A Comparative Study in digits recognition through Convolutional Neural Network and Support Vector Machines

Nizar Assad
Nizar.Assad@city.ac.uk

Abstract

To classify the MNIST dataset, this study compares Convolutional Neural Network (CNN) and Support Vector Machine (SVM) models in-depth. The study examines their implementation and performance features, illuminating the applicability of each strategy for the difficult task. While the SVM searches for the best hyperplanes for classification, the CNN uses convolutional layers to extract hierarchical visual information [1]. The experimental setup for the study entails designing a unique CNN architecture and evaluating SVM using linear and non-linear methods. Our comprehension of the capabilities of these models is improved by a thorough review of performance indicators, such as confusion matrices and accuracy scores. This study provides helpful advice for choosing the right models for picture classification tasks.

1. Introduction

The Convolutional Neural Network (CNN) and Support Vector Machine (SVM), two machine learning models, are compared and evaluated in this research. They are evaluated for their performance in discovering the best hyperparameter combinations and categorising the MNIST dataset, a collection of handwritten digit pictures. The foundation for this study is MNIST, which is frequently used for comparing various machine learning techniques [2]. The well-known and distinct CNN and SVM models are contrasted to identify their advantages and disadvantages in the difficult MNIST classification job. The article provides suggestions to researchers and professionals on how to choose appropriate models for picture categorization. The research entails using MNIST to train and assess both models, along with in-depth analysis and visualisations to comprehend model behaviour and the influence of hyperparameters. Predictive abilities are measured by performance indicators such as confusion matrices and accuracy ratings. The best-trained CNN and SVM models are compared in-depth in the report's conclusion, along with their advantages and shortcomings. The article ends with prospective future research topics in image classification and practical ramifications of the findings.

1.1. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are a type of deep learning architecture created specifically for problems involving images. Convolutional layers are used to separate out hierarchical features from the input data. The network can automatically learn key patterns and attributes present in the images thanks to these convolutional layers [1].
A customised architecture is developed for the CNN model used in this work to precisely address image processing needs. The performance of the CNN is improved by hyperparameter adjustment, enabling a thorough investigation of various learning rates, batch sizes, and epochs.

1.2. Support Vector Machines (SVM)

Within the classification and regression analysis fields of machine learning, a Support Vector Machine (SVM) stands as a reliable and commonly used machine learning approach. Its main idea revolves around finding the best hyperplane that can clearly separate data points from different classes while also maximising the margin between these classes. The SVM is given the ability to produce precise predictions for instances of data that haven't been seen before thanks to this hyperplane acting as a decision boundary. When it is impossible to separate data points linearly, the SVM performs exceptionally well. This is accomplished by using kernel functions to convert the input space into a higher-dimensional feature space, which enables the SVM to successfully handle complex relationships between data points [3].

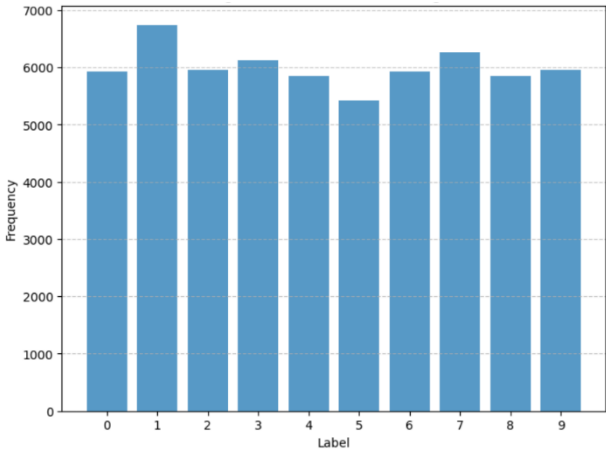
The SVM model is evaluated simultaneously using both linear and non-linear (kernelized) techniques. The Radial Basis Function (RBF) kernel is chosen as the best option for the non-linear SVM model after a careful analysis of kernel accuracies. The best pairing of the gamma parameter and the regularisation parameter (C) for the SVM model is found after a thorough investigation of hyperparameter optimisation.

