MNIST Dataset Introduction and Visualization

This notebook provides an introduction to the MNIST dataset and demonstrates various visualization techniques.

Dependencies

Import required libraries and modules

In [1]:
```
import sys
import os

# Add project root to Python path
current_dir = os.getcwd()
project_root = os.path.dirname(current_dir)
sys.path.append(project_root)

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.decomposition import PCA
from src.utils.mnist_visualization import visualize_mnist_image
```

Loading MNIST Dataset

Let's load the MNIST dataset from local CSV files.

In [2]:
```
# Load training data
train_data = pd.read_csv(os.path.join(project_root, 'data', 'train.csv'))
X = train_data.iloc[:, 1:].values  # All columns except the first (labels)
y = train_data.iloc[:, 0].values   # First column (labels)

print(f"Dataset shape: {X.shape}")
print(f"Number of classes: {len(np.unique(y))}")
```

```
Dataset shape: (60000, 784)
Number of classes: 10
```

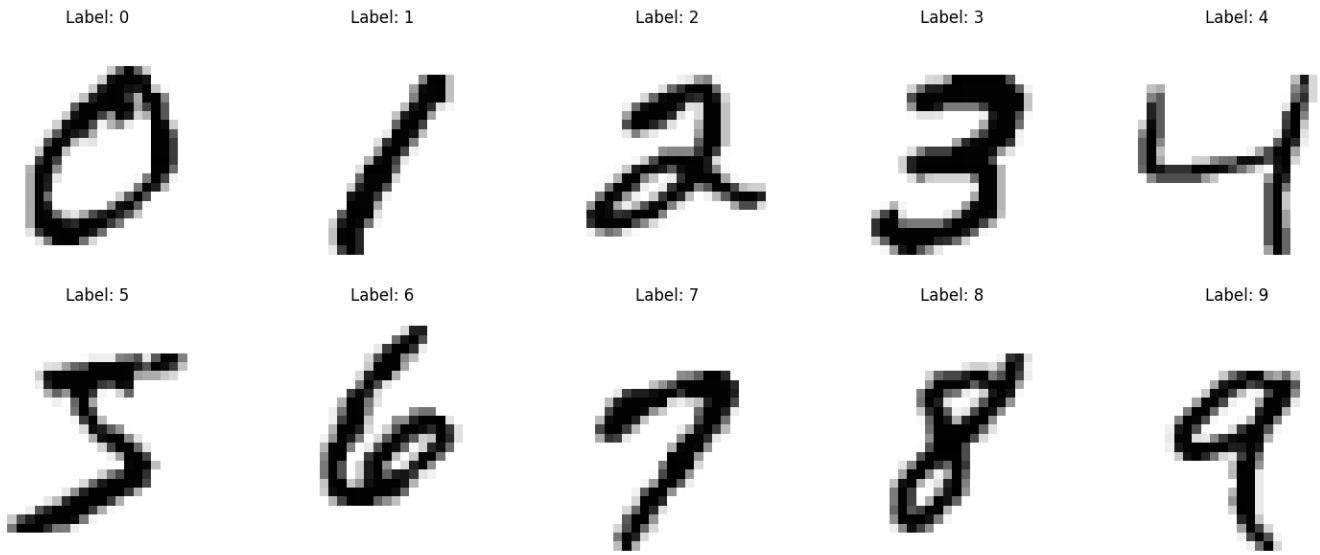
Visualizing Sample Images

Let's visualize one example for each digit class.

In [3]:
```
# Create a figure with subplots
fig, axes = plt.subplots(2, 5, figsize=(15, 6))
axes = axes.ravel()

# Get one example for each digit (0-9)
for digit in range(10):
    # Find the first occurrence of this digit
    idx = np.where(y == digit)[0][0]
    visualize_mnist_image(X[idx], digit, ax=axes[digit])

plt.tight_layout()
plt.show()
```

