SimCol-to-3D 2022 - 3D Reconstruction During Colonoscopy Challenge

Part of the Endoscopic Vision Challenge – MICCAI2022

SimCol-to-3D 2022 challenge details are available on the challenge website:

https://www.synapse.org/#!Synapse:syn28548633/wiki/

For any query, write to us: endovis-weiss-vision@ucl.ac.uk

Submission Guidelines – Test Phase

This document covers the technical details on preparing and submitting the docker submissions for the three sub-tasks of the Sim-to-3D 2022 challenge. Please also visit Sim-to-3D - 3D Reconstruction During Colonoscopy - syn28548633 - Wiki (synapse.org) and follow the writeup instructions for making a complete and valid submission.

Test data (without groundtruth) will be provided, and participants must submit their results as .zip file. A docker must also be submitted so that the organizers can validate the method.

The participants must create a valid docker file, following the guidelines in this document and upload it to the synapse platform.

Using the Synapse platform, the participants must submit the following files:

1. Docker image for Task 1
2. Docker image for Task 2
3. Docker image for Task 3
4. Prediction on test data as a single zip file
5. Report in LNCS format (no more than 4 pages) as a pdf.

Instructions on how to prepare your submission and make the submission are provided in this document.

1. Submission Docker Image Format

The docker image must read data from **/<patient folder>/<trajectory folder>** and output the resulting depth mask (for task 1) as .npy into **/<patient folder>/<trajectory folder>_OP/depth** or camera pose as (for task 2&3) as text file into **/<patient folder>/<trajectory folder>_OP/pose**

Note: Use the provided docker template for each task for integrating your code with the correct (required) output format. SimCol github link: https://github.com/anitarau/simcol

Follow the instructions in 'Testing docker image locally' to verify that your docker image is functioning as desired before submitting it on the Synapse.
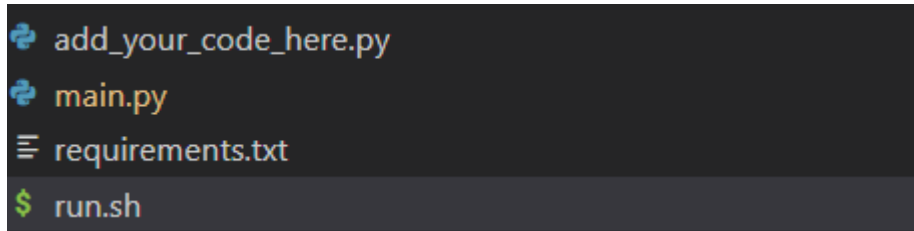
*Docker containers **must** contain an execution script called **run.sh** that takes the input path and output path as arguments. **Please note that any container entry point will be ignored**.*

```
#!/bin/bash

# $1 is the input path
# $2 is the output path

python main.py $1 $2
```

```
add_your_code_here.py
main.py
requirements.txt
run.sh
```

1.1.   Task 1 (Depth Estimation) Submission

For Task 1, the submitted docker will be used for testing in the following way:

> The submitted docker will scan all images from each subject's trajectory folder (e.g.
> /SyntheticColon_I_Test/Frames_S5) with the '.png' extension.
> The submitted docker will process each *.png image and output a depth prediction in *.npy float16 format.
> The submitted docker will store the output in /SyntheticColon_I_Test/Frames_S5_OP/depth
> folder.
> For example,

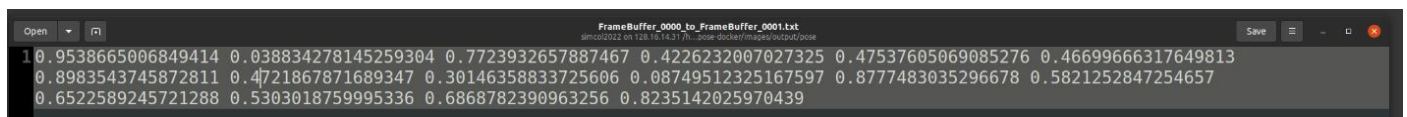> > **INPUT:** /SyntheticColon_I_Test/Frames_S5/FrameBuffer_000.png   …

> > **OUTPUT:** /SyntheticColon_I_Test/Frames_S5_OP/depth/FrameBuffer_000.npy
> > …

