Image classification in cortex-M4/M7

By Mohd Zahid Faiz
(MT2019514)

Image classification using CNN’s

-Using the Convolution Neural Network(CNN) the image classification is performed.

-CNN are a special type of neural network that classifies the image by classifying each image pixels.

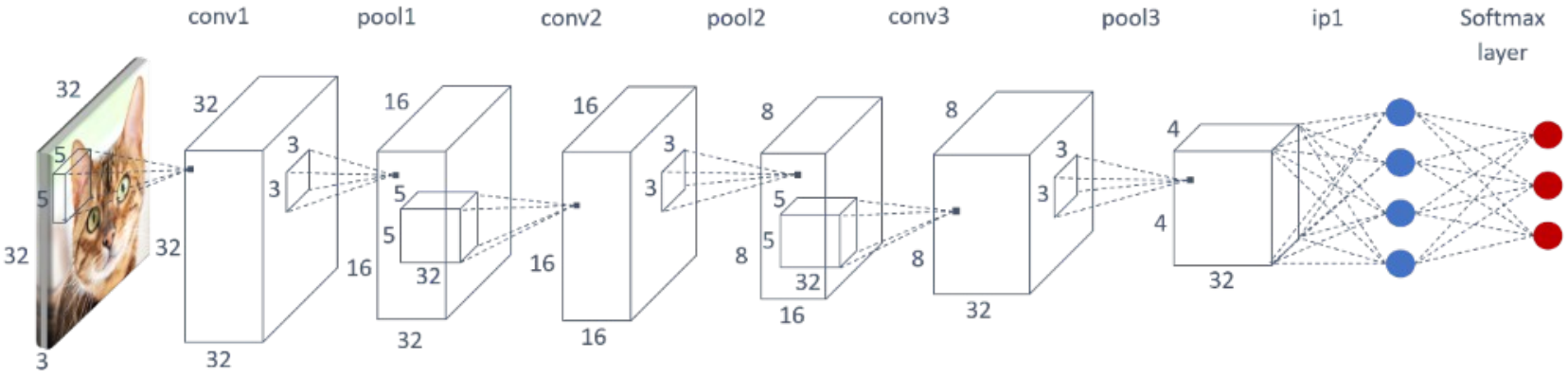


Image classification in CIFAR-10

-Cifar-10 dataset consists of 10 classes with 6000 images per class.

-Each image is of 32x32 size with 3 channels RGB and with 1byte per pixel resulting in 3.1Kb image.

-The pre processing is done with the help of a seperate script, where the image is resized to 32x32 and flattened as

Image = {R1,G1,B1, R2,G2,B2,.....}
The image is flattened and each pixel is represented by 3 values.This is stored in a file arm_nnexamples_cifar10_inputs.h

-The classification output will be scores for 10 classes. Higher the score higher the probability of the class.

