Exploring MNIST for Robust Handwritten Digit Classification

Mr. Pranav Dalvi
M.Sc Computer Science

Dr. Apurva Yadav
H.O.D MSc Computer Science

Dr. Neha Ansari
Lecturer M.Sc Computer Science

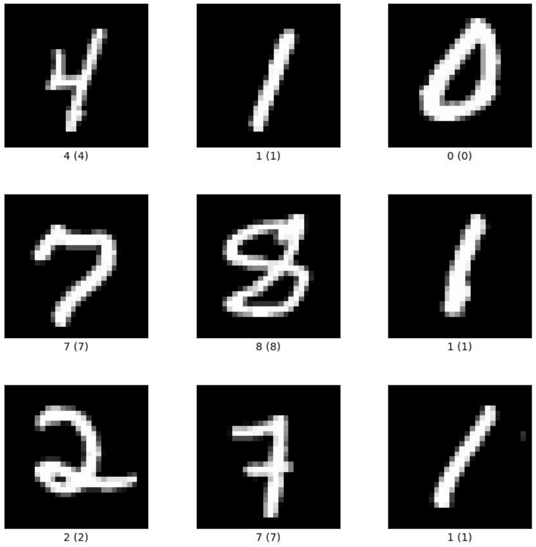
Institution: Kirti M. Doongursee College

Introduction

- Handwritten digit recognition is a critical component in various applications, ranging from automated postal mail sorting and bank check processing to digital document archiving and handwriting-based user interfaces.
- The Modified National Institute of Standards and Technology (MNIST) dataset has long been a benchmark for evaluating handwritten digit recognition algorithms. Introduced by LeCun et al.
- This research aims to explore the MNIST dataset for robust handwritten digit classification using Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Decision Trees (DT). By leveraging the capabilities of these models, we seek to enhance the performance of digit recognition systems.

Dataset

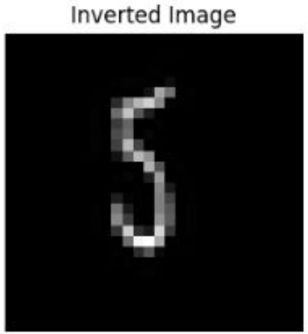
The Modified National Institute of Standards and Technology (MNIST) dataset has long been a benchmark for evaluating handwritten digit recognition algorithms. Introduced by LeCun et al., the MNIST dataset consists of 70,000 grayscale images of handwritten digits (60,000 for training and 10,000 for testing), each of size 28x28 pixels. This dataset has become a standard for testing and comparing various machine learning models.



https://www.tensorflow.org/datasets/catalog/mnist

Methodology

1. **Model Training**
   a. Pipeline Creation: Set up a machine learning pipeline with StandardScaler and the classifier (SVM, KNN, DT).
   b. Hyperparameter Tuning: Use GridSearchCV to find optimal hyperparameters.
2. **Evaluation**
   a. Cross-Validation: Perform 5-fold cross-validation to evaluate model performance.
   b. Test Set Evaluation: Assess the final model on a separate test set to determine accuracy.
3. **Algorithm Deployment**
   a. Load MNIST Dataset: Use fetch_openml to load the dataset.
   b. Train-Test Split: Split the dataset into training and testing sets.
   c. Fit Model: Train the model with the best hyperparameters identified by GridSearchCV.
   d. Predict and Visualize: Use the model to predict characters from new images and visualize the results.



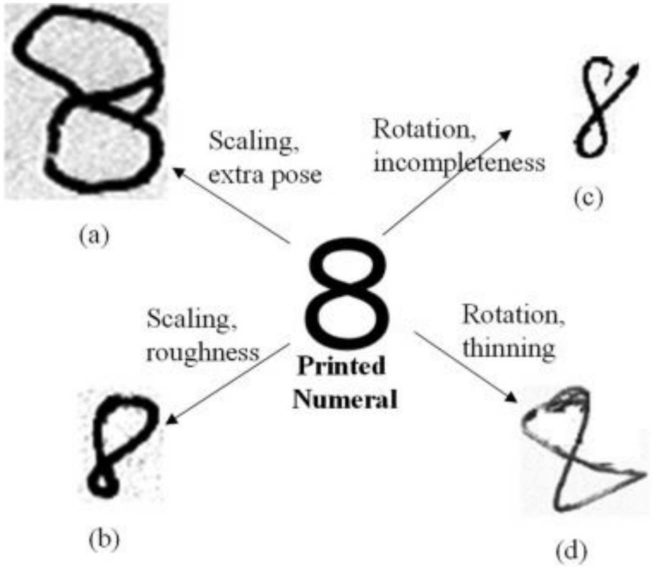
Results and Discussion

The performance of the SVM, KNN, and DT models was evaluated on the MNIST dataset using various metrics, including Test Accuracy, and Cross Validation Accuracy. The following table summarizes the best parameters and performance metrics for each model:

Model	Best Parameters	Cross Validation Accuracy	Test Accuracy
SVM	`C=10`, `gamma='scale'`, `kernel='poly'`	96.97%	97.18 %
KNN	n_neighbors=3`, `weights='distance'`, `metric='euclidean'`, `algorithm='auto' `	96.54%	96.62 %
DT	`criterion='entropy'`, `max_depth=30`, `min_samples_split=10`, `min_samples_leaf=5`, `splitter='best'`	85.69%	86.17 %

Challenges and Limitations:

- Achieving uniform accuracy across diverse handwriting styles.
- The variability in individual handwriting presents a significant challenge to the robustness of models like SVM, KNN, and DT, as these variations can significantly affect classification performance.
- The need for extensive preprocessing and hyperparameter tuning is emphasized, as achieving high levels of accuracy necessitates rigorous and often complex adjustments to the models.
- These processes are vital to align the diverse characteristics of handwritten samples with the trained models, ensuring that the recognition system can effectively interpret a wide range of handwritten inputs.



https://www.sciencedirect.com/science/article/pii/S1319157822002270

Future Enhancement

- **Advanced Models:** While these models performed well, exploring advanced deep learning architectures, such as deeper Convolutional Neural Networks (CNNs), could potentially yield even better results.
- **Automation and Scalability:** Develop automated preprocessing pipelines and scalable training frameworks to handle larger datasets and more complex models efficiently.
- **Dataset Augmentation:** Incorporate more diverse handwriting samples to enhance model robustness against varying styles and noise.
- **Real-World Application:** Extend the research to include real-world handwritten documents and multilingual character recognition to evaluate model performance in practical scenarios.

References

1. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. _Proceedings of the IEEE_, 86(11), 2278-2324. https://doi.org/10.1109/5.726791
2. Cortes, C., & Vapnik, V. (1995). SupportVector Networks. _Machine Learning_, 20, 273-297. https://doi.org/10.1007/BF00994018
3. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
4. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. _Journal of Machine Learning Research_, 12, 2825-2830. https://jmlr.org/papers/v12/pedregosa11a.html
5. Salakhutdinov, R., & Hinton, G. E. (2007). Learning a Nonlinear Embedding by Preserving Class Neighbourhood Structure. _Artificial Intelligence and Statistics (AISTATS)_.
6. Schölkopf, B., & Smola, A. J. (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.
7. Platt, J. (1998). Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Microsoft Research Technical Report MSR-TR-98-14.