CIFAR 10: Deep Learning Models

Introduction

- CIFAR 10 dataset has 50,000 train data and 10,000 test data of 32x32 RGB colour images of 10 categories of objects
- This project compares several deep learning architectures with hyperparameter tuning and also data augmentation
- The architectures are **Multi-Layer Perceptron**, **LeNet CNN and 3 block VGG**. **Data augmentation** is also attempted on the VGG model
- Early stopping is implemented throughout the models to have consistency when the model reaches accuracy saturation (min delta 0.01, patience =10)
- In this real world of limited resources, **the objective is to find the best model with highest accuracy for the same computer resources**

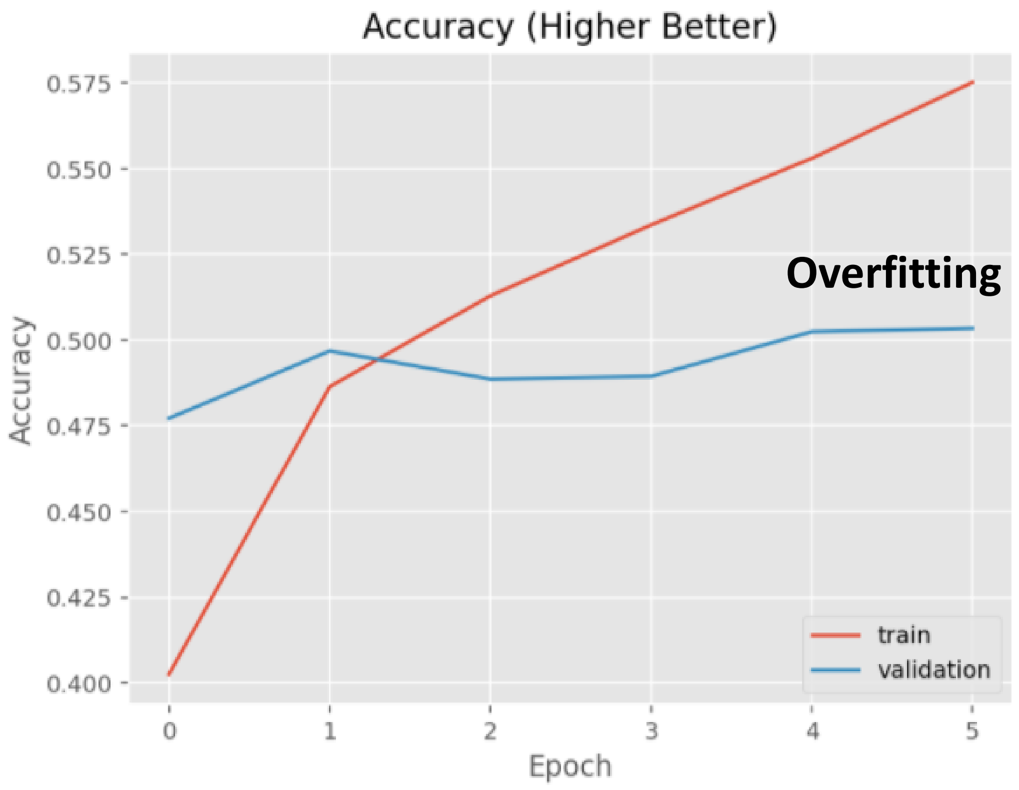
Data Pre-processing

- The data is relatively clean
- Minimal data pre-processing to normalise image colour values of 0 to 255 (representing the colour values for each RGB channel) to floating point between 0 to 1
- One-hot encoding is applied on the target object classification

Model 1: Multi-Layer Perceptron

- The MLP model is first implemented with 2 dense 512-neurons layers and RMSprop optimiser. The best accuracy is 50%
- As there is overfitting with validation accuracy diverging from training accuracy, the number of neurons is reduced to 256, and then 128, with dropouts and batch normalisation also applied
- The model converges at 128 neurons and early stopping at epoch 11 with accuracy also reaching 50%. This is the optimised MLP model that minimises computer resources while giving the same baseline accuracy

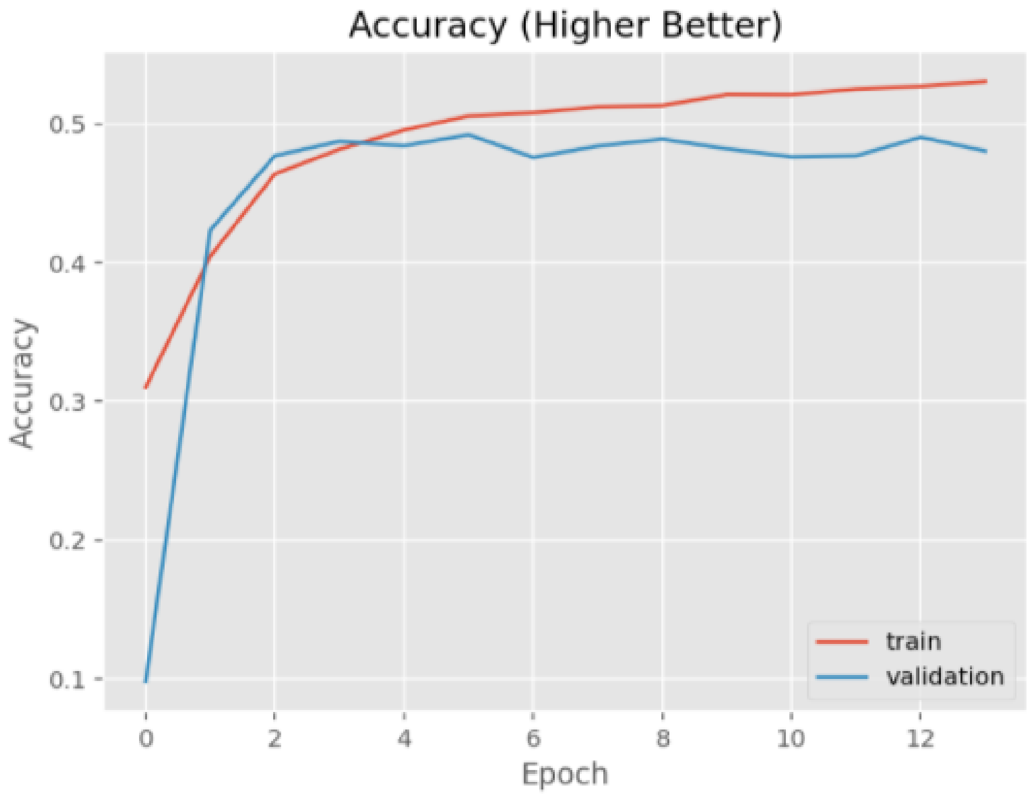
Model 1: Multi-Layer Perceptron

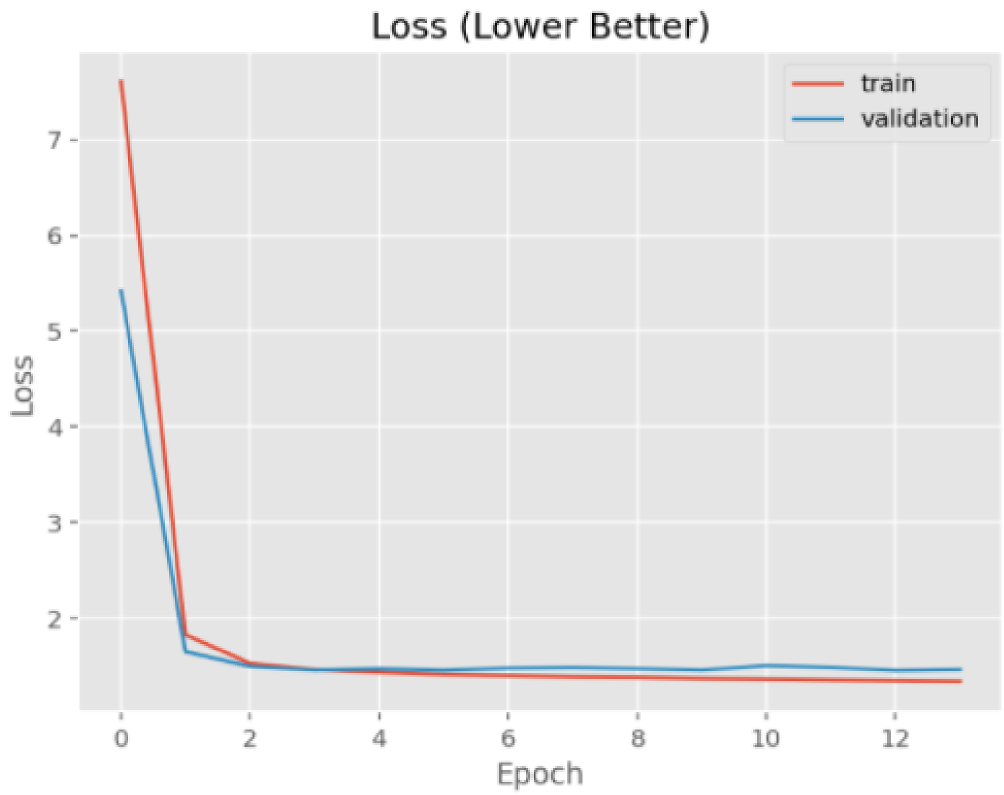


- Model: MLP of 2 Dense 512 neuron layers
- Optimiser: RMSprop
- **Best accuracy at 50% (MLP Baseline)**

```
Test loss: 1.4474800825119019
Test accuracy: 0.504800021648407
```

