APPROACH FOR THE PROBLEM:

We had divided the whole problem into two parts:

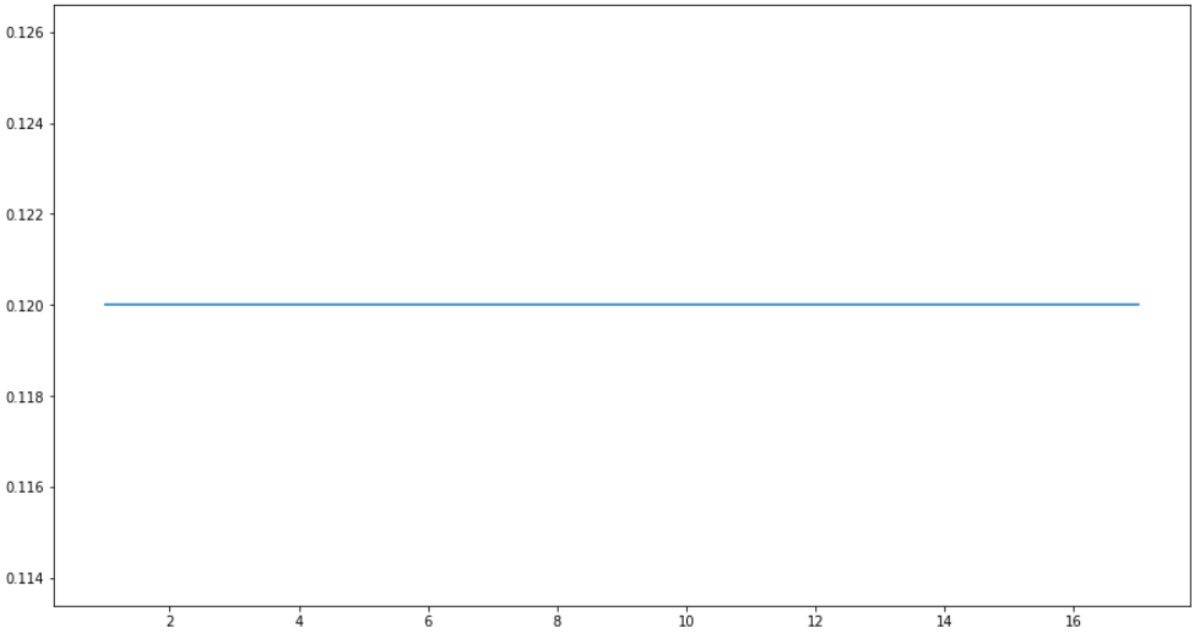
1. Deciding if a given image contains defects.
2. Predicting the bounding box for the images (which are predicted to be defective, by the model)

PART: 1 PREDICTING THE CLASS LABEL FOR THE IMAGE

- Firstly we predicted the class for the images, whether they are defective images or not. So for predicting the class labels we have used the RESNET 50 model and trained it on the training dataset.

- So in making the model we have used some hyperparameters and used a softmax activation function along with adam optimizer and learning rate as 0.0001.

- In this model, we have taken the loss as binary cross-entropy and fixed the number of output neurons to 1 as we want to classify images into defective or not-defective.



PART: 2 PREDICTING THE BOUNDING BOX

Having finished the prediction part of the image, we now take the images which have been predicted to be defective.

TRAINING THE MODEL:

- Firstly, we thought of using the latest state-of-the-art object detection model, DEtection TRansformer (DETR). The model tries to predict n number of bounding boxes and a class label corresponding to it. However, due to the limited hardware constraints, we were unable to train the model, since the GPU memory always would get out of memory.

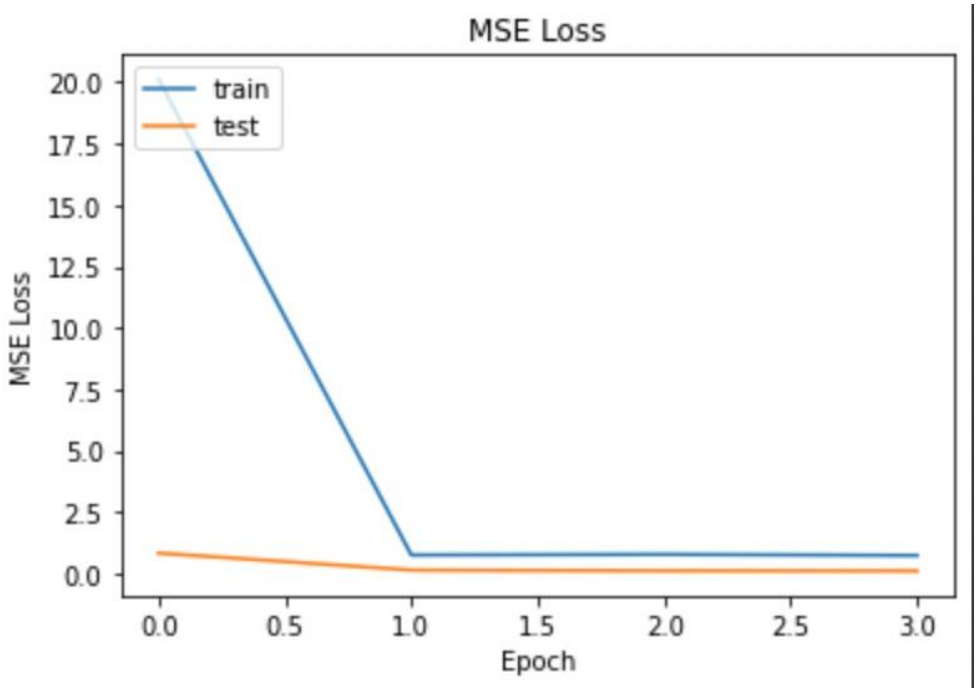
- So, we thought of changing our thought process and use the RESNET 18, (because all other models which we thought would help in making bounding boxes would get out of memory).

- Now, since **RESNET 18** has a fixed number of output neurons, we prepared data accordingly. Since most of the data had only one defect (i.e one bounding box), we took all the images, and then selected only one of the bounding boxes for our training process.

- For the purpose of Augmentation, we just applied the grayscale, and normalize the pixel augmentation (implemented in the PyTorch Framework).

TRAINING THE MODEL:

- For the purpose of training the model, we set the hyperparameters as follows:
    - **BATCH_SIZE = 4;**
    - **LEARNING RATE = 5e-5**
    - **Optimizer: Adam**
    - **EPOCHS: 10**
    - **LOSS: Mean Squared Error**
- And, having trained the model, we saved the model on the basis of the Validation MSE Loss
- The results, that we obtained are as follows:



- The validation loss was near 0.1 and the training loss was near 0.6.
- We believe that the results could have been better, if we had used the Transformer for the task of predicting the bounding box, however, due to hardware constraints, we had to resort our option to some affordable model.