

СОГЛАСОВАНО

УТВЕРЖДАЮ

Сторона ЗАКАЗЧИКА

Сторона ИСПОЛНИТЕЛЯ

Доцент кафедры  
ИАНИ ННГУ, к.т.н.

Профессор кафедры  
ИАНИ ННГУ, д.т.н.

\_\_\_\_\_  
С.М. Липкин  
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

\_\_\_\_\_  
Н.В. Старостин  
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

## **СИСТЕМА АНАЛИЗА ДЕФЕКТОВ ПОВЕРХНОСТИ ТРУБОПРОВОДА С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ**

**Руководство системного программиста**

**Этап 2. Разработка ПО «Око Эйлера» и программной документации**

**НИР «Разработка и реализация программного обеспечения анализа дефектов  
поверхности трубопровода с использованием методов машинного обучения»**

**(Шифр ПО «Око Эйлера»)**

Ответственные исполнители

\_\_\_\_\_ А. Арутюнян

\_\_\_\_\_ А.И. Седова

\_\_\_\_\_ Ю.А. Скуридин

\_\_\_\_\_ Д.И. Турков

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

2021 г.

## АННОТАЦИЯ

Руководство системного программиста представляет собой информацию по установке и настройке программы анализа дефектов поверхности трубопровода (ПО «Око Эйлера»).

Руководство предназначено для системных программистов, отвечающих за функционирование программы.

## **СОДЕРЖАНИЕ**

АННОТАЦИЯ	2
ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ	4
СТРУКТУРА ПРОГРАММЫ	5
НАСТРОЙКА ПРОГРАММЫ	6
ПРОВЕРКА ПРОГРАММЫ	7
СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ	8

## 1. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

### 1.1 Назначение программы

Система анализа дефектов поверхности трубопровода (ПО «Око Эйлера») предназначена для анализа внутренней поверхности труб нефтепровода.

### 1.2 Функции программы

ПО «Око Эйлера» выполняет следующие функции:

- осуществляет чтение исходных данных в виде графического файла формата .png;
- осуществляет проверку исходных данных на соответствие формату;
- строит маску обнаруженных дефектов;
- предоставляет возможность пользователю ввести маску дефектов для своего изображения;

### 1.3 Обеспечивающие технические и программные средства

Для функционирования ПО «Око Эйлера» необходимы следующие технические и программные средства на стороне сервера:

- ЭВМ на базе процессора с PR-рейтингом не менее 2000, оперативной памятью не менее 4ГБ, доступной дисковой памятью в размере не менее 100ГБ, подключение к сети Интернет с пропускной способностью не меньше 100 Мбит/сек;
- программное обеспечение – операционная система Windows 10, CUDA, CUDnn, Python версии не ниже 3.7 с установленными модулями Flask, NumPy, Pandas, Pillow, PyTorch, Torchvision;

для клиентской стороны:

- ЭВМ на базе процессора с PR-рейтингом не менее 2000, оперативной памятью не менее 4ГБ, доступной дисковой памятью в размере не менее 32ГБ, монитором, клавиатурой и мышью; (ПК или мобильное устройство, подключённое к сети Интернет; браузер, поддерживающий HTML5)
- операционная система Windows 10 с установленным браузером Internet Explorer версии 10.0.9200.16521 и выше (или аналогичными версиями других браузеров)

## 2. СТРУКТУРА ПРОГРАММЫ

app.py	Подсистема взаимодействия с пользователем
dict.pth Data.py Simple_Unet.py Training_functions.py Training_script.py benchmark.py	Подсистема сегментации
Dataset generator/dataset_generator.py	Подсистема генерации обучающих данных

### 3. НАСТРОЙКА ПРОГРАММЫ

#### 3.1 Установка зависимостей

Перед установкой программы на сервер необходимо установить программное обеспечение, описанное в пункте 1.3 настоящего руководства, в соответствии с их инструкциями по установке:

- Установить python версии не менее 3.7 с офф. сайта <https://www.python.org/downloads/>
- Убедиться, что расположение интерпретатора python добавлено в системные пути.
- Создать папку для репозитория defl, полный адрес установки может состоять из символов латиницы и знака “\_”.
- Зайти в терминал (командную строку)
- `cd defl`
- Установить модули:  

```
python -m venv .venv
python -m pip install --upgrade pip
python -m pip install numpy
python -m pip install torch
python -m pip install pandas
python -m pip install torchvision
python -m pip install Pillow
```

#### 3.2 Установка программы

Для установки программы необходимо распаковать архив с файлами из <https://github.com/EulerDL/defl.git> в выбранную директорию.

