Visualizing What Convolutional Neural Networks Learn: Feature Maps & Filters

Author: Praneeth Kumar Chapalabandla

GitHub: https://github.com/praneeth04-ai429/neural_network_

Abstract

The tutorial is a friendly introduction to the learning process of Convolutional Neural Networks (CNNs) in the process of learning hierarchical visual features. With a small CNN trained on the Fashion-MNIST dataset, we investigate convolutional filters, feature maps at various levels and how the high-level representations develop starting with the low-level ones.

Learning Objectives

- Understand how CNNs learn visual features through convolutional filters.

- Visualize intermediate feature maps and interpret what they represent.

- Relate CNN representations to model performance and architecture.

1. Introduction

Convolutional Neural Networks (CNNs) are now the pillars of computer vision in the modern world. They are also very good at image classification, detection and segmentation because they learn spatially structured representations. CNNs are able to learn features automatically by recognizing edges, textures, shapes and object-level patterns in an automatic and unsupervised manner.

This tutorial is devoted to the interpretation of the CNNs as they are learned through visualization of their internal elements, namely the convolutional filters and the feature maps produced by them. Such visualizations are useful in demystifying the working of deep learning models and enable practitioners to learn more about the behavior of the models.

2. Dataset: Fashion-MNIST

The Fashion-MNIST dataset is a recent version of the MNIST dataset, with 10 categories of 70,000 grayscale images of clothing. Each image is 28×28 pixels. The data has been utilized extensively as a benchmark of lightweight computer vision models and is best suited to illustrate CNN feature learning.

Classes include:

• T-shirt/top

• Trouser

• Pullover

• Dress

• Coat

• Sandal

• Shirt

• Sneaker

• Bag

• Ankle boot

3. Model Architecture

Our small CNN is made up of 2 convolutional layers and a fully connected classifier. This is a simple architecture that can extract meaningful visual features and ensure that the computational cost is low.

The architecture includes:

• Conv Layer 1: 8 filters (3×3), ReLU activation

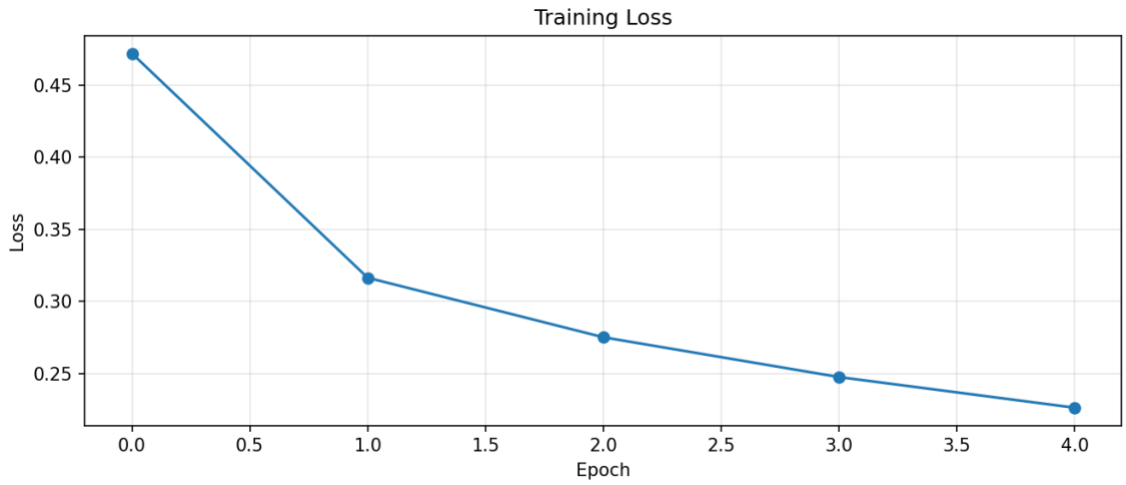
• Conv Layer 2: 16 filters (3×3), ReLU activation

