☑ checked.

☐ unchecked.

☒ not done.

☑ We have read all the instruction carefully and followed them to our best ability.

☑ We have written the name, roll no in report.

☑ Run sanity_check.sh.

☑ We will be submitting only single submission on behalf of our team.

☑ We have not included unnecessary text, pages, logos in the assignment.

☑ We have not used any high level APIs(Keras, Estimators for e.g.).

☑ We have not copied anything for this assignment.

1. Observations and Inferences.
   (a) Configuration and training details of your best performing model.

   Solution: Convolution layers:
   All convolution layers is set to batch normalize is True. conv1 - - conv2 pool1 - - conv3 - - conv4 - - pool2 - - conv5 - - conv6 - - pool3 - - fullyconnected layer and then dropout.
   Dropout is after flat layer with keep probability value 0.75.
   Training hyperparameters:
   lr : 0.0007 batchsize : 128 init : Xavier epochs : 27 DataAugment : 1

   (b) A plot of the learning curve showing iterations on the x-axis and negative log likelihood over labels on the y-axis.A single plot showing both the training loss and the validation loss.

   Solution:

(c) The performance on the test data of the model that performs best on the validation data.

Solution: This is our performance on test data that performs best on validation data. Score = 0.48109

(d) The parameter setting which gave you the best results.

Solution: lr : 0.0007 batchsize : 128 init : Xavier epochs : 27 DataAugment : 1

(e) Dimensions of the input and output at each layer

Solution: Convolutional Layer 1 : Inputs = 64 * 64 * 3
Filter size = 5 * 5 * 3 and 32 such filters. Output = 32 * 64 * 64
Convolutional Layer 2 : Inputs = 32 * 64 * 64
Filter size = 32 * 5 * 5 and 32 such filters. Output = 32 * 64 * 64
Pool Layer 1 : Inputs = 32 * 64 * 64 Output = 32 * 32 * 32
Convolutional Layer 3 : Inputs = 32*32*32
Filter size = 32 * 3 * 3 and 64 such filters. Output = 64 * 32 * 32
Convolutional Layer 4 : Inputs = 64 * 32 * 32
Filter size = 64 * 3 * 3 and 64 such filters. Output = 64 * 32 * 32
Pool Layer 2 : Inputs = 64 * 32 * 32 Output = 64 * 16 * 16
Convolutional Layer 5 : Inputs = 64 * 16 * 16
Filter size = 64 * 3 * 3 and 64 such filters. Output = 64 * 16 * 16
Convolutional Layer 6 (Padding Valid) : Inputs = 64 * 16 * 16

Filter size = 64 * 3 * 3 and 128 such filters. Output = 128 * 14 * 14
Pool Layer 2 : Inputs = 128 * 14 * 14. Output = 128 * 7 * 7
FC1: fully connected layer with Inputs = 6272
Output = 256 (i.e, number of neurons is 256)
SOFTMAX: softmax layer for classification. Inputs = 256 Output = 20

(f) Exactly how many parameters does your network have? How many of these are in the fully connected layers and how many are in the convolutional layers?

Solution: Convolutional Layer 1 : Parameters = 5 * 5 * 32 + 32 = 832
Convolutional Layer 2 : Parameters = 5 * 5 * 32 + 32 = 832
Pool 1 : Parameters = 0
Convolutional Layer 3 : Parameters = 3 * 3 * 64 + 64 = 640
Convolutional Layer 4 : Parameters = 3 * 3 * 64 + 64 = 640
Pool 2 : Parameters = 0
Convolutional Layer 5 : Parameters = 3 * 3 * 64 + 64 = 640
Convolutional Layer 6 : Parameters = 3 * 3 * 128 + 128 = 1280
Pool 3 : Parameters = 0
FC1 : Parameters = 6272 * 256 + 256 = 1605888
SOFTMAX Layer : Parameters = 256 * 20 + 20 = 5140
Total number of parameters = 1615892

(g) Exactly how many neurons does your network have? How many of these are in the fully connected layers and how many are in the convolutional layers?

