

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»
(Самарский университет)

Институт информатики и кибернетики
Кафедра технической кибернетики

Отчет по лабораторной работе №2

Дисциплина: «Технологии искусственного интеллекта»

Тема: «**Docker Compose**»

Выполнил: Каспаров И.А.

Группа: 6232-010402D

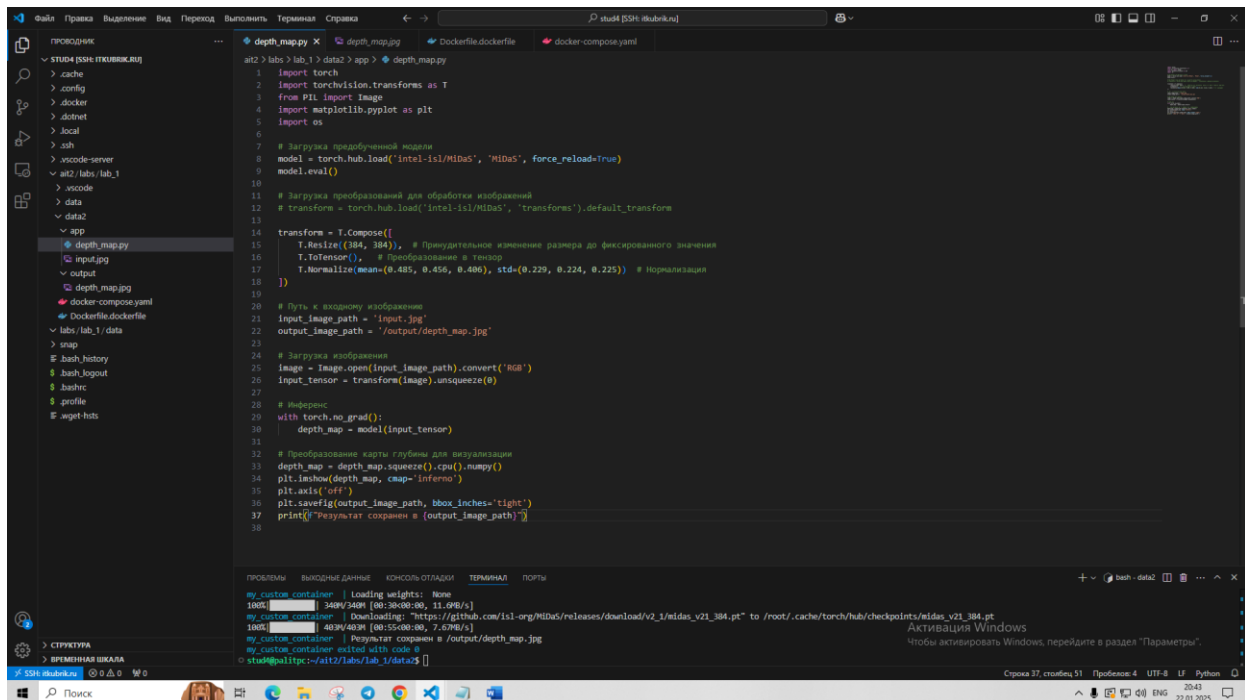
Самара 2024

Задание 1

Запустить предобученную нейронку с использованием pytorch внутри контейнера. Для создания контейнера использовать Docker Compose.

1. Собрать контейнер с установленным PyTorch (CPU или GPU версия).
2. Написать скрипт обработки изображений с использованием нейросети. Можно выбрать любую понравившуюся модель/задачу обработки изображения нейросетью.
3. Запустить контейнер командой:
`docker compose -f <имя_конфига.yaml> up`
4. Запустить скрипт с реализованным алгоритмом в контейнере в примонитрованной внутри контейнера папке. Результат обработки сохранить в локальной директории контейнера.
5. Убедиться в появлении результата в директории хоста.

