ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ
«ГРАНИТ-К»
(ООО НПП «ГРАНИТ-К»)

АВТОМАТИЧЕСКИЙ АНАЛИЗАТОР ТОМОГРАММ ИЗДЕЛИЙ ИЗ ГЕТЕРОГЕННЫХ МАТЕРИАЛОВ (АОС НК)

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Версия 2.2 от «15» сентября 2026 г.

Москва, 2026

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ 3
2. УСТАНОВКА ПО 4
3. ОБРАБОТКА ДВУМЕРНЫХ ИЗОБРАЖЕНИЙ 7
4. ОБРАБОТКА ТРЕХМЕРНЫХ ТОМОГРАММ 9
5. УСТРАНЕНИЕ НЕПОЛАДОК 11
6. УДАЛЕНИЕ АОС НК 12

1. ВВЕДЕНИЕ

Автоматический анализатор томограмм изделий из гетерогенных материалов (АОС НК) предназначен для автоматизированной обработки двух- и трехмерных томограмм объектов контроля, автоматизированного поиска дефектов на двух- и трехмерных томограммах объектов контроля, представленных в виде изображений в растровом формате, составления трехмерной модели образца с учетом найденных дефектов и экспорта полученной модели в формате, совместимом с программами конечноэлементного анализа.

2. УСТАНОВКА ПО

2.1 Требуется ЭВМ с операционной системой семейства GNU/Linux, с предварительно установленной системой контейнеризации *docker*, и пользователь, обладающий правами администратора. Для установки системы контейнеризации *docker* рекомендуется воспользоваться документацией на дистрибутив или непосредственно документацией системы контейнеризации *docker*, доступной по адресу https://docs.docker.com/engine/install/.

Для систем семейства Ubuntu можно выполнить команду *sudo apt install docker.io*, однако версия docker, поставляемого в комплекте с операционной системой может быть устаревшей.

Рекомендуется добавить пользователя в группу *docker* для более безопасной и комфортной работы (смотрите документацию для своего дистрибутива операционной системы): *gpasswd -a PLACE_USERNAME_HERE* docker

Для корректной работы АОС НК требуется не менее 6 Гб дискового пространства в директории */var/lib/docker*

2.2 Распакуйте скачанный архив с программой в домашнюю директорию. Будет получено два файла – *aos-nk-server-0.1.tar.gz* и *run.sh*. Первый файл является контейнером *docker*, второй – скриптом для запуска контейнера.



Рисунок 2.1 – Контейнер и скрипт запуска

2.3 Все дальнейшие действия необходимо выполнять в командном интерпретаторе *bash*, поставляемом в комплекте стандартных программ системы. Создайте директории для входных и выходных данных АОС НК.

```
$ mkdir /input /output
```

Рисунок 2.2 – Создание директории

2.4 В файле *run.sh* измените значения переменных *INPUT* и *OUTPUT* и задайте им абсолютные пути до созданных директорий. При необходимости измените зачение *TASKS_LIMIT*, определяющее число параллельно обрабатываемых заданий.

```
INPUT=/input/
OUTPUT=/output/

TASKS_LIMIT=16
SCAN_INTERVAL=60

docker run -d \
--name aos-nk-server \
--restart=unless-stopped \
-e SCAN_INTERVAL=$SCAN_INTERVAL \
-e TASKS_LIMIT=$TASKS_LIMIT \
-e USE_CUDA=0 \
-v $INPUT:/input/ \
-v $OUTPUT:/output/ \
aos-nk-server:v0.1
```

Рисунок 2.3 - Переменные INPUT и OUTPUT

2.5 Импортируйте контейнер. Для этого выполните команду *docker load < aos-nk-server-0.1.tar.gz*

2.6 Выполните скрипт *run.sh*, выполнив команду *./run.sh*. Как подтверждение запуска будет выведен присвоенный контейнеру код. Пример кода показан на рисунке ниже:

```
63f8b4b90749b86aa25b081a0488abd839118c4c3b1f9c5c16e441931ef85ce5
```