• Max-Pooling Layer: 2×2

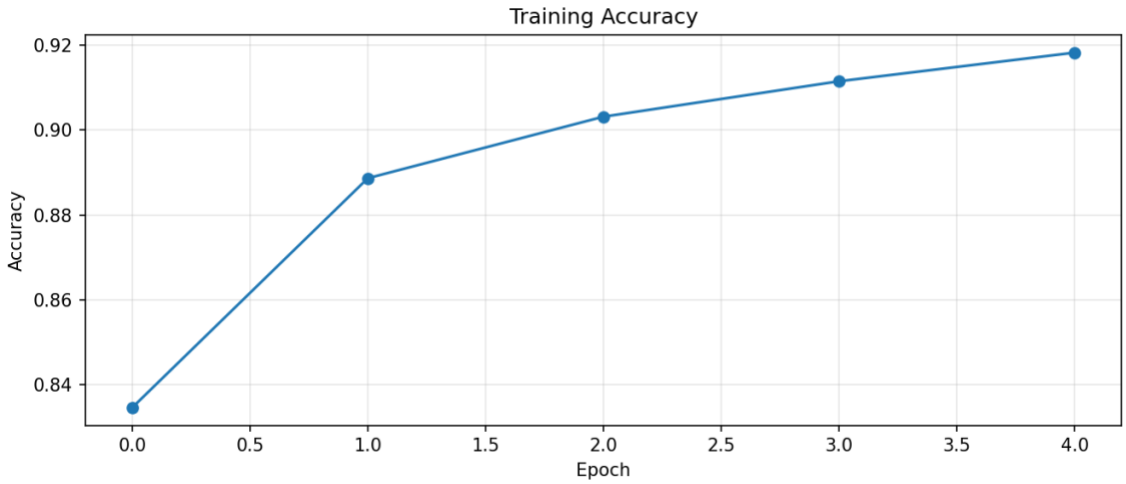
• Fully Connected Output Layer: 10 classes

4. Training Process

The CNN is trained for 5 epochs using the Adam optimizer and cross-entropy loss. Training loss and accuracy are recorded per epoch to evaluate learning progress.



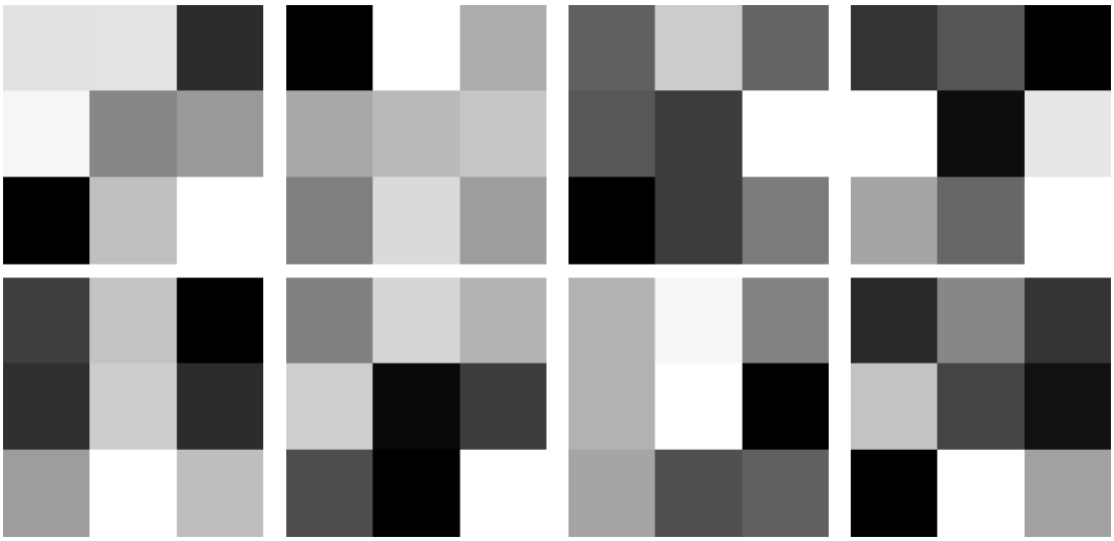
Training loss across epochs: Line plot showing decreasing loss over time.



Training accuracy across epochs : Line plot showing increasing accuracy over time.

5. Visualizing Learned Filters

The first convolutional layer learns basic feature detectors such as edges, corners, and gradients. Each filter can be visualized as a small 3×3 grayscale image representing its learned weights.

