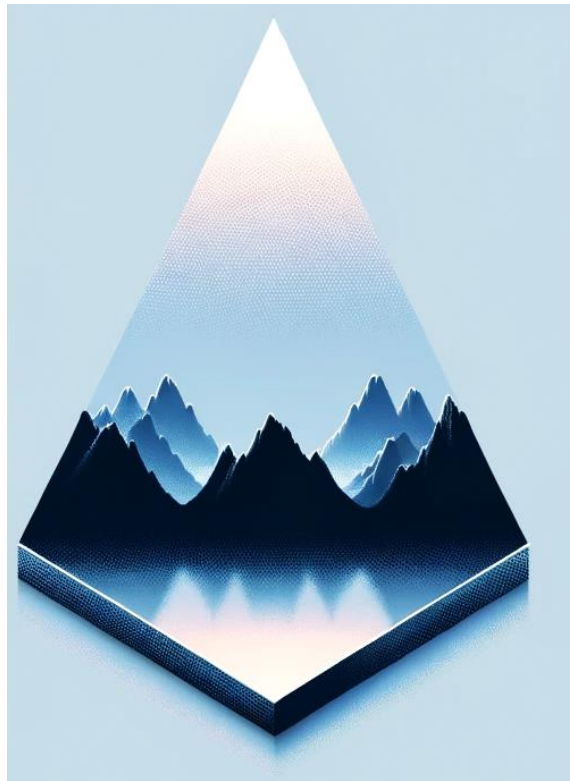


Рисунок 7 Исходное изображение

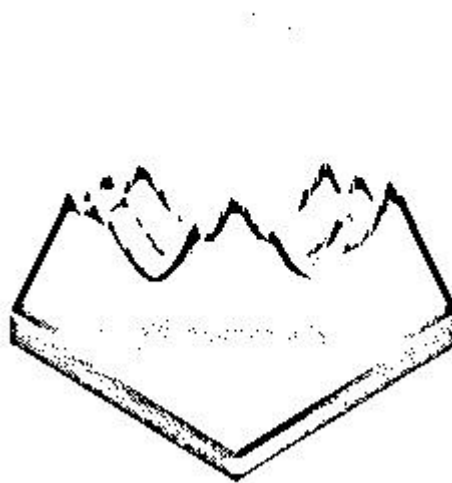


Рисунок 8 Преобразованное изображение

Задание 4.

Подключиться к камере. (Если камеры нет, попросить её у преподавателя).
Используя пороговые преобразования, добиться того, чтобы Вы попали в одну

группу, фон – в другую. Какая функция для этого подошла лучше всего? Ручной поиск, адаптивный или автоматический?

Код программы:

```
import cv2

cap = cv2.VideoCapture(0) # Создание экземпляра класса для захвата видео с
камеры 0
while(True):
    __, frame = cap.read() # Чтение одного кадра
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Перевод кадра в чб
    # Адаптивное изображение
    frame = cv2.adaptiveThreshold(frame, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 21, 8)
    cv2.imshow('frame', frame) # Отображение кадра
    # Ожидание нажатия на клавиши в течение 1 мс
    key=cv2.waitKey(1)
    # Нажали Y - сохранить кадр
    if key == 113:
        cv2.imwrite('task_4.jpg', frame)
    # Нажали ESC - выйти
    elif key==27:
        cv2.destroyAllWindows()
        break
```

Результат работы:



Рисунок 9 Результат работы программы

Задание 5.

На изображении выделить разметку.

Код программы:



Рисунок 10 Исходное изображение

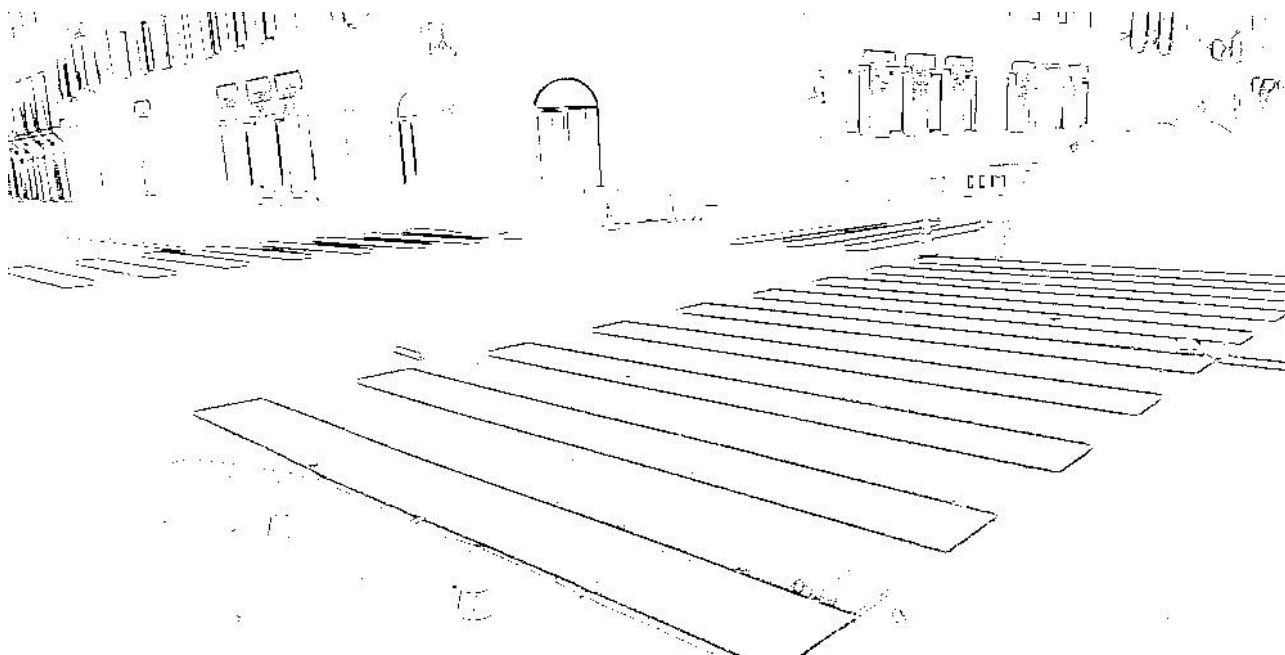


Рисунок 11 Преобразованное изображение

Вывод: В данной лабораторной работе проведено ознакомление с пороговыми преобразованиями для разделения пикселей на группы по признаку яркости. Чаще всего для обработки изображений использовалось адаптивное преобразование, которое позволяло выделять очертания объекта. Ручное преобразование было использовано для выделения девушки на фоне, сделав фон абсолютно чёрным, а девушку неизменной.