Solution:
Input : 12288 Convolutional Layer 1 : Neurons = 5 * 5 * 32 = 800
Convolutional Layer 2 : Neurons = 5 * 5 * 32 = 800
Pool 1 : Neurons = 0
Convolutional Layer 3 : Neurons = 3 * 3 * 64 = 576
Convolutional Layer 4 : Neurons = 3 * 3 * 64 = 576
Pool 2 : Neurons = 0
Convolutional Layer 5 : Neurons = 3 * 3 * 64 = 576
Convolutional Layer 6 : Neurons = 3 * 3 * 128 = 1092
Pool 3 : Neurons = 0
FC1 : Neurons = 256 = 256
SOFTMAX Layer : Neurons = 20 = 20
Total number of Neurons = 4696

(h) What was the effect of using batch normalization?

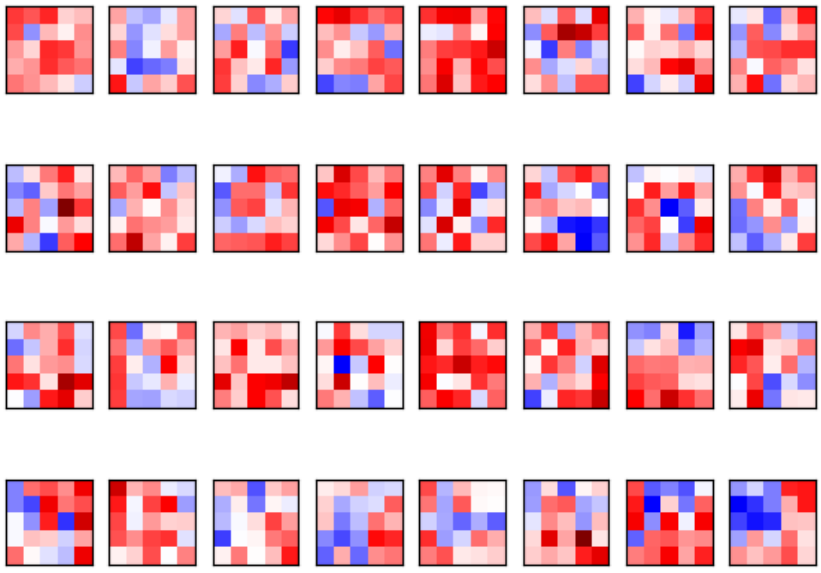
Solution:

Batch normalization is Off.

Batch normalization is ON. The effect of using batch normalization testing accuracy increases.

(i) Plot all the 32 layer-1 (Conv1)
lters in an 4 by 8 grid. Do you observe any interesting patterns?

Solution:

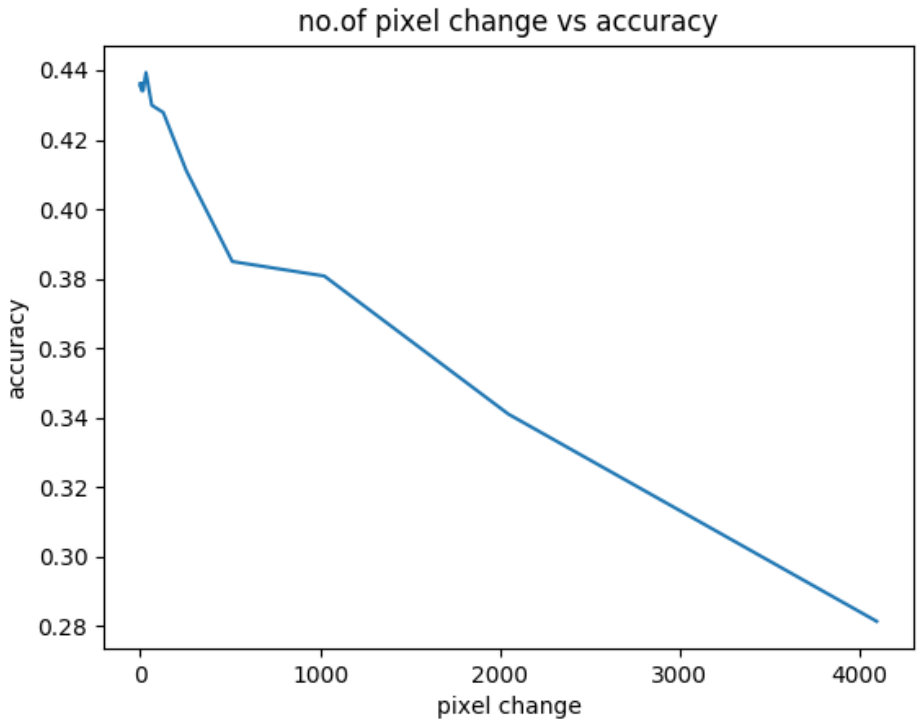


Most of the images contain red colour patterns in excess then some blue coloured.