Model 1: Multi-Layer Perceptron



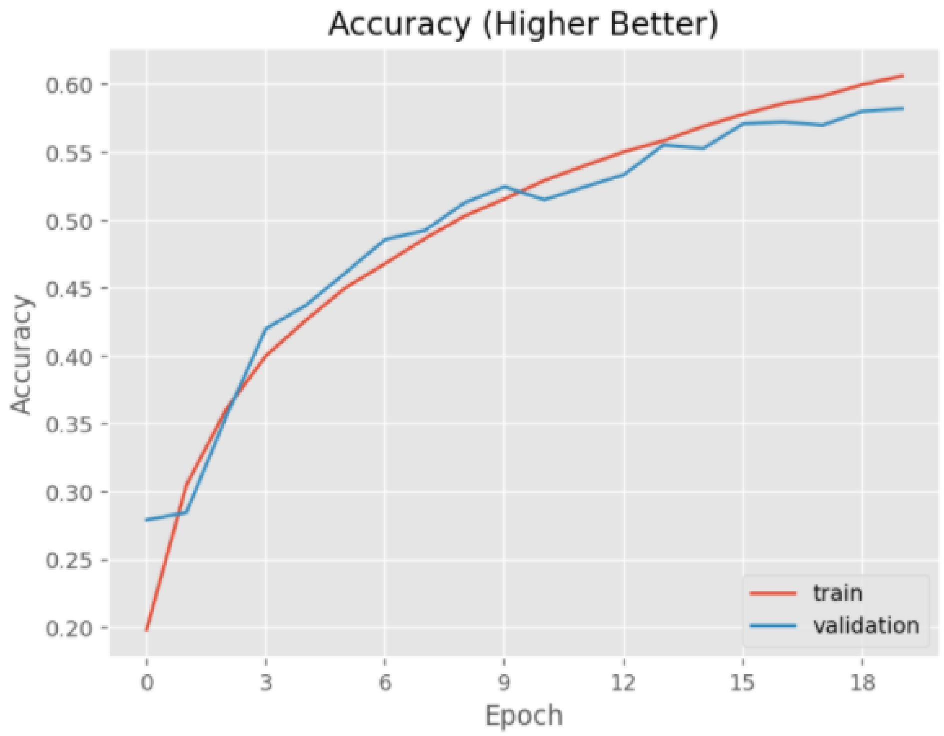


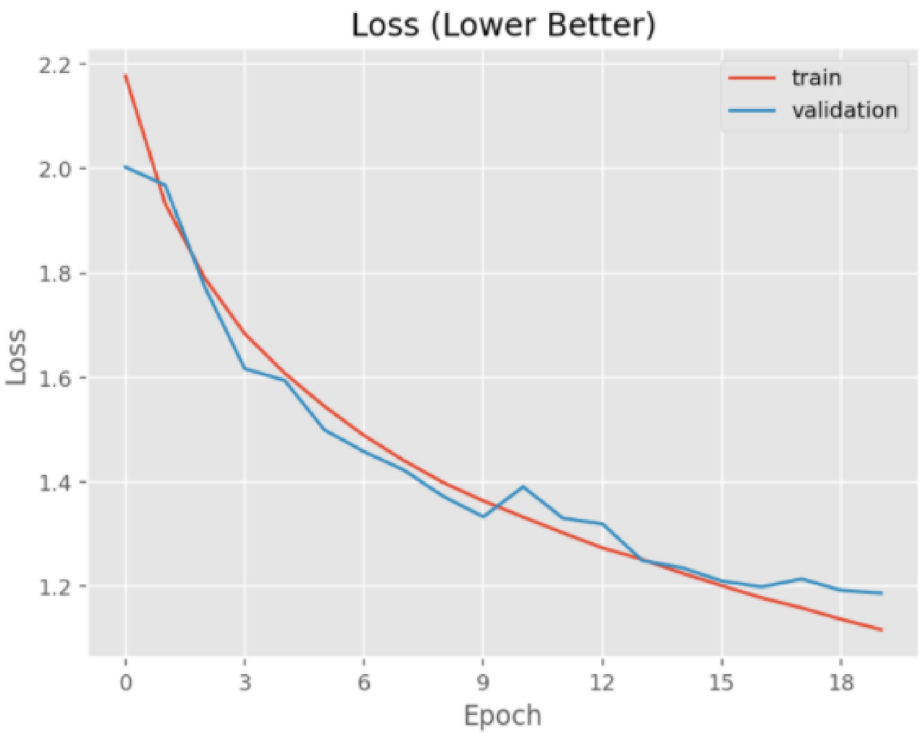
- Model: MLP of 2 Dense 128 neuron layers, dropout (0.2 & 0.4) and batch normalisation applied to reduce overfitting
- Optimiser: adam
- **Best accuracy at 49% (MLP Optimised)**
- **(Early stopping at epoch 11, each epoch 11s, batch size 100)**

```
Test loss: 1.457458257675171
Test accuracy: 0.48910000920295715
```

Model 2: LeNet-5 CNN

- The LeNet model is implemented with 3 convolutional blocks (6-16-120, k=5) and 2 dense layers with SGD or adam optimisers. The accuracy ranges from 60 to 69%