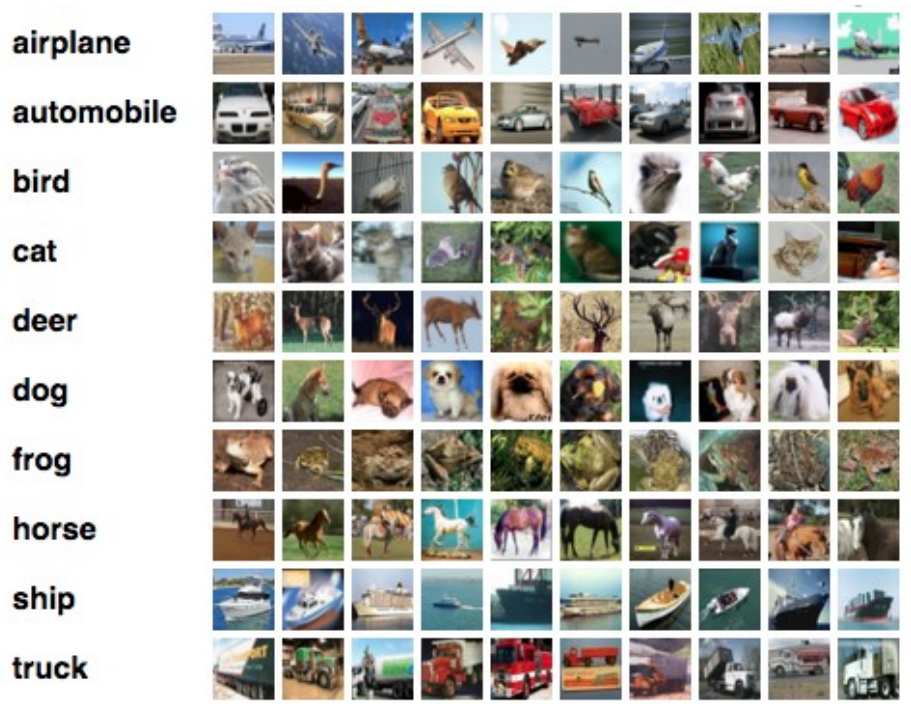


Image classification in CIFAR-10

-The model architechture used here is taken from the repository https://github.com/BVLC/caffe.

-The model is trained on CIFAR-10 image classifiaction and the weights are saved as arm_nnexamples_cifar10_weights.h. Similar for the parameters also are saved as arm_nnexamples_cifar10_parameter.h. Both of these files require a space of 32.3 KB to store weights, 40 KB for parameters.

-The size contrainst is the most important aspect if we are deploying that on an ARM processor.

-One of the first challange I faced here is the limitation of RAM/ROM in the simulator keil-MDK. Keil supports only 20Kb or below space for the free versions and the non availability of the board also was not helping.

Selecting a board for simulation

-The target processor used is Cortex-M4/Cortex-M7 and for that the board selected is from Stmicroelectronics named STM32F407VGT6.

-STM32F407VGT6 microcontroller features a 32-bit ARM® Cortex® -M4 with FPU core, 1-Mbyte Flash memory, and 192-Kbyte RAM at a maximum power consumption capped at 465mW.

-This project is configured to run both on simulator keil-MDK as well as in STM32F407VGT6.

Assignment on neural network

-The Neural network assignment done before is the basic building block of CNN's.The mathematical implimentation is same with some minor changes in the formula.

For understanding the neural networks, consider the below architecture, here X1 , X2 are the inputs and b is the bias term, The weighted linear combination that is the term

$$z = (w1.X1 + w2.X2 + b)$$

This regression is passed through an activation function,

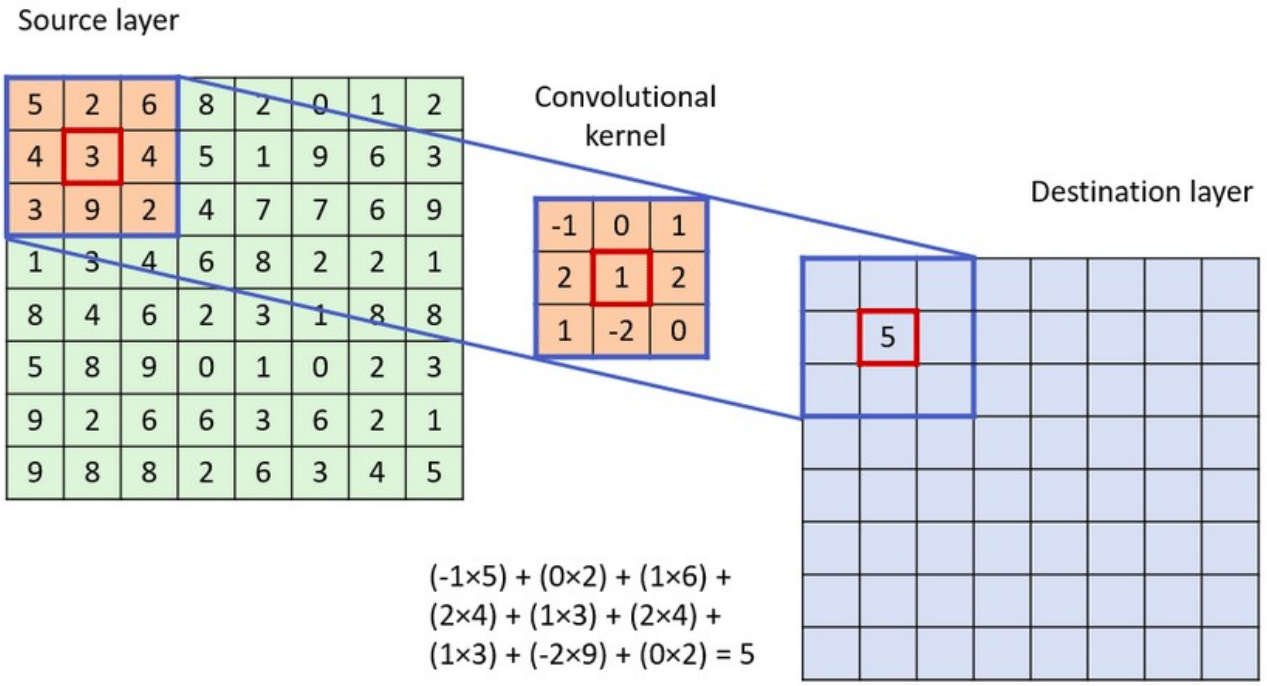
$$Y = f(w1.X1 + w2.X2 + b)$$
$$= f(z)$$

The activation function could be sigmoid or tanh or Relu.

Similarly for the CNN's the maths implimented in each layer is shown in the coming slides.

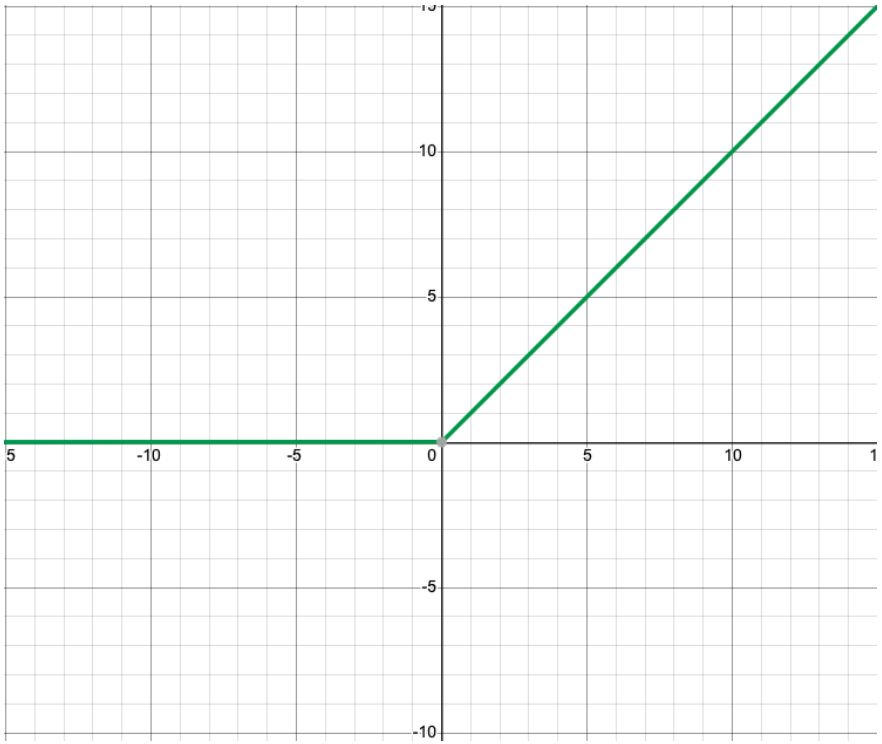
AND			
A=1	B=0	C=0	Result Y=0
A=1	B=0	C=1	Result Y=0
A=1	B=1	C=0	Result Y=0
A=1	B=1	C=1	Result Y=1
OR			
A=1	B=0	C=0	Result Y=0
A=1	B=0	C=1	Result Y=1
A=1	B=1	C=0	Result Y=1
A=1	B=1	C=1	Result Y=1
NOT			
A=1	B=0	C=0	Result Y=1
A=1	B=0	C=1	Result Y=1
A=1	B=1	C=0	Result Y=0
A=1	B=1	C=1	Result Y=0
NAND			
A=1	B=0	C=0	Result Y=1
A=1	B=0	C=1	Result Y=1
A=1	B=1	C=0	Result Y=1
A=1	B=1	C=1	Result Y=0
NOR			
A=1	B=0	C=0	Result Y=1
A=1	B=0	C=1	Result Y=0
A=1	B=1	C=0	Result Y=0
A=1	B=1	C=1	Result Y=0