1) Скрипт depth_map:



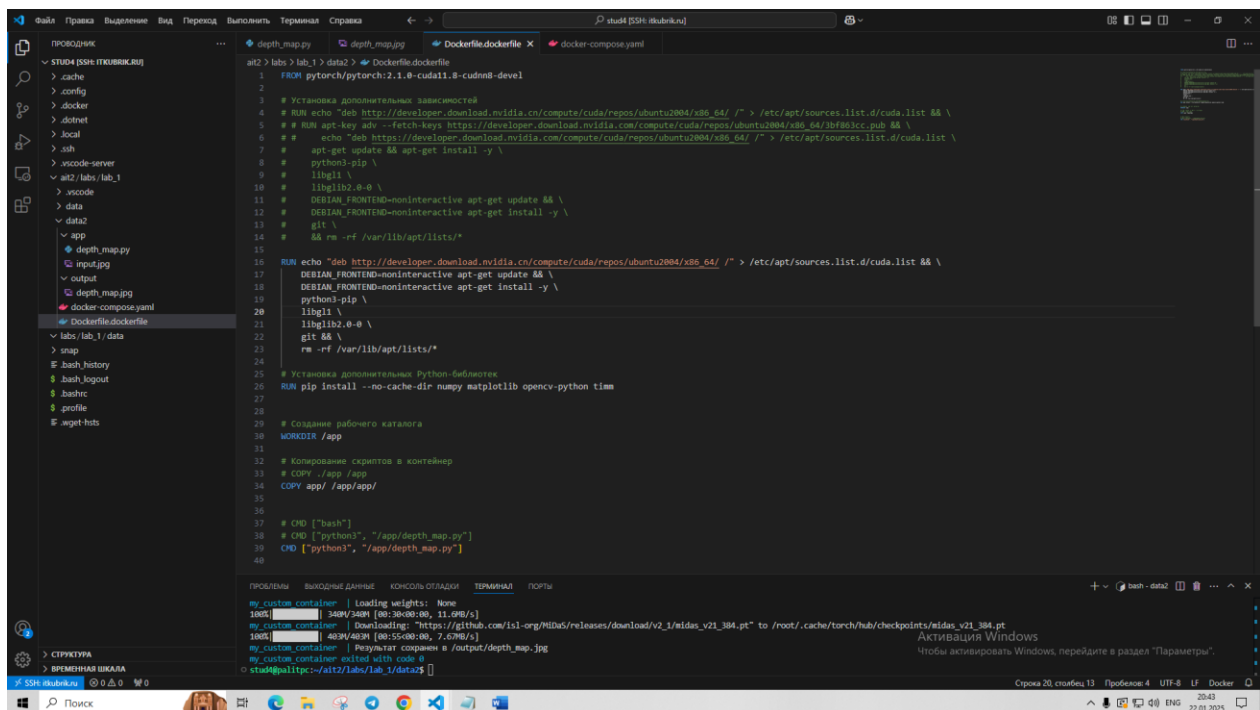
```
1 import torch
2 import torchvision.transforms as T
3 from PIL import Image
4 import matplotlib.pyplot as plt
5 import os
6
7 # Загрузка преобученной модели
8 model = torch.hub.load('intel-isl/MiDaS', 'MiDaS', force_reload=True)
9 model.eval()
10
11 # Загрузка преобученной для оптического изображения
12 # transform = torch.hub.load('intel-isl/MiDaS', 'transforms').default_transform
13
14 transform = T.Compose([
15     T.Resize((384, 384)), # Присутительное изменение размера до фиксированного значения
16     T.ToTensor(), # Преобразование в тензор
17     T.Normalize(mean=(0.485, 0.456, 0.406), std=(0.229, 0.224, 0.225)) # Нормализация
18 ])
19
20 # Путь к входному изображению
21 input_image_path = 'input.jpg'
22 output_image_path = '/output/depth_map.jpg'
23
24 # Загрузка изображения
25 image = Image.open(input_image_path).convert('RGB')
26 input_tensor = transform(image).unsqueeze(0)
27
28 # Инференс
29 with torch.no_grad():
30     depth_map = model(input_tensor)
31
32 # Преобразование карты глубины для визуализации
33 depth_map = depth_map.squeeze().cpu().numpy()
34 plt.imshow(depth_map, cmap='inferno')
35 plt.axis('off')
36 plt.savefig(output_image_path, bbox_inches='tight')
37 print("Результат сохранен в " + output_image_path)
```