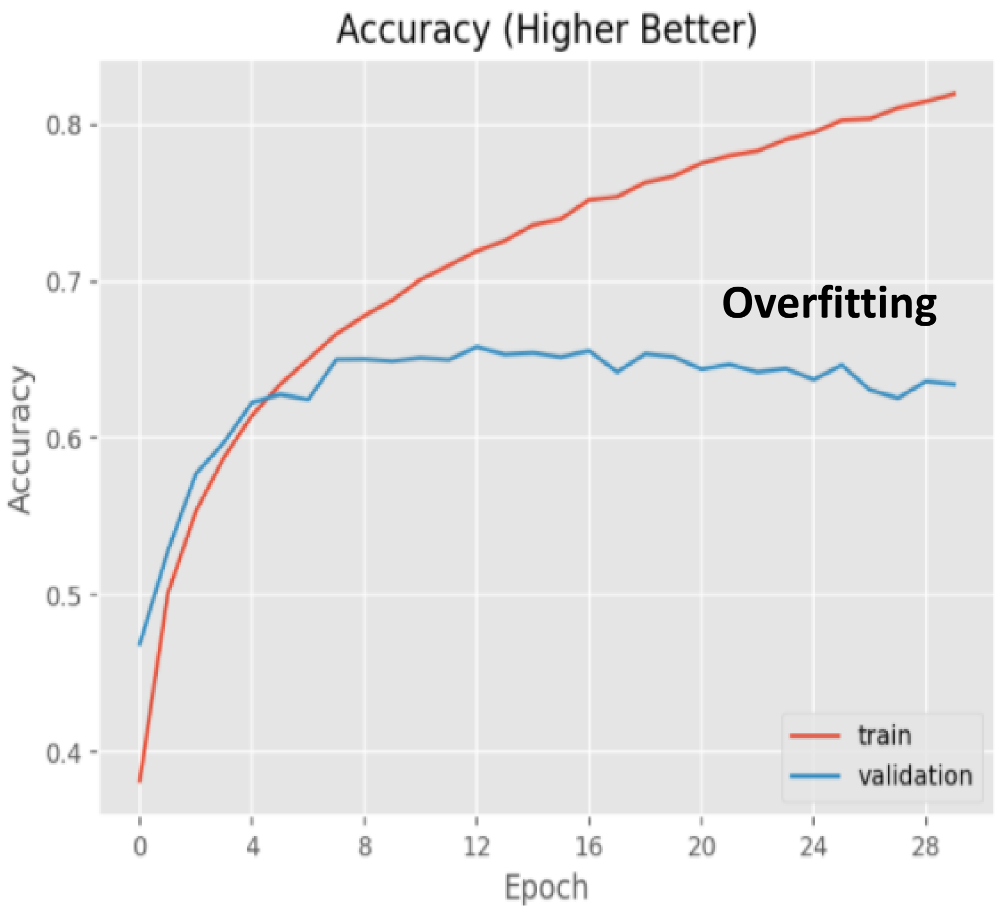


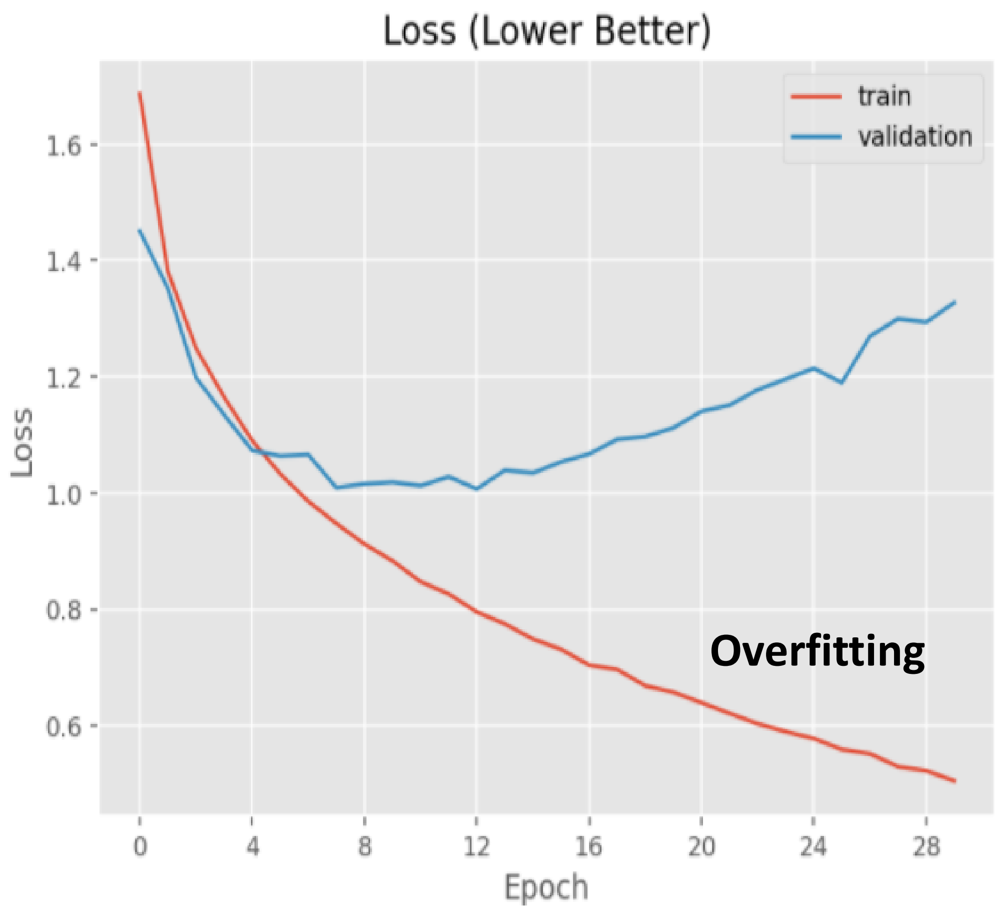


- Model: LeNet-5 Baseline
- Optimiser: SGD
- **Best accuracy at 57% (LeNet Baseline)**

```
Test loss: 1.2140653133392334
Test accuracy: 0.5728999972343445
```

Model 2: LeNet-5 CNN

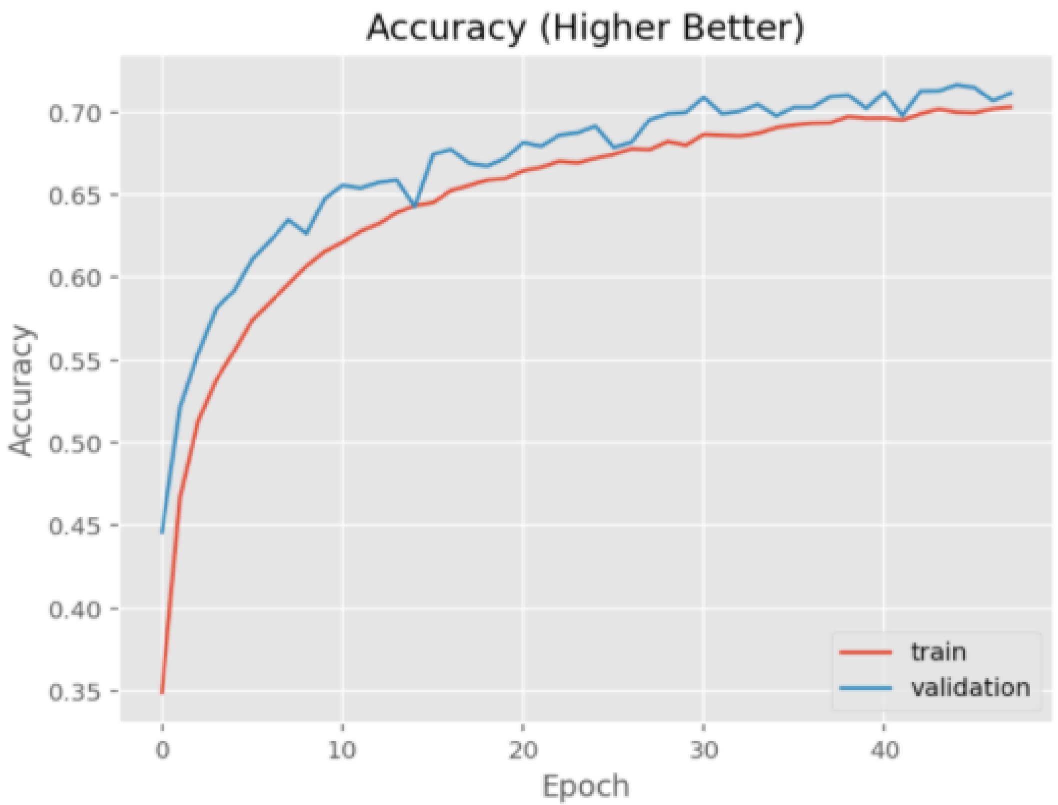


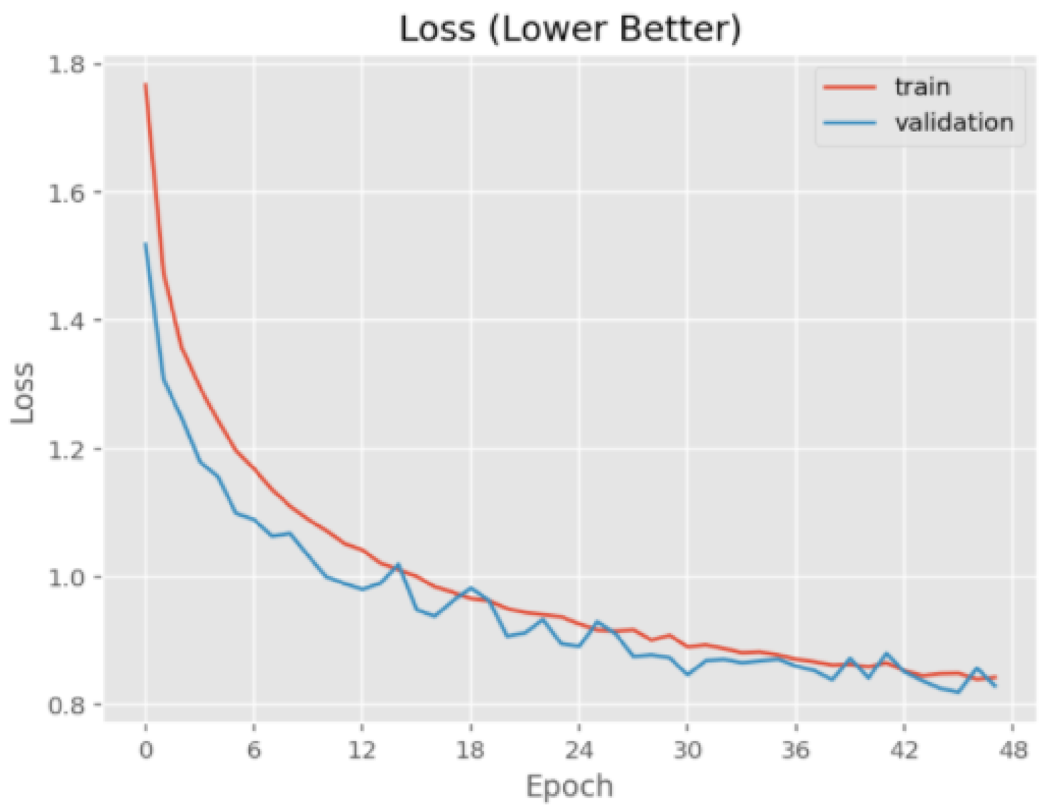


- Model: LeNet-5
- Optimiser: adam
- **Accuracy at 62% (with adam optimiser)**

```
Test loss: 1.4061318635940552
Test accuracy: 0.621399998664856
```