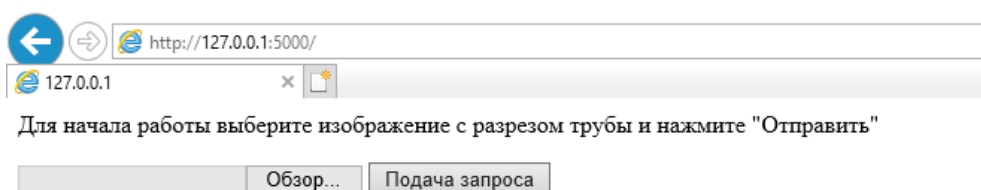
#### 3.3 Запуск сервера

- `./venv/scripts/activate` - активация виртуального окружения
- `set FLASK_APP=app.py` - убеждаемся, что app.py установлен в качестве исполняемого файла в окружении
- `flask run` - запуск приложения

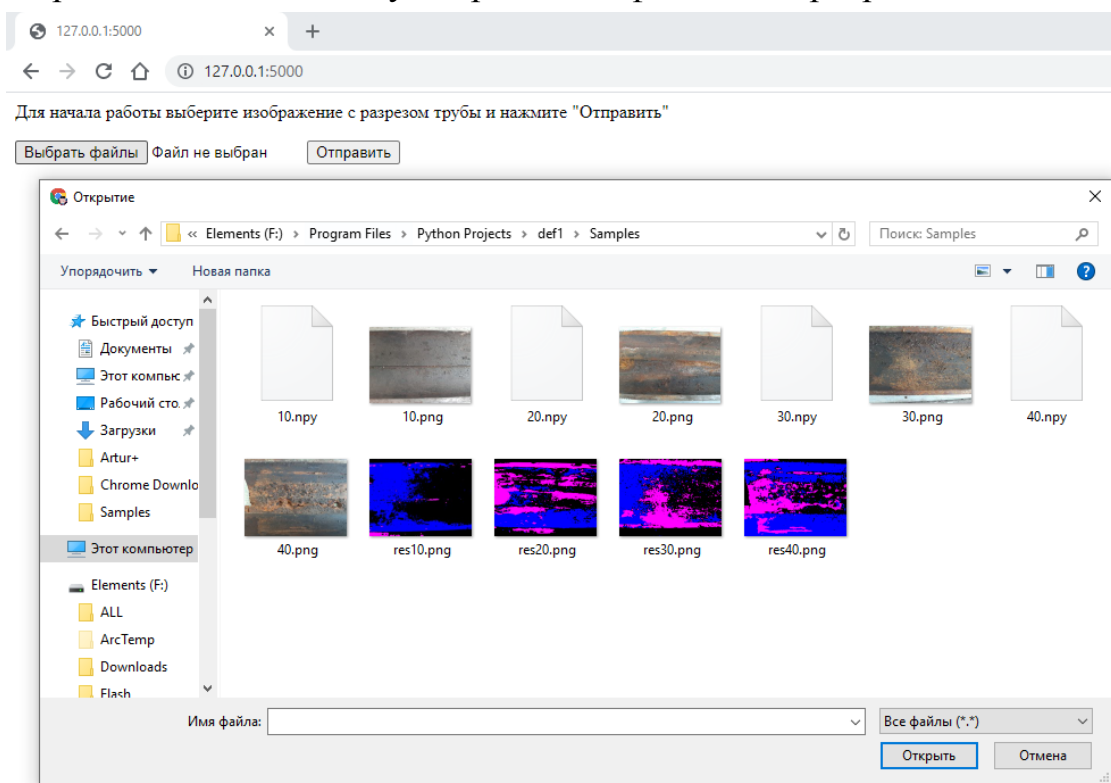
```
F:\Program Files\Python Projects\defl>.venv\Scripts\activate
(.venv) F:\Program Files\Python Projects\defl>set FLASK_APP=app.py
(.venv) F:\Program Files\Python Projects\defl>flask run
* Serving Flask app "app.py"
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

#### 4. ПРОВЕРКА ПРОГРАММЫ

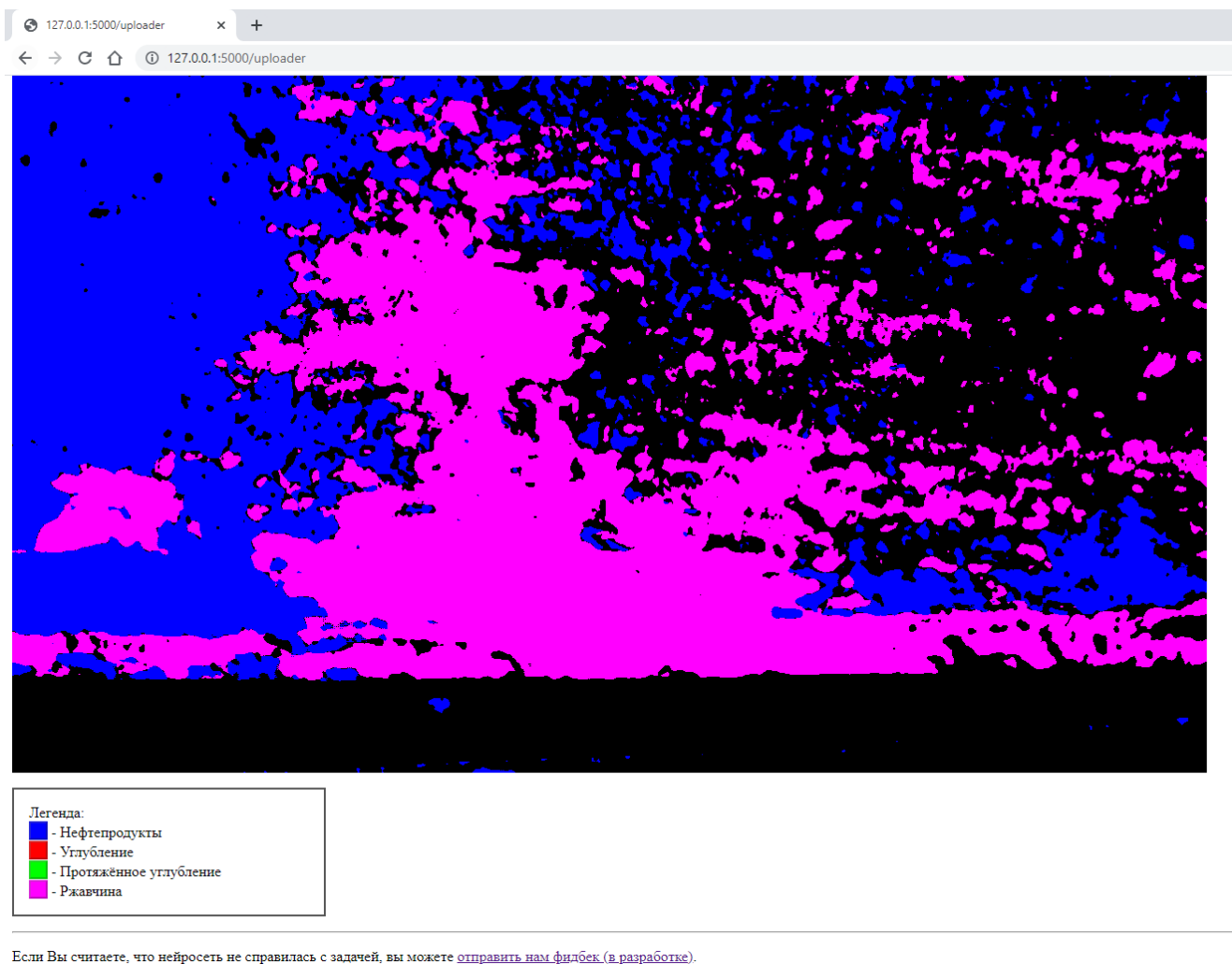
Для проверки работоспособности программы необходимо убедиться в успешном запуске сервера на хосте и возможности подключиться к серверу. Необходимо запустить Internet Explorer и ввести в адресную строку адрес хоста (например, `http://127.0.0.1:5000/`).