PROBLEMY ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТАЗОВ ТЕРМИНАЛ ПОРТЫ

```
my_custom_container | Loading weights: None
100% | 348K/348K | (0:30:00:00, 11.0MB/s)
my_custom_container | Downloading: "https://github.com/intel-isl/MiDaS/releases/download/v2.1/midas_v21_384.pt" to /root/.cache/torch/hub/checkpoints/midas_v21_384.pt
100% | 483K/483K | (0:55:00:00, 7.6MB/s)
my_custom_container | Результат сохранен в /output/depth_map.jpg
my_custom_container exited with code 0
stud@balltcr-ait2/labs/lab_1/data2$
```

Этот код выполняет задачу создания карты глубины (depth map) из входного изображения с использованием преобученной модели MiDaS.

2) Dockerfile:

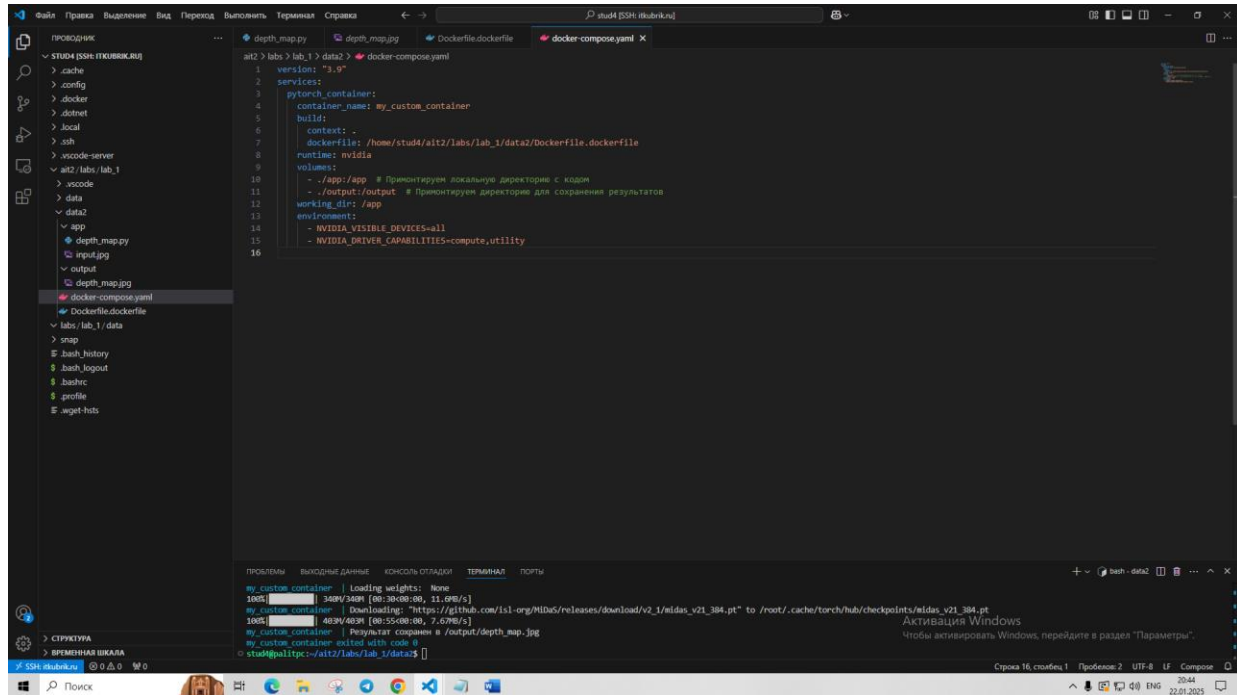


```
1 FROM pytorch/pytorch:2.1.0-cuda11.8-cudnn8-devel
2
3 # Установка дополнительных зависимостей
4 # RUN echo "deb http://developer.download.nvidia.cn/compute/cuda/repos/ubuntu2004/x86_64/ /" > /etc/apt/sources.list.d/cuda.list && \
5 # RUN apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/3bf863cc.pub && \
6 # # echo "deb https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/ /" > /etc/apt/sources.list.d/cuda.list \
7 # apt-get update && apt-get install -y \
8 # python3-pip \
9 # libgl1 \
10 # libglu1 \
11 # DEBIAN_FRONTEND=noninteractive apt-get update && \
12 # DEBIAN_FRONTEND=noninteractive apt-get install -y \
13 # git \
14 # && rm -rf /var/lib/apt/lists/*
15
16 RUN echo "deb http://developer.download.nvidia.cn/compute/cuda/repos/ubuntu2004/x86_64/ /" > /etc/apt/sources.list.d/cuda.list && \
17 DEBIAN_FRONTEND=noninteractive apt-get update && \
18 DEBIAN_FRONTEND=noninteractive apt-get install -y \
19 python3-pip \
20 libgl1 \
21 libglu1 \
22 git && \
23 rm -rf /var/lib/apt/lists/*
24
25 # Установка дополнительных Python-библиотек
26 RUN pip install --no-cache-dir numpy matplotlib opencv-python timm
27
28 # Создание рабочего каталога
29 WORKDIR /app
30
31 # Копирование скриптов в контейнер
32 # COPY ./app /app
33 COPY app/ /app/app/
34
35 # CMD ["bash"]
36 # CMD ["python", "/app/depth_map.py"]
37 CMD ["python", "/app/depth_map.py"]
```

