# DPTのインストール
## （仮想環境の構築）

DPTは，Yolov7の仮想環境を利用します

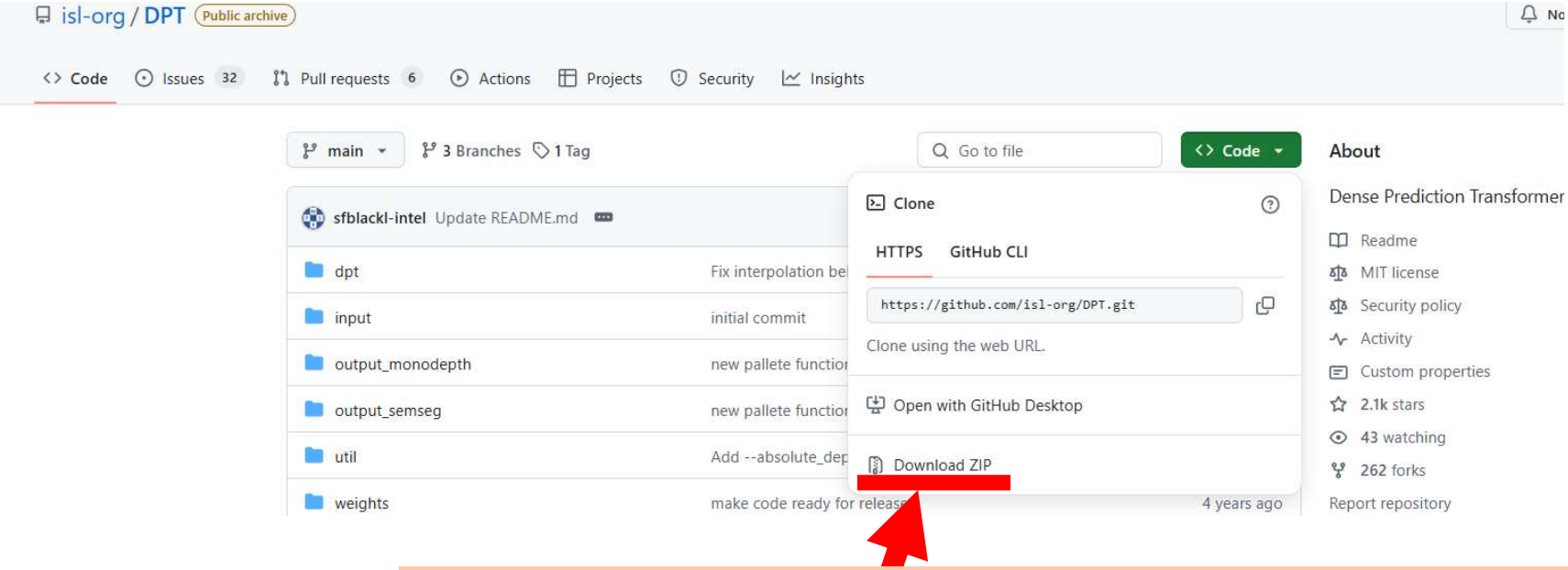
# Yolov7の仮想環境のターミナルを開いてください.

「pip install timm==0.4.5」と打ち込んで実行してください.

C:¥WINDOWS¥system32¥cmd.exe

```
(SadokuYolov7_2) C:¥Users¥bird04>pip install timm==0.4.5
```

DPTのページに（https://github.com/isl-org/DPT）行ってください




ZIPファイル（DPT-main.zip）をダウンロードする

---

```
python run_segmentation.py
```

3. The results are written to the folder `output_monodepth` and `output_semseg`, re

Use the flag `-t` to switch between different models. Possible options are `dpt_hybr`

ページの下に行って，
これをクリックして，
　　モデルファイルをダウンロードする，
(dpt_hybrid_nyu-2ce69ec7.pt)

els:

