***Festival recognition***

Introduction

Festivals are a time for bonding; a time for cross-cultural exchange; a time to forgive and forget. Whether it is a National festival like independence day or Lohri in the north, Bihu in the east, Pongal in the south, or Ganesh Chaturthi in the west, the fervors of festivities is infectious and what were once considered regional celebrations, are today a cause for exuberance all over the country. In fact India can probably claim to have the maximum number of holidays in the world for festivals. So why not to make a project on the same.

This project is based on image classification on topic recognition of festival.For this , you have to pass an image and you will get to know that it belongs to which festival. For input ,images of festivals are from

1. Diwali
2. Holi
3. Ganesh Chaturthi
4. EID
5. Christmas
6. Lohri
7. Independence Day

Objective

The objective of **image classification** is to identify and portray, as a unique gray level (or color), the features occurring in an **image** in terms of the object or type of land cover these features actually represent on the ground. **Image classification** is perhaps the most important part of digital **image** analysis.

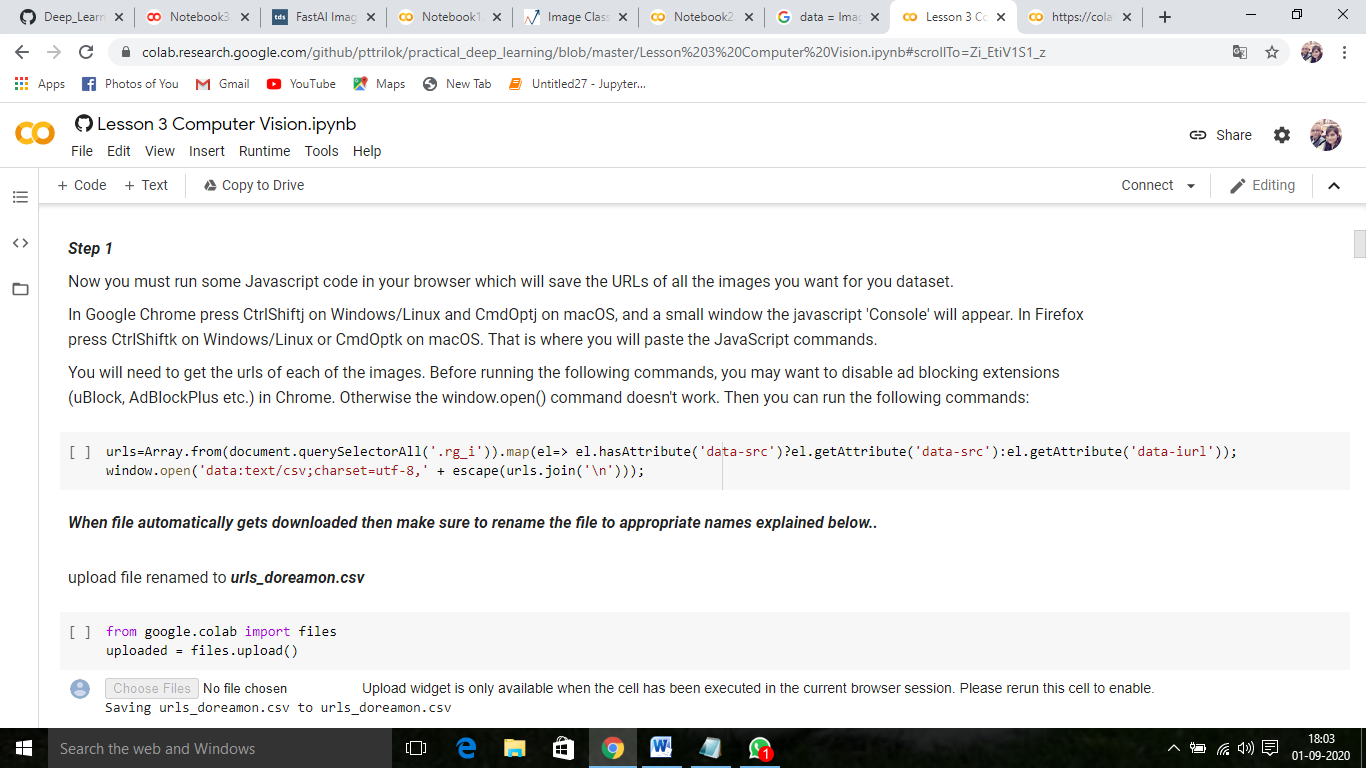
As a part of this project, we aim to predict the Festival category .An image falls into from 7 categories –Diwali , Holi , Ganesh Chaturthi,Independence Day, EID, Christmas ,Lohri . We do this by using convolutional neural networks and fastai Library for this tasks.

Dataset formation

Dataset for the objective is made by using the scrapping Technique. Every character images available on internet is taken separately and then the put in the source folder for the use. It is then uploaded on the kaggle database for direct retrieval into collab notebook using the api commands

The FastAI library provides a lot of different [datasets](https://course.fast.ai/datasets) which can be loaded in directly, but it also provides functionality for downloading images given a file containing the URLs of these images.

In order to get the URLs, we will navigate to [Google Images](https://images.google.com/), search for our category of choice, scroll down until enough images are loaded and then execute the following javascript code from the developer console.



After we downloaded the csv files we can download the data using the download\_images method.

**Kaggle:**

Kaggle, a subsidiary of [Google LLC](https://en.wikipedia.org/wiki/Google_LLC), is an online community of [data scientists](https://en.wikipedia.org/wiki/Data_science) and [machine learning](https://en.wikipedia.org/wiki/Machine_learning) practitioners. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges*.*

It provides services for downloading the public data already present in it. We can also create our own dataset and upload on kaggle servers and can make it public for public use. It is one of the best way to retrieve data into google colab directly from kaggle with very high server speed using the api commands provided by the kaggle.

Framework(Fastai)

**Fastai** is a deep learning library which provides practitioners with high-level components that can quickly and easily provide state-of-the-art results in standard deep learning domains, and provides researchers with low-level components that can be mixed and matched to build new approaches. It aims to do both things without substantial compromises in ease of use, flexibility, or performance. This is possible thanks to a carefully layered architecture, which expresses common underlying patterns of many deep learning and data processing techniques in terms of decoupled abstractions. These abstractions can be expressed concisely and clearly by leveraging the dynamism of the underlying [**Python**](https://www.python.org/) language and the flexibility of the **[PyTorch](https://pytorch.org/)** library. fastai includes:

* A new type dispatch system for Python along with a semantic type hierarchy for tensors
* A GPU-optimized computer vision library which can be extended in pure Python
* An optimizer which refactors out the common functionality of modern optimizers into two basic pieces, allowing optimization algorithms to be implemented in 4-5 lines of code
* A novel 2-way callback system that can access any part of the data, model, or optimizer and change it at any point during training
* A new data block API

...and much more.

We have used this library to successfully create a complete deep learning course, which we were able to write more quickly than using previous approaches, and the code was more clear. The library is already in wide use in research, industry, and teaching.

Methods

## Resnet :

It is one of the most efficient ways to train our model, it is prebuilt in the fastai framework that we have used to prepare our data for the training purpose.

ResNet-50 is a convolutional neural network that is 50 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database

**ResNet-50 is a deep residual network. The “50” refers to the number of layers it has. It’s a subclass of convolutional neural networks, with ResNet most popularly used for image classification.**

## CNN:

CNN

In machine learning, Convolutional Neural Networks (CNN or ConvNet) are complex feed forward neural networks. CNNs are used for image classification and recognition because of its high accuracy. It was proposed by computer scientist Yann LeCun in the late 90s, when he was inspired from the human visual perception of recognizing things. The CNN follows a hierarchical model which works on building a network, like a funnel, and finally gives out a fully-connected layer where all the neurons are connected to each other and the output is processed.

System Requirement

We have used the Notebook Provided by the google, i.e Google Colab. It is linux based environment provided by the google free of cost. The google also provides very fast severs to download the dataset as well free GPU and TPU to boost the time required to train the working model.

Libraries Used:

1. Fastai
2. Pandas
3. openCV

Steps To be followed for installing the Above Libraries,

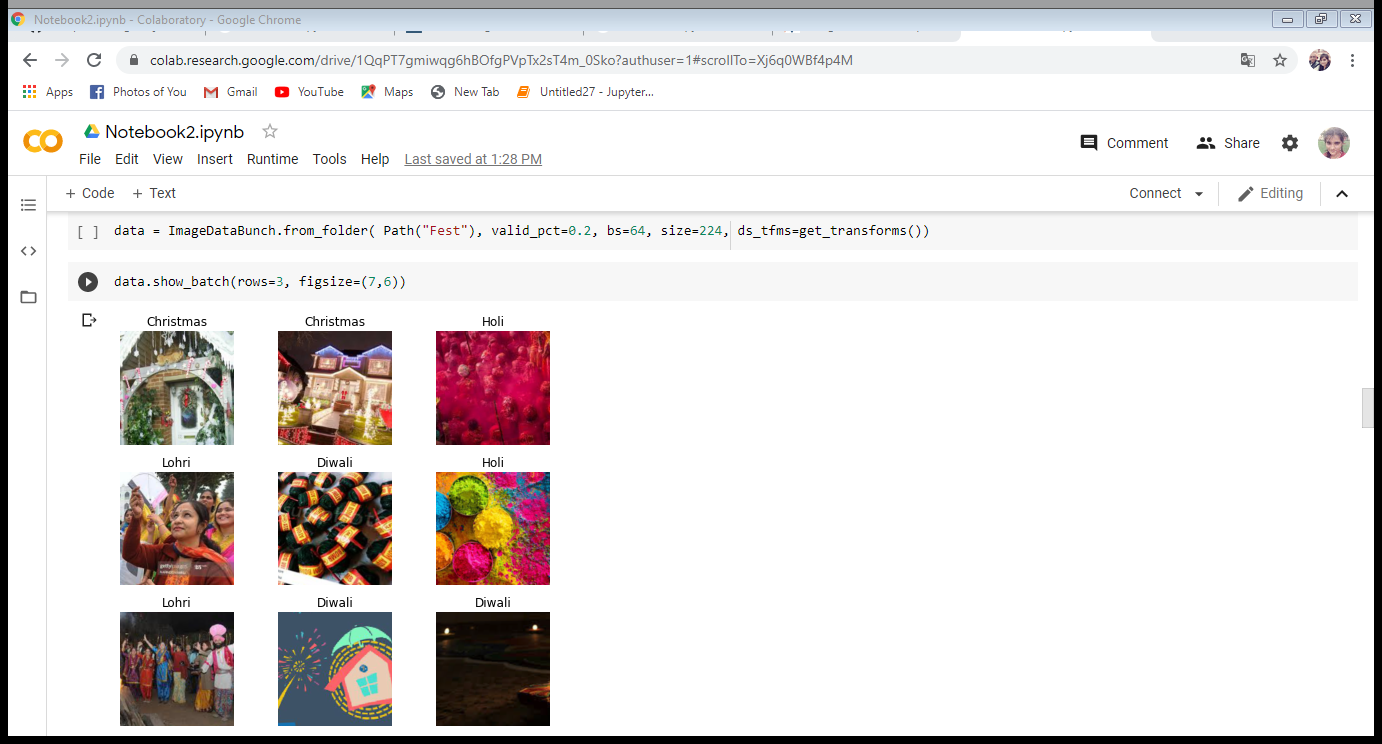
Install the following dependencies via pip:

1. !pip install fastai(‘!’ used in google colab only )

2. !pip install pandas(‘!’ used in google colab only)

3. !pip install openCV(‘!’ used in google colab notebook only)

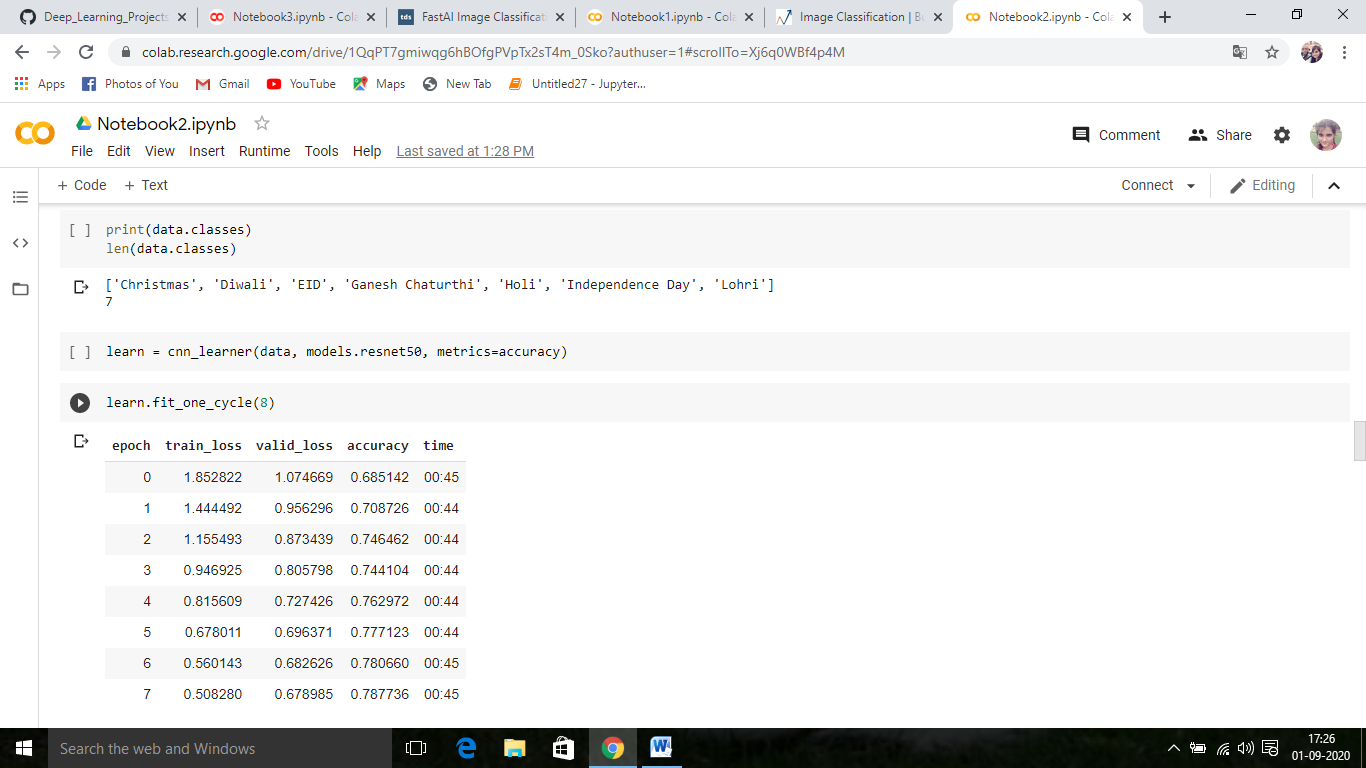
Project



FastAI has specific data objects called databunches which are needed to train a model. These databunches can be created in two main ways.

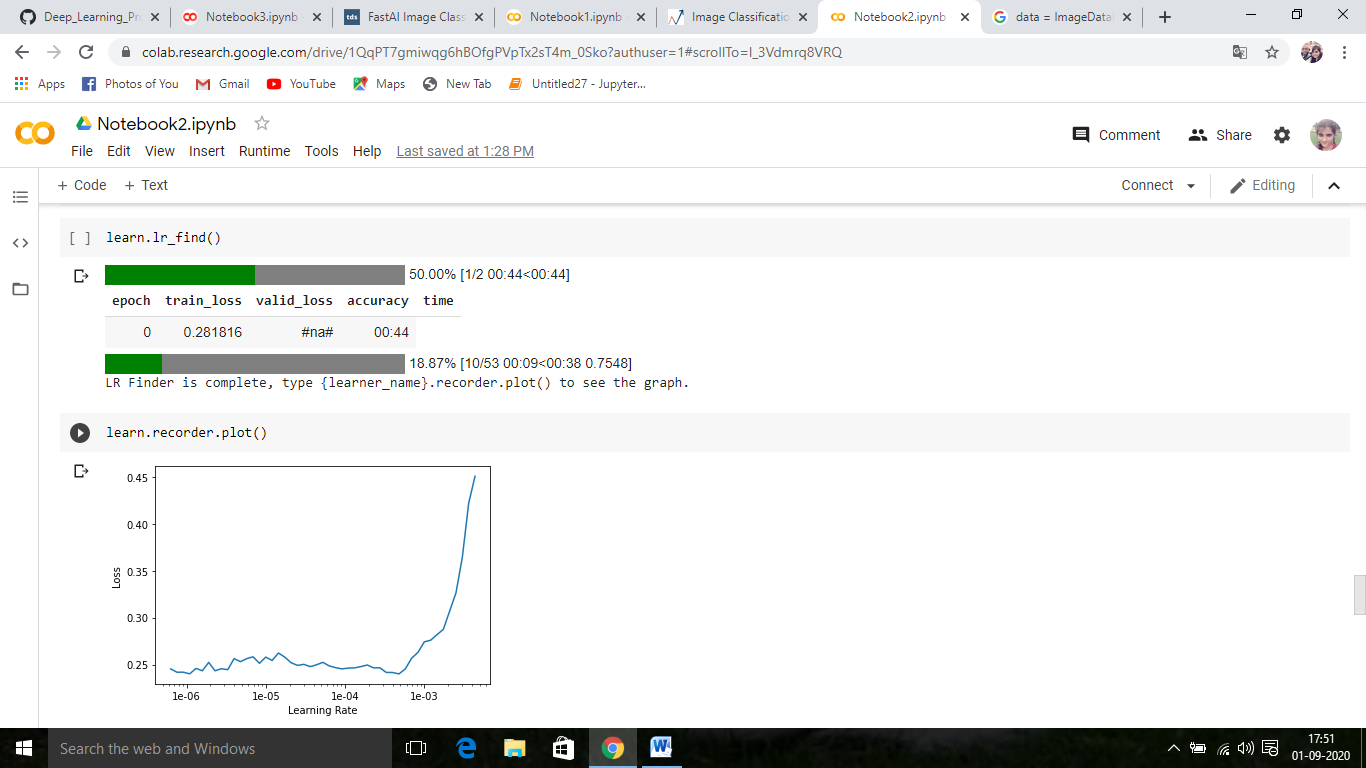
The first way is to use problem-specific methods like the ImageDataBunch.from\_folder which can be used to load data that has the following structure. And this is what we are doing.

We can show a random batch of images using the show\_batch method.

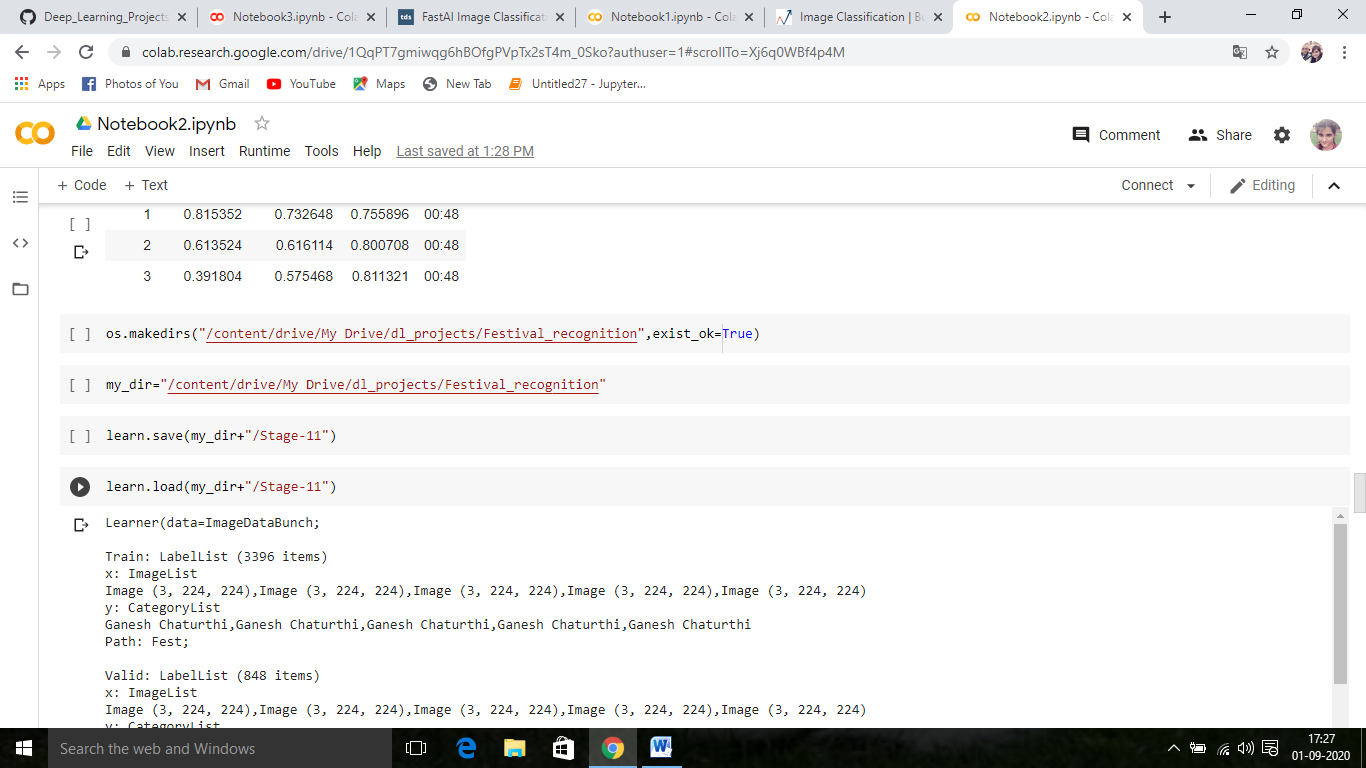


In this step, we will train the model on the training set images and validate it using, you guessed it, the validation set and get the accuracy.

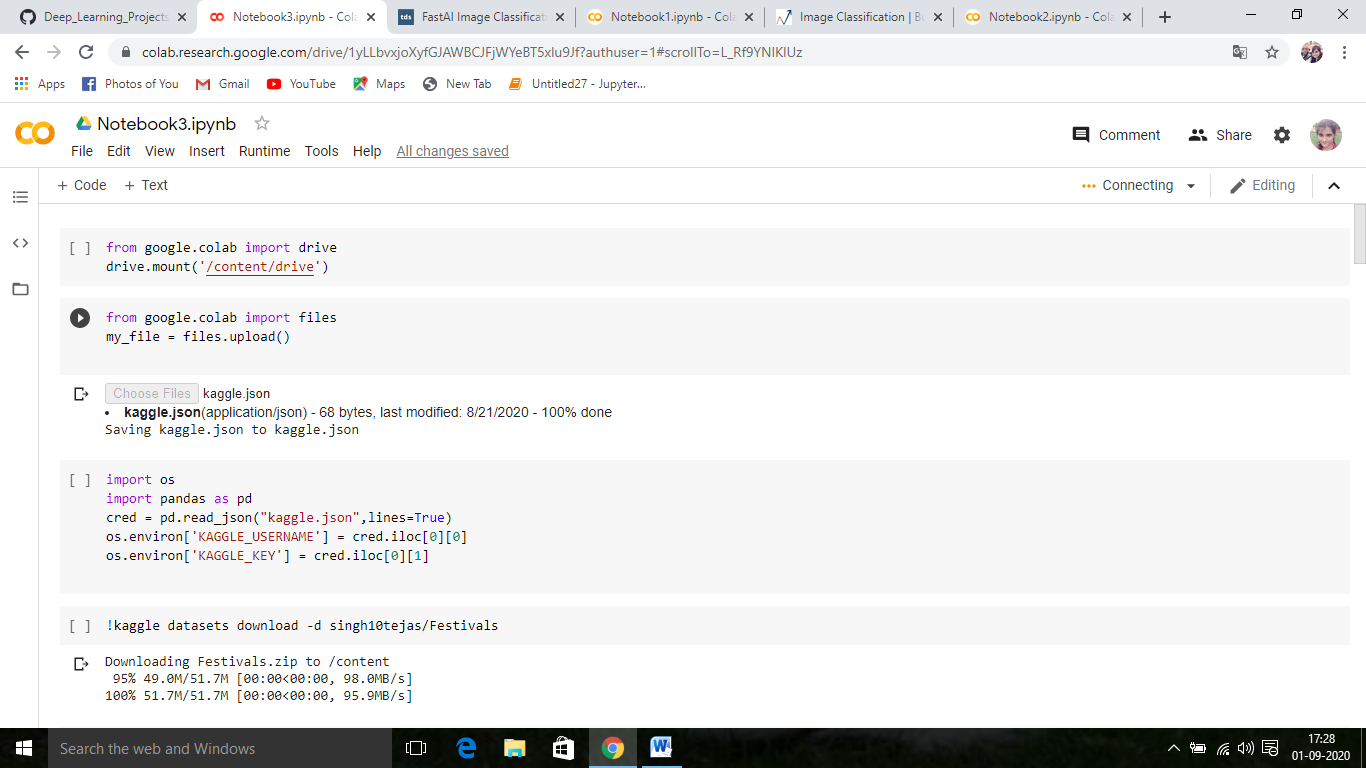
To train the layers we can use the fit or fit\_one\_cycle method. The fit method is the “normal” way of training a neural net with a constant learning rate, whilst the fit\_one\_cycle method uses something called the [**1 cycle policy**,](https://sgugger.github.io/the-1cycle-policy.html) which basically changes the learning rate over time to achieve better results.



To find the perfect learning rates we can use the lr\_find and recorder.plot methods which create a plot that relates the learning rate with the loss.



Using these lines we can save our model and load in google drive



The first line is used to mounting your Google Drive. Run those two lines of code and get the authorization code by loggin into your Google account. Then, paste the authorization code and press Enter.

**Import data into Google Colaboratory,**

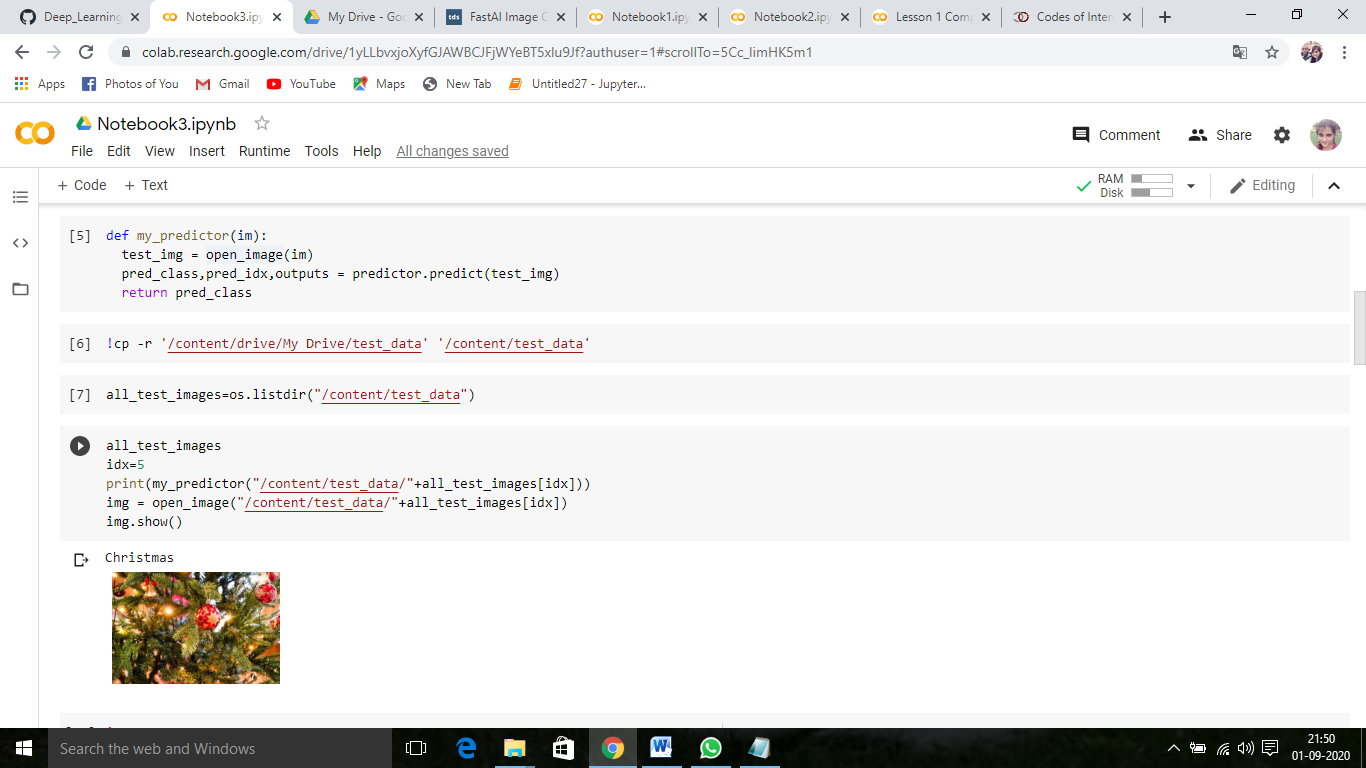
 To upload file, files module under google.colab should be imported in advance. Then use files.upload() function to upload CSV or TXT file. upload file using the upload button control Running below commands will allow us to upload data files into the colab environment

Result

# Before prediction



# After prediction



# Thank You !