Proceedings of CONF-CDS 2026: Machine Learning and Neural Network Applications in Engineering
DOI: 10.54254/2755-2721/2026.AST35278

Robustness of Convolutional Neural Networks to Partial Test-Set Image Corruption on Fashion-MNIST

Anthony Huang

International School of Beijing (ISB), Beijing, China
anthony.huang@student.isb.bj.edu.cn

Abstract. Image classifiers often report high accuracy on clean benchmark data, but in the real world, their inputs are not always clean. This paper tests how two convolutional models respond when part of the fashion-MNIST test set is deliberately corrupted. A custom four-layer convolutional neural network and an ImageNet-pretrained ResNet-18 were trained only on clean Fashion-MNIST images. A fixed corruption, Gaussian blur followed by contrast enhancement, was then applied to 0%, 10%, 30%, 50%, 70%, and 100% of the official test images. Both models used the same corrupted image set. Accuracy, mean predictive entropy, and t-distributed stochastic neighbor embedding plots were used to compare the runs. The custom model reached 91.31% accuracy on clean data but fell to 10.82% when every test image was corrupted. ResNet-18 started lower, at 88.10%, but reached 20.44% at 100% corruption. Entropy also rose with the corrupted percentage, from 0.1729 to 0.5379 for the custom model and from 0.1517 to 0.3073 for ResNet-18. These suggest that both clean-trained models were fragile under this specific corruption, while ResNet-18 held up better in this run. A possible explanation is that its deeper residual structure and ImageNet-pretrained layers preserved more coarse shape information after blur weakened fine local details.

Keywords: convolutional neural network, corruption robustness, Fashion-MNIST, predictive entropy, transfer learning

1. Introduction

Deep convolutional neural networks are now a standard tool for image classification because they can learn useful visual features from data [1]. High benchmark accuracy can still hide a weakness: the test images are usually prepared in the same way as the training images. In practical settings, images may be blurred, over-enhanced, compressed, or affected by lighting and sensor problems. Fashion-MNIST is a useful small benchmark for this issue because it contains 70,000 labeled 28 x 28 grayscale clothing images in ten balanced classes [2].

Earlier studies have shown that image-quality changes can sharply reduce neural-network accuracy. Blur and noise can be especially damaging [3], and common-corruption benchmarks show that high clean-image accuracy does not guarantee good performance on blurred, noisy, or degraded test images [4]. Blur is especially important for Fashion-MNIST because the images are only 28 x 28 pixels. Even mild smoothing can remove edges and small shape details that help distinguish

DOI: 10.54254/2755-2721/2026.AST35278

clothing classes [5]. Therefore, clean-image accuracy and corrupted-image accuracy should be reported separately.

In this experiment, one fixed blur-and-contrast corruption is applied to different shares of the Fashion-MNIST test set. A compact CNN trained from scratch is compared with a modified, ImageNet-pretrained ResNet-18. The goal is to measure accuracy, track predictive entropy, and inspect feature plots under clean and fully corrupted conditions. Corruption intensity stays the same throughout the experiment. Each selected image receives the same transformation, so the intermediate test sets are mixtures of clean and corrupted images, providing a simple baseline for future experiments with repeated runs and additional corruption types.

2. Related work

2.1. Convolutional architectures and transfer learning

Residual networks use skip connections that make deep networks easier to optimize by learning residual mappings [6]. ResNet-18 is often used for transfer learning because most of its convolutional layers can start from ImageNet weights and then adapt to a smaller dataset. Transferred features often help, but their benefit depends on how closely the source and target tasks match [7]. Larger comparisons show the same pattern: ImageNet pretraining can improve performance on many datasets, but the gains are smaller or inconsistent for some small and fine-grained tasks [8]. Therefore, ResNet-18 still needs to be tested directly on Fashion-MNIST rather than assumed to be stronger.

Model architecture and pretraining can also influence what visual cues a network relies on. For example, ImageNet-trained CNNs often depend strongly on local texture rather than overall shape, and this texture bias can affect robustness under image corruption [9].

