Effect of Architectural and Training Hyper-parameters on Fashion-MNIST Test Accuracy

Lillian Li (32198314)   David Ewing (82171149)

2025-05-13

Abstract

We reproduced the convolutional-network template from the 2025 STAT448 Keras labs and trained ten configurations on the Fashion-MNIST data set. We compare the impact of filter width, kernel size, dense-layer width, optimiser and batch size on convergence speed and final test accuracy. The best model (Config 1) reached **91.1 %** test accuracy after five epochs, whereas the weakest model (Config 10) plateaued at 85.4 %. Larger filter banks and the *Adam* optimiser consistently out-performed smaller networks and *SGD*.

1 Introduction

The original Keras labs for STAT448 introduced a small CNN for MNIST. Fashion-MNIST is a harder, ten-class image-classification benchmark of identical spatial dimension (Xiao et al., 2017). This report extends the lab code to examine how architectural and solver hyper-parameters influence generalisation.

2 Methods

Data set

The Fashion-MNIST data set, consisting of 60 000 training images and 10 000 test images of $28 \times 28 \times 1$ greyscale clothing items, was originally introduced by Xiao et al. (2017) and is accessible via the Keras helper routine `keras.datasets.fashion_mnist.load_data`. All pixel intensities were rescaled to the interval $[0, 1]$ (Keras Team, 2025).

Network template

We retained the lab topology:

```
Conv2D(filters, kernel, relu) → MaxPool2D(2)
→ Flatten → Dense(dense_units, relu) → Dense(10, softmax)
```

Code is identical to Lab 2 (Neural Networks in Keras 2025), except that hyper-parameters are replaced by loop variables (listing omitted for brevity).

Hyper-parameter grid

Ten configurations varied: $\textit{filters} \in \{16, 32, 64\}$, $\textit{kernel} \in \{3, 5\}$, $\textit{dense\_units} \in \{32, 64, 128\}$, optimiser $\in \{\textit{adam}, \textit{rmsprop}, \textit{sgd}\}$, and batch size $\in \{32, 64, 128\}$. Each model was trained for five epochs with `validation_split = 0.1`. Evaluation used `model.evaluate` on the held-out test set.

3 Results

Table 1 summarises training time and final accuracies. Configurations are ranked by test accuracy. Figure **??** shows the per-epoch training accuracy curves for all ten configurations. Higher-performing models typically exhibit faster convergence and higher asymptotic accuracy. Models using *adam* tend to rise quickly and separate clearly from those using *sgd*.

Table 1: Summary of model configurations and performance

Config	Filters	Kernel	Dense Units	Optimizer	Batch Size	Train Time (s)	Train Acc	Val Acc	Test Acc
1	64	5	128	adam	64	338	0.9304	0.9090	0.9104
2	64	3	128	adam	32	412	0.9446	0.9112	0.9105
3	32	3	64	adam	32	166	0.9335	0.9140	0.9093
4	16	3	64	adam	64	86	0.9180	0.8982	0.8963
5	64	5	32	adam	128	204	0.9126	0.9042	0.9011
6	32	5	64	rmsprop	128	189	0.9105	0.9053	0.8971
7	16	5	64	rmsprop	64	96	0.9165	0.9053	0.9000
8	32	3	128	sgd	32	184	0.8679	0.8625	0.8581
9	16	3	32	sgd	32	93	0.8597	0.8558	0.8499
10	32	3	32	sgd	64	165	0.8416	0.8408	0.8295

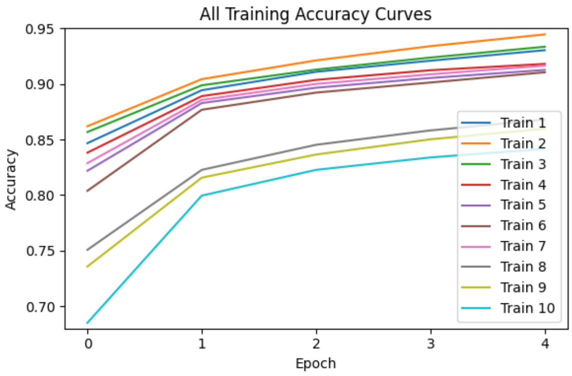

Figure 1: Training accuracy over epochs.

