

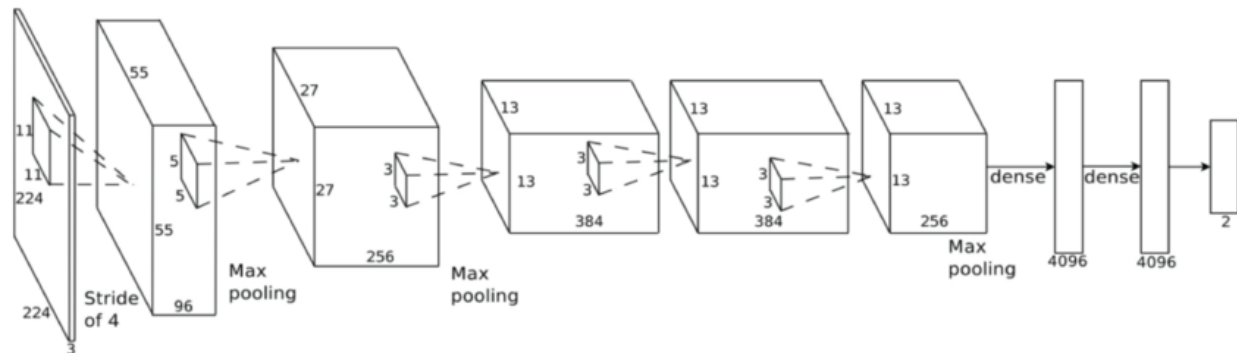
## Assignment 2: Experiment 3

Team 10

[Link to experiments sheet](#), [Link to Colab](#)

### Implement the Alexnet models in Keras.

Summarize Accuracy measures for Train, Test and Validation for the best model:



### Key Takeaways of Alexnet:

- It contains 5 convolutional layers and 3 fully connected layers.
- Unlike previous models Alexnet used 'Relu' activation instead of 'Tanh' to add non-linearity.
- Use dropout instead of regularisation to deal with overfitting. However the training time is doubled with the dropout rate of 0.5.

The image size in the above architecture chart should be  $224 * 224$ , but we were asked to work on  $32*32$  image size. So, we used the  $32*32$  image data and applied it to Alexnet. Also, made some changes to the model and then trained the model.

Some of the changes to the original alexnet are:

- Changed the input to (32,32)
- Changed the final softmax layer to 200 output classes to match the problem.
- Changes are made to pooling and padding layers.\*

\* (Tested with and without making changes)

When compared with the best model we created in the problem 1. AlexNet performed way lower.

## Assignment 2: Experiment 3

### Team 10

#### Summary of the model:

Layer (type)	Output Shape	Param #
=====		
conv2d_11 (Conv2D)	(None, 16, 16, 96)	2688
activation_19 (Activation)	(None, 16, 16, 96)	0
max_pooling2d_7 (MaxPooling2D)	(None, 8, 8, 96)	0
batch_normalization_17 (Batch Normalization)	(None, 8, 8, 96)	384
conv2d_12 (Conv2D)	(None, 8, 8, 256)	2973952
activation_20 (Activation)	(None, 8, 8, 256)	0
max_pooling2d_8 (MaxPooling2D)	(None, 4, 4, 256)	0
batch_normalization_18 (Batch Normalization)	(None, 4, 4, 256)	1024
conv2d_13 (Conv2D)	(None, 4, 4, 384)	885120
activation_21 (Activation)	(None, 4, 4, 384)	0
batch_normalization_19 (Batch Normalization)	(None, 4, 4, 384)	1536
conv2d_14 (Conv2D)	(None, 4, 4, 384)	1327488
activation_22 (Activation)	(None, 4, 4, 384)	0
batch_normalization_20 (Batch Normalization)	(None, 4, 4, 384)	1536
conv2d_15 (Conv2D)	(None, 4, 4, 256)	884992

**Assignment 2: Experiment 3**  
**Team 10**

activation_23	(Activation)	(None, 4, 4, 256)	0
max_pooling2d_9	(MaxPooling2	(None, 2, 2, 256)	0
batch_normalization_21	(Batc	(None, 2, 2, 256)	1024
flatten_3	(Flatten)	(None, 1024)	0
dense_9	(Dense)	(None, 4096)	4198400
activation_24	(Activation)	(None, 4096)	0
dropout_7	(Dropout)	(None, 4096)	0
batch_normalization_22	(Batc	(None, 4096)	16384
dense_10	(Dense)	(None, 4096)	16781312
activation_25	(Activation)	(None, 4096)	0
dropout_8	(Dropout)	(None, 4096)	0
batch_normalization_23	(Batc	(None, 4096)	16384
dense_11	(Dense)	(None, 200)	819400
activation_26	(Activation)	(None, 200)	0
=====			

## **Assignment 2: Experiment 3**

### **Team 10**

The best test accuracy and test loss obtained from the AlexNet were **0.191** and **4.4573** respectively. Although, a train accuracy and train loss of 0.8437 and 0.5228 were obtained the model performed poorly on the test data. [Experiment Results Here.](#)

Some of the possible reasons are:

- The model was developed for 224\*224 not for 32\*32
- Images are 32x32 and are using an initial kernel of 11x11. This might result in losing a lot of information.

**Approach to further Improve the accuracy:** Changed the kernel size to (3,3) instead of (11,11). This did not help improve the accuracy.

### **Best Model so far on Alexnet:**

Base alexnet as shown above +

Optimizer = 'Adam', batch\_size=32 and epochs=10.

### **Sources:**

1. <https://medium.com/@smallfishbigsea/a-walk-through-of-alexnet-6cbd137a5637>
2. <https://www.artificial-intelligence.video/keras-9-1-buildtrain-alexnet-with-cifar-10>
3. <https://shaoanlu.wordpress.com/2017/05/29/sgd-all-which-one-is-the-best-optimizer-dogs-vs-cats-toy-experiment/>