SegPC-2021

This is the official repository for the ISBI 2021 paper **Transformer Assisted Convolutional Neural Network for Cell Instance Segmentation** by Deepanshu Pandey, Pradyumna Gupta, Sumit Bhattacharya, Aman Sinha, Rohit Agarwal.

Getting Started

We recommend using Python 3.7 for running the scripts in this repository. The necessary packages can be installed using *requirements.txt* in the respective folders. Since all of our work has been done on Google Colaboratory, the *requirements.txt* may have more packages/modules than is actually required and it might take quite long to install everything. Hence,for such a case, the folders of both the models also contain an *essential-requirements.txt* file which contains some essential packages that need to installed beforehand, while the other fundamental packages can be installed later as their need shows up as an error when running the given training and inference scripts.

To run this repository, following the given steps using the sections mentioned in the subsequent sections:

1) Prepare the data in COCO format

2) Run the training script for Cascade Mask RCNN / DetectoRS

3) Run the inference script for Cascade Mask RCNN / DetectoRS

4) Run the ensemble script

Data Preparation

Note : This step is not required for inference.

All the models present in the paper require data in COCO format to train. Hence , to train the models the images and masks need to be resized and a json file in COCO format is required. The dataset_preparation.py script in the utils folder can be used to perform these tasks. The following flags need to be used for running the dataset_preparation.py script:

```
usage: dataset_preparation.py [-h] --img_root IMG_ROOT --mask_root MASK_ROOT --dest_root DEST_ROOT

arguments:
  -h, --help            show this help message and exit
  --img_root IMG_ROOT   path to the folder where the images are saved
  --mask_root MASK_ROOT
                        path to the folder where gt instances are saved
  --dest_root DEST_ROOT
                        path to the folder where the COCO format json file and resized masks and images will be saved
```

Cascade Mask RCNN

For installation of required packages:

```
$ cat Cascade_Mask_RCNN_X152/requirements.txt | xargs -n 1 pip3 install
```

Train

The following flags need to be used to run CMRCNN_X152_train.py:

```
usage: CMRCNN_X152_train.py [-h] --backbone {Original,Effb5,Transformer_Effb5} --train_data_root TRAIN_DATA_ROOT
--training_json_path TRAINING_JSON_PATH --val_data_root VAL_DATA_ROOT --validation_json_path VALIDATION_JSON_PATH
--work_dir WORK_DIR [--iterations ITERATIONS] [--batch_size BATCH_SIZE]

arguments:
  -h, --help            show this help message and exit
  --backbone {Original,Effb5,Transformer_Effb5}
                        The backbone to be used from the given choices
  --train_data_root TRAIN_DATA_ROOT
                        path to training data root folder
  --training_json_path TRAINING_JSON_PATH
                        path to the training json file in COCO format
  --val_data_root VAL_DATA_ROOT
                        path to validation data root folder
  --validation_json_path VALIDATION_JSON_PATH
                        path to validation json file in COCO format
  --work_dir WORK_DIR   path to the folder where models and logs will be saved
  --iterations ITERATIONS
  --batch_size BATCH_SIZE
```

Inference

The following flags need to be used while running CMRCNN_X152_inference.py:

```
usage: CMRCNN_X152_inference.py [-h] --backbone {Original,Effb5,Transformer_Effb5}
--saved_model_path SAVED_MODEL_PATH --input_images_folder INPUT_IMAGES_FOLDER --save_path SAVE_PATH

arguments:
  -h, --help            show this help message and exit
  --backbone {Original,Effb5,Transformer_Effb5}
                        The backbone to be used from the given choices
  --saved_model_path SAVED_MODEL_PATH
                        path to the saved model which will be loaded
  --input_images_folder INPUT_IMAGES_FOLDER
                        path to the folder where images to inference on are
                        kept
  --save_path SAVE_PATH
                        path to the folder where the generated masks will be
                        saved
```

DetectoRS

Preparation script should be run with the following command before running any other file in the DetectoRS folder :

`$ bash mmdetection_preparation.sh`

For installation of required packages:

`$ cat DetectoRS/requirements.txt | xargs -n 1 pip3 install`

Train

The following flags need to be used while running DetectoRS_train.py:

```
usage: DetectoRS_train.py [-h] --backbone {Original,Effb5,Transformer_Effb5} --train_data_root TRAIN_DATA_ROOT
--training_json_path TRAINING_JSON_PATH [--train_img_prefix TRAIN_IMG_PREFIX] [--train_seg_prefix TRAIN_SEG_PREFIX]
--val_data_root VAL_DATA_ROOT --validation_json_path VALIDATION_JSON_PATH [--val_img_prefix VAL_IMG_PREFIX]
[--val_seg_prefix VAL_SEG_PREFIX] --work_dir WORK_DIR [--epochs EPOCHS] [--batch_size BATCH_SIZE]

arguments:
  -h, --help            show this help message and exit
  --backbone {Original,Effb5,Transformer_Effb5}
                        The backbone to be used from the given choices
  --train_data_root TRAIN_DATA_ROOT
                        path to training data root folder
  --training_json_path TRAINING_JSON_PATH
                        path to the training json file in COCO format
  --train_img_prefix TRAIN_IMG_PREFIX
                        prefix path ,if any, to be added to the train_data_root path to access the input images
  --train_seg_prefix TRAIN_SEG_PREFIX
                        prefix path ,if any, to be added to the train_data_root path to access the semantic masks
  --val_data_root VAL_DATA_ROOT
                        path to validation data root folder
  --validation_json_path VALIDATION_JSON_PATH
                        path to validation json file in COCO format
  --val_img_prefix VAL_IMG_PREFIX
                        prefix path ,if any, to be added to the val_data_root path to access the input images
  --val_seg_prefix VAL_SEG_PREFIX
                        prefix path ,if any, to be added to the val_data_root path to access the semantic masks
  --work_dir WORK_DIR   path to the folder where models and logs will be saved
  --epochs EPOCHS
  --batch_size BATCH_SIZE
```

Note: DetectoRS requires semantic masks along with instance masks during training , hence the arguments - train_seg_prefix and val_seg_prefix

Inference

The following flags need to be used while running DetectoRS_inference.py:

```
usage: DetectoRS_inference.py [-h] --backbone {Original,Effb5,Transformer_Effb5}
--saved_model_path SAVED_MODEL_PATH --input_images_folder INPUT_IMAGES_FOLDER --save_path SAVE_PATH

arguments:
  -h, --help            show this help message and exit
  --backbone {Original,Effb5,Transformer_Effb5}
                        The backbone to be used from the given choices
  --saved_model_path SAVED_MODEL_PATH
                        path to the saved model which will be loaded
  --input_images_folder INPUT_IMAGES_FOLDER
                        path to the folder where images to inference on are kept
  --save_path SAVE_PATH
                        path to the folder where the generated masks will be saved
```

Ensemble

Apart from the individual models, the paper also presents the scores of ensemble of any three models. The ensemble.py script in the utils folder can be used for making ensemble of the outputs of three models , using the following flags :

```
usage: ensemble.py [-h] --model1_predictions MODEL1_PREDICTIONS
--model2_predictions MODEL2_PREDICTIONS --model3_predictions MODEL3_PREDICTIONS
--final_predictions FINAL_PREDICTIONS

arguments:
  -h, --help            show this help message and exit
  --model1_predictions MODEL1_PREDICTIONS
                        path to the predictions of first model
  --model2_predictions MODEL2_PREDICTIONS
                        path to the predictions of second model
  --model3_predictions MODEL3_PREDICTIONS
                        path to the predictions of third model
  --final_predictions FINAL_PREDICTIONS
                        path where the ensembled outputs should be saved
```

Results and Models

Method	Backbone	Download
Cascade Mask R-CNN	Original(ResNet)	model
DetectoRS	Original(ResNet)	model
Cascade Mask R-CNN	EfficientNet-b5	model
DetectoRS	EfficientNet-b5	model
Cascade Mask R-CNN	EfficientNet-b5+ViT	model
DetectoRS	EfficientNet-b5+ViT	model