2.2. Robustness evaluation

Corruption robustness is often tested by applying fixed transformations at test time and comparing the result with clean-image accuracy [4]. Training with similar corruptions can improve robustness [10]. However, the two models in this study were not trained on corrupted images but only on clean ones. Feature plots and predictive entropy were also used to examine the effect beyond accuracy. t-Distributed stochastic neighbor embedding (t-SNE) shows whether feature vectors from the same class still form local groups under various corruption levels in a two-dimensional view [11], while predictive entropy summarizes how concentrated the SoftMax outputs are.

3. Methodology

3.1. Dataset and preprocessing

Fashion-MNIST contains 60,000 official training images and 10,000 official test images [2]. The implementation randomly divided the official training set into 42,000 training images and 18,000 validation images using seed 42. The official 10,000-image test set remained separate and was used for all reported test results. Images were converted to tensors and normalized. No corruption-based data augmentation was applied during training.

DOI: 10.54254/2755-2721/2026.AST35278

3.2. Corruption protocol

The fixed corruption combined a Gaussian blur with a 5 x 5 kernel and contrast adjustment with a factor of 2.5. In the implemented pipeline, both operations were applied after tensor normalization. Applying the transformations in normalized space changes their numerical interpretation because values no longer occupy the original grayscale range and contrast is adjusted around the normalized tensor statistics. The procedure is reproducible but corresponds less directly to ordinary image acquisition defects than corruption applied before normalization. Six test conditions were evaluated: clean data and test sets with 10%, 30%, 50%, 70% or 100% of images corrupted.

For each nonzero proportion, image indices were sampled without replacement and stored in a degradation indices file. Both models used the same stored indices at a given proportion. The 10%, 30%, 50%, and 70% subsets were sampled independently rather than constructed as nested subsets; the 100% condition contained all test images. The intermediate conditions therefore measure aggregate accuracy on mixed clean-corrupted test sets. The 100% condition most directly measures performance on the fixed corruption itself.

3.3. Model architectures

The custom CNN contained four 3 x 3 convolutional layers with 32, 64, 128 and 128 output channels. ReLU activations followed each convolution, and 2 x 2 max pooling followed the first two convolutional blocks. Adaptive average pooling reduced the final feature maps to one value per channel. The classifier used a 128-to-64 fully connected layer, ReLU, dropout with probability 0.3, and a ten-class output layer. The architecture contained 249,162 trainable parameters.

The second model was torchvision's ResNet-18 initialized with ImageNet weights. Its original three-channel first convolution was replaced by a randomly initialized one-channel convolution with a 7 x 7 kernel, stride 2, and 64 output channels. The final classifier was replaced by a randomly initialized ten-class layer. The remaining pretrained parameters were retained, and all layers were fine-tuned. The modified network contained approximately 11.18 million trainable parameters.

3.4. Training procedure

Both models were trained for at most 40 epochs with a batch size of 64 and cross-entropy loss. The custom CNN used Adam optimization with a learning rate of 0.001 and weight decay of 0.0001. ResNet-18 used AdamW, a weight-decay variant of Adam, with a learning rate of 0.0001 and the same weight decay. Training stopped after five consecutive epochs without validation-loss improvement. The evaluation used the model state present when early stopping occurred. Seed 42 was set for PyTorch and NumPy. The recorded runs used the PyTorch Metal Performance Shaders backend on Apple Silicon.

3.5. Evaluation

Top-1 classification accuracy was calculated for each test condition. Accuracy differences are reported in percentage points. Mean predictive entropy was calculated from the ten-class softmax distribution for every test image and averaged across the test set. Higher entropy indicates a less concentrated predictive distribution.

Feature vectors were extracted after adaptive pooling from each model's final convolutional representation. t-SNE projected approximately 2,048 feature vectors per condition into two

DOI: 10.54254/2755-2721/2026.AST35278

dimensions using perplexity 30 and random seed 42. Separate projections were fitted for each model and condition. Visual comparisons therefore focus on within-panel class mixing rather than distances or directions across panels.