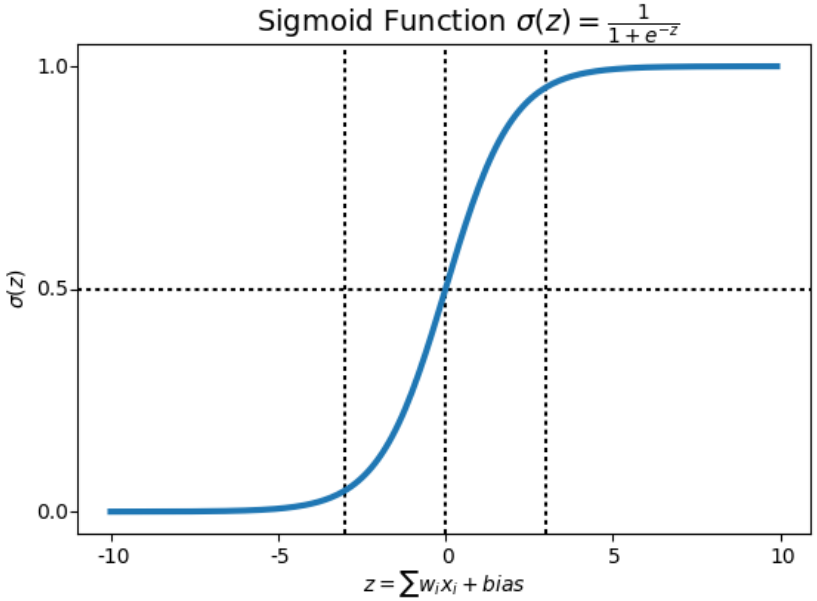
Convolution Block



Single layer of convolution with Relu activation.