Model 2: LeNet-5 CNN



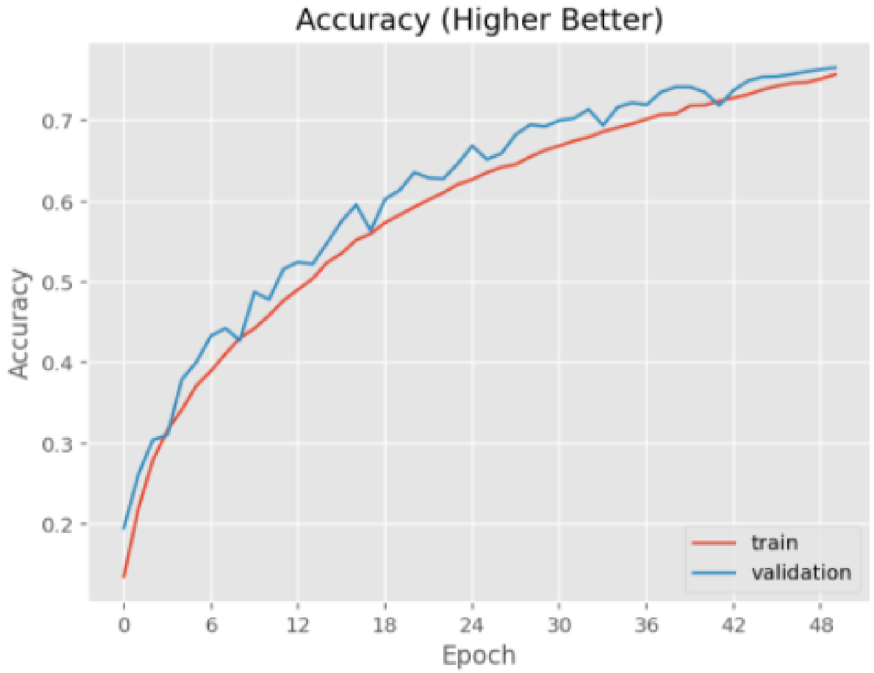


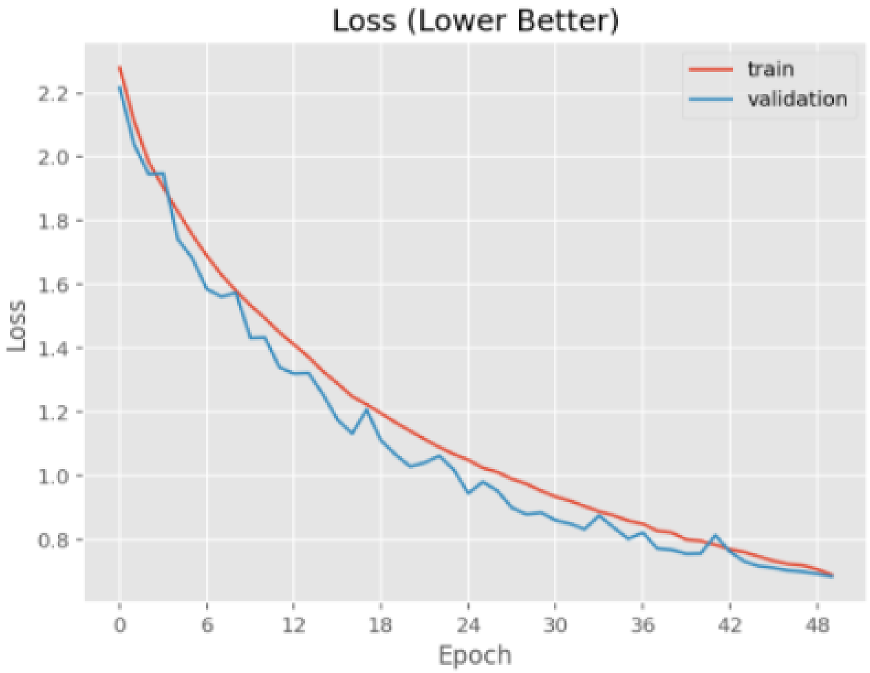
- Model: LeNet-5, with 2 dropout layers 0.2, 0.4 added to avoid overfitting
- Optimiser: adam
- **Best accuracy at 69% (LeNet optimised)**
- **Early stopping at epoch 48, each epoch 2s**

```
Test loss: 0.8936426639556885
Test accuracy: 0.6919999718666077
```

Model 3: VGG 3 Block

- The VGG model is implemented with 3 convolutional blocks (32-64-128, double layers, k=3) and 2 dense layers with SGD (with learning rate, momentum tuning) or adam optimisers. The accuracy ranges from 76 to 85%

