Object Classification

Machine Learning vs Deep Learning

Data Collection & PreProcessing

Importing necessary libraries and packages are imported. The MNIST dataset is downloaded and split into training and testing sets. Labeling train and test datasets

Feature Extraction with SIFT and Bag-of-Words

To process images in a dataset for feature extraction using the SIFT (Scale-Invariant Feature Transform) method, the following steps are typically followed:

1. Convert PyTorch Tensor Images to 8-bit Numpy Arrays:
First, images in the dataset, initially represented as PyTorch tensors, need to be converted to 8-bit numpy arrays. This conversion is essential because the SIFT feature extractor requires images in a numpy array format.

2. Dataset Images Conversion:
Each image in the dataset is individually converted into a numpy array. This ensures that all images are in the correct format for further processing.

3. SIFT Feature Extraction:
The SIFT algorithm is then applied to each image to extract key features. SIFT is a powerful feature extraction technique that identifies key points in images, which are invariant to scale and rotation, making them useful for various image analysis tasks.

4. Create Histograms of Visual Words:
Using the extracted SIFT features, histograms of visual words are created through k-means clustering. K-means clustering groups the SIFT features into clusters, and the histogram represents the frequency of these clusters (visual words) in each image.

5. Scale Data Using StandardScaler :
Finally, the data is scaled using StandardScaler to standardize the features by removing the mean and scaling to unit variance. This step ensures that the features are on the same scale, which can improve the performance of machine learning models.

Machine Learning with Bag-of-Words Representation

1. Extract Valid Labels:

Ensure that valid labels are extracted and aligned with the images containing SIFT descriptors. This step is crucial for supervised learning, where each image (with its corresponding SIFT descriptors) must have a label indicating its class.

2. Object Classification Using Machine Learning and Deep Learning:

- PCA for Dimensionality Reduction: Principal Component Analysis (PCA) is applied to the dataset to reduce the dimensionality of the feature space. This step helps in retaining the most important features while reducing computational complexity.
- Training the SVC: A Support Vector Classifier (SVC) is trained on the PCA-reduced data. The SVC is a powerful classification algorithm that works well for high-dimensional data.

Validation & Visualization

3. Evaluate Model Performance:

The model's performance is assessed using several metrics:

- Validation Accuracy: This metric provides an overall accuracy measure of the model on the validation dataset.
    - Validation Accuracy: **0.10515991471215352**

- Classification Report: This report includes precision, recall, and F1-score for each class, providing a detailed performance overview.
    - Validation Classification Report:

Class	Precision	Recall	F1-Score	Support
0	0.11	0.11	0.11	1203
1	0.11	0.22	0.15	1292
2	0.09	0.09	0.09	1168
3	0.12	0.1	0.11	1200
4	0.1	0.09	0.1	1150
5	0.11	0.08	0.09	1082
6	0.11	0.09	0.1	1126
7	0.1	0.1	0.1	1194
8	0.09	0.07	0.08	1157
9	0.09	0.07	0.08	1153

Deep Learning with CNN

1. Define the CNN Model:

Start by defining the Convolutional Neural Network (CNN) architecture. This typically involves specifying the number of layers, types of layers (convolutional, pooling, fully connected), activation functions, and other hyperparameters.

2. Compile the Model:

Compile the CNN model using the Adam optimizer and sparse categorical crossentropy loss. This step configures the model for training by setting the optimization strategy and the loss function.

3. Prepare the Data for Training:

Preprocess the MNIST dataset to make it suitable for training. This involves normalizing the pixel values, converting labels to the appropriate format, and splitting the data into training and testing sets.

4. Train and Evaluate the CNN Model:

Train the CNN model on the MNIST dataset. During training, the model learns to recognize patterns and features in the images. After training, evaluate the model's performance on the test dataset to measure its accuracy and effectiveness.

5. Test Accuracy:

Measure the accuracy of the CNN model on the test dataset.

- Test Accuracy: `0.9900000095367432`

6. CNN Classification Report:

Generate a classification report for the CNN model, which includes precision, recall, and F1-score for each class.

