```
# 利用可能なディストリビューションを確認
wsl --list --online
# 今回はUbuntu-20.04を使用
wsl --install -d Ubuntu-20.04
# 別ウィンドウが起動



# Ubuntuアカウントの設定
Enter new UNIX username: interface
New password: YOURPASS
Retype new password: YOURPASS
# パッケージのアップデート
sudo apt-get update
sudo apt-get upgrade
# GPUをWSL2で認識しているか（図5 ①GPU確認）
nvidia-smi

# +-----------------------------------------------------------------------------+
# | NVIDIA-SMI 525.65       Driver Version: 527.56       CUDA Version: 12.0     |
# |-------------------------------+----------------------+----------------------+
# | GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
# | Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
# |                               |                      |               MIG M. |
# |===============================+======================+======================|
# |   0  NVIDIA GeForce ...   On  | 00000000:01:00.0  On |                  N/A |
# |  0%   48C    P8    17W / 240W |    424MiB /  8192MiB |      8%      Default |
# |                               |                      |                  N/A |
# +-------------------------------+----------------------+----------------------+



sudo vim /etc/wsl.conf
# 下記を記述
#
# [boot]
# systemd=true



# インストールスクリプトの実行
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
sudo systemctl --now enable docker
# Dockerが使用可能か確認
sudo docker run hello-world



リスト1　NVIDIA Container Toolkitのインストール

# パッケージ リポジトリと GPG キーをセットアップ
# （参考）下記リンクにコマンドが記載されています
# https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html
distribution=$(. /etc/os-release;echo $ID$VERSION_ID) ¥
      && curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o
/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg ¥
      && curl -s -L https://nvidia.github.io/libnvidia-container/$distribution/libnvidia-container.list | ¥
            sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg] https://#g' | ¥
            sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list

sudo apt-get update
sudo apt-get install -y nvidia-docker2
sudo systemctl restart docker

# CUDAコンテナを動かし、上記nvidia-smiと同じ結果が得られたら成功（図5 ②GPU確認）
sudo docker run --rm --gpus all nvidia/cuda:11.6.2-base-ubuntu20.04 nvidia-smi



リスト2　/etc/docker/daemon.jsonの修正

{
    "runtimes": {
        "nvidia": {
            "path": "nvidia-container-runtime",
            "runtimeArgs": []
        }
    },
    "default-address-pools": [
            {
```

```
                    "base": "172.30.0.0/16",
                    "size": 24
                }
    ]
}
```

```
# NGCコンテナのイメージをダウンロード
sudo docker pull nvcr.io/nvidia/pytorch:21.04-py3
# NGCコンテナを実行
sudo docker run --gpus all -it --rm nvcr.io/nvidia/pytorch:21.04-py3
```

```
# GPUが有効か確認（図5 ③GPU確認）
python
>>> import torch
>>> print(torch.cuda.is_available())
# True
```

```
sudo docker run --gpus all -it --rm --shm-size=1g -p 8888:8888 -p 8000:8000  nvcr.io/nvidia/pytorch:21.04-py3
```

```
jupyter-lab --port=8888 --ip=0.0.0.0 --allow-root --NotebookApp.token=''
```

```
# 変更前
# url = 'http://images.cocodataset.org/val2017/000000039769.jpg'
# im = Image.open(requests.get(url, stream=True).raw)
# 変更後
im = Image.open("YOUR_IMAGE.JPG")
```

```
git clone https://github.com/jsbroks/coco-annotator.git
cd coco-annotator/
```

```
git clone https://github.com/daedalus-jp/detr_interface.git
cd detr_interface
```

リスト3　学習済みモデルをダウンロードし，それを基にfine-tuningを行う

```
import torch
# Get pretrained weights
checkpoint = torch.hub.load_state_dict_from_url(
            url='https://dl.fbaipublicfiles.com/detr/detr-r50-e632da11.pth',
            map_location='cpu',
            check_hash=True)

# Remove class weights
del checkpoint["model"]["class_embed.weight"]
del checkpoint["model"]["class_embed.bias"]

# Save
torch.save(checkpoint, 'detr-r50.pth')
```

リスト4　学習の実行方法

```
cd detr
python3 main.py --dataset_file custom --coco_path ../data/custom --output_dir ./outputs/ --resume ./detr-r50.pth --num_workers
8 --batch_size 2 --num_classes 1 --epochs 80 --lr 1e-5
```

リスト5　torch.load(*)の引数にcheckpoint.pthのパスを記載する

```
finetuned_model = torch.hub.load('facebookresearch/detr',
                         'detr_resnet50',
                         pretrained=False,
                         num_classes=1)

checkpoint = torch.load('./detr/outputs/checkpoint.pth', map_location='cpu')
finetuned_model.load_state_dict(checkpoint['model'], strict=False)
finetuned_model.eval()
```

リスト6　推論時にしきい値の変更を行うことができる

```
folder_path="./data"
paths=sorted(glob.glob(folder_path+"/test/*.jpg"))

folder_path+="/result"
#folder_path+="/result_without_dataaug"

for path in paths:
    filename=path.split("jpg")[0][:-1]
    input_image = Image.open(filename+".jpg").convert('RGB')
    inference(folder_path,filename,input_image,finetuned_model,labels, threshold=0.6)
```