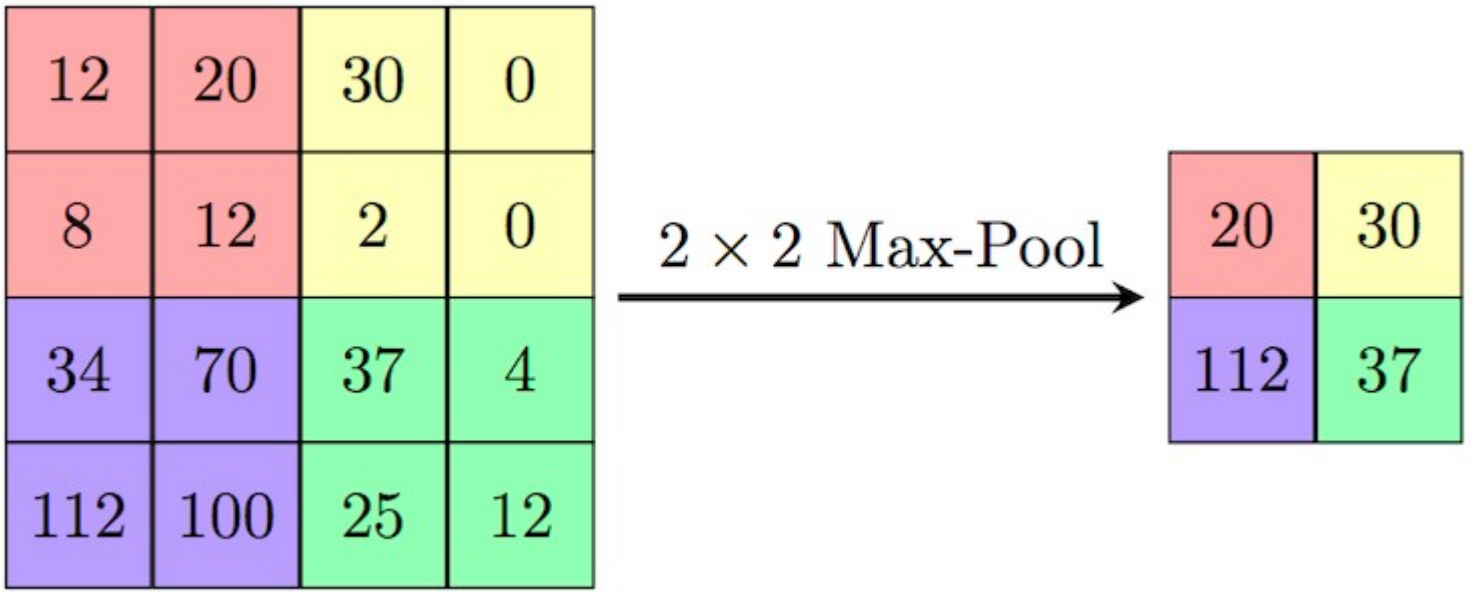
Activation functions





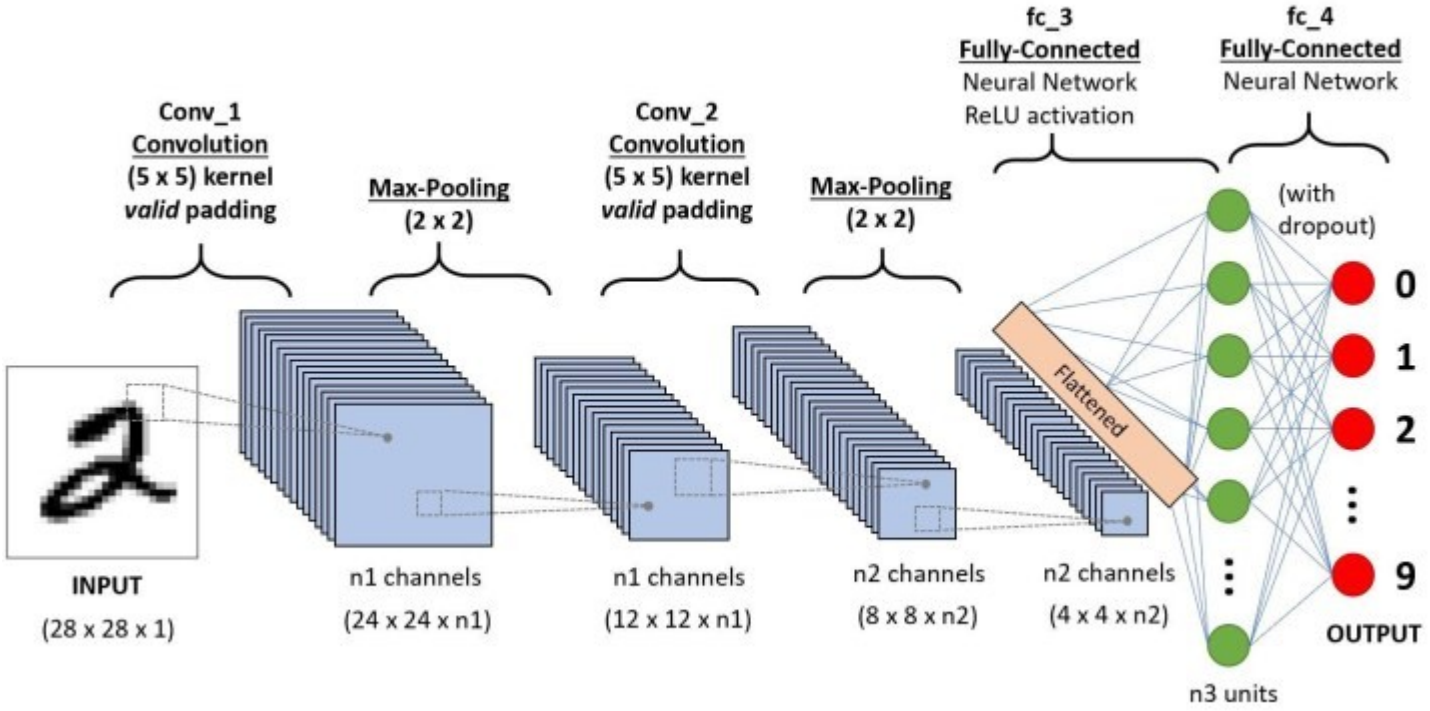
Relu activation layer

Maxpooling



Maxpooling layer working.

Final architechture



- Fully Connected layer (FC) is similar to the NN and the last layer is the sigmoid activation function.