Рисунок 2.4 – Пример кода контейнера

2.7 Для подтверждения старта программы подождите примерно одну

минуту и выполните команду *docker ps*. В колонке *STATUS* должна быть запись *Up ... seconds* или *Up ... minutes*

```
CONTAINER ID   IMAGE                      COMMAND                  CREATED          STATUS
63f8b4b90749   aos-nk-server:v0.1         "dumb-init python -u…"   43 minutes ago   Up 43 minutes
```

Рисунок 2.5 – Состояние колонки STATUS

2.8 В выводе *docker logs КОД_КОНТЕЙНЕРА* должны быть строки следующего вида

```
Starting...
Scan interval = 60, tasks limit = 8
Starting infinite loop...
```

Рисунок 2.6 – Пример лога

3. ОБРАБОТКА ДВУМЕРНЫХ ИЗОБРАЖЕНИЙ

В директорию для входных файлов поместите директорию с одним изображением. Пример размещения файлов показан на рисунке. В директории *input*, указанной в скрипте *run.sh* размещена директория *5-single*, в которой размещен файл с изображением для анализа.

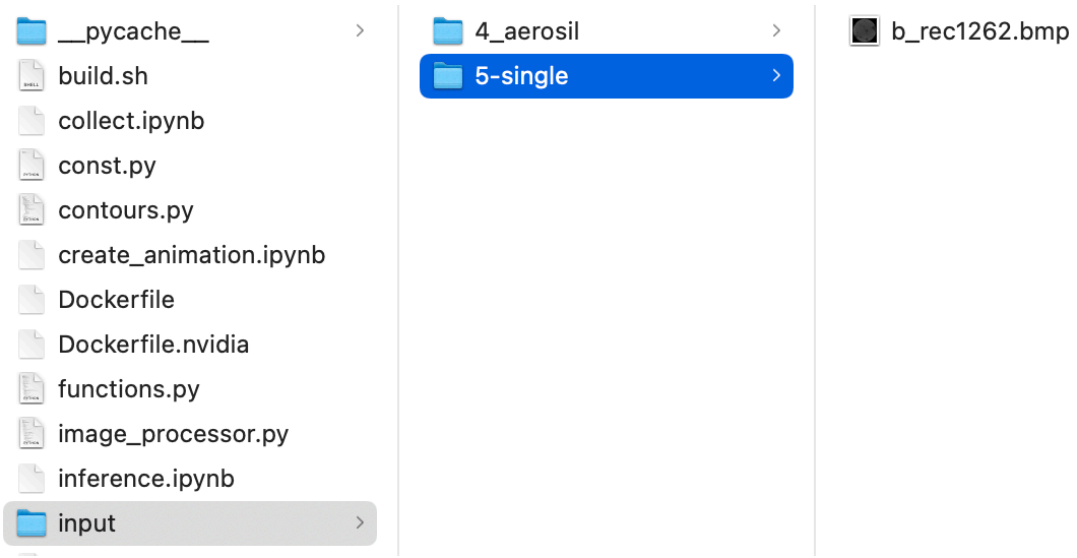


Рисунок 3.1 – Размещение файлов

АОС НК начнет обработку изображения в течение 60 секунд, создаст в директории для выходных данных директорию с названием, соответствующим входной директории, и разместит там файл-изображение с маской дефектов, файл-изображение с выделенными дефектами и таблицу с параметрами дефектов. Пример размещения файлов показан на рисунке. В директории *output*, указанной в скрипте *run.sh* размещена директория *5-single*, в которой размещены файлы с маской (название *b_rec1262.bmp* соответствует входному файлу), выделенными дефектами (contours.bmp) и таблицей с параметрами дефектов (contours.csv).