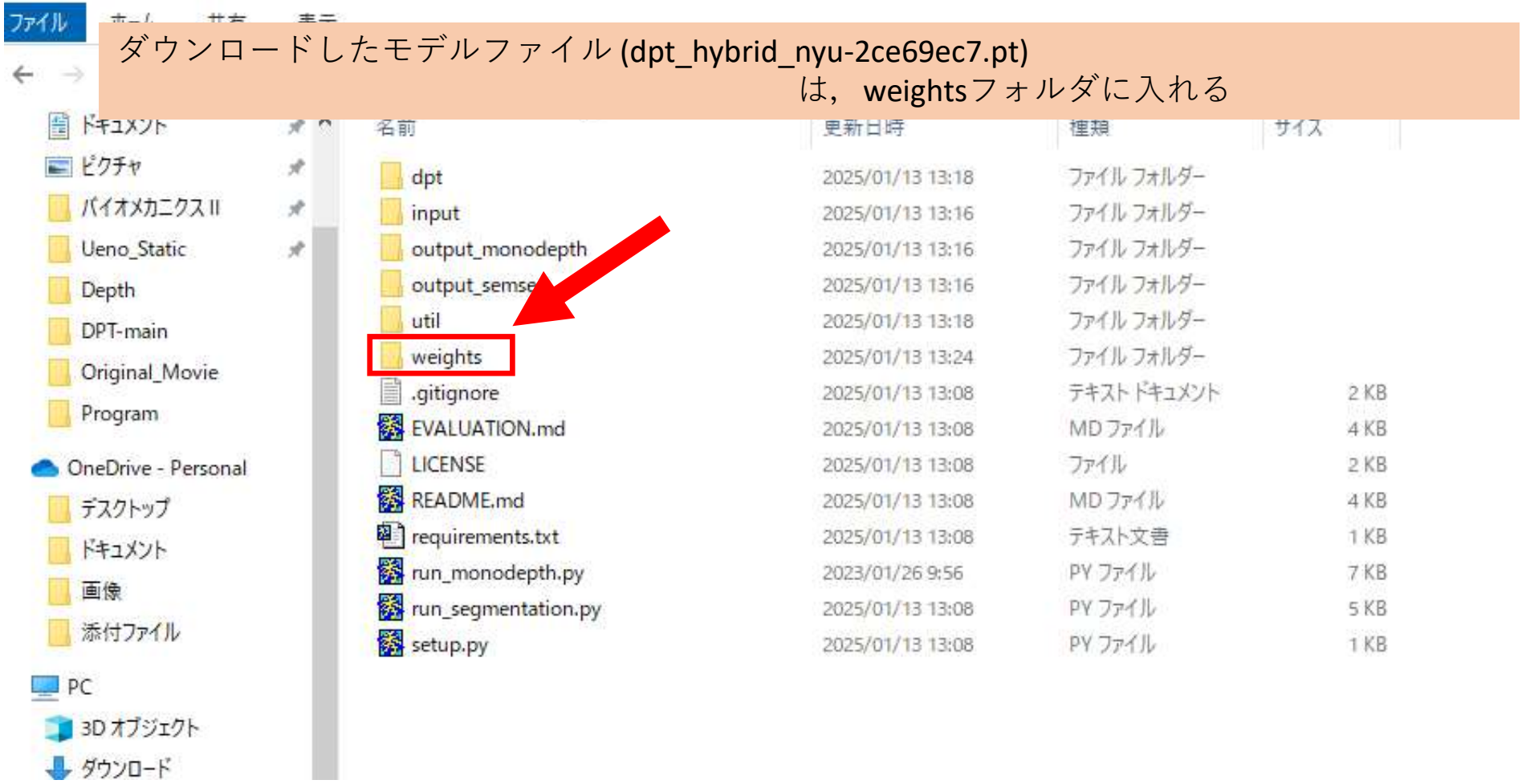
fi ed on KITTI: dpt_hybrid_kitti-cb926ef4.pt Mirror