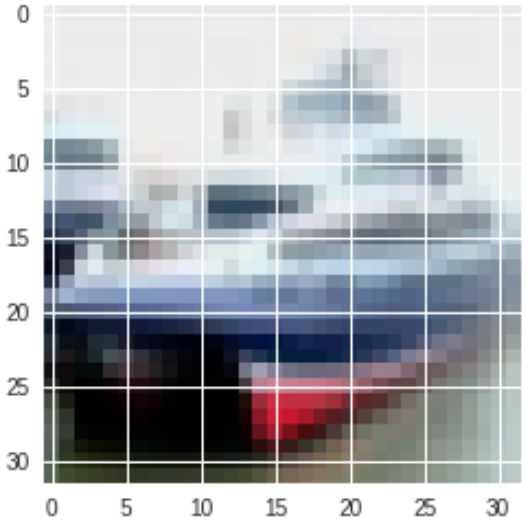
CMSIS library used

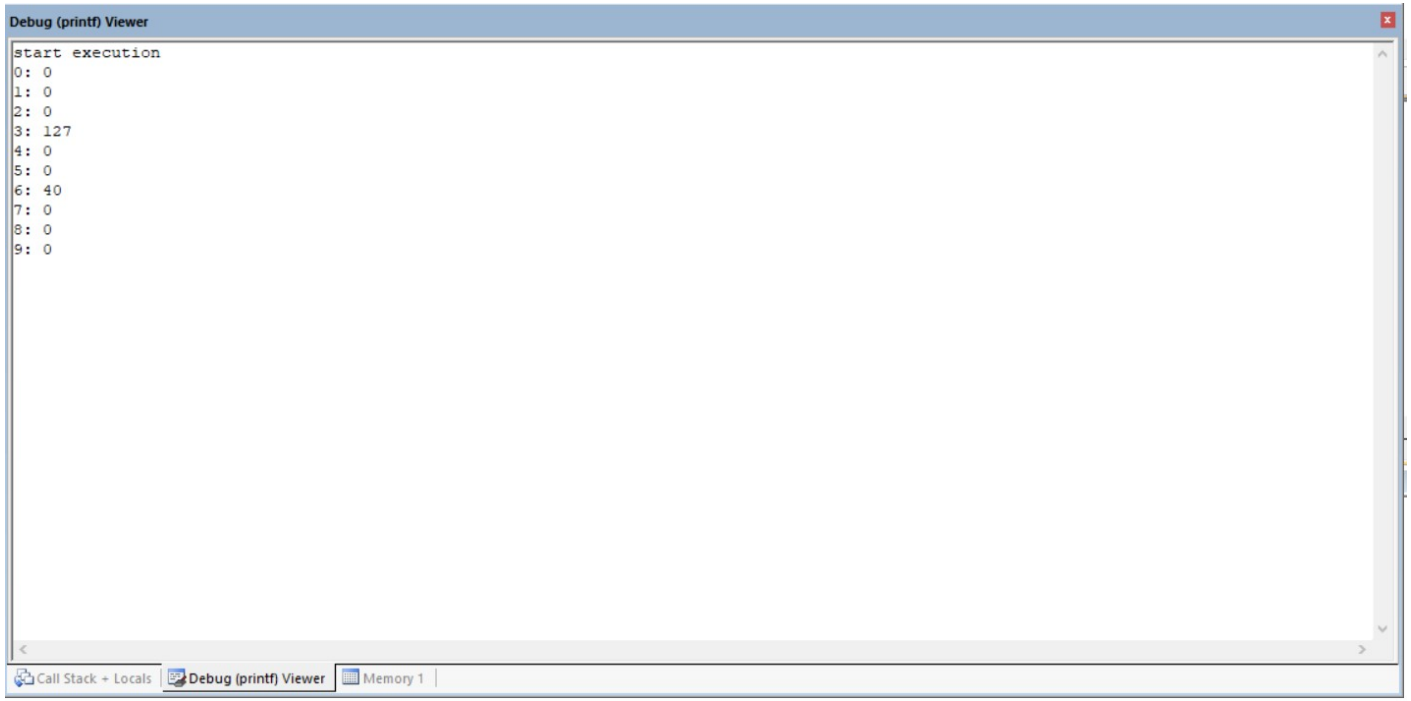
-CMSIS NN software library, a collection of efficient neural network kernels developed to maximize the performance and minimize the memory footprint of neural networks on Cortex-M processor cores.

-These are following functions used,

- arm_convolve_HWC_q7_RGB()
- arm_convolve_HWC_q7_fast()
- arm_relu_q7()
- arm_maxpool_q7_HWC()
- arm_avepool_q7_HWC()
- arm_fully_connected_q7_opt()
- arm_fully_connected_q7()

Result





'airplane':0,'automobile':1,'bird':2,'cat':3,'deer':4,'dog':5,'frog':6,'horse':7,
'ship':8,'truck':9