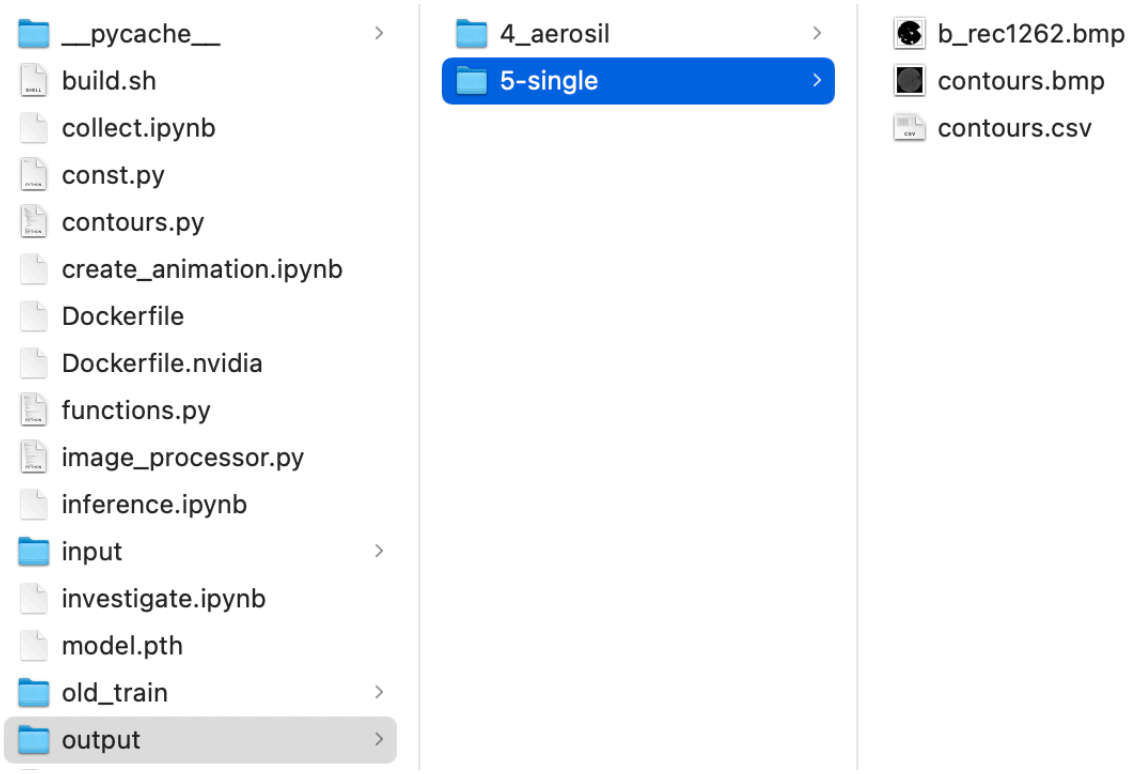


Рисунок 3.2 – Размещение файлов

При выполнении команды *docker logs КОД_КОНТЕЙНЕРА* будут строки следующего вида.

```
Found new tasks: ['5-single']
Started process_single for /input/5-single
Downloading: "https://dl.fbaipublicfiles.com/semiweaksupervision/model_files/semi_v
100%|██████████| 97.7M/97.7M [00:05<00:00, 17.7MB/s]
Finished process_single for /input/5-single
```

Рисунок 3.3 – Выполнение команды docker logs КОД_КОНТЕЙНЕРА

4. ОБРАБОТКА ТРЕХМЕРНЫХ ТОМОГРАММ

В директорию для входных файлов поместите директорию с изображениями, входящими в томограмму. Пример размещения файлов показан на рисунке. В директории *input*, указанной в скрипте *run.sh* размещена директория *4_aerosil*, в которой размещены файлы с изображениями для анализа.

