Государственное учреждение образования

“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Кафедра: Интеллектуальных информационных технологий

Дисциплина: Обработка изображений в интеллектуальных системах

**Отчет по лабораторной работе №6**

**“Распознавание объектов”**

Выполнил:

Студент гр. 221703 Вечорко Д. Н.

Проверил: Сальников Д.А.

Минск 2024

**Распознавание объектов состоит из нескольких этапов**:

* **Идентификация объектов**: Определение присутствия конкретных объектов в изображении (например, распознавание лиц, автомобилей, животных и т.д.).
* **Классификация**: Разделение обнаруженных объектов на категории (например, различение собак и кошек на изображениях).
* **Сегментация**: Выделение конкретных частей изображения в отдельные объекты или области.

**Цель:**

Реализовать программу, позволяющую осуществить поиск объектов на изображении.

**Ход работы:**

1. Описание алгоритма:

* Инициализируем детектор MTCNN
* Загружаем изображение, на котором хотим распознать лицо.
* Загружаем изображение, которое будет использоваться для замены лица.
* Выполняем распознавание лиц с помощью детектора и помещаем все найденные лица в массив.
* Для каждого найденного лица изменяем размер изображения замены под размеры распознанного лица.
* Заменяем распознанное лицо на изображение замены в исходном изображении.
* Выводим результат с замененными лицами.

1. Средства разработки:

* Язык программирования - Python.
* Библиотеки matplotlib, cv2, mtcnn.
* Классификатор MTCNN (Multi-Task Cascaded Convolutional Networks.

#### Основные этапы работы MTCNN:

1. **Обнаружение лиц (Face Detection)**: MTCNN использует каскад из трех сетей:
   * P-Net (Proposal Network): выявляет области, в которых может находиться лицо.
   * R-Net (Refinement Network): уточняет эти области.
   * O-Net (Output Network): финализирует границы лица и предсказывает ключевые точки.
2. **Определение ключевых точек лица (Facial Landmarks)**: Ключевые точки включают глаза, нос и углы рта, которые могут быть полезны для анализа позы лица.
3. **Замена изображения**

3. Результат работы:





Код:

from mtcnn import MTCNN  
import cv2  
import matplotlib.pyplot as plt  
  
# Инициализация детектора  
detector = MTCNN()  
  
# Загрузка изображения с лицом  
image = cv2.cvtColor(cv2.imread('../3.jpg'), cv2.COLOR\_BGR2RGB)  
  
# Загрузка изображения для замены лица  
replacement\_face = cv2.cvtColor(cv2.imread('../cat.png'), cv2.COLOR\_BGR2RGB)  
  
# Распознавание лиц  
faces = detector.detect\_faces(image)  
  
# Отметка лиц на изображении и замена  
for face in faces:  
 x, y, width, height = face['box']  
  
 # Изменение размера изображения замены под размер распознанного лица  
 replacement\_face\_resized = cv2.resize(replacement\_face, (width, height))  
  
 # Замена лица в исходном изображении  
 image[y:y + height, x:x + width] = replacement\_face\_resized  
  
# Отображение результата  
plt.imshow(image)  
plt.axis('off')  
plt.show()

Вывод:

Изучили модель MTCNN и научились с ней работать.