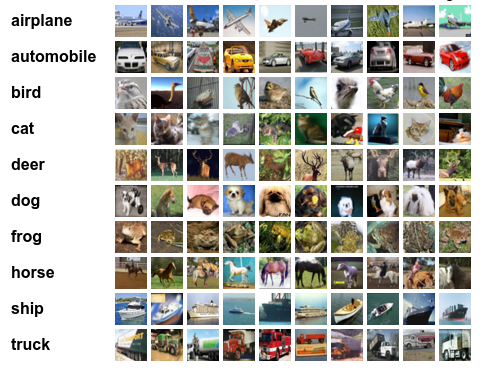
**Image Classification on CIFAR-10**

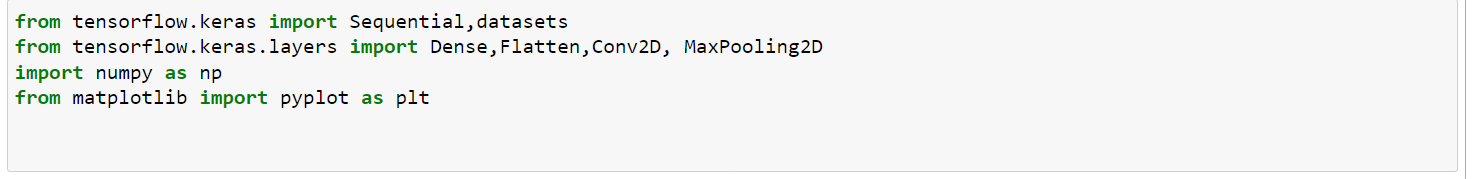


**Introduction:**

In this blog, we will build and train an image classifier CNN on the CIFAR-10 dataset using TensorFlow.

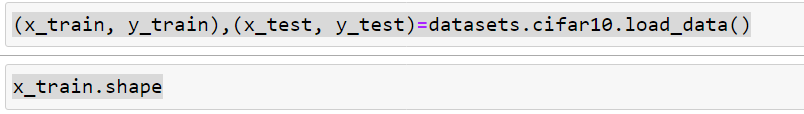
**Importing Libraries:**

We will only import the TensorFlow library along with



**Dataset**

The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. We can access this dataset directly through the tensorflow library.



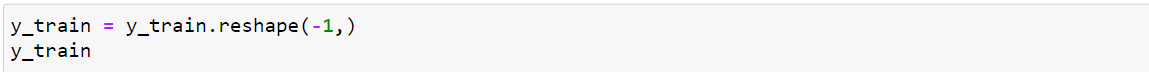


The data come already separated in training and test subsets . We will use the test data as our validation data and score the performance of the model with the help of validation accuracy.

**Pre-Processing:**

We will now pre-process our data before fitting the data in our model to reduce the problem of overfitting .

**Reshaping the data**

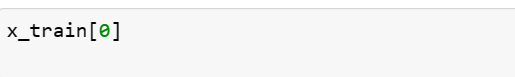




**##Observe the Data**

****

**Normalizing images:**

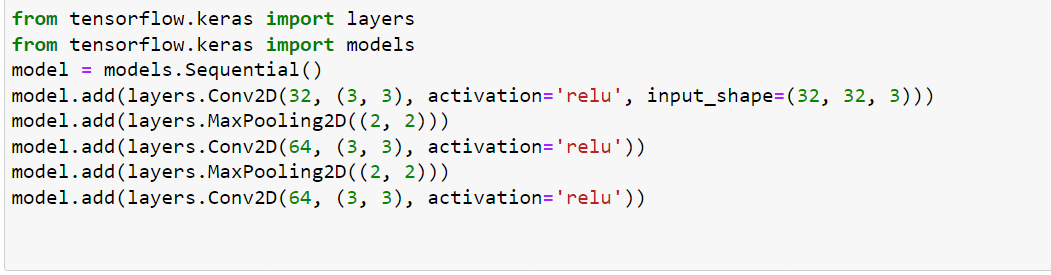




**Create The Convolutional Base:**

The below code defines the convolutional base using a common pattern: a stack of Conv2D and MaxPooling2D layers.

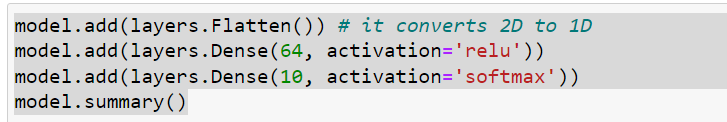
In this example, we will configure your CNN to process inputs of shape (32, 32, 3), which is the format of CIFAR images. we can do this by passing the argument input shape to first layer.



**Add Dense layers on top**

To complete the model, we will feed the last output tensor from the convolutional base (of shape (4, 4, 64)) into one or more Dense layers to perform classification. Dense layers take vectors as input (which are 1D), while the current output is a 3D tensor.

First, We will flatten (or unroll) the 3D output to 1D, then add one or more Dense layers on top. CIFAR has 10 output classes, so We use a final Dense layer with 10 outputs.



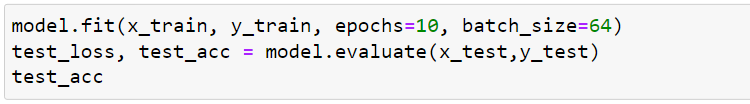
**Compiling and Fitting the Model:**

Since our data is pre-processed and our model is also ready, it’s time to compile the model.

We will use the Adam Optimizer and We will be using **loss = “sparse\_categorical\_crossentropy”**and accuracy as our metric for scoring our results.



After successfully compiling our model, we will now call fit function



After training the model for 10 epochs we get an accuracy of about  71%

**Conclusion**

We have successfully trained the CIFAR-10 dataset and have achieved desent accuracy .