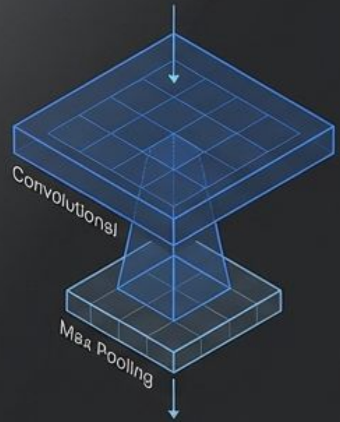
Deep Learning & Transfer Learning

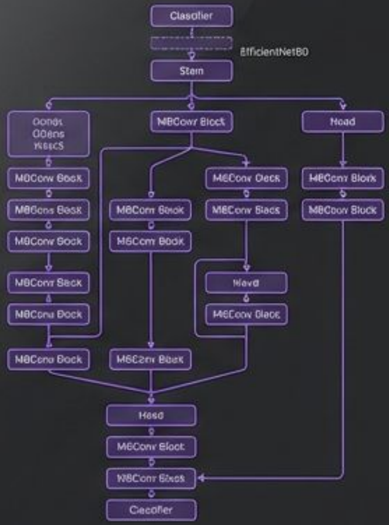
A Comparative Analysis of Image Classification on CIFAR-10

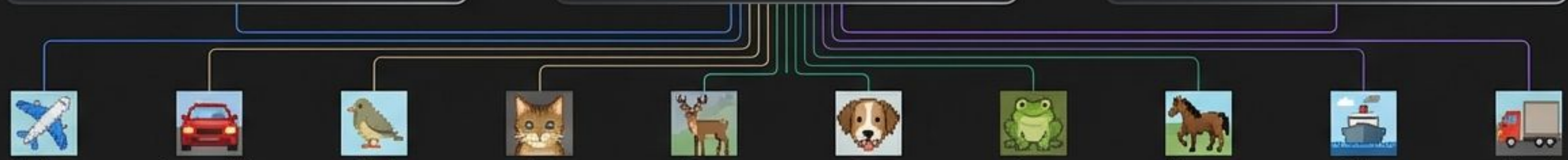
Baseline



Deep Scaling

Transfer Learning





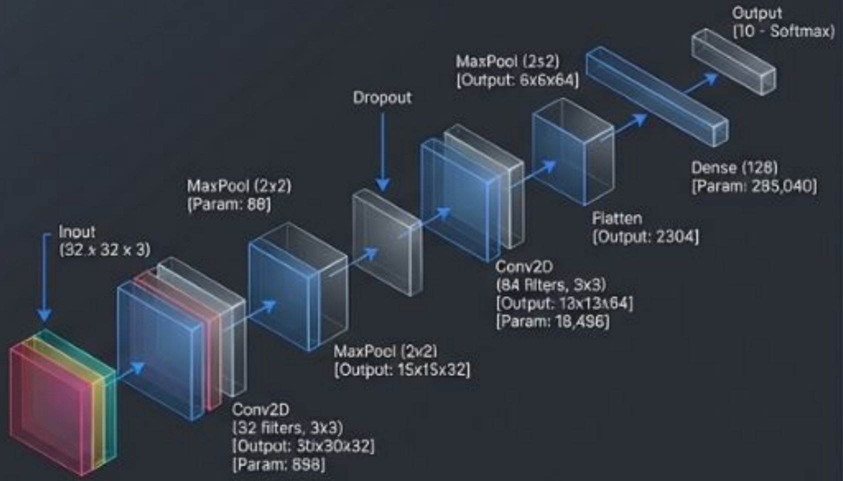
The Baseline: Simple CNN

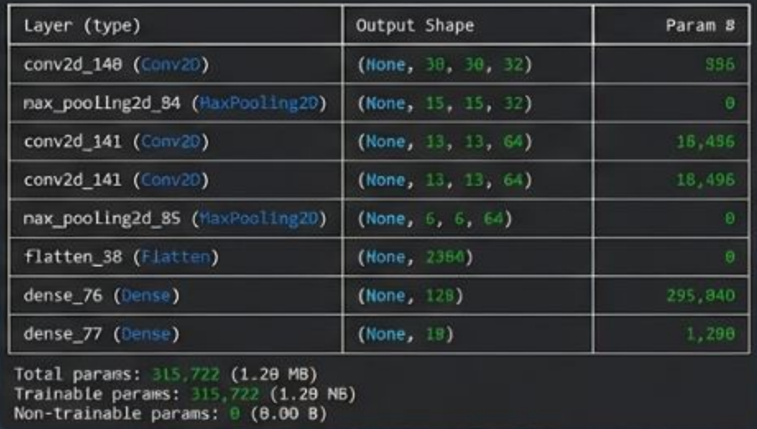
Model Stats

Total Parameters: 315,722

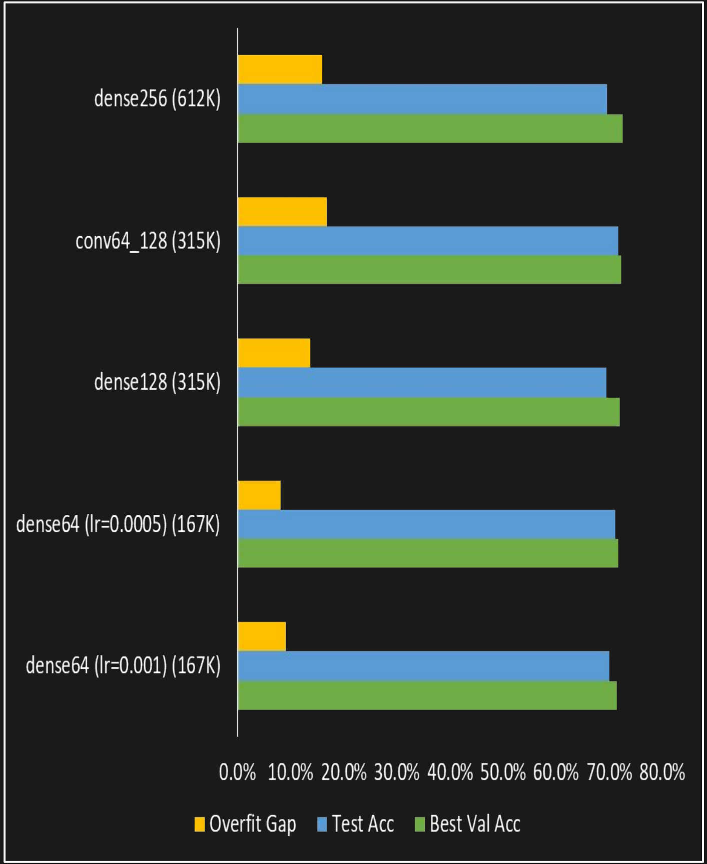
Accuracy: ~64% - 68%

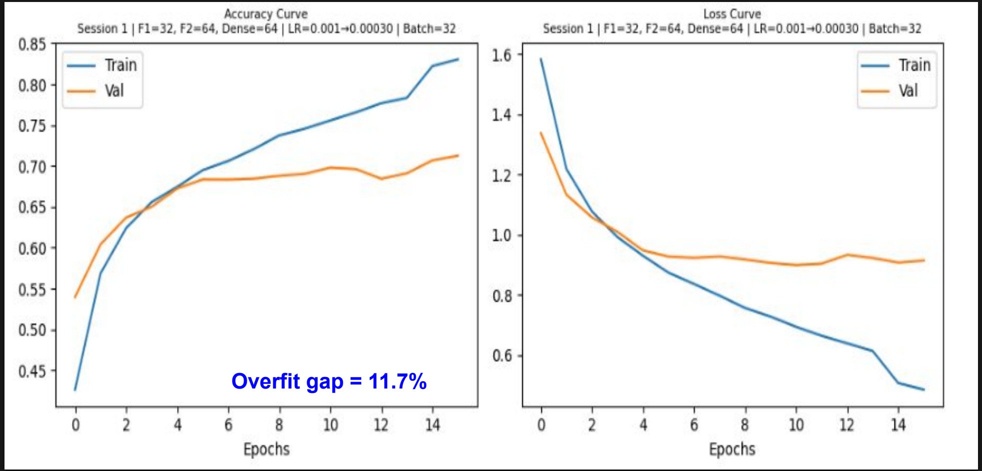
Structure: 2 Conv Blocks + Dense

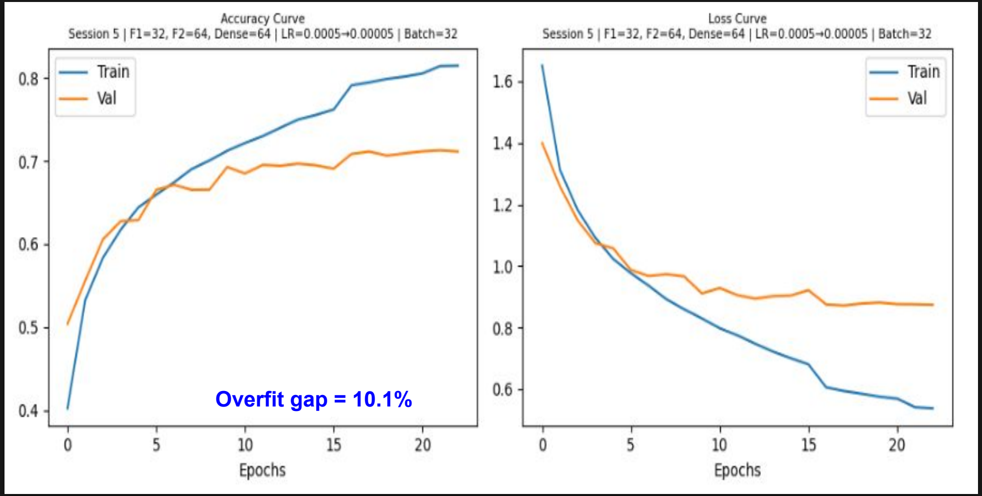




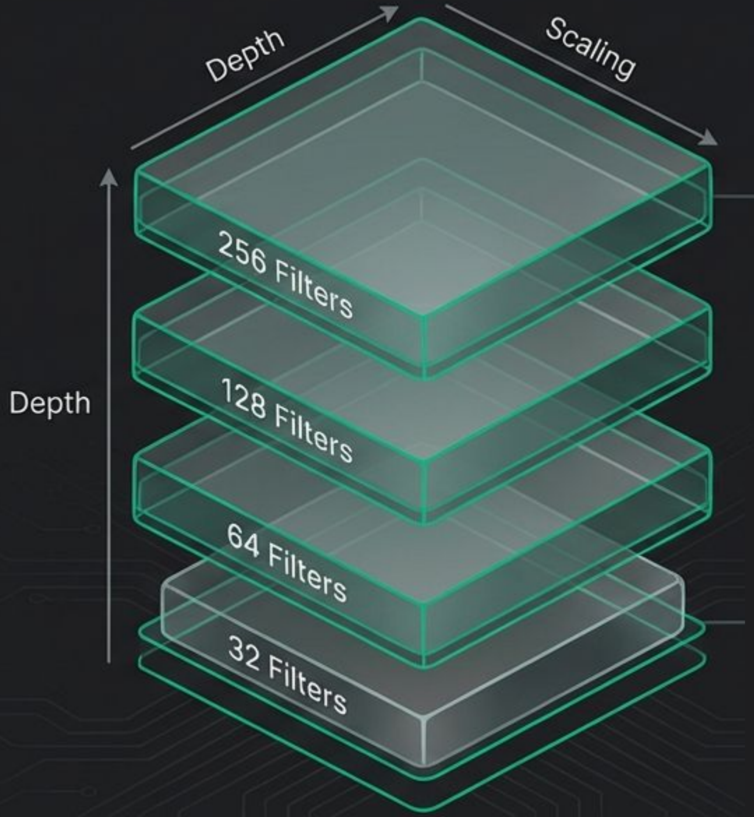
```
Layer (type)                      Output Shape            Param #
conv2d_140 (Conv2D)               (None, 30, 30, 32)          896
max_pooling2d_84 (MaxPooling2D)   (None, 15, 15, 32)            0
conv2d_141 (Conv2D)               (None, 13, 13, 64)       18,496
conv2d_141 (Conv2D)               (None, 13, 13, 64)       18,496
max_pooling2d_85 (MaxPooling2D)   (None, 6, 6, 64)              0
flatten_38 (Flatten)              (None, 2304)                  0
dense_76 (Dense)                  (None, 128)             295,840
dense_77 (Dense)                  (None, 10)                1,290
Total params: 315,722 (1.20 MB)
Trainable params: 315,722 (1.20 MB)
Non-trainable params: 0 (0.00 B)
```

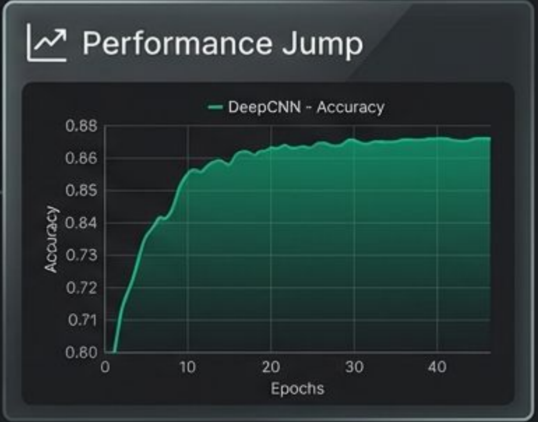

dense256 (612K)
conv64_128 (315K)
dense128 (315K)
dense64 (lr=0.0005) (167K)
dense64 (lr=0.001) (167K)
0.0% 10.0% 20.0% 30.0% 40.0% 50.0% 60.0% 70.0% 80.0%
Overfit Gap
Test Acc
Best Val Acc


Accuracy Curve
Session 1 | F1=32, F2=64, Dense=64 | LR=0.001→0.00030 | Batch=32
Train
Val
Epochs
Overfit gap = 11.7%
Loss Curve
Session 1 | F1=32, F2=64, Dense=64 | LR=0.001→0.00030 | Batch=32
Train
Val
Epochs

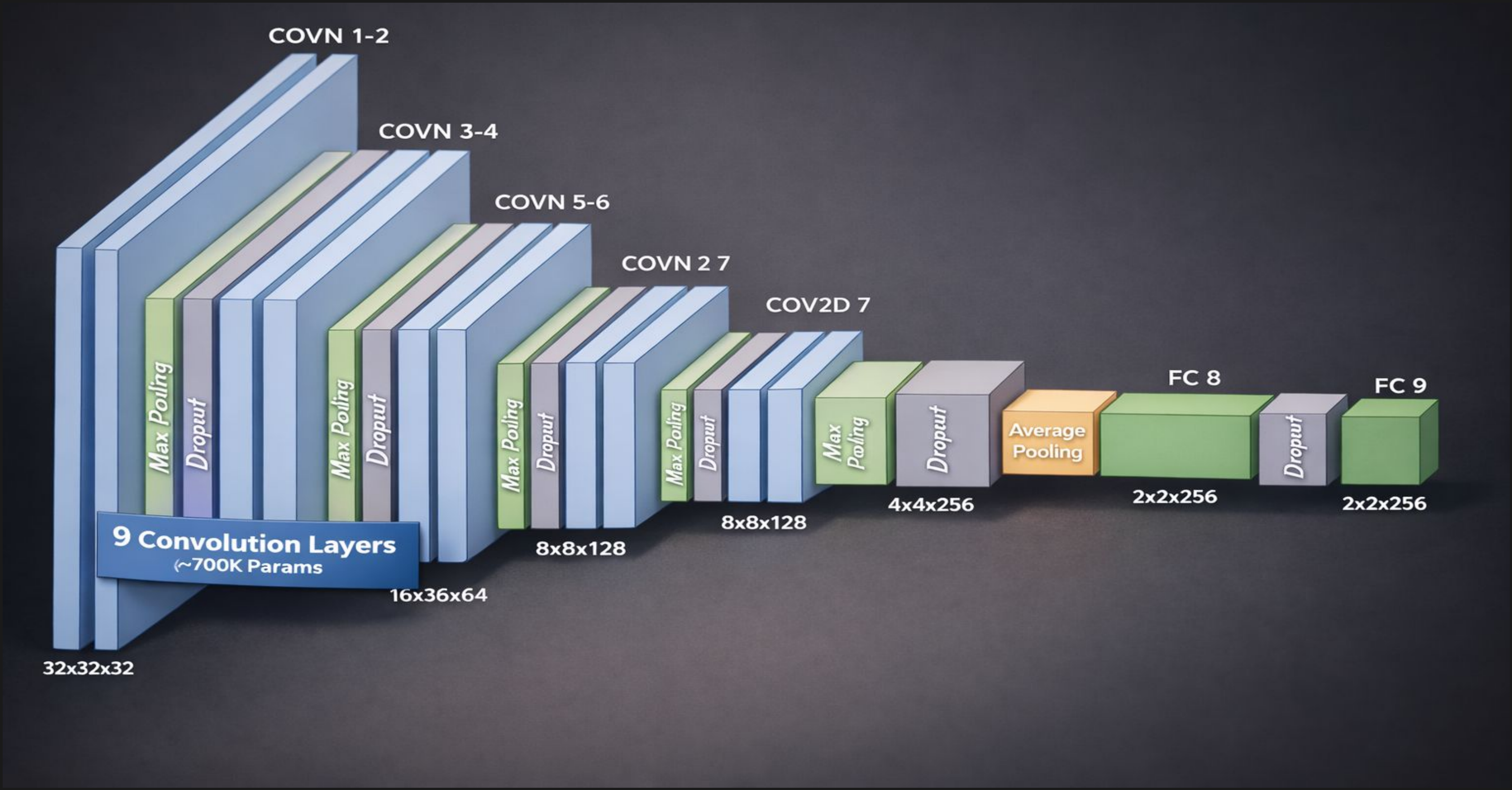

Accuracy Curve
Session 5 | F1=32, F2=64, Dense=64 | LR=0.0005→0.00005 | Batch=32
Train
Val
Epochs
Overfit gap = 10.1%
Loss Curve
Session 5 | F1=32, F2=64, Dense=64 | LR=0.0005→0.00005 | Batch=32
Train
Val
Epochs

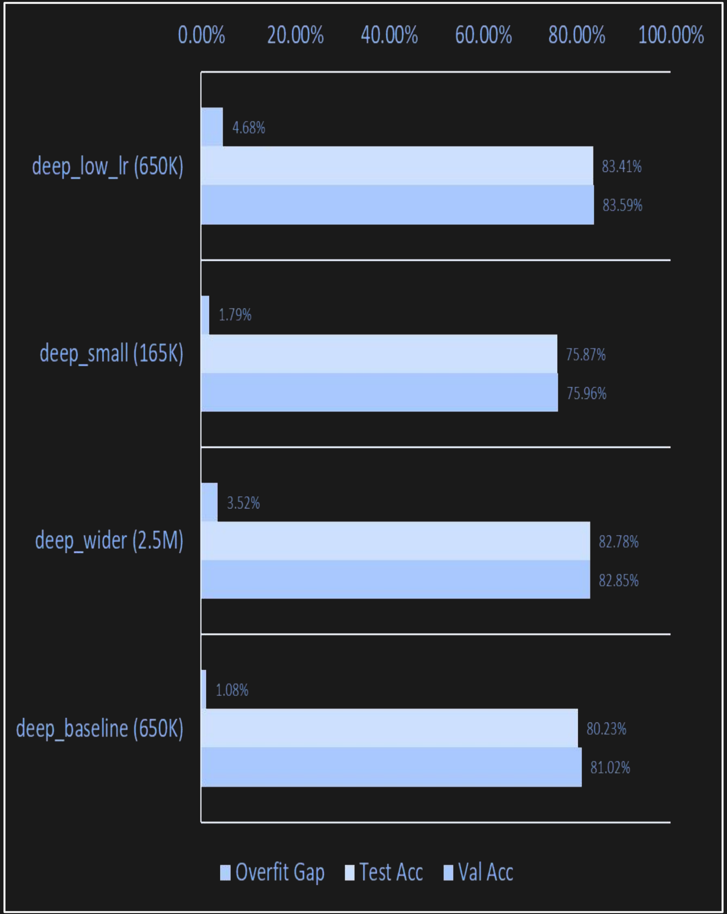
Escalation: Deep CNN Architecture



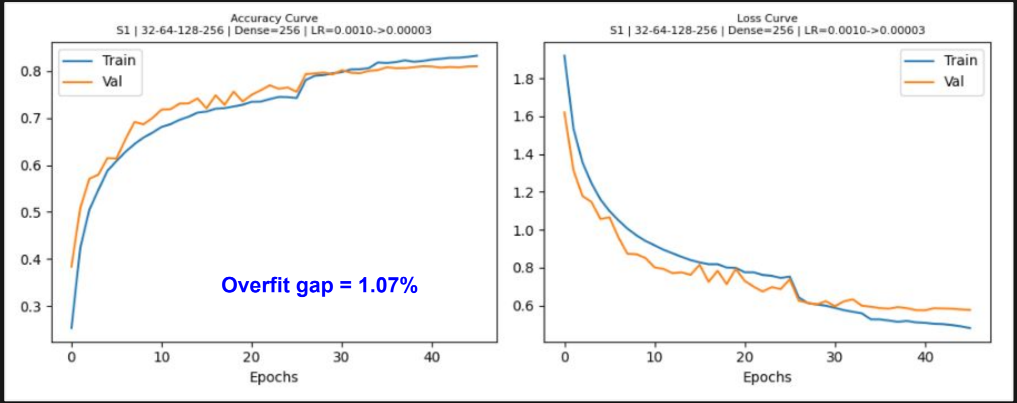


Deeper CNN Architecture

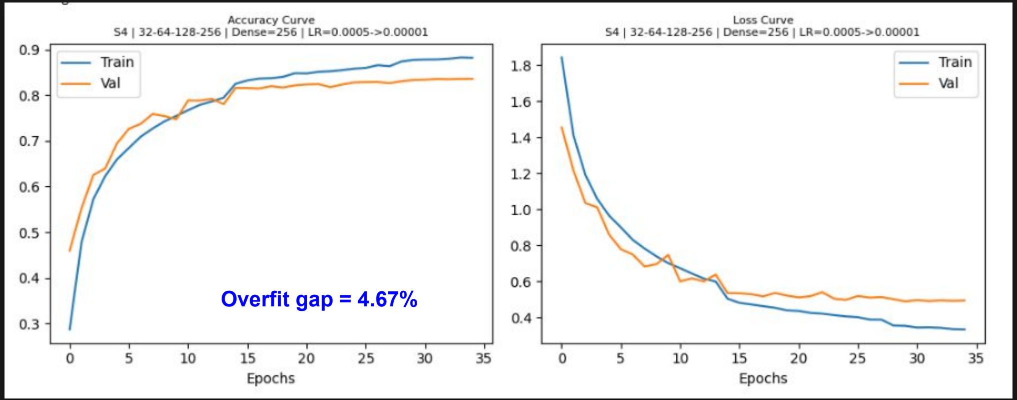




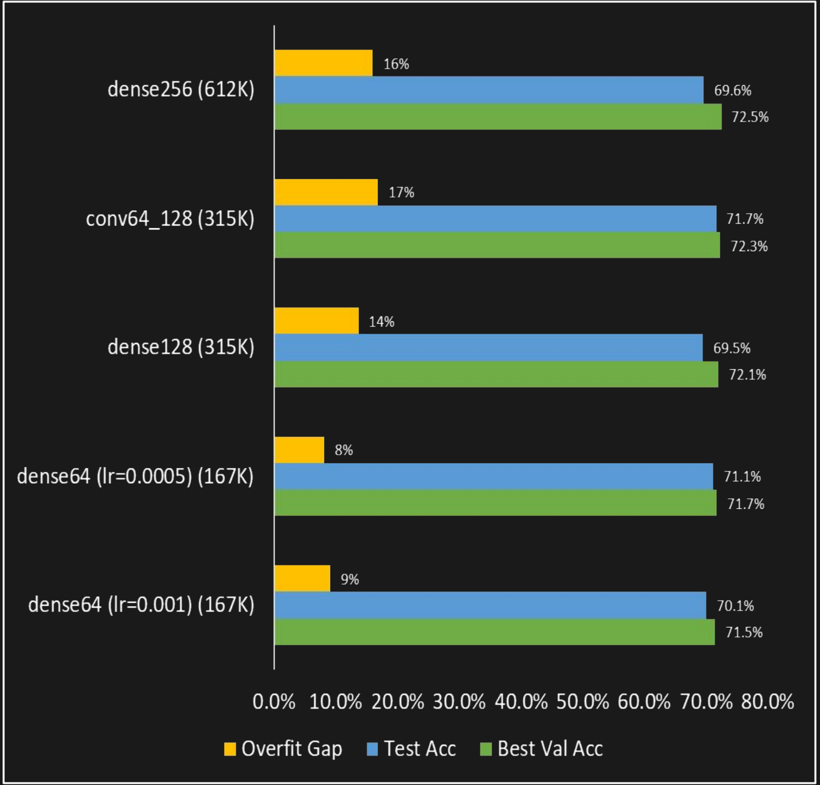
deep_baseline(650K): F1=32, F2=64, F3= 128, F4= 256, Dense=256, Batch=32, Epochs= ~~50~~ 46 LR= ~~0.001~~ 0.00003

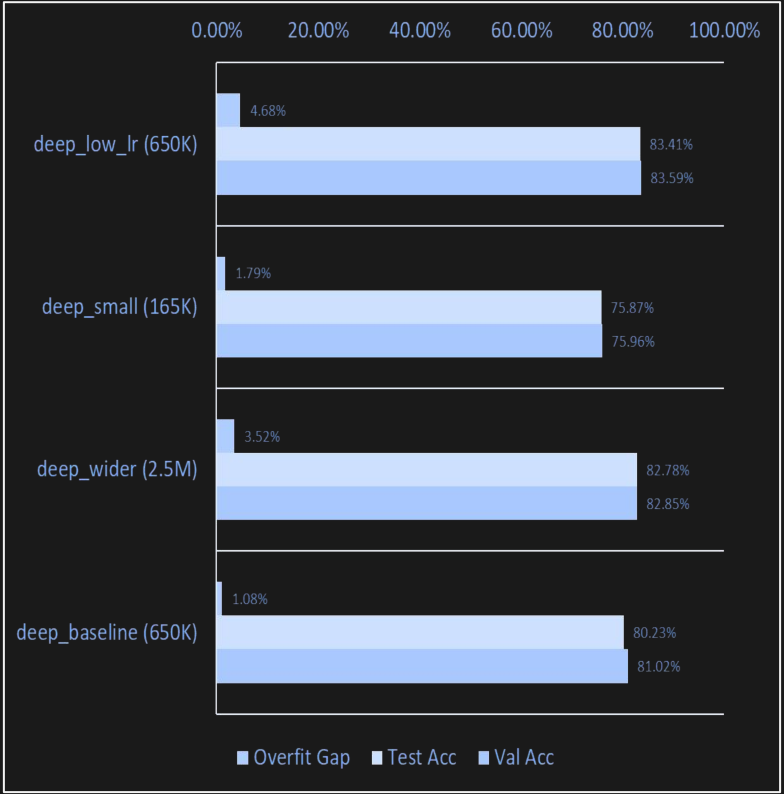


deep_small(165K): F1=32, F2=64, F3= 128, F4= 256, Dense=256, Batch=32, Epochs=~~50~~ 35, LR = ~~0.0005~~ 0.00001



