Things done:

1) Getting the dataset ready for further use (like change of format)
2) Researched models that are feasible to use for our purpose
3) Testing of the models

Process in Detail:

1) Getting the dataset ready for further use (like change of format)

   Upload the annotated data to account created in https://app.roboflow.com/

   Here we can do data augmentation, format conversion and also enble us to use it directly in platforms like python, c++ etc.

   For the above purpose refer: https://www.youtube.com/watch?v=76E6esnez8E
2) Researched models that are feasible to use for our purpose

   Short listed some good models:
   (i) RCNN
   (ii) Mask RCNN
   (iii) Fast RCNN
   (iv) Faster RCNN
   (v) R-FCN
   (vi) YOLO
   (vii) Blitznet
   (viii) Retinanet

   I decided not to use (i) as it is not very accurate, (ii) as we don't need masked region of the pre cancer

3) Testing the model

   (i) Faster RCNN:
   There are a lot of different backbones to try here. I have tried the following :

fasterrcnn_darknet, fasterrcnn_mobilenetv3_large_fpn, fasterrcnn_squeezenet1_1, resnet50_fpn, resnet50_fpn_v2, vitdet, vitdet_tiny, fasterrcnn_efficientnet_b0, fasterrcnn_efficientnet_b4

Out of the above only resnet101, resnet50_fpn and fasterrcnn_efficientnet_b0 gave good results. Also see Vitdet.

(ii) Retinanet:

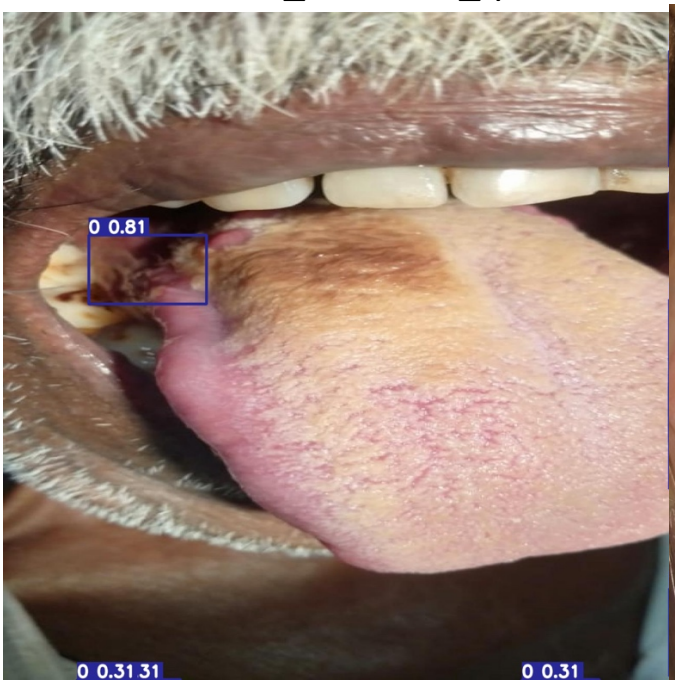
The result were not very good, but when precancer was detected it was correctly.

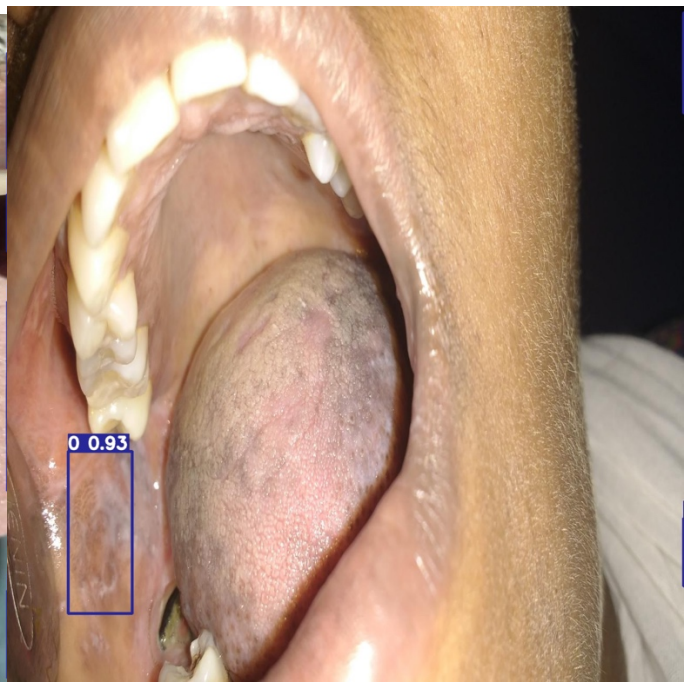
(iii) Yolov5:

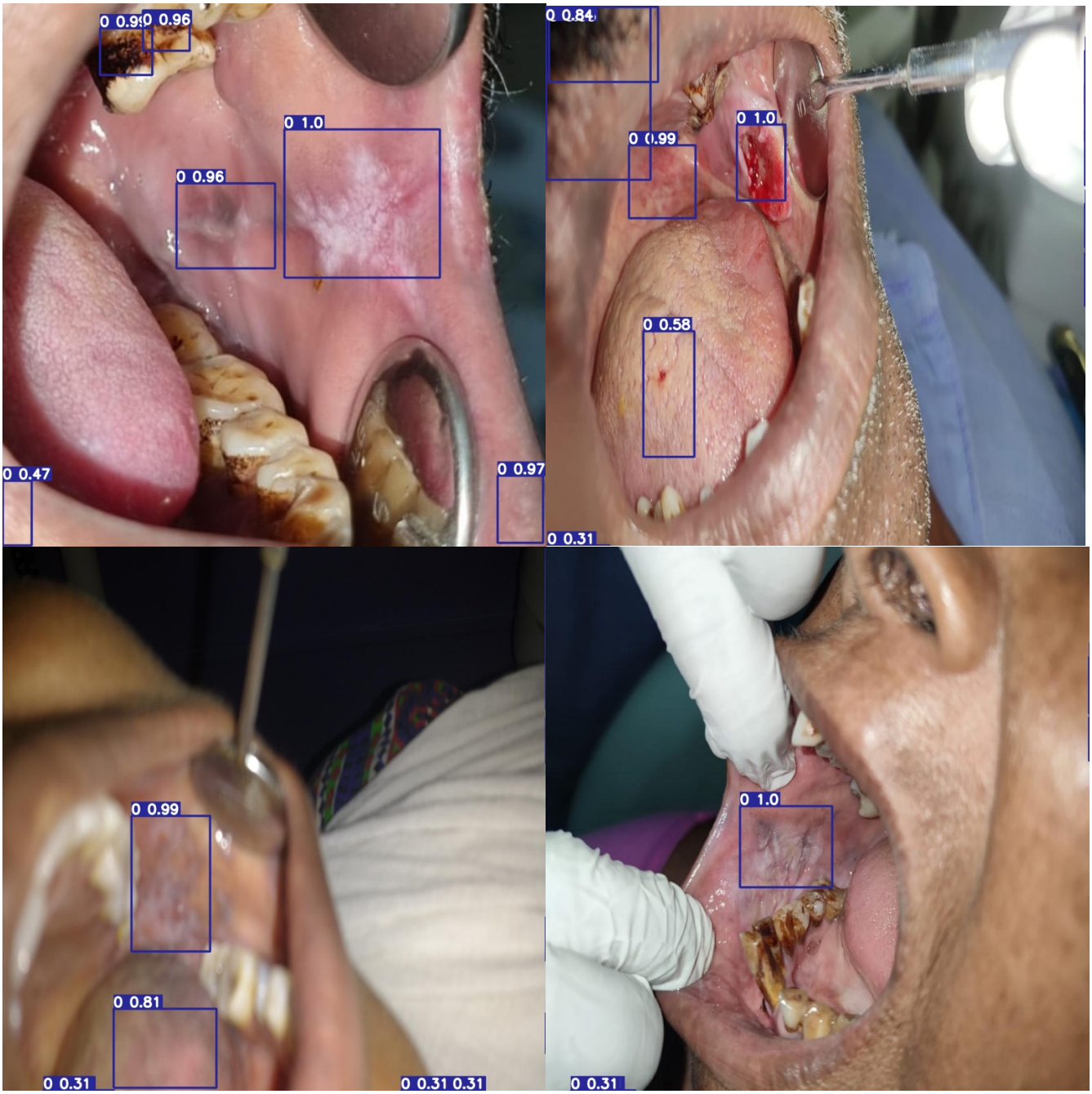
Accuracy was good but it also gave many false prediction.

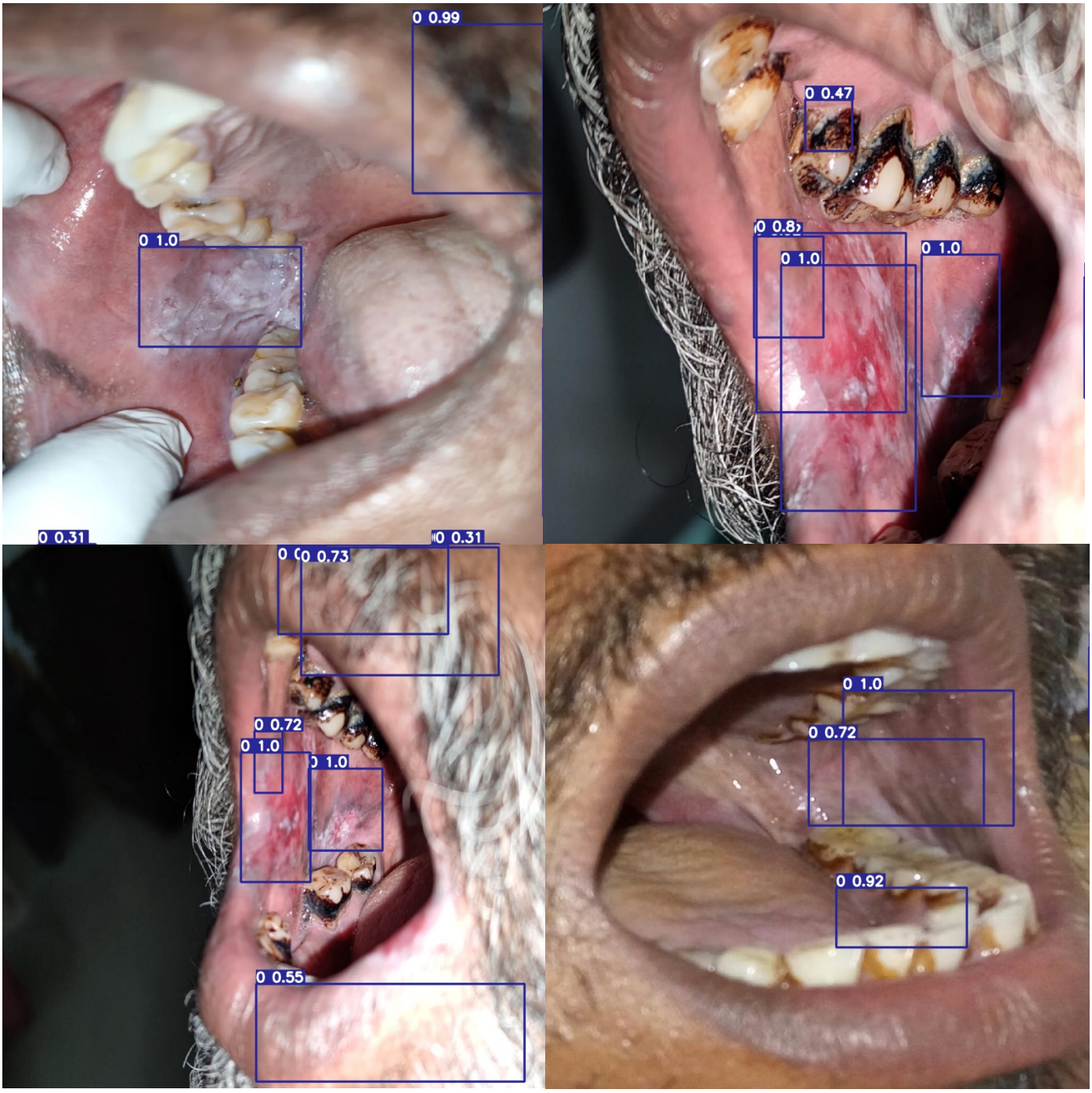
Results: (using 239 images in total)

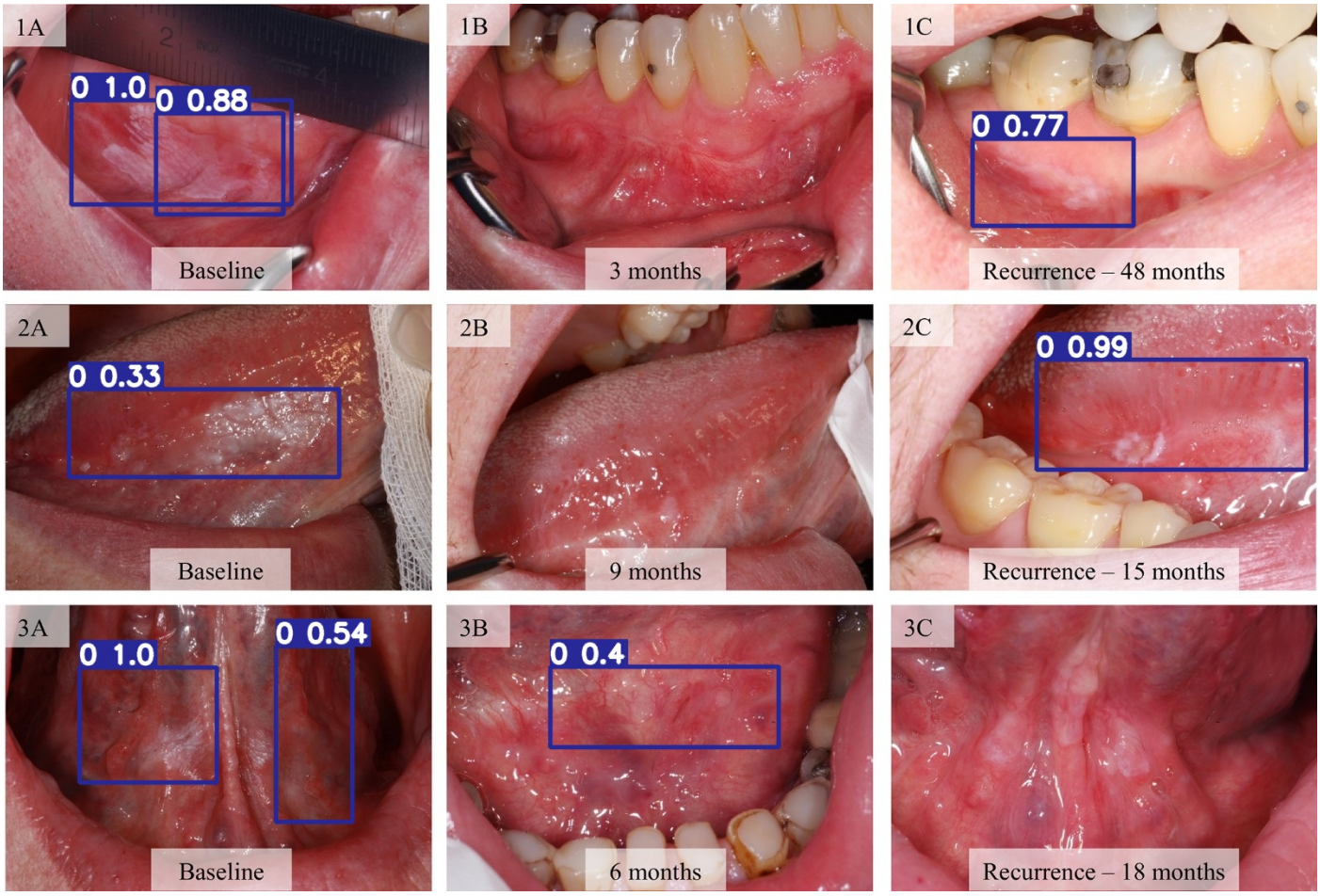
Fasterrcnn_resnet50_fpn



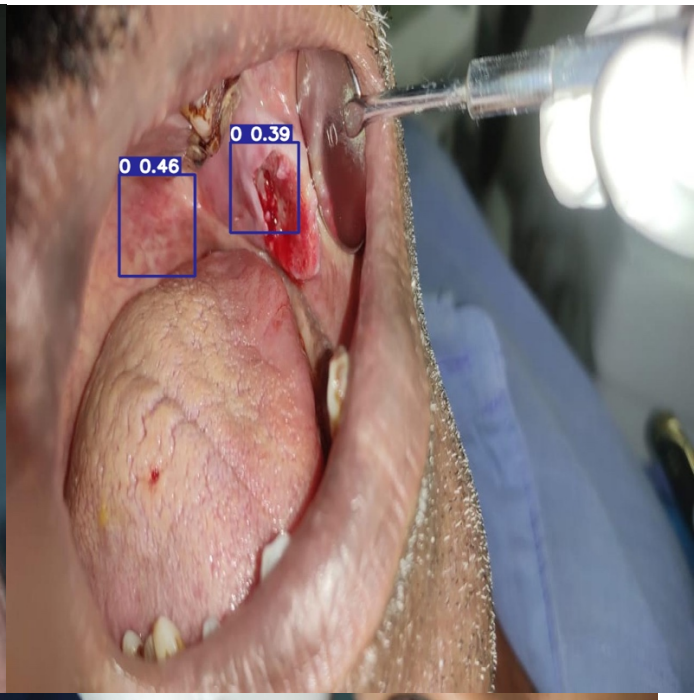



0 0.99
0 0.96
0 1.0
0 0.96
0 0.47
0 0.97
0 0.84
0 1.0
0 0.99
0 0.58
0 0.31
0 0.99
0 0.81
0 0.31
0 0.31 0.31
0 1.0
0 0.31


0 0.99
0 1.0
0 0.47
0 1.0
0 1.0
0 0.31
0 0.31
0 0.73
0 0.72
0 1.0
0 1.0
0 0.55
0 1.0
0 0.72
0 0.92