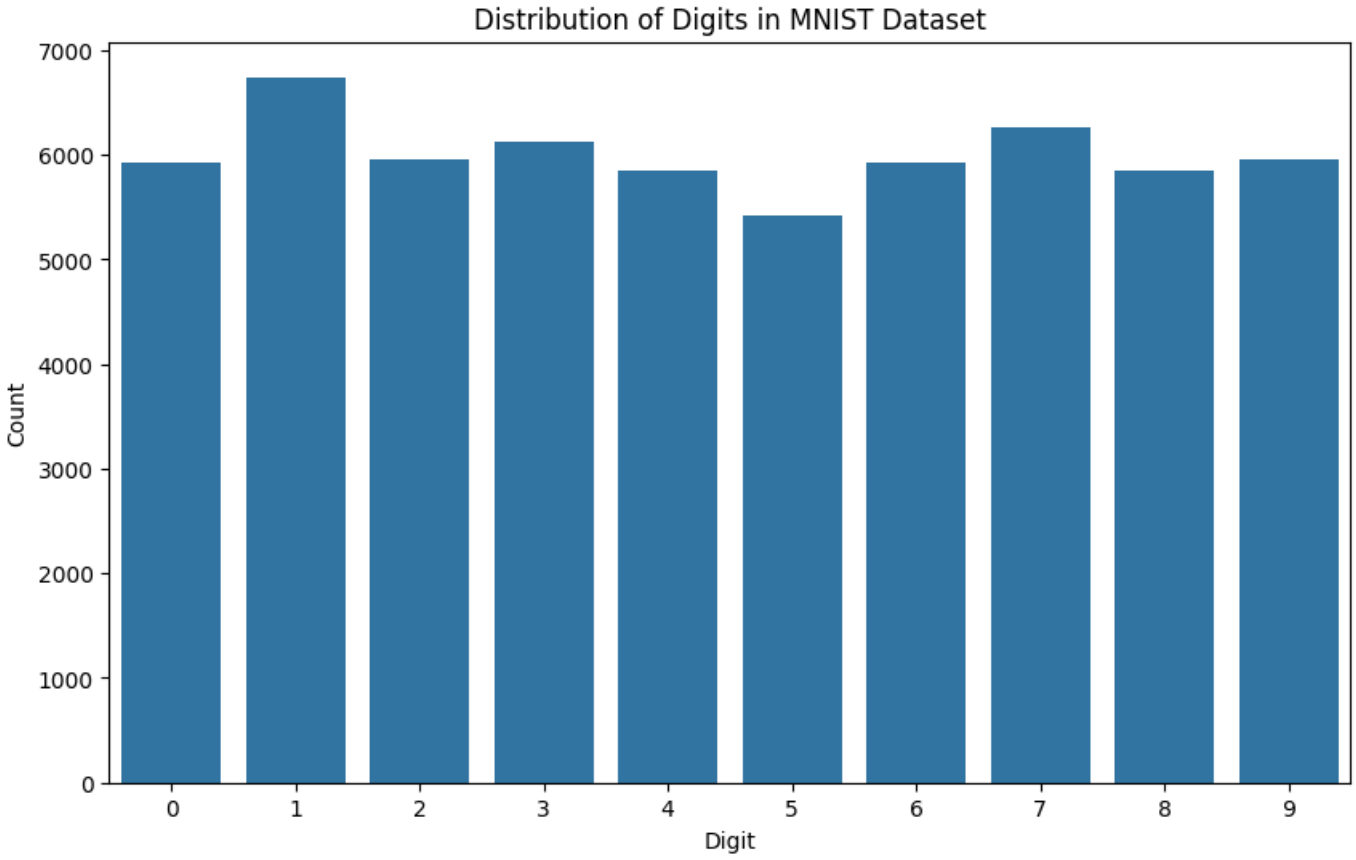


Class Distribution

Let's examine the distribution of classes in the dataset.

In [4]:
```
# Count occurrences of each digit
unique, counts = np.unique(y, return_counts=True)

# Create bar plot
plt.figure(figsize=(10, 6))
sns.barplot(x=unique, y=counts)
plt.title('Distribution of Digits in MNIST Dataset')
plt.xlabel('Digit')
plt.ylabel('Count')
plt.show()
```