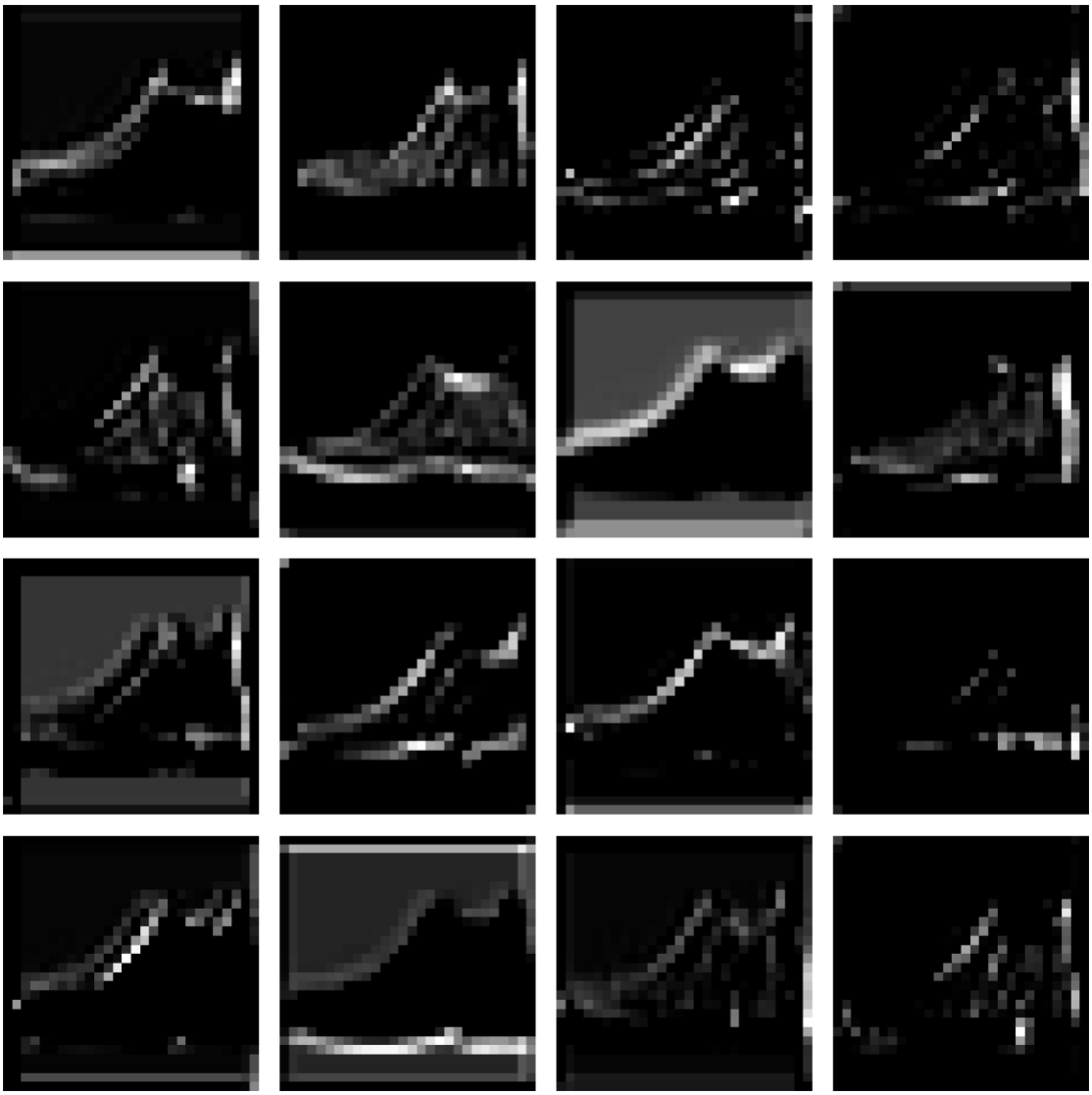


Filters from the first convolutional layer : Grid of grayscale 3×3 learned filters.

6. Visualizing Feature Maps

Feature maps show how each filter responds to a particular input image. Early layers tend to highlight edges and simple textures, while deeper layers capture more abstract shapes, such as outlines of shoes or bags.

Feature maps from the first convolutional layer : Grid of 8 feature activations.



Feature maps from the second convolutional layer : Grid of 16 deeper feature activations.

7. Discussion

The visualizations reveal a hierarchical progression: early layers detect low-level signals such as edges and textures, while deeper layers form higher-level abstractions that reflect meaningful shapes. This hierarchical learning is one of the core strengths of CNNs and explains their performance on image-based tasks.

8. Ethical Considerations

Interpreting CNN visualizations can help practitioners detect biases or unexpected model behaviour. Although Fashion-MNIST is a neutral dataset, deeper real-world applications must address fairness, transparency, and the societal implications of automated visual systems.

References

1. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition.
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
3. Dumoulin, V., & Visin, F. A guide to convolution arithmetic for deep learning.
4. PyTorch Documentation: https://pytorch.org/docs/stable/
5. Fashion-MNIST Dataset: https://github.com/zalandoresearch/fashion-mnist