4. Results

4.1. Accuracy and predictive entropy

Table 1. Test accuracy and mean predictive entropy by corrupted test-set proportion

Corrupted test images (%)	Custom CNN accuracy (%)	ResNet-18 accuracy (%)	Custom CNN entropy	ResNet-18 entropy
0	91.31	88.10	0.1729	0.1517
10	83.18	81.41	0.2085	0.1662
30	66.79	67.19	0.2877	0.2001
50	51.10	54.23	0.3552	0.2297
70	34.98	40.68	0.4332	0.2569
100	10.82	20.44	0.5379	0.3073

Table 1 shows the measured accuracy and mean predictive entropy values of the Custom CNN and ResNet-18 models at different corruption rates. Accuracy fell steadily as more test images were corrupted. The custom CNN was 3.21% higher on clean data and still 1.77 points higher at 10% corruption. After that point, the order changed. ResNet-18 was ahead by 0.40 points at 30%, 3.13 points at 50%, 5.70 points at 70%, and 9.62 points when every image was corrupted. From clean to full corruption, the custom CNN lost 80.49% while ResNet-18 lost 67.66 points.

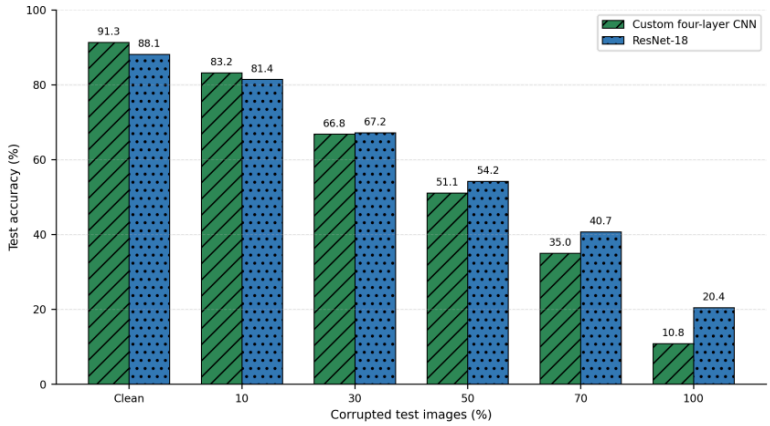

Figure 1. Test accuracy of the custom CNN and ResNet-18 as the proportion of test images receiving the fixed blur-and-contrast corruption increases

Figure 1 shows the measured test accuracy (%) vs the test image corruption level (%) for both models. The middle columns are mixed-test-set results. A 50% condition, for example, contains 5,000 clean and 5,000 corrupted test images. The 100% condition means every image is corrupted by going through the same post-normalization blur-and-contrast operation. Both models were weak there, but ResNet-18 still produced nearly twice the custom CNN's accuracy.

DOI: 10.54254/2755-2721/2026.AST35278

As the proportion of corrupted images increased, mean predictive entropy rose for both models, indicating that their predictions became less confident overall. ResNet-18 maintained lower entropy than the custom CNN across all conditions. This might initially suggest better-calibrated or more confident predictions. However, at 100% corruption, ResNet-18 achieved only 20.44% accuracy while maintaining a relatively low entropy of 0.3073. This pattern is consistent with overconfidence, although calibration metrics such as Expected Calibration Error would be required to verify this directly.