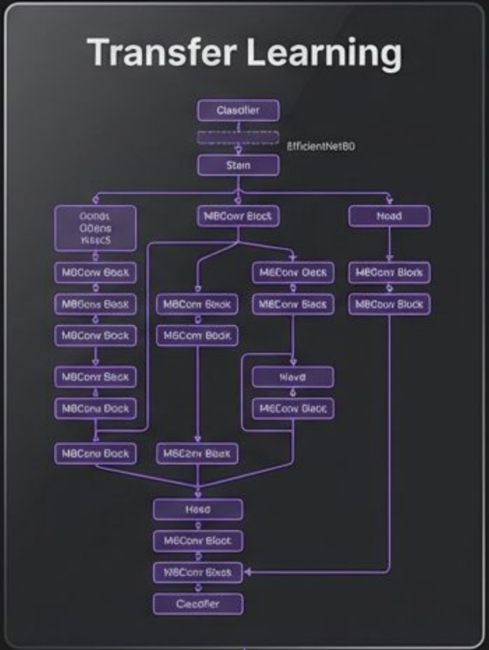
Simple CNN vs Deeper CNN

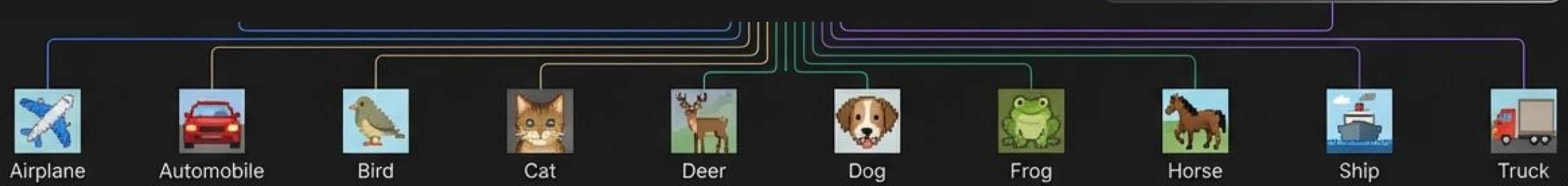




Transfer Learning: Pretrained Models

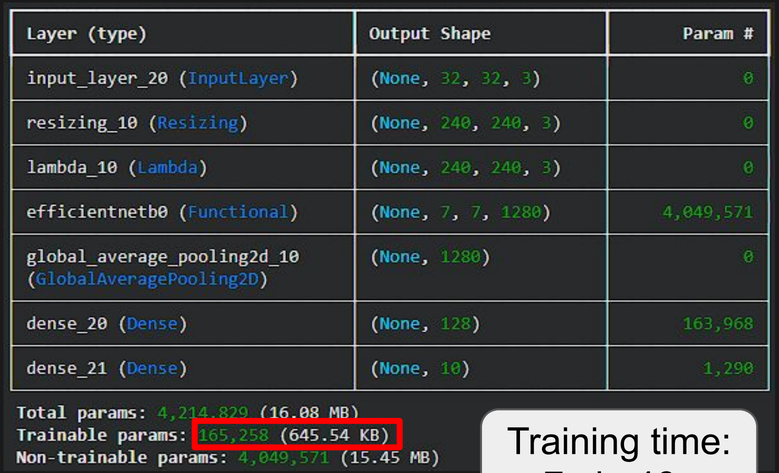
EfficientNet B0





EfficientNetB0

Retrain almost 0% of the layers:



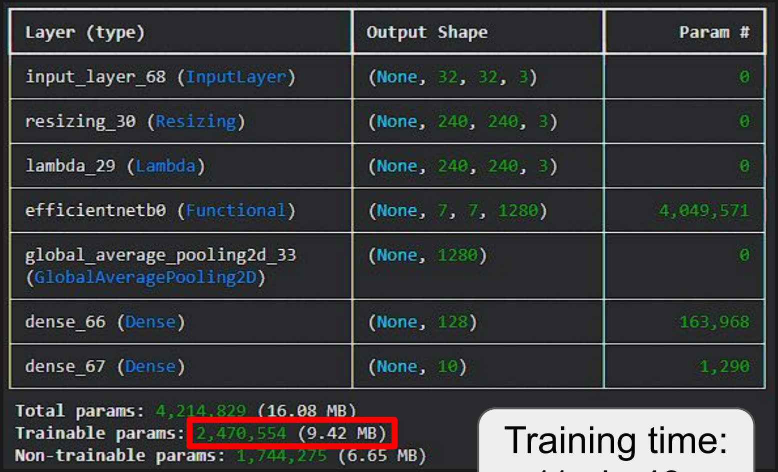
Layer (type)	Output Shape	Param #
input_layer_20 (InputLayer)	(None, 32, 32, 3)	0
resizing_10 (Resizing)	(None, 240, 240, 3)	0
lambda_10 (Lambda)	(None, 240, 240, 3)	0
efficientnetb0 (Functional)	(None, 7, 7, 1280)	4,049,571
global_average_pooling2d_10 (GlobalAveragePooling2D)	(None, 1280)	0
dense_20 (Dense)	(None, 128)	163,968
dense_21 (Dense)	(None, 10)	1,290

```
Total params: 4,214,829 (16.08 MB)
Trainable params: 165,258 (645.54 KB)
Non-trainable params: 4,049,571 (15.45 MB)
```

Training time:
7min 46s

accuracy:
~92 %

retrain 20% of the layers:



Layer (type)	Output Shape	Param #
input_layer_68 (InputLayer)	(None, 32, 32, 3)	0
resizing_30 (Resizing)	(None, 240, 240, 3)	0
lambda_29 (Lambda)	(None, 240, 240, 3)	0
efficientnetb0 (Functional)	(None, 7, 7, 1280)	4,049,571
global_average_pooling2d_33 (GlobalAveragePooling2D)	(None, 1280)	0
dense_66 (Dense)	(None, 128)	163,968
dense_67 (Dense)	(None, 10)	1,290

```
Total params: 4,214,829 (16.08 MB)
Trainable params: 2,470,554 (9.42 MB)
Non-trainable params: 1,744,275 (6.65 MB)
```

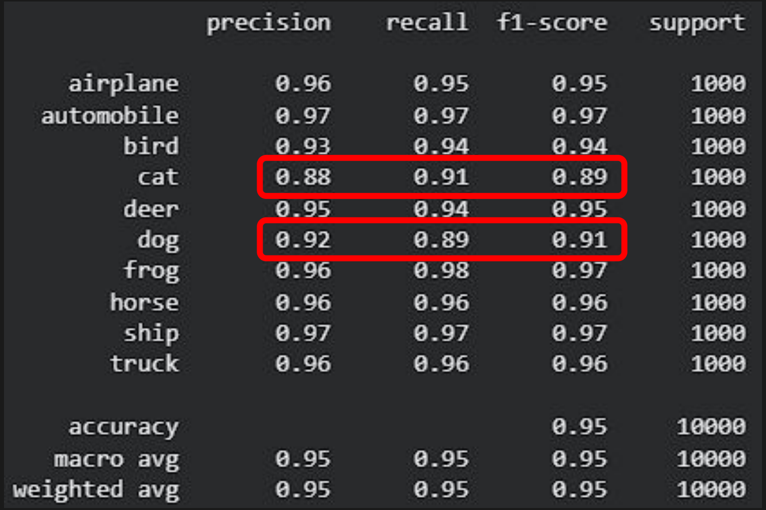
Training time:
11min 43s

accuracy:
~95 %

- Same architecture
- difference in “Trainable params”, training time and accuracy