1.2.   Task 2 (Camera Pose Estimation) Submission

For Task 2, the submitted docker will be used for testing in the following way:

- The submitted docker will scan all images from each subject's trajectory folder (e.g.
  /SyntheticColon_I_Test/Frames_S5) with the '.png' extension and ensure they are in ascending order.
- The submitted docker will process two consecutive frames *(k, k+1)* and output 1 x 16 flatten of the 4x4 relative camera pose [R t; 0 0 0 1], where R is a 3x3 rotation matrix, and t is a 3x1 translation vector between the two consecutive frames *(k, k+1)*.
- The submitted docker will store the output transformation as a txt file in
  /SyntheticColon_I_Test/Frames_S5_OP/pose folder.

```
FrameBuffer_0000_to_FrameBuffer_0001.txt
simcol2022 on 128.16.14.317h...pose-docker/images/output/pose
1 0.9538665006849414 0.038834278145259304 0.7723932657887467 0.4226232007027325 0.47537605069085276 0.46699666317649813
  0.8983543745872811 0.4721867871689347 0.30146358833725606 0.08749512325167597 0.8777483035296678 0.5821252847254657
  0.6522589245721288 0.5303018759995336 0.6868782390963256 0.8235142025970439
```

- The output txt file will have the name <frame k>_to_<frame k+1> (e.g.
  FrameBuffer_0000_to_FrameBuffer_0001) with a txt extension.

> For example,

> > **INPUT:**
> > /SyntheticColon_I_Test/Frames_S5/FrameBuffer_000.png
> > /SyntheticColon_I_Test/Frames_S5/FrameBuffer_001.png
> > /SyntheticColon_I_Test/Frames_S5/FrameBuffer_003.png
> > …

OUTPUT:

```
/SyntheticColon_I_Test/Frames_S5_OP/Pose/FrameBuffer_0000_to_FrameBuffer_0001.txt
/SyntheticColon_I_Test/Frames_S5_OP/Pose/FrameBuffer_0001_to_FrameBuffer_0002.txt
…
```

1.3. Task 3 (Generalization to Real Data) Submission

Like Section 1.2, with the output folder named as `/SyntheticColon_I_Test/Frames_S5_OP/pose_t3`

1.4. Docker image format – Dummy Examples for Task 1, Task 2 and Task 3

To facilitate the submission, we have provided dummy docker images for both Task 1, Task 2 and Task 3, which shows the path to set for the input and output folders and the format of the output depth ('npy for Task 1) and 4x4 relative pose flatten to 1D (1x16 vector) ('txt' for Task 2).

Visit the EndoVis-Sim-to-3D 2022 GitHub page (below mentioned link) for accessing these docker dummy examples

https://github.com/anitarau/simcol

This GitHub repository also contains scripts for visualising the output.

1.5. Submission instructions

Each team/participant needs to fulfil the following submission criteria to qualify in the challenge leaderboard.

- A **docker container** for each subtask uploaded to Synapse using Synapse project. (**Complete docker submission instructions are provided in this document – Sec. 3**)
- **Prediction on test data** should be submitted as a single zip file. **Visit Sec. 4 for details on how to submit**.
- **Writeup of the method** used (for each task), submitted as a pdf file uploaded to the Team Synapse project and submitted to the SimCol3D Challenge. **Visit Sec. 5 for details on how to submit**.
- **Pre-recorded short presentation**: MICCAI2022 and Endovis2022 challenge will be hybrid. Each team/participant must submit a pre-recorded video presentation of their methods. One member of the participating team should attend the challenge and present their method in person. Further details on the duration of the talk will be published closer to the challenge day.

1.6. Becoming a certificate user

<mark>**IMPORTANT**: To use all docker functionality on Synapse, you must be a certified user. You can become a certified user by filling out the following quiz:</mark>

https://www.synapse.org/#!Quiz:

2. Building and testing docker images

2.1. Setup docker

Docker is required for submissions: https://www.docker.com/get-started/

The base docker image can be chosen freely. It is suggested to use one of the following:

Pytorch: https://hub.docker.com/r/pytorch/pytorch/

Tensorflow: https://www.tensorflow.org/install/docker

For GPU support the NVIDIA Container Toolkit may need to be installed: https://github.com/NVIDIA/nvidia-docker

2.2.   Building docker images

Download the dummy from our GitHub: https://github.com/anitarau/simcol.