Pixel Value Distribution with PCA

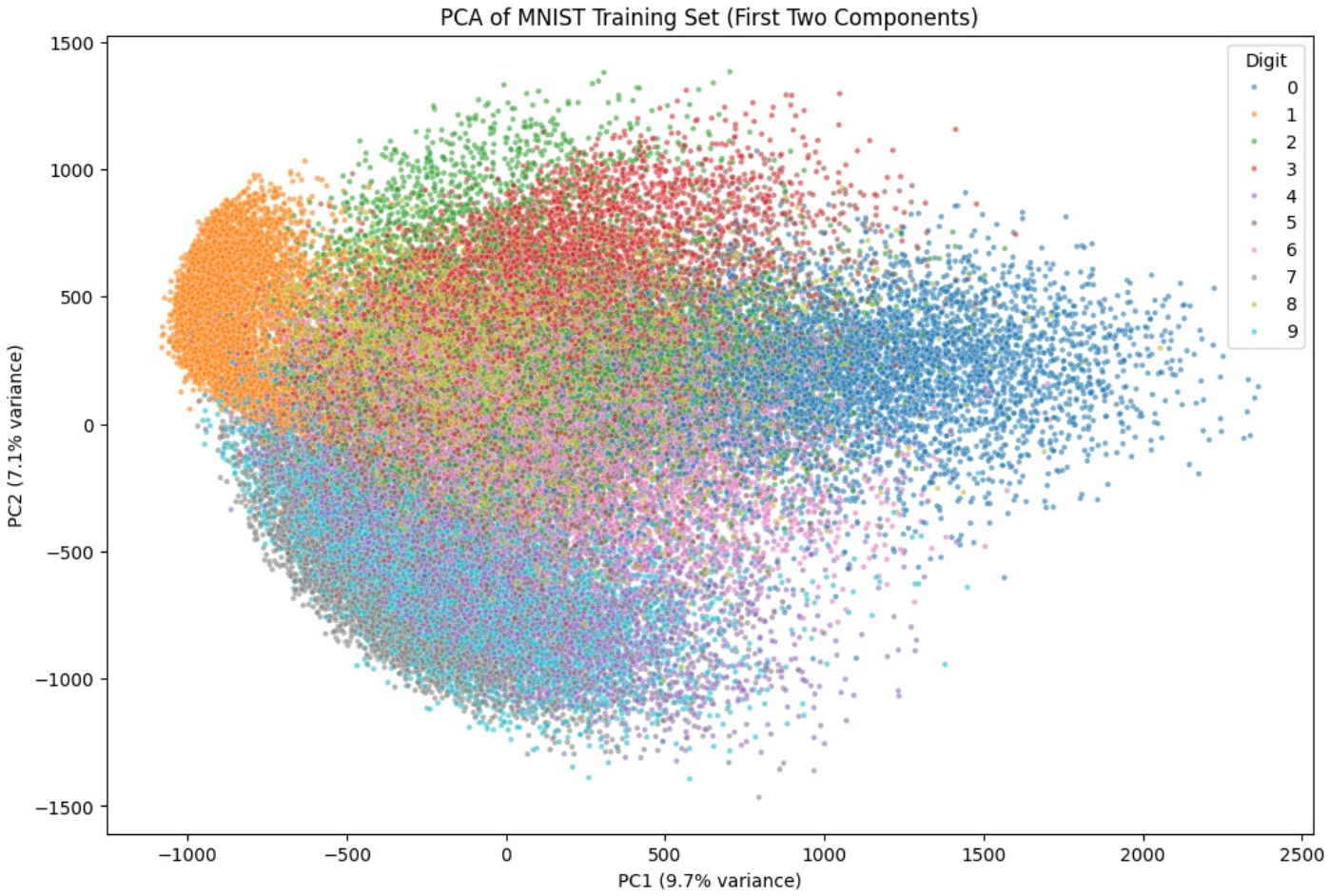
Let's analyze the distribution of pixel values in the dataset, using

In [5]:

```
# Perform PCA and plot
plt.figure(figsize=(12, 8))
pca = PCA(n_components=2)
X_pca = pca.fit_transform(train_data.drop('label', axis=1))
df_pca = pd.DataFrame(X_pca, columns=['PC1', 'PC2'])
df_pca['label'] = train_data['label']

# Plot PCA results with a categorical color palette
sns.scatterplot(
    data=df_pca,
    x='PC1',
    y='PC2',
    hue='label',
    palette='tab10',  # Palette with good contrast for categorical data
    alpha=0.6,
    s=10
)
plt.title("PCA of MNIST Training Set (First Two Components)")
plt.xlabel(f"PC1 ({pca.explained_variance_ratio_[0]:.1%} variance)")
plt.ylabel(f"PC2 ({pca.explained_variance_ratio_[1]:.1%} variance)")
plt.legend(title='Digit')

# Show all plots
plt.show()
```



In [ ]: