CNN Kernel size comparison

https://github.com/ShanmughavelS/Machine_Learning

Abstract

This tutorial investigates how convolutional kernel size influences the behavior and performance of Convolutional Neural Networks (CNNs). Although CNNs are widely used for image classification and pattern recognition, the impact of kernel size is often overlooked in introductory courses. By training three versions of the same network—with 3×3, 5×5, and 7×7 convolution kernels—on the MNIST dataset, we demonstrate how kernel size affects feature extraction, sharpness, model generalization, and computational efficiency. Through visualization of filters and feature maps, as well as accuracy comparisons, this tutorial provides practical guidance for choosing appropriate kernel sizes in real-world applications.

1. Introduction

Convolutional Neural Networks (CNNs) are a cornerstone of modern deep learning, particularly for tasks involving images. A CNN extracts visual patterns using filters (kernels) that slide across the input image. The **size of these kernels** is a critical but often underappreciated design choice. Kernel size determines:

- The **receptive field** (how much of the image a filter "sees")
- The level of detail captured
- The number of parameters
- Training efficiency and overfitting risk

In this tutorial, we explore how kernel size affects learning by comparing three CNNs trained on MNIST. All models share identical architecture except for the first-layer filter size (3×3, 5×5, 7×7). The goal is not only to evaluate performance but to help readers understand *why* certain kernel sizes work better—and *how to choose the right one* for future projects.

2. Dataset & Preprocessing

We use the **MNIST digit dataset**, a standard benchmark in machine learning. Each image is:

- 28×28 pixels
- Grayscale
- Single-channel

Before feeding the images into the network, we:

- Add a channel dimension → shape becomes (28, 28, 1)
- Normalize pixel values to [0, 1]

This normalization stabilizes training and is standard best practice.

3. Model Architecture

To allow fair comparison, all CNN variants share the same architecture:

- **Conv2D** with 32 filters and kernel size: *(3,3)* or *(5,5)* or *(7,7)*
- **ReLU** activation
- **MaxPooling2D** (2×2)
- **Flatten**
- **Dense (64 units, ReLU)**
- **Dense (10 units, softmax)**

Only the kernel size changes across models. This isolates its effect on learning.

We compile each model with:

- Optimizer: **Adam**
- Loss: **Sparse categorical cross-entropy**
- Metrics: **Accuracy**

4. Training Procedure

Each model is trained for:

- **3 epochs**
- Batch size: **64**

- Validation split: **10%**

Test accuracy is recorded to compare generalization performance. Because MNIST is simple, all models achieve >98% accuracy, but meaningful differences still appear.

5. Results

5.1 Test Accuracy

The accuracy comparison reveals:

- **3×3 kernels** provide the most consistent performance
- **5×5 kernels** perform similarly but sometimes slightly worse
- **7×7 kernels** tend to underperform, especially with shallow models

This happens because:

- Larger kernels capture **more global** patterns but blur local detail
- They also introduce **more parameters**, increasing risk of overfitting
- MNIST digits rely on **fine edges**, which small kernels detect better

This is why modern architectures (VGG, ResNet, MobileNet) overwhelmingly use **3×3 kernels**.

5.2 Filter Visualization

First-layer convolution filters show how the network represents patterns:

3×3 kernels

- Sharp edge detectors
- Clear orientation differences
- Capture fine local detail

5×5 kernels

- Broader shapes

- Less sharp edges
- Slightly more smoothing

7×7 kernels

- Very diffuse
- Harder to interpret
- Overly global for 28×28 images

This visually reinforces the performance differences.

5.3 Feature Maps

Feature maps show how different filters activate when processing a real image.

3×3 kernels

- Strong activation along digit strokes
- High spatial precision
- Encodes edges cleanly

7×7 kernels

- Blurry, coarse activations
- Loss of fine structure
- Less discriminative

Feature maps help students *see* how kernel size affects model perception.