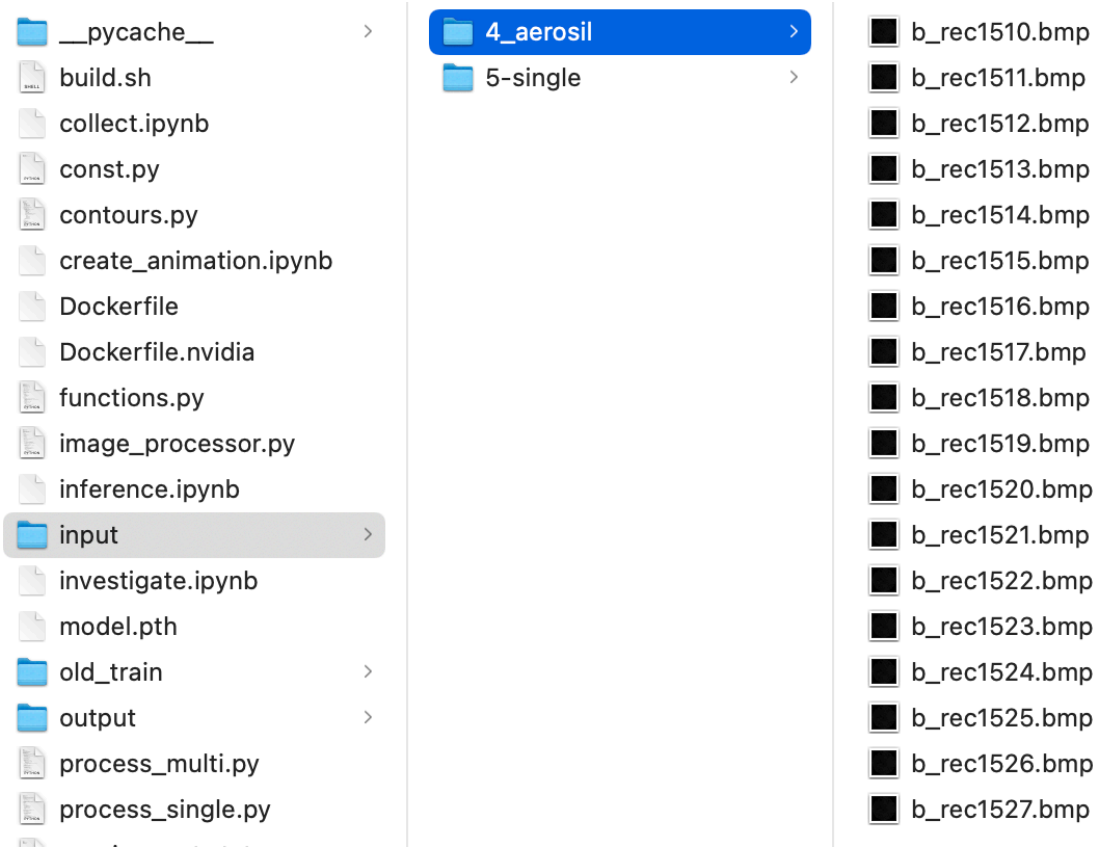


Рисунок 4.1 – Пример размещения файлов

АОС НК начнет обработку изображения в течение 60 секунд, создаст в директории для выходных данных директорию с названием, соответствующим входной директории, и разместит там файлы-изображения с масками дефектов, файл-модель в формате *stl* и текстовый файл с записью об объемной доле дефектов в образце.

Пример размещения файлов показан на рисунке. В директории *output*, указанной в скрипте *run.sh* размещена директория *4_aerosil*, в которой

размещены файлы с масками, информацией о содержании дефектов и моделью в формате *stl*.