Below is a walk-through example for Task 1 docker submission. Follow the same steps (with Task 2 or Task 3 tags) to make submissions for Task 2 and Task 3.

Integrate your code within the dummy docker folder (SimCol3D-task1-depth-docker) by adding your code to 'add_your_code_here.py' file.

Ensure that you are in the top-level directory of the project where the Dockerfile is located.

Build the image using the command:

```
docker build -t <image_name> .
```

```
SimCol3D-task1-depth-docker$ docker build -t simcol2022_task1_ymtest:v1 .
```

Important note: Use the format simcol2022_<taskname>_<teamname>:<version>

where,

> <taskname> is either *task1*, *task2 or task3* depending on the task you are submitting to
> <teamname> is the acronym of your team that you specified at the time of team registration. In the above example, it is *ymtest*
> <version> is the version number for docker.

2.3.   Testing docker image locally

To test the docker image locally, you will need a directory with example input images and an empty directory to output resulting image/txt files.

You can test your image using the following command

```
docker run --gpus all -it --rm -v "$PWD/SyntheticColon_I_Test:/data"
<image_name> /data/Frames_S5 /data/Frames_S5_OP
```

```
[simcol2022@Dan-DGX-Station:~/Code/ymjin/SimCol3D-task1-depth-docker$ docker run --gpus all -it --rm -v
"$PWD/SyntheticColon_I_Test:/data" <image_name> /data/Frames_S5 /data/Frames_S5_OP
```

3. Docker submission on the Synapse platform

3.1.   Create a new project on Synapse

In order to submit a docker container, you first need to create a new project on the Synapse platform with the challenge name and your team:

SimCol2022_<teamname>

Create a New Project   ✕

Project Name
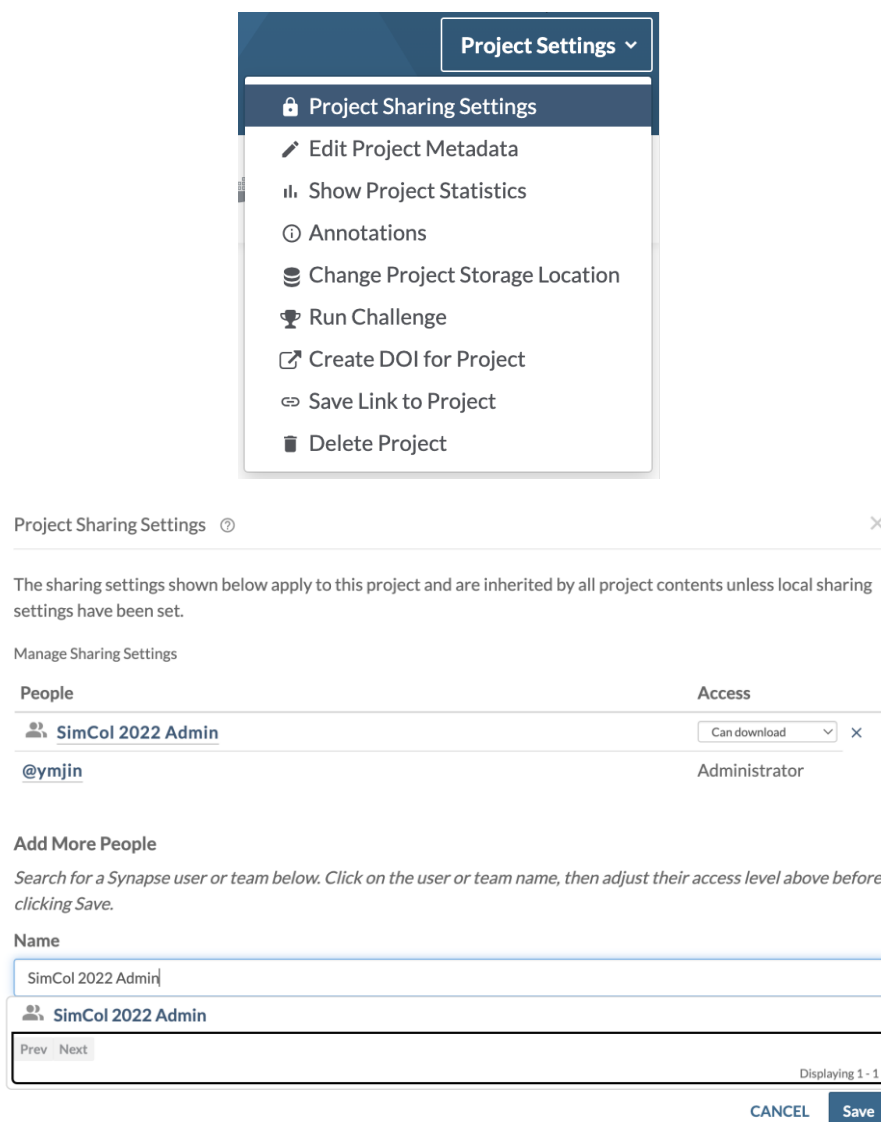
SimCol2022_YMTest|

Cancel   **OK**

Note: YMTest is the team name in these instructions.

Once the project is created,

- Note its Synapse ID (e.g., for this example the Synapse ID is syn34226362)
- Give download permission of the project to the SimCol organising team (SimCol 2022 Admin) by following the below mentioned screenshots.

3.2.   Login to Synapse with docker

```
docker login -u <username> docker.synapse.org
```

(Enter Synapse password)

```
simcol2022@Dan-DGX-Station:~/Code/ymjin/SimCol3D-task1-depth-docker$ docker login -u ymjin docker.synapse.org
Password:
WARNING! Your password will be stored unencrypted in /home/simcol2022/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

3.3.   Tag the submission docker image

The synapse ID of the project created in Step 4.1 can be used to upload.

```
docker tag <imagename_local> docker.synapse.org/<Synapse_project_ID>/<imagename_synapse>
```

```
/SimCol3D-task1-depth-docker$ docker tag simcol2022_task1_ymtest:v1 docker.synapse.org/syn34226362/simcol2022_task1_ymtest:v1
/SimCol3D-task1-depth-docker$ ▮
```

3.4. Push the docker image to Synapse

