# System Requirements Specification

# Index

## For

# Polyp Segmentation Application

Version 1.0

# POLYP SEGMENATION APPLICATION

## System Requirement Specification

# 1 PROJECT ABSTRACT

The **Polyp Segmentation Application** is an ML code test for candidate for checking their coding skills.

**Following are the requirement specifications**:

| **File Name** | **Module Names** | **Functionality** | **Problem Statement for Candidate** |
|---|---|---|---|
| dataloader.py | PolypDatasetLoader class and all its dataloader functionalities | It contains the class that loads the data from the folder and returns the image and mask as pytorch tensors. | You have the template for the class. Code all the functionalities mentioned in the class for loading the data, augmenting it and retuning the image and mask pytorch supported tensor |
| loss.py | DiceBCELoss class | It contains the class used to calculate the error between original mask and predicted mask. | You have to first flatten both the tensors then calculate dice loss and binary cross entropy loss between them. As a result, sum them and return them as final loss |
| metric.py | dice_cofficient function | It contains the dice_cofficient function that calculates the similarity between | You have to first flatten both the tensors then calculate dice coefficient between the mask tensors. |

| | | | |
|---|---|---|---|
| | | original mask and predicted mask | As the result, return the dice coefficient. |
| model.py | PolypModel class | It contains class for creating the model which takes input images of Bx3x512x512 where B is the number of batches. It outputs the mask of shape Bx1x512x512 | You have to create a model that takes input image of the given size and returns a mask of that of given size. Make it complex to understand the features and it should return a jaccard coefficient value on the test samples be more that 80%. |
| train_helper.py | __initialize_hyperparameters function | It contains code for initializing the optimizer, schedular and loss function | You have to add any type of optimizer and schedular of your choice. |

# 2. TEMPLATE CODE STRUCTURE

## 2.1 Package: PolypSegmentationApplication

**Resources**

| Names | Resource | Remarks | Status |
|---|---|---|---|
| Package Structure | | | |
| PolypDatasetLoader | dataloader file | Contains the PolypDatasetLoader class that loads the data and returns the image and mask as tensors | Not implemented |
| DiceBCELoss | loss file | Contains the DiceBCELoss class that is used to calculate the margin of error between original mask and predicted mask. | Not Implemented |

| | | | |
|---|---|---|---|
| dice_cofficient | metric file | Contains the dice_cofficient function that calculates the similarity between original mask and predicted mask | Not Implemented |
| __initialize_hyperparameters function | train_helper file | It contains code for initializing the optimizer, schedular and loss function | Partially Implemented |
| PolypModel Class | model file | It contains class for creating the model which takes input images of Bx3x512x512 where B is the number of batches. It outputs the mask of shape Bx1x512x512 | Not implemented |

### 5.2 Package: PolySegmentationApp.Tests

**Resources**

**All the tests file contains the testing code for evaluation. Don’t change or edit it.**

## 3. Commands for training and testing

After completing the code, you can use the commands given below for installing all the important packages, to train your model and to test your model:

1. **To setup environment:**

```
pip install -r requirements.txt
```

2. **To launch application:**

```
python3 main.py -d “data/PNG” -i “Original” -m “Ground Truth” -b 2 -e 100
```

3. **To run Test cases:**

```
python3 testing.py
```