4.2. Feature-space visualization

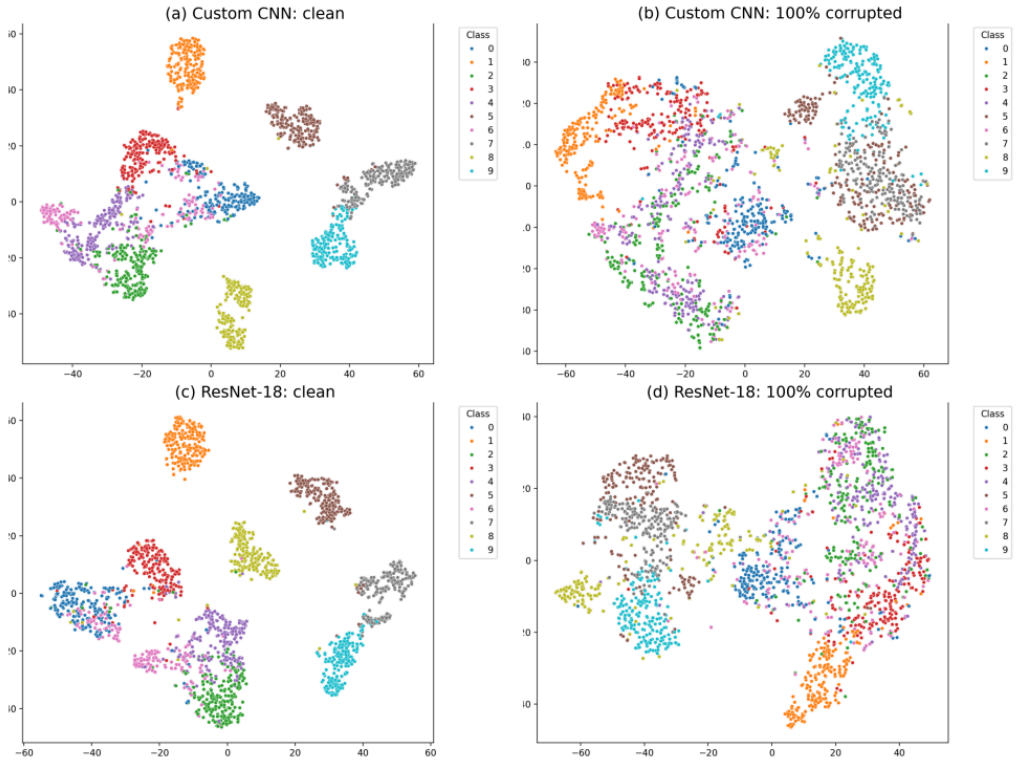

Figure 2. Independent t-SNE projections of custom CNN and ResNet-18 feature vectors for clean and fully corrupted test images. Colors denote Fashion-MNIST classes. (a) custom CNN at 0% test image corruption rate shows clean boundary separations among classes; (b) custom CNN at 100% test image corruption rate shows blurred boundaries among classes; (c) Resnet-18 at 0% test image corruption rate shows clear boundary separation among classes; (d) Resnet-18 at 100% test image corruption rate shows blurred boundaries among classes

The clean projections showed several class-dominated regions, especially for clothing types with clearer shape differences. In the saved plots, classes such as trousers and bags appeared more separated than shirt-like categories. The fully corrupted panels were less organized. Colors that had been grouped together spread into wider areas, and similar classes became harder to separate. This matches the accuracy drop in Figure 1. Since each t-SNE map was fitted separately, the panels are useful for checking class mixing [12].

DOI: 10.54254/2755-2721/2026.AST35278

5. Discussion

The clean accuracies alone do not fully capture the differences between the two models. The results in Table 1 reveal a different pattern once the corruptions were introduced. On clean Fashion-MNIST, the custom CNN looked like a better model. Once corrupted images became common, that advantage faded. By the 50% condition, ResNet-18 was already 3.13% ahead, and at full corruption the gap reached 9.62 points. While ResNet-18's 20.44% accuracy at full corruption is still weak, the data suggests that the smaller model won on clean images, while the larger pretrained model lost less under this stress test.

One possible explanation is that the custom CNN had fewer alternative feature representations once the blur removed fine detail. Because it was trained from scratch on tiny 28×28 Fashion-MNIST images, it likely relied heavily on local edges and textures that the 5×5 Gaussian blur would have smoothed away. ResNet-18, by contrast, has many more layers and residual connections, so even if early layers lost some high-frequency information, later layers could still piece together coarser shape cues. The ImageNet pretraining probably also helped by giving the model a broader set of visual features to begin with. The comparison is a bit imperfect: the two models were trained with different optimizers, learning rates, and vastly different parameter counts, any of which factors could have contributed to the gap as well.

The order of preprocessing is important. Because normalization happened before the blur and contrast steps, the contrast adjustment was applied to already-standardized values instead of raw pixel intensities. Running the same operations in the opposite order would likely have produced a different kind of corruption. Therefore, the results are tied to this specific pipeline, not just to the labels "Gaussian blur" and "contrast."

The mixed percentages also make the accuracy numbers harder to read clearly. Since the corrupted subsets at 10%, 30%, 50% and 70% were sampled independently, differences between those points could partly reflect which images got picked rather than how the model responded. It would have been cleaner to use nested subsets and to report accuracy on the clean versus corrupted images separately within each condition. Repeating the sampling across a few seeds would have helped too.