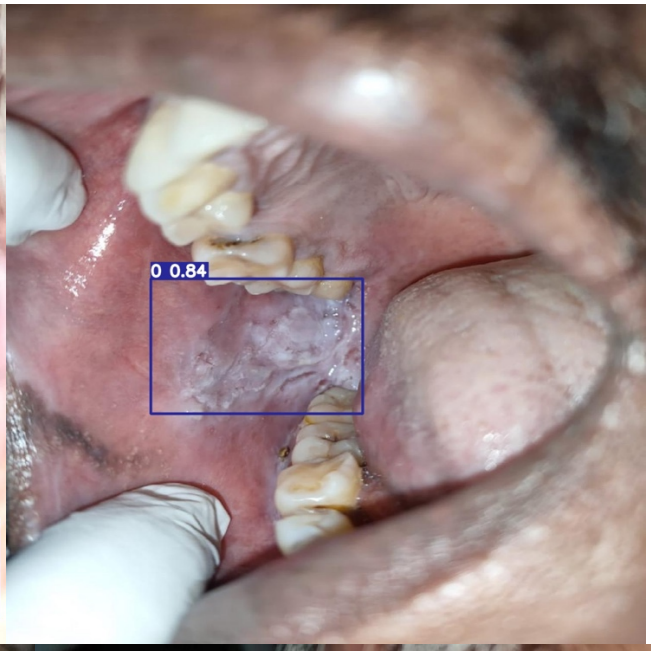
fasterrcnn_resnet101


0 0.46
0 0.39

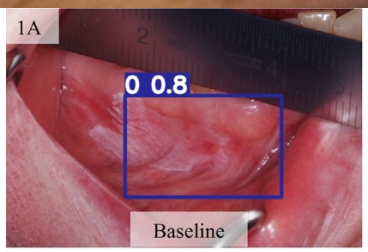

0 0.44


0 0.99


0 0.34


0 0.84


0 0.46


1A
0 0.8
Baseline

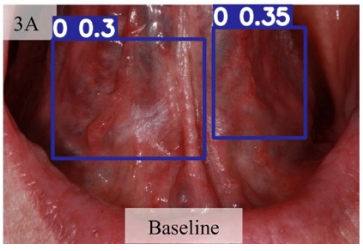

1B
3 months


1C
Recurrence – 48 months

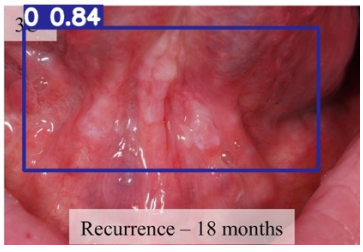

2A
Baseline


2B
9 months

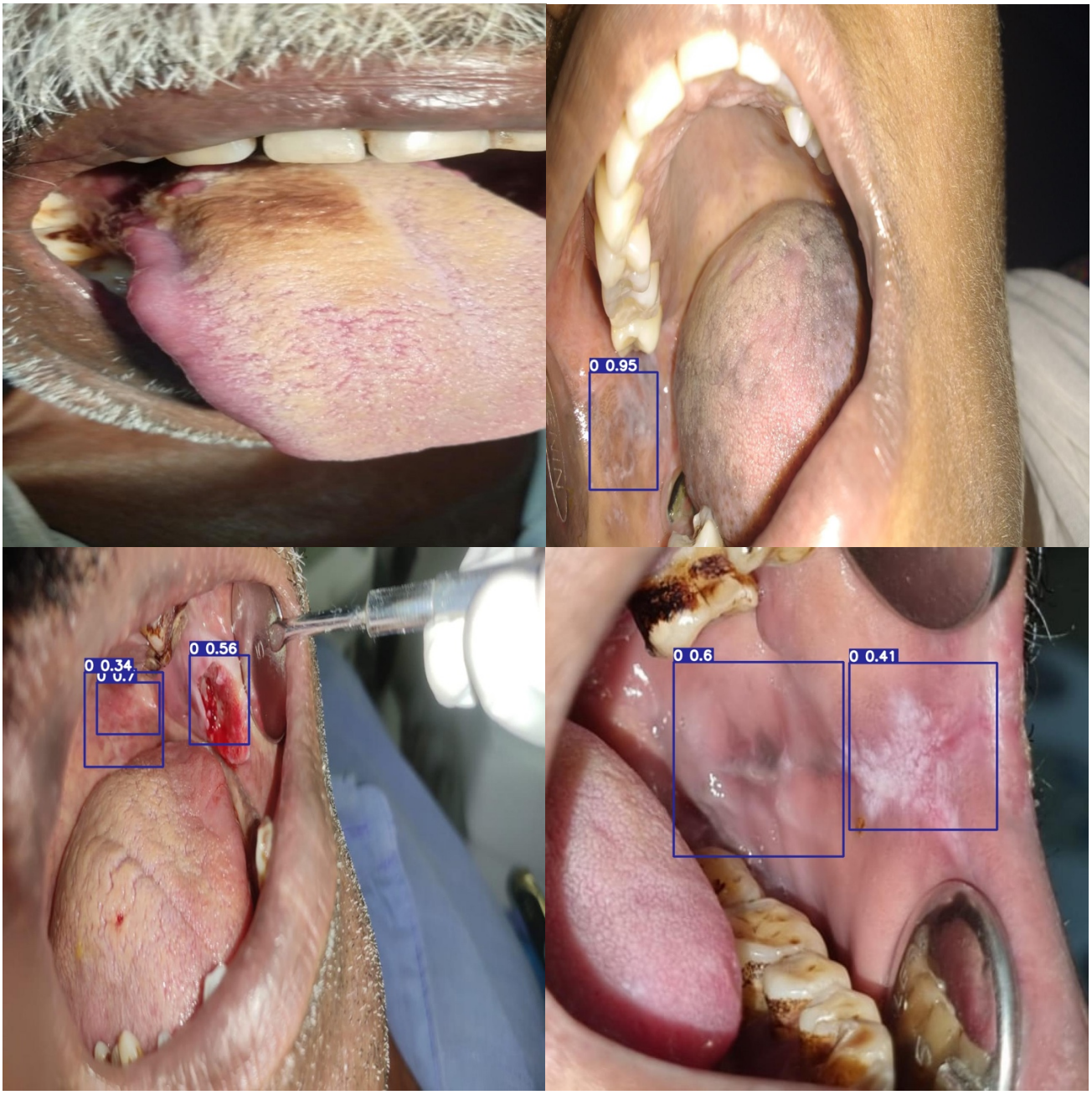

2C
Recurrence – 15 months


3A
0 0.3
0 0.35
Baseline

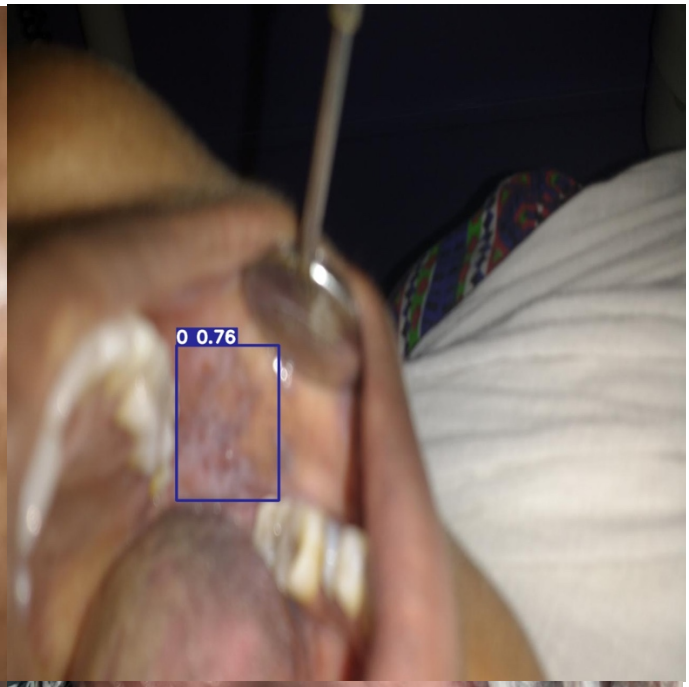

3B
6 months

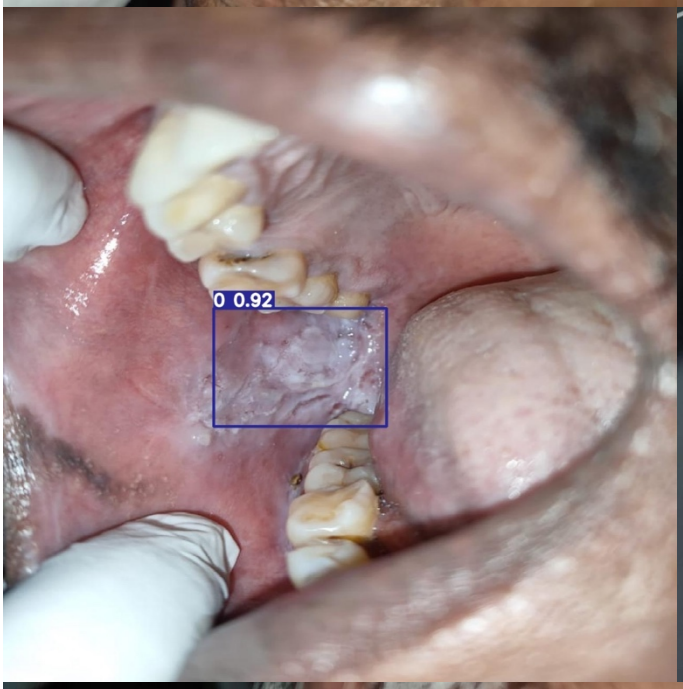

0 0.84
Recurrence – 18 months

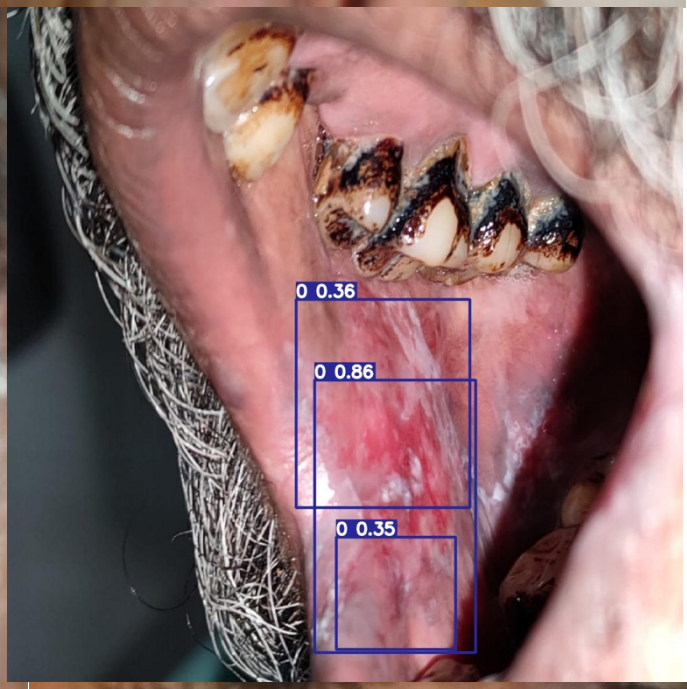
fasterrcnn_efficientnet_b0

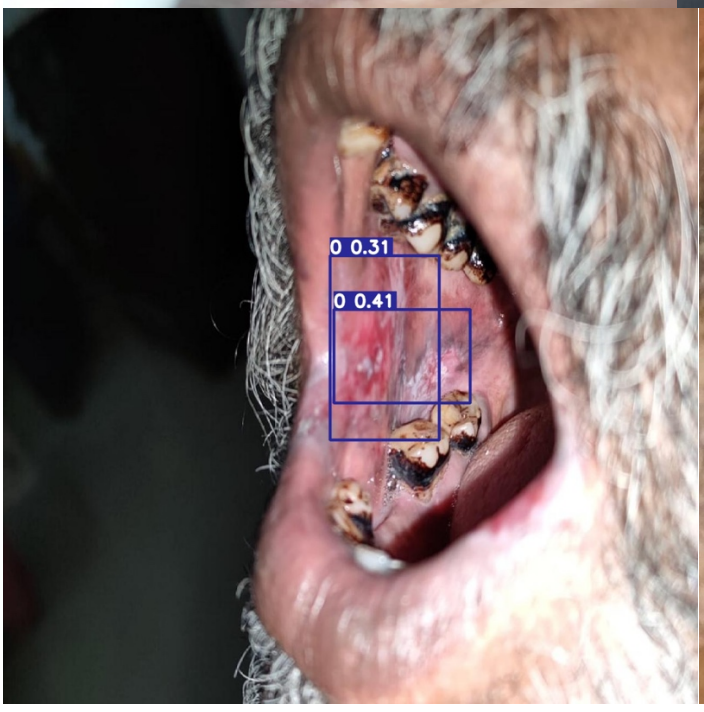


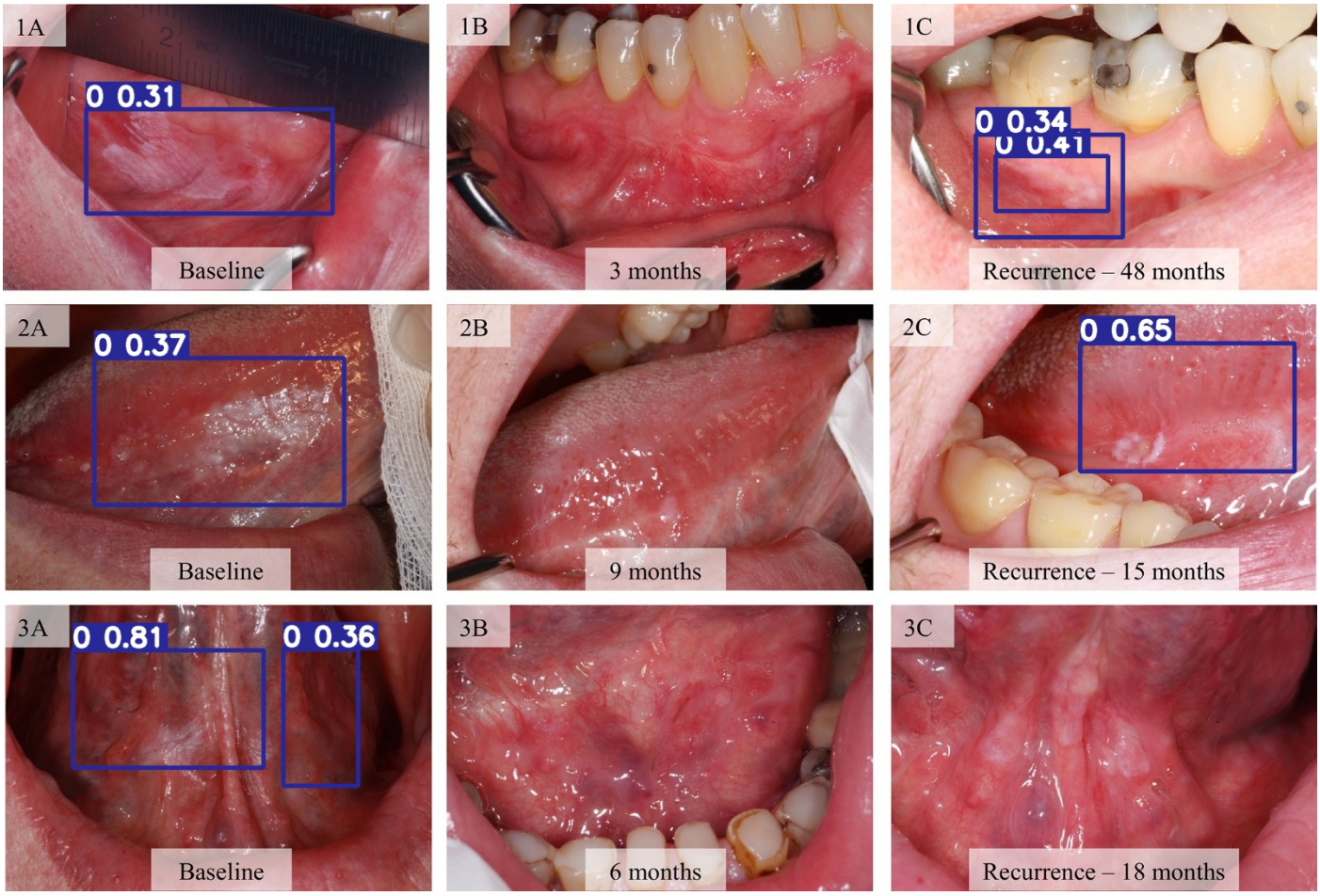

0 0.99
0 0.51