2. Dataset

The MNIST dataset is a sizable collection of 70,000 grayscale pictures, each of which measures 28 by 28 pixels and shows handwritten numbers. This collection provides a solid foundation for assessing the effectiveness of machine learning models in tasks related to digit identification and image categorization [2]. This dataset's pixel values, which range from 0 to 255, provide a subtle depiction of the handwritten numbers. The MNIST dataset's scrupulously preserved equilibrium in class distribution is one of its most notable features. Figure 1 shows that the 10 separate digit classes, which range from 0 to 9, are evenly distributed, preventing any potential bias and guaranteeing a balanced foundation for model training.

Remarkably, each digit class consists of a sizable subset of roughly 6,000–7,000 samples, offering plenty of data for effective learning. A further examination of pixel-level characteristics reveals a mean pixel value of 0.1307, indicating the dominant brightness in the photos. The pixel intensities cover the entire grayscale spectrum, varying between 0.0 and 1.0, capturing a broad spectrum of tones. The dataset's variability is attested to by the standard deviation of 0.3081, which highlights the dispersion of pixel values. In conclusion, the MNIST dataset is a cornerstone for the research of digit recognition and picture classification due to its even class distribution and pixel-level complexity. Its well-balanced structure and rich pixel properties make it easier to explore and create machine learning algorithms for various types of visual data processing.

Figure 1 - Histogram of Labels in the Training Data



3. Methods

3.1. Methodology

We describe the thorough technique used for the MNIST digit classification problem in this section. This comprehensive strategy required careful data preprocessing, careful model architecture creation, exacting hyperparameter tuning, and methodical model evaluation. It was also carefully compared how well Convolutional Neural Networks and Support Vector Machines handled the classification task made available by the MNIST dataset. Loaded and divided into training, validation, and test sets was the dataset. For the models to work, images were converted into tensor format. The photos were flattened and normalised for the SVM technique by scaling to a range of [0, 1]. Additionally, stratified sampling was used to further divide the training data into training and validation sets for SVM.

3.2. Architecture and Parameters used for the CNN

The Convolutional Neural Network (CNN) architecture was carefully constructed to take advantage of the spatial hierarchies found in images. Convolutional layers and completely connected layers made up its two main parts. Through a sequence of filters, followed by a rectified linear unit (ReLU) activation for each filter, the convolutional layers extracted complex information from the input images. The most important information was caught by later max-pooling layers, which also helped with dimensionality reduction and translation invariance [4]. Fully connected layers made up the CNN's top layer, successfully translating newly learnt information to appropriate class labels. The network architecture was designed to provide hierarchical pattern recognition and feature extraction.
A methodical grid search was used to approach the CNN's hyperparameter tunning. Exploring different combinations of hyperparameters, such as learning rate, batch size, and number of epochs, was a part of this process. The batch size affected the quantity of training samples processed prior to updating the model's parameters, whereas the learning rate affected the step size during gradient descent optimisation. The total iterations throughout the full training dataset depended on the number of epochs. To balance effective convergence with avoiding overfitting, several hyperparameters were chosen and adjusted [5].

3.3. Architecture and Parameters used for the SVM

Accurate and reliable findings were made possible by the architecture of the Support Vector Machine (SVM) models used for the MNIST digit classification problem. We investigated the Linear SVM and the Non-linear SVM using a Radial Basis Function (RBF) kernel in our search for the best performance.
A fundamental model, the Linear SVM, sought to identify the best hyperplane for separating the various data classes inside the input feature space. This strategy mostly made use of linear decision boundaries. Despite being straightforward, the Linear SVM demonstrated limits in capturing complex and nonlinear relationships in the dataset, which reduced the overall accuracy of its classification.
The Non-linear SVM with the RBF kernel, however, has become a strong substitute. We gave the SVM the ability to translate the data into a higher-dimensional space where complicated relationships and patterns could be more easily seen by utilising the RBF kernel. This change improved the SVM's capacity to precisely categorise the digits found in the MNIST dataset by making it easier to capture complex decision boundaries.

The performance of the SVM models was improved significantly by hyperparameter adjustment. The gamma value, which affects the RBF kernel's form, and the regularisation parameter C, which balances the trade-off between training and testing errors, were carefully optimised for the Non-linear SVM. The model's improved classification performance was ultimately a result of the rigorous parameter tunning that made sure it was perfectly tailored to the nuances of the dataset. Performance of the SVM models was enhanced by the architectural design, strategic use of the RBF kernel, thorough hyperparameter optimisation, and other factors working together.