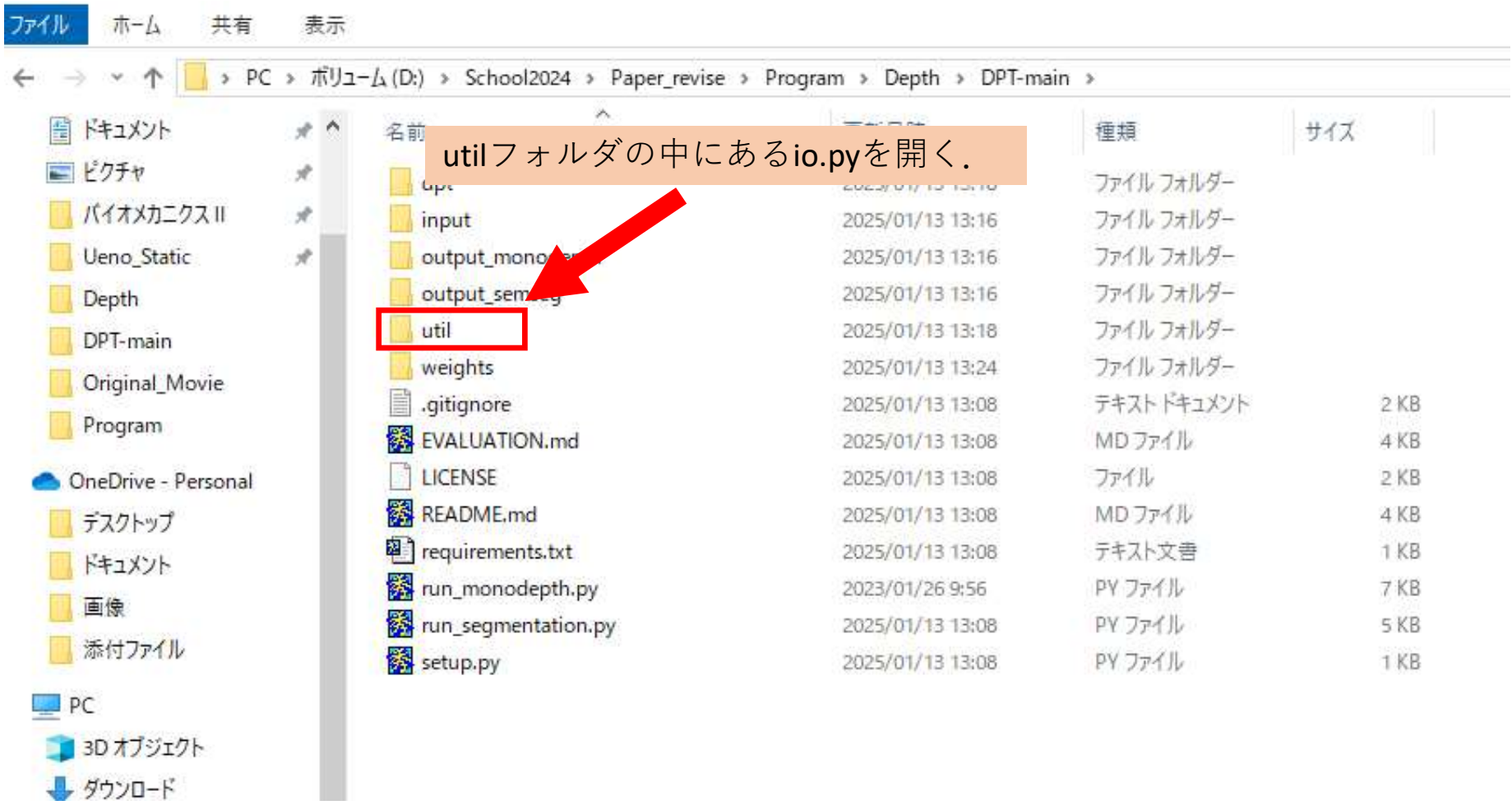
finetuned on v2: dpt_hybrid_nyu-2ce69ec7.pt Mirror

