DUSS README FILE

1. Setup and activate the environment:
    1. Run CMD: conda env create -f myenv.yml
    2. Run CMD: conda activate EnvName

2. Data preprocessing:
    1. Download Suim dataset (URL:) and place train & val split in one folder. Inside train/val folder keep image in “images” and masks in “mask” folder.
    2. Run CMD: cd utils
    3. Run CMD: python true_annotations.py –data_dir ‘path/to/datatset-directory’ –split ‘train’ –mask_category ‘coarse’
        - python true_annotations.py --data_dir '/home/multimedia/VisionLab/DUSS/data' --split ‘train’ --–mask_category ‘coarse’

3. Segment discovery and saving segment crops:
    1. Run CMD: cd ..
    2. Run CMD: python segment_discovery.py --head 0 --head_plus 0 --device cpu --split "train" --vit_size vitb --patch_size 8 --data_dir “path/to/datatset-directory”
        - python segment_discovery.py --head 0 --head_plus 0 --device cpu --split "train" --vit_size vitb --patch_size 8 --data_dir "./data"
        - Two CSV files and their corresponding CROP datasets will be saved in outputs folder.
        - The CSV/CROP file name with and without containe information of raw segments and processed segmentations, respectively.

4. Segment-wise feature (CLS) extraction from DINOs’ vision transformer:
    1. Download pretrained weight of DINO_ViTb or required model from dino github repo.
    2. Run CMD: python CLS_extractor.py --crop_path ".path/to/crop_directory" --pretrained_weight 'path/to/dinovit_models’/pretrained_weight' --batch_size_per_gpu 8 --arch vit_base patch_size 8 --split train --viz_validSegs “True_for_storing_valid_segment_seperately”
        - python CLS_extractor.py --crop_path "./outputs/crop_dir" --pretrained_weight 'checkpoints/b8_checkpoint.pth' --batch_size_per_gpu 8 --arch vit_base --patch_size 8 --split train --viz_validSegs True
    - NOTE: Alternatively, for segment-wise feature extraction from MoCos’ CNN backbone:
        - Run CMD: python feat_extractore.py --model 'resnet50_mocov2' --device cpu –crop_path “path/to/crop/dir” --viz_validSegs “True_for_storing_valid_segment_seperately”
            - python feat_extractor.py --model 'resnet50_mocov2' --device cuda:0 --crop_path ./outputs/crop_dir --viz_validSegs True

5. Segment-wise pseudo labeling:
    1. Run CMD: python seg_pseudoLabeling.py --csv True --n_clusters 6 --save_clusters True
        - set “--csv” attribute True for generating pseudo labels of vision transformers’ feature set (CLS). For CNN-based set False.
        - The csv file (train_PlabelLUV_Plus.csv) containing pseudolabeling information of valid segments will be saved in output directorey. Also, centroide information will be saved as npy file.
        - If “--save_clusters” is True then cluster will be saved in outputs directory.

6. Generate initial pseudo-annotated segmentation masks:
   1. Run CMD: python generate_PseudoMapV2.py --split train --n_clusters 6 --patch_size 8 --device cuda:0
      - The pseudo masks will be saved into the outputs directory in seperate folder.

7. Evaluate the initial pseudo-annotated segmentation masks:
   1. Run CMD: python evalV2.py --split train --n_clusters 6 --t_clusters 6 --data_dir './data' --device cpu
      1. '--n_clusters' and '—t_clusters' indicated number of pseudo clusters and true clusters, respectively.

8. Visualize colored pseudo masks:
   1. Run CMD: cd utils
   2. Run CMD: python color_Mask.py --n_clusters 6 --t_clusters 6 --split train --mask_path 'path/to/grayMask' --map_path 'path/to/map.csv'