4. Results, Findings & Evaluation

4.1. Model Selection

The process of model selection for the MNIST digit classification task unfolded through a systematic sequence of steps designed to thoroughly evaluate and choose the most proficient model. Subsequently, an optimized Convolutional Neural Network (CNN) underwent meticulous evaluation through hyperparameter tuning and convergence analysis. Meanwhile, two Support Vector Machine (SVM) architectures, Linear SVM and Non-linear SVM with an RBF kernel, were explored and thoroughly evaluated. A meticulous comparative analysis ensued, encompassing performance metrics, confusion matrices, classification reports, and Receiver Operating Characteristic (ROC) analysis. The subsequent sections delve into detailed outcomes and insights from this comprehensive model selection approach.

4.2. Algorithm Comparison

In this section, we embark on a comprehensive exploration and juxtaposition of two robust machine learning models, the Convolutional Neural Network (CNN) and the Support Vector Machine (SVM), in the context of the challenging MNIST digit classification task. Our scrutiny extends across multiple dimensions, encompassing architectural intricacies, hyperparameter optimization, performance metrics, and model robustness.

4.2.1. Convolutional Neural Network (CNN)

The chosen CNN architecture was meticulously designed to harness the intrinsic spatial hierarchies inherent to images. Comprising a cascade of convolutional and fully connected layers, the CNN excelled in capturing intricate features and hierarchies, substantiated by the Confusion Matrix and Classification Report.

The CNN achieved a commendable test accuracy of 0.9362, reflecting its adeptness in discerning digit patterns. The Confusion Matrix **(Figure 1)** showcases a comprehensive picture of its classification performance. Evidently, the CNN exhibits robust performance with a test accuracy of 0.9362. It is particularly effective in correctly identifying digits '0,' '1,' and '4.' However, it shows some challenges in distinguishing between digits '2' and '7,' as well as between '5' and '3.' These misclassifications are crucial for further analysis and optimization.

Precision and recall rates for each class further underscore its reliability. Notably, the weighted F1-score of 0.94 reinforces the CNN's ability to maintain a harmonious balance between precision and recall, indicative of its suitability for real-world scenarios.

Hyperparameter tuning played a pivotal role in optimizing the CNN's performance. Through an exhaustive grid search, parameters such as learning rate, batch size, and epochs were systematically adjusted. The resultant CNN demonstrated a well-calibrated learning process, allowing for efficient convergence while mitigating overfitting. This adaptive fine-tuning facilitated the model's ability to generalize effectively and perform robustly on unseen data.

4.2.2. Support Vector Machine (SVM)

Within the SVM realm, our exploration encompassed two distinct architectures: the Linear SVM and the Non-linear SVM with an RBF kernel. The latter emerged as the victor, achieving an impressive test accuracy of 0.9689. This heightened accuracy can be attributed to the RBF kernel's capacity to transform input data into a higher-dimensional space, thereby enabling nuanced pattern recognition. The Confusion Matrix **(Figure 2)** for the Non-linear SVM with an RBF kernel underscores its superior performance with a remarkable test accuracy of 0.9689. The SVM model exhibits exemplary precision and recall across various classes, further reflected in the weighted F1-score of 0.97. This model demonstrates exceptional accuracy in classifying digits '0,' '1,' '2,' and '4.' It shows slight challenges in distinguishing between '3' and '8,' as well as between '9' and '4.' These findings emphasize the SVM's prowess in capturing intricate patterns and achieving high accuracy.
Hyperparameter tuning for the Non-linear SVM delved into K-Fold Cross-Validation, ensuring robust parameter selection. The optimization of gamma and regularization parameter C underscores the meticulousness in parameter selection, reflecting a balance between capturing intricate decision boundaries and mitigating overfitting.

4.2.3. Comprehensive Evaluation

The micro-average Receiver Operating Characteristic (ROC) curve **(Figure 3)** was plotted for both models, showing their ability to distinguish between classes. Surprisingly, both models achieved an Area Under the Curve (AUC) of 0.99, indicating high discriminative power across different classes. This suggests that both models are capable of effectively separating the digits, even though they have different underlying mechanisms.
Upon close inspection of the Confusion Matrices, it becomes evident that the Non-linear SVM with the RBF kernel surpasses the CNN in terms of overall classification performance. The SVM model showcases superior accuracy and precision, particularly in identifying complex patterns and distinguishing between digit classes. While the CNN demonstrates commendable performance, its ability to handle intricate variations in the dataset appears slightly constrained in comparison. Consequently, the Non-linear SVM with the RBF kernel emerges as the victor, demonstrating its potential for intricate pattern recognition tasks.