0 0.8


0 0.76

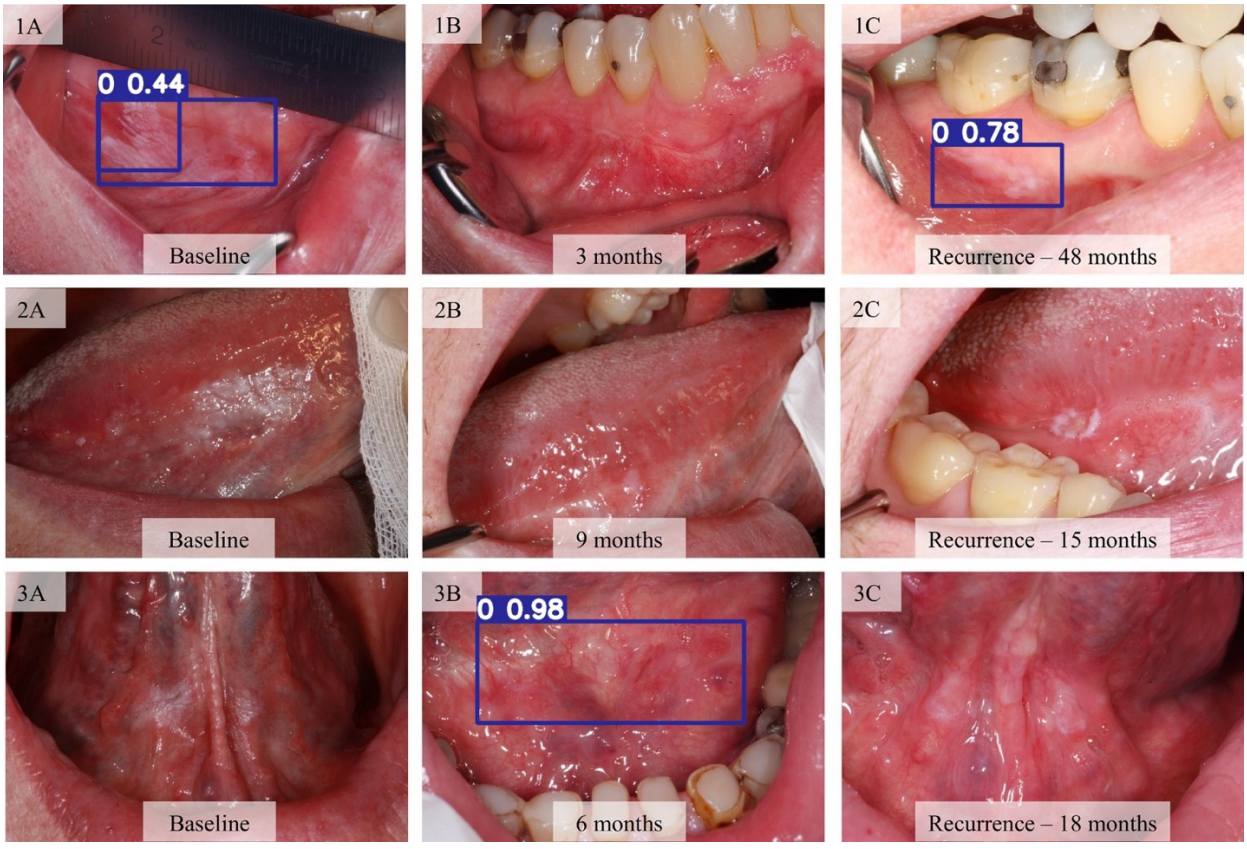

0 0.92


0 0.36
0 0.86
0 0.35

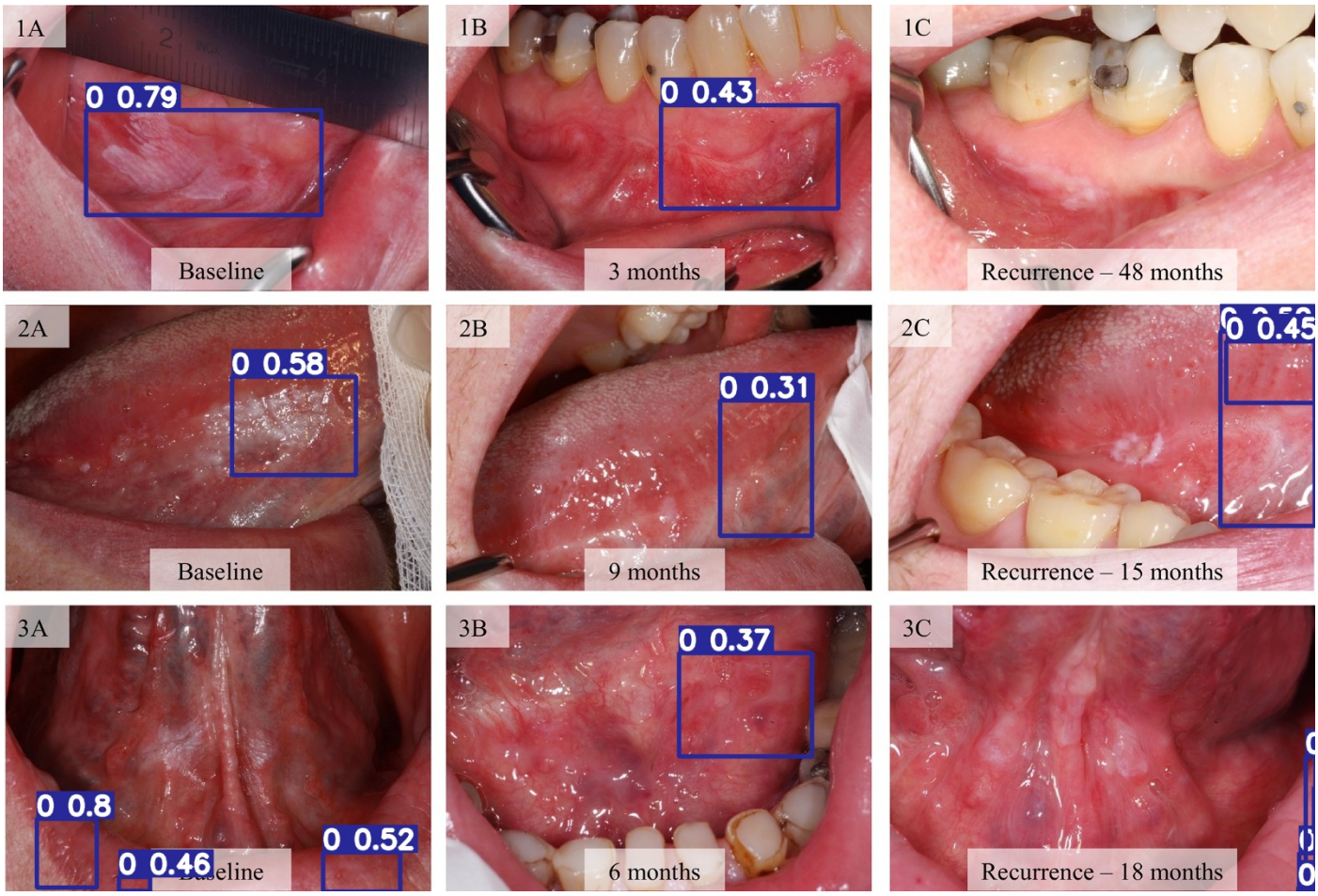

0 0.31
0 0.41



fasterrcnn_mobilenetv3_large_fpn

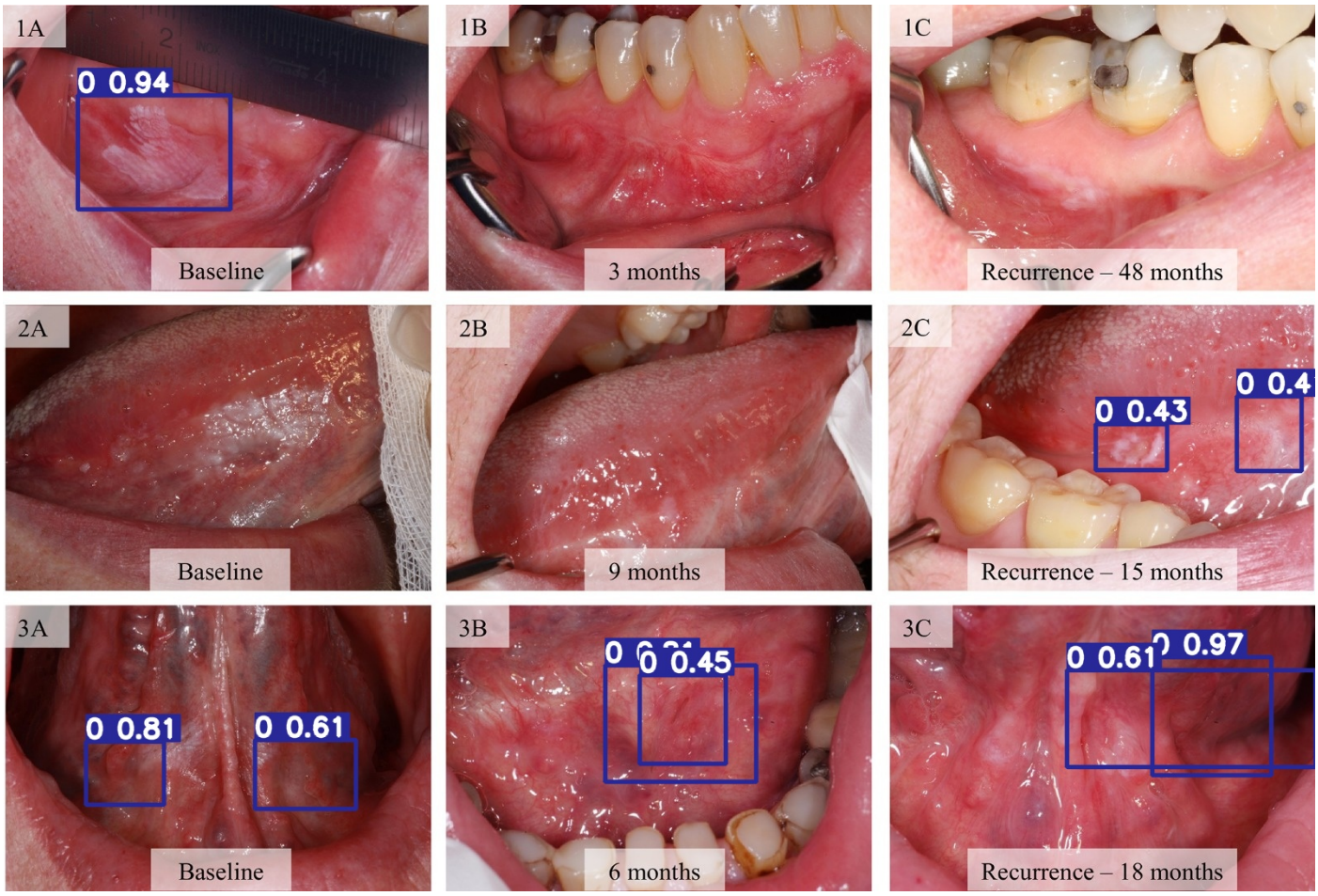


fasterrcnn_squeezenet1_1

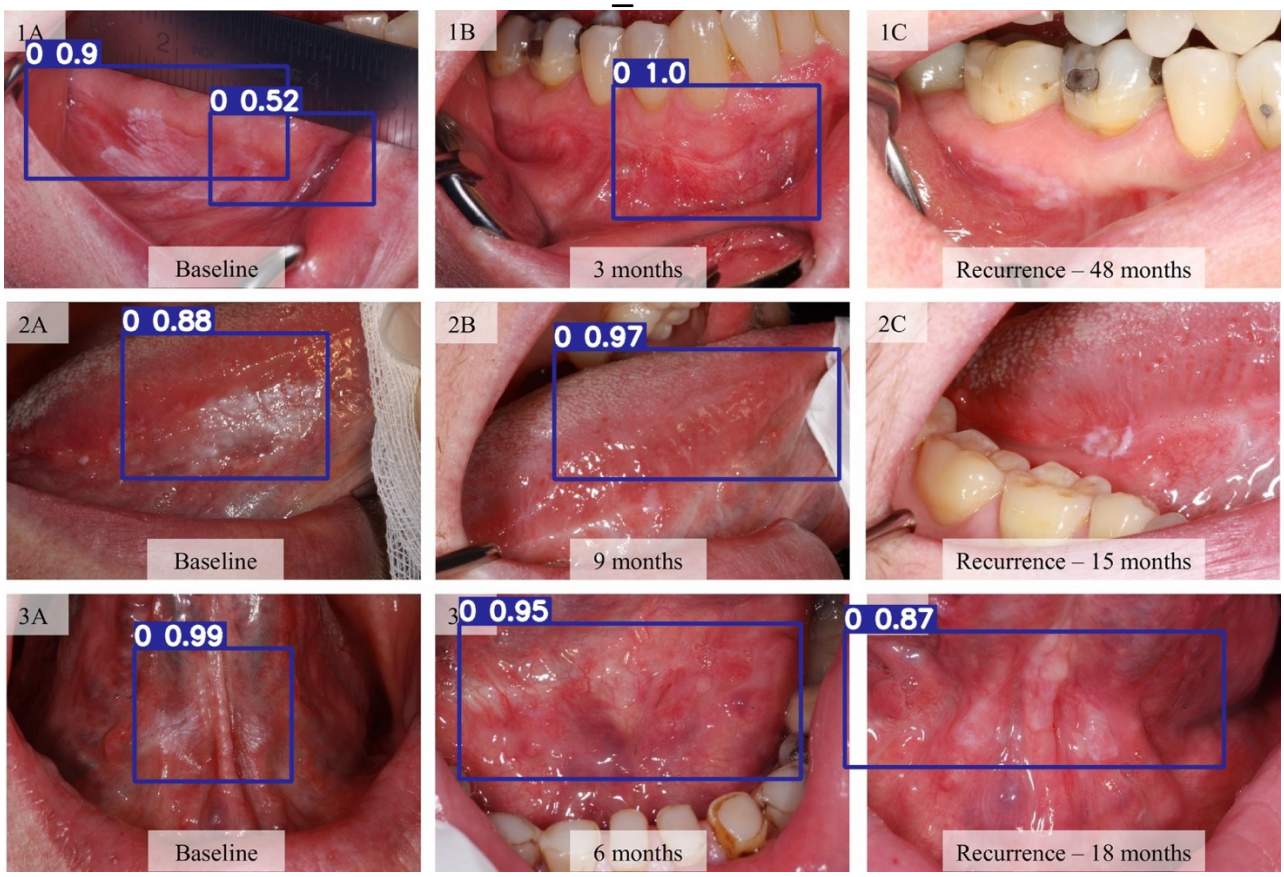


vitdet_tiny

Fasterrcnn_vitdet



Fasterrcnn_darknet



Procedure for various models:

i) Retina net

To train :

!python train.py --dataset coco --coco_path <path of .coco file of the dataset> --depth <depth of network> --epochs <number of epochs>

To test a coco dataset:

!python visualize.py --dataset coco --coco_path <path of .coco file of the dataset> --model <path of .pt file of weights obtained during training>

To test a file of images:

python visualize_single_image.py --image_dir <directory of test images> --model_path <.pt file path> --class_list <.csv of classlist path>

what is classlist : we have to make a csv with classes of the following format,
class_key, class_name . Also include 0,background.

Visit : https://github.com/yhenon/pytorch-retinanet for further information

The data should of a .json format and the names should be very specific:
Steps:

1) Create file named “oral_precancer.v1i.coco” inside the cloned repo.
2) Make two directory “annotations”,”images”
3) In “annotations” : rename annotation of train data as “instances_train2017.json” and of validation as “instances_val2017.json”
4) In “images” : store train images in “train2017” and validation images in “val2017”

ii) Faster RCNN

To train:

python train.py --data <location of .yaml file of the json format data> --epochs <number of epochs> --model <model name> --name <name of file in which to save the results> --batch <batch size>

To test:

python inference.py --input <location of directory of test images> --weights <path of .pth either best_model.pth or last_model_state.pth>

the results of training are saved at ./outputs/training file

the inferences are saved at ./outputs/inference

to find all available models check ./models

go to /home/aimlgpu2/yash/oral_pre_cancer/fastercnn-pytorch-training-pipeline on the server to get an idea as it is already implemented there.

For further info on this visit : https://github.com/sovit-123/fasterrcnn-pytorch-training-pipeline
It also contains info on how to make the .yaml file.

You can also change the model used in efficient net file by going to fasterrcnn_efficientnet_b0.py but you will also have to change the "backbone.out_channels =" (for b0,b1 : 1280, b2:1408, b6:2304 , you can find this value from error that is thrown if you can model and don't change this value).

You can also freeze the weights of backbone:

```
for param in model.backbone.parameters():
      param.requires_grad = False
```

Just add the above line

You can find all of this stuff on this folder on gpu2 server : "/home/aimlgpu2/yash/ oral_pre_cancer" , ip : sftp://192.168.15.5