

Install and configure Docker as shown in the link:

<https://docs.docker.com/desktop/windows/install>

Writing a Dockerfile for our application:

The Dockerfiles contain instructions for building the image. Lines of this file begin with instructions typed in capital letters. After the instructions come their arguments. Instructions, when building an image, are processed from top to bottom.

```
FROM pytorch/pytorch:latest

RUN apt-get -y update
RUN apt-get -y install git

RUN apt-get -y install ffmpeg libsm6 libxext6

RUN pip install -U pip
RUN pip install torchvision

WORKDIR /yolop

RUN git clone https://github.com/hustvl/YOLOP.git
RUN cd YOLOP && pip install -r requirements.txt

WORKDIR YOLOP
```

Let's see what each line means:

FROM pytorch/pytorch:latest

We make it clear to docker that when creating an image, the first thing to do is to download and install pytorch from the Docker repository, and the latest version. (PyTorch is a machine learning framework for the Python language, required for the selected project)

RUN apt-get -y update

This line tells Docker to update the packages from the base image.

RUN apt-get -y install git

RUN apt-get -y install ffmpeg libsm6 libxext6

With these two commands, install the git utility into the image to access the repository and libraries for working with computer vision algorithms, machine learning and image processing.

RUN pip install -U pip

This command updates the Python package manager.

RUN pip install torchvision

This command installs the Torchvision library, which contains model architectures and common image transformations for computer vision.

WORKDIR /yolop

Change the working directory to /yolop, if it did not exist, it is automatically created.

RUN git clone https://github.com/hustvl/YOLOP.git

We clone the repository with the project from github.

RUN cd YOLOP && pip install -r requirements.txt

We go to the directory with the project and install the python dependencies that the author of the project indicated in the *requirements.txt* file.

WORKDIR YOLOP

Change the working directory to the directory with the project.

Open the Windows command prompt and go to the folder with the created Dockerfile.

(It is important that the Dockerfile does not have an extension - windows sometimes adds the extension automatically)

Image assembly.

To do this, in the terminal, being in the root folder of the project (where the Dockerfile is), run the command:

docker build --tag yolop .

docker - calling the docker application.

build - building the image according to the instructions from the Dockerfile.

--tag <name> - the application will be available by the name specified here.

. - indicates that you need to build the project from the current folder.

Next, the assembly of the project begins, upon successful completion of which we will receive ready image with the project.

We create and run the container.

To do this, run the command in the terminal:

```
docker run yolop python tools/demo.py --source inference/images
```

docker - calling the docker application.

run yolop - create and run a container from the yolop image.

```
python tools/demo.py --source inference/images
```

to executing a command inside a container (the command is taken from the project instructions on github).

Let's see the result.

Upon successful completion of the previous paragraph, images processed by machine vision algorithms are created inside the container. To view images need to be copied from the container.

To do this, first find out the **id** of the container with the following command:

```
docker ps -a
```

Copy the CONTAINER ID of the desired container.

Next, execute the following command, where instead of **<CONTAINER ID>** we insert the **id** of the desired container:

```
docker cp <CONTAINER ID>:/yolop/YOLOP/inference .
```

In the same folder where the Dockerfile is located, the inference folder should appear, in which the images folder contains the source images, and the output folder contains the images processed by the algorithm.

To upload our images to the container.

To load images into the container, the command from the previous paragraph is used, only the point of departure and destination are changed in places:

```
docker cp ./images <CONTAINER ID>:/yolop/YOLOP/inference
```

The images folder must contain images with a jpg extension and a resolution of 1280*720px

For example:

https://upload.wikimedia.org/wikipedia/commons/4/4a/One_of_the_sections_of_the_Rublevo-Uspenskoe_highway.jpg

Apply the algorithm to the loaded images.

To execute a command in the created container, you need to run the following command:

```
docker exec <NAME> python tools/demo.py --source inference/images
```

The name of the container <NAME> can be found by the command

```
docker ps -a
```

Useful command:

You can run the container interactively with the following command:

```
docker exec -it <NAME> bash
```

After that, the bash terminal will be available, in which you can view the contents of the container and navigate through directories using the **UNIX** `ls` and `cd` commands.