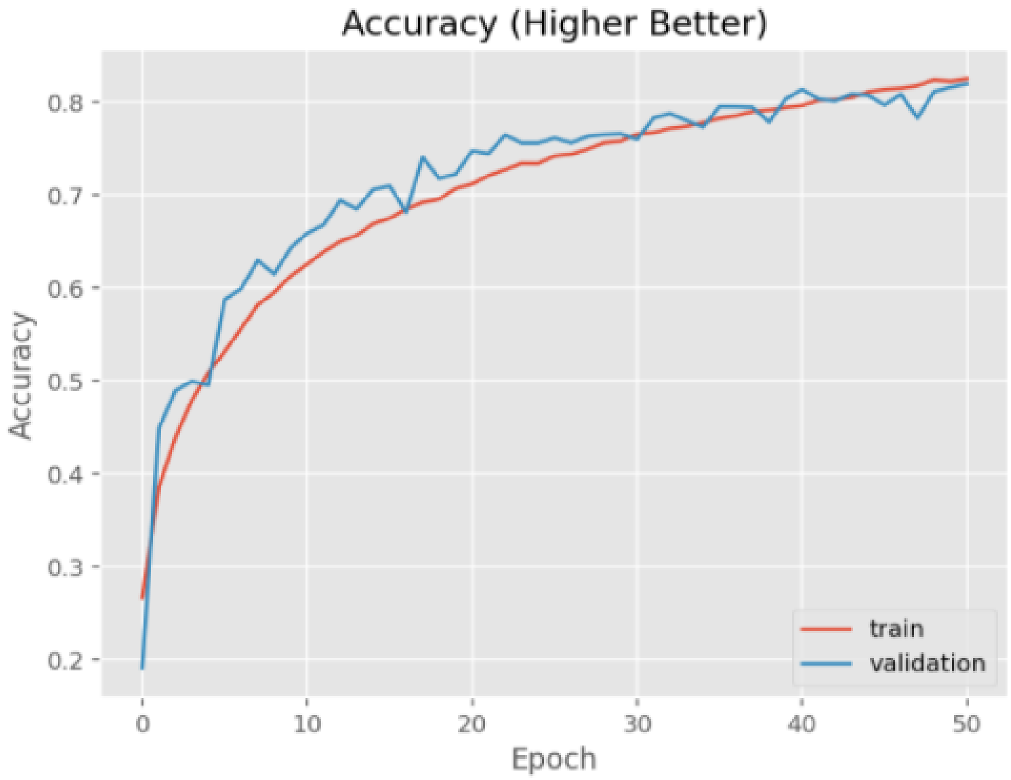


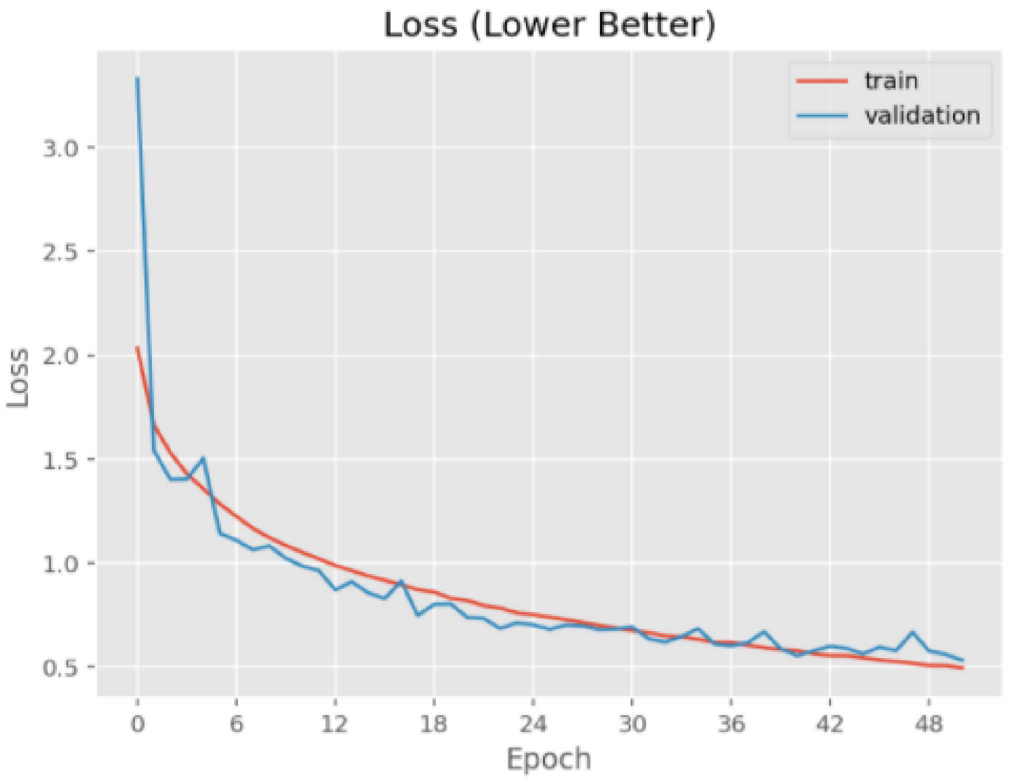


- Model: VGG 3 Block Baseline
- Optimiser: SGD
- **Best accuracy at 76% (VGG Baseline)**

```
Test loss: 0.70351344347000012
Test accuracy: 0.7601000070571899
```

Model 3: VGG 3 Block

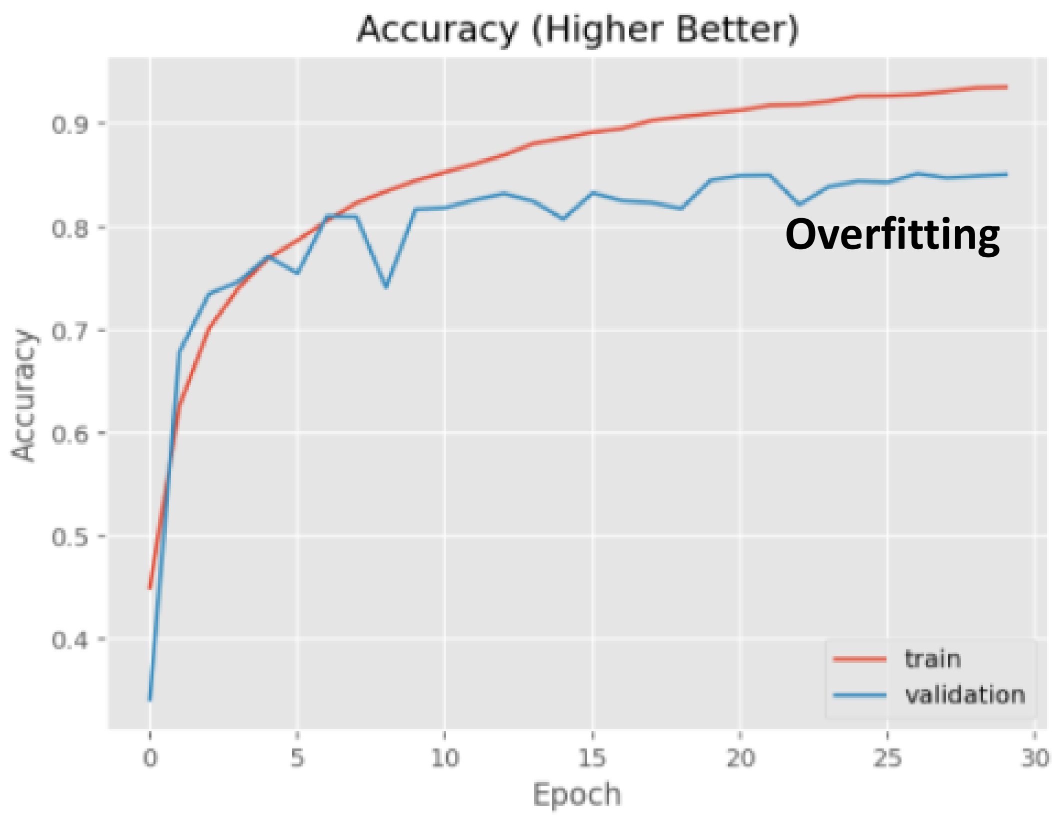


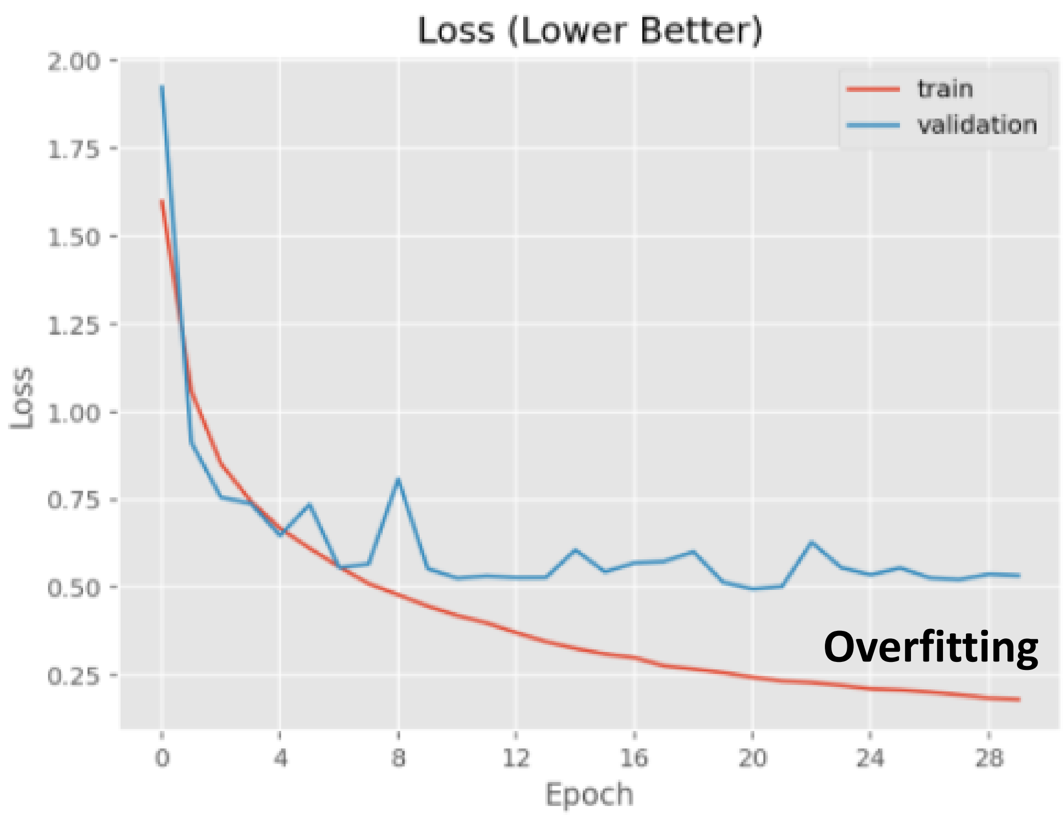


- Model: VGG 3 Block
- Optimiser: SGD with learning rate 0.001, momentum 0.9
- **Accuracy at 80% , early stop 51 epoch**

```
Test loss: 0.5660185813903809
Test accuracy: 0.8047999739646912
```

