1. <u>Testing:</u>

Program name:new_predict.py

- Generate metrics such as Average IOU, F1 score, Accuracy, Precision, Recall.
- Save results images in directory specified.

2. <u>Training:</u>

Program name: train.py

- used to train U-Net with backbones such as VGG19, ResNet - 50, ResNet - 101
-Will also generate metrics graphs at end of training, such as training and validation loss, learning rate, training and validation IOU.

3. <u>Data Cleaning:</u>

Program name: data_cleaning.py

Select images and corresponding binary masks such that amount of foreground in binary masks is within certain desired range.

4. <u>Threshold selection:</u>

Program name: iou selection.ipynb
Algorithm which finds out for which value of binary threshold value the model gives better IOU with ground truth binary masks.

5. <u>Data Generator:</u>

Program name: data.py

Data generator is used to read training and validation of data on fly during training and apply custom designed augmentation pipeline.

6. <u>Loss functions:</u>

Program name: project/segmentation_models/losses.py

Has various loss function defined such as binary_crossentropy, binary_focal_loss, dice_loss.

7. <u>Metrics functions:</u>

Program name: project/segmentation_models/metrics.py

Has various metric functions defined such as IOU score, F score, Precision, Recall

8. U Net Model building:

Program name: project/segmentation_models/models/unet.py

9. Various model backbone architectures:

Program name: project/segmentation_models/backbones/classification_models-master/classification_models-master/classification_models/models_factory.py

10. Model weights, training graphs

All the model weights and related graphs are stored in folder named "train_summary".

11. Training and testing data set:

Training images and it's corresponding binary masks are kept in folder "dataset/train/images_new_selected/" and "dataset/train/images_new_selected/masks_new_selected".

Testing images and it's corresponding binary masks are kept in folder "dataset/test/images_new_selected/" and "dataset/test/masks_new_selected/".

12. Anaconda environment replication:

Use the following command in anaconda prompt:

```
conda env create -f environment.yml
```

It uses given environment.yml file in project folder to install necessary libraries.