



у меня есть python функция

Пример использования:

```
result = face_recognition_dct(
    'test_face.jpg',
    ['reference1.jpg', 'reference2.jpg'],
    block_size=8,
    num_coefficients=15,
    threshold=0.85
)
```

print("Тот же человек" if result else "Другой человек")

напиши программу, которая будет использовать эту функцию. я хочу чтобы на вход она получала директорию с изображениями .jpg человека, которые будут использоваться для вычисления эталонов. а также директорию с тестовыми изображениями. на выходе хочу видеть номер тестового изображения и результат для него.

```
import os
from pathlib import Path

def process_face_recognition(reference_dir, test_dir, block_size=8, num_coefficients=15,
    """
    Обрабатывает распознавание лиц для всех тестовых изображений.

    Args:
        reference_dir: путь к директории с эталонными изображениями
        test_dir: путь к директории с тестовыми изображениями
        block_size: размер блока для DCT
        num_coefficients: количество коэффициентов
        threshold: порог сравнения
    """
    # Получаем список эталонных изображений
```

```

reference_images = [
    os.path.join(reference_dir, f)
    for f in os.listdir(reference_dir)
    if f.lower().endswith('.jpg')
]

if not reference_images:
    print("Ошибка: эталонные изображения не найдены")
    return

print(f"Найдено эталонных изображений: {len(reference_images)}")
print("-" * 50)

# Получаем список тестовых изображений
test_images = sorted([
    f for f in os.listdir(test_dir)
    if f.lower().endswith('.jpg')
])

if not test_images:
    print("Ошибка: тестовые изображения не найдены")
    return

# Обрабатываем каждое тестовое изображение
for idx, test_image in enumerate(test_images, 1):
    test_path = os.path.join(test_dir, test_image)

    result = face_recognition_dct(
        test_path,
        reference_images,
        block_size=block_size,
        num_coefficients=num_coefficients,
        threshold=threshold
    )

    status = "Тот же человек ✓" if result else "другой человек ✗"
    print(f"Изображение #{idx} ({test_image}): {status}")

print("-" * 50)
print(f"Обработано изображений: {len(test_images)}")

if __name__ == "__main__":
    # Укажите пути к директориям
    REFERENCE_DIR = "path/to/reference_images"
    TEST_DIR = "path/to/test_images"

    process_face_recognition(
        reference_dir=REFERENCE_DIR,
        test_dir=TEST_DIR,
        block_size=8,
        num_coefficients=15,
        threshold=0.85
    )

```

Программа автоматически находит все .jpg файлы в обеих директориях, обрабатывает их и выводит пронумерованные результаты.

Хотите добавить сохранение результатов в файл или визуализацию?

параметры функции изменились. перепиши код. выводи в результатах максимальное сходство.

Пример использования:

```
is_match, similarity = face_recognition_dct(  
    'test_face.jpg',  
    ['reference1.jpg', 'reference2.jpg'],  
    block_size=8,  
    num_coefficients=15,  
    threshold=0.85  
)  
  
print(f"Результат: {'Тот же человек' if is_match  
else 'Другой человек'}")  
  
print(f"Максимальное сходство: {similarity:.4f}")
```

```
import os  
from pathlib import Path  
  
def process_face_recognition(reference_dir, test_dir, block_size=8, num_coefficients=15,  
    """  
        Обрабатывает распознавание лиц для всех тестовых изображений.  
  
    Args:  
        reference_dir: путь к директории с эталонными изображениями  
        test_dir: путь к директории с тестовыми изображениями  
        block_size: размер блока для DCT  
        num_coefficients: количество коэффициентов  
        threshold: порог сравнения  
    """  
        # Получаем список эталонных изображений  
        reference_images = [
```

```

        os.path.join(reference_dir, f)
        for f in os.listdir(reference_dir)
        if f.lower().endswith('.jpg')
    ]

if not reference_images:
    print("Ошибка: эталонные изображения не найдены")
    return

print(f"Найдено эталонных изображений: {len(reference_images)}")
print("-" * 50)

# Получаем список тестовых изображений
test_images = sorted([
    f for f in os.listdir(test_dir)
    if f.lower().endswith('.jpg')
])

if not test_images:
    print("Ошибка: тестовые изображения не найдены")
    return

# Обрабатываем каждое тестовое изображение
for idx, test_image in enumerate(test_images, 1):
    test_path = os.path.join(test_dir, test_image)

    is_match, similarity = face_recognition_dct(
        test_path,
        reference_images,
        block_size=block_size,
        num_coefficients=num_coefficients,
        threshold=threshold
    )

    status = "Тот же человек ✓" if is_match else "Другой человек ✗"
    print(f"Изображение #{idx} ({test_image}): {status} | Сходство: {similarity:.4f}")

print("-" * 50)
print(f"Обработано изображений: {len(test_images)}")

if __name__ == "__main__":
    # Укажите пути к директориям
    REFERENCE_DIR = "path/to/reference_images"
    TEST_DIR = "path/to/test_images"

    process_face_recognition(
        reference_dir=REFERENCE_DIR,
        test_dir=TEST_DIR,
        block_size=8,
        num_coefficients=15,
        threshold=0.85
    )

```

Теперь программа выводит максимальное сходство для каждого тестового изображения.