```
docker push docker.synapse.org/<Synapse ID>/<imagename_synapse>
```

```
[simcol2022@Dan-DGX-Station:~/Code/ymjin/SimCol3D-task1-depth-docker$ docker push docker.synapse.org/syn34226362/simcol2022_task1_ymtest:v1
The push refers to repository [docker.synapse.org/syn34226362/simcol2022_task1_ymtest]
c1948ba5393a: Pushed
24700f4458c2: Pushed
df30fa2820e1: Mounted from syn34221648/simcol3d_task2_sbtest
f567fecf866f: Mounted from syn34221648/simcol3d_task2_sbtest
df4e089532f6: Mounted from syn34221648/simcol3d_task2_sbtest
5f08512fd434: Mounted from syn34221648/simcol3d_task2_sbtest
c7bb31fc0e08: Mounted from syn34221648/simcol3d_task2_sbtest
50858308da3d: Mounted from syn34221648/simcol3d_task2_sbtest
v1: digest: sha256:6af0e5dad7f529b5f161418cb53a5ee746dda17a2975c0707db22fea4771637b size: 1992
```

3.5. Verify the docker image push

You can verify on the Synapse website if the push was successful. If the download fails, it might be because you are not a <mark>certified user</mark>.

3.6. Submit the docker image to the challenge

Under your project SimCol2022_<team_name>, press the 'Docker' tab

Click on the docker that you wish to submit to the challenge

Then, click on 'Docker Repository Tools' → 'Submit Docker Repository to Challenge'

Select the docker 'Tag' you would like to submit.

Next, select the 'Challenge task' you would like to submit.

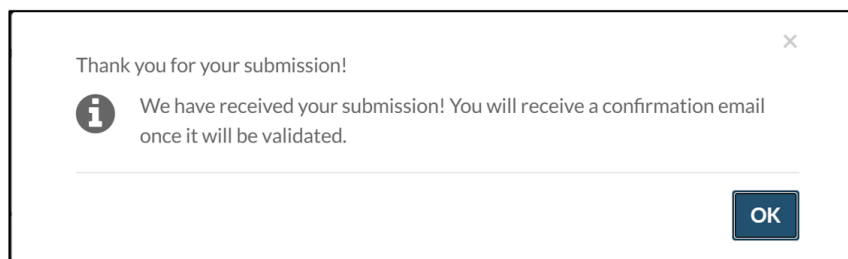Next, select 'I am submitting as an individual'.

Note that team registration is taking place through a separate email. On the Synapse Challenge website, **only the lead participant of the team can make the submission** to avoid voiding the Participation Policy.

You would receive an <u>email within 12 hours</u> confirming if the submission made is valid.

Your submission will appear in the Submissions table for the specific task.
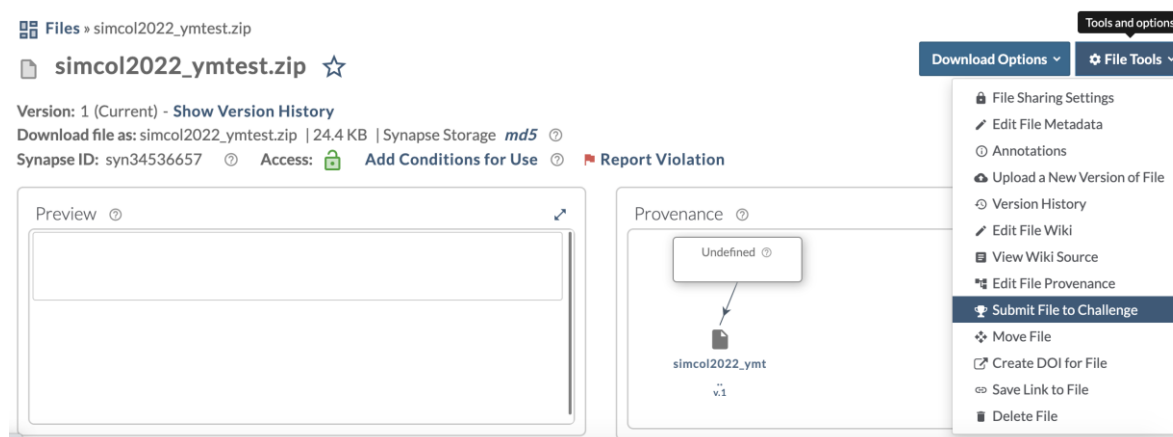

4. Prediction Zip file submission

The Prediction folder must be structured in the tree, shown in [this file](#).

Follow the following steps for submitting the prediction zip file.