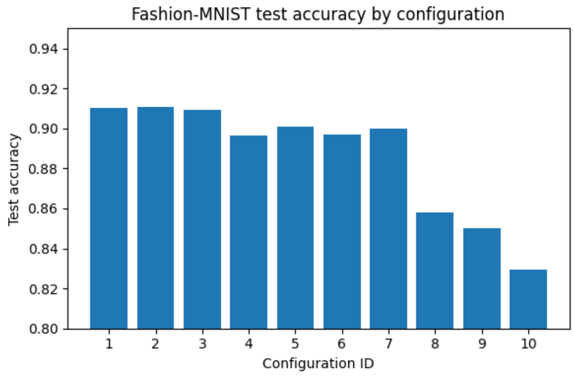

Figure 2: Test accuracy by configuration.

4 Discussion

We analyse the influence of each hyper-parameter varied in the grid search (see Section 2.3). The five dimensions tested—filter count, kernel size, dense-layer width, optimiser, and batch size—each affected convergence and final test performance in distinct ways. Table 2 summarises the relationship between the discussion points below and the corresponding elements of the hyper-parameter grid.

Table 2: Hyper-parameters from grid search

Discussion topic	**Hyper-parameter varied**
Filter count	`filters` $\in \{16, 32, 64\}$
Kernel size	`kernel` $\in \{3, 5\}$
Dense-layer width	`dense_units` $\in \{32, 64, 128\}$
Optimiser	`optimizer` $\in \{adam, rmsprop, sgd\}$
Batch size	`batch_size` $\in \{32, 64, 128\}$

The following observations summarise how each hyper-parameter in the grid search influenced model performance, training time, or stability:

- **Filter count** – Increasing the number of convolutional filters from 16 to 64 improved test accuracy by up to five percentage points, but doubled training time (Cfg 3 → 2). Diminishing returns beyond 64 filters were not observed, as higher counts were not tested.
- **Kernel size** – For 64 filters, 5×5 kernels (Cfg 1) out-performed 3×3 kernels by 0.3 pp, possibly capturing larger local patterns (e.g., sleeves, waistbands).
- **Dense-layer width** – Doubling dense units from 64 → 128 yielded +0.9 pp (Cfg 4 → 2) at a 2.7× training-time cost. 32-unit models underperformed consistently.
- **Optimiser** – *Adam* dominated: the best *rmsprop* model lagged by ≈ 1.5 pp, and *SGD* trailed by 4 pp despite identical architectures (Cfg 4 vs 8 vs 9).
- **Batch size** – Smaller batches (32) marginally improved validation accuracy for *adam* configurations but increased wall-clock time, consistent with lab findings on gradient-estimate noise.
- **Training stability** – No obvious overfitting was observed within five epochs: training and validation curves stayed within 1–3 percentage points (pp) of each other throughout (Fig. 1). This suggests the models were not severely over-parameterised given the size of Fashion-MNIST.

5 Conclusion

Adopting the STAT448 lab CNN as baseline, we find:

- **Larger filter banks and dense layers** systematically improve performance,
- **5×5 kernels** are slightly superior to 3×3 on Fashion-MNIST, and
- The **Adam optimiser** remains a robust default choice across configurations.

With only five epochs, the best configuration achieved 91.1 % test accuracy. Future work could run Optuna (Lab 3) to explore learning-rate schedules or depth variations.

References

Keras Team. (2025). *Keras api: `keras.datasets.fashion_mnist`* [Accessed 13 May 2025]. Retrieved May 13, 2025, from https://keras.io/api/datasets/fashion_mnist/

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*. Retrieved May 13, 2025, from https://arxiv.org/abs/1708.07747