**1 Часть:**

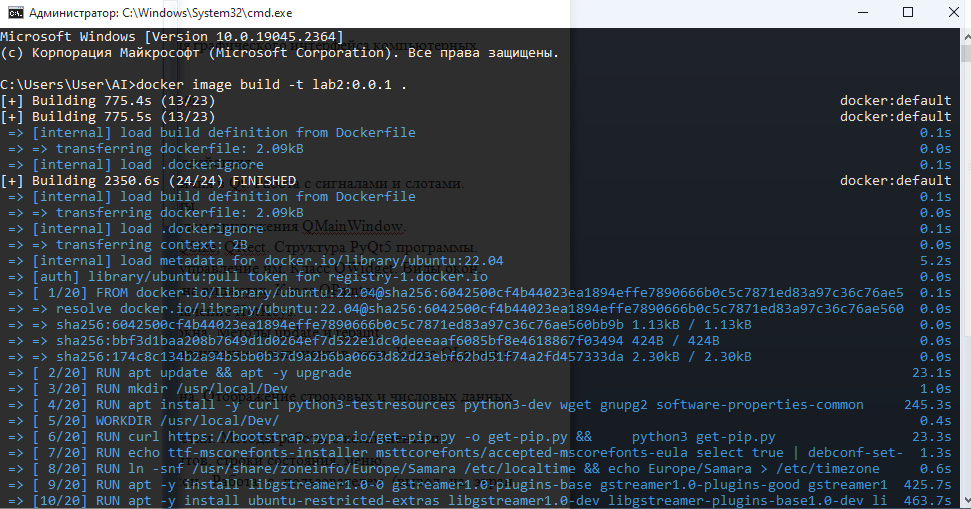
**Описание:**

В данной программе выполняется пороговая обработка изображения, сохраняется результат в файл и копируется в указанную целевую папку.

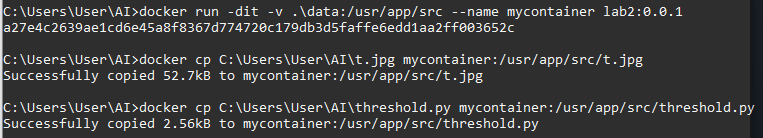
**Исходное изображение:**

****

**Билд проекта:**



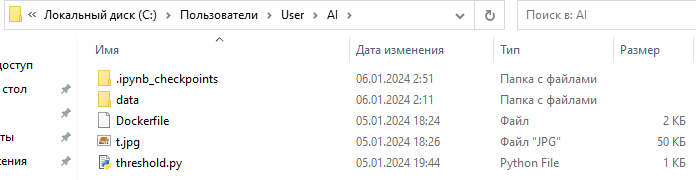
**Запуск проекта и копирование данных из локальной машины в файловую систему компьютера:**

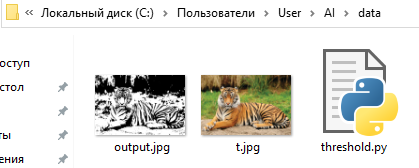


**Запуск скрипта:**



**Файловая система:**

****



**Обработанное изображение:**

****

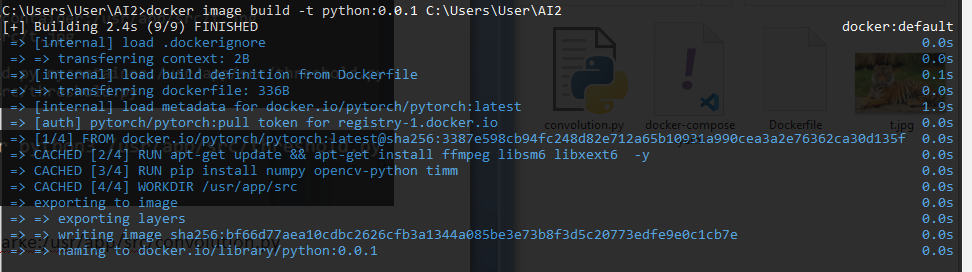
**2 Часть:**

**Описание:**

В данной программе используется простая архитектура с двумя сверточными слоями. Первый сверточный слой имеет 64 канала с размером ядра 3x3, а второй сверточный слой преобразует 64 канала обратно в 3 канала. Между сверточными слоями используется функция активации ReLU для введения нелинейности.

**Билд проекта:**

docker image build -t python:0.0.1 C:\Users\User\AI2



**Запуск проекта:**

docker run -dit -v C:\Users\User\AI2\data:/usr/app/src python:0.0.1



**Копирование данных из локальной машины в файловую систему компьютера:**

docker cp C:\Users\User\AI2\convolution.py distracted\_clarke:/usr/app/src/convolution.py



docker cp C:\Users\User\AI2\t.jpg distracted\_clarke:/usr/app/src/t.jpg



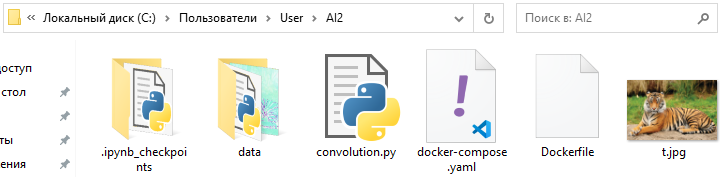
**Запуск скрипта:**

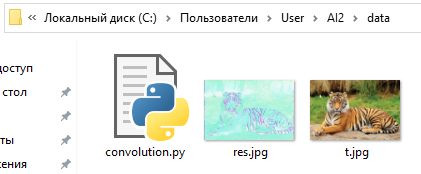
docker exec -it distracted\_clarke python3 /usr/app/src/convolution.py



В первой части все аналогично, поэтому так подробно не описывал.

**Файловая система:**





**Полученное изображение:**