1. In the project that you created (make sure you have already created the project by following Sec 3.1 and have given download permission to SimCol 2022 Admin).
2. Upload the zip file under 'Files' tab in your project.
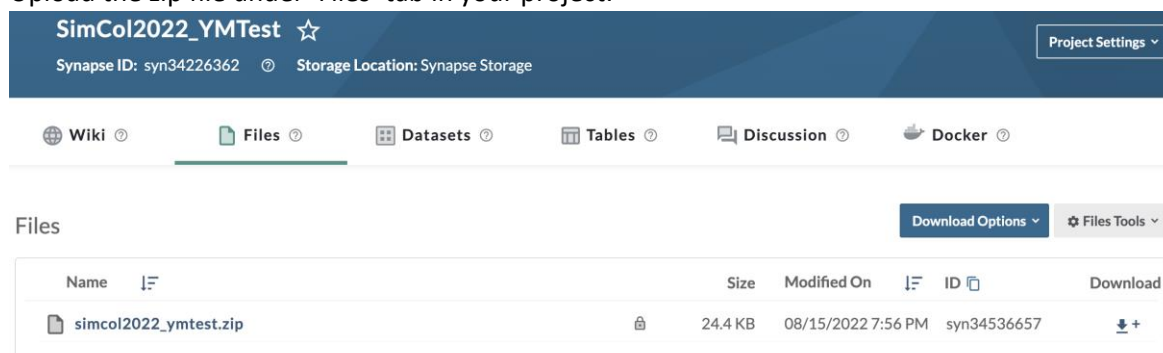
3. Click on the uploaded zip file. Then go to File tools -> Submit File to Challenge

4. Then enter submission name. It must be in the format SimCol2022_<Team name>_Prediction.zip. See sample below, where team name is 'ymtest'. And submit it to 'SimCol3D - Predictions as a single zip file submission'.


5. Writing up submission

- The writeup should be no more than 4 pages in LNCS format.
    o Please visit THIS LINK for the method writeup guidelines.
    o For submitting the writeup, follow the following steps:
1. In the project that you created (make sure you have already created the project by following Sec 3.1 and have given download permission to SimCol 2022 Admin).
2. Upload the writeup file under 'Files' tab in your project.
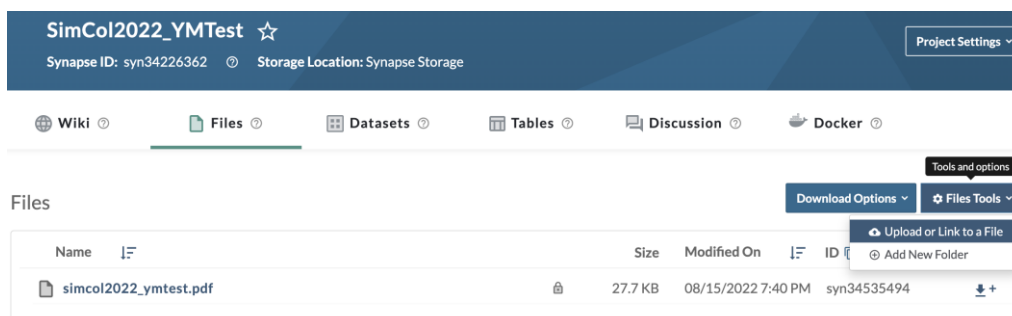
3. Click on the uploaded pdf file. Then go to File tools -> Submit File to Challenge
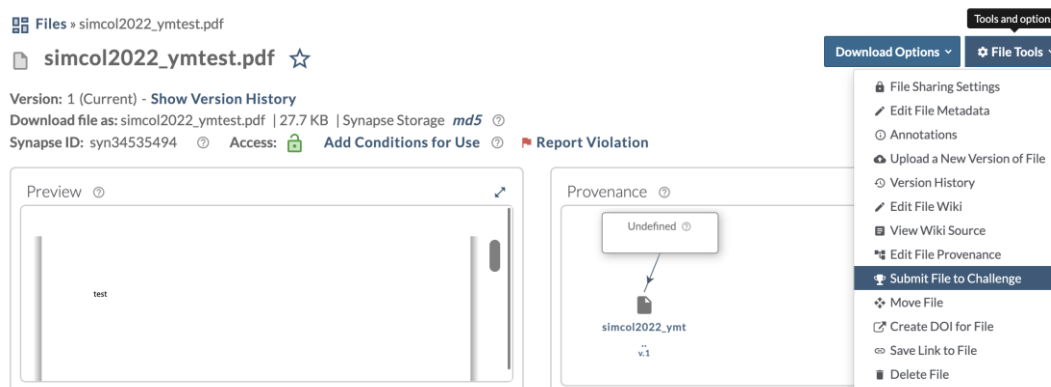
4. Then enter submission name. It must be in the format SimCol2022_<Team name>_report. See sample below, where team name is 'ymtest'. And submit it to 'SimCol3D challenge report submission'.

Submit to Challenge   ✕

Select the challenge(s) below that you would like to submit to

- ○ Eval Q Test   ⊘
- ○ SimCol3D - Predictions as a single zip file submission
- ● SimCol3D challenge report submission
- ○ SimCol3D Task 1 - Depth Prediction (synthetic Data) - docker
- ○ SimCol3D Task 2 - Camera Pose Prediction (Synthetic Data) - docker
- ○ SimCol3D Task 3 - Generalization (Real Data) - docker
- ○ Surgical Workflow and Skill Analysis Subchallenge Docker

Submission   ⊘

- ○ Surgical Workflow and Skill Analysis Subchallenge Final Write-Up

Submission name (optional)

simcol2022_vmtest_report

CANCEL   **Next**

For any query, write to us: endovis-weiss-vision@ucl.ac.uk or post on the EndoVis-SimCol2022 Slack forum.