На открывшейся веб-странице нажать на кнопку выбора файла, в открывшемся окне проводника найти и выбрать тестовое изображение, после чего на веб-странице нажать кнопку отправления файла на сервер.



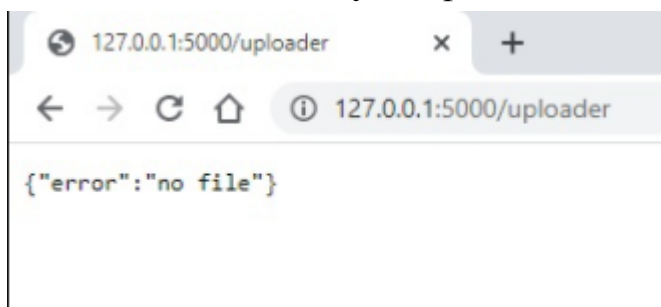
На странице `host/uploader` будет изображён результат работы нейросети.



## 5. СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

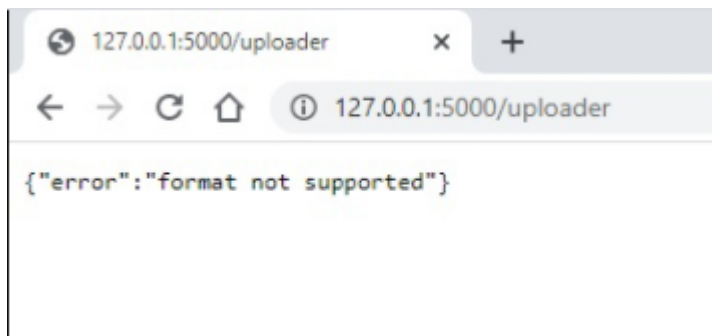
На веб-странице отображаются ошибки при некорректном вводе изображения или некорректной работе приложения:

- 'error': 'no file' - не получен файл на ввод

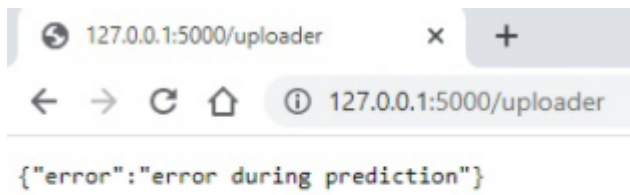


- 'error': 'format not supported' - формат входного файла не соответствует PNG.





- 'error': 'error during prediction' - проблема работы приложения.



В терминале сервера отображаются сообщения:

- “GET / HTTP/1.1” 200 - Изображение принято сервером;
- “POST /uploader HTTP/1.1” 200 - Изображение отправлено сервером.