(j) Apply guided back propagation on any 10 neurons in the Conv6 layer and plot the images which excite this neuron. The idea again is to discover interesting patterns which excite some neurons.

Solution:

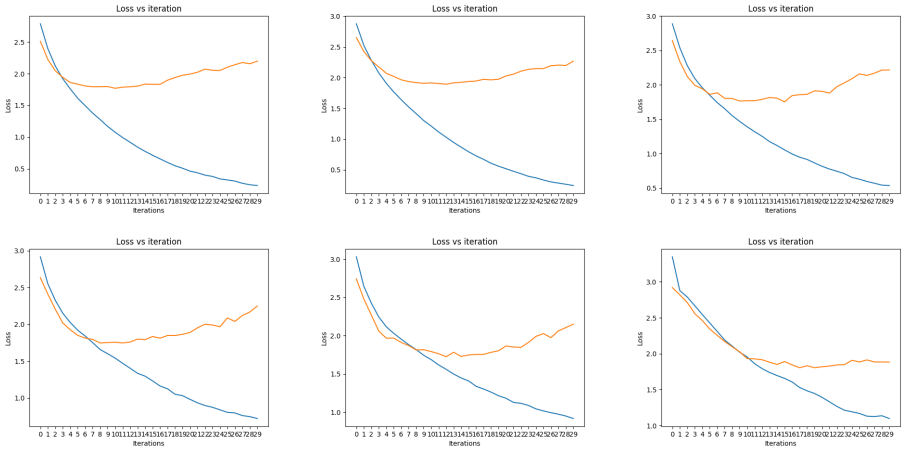
(k) Try to find different ways in which you can fool the network. One way is to randomly change some pixels in the image such that the class is changed. Report the plot of accuracy v/s number of pixels changed on the test set.

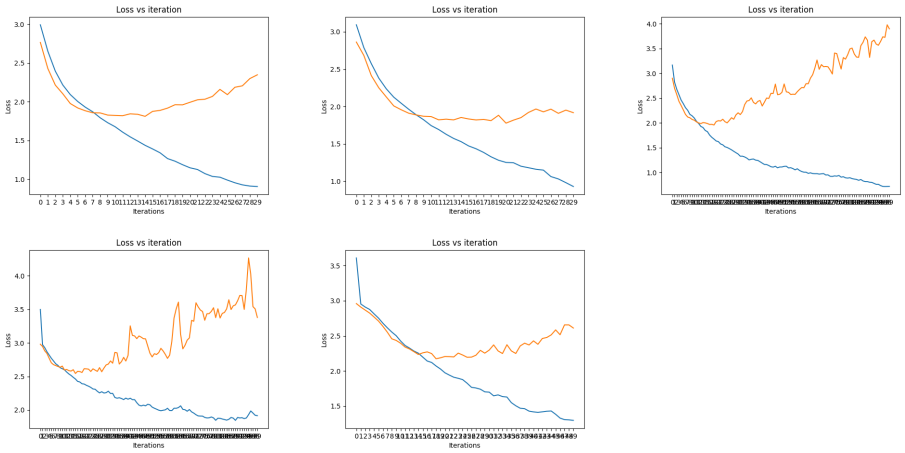


Solution:
The network can be fooled with images with full noise.

(l) Hyperparamater tuning for (learning rate, init, batch size) .

Solution:
Following 11 images are some test results with learning rate varying from 0.00008 to 0.008, batch size from 64 to 512 with various different combinations. These include Xavier and He initializations.





Hyperparameter tuning is done as follows:
1) Fix batch size as 64 for first set of iterations and then as 512 for second.
2) Try values of learning rate fron 0.1 to 0.00001 and we found best rate as 0.0007.
3) Xavier initialization is found to be better.
4) Now vary batch size and 128 is finalized.