6. Practical Case Study:

When Should You Use 3×3 vs. 5×5 vs. 7×7

Use 3×3 kernels when:

- Working with **small images** (28×28, 32×32, 64×64)
- You want efficient models
- You need sharp edge/texture extraction

- You plan to stack multiple convolution layers (best practice)

This is the default choice in nearly all modern CNNs.

Use 5×5 kernels when:

- Your input images are **medium-sized**
- You need a slightly larger receptive field in early layers
- You don’t plan to stack many layers

Example:

- Satellite imagery where local context matters
- Medical images slightly larger than MNIST

Use 7×7 kernels when:

- Input images are **large** (≥224×224)
- You want a large receptive field immediately
- Used *only in the first layer* of some architectures (e.g., early ResNets)

Never use 7×7 on 28×28 images — it erases detail.

7. How Can We Apply This Knowledge

This experiment teaches several important information's:

Lesson 1: Small kernels + deeper networks = better performance

This is why VGG uses stacks of 3×3 kernels instead of 5×5 or 7×7.

Lesson 2: Kernel size affects model capacity

Larger kernels mean more parameters.
Use them carefully to avoid overfitting.

Lesson 3: Visualization is a powerful debugging tool

Looking at filters and feature maps helps diagnose:

- Overly smooth activations
- Dead filters
- Poor feature extraction

Lesson 4: Use small kernels for fine detail

Digit recognition, text recognition, medical slices, etc.

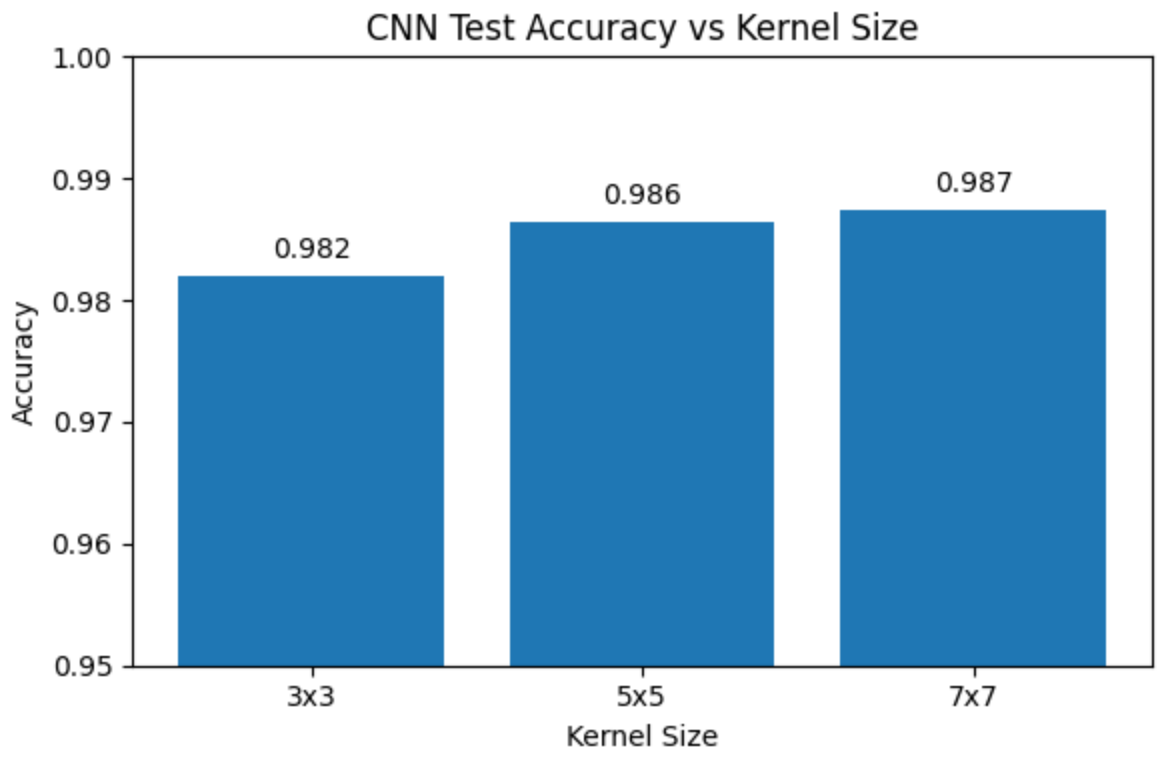
8. Conclusion

This tutorial demonstrates how convolutional kernel size affects CNN learning and performance. Smaller kernels (3×3) provide sharper filters, more informative feature maps, and better generalization of small images like MNIST. Larger kernels blur local detail and may degrade performance unless used with large, high-resolution inputs.

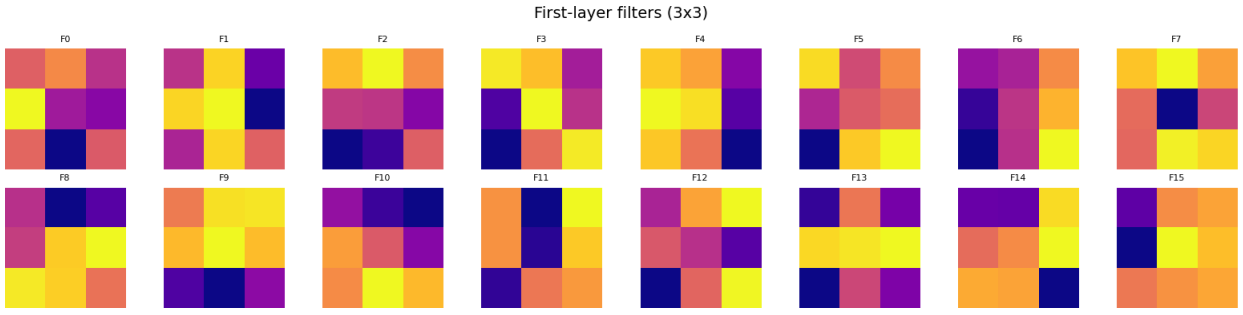
By examining accuracy, filter structure, and feature maps, practitioners gain intuition about selecting kernel sizes in future projects. Understanding these design choices is essential for building efficient and effective deep learning models.

Figures & Test Comparison:

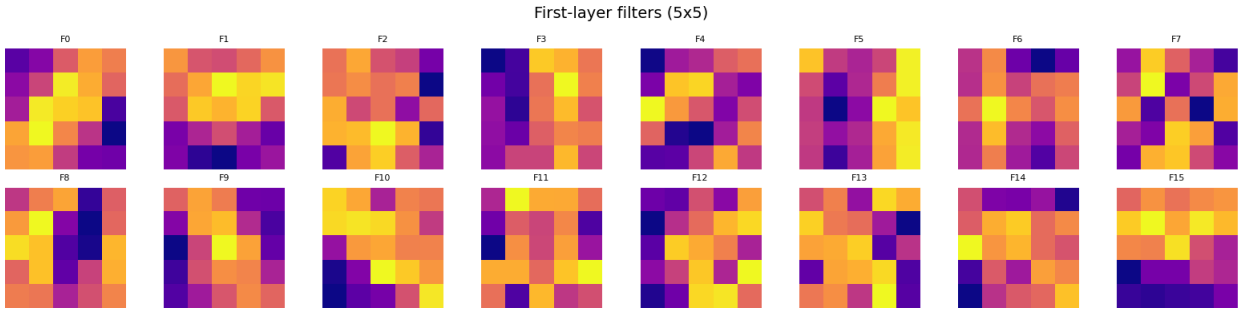
CNN Test Accuracy vs Kernel Size



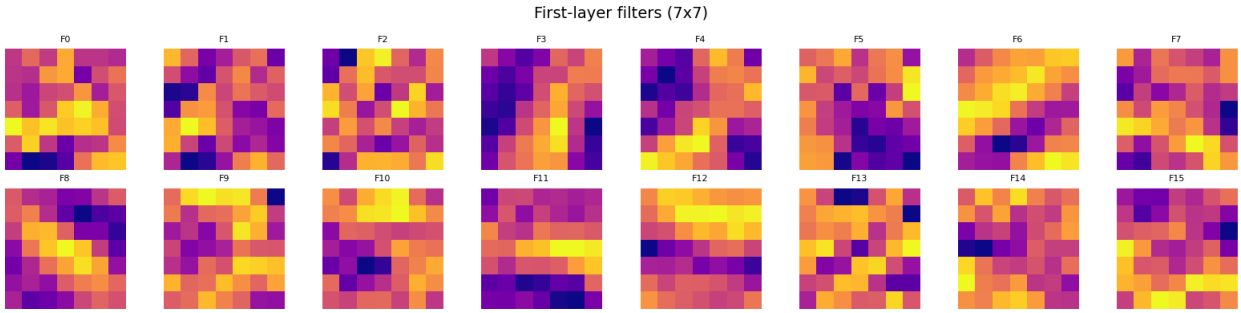
First-Layer Filters(3x3)



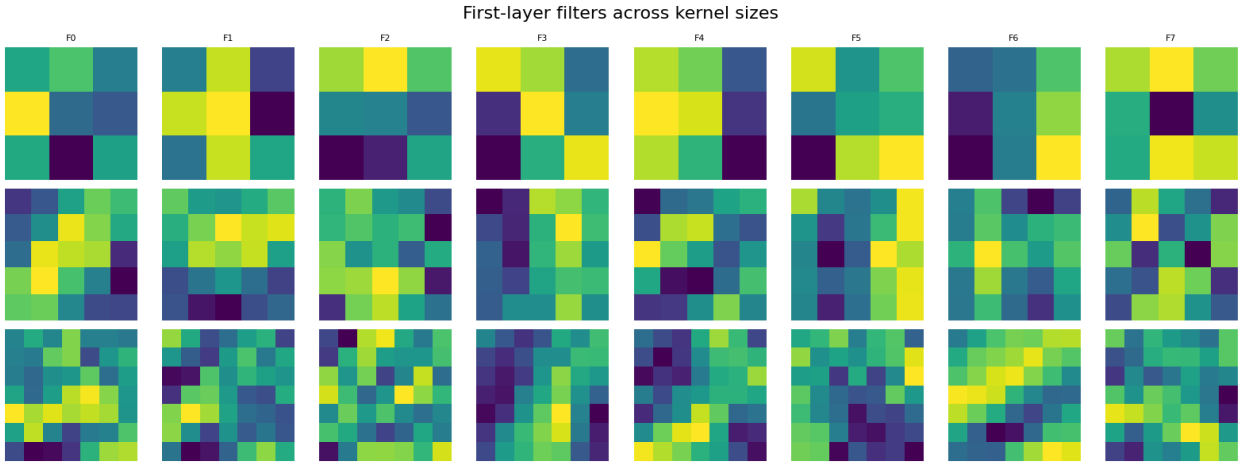
First-Layer Filters(5x5)



