

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Цифровая обработка изображений»**  
**Тема: Сегментация изображений**

Студентка гр. 7381

\_\_\_\_\_

Алясова А.Н.

Преподаватель

\_\_\_\_\_

Черниченко Д.А.

Санкт-Петербург

2021

### **Постановка задачи.**

Реализовать на языке Python с использованием библиотеки OpenCV программу, выполняющую сегментацию/кластеризацию изображения по цвету с использованием метода K-Means.

### **Входные данные.**

1. Цветное изображение в формате bmp, jpg.
2. Количество кластеров.

### **Выходные данные.**

Цветное изображение в формате bmp (result.bmp), являющиеся результатом сегментации (сегмент окрашивается средним цветом в кластере).

### Выполнение работы.

Загрузим цветное изображение в формате .jpg (рис.1).



Рисунок 1

Выберем количество кластеров  $k = 3$ :



Рисунок 2 – Сегментированное изображение с  $k = 3$

Выберем количество кластеров  $k = 6$ :

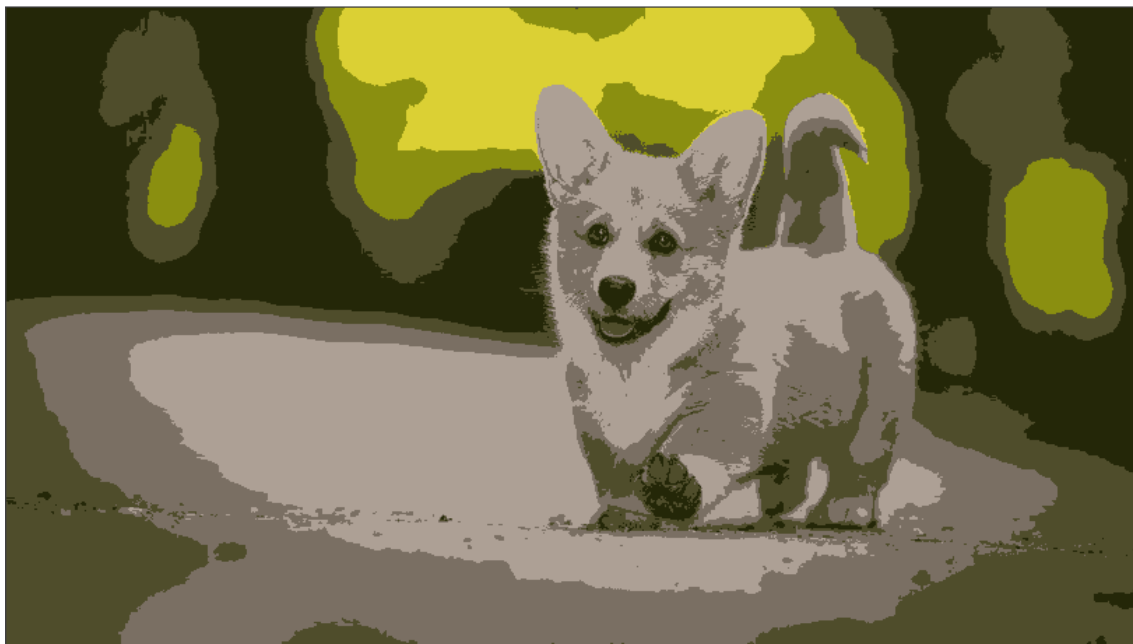


Рисунок 3 - Сегментированное изображение с  $k = 6$

Выберем количество кластеров  $k = 10$ :

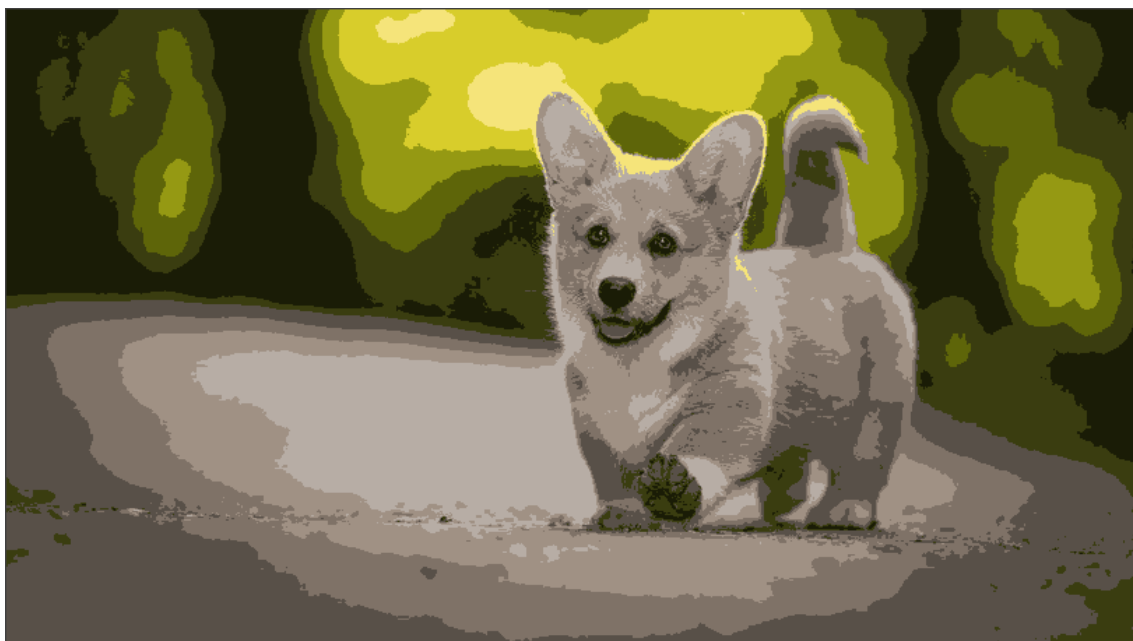


Рисунок 4 - Сегментированное изображение с  $k = 10$

Выберем количество кластеров  $k = 30$ :



Рисунок 5 - Сегментированное изображение с  $k = 30$

### **Выводы.**

В ходе выполнения данной работы была реализована программа на языке Python с использованием библиотеки OpenCV, выполняющая сегментацию/кластеризацию изображения по цвету с использованием метода K-Means.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД

```
import cv2
import numpy as np
import sys

filename = sys.argv[1]
k = int(sys.argv[2])
image = cv2.imread(filename)
cv2.imshow("Original image", image)
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 10,
1.0)
ret, label, center = cv2.kmeans(np.float32(image.reshape((-1, 3))), k,
None, criteria, 10, cv2.KMEANS_RANDOM_CENTERS)
center = np.uint8(center)
clusters = center[label.flatten()].reshape((image.shape))
cv2.imshow('Clusters', clusters)
cv2.imwrite('result.bmp', image)
cv2.waitKey(0)
```