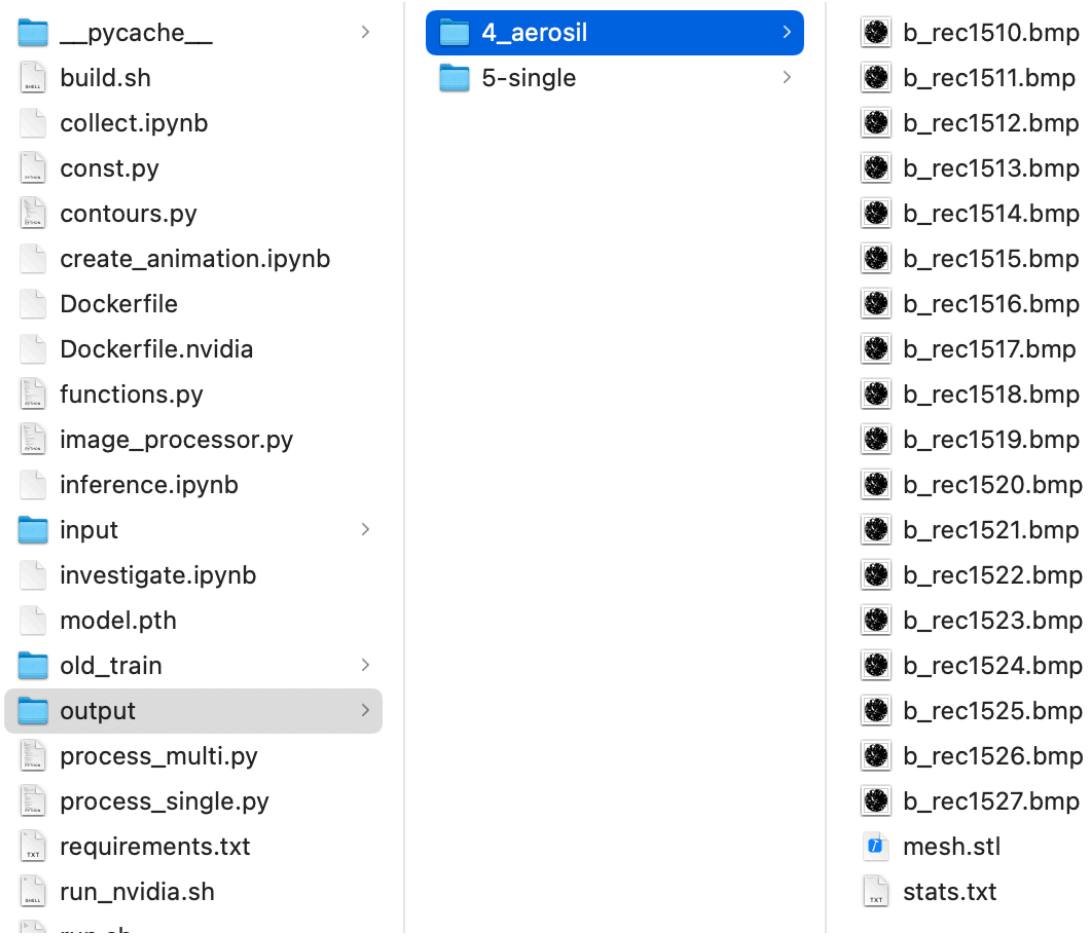


Рисунок 4.2 – Директория output

При выполнении команды *docker logs КОД_КОНТЕЙНЕРА* будут строки следующего вида.

```
Found new tasks: ['4_aerosil']
Started process_multi for /input/4_aerosil
/input/4_aerosil: all images are uploaded
Creating 3d model for /input/4_aerosil
Creating STL...
Finding exterior voxels: 100%|██████████| 18/18 [10:14<00:00, 34.14s/it]
Meshing: 100%|██████████| 22878815/22878815 [15:32<00:00, 24527.62it/s]s]
Exporting mesh to //output/4_aerosil/mesh.stl
Finished process_multi for /input/4_aerosil
```

Рисунок 4.3 – Выполнение команды *docker logs КОД_КОНТЕЙНЕРА*

5. УСТРАНЕНИЕ НЕПОЛАДОК

Для анализа работы программы выполните команду docker logs КОД_КОНТЕЙНЕРА. Примеры выполнения команды и текста ошибок показаны на рисунке.

```
(base) aruslantsev@gpu3z1:~/ndt/v2_ml_2d$ docker logs 66a
Traceback (most recent call last):
  File "/app/image_processor.py", line 9, in <module>
    from process_multi import process_multi
  File "/app/process_multi.py", line 12, in <module>
    from contours import find_contours
  File "/app/contours.py", line 5, in <module>
    import pandas as pd
ModuleNotFoundError: No module named 'pandas'
Traceback (most recent call last):
  File "/app/image_processor.py", line 9, in <module>
    from process_multi import process_multi
  File "/app/process_multi.py", line 12, in <module>
    from contours import find_contours
  File "/app/contours.py", line 5, in <module>
    import pandas as pd
ModuleNotFoundError: No module named 'pandas'
```

Рисунок 5.1 - Примеры выполнения команды и текста ошибок

В случае любой ошибки рекомендуется:

1. Выполнить перезапуск АОС НК командой *docker restart aos-nk-server;*
2. Проверить, что указанные директории для входных и выходных данных существуют и доступны для записи;
3. Проверить, что входные изображения возможно открыть и просмотреть;
4. Проверить, что достаточно дискового пространства для выходных файлов;
5. В случае повторного возникновения ошибки связаться с администратором.

6. УДАЛЕНИЕ АОС НК

Выполните следующую последовательность команд:

1. docker stop aos-nk-server
2. docker rm aos-nk-server
3. docker system prune -a -f