Model 3: VGG 3 Block

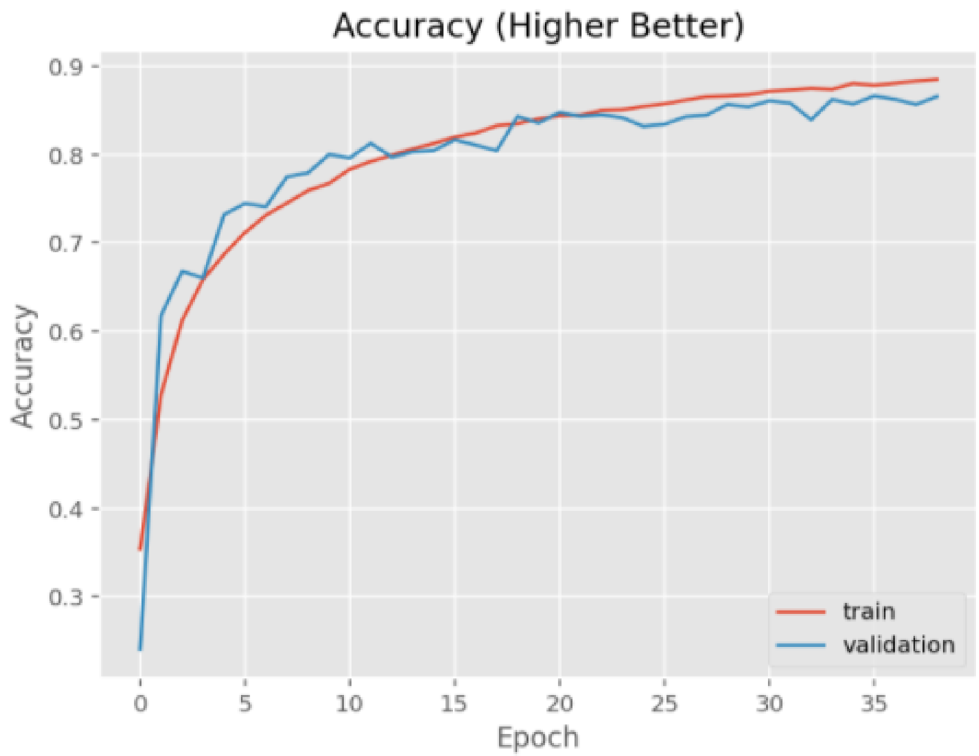


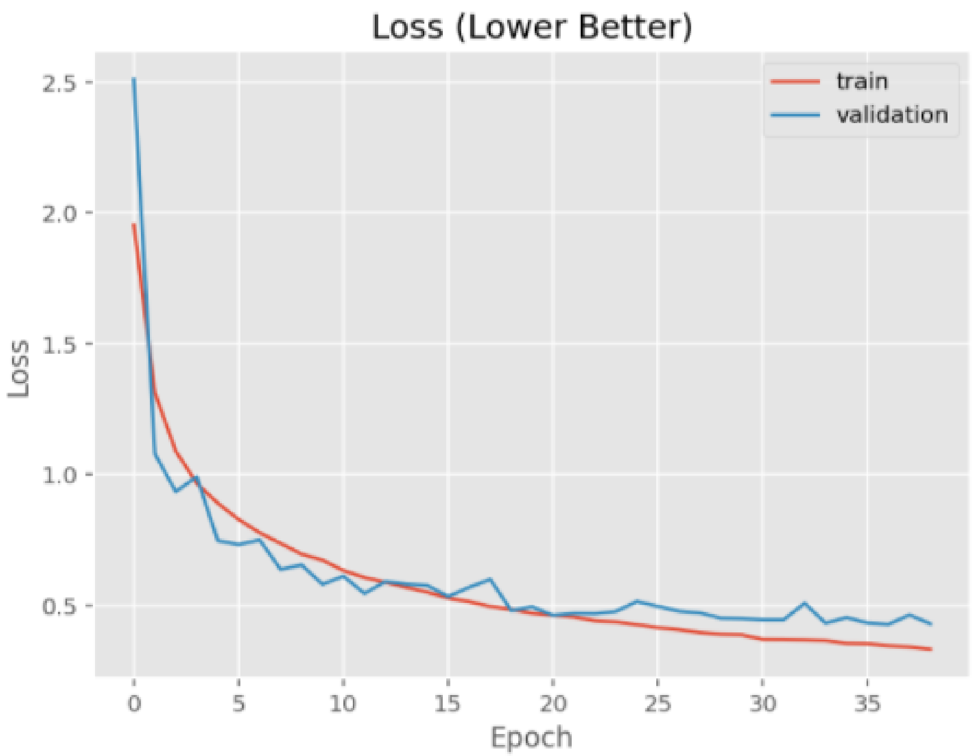


- Model: VGG 3 Block
- Optimiser: adam, with batch normalisation after each VGG block & dense layer
- **Accuracy at 83% , early stop 37 epoch**

```
Test loss: 0.5596263408660889
Test accuracy: 0.8305000066757202
```

Model 3: VGG 3 Block



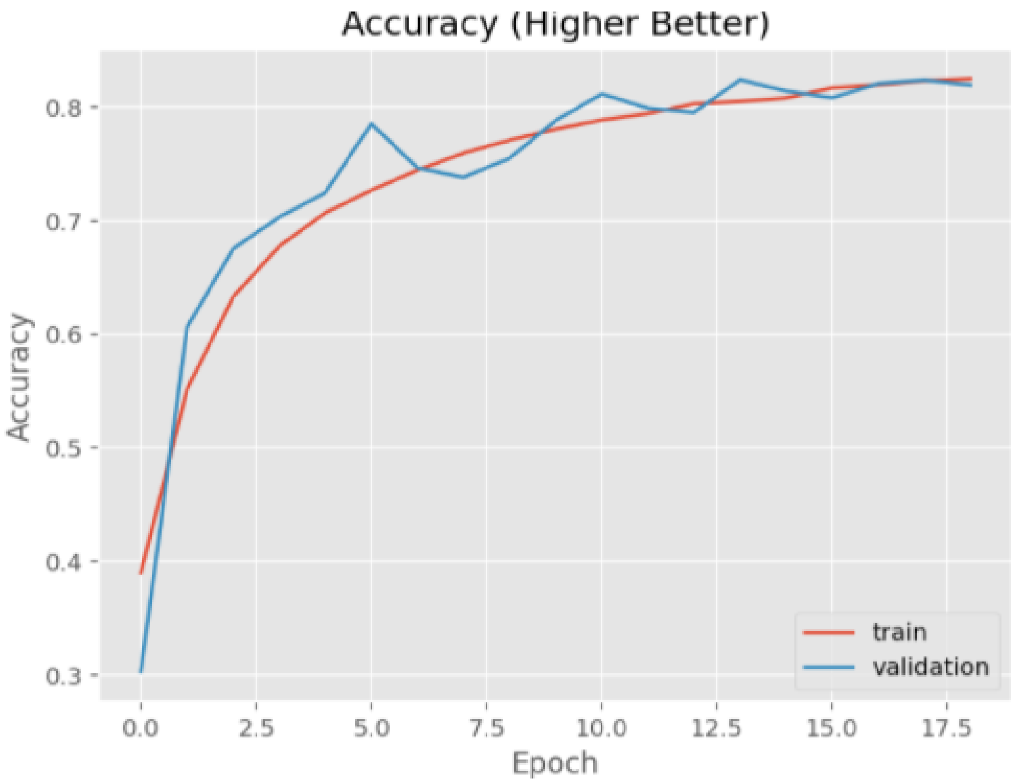


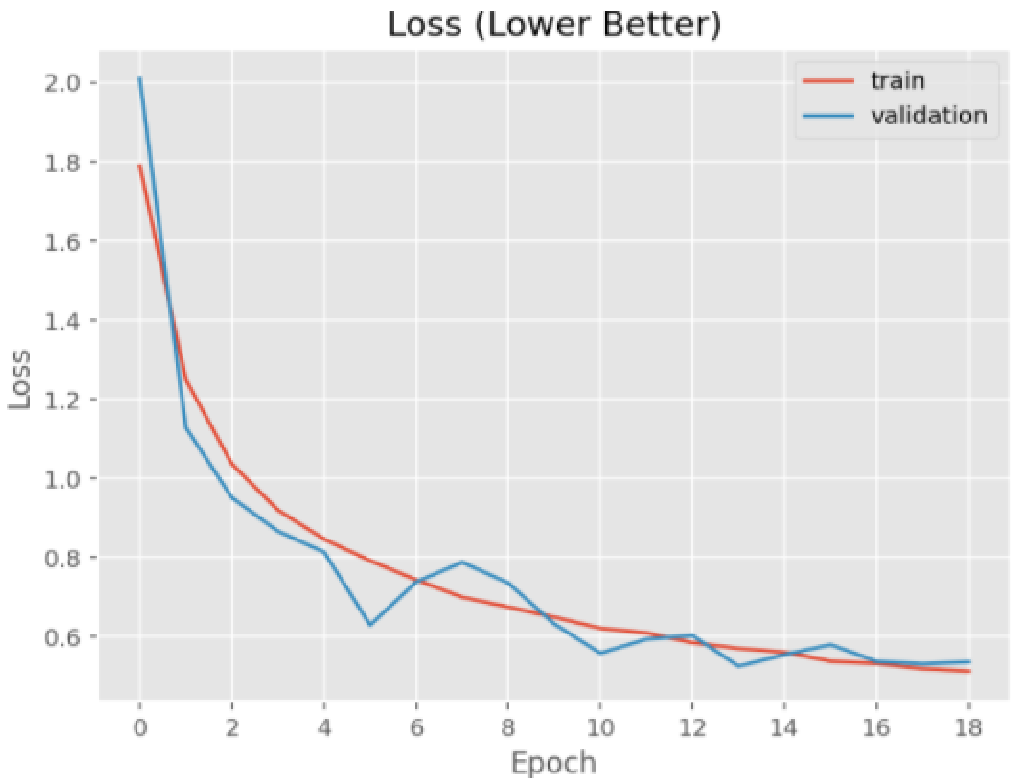
- Model: VGG 3 Block
- Optimiser: adam, with batch normalisation after each VGG layers & dense layer, plus dropout layers added at 0.3,0.4,0.5 & 0.6
- **Best Accuracy at 85% , early stop 39 epoch, epoch 6s (Optimised VGG model)**

```
Test loss: 0.44637230038642883
Test accuracy: 0.8515999913215637
```

Model 4: Data Augmentation & VGG-3

- Randomised Data augmentation used on optimised VGG-3 architecture. Height & width shift 0.1 & horizontal flip.



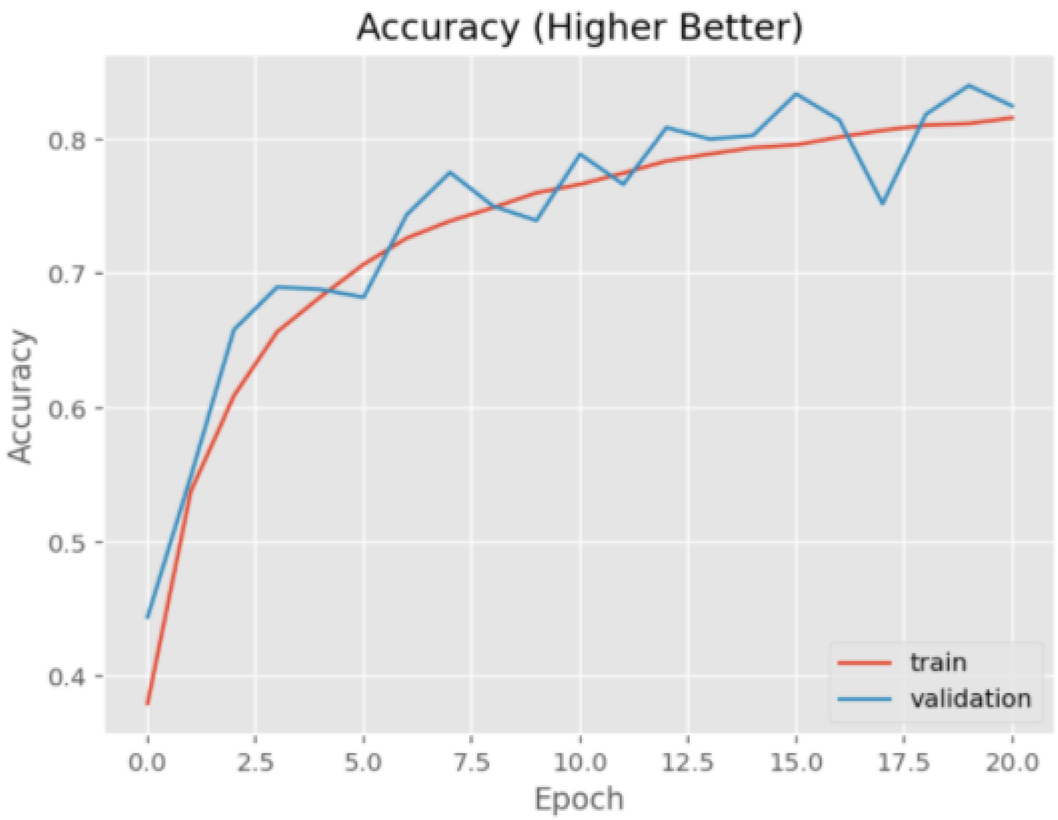


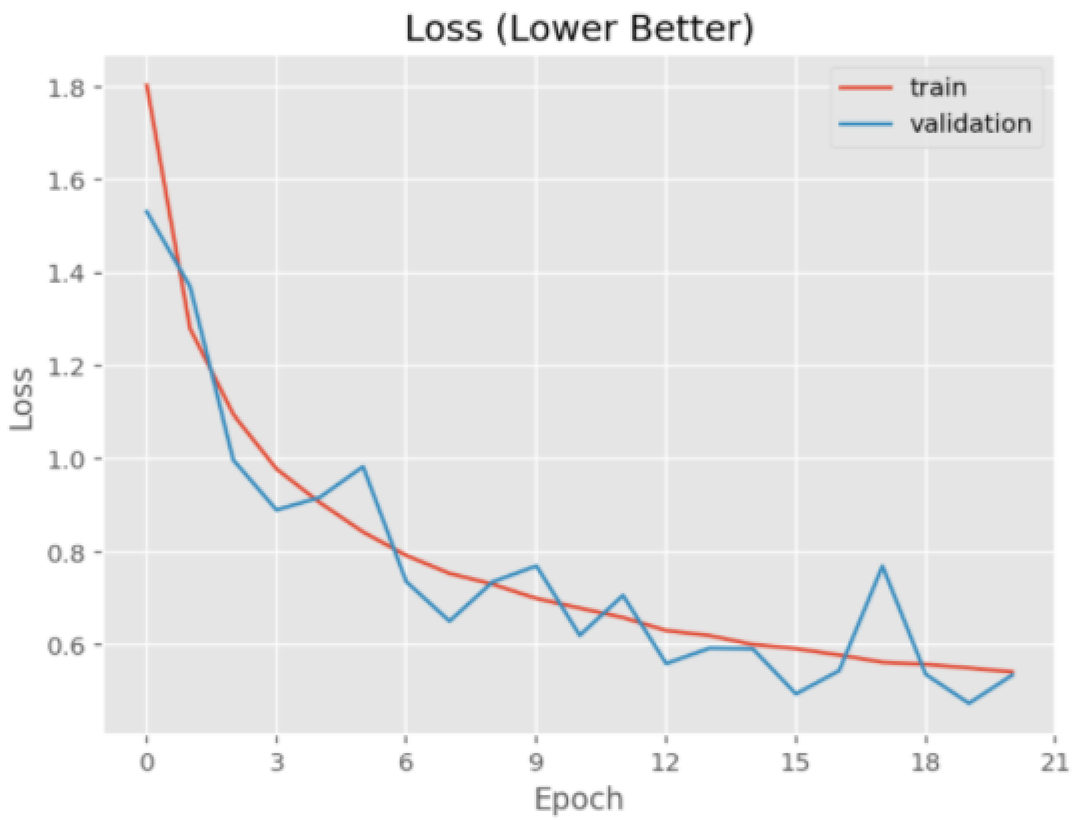
- Model: VGG 3 & Data Augmentation
- Optimiser: adam
- **Accuracy at 82%**

```
Test loss: 0.5262131094932556
Test accuracy: 0.8238000273704529
```

