# ***Занятие 24.*** *В поисках цвета: как находить объекты на изображениях*

## **О занятии**

**Краткое содержание:**

На этом занятии мы узнаем, как OpenCV используется для поиска объектов по цвету с помощью метода пороговой сегментации. Настроим рабочую среду для написания кода. Познакомимся с цветовыми пространствами HSV и BGR а также преобразуем изображение из одного пространства в другое. Научимся строить и вращать трехмерный график цветового рассеивания, найдем пороговые значения для определения нужных цветов. В заключение займемся сегментацией изображения по выбранным цветам, накладывая маски на изображение.

**Ключевые навыки:**

* Знание метода пороговой сегментации.
* Преобразование из одного цветового пространства в другое.
* Построение трехмерного графика цветового рассеивания.
* Установка порогов для определения нужного диапазона цветов.
* Сегментация изображения по выбранным цветам.

**Необходимые инструменты:**

Гигачат, Visual Studio Code.

**Термины:**

***Пороговая сегментация*** — метод, в котором устанавливается пороговое значение для разделения изображения на две части.

***Цветовое пространство HSV*** — это модель представления цвета, включающая три компонента: Hue (Оттенок), измеряемый в градусах от 0 до 360, Saturation (Насыщенность), измеряемую в процентах, и Value (Яркость), также измеряемую в процентах.

***Трехмерный график*** — это графическое представление данных в трех измерениях, где каждая точка отображает значение по трем осям.

## 

## **Практическая работа**

### Задание 1.

Прежде чем приступить к преобразованию изображения в нужное цветовое пространство, необходимо настроить рабочую среду:

1. Создать новую папку «Сегментация».
2. В Visual Studio Code в левом верхнем углу нажать кнопку «Файл», выбрать «Открыть папку», затем выбрать на компьютере созданную папку «Сегментация» и нажать «Выбор папки».
3. Создать в этой же папке “Segmentation”новый файл с расширением .py, а также добавить эту папку все необходимые для урока изображения.
4. Убедиться, что установлены библиотеки OpenCV, MatPlotLib и NumPy.

#### Задание для самостоятельной работы:

Настройте рабочую среду, следуя рекомендациям в материалах к занятию.

### Задание 2.

Возьмите одну из подготовленных картинок под названием tiger1.png и преобразуйте ее в цветовое пространство HSV. Картинка находится в материалах к занятию в файле «tiger1.png».

Для начала подключите библиотеки:

* OpenCV — для обработки изображения.
* Matplotlib — для построения графиков.
* NumPy — для вычислений.

OpenCV по умолчанию использует цветовое пространство BGR, которое аналогично RGB, но с другим порядком цветов. Поэтому вам нужно сначала преобразовать изображение из BGR в RGB, а затем в HSV, чтобы цвета отображались правильно. Для этого используйте функцию imread, чтобы загрузить изображение в формате BGR: tiger = cv2.imread('tiger1.png').

Далее выполните два преобразования:

1. Из BGR в RGB.
2. Из RGB в HSV.

#### Задание для самостоятельной работы:

Выполните преобразование изображения в HSV, применив данный код:

import cv2

import matplotlib.pyplot as plt

import numpy as np

from matplotlib import colors

tiger = cv2.imread('tiger1.png')

tiger = cv2.cvtColor(tiger, cv2.COLOR\_BGR2RGB)

tiger = cv2.cvtColor(tiger, cv2.COLOR\_BGR2RGB)

### Задание 3.

Самостоятельно перенесите код по построению трехмерного графика к себе в файл. Он находится в материалах к занятию в файле “Segmentation\_1.py”

### Задание 4.

**Промпт:** Объясни этот код кратко:   
 pixel\_colors = tiger.reshape((np.shape(tiger)[0]\*np.shape(tiger)[1], 3))

norm = colors.Normalize(vmin=-1.,vmax=1.)

norm.autoscale(pixel\_colors)

pixel\_colors = norm(pixel\_colors).tolist()

axis.scatter(h.flatten(), s.flatten(), v.flatten(), facecolors=pixel\_colors, marker=".")

axis.set\_xlabel("Hue")

axis.set\_ylabel("Saturation")

axis.set\_zlabel("Value")

plt.show()

**Возможный результат:**

Этот код создает цветовой спектр изображения тигра:

1. Создается переменная pixel\_colors с цветами каждого пикселя изображения.
2. Функция reshape преобразует переменную в нужный формат.
3. Применяется инструмент Normalize из Matplotlib для нормализации цветов.
4. Цвета преобразуются в список.
5. Функция scatter визуализирует цветовые спектры пикселей.
6. Устанавливаются названия осей графика и отображается результат.

### Задание 5.

Для определения нужного диапазона оранжевого цвета установите пороговые значения. Оцените значения по осям на графике. Для этого поворачивайте график, двигая мышь с зажатой левой кнопкой. Для более точных значений смотрите в правый нижний угол графика, наведя мышку на нужную область. Обращайте внимание только на целые числа.

Затем составьте светлую и темную версию, прописав в каждом три целых числа. Получите значения нижнего и верхнего порога оранжевого цвета.

light\_orange = (11, 80, 0)

dark\_orange = (19, 255, 255)

#### Задание для самостоятельной работы:

Попробуйте таким же образом определить пороговые значения белого цвета, используя тот же график цветового рассеивания.

### Задание 6.

Самостоятельно перенесите код для сегментации изображения по выбранным цветам. Он находится в материалах к занятию в файле “Segmentation\_2.py”

## 

## **Итоги занятия**

Сегодня на занятии вы:

* Познакомились с пороговой сегментацией.
* Узнали новые модели цветового пространства.
* Научились строить трехмерный график цветового рассеивания.
* Приобрели навык установки порогов для поиска.

Вы хорошо потрудились! Увидимся на следующем занятии!