PROBLEMY ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТАЗОВ ТЕРМИНАЛ ПОРТЫ

```
my_custom_container | Loading weights: None
100% | 348K/348K | (0:30:00:00, 11.0MB/s)
my_custom_container | Downloading: "https://github.com/intel-isl/MiDaS/releases/download/v2.1/midas_v21_384.pt" to /root/.cache/torch/hub/checkpoints/midas_v21_384.pt
100% | 483K/483K | (0:55:00:00, 7.6MB/s)
my_custom_container | Результат сохранен в /output/depth_map.jpg
my_custom_container exited with code 0
stud@balltcr-ait2/labs/lab_1/data2$
```

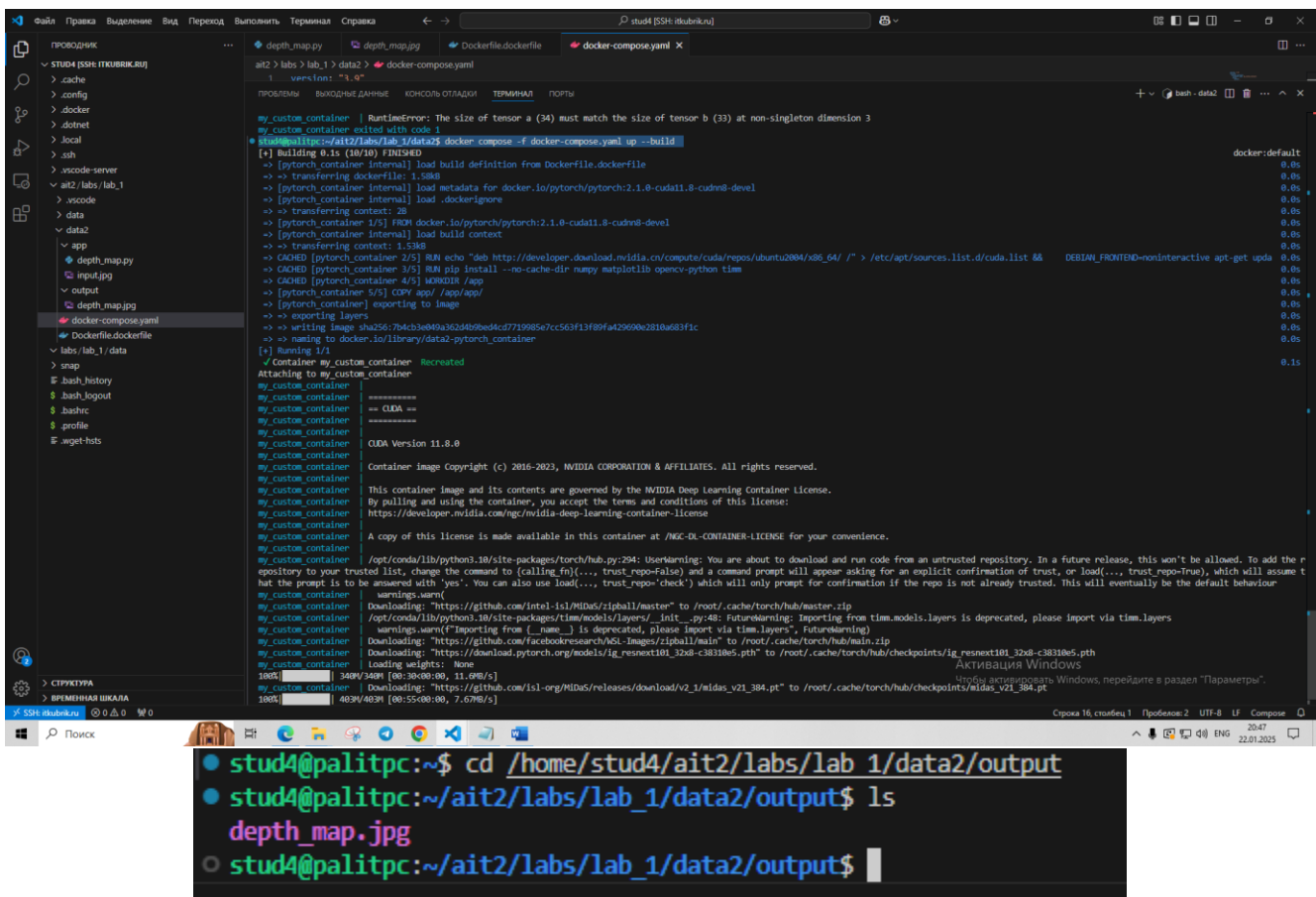
3) Docker-compose.



```
al2 > labs > lab_1 > data2 > docker-compose.yml
1 version: "3.9"
2 services:
3   pytorch_container:
4     container_name: my_custom_container
5     build:
6       context: .
7       dockerfile: /home/stud4/ait2/labs/lab_1/data2/Dockerfile.dockerfile
8     runtime: nvidia
9     volumes:
10      - ./app/app : Промониторим локальную директорию с кодом
11      - ./output/output : Промониторим директорию для сохранения результатов
12     working_dir: /app
13     environment:
14      - NVIDIA_VISIBLE_DEVICES=all
15      - NVIDIA_DRIVER_CAPABILITIES=compute,utility
16
```

```
my_custom_container | Loading weights: None
100% | 349V/349M [00:30:00:00, 11.09M/s]
my_custom_container | Downloading: "https://github.com/isl-org/MiDaS/releases/download/v2.1/midas_v21_384.pt" to /root/.cache/torch/hub/checkpoints/midas_v21_384.pt
100% | 403V/403M [00:55:00:00, 7.67M/s]
my_custom_container | PyTorch сохранен в /output/depth_map.jpg
my_custom_container exited with code 0
stud4@palitpc:~/ait2/labs/lab_1/data2$
```

