МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное   
образовательное учреждение высшего образования  
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики  
Кафедра технической кибернетики

**Отчет по лабораторной работе №2**

Дисциплина: «Системы обработки изображений»

Тема: **«Математическое представление изображений. Улучшение качества и препарирование изображений»**

Выполнил: Пшенин В.И.

Группа: 6133-010402D

Самара 2020

**Задание на лабораторную работу**

1. По гистограмме входного изображения определить значение порога яркостей, обеспечивающего оптимальное разделение объекта и фона. Осуществить пороговую обработку входного изображения с найденным пороговым значением.
2. Определить динамический диапазон входного изображения. Осуществить линейное контрастирование входного изображения в заданный динамический диапазон яркостей.
3. Осуществить препарирование изображения с заданной препарирующей функцией. Виды препарирующих функций для каждого варианта определены ранее.

|  |  |
| --- | --- |
| **№ варианта** | **Вид функции** |
| **4** | ris11_08 |

**Задание 1**

По гистограмме входного изображения определить значение порога яркостей, обеспечивающего оптимальное разделение объекта и фона. Осуществить пороговую обработку входного изображения с найденным пороговым значением.



Рисунок 1 – Входное изображение

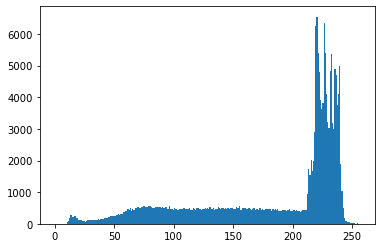


Рисунок 2 – Гистограмма входного изображения

Результат пороговой обработки с порогом равным 190 отражен на рисунке 3.

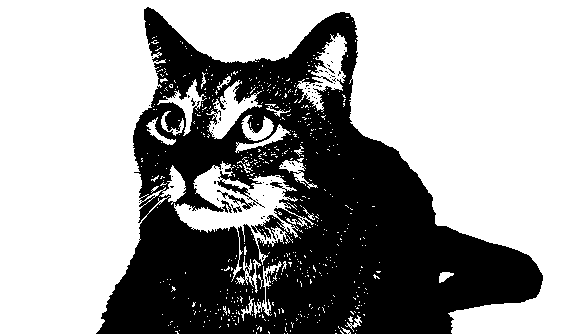


Рисунок 3 – Результат пороговой обработки входного изображения

**Задание 2**

Определить динамический диапазон входного изображения. Осуществить линейное контрастирование входного изображения в заданный динамический диапазон яркостей.



Рисунок 4 – Входное изображение

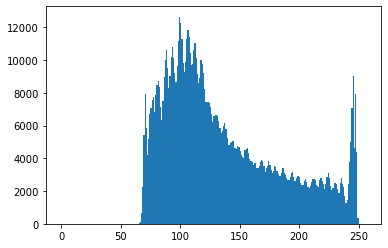


Рисунок 5 – Гистограмма входного изображения

Динамический диапазон изображения (53, 250). Осуществим линейное контрастирование входного изображения в заданный динамический диапазон яркостей. На рисунке 6 гистограмма изображения после линейного контрастирования.

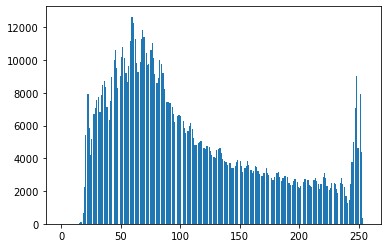


Рисунок 6 – Гистограмма изображения после линейного контрастирования

На рисунке 7 результат линейного контрастирования входного изображения в заданный динамический диапазон яркостей.



Рисунок 7 – Результат линейного контрастирования входного изображения

**Задание 3**

Осуществить препарирование изображения с заданной препарирующей функцией. Виды препарирующих функций для каждого варианта определены ранее.

В качестве входного изображения выберем рисунок 4.

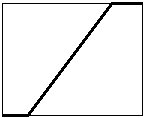


Рисунок 8 – Препарирующая функция

После осуществления препарирования изображения с заданной препарирующей функцией гистограмма изображения изменилась (рисунок 9), а результат операции изображен на рисунке 10.

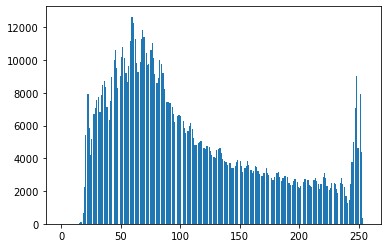


Рисунок 9 – Гистограмма изображения после препарирования изображения



Рисунок 10 – Результат препарирования изображения заданной препарирующей функцией

**ПРИЛОЖЕНИЕ А**

**Код программы**

import numpy as np

import matplotlib.pyplot as plt

from PIL import Image

# In[2]:

img = Image.open("cat.jpg").convert('L')

# In[3]:

img

# In[4]:

a = np.array(img.convert('L'))

a

# In[5]:

plt.hist(a.flatten(),256,[0,256])

plt.show()

# In[6]:

def threshold(intensity):

thr = 190

iI = intensity

iO = 0

if iI > thr:

iO = 255

return iO

# In[7]:

img2 = img.point(threshold)

img2

# In[3]:

img = Image.open("low.png")

img

# In[4]:

a = np.array(img)

plt.hist(a.flatten(),256,[0,256])

plt.show()

# In[6]:

minI = a.min()

maxI = a.max()

minO = 0

maxO = 255

minI, maxI

# In[7]:

def normalize(intensity):

iI = intensity

iO = ((iI - minI) / (maxI - minI)) \* (maxO - minO) + minO

return iO

# In[8]:

img2 = img.point(normalize)

img2

# In[9]:

a = np.array(img2)

plt.hist(a.flatten(),256,[0,256])

plt.show()

# In[10]:

a.min(),a.max()

# In[14]:

def taskNormalize(intensity):

iI = intensity

left = 53

right = 250

if iI < left:

return 0

if iI > right:

return 255

iO = ((iI - left) / (right - left)) \* (maxO - minO) + minO

return iO

# In[15]:

img3 = img.point(taskNormalize)

img3

# In[16]:

a = np.array(img3)

plt.hist(a.flatten(),256,[0,256])

plt.show()