In the hidden layers Relu activation function is applied to increase the non-linearity in images and In the final dense layer **softmax activation is applied because it** allows the neural network to run a multi-class function (dog, cat, car etc.). I used a 3x3 filter starting with 32 filters and zero paddings and a stride of 1.

First structure

First the conv2d_1 using relu had an input volume of 32*32*3 and output shape of 32*32*32 it fired to another conv2d_2 using relu had an output shape of 32* 32* 32 then max_pooling2d_1 layer was applied with 16, 16, 32 output shape resulting in 8192 after flattening it then 1024 after applying a dropout of .2 and a dense layer using Relu activation function and finally an output shape of 10 after adding a dropout of .2 and applying a dense layer using softmax activation function (shown in fig 1)

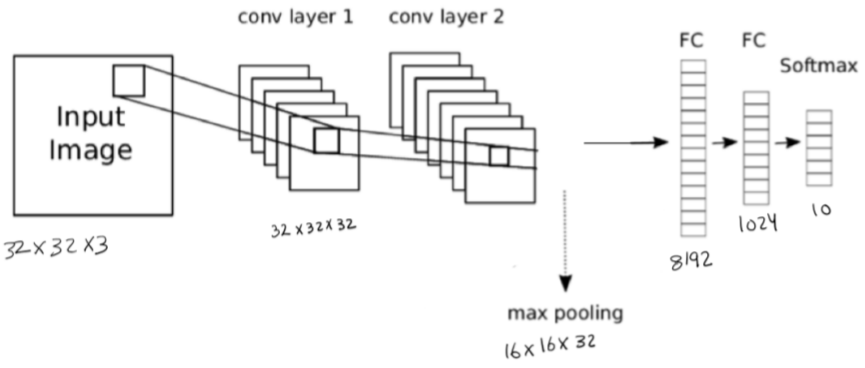
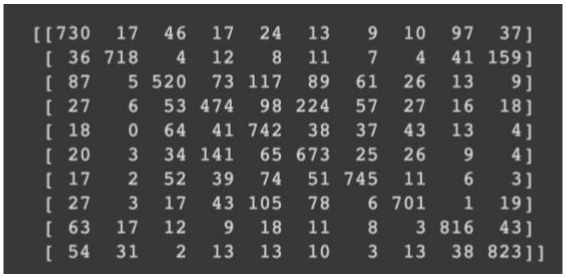


Fig 1

Confusion matrix:



```
[[730  17  46  17  24  13   9  10  97  37]
 [ 36 718   4  12   8  11   7   4  41 159]
 [ 87   5 520  73 117  89  61  26  13   9]
 [ 27   6  53 474  98 224  57  27  16  18]
 [ 18   0  64  41 742  38  37  43  13   4]
 [ 20   3  34 141  65 673  25  26   9   4]
 [ 17   2  52  39  74  51 745  11   6   3]
 [ 27   3  17  43 105  78   6 701   1  19]
 [ 63  17  12   9  18  11   8   3 816  43]
 [ 54  31   2  13  13  10   3  13  38 823]]
```

recall: .69
precision: .70

Second structure

First the conv2d_1using Relu had an input volume of 32*32*3 and output shape of 32*32*32 it fired to another conv2d_2 using Relu that had an output shape of 32* 32* 32 then max_pooling2d_1 layer was applied resulted in 16* 16* 32 output shape. A conv2d_3 layer with Relu was added results 16* 16* 64 Output shape. And conv2d_4 using outputs results a 16* 16* 64 output shape. Then max_pooling2d_2 outputs 8*8* 64 resulting in 4096 after flattening it then 1024 after applying a drop out of .2 and a dense layer using Relu activation function and finally an output shape of 10 after adding a dropout of .2 and applying a dense layer using softmax activation function (shown in fig 2)

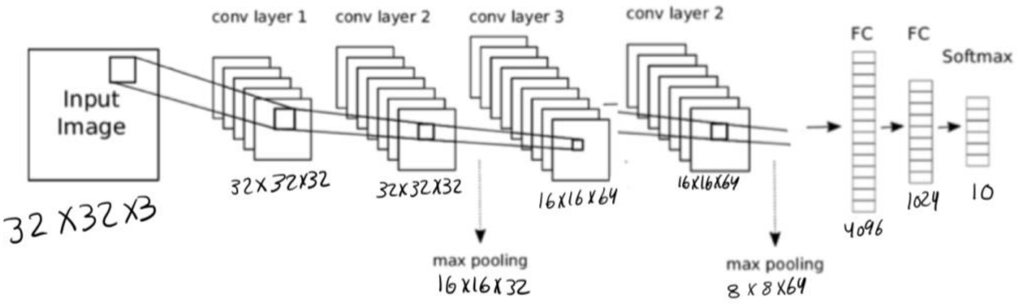
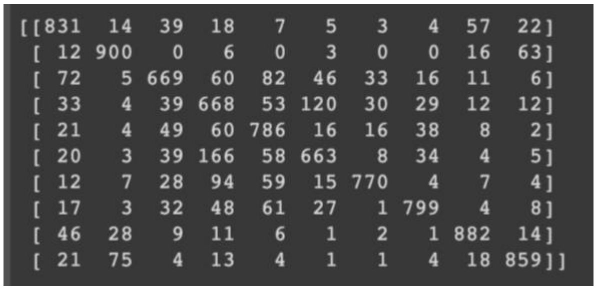


