**Aim**

The aim of this project is to classify a landmark based on the image uploaded to the server through the android application using neural networks.

**Server End**

On the backend we have a Django server running which takes in a base64 encoded string and outputs the landmark details. The internal workings of the server are as follows:

* Initially there are images of various landmarks kept on the server. All these images are resized into a 64 X 64 px dimension with the help of a python script. These form our training dataset.
* While compiling our model we first traverse over our training dataset and convert all the images into numpy arrays and assign labels to each image. Furthermore we make these labels categorical using the LabelEncoder module in Python.
* Once the above two steps are finished we can create our model. The neural network we design is shown pictorially as below :
* Once the model is constructed we fit our existing training dataset into this model with the *model.fit()* function. Upon success we save the model in a .h5 file on the server.
* In our server for each image we load this model from the saved .h5 file and we make predictions with the *model.predict()* function available to us and deliver the output as a JSON string from our RESTful API.

**Application**

The android application contains the following features:

* A image can be captured from the mobile through the built in camera application. The selected image will then show up in the application and you can verify if it is suitable for uploading to the server.
* You can also select an already existing image from the mobile gallery.
* Once an image is selected you have the option to upload it to the server. The image is encoded with base64 encoding and is then sent as a POST request to the Django server.
* Once image is successfully uploaded to the server we get the classification output as a JSON string and read out to the user via the Text to Speech service in the Android SDK.