Run with

```
python run_monodepth -t [dpt_hybrid_kitti|dpt_hybrid_nyu]
```

ダウンロードしたモデルファイル (dpt_hybrid_nyu-2ce69ec7.pt)
は， weightsフォルダに入れる




ファイル
ホーム
共有
表示
PC › ボリューム (D:) › School2024 › Paper_revise › Program › Depth › DPT-main ›
ドキュメント
ピクチャ
バイオメカニクスII
Ueno_Static
Depth
DPT-main
Original_Movie
Program
OneDrive - Personal
デスクトップ
ドキュメント
画像
添付ファイル
PC
3D オブジェクト
ダウンロード
名前
種類
サイズ
utilフォルダの中にあるio.pyを開く.
input 2025/01/13 13:16 ファイル フォルダー
output_mono 2025/01/13 13:16 ファイル フォルダー
output_sem 2025/01/13 13:16 ファイル フォルダー
util 2025/01/13 13:18 ファイル フォルダー
weights 2025/01/13 13:24 ファイル フォルダー
.gitignore 2025/01/13 13:08 テキスト ドキュメント 2 KB
EVALUATION.md 2025/01/13 13:08 MD ファイル 4 KB
LICENSE 2025/01/13 13:08 ファイル 2 KB
README.md 2025/01/13 13:08 MD ファイル 4 KB
requirements.txt 2025/01/13 13:08 テキスト文書 1 KB
run_monodepth.py 2023/01/26 9:56 PY ファイル 7 KB
run_segmentation.py 2025/01/13 13:08 PY ファイル 5 KB
setup.py 2025/01/13 13:08 PY ファイル 1 KB

io.pyを①から②に変更する.

①

```
def write_depth(path, depth, bits=1, absolute_depth=False):
    """Write depth map to pfm and png file.

    Args:
        path (str): filepath without extension
        depth (array): depth
    """
    write_pfm(path + ".pfm", depth.astype(np.float32))

    if absolute_depth:
        out = depth
    else:
        depth_min = depth.min()
        depth_max = depth.max()

        max_val = (2 ** (8 * bits)) - 1

        if depth_max - depth_min > np.finfo("float").eps:
            out = max_val * (depth - depth_min) / (depth_max - depth_min)
        else:
            out = np.zeros(depth.shape, dtype=depth.dtype)

    if bits == 1:
        cv2.imwrite(path + ".png", out.astype("uint8"), [cv2.IMWRITE_PNG_COMPRESSION, 0])
    elif bits == 2:
        cv2.imwrite(path + ".png", out.astype("uint16"), [cv2.IMWRITE_PNG_COMPRESSION, 0])

    return


def write_segm_img(path, image, labels, palette="detail", alpha=0.5):
    """Write depth map to pfm and png file.
```

これを実行しないようにする.

②

```
def write_depth(path, depth, bits=1, absolute_depth=False):
    """Write depth map to pfm and png file.

    Args:
        path (str): filepath without extension
        depth (array): depth
    """
#    write_pfm(path + ".pfm", depth.astype(np.float32))

    if absolute_depth:
        out = depth
    else:
        depth_min = depth.min()
        depth_max = depth.max()

        max_val = (2 ** (8 * bits)) - 1

        if depth_max - depth_min > np.finfo("float").eps:
            out = max_val * (depth - depth_min) / (depth_max - depth_min)
        else:
            out = np.zeros(depth.shape, dtype=depth.dtype)

    if bits == 1:
        cv2.imshow( "Result", out.astype("uint8") )
        cv2.waitKey(1)
        cv2.imwrite(path + ".png", out.astype("uint8"), [cv2.IMWRITE_PNG_COMPRESSION, 0])
    elif bits == 2:
        cv2.imshow( "Result", out.astype("uint16")  )
        cv2.waitKey(1)
        cv2.imwrite(path + ".png", out.astype("uint16"), [cv2.IMWRITE_PNG_COMPRESSION, 0])

    return
```

これを書き加える

# DPT-mainフォルダに，DPT-Human.pyを入れる．