Fig 2

Confusion matrix:


```
[[831  14  39  18   7   5   3   4  57  22]
 [ 12 900   0   6   0   3   0   0  16  63]
 [ 72   5 669  60  82  46  33  16  11   6]
 [ 33   4  39 668  53 120  30  29  12  12]
 [ 21   4  49  60 786  16  16  38   8   2]
 [ 20   3  39 166  58 663   8  34   4   5]
 [ 12   7  28  94  59  15 770   4   7   4]
 [ 17   3  32  48  61  27   1 799   4   8]
 [ 46  28   9  11   6   1   2   1 882  14]
 [ 21  75   4  13   4   1   1   4  18 859]]
```

recall: .78
precision: .79

Third structure

First the conv2d_1using Relu had an input volume of 32*32*3 and output shape of 32*32*32 it fired to a max_pooling2d_1 that had an output shape of 16* 16* 32 then conv2d_2 using relu results in 16*16* 64 output shape after that a max_pooling2d_2 results a 8* 8*64 output shape. 512 output shape after applying a drop out of .5 and a dense layer using Relu activation function and finally an output shape of 10 after applying a dense layer using softmax activation function (shown in fig 3)

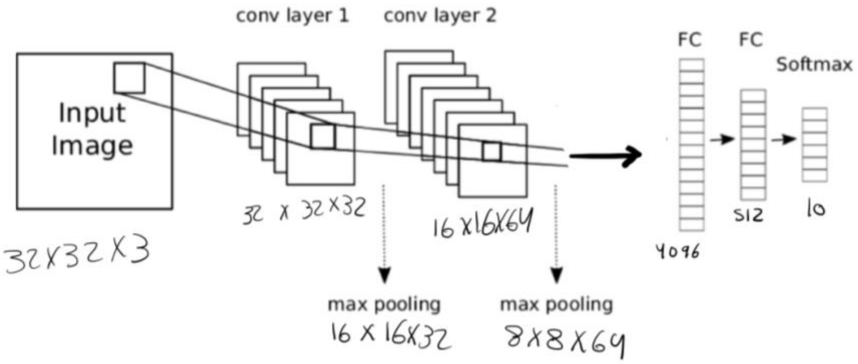
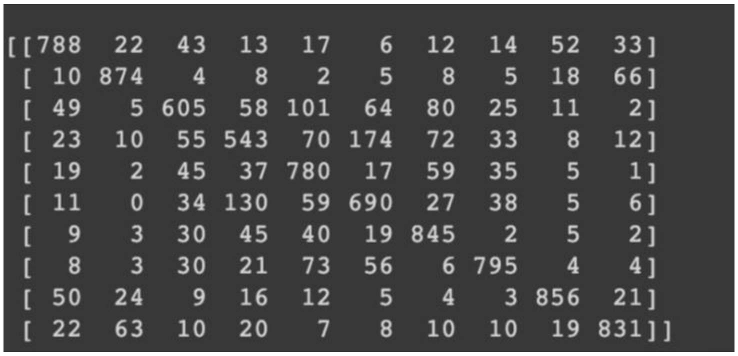


Fig 3

Confusion matrix:



```
[[788  22  43  13  17   6  12  14  52  33]
 [ 10 874   4   8   2   5   8   5  18  66]
 [ 49   5 605  58 101  64  80  25  11   2]
 [ 23  10  55 543  70 174  72  33   8  12]
 [ 19   2  45  37 780  17  59  35   5   1]
 [ 11   0  34 130  59 690  27  38   5   6]
 [  9   3  30  45  40  19 845   2   5   2]
 [  8   3  30  21  73  56   6 795   4   4]
 [ 50  24   9  16  12   5   4   3 856  21]
 [ 22  63  10  20   7   8  10  10  19 831]]
```

recall: .76
precision: .76

Hence the second structure outperformed the first and the third but all of them suffered from overfitting so in future work regularization will be considered.