Dataset: <https://www.kaggle.com/datasets/muhammad0subhan/fruit-and-vegetable-disease-healthy-vs-rotten?resource=download>

Docker command: docker run -t --rm -p 8501:8501 -v E:/SRUSHH/study/Python/Projects/freshOrNot:/verdant tensorflow/serving --rest\_api\_port=8501 --model\_config\_file=/verdant/models/models.config

Challenge:

#1:

Splitting dataset: in tensorflow, ImageDatagenerator, if we give 0.x validation split even along with shuffle=true, it will take x% data from training data before shuffling the data. Means, split occurs before shuffle. So, it is better to have separate validation dataset if the data is not shuffled properly.

Solution: I used python module split-folder for splitting data into train, test and validation data and give validation dataset instead of validation split. Also, it helped me better in testing data also.

#2(difficult one)

Predicting single image using trained model:

I trained the model. Achieved 99% test accuracy. But when I tried to predict set of few images instead of whole batch I got very weird result.

#3(dataset size large)

#4: passing base model path as relative not absolute. I though / wont make any difference but without / in beginning, it is considered as relative path only

#5: tf-serving was prediction a same class constantly  
solution: while using tf-serving u manually need to set training=Fasle or else it will unexpected results. Means, before deploying model to tf serving make sure it is in inference mode

#6: CORS policy

set NODE\_OPTIONS=--openssl-legacy-provider

npm start

#7: Non image/corrupt files as input

Soln: <https://stackoverflow.com/questions/68191448/unknown-image-file-format-one-of-jpeg-png-gif-bmp-required>