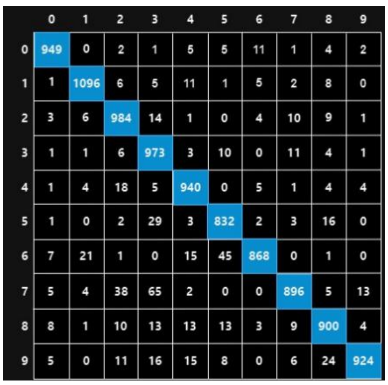


Figure 1 Confusion Matrix for CNN

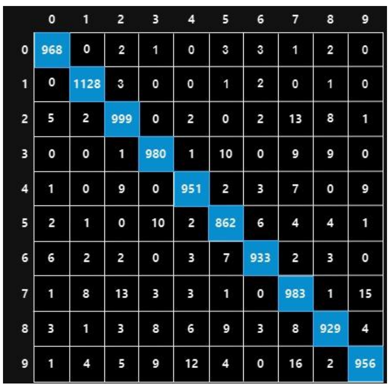


Figure 2 Confusion Matrix for SVM

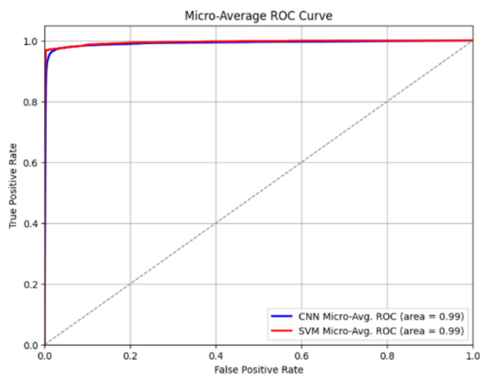


Figure 3 Micro-Average ROC Curve

5. Conclusion

In conclusion, our comprehensive evaluation of the Convolutional Neural Network (CNN) and the Support Vector Machine (SVM) for the MNIST digit classification task reveals intriguing insights into their respective strengths and capabilities. While the CNN showcases commendable accuracy and balanced performance, its spatial hierarchies and feature extraction capabilities contribute to its robustness. On the other hand, the Non-linear SVM with the RBF kernel stands out with its remarkable accuracy, exceptional precision, and adeptness in capturing intricate patterns [6]. The meticulous hyperparameter tuning and the RBF kernel's transformative power contribute to its superiority in digit classification.

Furthermore, the micro-average ROC curve's remarkable Area Under the Curve (AUC) of 0.99 for both models underscores their high discriminative ability, implying their efficacy in distinguishing between different digit classes. This finding is particularly significant as it highlights the models' consistent performance across diverse classes.

In the context of the MNIST digit classification task, the Non-linear SVM with the RBF kernel emerges as the winner, offering unparalleled accuracy and precision. However, it is essential to acknowledge that both models contribute significantly to the field of digit classification, each bringing its unique strengths. The insights gleaned from this study empower data scientists and machine learning practitioners to make informed decisions when selecting a model for intricate pattern recognition tasks, ultimately advancing the realm of image classification and pattern analysis [7].

As the realm of machine learning and deep learning continues to evolve, these findings lay the groundwork for further explorations and optimizations, contributing to the ongoing advancement of accurate and robust digit classification algorithms.

6. Reference

[1] S. Virnodkar a, a, & b. (2020, September 15). CaneSat dataset to leverage convolutional neural networks for sugarcane classification Journal of King Saud University - Computer and Information Sciences.

[2] Etzold, D. (2022, January 13). MNIST‐dataset of handwritten digits. Medium. https://medium.com/mlearning-ai/mnist-dataset-of-handwritten-digits-f8cf28edafe

[3] Saini, A. (2023, July 7). Guide on Support Vector Machine (SVM) algorithm. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

[4] GeeksforGeeks. (2023, March 24). Introduction to convolution neural network. https://www.geeksforgeeks.org/introduction-convolution-neural-network/

[5] Rendyk. (2023, May 18). Tuning the hyperparameters and layers of neural network deep learning. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/05/tuning-the-hyperparameters-and-layers-of-neural-network-deep-learning/

[6] LeCun, Y., Cortes, C., & Burges, C. (1998). The MNIST Database of Handwritten Digits. Retrieved from http://yann.lecun.com/exdb/mnist/

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.