Both predictive entropy and t-SNE provided useful supporting context, although each has limitations. The increase in entropy tracked the accuracy drop reasonably well. ResNet-18 maintained relatively low entropy (0.3073) despite only achieving 20.44% accuracy under full corruption, suggesting possible overconfidence, a known calibration problem in neural networks. The t-SNE plots showed more class mixing after corruption, but they're hard to compare directly across panels since each one was fit separately.

Several aspects of the experimental design could be improved in future work. Testing blur and contrast in isolation would make it easier to see which one hurts performance. Restoring the best checkpoint before evaluating would also be more standard practice. And ResNet-18's relatively heavy stem (7×7 stride-2 conv + max pooling) might just be a bad fit for 28×28 grayscale images to begin with.

6. Conclusion

This research compared a custom four-layer CNN with a modified ImageNet-pretrained ResNet-18 on Fashion-MNIST under one fixed test-time corruption. The corruption combined a Gaussian blur with a 5 x 5 kernel and contrast adjustment with a factor of 2.5. It was applied to the official test set

DOI: 10.54254/2755-2721/2026.AST35278

while both models were trained only on clean images. The percentage labels refer to the share of corrupted test images.

Both models lost accuracy once corrupted examples were applied. The custom CNN declined from 91.31% clean accuracy to 10.82% when all test images were corrupted. ResNet-18 declined from 88.10% to 20.44%. ResNet-18 was less accurate on clean data but kept more performance under complete corruption. Its advantage in the final condition could likely come from pretraining, deeper features, residual blocks, larger capacity, and/or ImageNet initialization. ResNet-18 remained relatively confident even when many classifications were incorrect, suggesting that confidence-based measures, such as predictive entropy, provide useful information beyond accuracy alone.

These results suggest that clean-image benchmark performance may substantially overestimate real-world robustness, even for architectures that perform well on standard test data. Future improvements could apply corruptions before normalization, restore the best validation checkpoint, separate blur from contrast adjustment, and repeat runs with multiple seeds. Mixed test sets should also report clean and corrupted subsets separately.

Although Fashion-MNIST is a relatively simple benchmark, the findings highlight a broader challenge in machine learning: models that perform well under controlled evaluation conditions may not remain reliable when input quality changes. Even a single fixed corruption caused substantial performance degradation in both networks. Evaluating robustness alongside standard accuracy metrics may therefore provide a more realistic assessment of model behavior and help guide the development of image classifiers that generalize more effectively to real-world conditions.

References

[1] LeCun, Y., Bengio, Y., Hinton, G. (2015) Deep learning. Nature, 521: 436-444.
[2] Xiao, H., Rasul, K., Vollgraf, R. (2017) Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv: 1708.07747.
[3] Dodge, S., Karam, L. (2016) Understanding how image quality affects deep neural networks. International Conference on Quality of Multimedia Experience, 1-6.
[4] Hendrycks, D., Dietterich, T. (2019) Benchmarking neural network robustness to common corruptions and perturbations. International Conference on Learning Representations.
[5] Vasiljevic, I., Chakrabarti, A., Shakhnarovich, G. (2016) Examining the impact of blur on recognition by convolutional networks. arXiv preprint arXiv: 1611.05760.
[6] He, K., Zhang, X., Ren, S., Sun, J. (2016) Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition, 770-778.
[7] Yosinski, J., Clune, J., Bengio, Y., Lipson, H. (2014) How transferable are features in deep neural networks? Advances in Neural Information Processing Systems, 27: 3320-3328.
[8] Kornblith, S., Shlens, J., Le, Q.V. (2019) Do better ImageNet models transfer better? IEEE Conference on Computer Vision and Pattern Recognition, 2661-2671.
[9] Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W. (2019) ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. International Conference on Learning Representations.
[10] Shorten, C., Khoshgoftaar, T.M. (2019) A survey on image data augmentation for deep learning. Journal of Big Data, 6: 60.
[11] van der Maaten, L., Hinton, G. (2008) Visualizing data using t-SNE. Journal of Machine Learning Research, 9: 2579-2605.
[12] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q. (2017) On calibration of modern neural networks. Proceedings of Machine Learning Research, 70: 1321-1330.