We start by testing the impact of different batch size.

We run this test with all default setting as the original paper except changing batchsize. We test larger batch sizes vs small batch sizes and since the code is tested based on CIFAR-10 dataset which has 50,000 training images(divisible by 10), we also include fully divisible batch size 200, 100 and 50 as options. From Table-1, we noticed that fully divisible batch size tends to have higher training time per epoch but less epoch to finish training. In addition to the training time, less batch size works better on CIFAR-10 dataset and when batch size is 100, resNet is able to reach 86.03% accuracy, which is higher than any other batch size by 1% to 2%.

BatchSize(resNet18)	Accuracy	Time per epoch	Total Time Training
256	0.8478	41s	1845s
128	0.8482	55s	2035s
64	0.8593	92s	4743s
32	0.8409	155s	5776s
200	0.8480	41s	1804s
100	0.8603	67s	3350s
50	0.8517	105s	4387s

Table-1 resNet-18 with different batch size

The original author uses basic block in resNet18 and resNet34 but apply bottleneck block in the deeper resNet such as resNet152 because they states that bottleneck architecture could dramatically decrease the time complexity with deeper layers. Just like what author does from resNet34 to resNet50, we replaced basic block in resNet18 with bottleneck block leads to 8 more layers in the models(resNet26). ResNet26's time per epoch is only incremented by 2s.

	Accuracy	Time per epoch	Total Time Training
resNet18	0.8603	30s	3350s
resNet18-bottleNeck	0.8471	32s	1472s

Table-2 Basic Block vs BottleNeck Block

Based on Table-3, we verified that applying dropout after batch normalization leads to better accuracy than original bottleneck block with same training time per epoch.