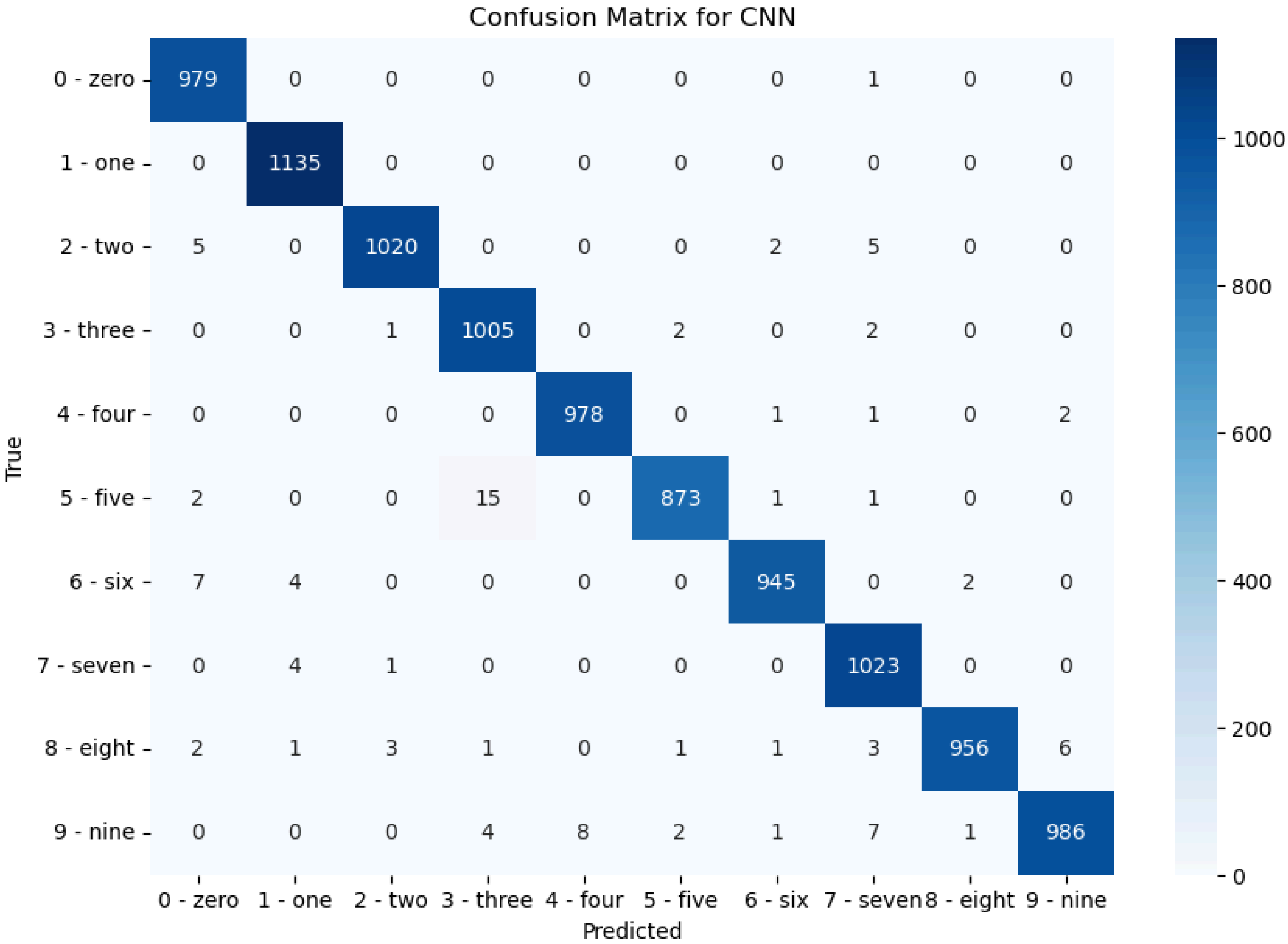
- CNN Classification Report:

Class	Precision	Recall	F1-Score	Support
0	0.98	1.0	0.99	980
1	0.99	1.0	1.0	1135
2	1.0	0.99	0.99	1032
3	0.98	1.0	0.99	1010
4	0.99	1.0	0.99	982
5	0.99	0.98	0.99	892
6	0.99	0.99	0.99	958
7	0.98	1.0	0.99	1028
8	1.0	0.98	0.99	974
9	0.99	0.98	0.98	1009

7. CNN Confusion Matrix:

Create a confusion matrix to visualize the model's performance. The confusion matrix shows the true versus predicted class distributions, helping to identify areas where the model may be struggling.

- CNN Confusion Matrix:



Conclusion

In this project, both machine learning and deep learning approaches were employed for object classification. The performance of the models was evaluated using various metrics such as precision, recall, F1-score, and accuracy. The results indicate that deep learning, specifically using a Convolutional Neural Network (CNN), achieved better performance compared to traditional machine learning methods.

Key Points:

1. Machine Learning with Bag-of-Words Representation:

- The model was trained using SIFT features and k-means clustering.
- The Support Vector Classifier (SVC) was used on PCA-reduced data.
- The classification report showed an overall accuracy of 11%, with relatively low precision, recall, and F1-scores across all classes.

2. Deep Learning with CNN:

- The CNN model was defined, compiled with the Adam optimizer and sparse categorical crossentropy loss, and trained on the MNIST dataset.
- The CNN achieved a test accuracy of 99%.
- The CNN classification report demonstrated high precision, recall, and F1-scores, with an overall accuracy significantly higher than the machine learning approach.

Conclusion:

The deep learning approach using a CNN significantly outperformed the machine learning approach based on Bag-of-Words representation and SIFT features. The CNN model demonstrated superior accuracy and more consistent performance across all classes, making it a more effective solution for object classification tasks. The high accuracy and robust performance of the CNN highlight the potential of deep learning techniques in achieving state-of-the-art results in image classification.