Model:

```
# Input: 28x28x1 (height, width, channels)
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
# Output: 26x26x32 (Filter size 3x3 reduces dimensions by 2)
model.add(layers.MaxPooling2D((2, 2))) # Output: 13x13x32 (Halves dimensions)

# 2. Second Convolutional Block
model.add(layers.Conv2D(64, (3, 3), activation='relu')) # Output: 11x11x64
model.add(layers.MaxPooling2D((2, 2))) # Output: 5x5x64

# 3. Third Convolutional Block
model.add(layers.Conv2D(64, (3, 3), activation='relu')) # Output: 3x3x64

# --- 4. Classifier Block (Classification) ---
# Flatten the 3D tensor output into a 1D vector for the Dense layers.
model.add(layers.Flatten()) # Output: 3*3*64 = 576 units

# Dense Hidden Layer
model.add(layers.Dense(64, activation='relu'))

# Output Layer: 10 units for 10-way classification, using softmax for probability distribution.
model.add(layers.Dense(10, activation='softmax'))
#For hidden layers usually use relu
#For output use softmax or sigmoid
```

Result:

```
--- Final Evaluation on Test Data ---


Test Loss: 0.0196

Test Accuracy: 0.9933


--- Performance Comparison ---

Dense Network Accuracy (Ch 2): 0.9780

ConvNet Accuracy (Ch 5): 0.9933

Conclusion: The ConvNet significantly outperforms the Dense network.
```

Summary

The basic ConvNet architecture achieved a high test accuracy of **99.33%**, significantly **outperforming the previously tested densely-connected network** (97.80%). This demonstrates the superior ability of convolutional layers to learn local, hierarchical patterns directly from raw image pixel data.