EfficientNetB0

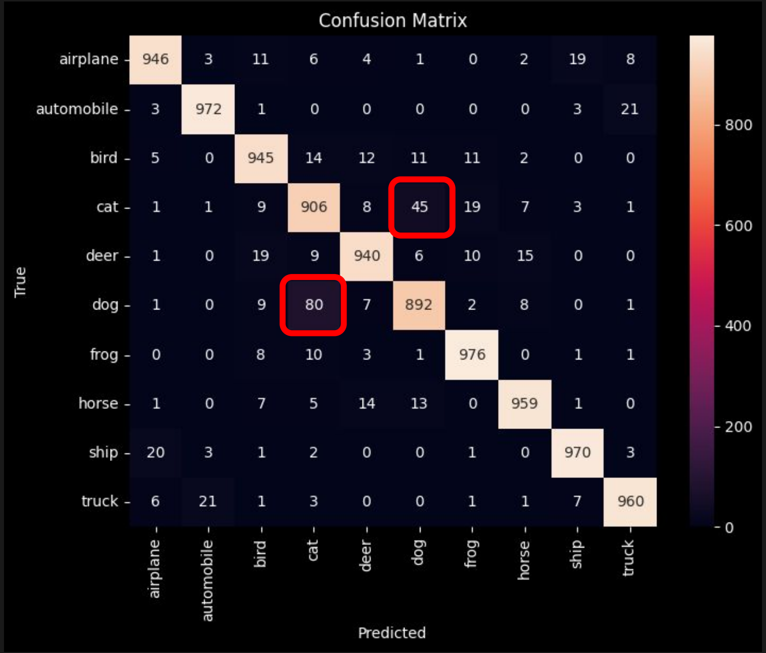
retrain 20% of the layers:



```
              precision    recall  f1-score   support

    airplane       0.96      0.95      0.95      1000
  automobile       0.97      0.97      0.97      1000
        bird       0.93      0.94      0.94      1000
         cat       0.88      0.91      0.89      1000
        deer       0.95      0.94      0.95      1000
         dog       0.92      0.89      0.91      1000
        frog       0.96      0.98      0.97      1000
       horse       0.96      0.96      0.96      1000
        ship       0.97      0.97      0.97      1000
       truck       0.96      0.96      0.96      1000

    accuracy                           0.95     10000
   macro avg       0.95      0.95      0.95     10000
weighted avg       0.95      0.95      0.95     10000
```

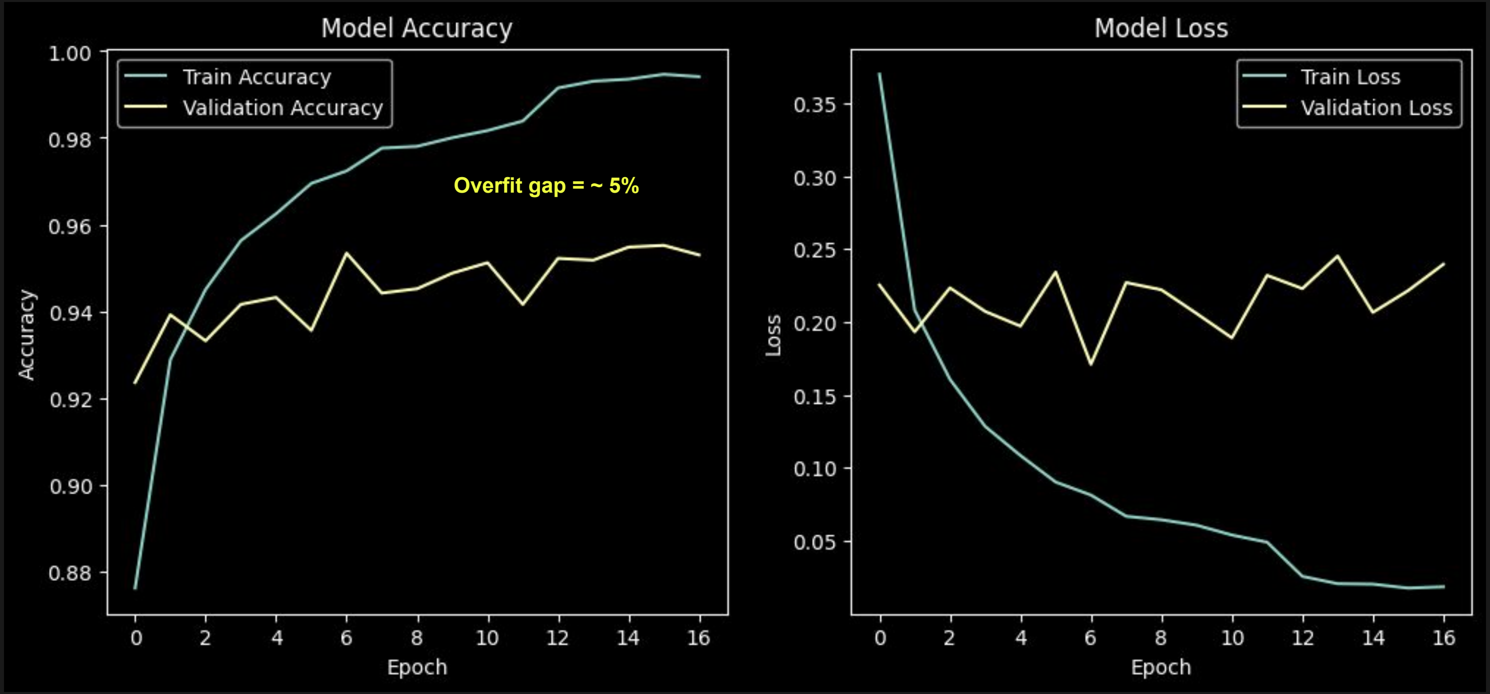


Confusion Matrix

True \ Predicted	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	946	3	11	6	4	1	0	2	19	8
automobile	3	972	1	0	0	0	0	0	3	21
bird	5	0	945	14	12	11	11	2	0	0
cat	1	1	9	906	8	45	19	7	3	1
deer	1	0	19	9	940	6	10	15	0	0
dog	1	0	9	80	7	892	2	8	0	1
frog	0	0	8	10	3	1	976	0	1	1
horse	1	0	7	5	14	13	0	959	1	0
ship	20	3	1	2	0	0	1	0	970	3
truck	6	21	1	3	0	0	1	1	7	960