4) Запуск собранного контейнера.



```
al2 > labs > lab_1 > data2 > docker-compose.yml
1 version: "3.9"
2 services:
3   pytorch_container:
4     container_name: my_custom_container
5     build:
6       context: .
7       dockerfile: /home/stud4/ait2/labs/lab_1/data2/Dockerfile.dockerfile
8     runtime: nvidia
9     volumes:
10      - ./app/app : Промониторим локальную директорию с кодом
11      - ./output/output : Промониторим директорию для сохранения результатов
12     working_dir: /app
13     environment:
14      - NVIDIA_VISIBLE_DEVICES=all
15      - NVIDIA_DRIVER_CAPABILITIES=compute,utility
16
```

```
my_custom_container | Runtime error: The size of tensor a (34) must match the size of tensor b (33) at non-singleton dimension 3
my_custom_container exited with code 1
stud4@palitpc:~/ait2/labs/lab_1/data2$ docker compose -f docker-compose.yml up --build
[+] Building 0.1s (18/18) FINISHED
=> [pytorch_container Internal] load build definition from Dockerfile.dockerfile
=> [pytorch_container Internal] load metadata for docker.io/pytorch/pytorch:2.1.0-cuda11.8-cudnn8-devel
=> [pytorch_container Internal] load .dockerignore
=> [pytorch_container Internal] transfering context: 2B
=> [pytorch_container 1/5] FROM docker.io/pytorch/pytorch:2.1.0-cuda11.8-cudnn8-devel
=> [pytorch_container Internal] load build context
=> [pytorch_container Internal] transfering context: 1.53kB
=> CACHED [pytorch_container 2/5] RUN echo "set http://developer.download.nvidia.cn/compute/cuda/repos/ubuntu2004/x86_64/ /" > /etc/apt/sources.list.d/cuda.list && DEBIAN_FRONTEND=noninteractive apt-get update
=> CACHED [pytorch_container 3/5] RUN pip install --no-cache-dir numpy matplotlib opencv-python timm
=> CACHED [pytorch_container 4/5] WORKDIR /app
=> [pytorch_container 5/5] COPY app/ /app/app/
=> [pytorch_container Internal] exporting to image
=> [pytorch_container Internal] writing image sha256:7b4cb3b840a362d49b6dc4d771985e7cc563f13f89fa429690e2810a683f1c
=> [pytorch_container Internal] naming to docker.io/library/data2-pytorch_container
[+] Naming 1/1
Container my_custom_container Recreated
Attaching to my_custom_container
my_custom_container | =====
my_custom_container | == CUDA ==
my_custom_container | =====
my_custom_container | CUDA Version 11.8.0
my_custom_container | Container image Copyright (c) 2016-2023, NVIDIA CORPORATION & AFFILIATES. All rights reserved.
my_custom_container | This container image and its contents are governed by the NVIDIA Deep Learning Container License.
my_custom_container | By pulling and using the container, you accept the terms and conditions of this license:
my_custom_container | https://developer.nvidia.com/ngc/nvidia-deep-learning-container-license
my_custom_container | A copy of this license is made available in this container at /NGC-CL-CONTAINER-LICENSE for your convenience.
my_custom_container | /opt/conda/lib/python3.10/site-packages/torch/hub.py:294: UserWarning: You are about to download and run code from an untrusted repository. In a future release, this won't be allowed. To add the repository to your trusted list, change the command to (calling fn(..., trust_repo=False) and a command prompt will appear asking for an explicit confirmation of trust, or load(..., trust_repo=True), which will assume that the prompt is to be answered with 'yes'. You can also use load(..., trust_repo='check') which will only prompt for confirmation if the repo is not already trusted. This will eventually be the default behaviour
my_custom_container | warnings.warn(
my_custom_container | Downloading: "https://github.com/isl-org/MiDaS/releases/download/v2.1/midas_v21_384.pt" to /root/.cache/torch/hub/main.zip
my_custom_container | /opt/conda/lib/python3.10/site-packages/timm/models/layers/_init_.py:48: FutureWarning: Importing from timm.models.layers is deprecated, please import via timm.layers
my_custom_container | warnings.warn(f"Importing from {__name__} is deprecated, please import via timm.layers", FutureWarning)
my_custom_container | Downloading: "https://github.com/facebookresearch/HSI-Images/zipball/main" to /root/.cache/torch/hub/main.zip
my_custom_container | Loading weights: None
my_custom_container | Downloading: "https://download.pytorch.org/models/ig_resnext101_32x8-c3b19e5.pth" to /root/.cache/torch/hub/checkpoints/ig_resnext101_32x8-c3b19e5.pth
100% | 349V/349M [00:30:00:00, 11.09M/s]
my_custom_container | Downloading: "https://github.com/isl-org/MiDaS/releases/download/v2.1/midas_v21_384.pt" to /root/.cache/torch/hub/checkpoints/midas_v21_384.pt
100% | 403V/403M [00:55:00:00, 7.67M/s]
```

```
stud4@palitpc:~$ cd /home/stud4/ait2/labs/lab_1/data2/output
stud4@palitpc:~/ait2/labs/lab_1/data2/output$ ls
depth_map.jpg
stud4@palitpc:~/ait2/labs/lab_1/data2/output$
```

Исходное изображение:



Карта глубины изображения:

