**TensorFlow**: is a Python library for fast numerical computing created and released by Google.

It is a foundation library that can be used to create Deep Learning models directly

or by using wrapper libraries that simplify the process built on top of TensorFlow.

It was created and is maintained by Google and released under the Apache 2.0 open source license.

The API is nominally for the Python programming language,

although there is access to the underlying C++ API.

Unlike other numerical libraries intended for use in Deep Learning like Theano,

TensorFlow was designed for use both in research and development and in production systems,

not least RankBrain in Google search and the fun DeepDream project

**Keras:** Keras is an open-source software library that provides a Python interface

for artificial neural networks.

Keras acts as an interface for the TensorFlow library.

The Python Imaging Library adds image processing capabilities

to your Python interpreter.

This library provides extensive file format support,

an efficient internal representation,

and fairly powerful image processing capabilities.

The core image library is designed for fast access to data stored

in a few basic pixel formats. It should provide a solid foundation for

a general image processing tool.

**Sklearn train\_test\_split**

**train\_test\_split** is a function in **Sklearn** model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, **Sklearn train\_test\_split** will make random partitions for the two subsets.

**to\_categorical**()

Using the method **to\_categorical**() , a numpy array (or) a vector which has integers that represent different categories, can be converted into a numpy array (or) a matrix which has binary values and has columns equal to the number of categories in the data. ... num\_classes: Total number of classes.

**Keras.sequence**

A **Sequential model** is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor. A **Sequential model** is not appropriate when: Your **model** has multiple inputs or multiple outputs. Any of your layers has multiple inputs or multiple outputs.

**Ephocs:**

Ephocs is the number of times you give the data to the neural network

If we give less value to the ephocs then it shows the graph in a straight line which is underfitted. Whereas ephocs with higher value shows the exact curves in the graph which is Overfitted

**Overfitting**: Good performance on the training data, poor generliazation to other data.

**Underfitting**: Poor performance on the training data and poor generalization to other data.