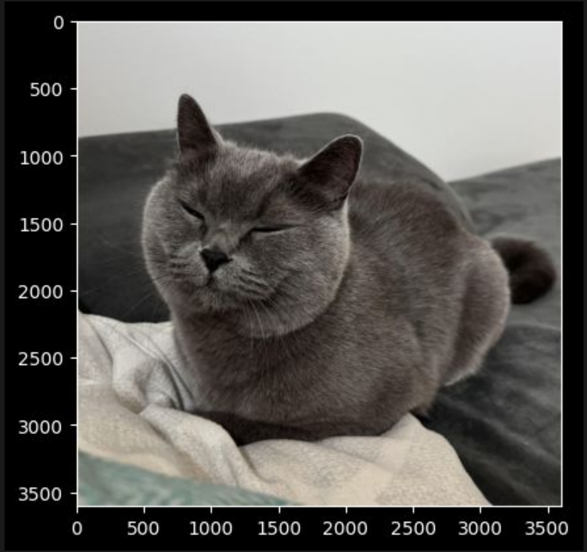
EfficientNetB0

retrain 20% of the layers:

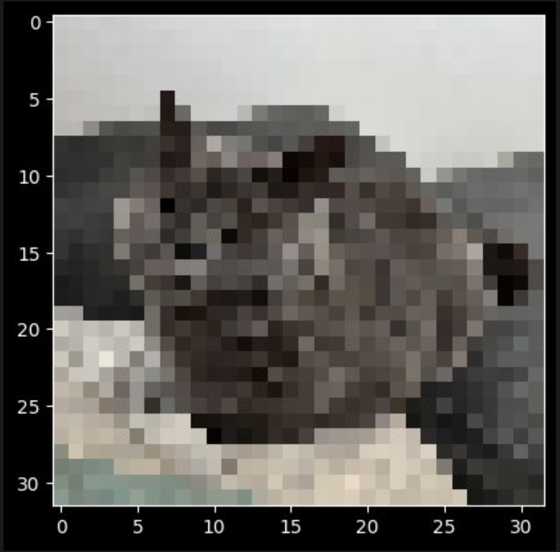


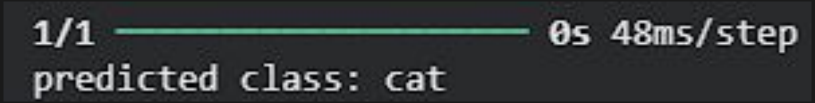
Predicting the class of our own pet

Original picture 3605x3605 px:



Resized picture 32x32 px:

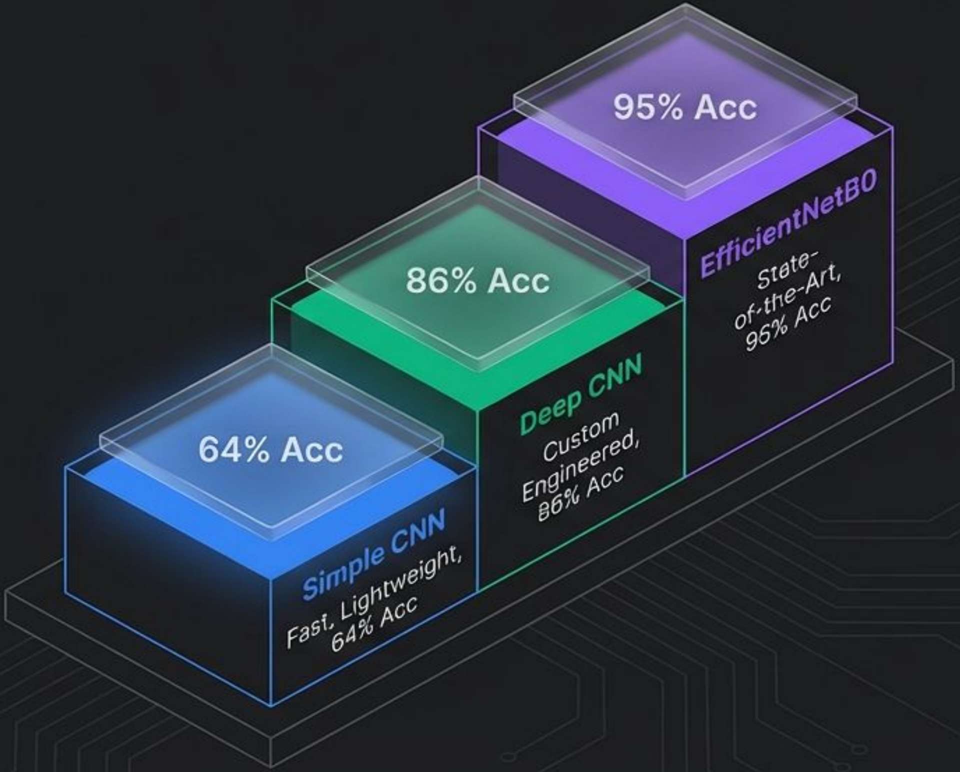




```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
predicted class: cat
```

Engineering Editorial

Conclusion: Performance vs. Cost



Final Verdict:

Transfer Learning achieves a decisive victory in accuracy (95%), offering the best performance-to-effort ratio for general classification.

However, this comes with a 7M parameter footprint and increased pre-processing overhead.