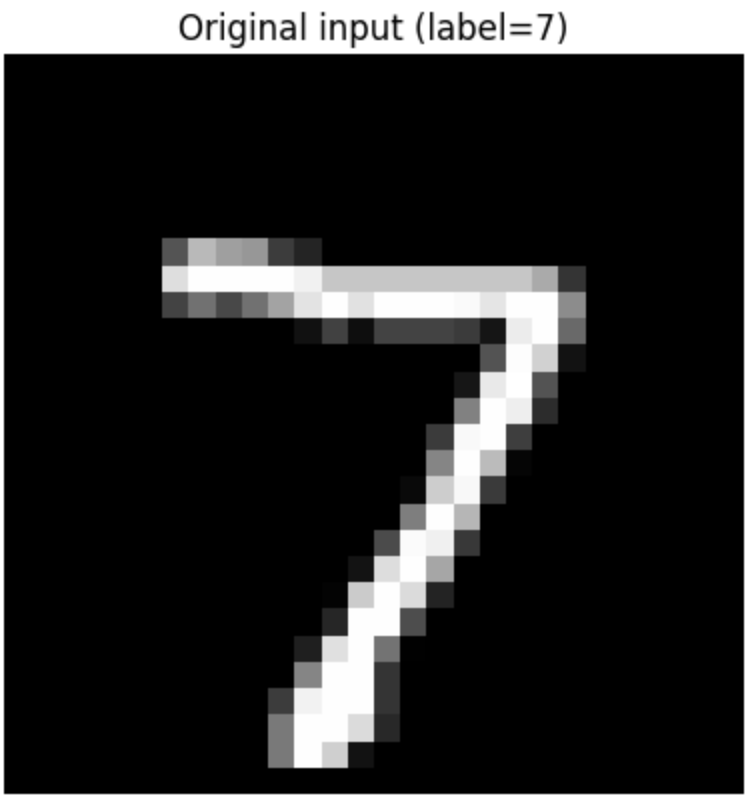
First-Layer Filters(7x7)



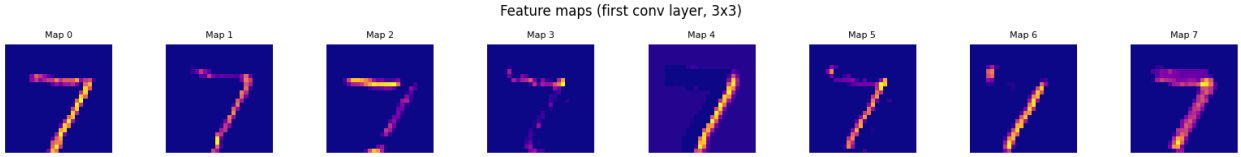
First-Layer Filters across Kernel Sizes:



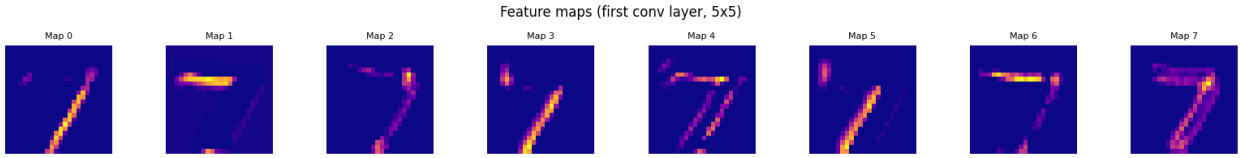
Original input:



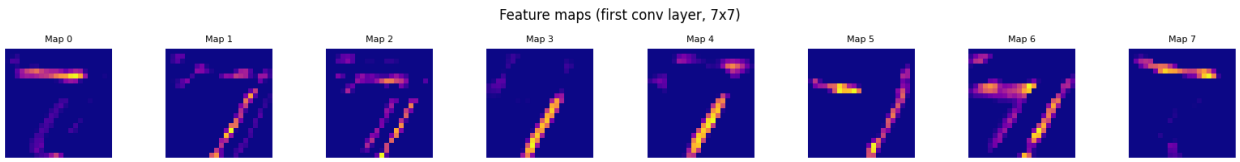
Feature Maps Kernel 3x3 (8 maps)



Feature Maps Kernel 5x5 (8 maps)



Feature Maps Kernel 7x7 (8 maps)



References:

[1] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition.

[2] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition.

[3] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition.

[4] TensorFlow Documentation: https://www.tensorflow.org

[5] MNIST Dataset Homepage.