Approach/Methods

- Framework: tf.keras API
- Deployment: AWS, Heroku
- Fronend : HTML
- Backend: Flask, Python

Stage - 1 (only for deployment) :

- In the initial stage, I have used the Simple Unet model and trained it on 3600 image samples with Adam optimizer and cross-entropy as the loss function.

- **IOU(**Intersection over Union) has been used as an evaluation metric.

- The original Image size was 101 X 101. So I have resized the image with 128 X 128.

- After training has saved model architecture and model weights.

- I have made an app.py file and in that, I have used a saved model to get output.

- I have deployed the final flask app on AWS EC2 Instance and on Heroku also.

- I tried hard to deploy the app on the GCP App engine but i was facing an error of L2 cache and daily server not responding errors.

- I don’t have a premium account of AWS so I haven’t tried to deploy using AWS Lambda service.

Stage - 2 (Improved performance metric) :

- To Improve model performance I have built different model architecture with the same loss function, optimizer, and performance metric as a stage 1 training process.

- In this stage, the model ran for 29 epochs and I got a validation IOU of 0.79.

- After training, I tried to find the best threshold value from the range between 0.3 to 0.7 and I got 0.6 as the best threshold value which has given 0.75 private scores and a 0.726 public score on Kaggle.

Stage -3 (Final model)

- In the third try, I have used Yakubovskiy’s Github library called Segmentation Models, which was fairly easy. Segmentation models are a Keras-based Python library that implements four popular segmentation architectures and dozens of ImageNet pre-trained backbones that can easily be combined into the segmentation model of choice.

- I have used Unet model with resnet50 as the backbone and imagenet pretrained weight as encoder weights. I have not to train encoder weight because of the large number of training parameters. I have only trained the decoder network of Unet model.

- I have trained this Unet model for 35 epochs with binary cross-entropy as loss function and Adam as optimizer with decaying learning rate.

- I have got a 0.82 IOU score on 400 validation images.

- I tried to find the best threshold value to get a better IOU score and I got 0.55 as the best threshold value which gives a 0.85 IOU score on the validation dataset.

- By using 0.55 as the threshold value I got a private score of 0.815 and a public score of 0.792 on 18000 test images.

Kaggle score :

4 submissions for sumit tagadiya Sort by Most recent

All Successful Selected

Submission and Description	Private Score	Public Score	Use for Final Score
submission_final.csv a minute ago by sumit Unet with Resnet50	0.79743	0.77718	☐
submission_final_new.csv 3 minutes ago by sumit Unet model with resnet50 backbone and Imagent weight as encoder weights.	0.81569	0.79284	☐
submission1.csv 7 hours ago by sumit second submission	0.73807	0.70853	☐
submission.csv 8 hours ago by sumit	0.75468	0.72631	☐

- You can find deployment on AWS here and on Heroku here.

You might be facing an issue in the loading of AWS deployment link because it is not a secure connection so please try to open the link in private mode.

Heroku takes slightly more time to load links because of 526 MB of ram only. So wait for a while it will load successfully though.

Problems which I had faced during this project :

Trining difficulties :

- When I was training my first initial model I was getting loss value negative large like -10k and -50k as binary cross-entropy.

- After so many tries I found the clue. The problem was in target image datatype though I have normalized both training and target images by dividing it with 255. But target images had not boolean data type that's why I was getting such a large value of binary cross-entropy.

- After making the target image data type as boolean I solved the issue and I was getting better results.

Deployment difficulties :

- As a first try, I decided to deploy the model on the GCP app engine. I tried to deploy my flask app on the free tier of the GCP app engine. My whole app code was pushed successfully as per the log streaming but it was failed at the final deployment push.

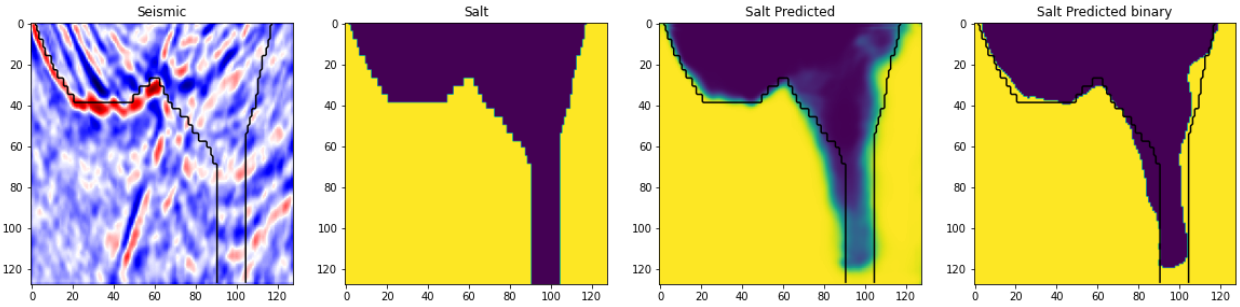
- I read every log stream and tried to figure out the problem and I had tried almost 10 times in 2 days but still, I was not able to deploy my flask app on the GCP app engine.

- So as an alternative to GCP I have deployed my flask app on AWS EC2 instance and also on the Heroku platform.

- I have not premium account of AWS so I haven't tried to deploy an app using AWS Lambda services.

Here are some of the Test results :

- **Result from validation data.**



In the above image yellow part is the salt area and the blue part is the background.

- **Result from test data**