**README**

**Introduction:**

Deep learning, a subset of machine learning has led to the development of Convolutional Neural Networks for predicting objects in an image and classifying the image on the basis of these identified similar patterns. This assignment demonstrates image classification over CIFAR-100 dataset using a CNN that results in an accuracy of around 73% over the test dataset.

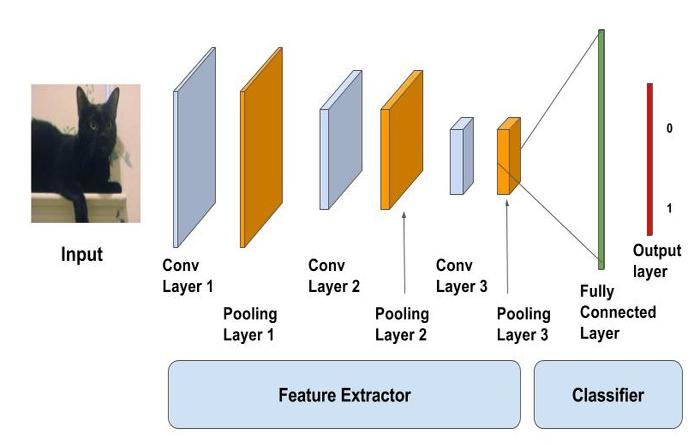
What is Image Classification?

Image Classification is an approach that tries to analyze an image as a whole. The objective behind the image classification task is to assign a label to an image on the basis of the pattern identified from it. While object detection deals with classification as well as localization as there can be multiple objects involved, image classification deals with a single object images only [1]. Steps involved are:

1. Flattening of the input dimension of the image to 1D that is width\_pixel\*height\_pixel.
2. Next step is normalization of the image pixel values that is dividing the pixel value by 255.
3. This step is followed by one-hot encoding over the categorical column.
4. Next step involves building of the model architecture with dense layers.
5. And the final step involves training the model and then making predictions [5].

**Approach:**

Convolution neural networks have convolutional layers, ReLU layers, pooling layers and a fully connected layer.



The convolutional layer applies convolution operation on the input and passes information to the following layer. Pooling layers do the task of combining output of a group of neurons into a single neuron in the following layer. Whereas the fully connected layers connect each neuron in the current layer to each neuron in the following layer [5]. A classic CNN architecture would look like

****

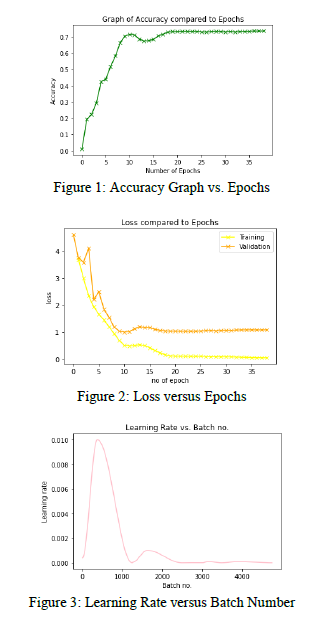
The CNN used has input, output and boolean value pool. Kernel size used is 3\*3. Normalization and ReLU activation function is used. The pooling layer used takes maximum value from a 2\*2 box. The model has 12 layers that include 5 convolution layer and 6 residual layers. Hyper parameters used are learning rate, gradient clipping, epochs, and weight decay and optimization functions. The learning rate is kept varying with number of epochs. At beginning it is set at peak and gradually decreased. Learning rate is set to 30% of the number of epochs. By training for 10 epochs 70% accuracy was achieved but training for 10 more epochs accuracy achieved was 72.5%.

**Dataset Details:**

The dataset used is named CIFAR-100 [4]. This dataset consists of tiny images of objects such as apple, bed, baby, bird, cat, deer, dog, ship, truck etc. There are 100 classes each containing 600 images present in this dataset. Each image has 3 RGB colour channels and a pixel dimension 32\*32 which makes the overall size per input equal to 32\*32\*32 that is 3072. For each class there are 500 training images and 100 testing images. Hence in all, the dataset has 50K training images and 10K testing images. These images were collected by hey Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. In this dataset, all he 100 classes are divided into 20 groups of superclasses. Hence, each image has a “fine” label associated with the class it belongs to and a “coarse” label associated with the “superclass” it belongs to.

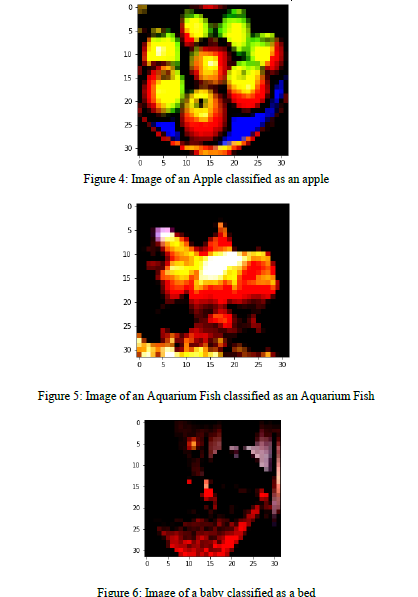
**Results:**

The accuracy of the CNN used for image classification over CIFAR-100 is 73%. Figure 1 shows the graph of accuracy versus number of epochs. Figure 2 shows the training and validation loss graph. Figure 3 shows the learning rate versus batch number.



Sample test image predictions for three test cases are.

Figure 4 has object apple and got classified as an apple. Figure 5 has object aquarium fish and got classified as an aquarium fish and the third image is that of a baby and got classified as a bed which is shown in figure 6.

****

**Installation Guidelines and Platform Details:**

**Dataset was downloaded from:** <https://www.kaggle.com/c/ml2016-7-cifar-100> and uploaded over drive. Mounting the drive, the entire code has been done over Google Colab. The comments in the code explain the steps in detail.

**References:**

1. Bakhshi, A., Chalup, S. and Noman, N., 2020. Fast evolution of cnn architecture for image classification. In *Deep Neural Evolution* (pp. 209-229). Springer, Singapore.
2. Basha, S.S., Dubey, S.R., Pulabaigari, V. and Mukherjee, S., 2020. Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, *378*, pp.112-119.
3. Sharma, N., Jain, V. and Mishra, A., 2018. An analysis of convolutional neural networks for image classification. *Procedia computer science*, *132*, pp.377-384.
4. https://www.kaggle.com/c/ml2016-7-cifar-100
5. https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb
6. https://medium.com/jovianml/image-classification-with-cifar100-deep-learning-using-pytorch-9d9211a696e