Deep Learning Exercise 3

Yefim Nudelman
ID: 209342096

Guy Baruch
ID: 207707472

Dataset split

We split the available training data into an 80/20 train/validation split (0.8/0.2), following standard practice for model selection and hyperparameter tuning.

Part 1 - Visualize the Data

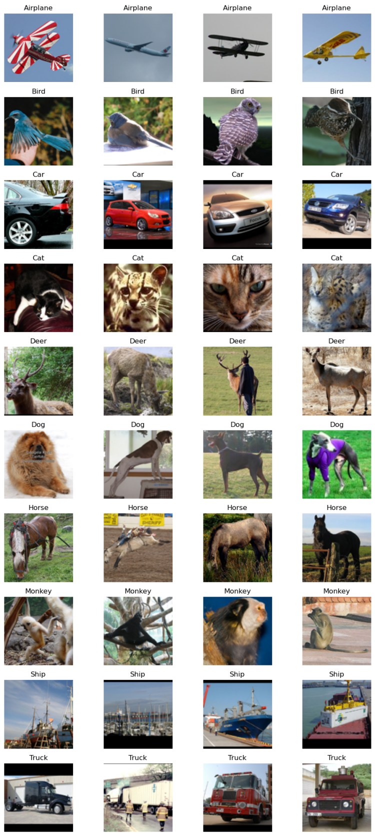

Figure 1: (Part 1) Visualization of classes.

Part 2

Data augmentations

We used the following augmentations:

- Random horizontal flip with probability 0.5.
- Random rotation up to 15°.
- Color jitter with brightness, contrast, and saturation jitter of 0.2, and hue jitter of 0.1.

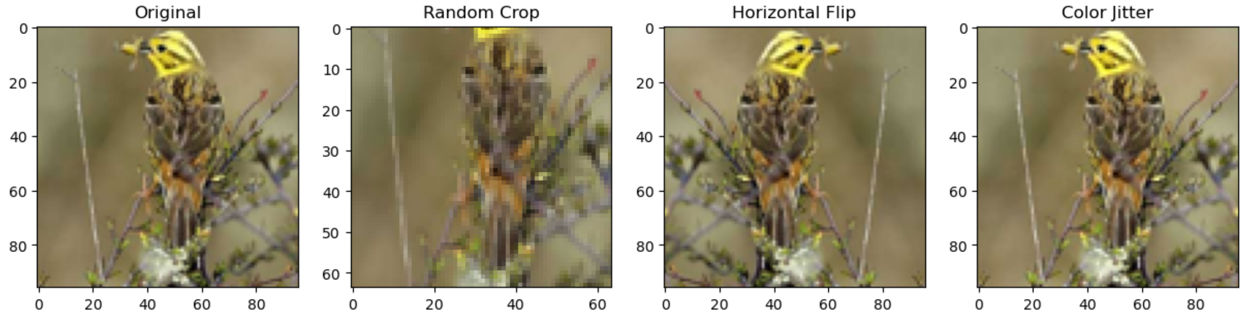

Figure 2: Examples of the applied data augmentations.

Logistic regression hyperparameter search

Grid search parameters

Hyperparameter	**Values searched**
Optimizer	Adam, SGD
Learning rate	10^{-2}, 10^{-3}, 10^{-4}
Batch size	64, 128
Weight decay	0, 10^{-3}, 10^{-4}
Epochs (grid search)	15

Table 1: Logistic regression grid-search hyperparameter ranges.

Top 10 configurations

Rank	Val. Acc.	Opt.	LR	Batch	WD
1	28.9%	Adam	10^{-4}	64	0
2	28.9%	Adam	10^{-4}	64	10^{-3}
3	28.9%	Adam	10^{-4}	64	10^{-4}
4	28.6%	Adam	10^{-4}	128	0
5	28.6%	Adam	10^{-4}	128	10^{-3}
6	28.6%	Adam	10^{-4}	128	10^{-4}
7	28.6%	SGD	10^{-4}	64	0
8	28.6%	SGD	10^{-4}	64	10^{-3}
9	28.6%	SGD	10^{-4}	64	10^{-4}
10	28.3%	SGD	10^{-4}	128	0

Table 2: Top 10 logistic regression configurations from the grid search.

Main insights

The best logistic regression results are obtained with a small learning rate (10^{-4}); all top-10 configurations use 10^{-4} regardless of optimizer, batch size, or weight decay. Validation accuracy is relatively flat across weight decay and batch size, and Adam and SGD perform similarly at this best learning rate.

All configurations in this grid search were evaluated for 15 epochs to quickly compare hyperparameters.

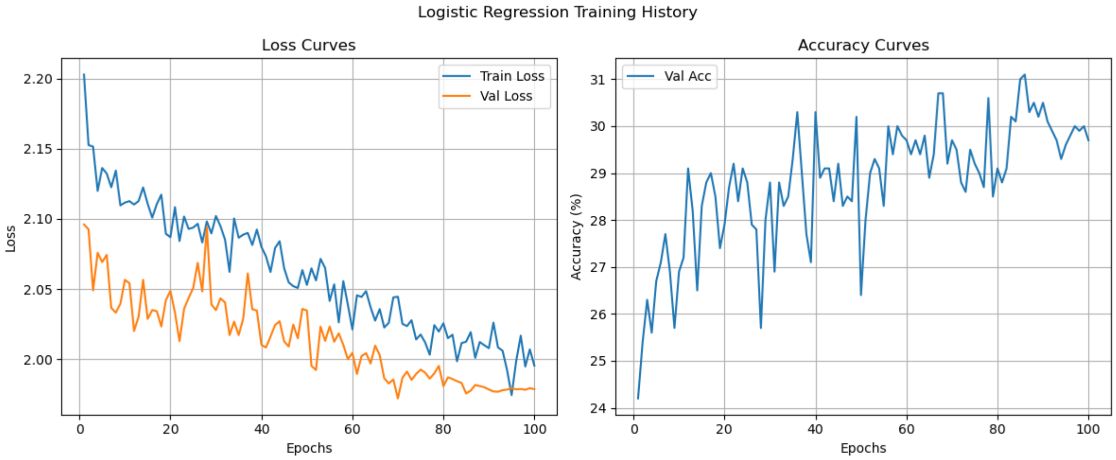

Figure 3: Logistic regression training/validation curves using the selected hyperparameters.

The final logistic regression model achieves a test accuracy of 30.62% on the 8000 test images.

Neural network hyperparameter search

Grid search parameters

Hyperparameter	**Values searched**
Hidden dimension	512, 1024
Optimizer	Adam, SGD
Learning rate	10^{-2}, 10^{-3}, 10^{-4}
Batch size	32, 64
Weight decay	0, 10^{-3}, 10^{-4}
Epochs (grid search)	15

Table 3: Neural network grid-search hyperparameter ranges.

Top 10 configurations

Rank	**Val. Acc.**	**Hidden**	**Opt.**	**LR**	**Batch**	**WD**
1	37.8%	1024	Adam	10^{-3}	64	0
2	37.0%	1024	SGD	10^{-2}	64	0
3	36.9%	512	Adam	10^{-3}	64	10^{-4}
4	36.8%	1024	Adam	10^{-4}	64	10^{-3}
5	36.7%	512	SGD	10^{-2}	64	10^{-4}
6	36.7%	1024	Adam	10^{-3}	32	0
7	36.7%	1024	SGD	10^{-2}	64	10^{-4}
8	36.5%	1024	Adam	10^{-4}	64	10^{-4}
9	36.5%	1024	SGD	10^{-3}	64	10^{-3}
10	36.4%	1024	Adam	10^{-4}	32	0

Table 4: Top 10 neural network configurations from the grid search.

Main insights

The best configurations favor a larger hidden dimension (1024 appears in 8 of the top 10) and relatively aggressive learning rates (10^{-3} or 10^{-2}). Both Adam and SGD can reach similar validation accuracy, with the top run using Adam at learning rate 10^{-3}.

All configurations in this grid search were evaluated for 15 epochs to quickly compare hyperparameters.

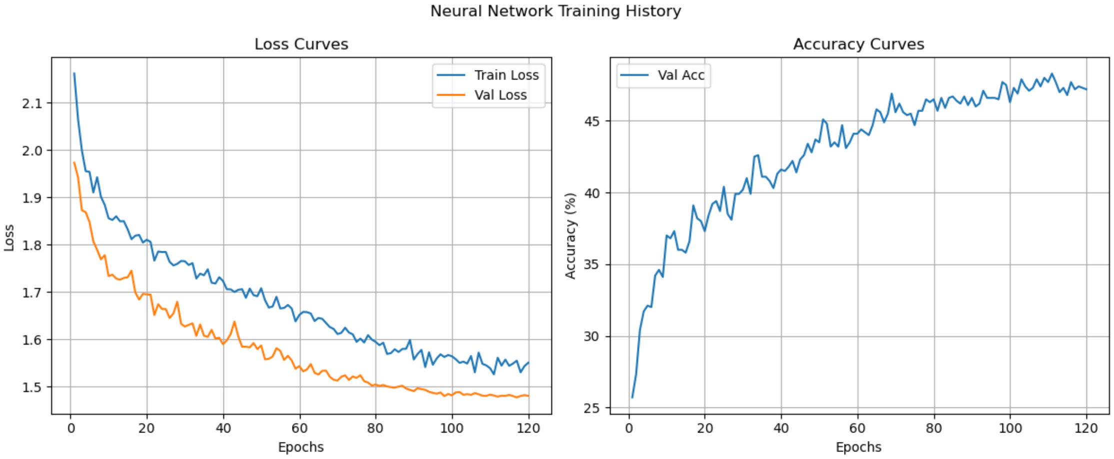

Figure 4: Neural network training/validation curves using the selected hyperparameters.

The final neural network model achieves a test accuracy of 46.02% on the 8000 test images.

CNN hyperparameter search

Grid search parameters

Hyperparameter	**Values searched**
Base width	16, 32
Optimizer	Adam, SGD
Batch size	32, 64
Weight decay	10^{-3}, 10^{-4}
Learning rate	10^{-2}, 10^{-3}, 5×10^{-4}
Epochs (grid search)	15

Table 5: CNN grid-search hyperparameter ranges.

Top 10 configurations

Rank	**Val. Acc.**	**Width**	**Opt.**	**Batch**	**WD**	**LR**
1	47.3%	32	SGD	32	10^{-4}	10^{-3}
2	46.4%	16	SGD	64	10^{-4}	10^{-2}
3	45.2%	32	SGD	64	10^{-3}	10^{-2}
4	44.4%	32	SGD	32	10^{-3}	10^{-3}
5	44.4%	32	SGD	64	10^{-4}	10^{-2}
6	44.3%	16	Adam	64	10^{-4}	5×10^{-4}
7	43.4%	16	SGD	64	10^{-3}	10^{-2}
8	43.1%	32	Adam	32	10^{-4}	5×10^{-4}
9	42.8%	16	SGD	32	10^{-4}	10^{-2}
10	42.8%	32	Adam	64	10^{-4}	5×10^{-4}

Table 6: Top 10 CNN configurations from the grid search.

Main insights

The top ranks are dominated by SGD (7 of the top 10), and the best validation accuracy (47.3%) is achieved with SGD using base width 32, batch size 32, weight decay 10^{-4}, and learning rate 10^{-3}. The learning rates 10^{-3} and 10^{-2} show up repeatedly among the best runs, while Adam only appears in 3 of the top 10 configurations (typically with learning rate 5×10^{-4}).

All configurations in this grid search were evaluated for 15 epochs to quickly compare hyperparameters; the final model training is run for a longer schedule.

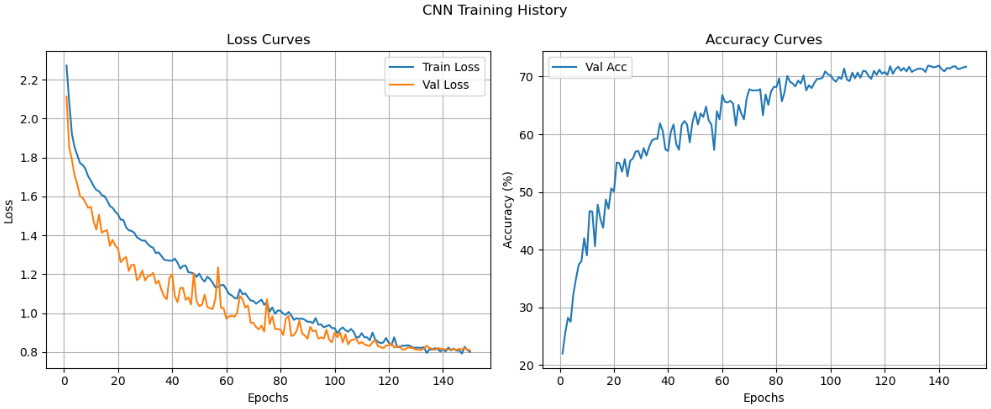

Figure 5: CNN final training curves (longer training schedule) using the selected hyperparameters.

The final trained CNN achieves a test accuracy of 73.86% on the 8000 test images.

MobileNet (frozen backbone) hyperparameter search

Grid search parameters

Hyperparameter	**Values searched**
Freeze backbone	True
Weight decay	10^{-3}, 10^{-4}
Hidden dimension	256, 512
Optimizer	Adam, SGD
Learning rate	10^{-2}, 10^{-3}, 10^{-4}
Batch size	32, 64
Epochs (grid search)	15

Table 7: MobileNet (frozen) grid-search hyperparameter ranges.

Top 10 configurations

Rank	**Val. Acc.**	**Hidden**	**Opt.**	**LR**	**Batch**	**WD**
1	76.5%	256	SGD	10^{-2}	64	10^{-3}
2	76.4%	256	Adam	10^{-3}	32	10^{-3}
3	76.1%	512	Adam	10^{-3}	64	10^{-3}
4	76.1%	512	Adam	10^{-3}	64	10^{-4}
5	76.0%	256	Adam	10^{-3}	64	10^{-4}
6	75.7%	512	Adam	10^{-4}	64	10^{-3}
7	75.6%	256	Adam	10^{-3}	64	10^{-3}
8	75.6%	256	Adam	10^{-4}	32	10^{-3}
9	75.6%	512	Adam	10^{-4}	32	10^{-3}
10	75.6%	512	SGD	10^{-2}	64	10^{-3}

Table 8: Top 10 MobileNet (frozen) configurations from the grid search.

Main insights

Freezing the backbone yields substantially higher validation accuracy than the earlier baselines; the best configuration reaches 76.5% validation accuracy. The top results favor weight decay 10^{-3} and either SGD with learning rate 10^{-2} or Adam with learning rate 10^{-3}, and both hidden dimensions (256 and 512) can perform well.

All configurations in this grid search were evaluated for 15 epochs to quickly compare hyperparameters.

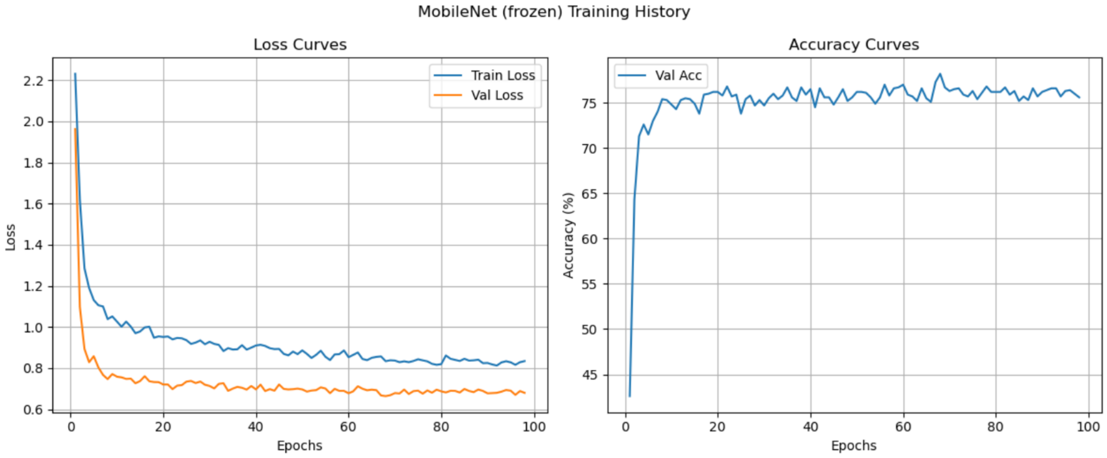

Figure 6: MobileNet (frozen backbone) final training curves using the selected hyperparameters.

The final trained MobileNet (frozen backbone) achieves a test accuracy of 76.80% on the 8000 test images.

MobileNet (fine-tuned) hyperparameter search

Grid search parameters

Hyperparameter	**Values searched**
Freeze backbone	False
Weight decay	10^{-3}, 10^{-4}
Hidden dimension	256, 512
Optimizer	Adam, SGD
Learning rate	10^{-2}, 10^{-3}, 10^{-4}
Batch size	32, 64
Epochs (grid search)	15

Table 9: MobileNet (fine-tuned) grid-search hyperparameter ranges.

Top 10 configurations

Rank	**Val. Acc.**	**Hidden**	**Opt.**	**LR**	**Batch**	**WD**
1	85.6%	256	Adam	10^{-4}	64	10^{-3}
2	85.6%	256	Adam	10^{-4}	64	10^{-4}
3	85.1%	512	Adam	10^{-4}	64	10^{-3}
4	85.0%	256	Adam	10^{-4}	32	10^{-3}
5	85.0%	512	Adam	10^{-4}	32	10^{-3}
6	85.0%	512	Adam	10^{-4}	32	10^{-4}
7	84.9%	256	Adam	10^{-4}	32	10^{-4}
8	84.6%	512	Adam	10^{-4}	64	10^{-4}
9	83.8%	256	SGD	10^{-2}	64	10^{-3}
10	83.8%	512	SGD	10^{-2}	64	10^{-4}

Table 10: Top 10 MobileNet (fine-tuned) configurations from the grid search.

Main insights

Fine-tuning the backbone improves validation accuracy substantially, reaching up to 85.6%. The top configurations overwhelmingly favor Adam with a small learning rate (10^{-4}); SGD configurations with learning rate 10^{-2} appear in the top 10 but underperform the best Adam runs. Both hidden dimensions (256 and 512) can work well, and both weight decay settings (10^{-3} and 10^{-4}) appear among the best results.

All configurations in this grid search were evaluated for 15 epochs to quickly compare hyperparameters.

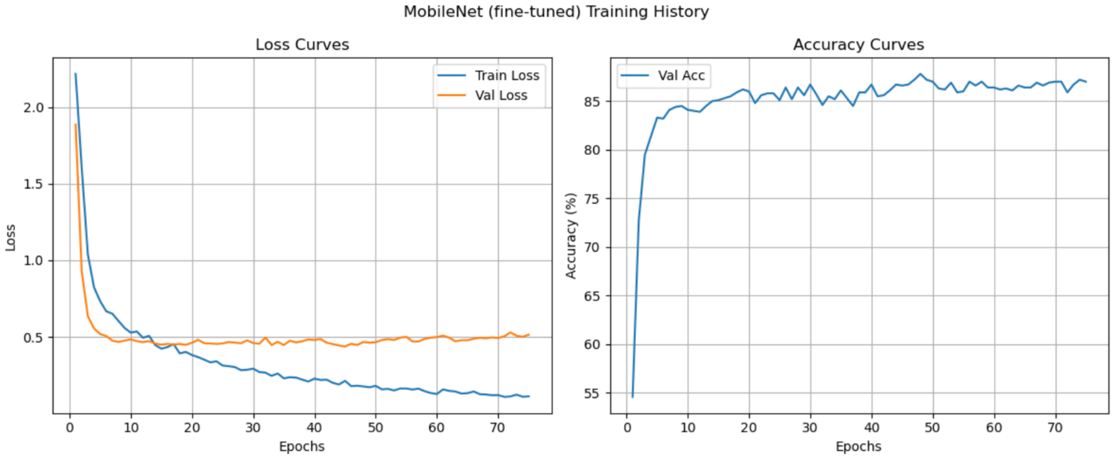

Figure 7: MobileNet (fine-tuned) final training curves using the selected hyperparameters.

The final trained MobileNet (fine-tuned) achieves a test accuracy of 87.81% on the 8000 test images.

How to Run the Code

To reproduce the results and figures shown in this report:

1. Open the project notebook by following this url https://colab.research.google.com/drive/1xdvZisZABRoAYy_ZLh7xfD6Zo-Ol7PRG?usp=sharing
2. uncomment relevant cells
3. Run all cells sequentially from top to bottom.

The notebook uses the project's Python modules to:

- Download and load the STL-10 dataset
- Split the training set into train and validation sets (80/20)
- Apply data augmentation during training and center cropping during testing
- Train and evaluate all required models
- Perform hyperparameter searches
- Generate the training and validation curves and final evaluation plots

Hardware

The code supports both CPU and GPU execution. If a GPU is available, it will be used automatically; otherwise, the code runs on CPU.