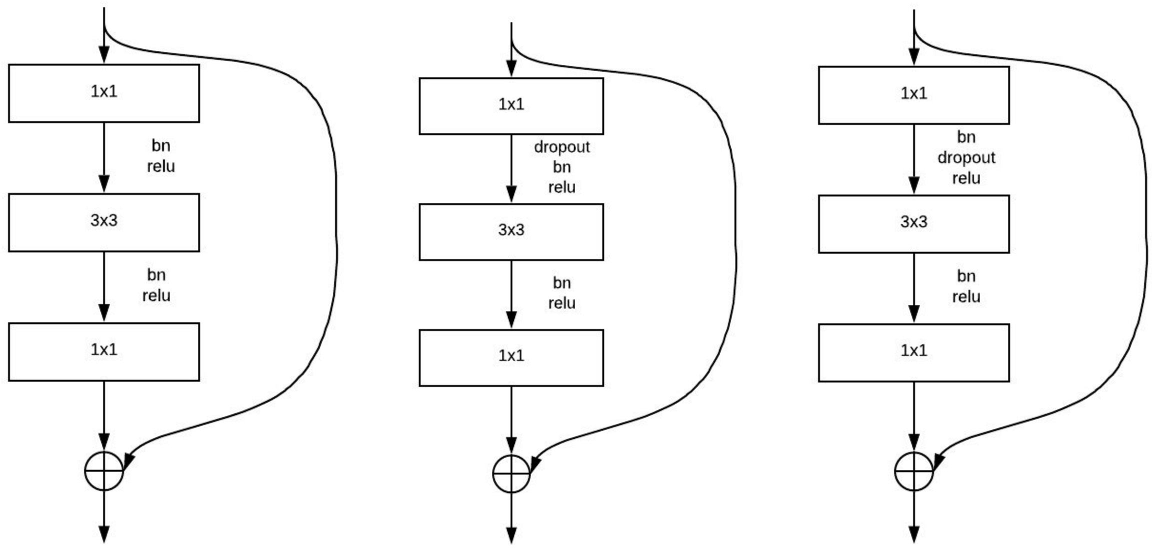


Image-1 Original bottleneck Block vs Dropout First Block vs Proposed Block

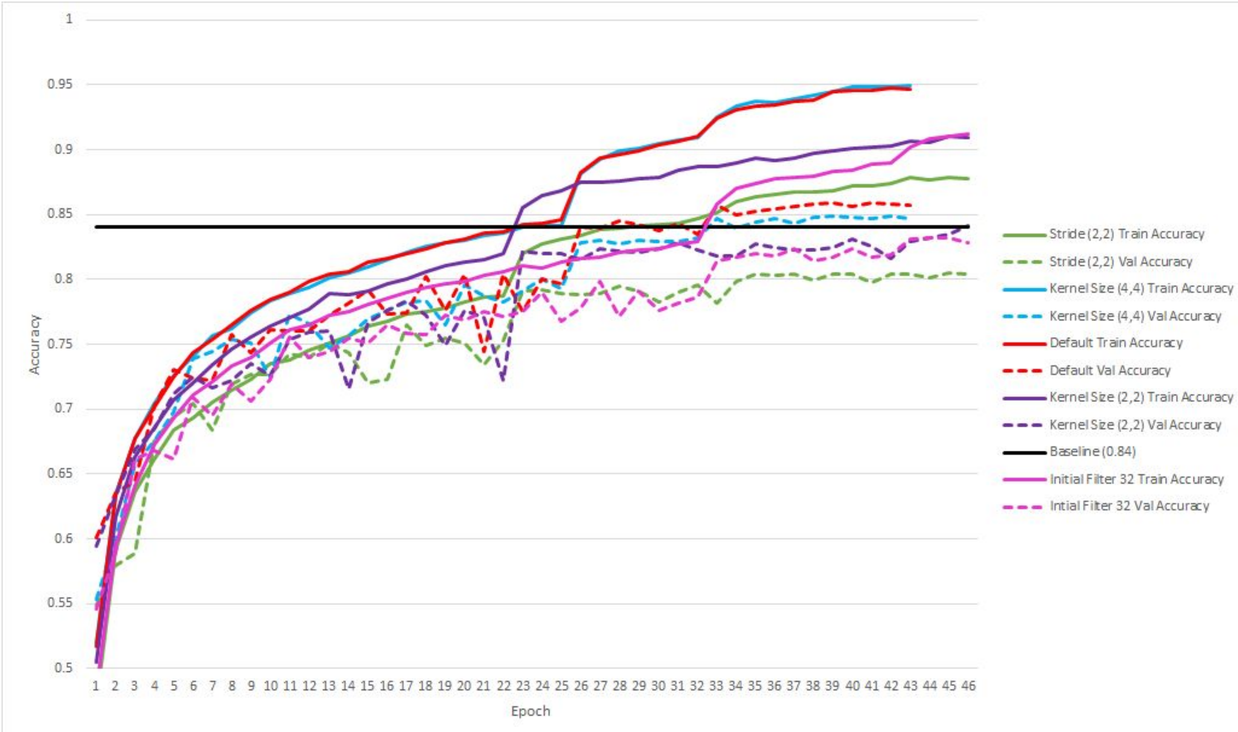
	Accuracy	Time per epoch
Original bottleneck Block	0.8471	32s
Dropout First Block	0.7812	36s
Proposed Block	0.8505	32s

Table-3 Original Block vs Dropout First Block vs Proposed Block

Figure 4 describes the results of a series of tests on ResNet18 (basic block version) where the default stride is set to (1,1). Note that for the initial layer of a block, the stride is always (2, 2). A test is then conducted where the stride is changed to (2,2) at all times, unchanged for first layer of block. The default kernel size for these blocks is (3,3) and these will be changed in 2 different tests: one test with a kernel size of (2,2) and the other with a kernel size of (4,4). The last test changes the initial filters from 64 to 32, in order to see its impact.

Observations:

- Kernel Size (3,3) and Kernel Size (4,4) have an identical training accuracy over the epoch however Kernel Size (4,4) is slower per epoch and also leads to a slightly lower final validation accuracy.
- Every other parameter change does not affect the epoch time, however they all performed worse, from 1.5% to 5% worse in accuracy by the final epoch.
-



Residual_block stride on ResNet18 (Basic Block)

Change	Accuracy	Time per epoch
Default	0.8568	30s
Stride from (1,1) to (2,2)	0.8042	30s
Kernel Size from (3,3) to (4,4)	0.8469	37s
Kernel Size from (3,3) to (2,2)	0.8416	29s
Filters start at 32 instead of 64	0.8283	28s

Reference:

[arXiv:1801.05134](https://arxiv.org/abs/1801.05134)