Model 4: Data Augmentation & VGG-3

- Increased transformation of rotation angle 15 added





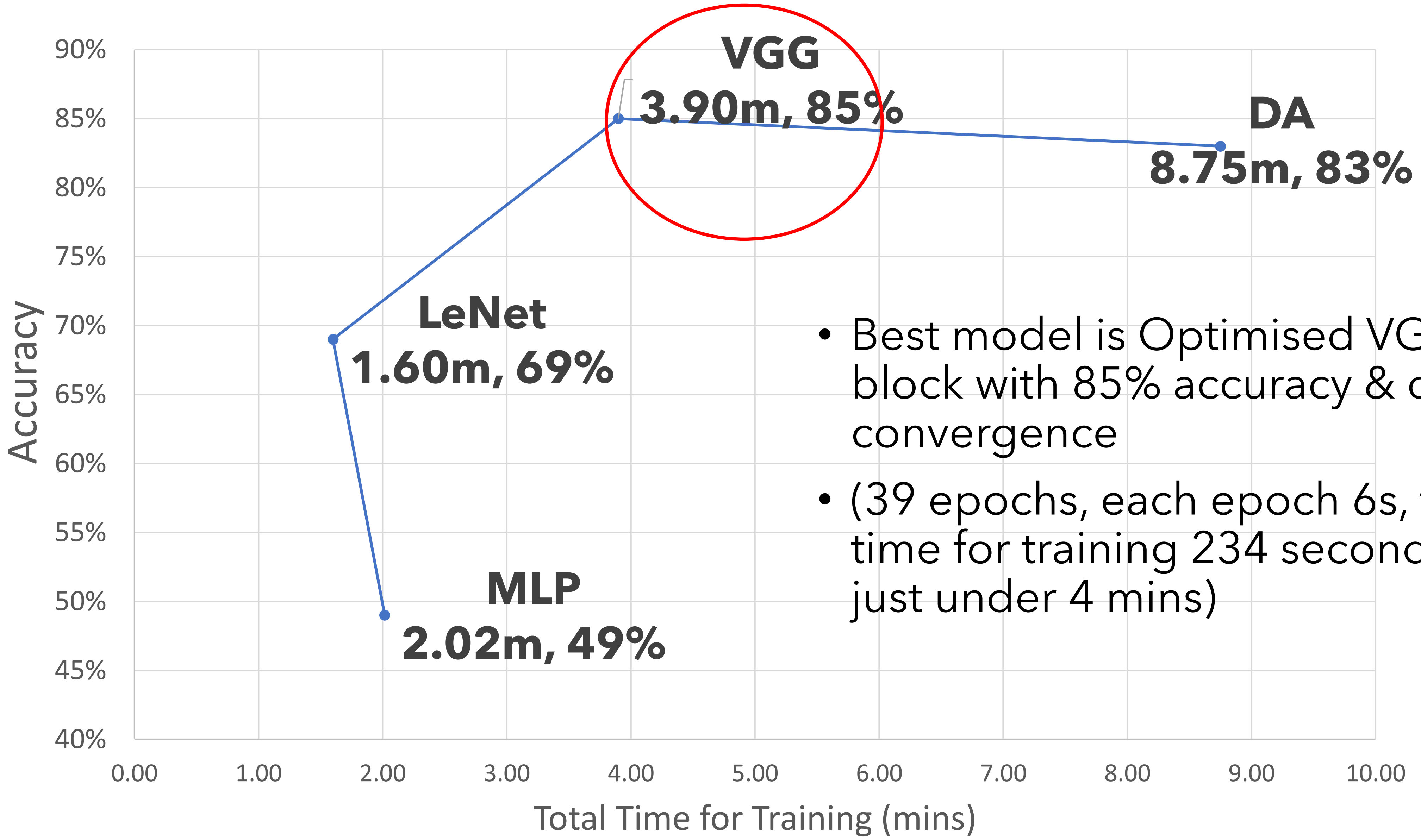
- Model: VGG 3 & Data Augmentation
- Optimiser: adam
- **Best Accuracy at 83%, early stop 21, epoch 25s**

```
Test loss: 0.4953373074531555
Test accuracy: 0.8342999815940857
```

Model Comparison

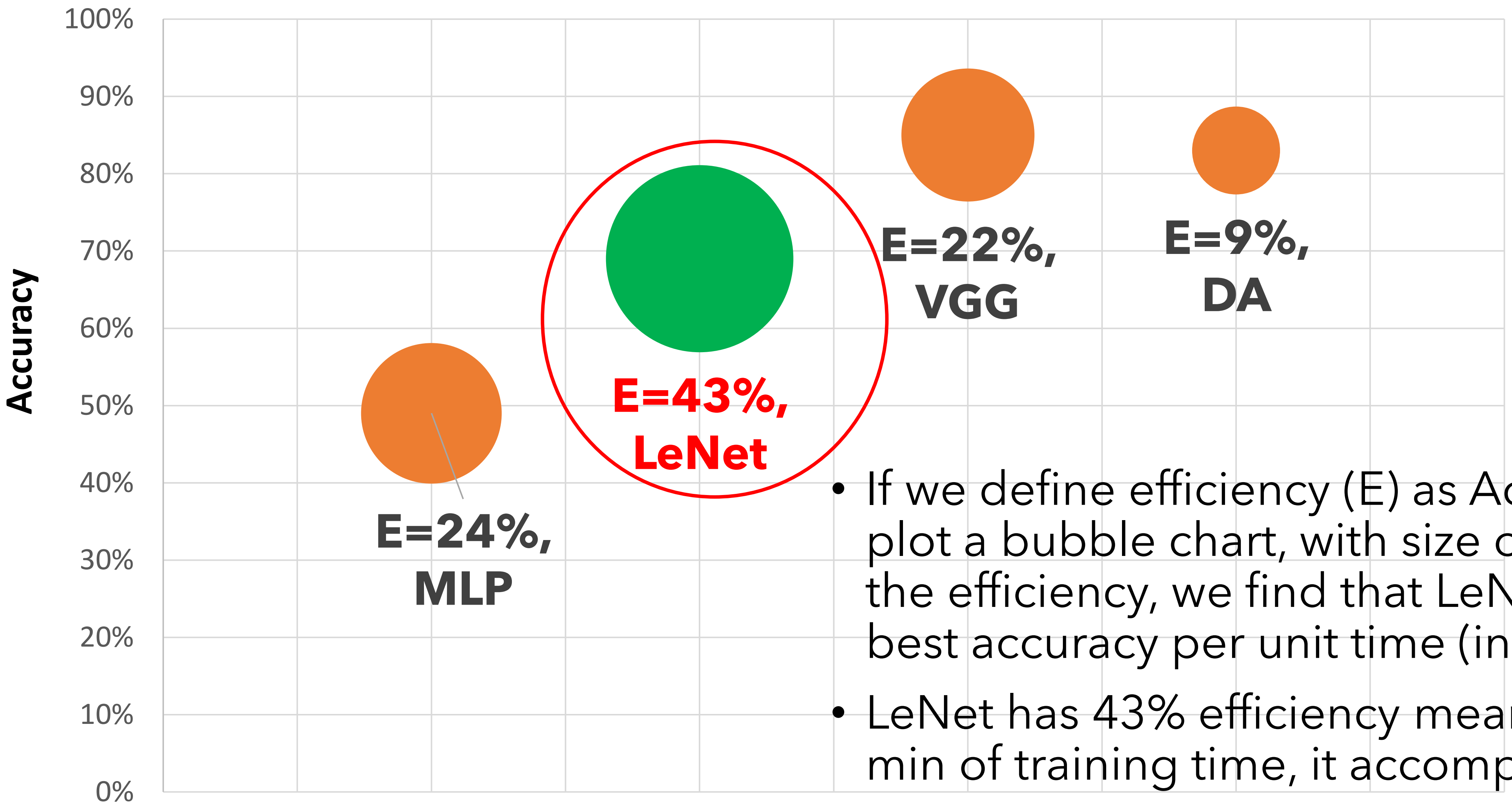
Model	Key Parameters	Accuracy	Early Stop/ Epoch	Epoch Time	Remarks
MLP	2 Dense Layers of 128 neurons Dopout layers 0.2,0.4 & batch normalisation Adam optimiser	49%	11	11s	Optimised MLP model with quick convergence after 11 epochs
LeNet-5	3 Conv layers (6,16,120) & 2 dense layers (84,10) Dropout 0.2, 0.4 Adam optimiser	69%	48	2s	**Optimised LeNet model with extremely quick per epoch time**
VGG-3	**3 convolutional blocks (32, 64,128 double layers) and 2 dense layers (128,10)** **Dropouts 0.3,0.4,0.5,0.6 & batch normalisation**	**85%**	**39**	**6s**	**Optimised VGG-3 model with highest accuracy with relatively quick convergence & epoch time**
Data Augmt & VGG	VGG architecture Randomised transformation Height, width shifts, rotation & horizontal flip	83%	21	25s	Marginal effect on accuracy. Each epoch time increased to 25s on-the-fly data augmentation

Model Comparison



- Best model is Optimised VGG-3 block with 85% accuracy & quick convergence
- (39 epochs, each epoch 6s, total time for training 234 seconds or just under 4 mins)

Model Comparison



- If we define efficiency (E) as Accuracy/Time plot a bubble chart, with size of bubble as the efficiency, we find that LeNet offers the best accuracy per unit time (in mins).
- LeNet has 43% efficiency meaning that for 1 min of training time, it accomplished 43% accuracy

Conclusion

- Computer resources are limited & best model should deliver high performance (accuracy) with fewer epochs & shorter epoch time
- On hyperparameters, adam optimiser performs well & batch normalisation, dropout and pooling reduce overfitting and help model to converge quickly
- LeNet produces higher accuracy at 69% compared to MLP at 49% even with a shorter training time (MLP 2 mins vs LeNet 1.6 mins) showing that convolutional layers CNN architecture offers higher performance for images classification
- **Best model is the Optimised VGG-3 block with 85% accuracy & quick convergence** (39 epochs, each epoch 6s, total training time of 234 seconds or less than 4 mins)
- **LeNet offers the best efficiency** (in terms of accuracy per unit of training time) and achieved 69% accuracy in just about one and half mins
- Data augmentation has marginal effect (slight drop of accuracy to 83%) as CIFAR 10 dataset is relatively "standardised" across the training and test sets