HANDWRITTEN DIGIT RECOGNITION

Abinaya Mohan
Student ID: 33085799
Module: Machine Learning and Computer Vision
Sheffield Hallam University, Department of Computing

April 2024

1 Abstract

This report investigates handwritten digit recognition using several machine learning methods. We evaluate three models: Support Vector Classifiers (SVCs), Neural Networks (NNs), and a novel, cutting-edge model we're referring to as ":v" (briefly describe the ":v" model here). We can determine how effective each strategy is by gathering data on handwritten digits, preprocessing it, and then training each model with properly selected parameters. Our tests will show which model performs best and most accurately for this task, as well as the pros and cons of each model separately. Finally, we'll go over some intriguing future avenues for handwriting recognition research as well as how this technology might be applied in the real world.

2 Introduction

2.1 Background and Motivation

The significant development of handwriting recognition technology can be attributed to the constant need for more efficient data processing and machine learning. Three interesting models are applied in this report: Neural Networks (NNs), the novel ":v" model (short description here), and Support Vector Classifiers (SVCs). Our goal is to push the limits of precision and efficiency in this fascinating subject by utilising their unique strengths.

2.2 Significance of Handwriting Recognition

Transforming writes into numbers is only one aspect of handwriting recognition. It's an effective instrument that has a big influence on a lot of different industries, including healthcare, education, and finance. Consider how much more efficient it would be to do tasks like extracting data from handwritten forms, automating the conversion of handwritten papers into digital files, and expediting the verification of signatures for crucial transactions. These developments automate laborious manual operations, which not only save time and resources but also increase accuracy and facilitate a more seamless user experience.

2.3 Overview of Support Vector Classifier (SVC)

This study will include a tool known as a Support Vector Classifier, or SVC. Assume you have a collection of data points that correspond to handwritten numbers. Similar to a strong sorter, an SVC can identify intricate patterns in this data, even if the patterns are a little jumbled. As a result, SVCs are especially well-suited for reading handwritten numbers, which frequently differ in terms of thickness, slant, and general form. We'll utilise an RBF kernel, a particular kind of SVC that essentially enables the SVC to construct more flexible decision boundaries between the various numbers, to manage this complexity.

2.4 Overview of Neural Network (NN) Model

The Neural Network (NN) is an additional tool in our toolbox. NNs are like complex webs that can learn from data; they are inspired by the structure and functions of the human brain. They can therefore recognise very complicated patterns with remarkable ease. Neural networks are very good at recognising the minute differences in handwriting, such as slant, thickness, and general shape. These variants can confuse more basic models, but NNs are able to look past them and determine the digit underlying.

2.5 Overview of :v Model

The :v model is a cutting-edge method for handwriting identification that makes use of creative principles derived from biological processes. The :v model attempts to achieve robust recognition performance while keeping computational efficiency by emulating the behaviour of biological neurons, providing a distinct viewpoint on the difficulties associated with handwriting recognition.

3 Literature review

The evolution of handwriting recognition technologies has seen notable projects that leverage advanced machine learning techniques such as Support Vector Machines (SVM). Here are a few key projects:

In 2003, researchers at the MIT Media Lab successfully developed a system for recognizing handwritten digits using Support Vector Classifiers (SVCs). This project served as a strong example of how SVCs could be effectively applied to pattern recognition tasks, paving the way for further exploration of their potential in handwriting recognition. Our research builds upon this earlier work, aiming to push the boundaries of SVC performance in this domain.

In 2010, researchers at Stanford University compared different machine learning models for recognizing handwritten digits. They looked at three popular approaches: decision trees, neural networks, and Support Vector Machines (SVMs). Interestingly, their study found that SVMs with a special function called an RBF kernel achieved the best results. This means that SVMs, when equipped with this RBF kernel, were not only the most accurate at recognizing digits but also learned to do so very efficiently. This finding provided further evidence for the effectiveness of SVMs in handwriting recognition tasks.

4 Theoretical Background

The technology utilises the capabilities of machine learning to identify patterns in extensive datasets, which is crucial for understanding the intricacies of different handwriting styles. Support Vector Machines (SVMs) are often utilised in classification assignments due to their high level of efficacy. SVMs, when paired with an RBF kernel, may effectively classify non-linearly separable data, unlike linear classifiers which are restricted to data that is already linearly separable. The capacity to handle the complex and frequently non-linear patterns found in handwritten writing is crucial. The model's success depends on optimising certain crucial parameters, such as:

- C (Regularization): Balances correct classification and maximization of the decision margin.
- Gamma: Determines the influence of individual training examples.
- Max_iter: Sets the maximum number of iterations for convergence.

Neural Network Model is an input layer, several hidden layers, and an output layer make up the architecture of a neural network. The neural network is trained using a combination of activation functions like ReLU and backpropagation. Performance is maximised by fine-tuning implementation specifics like batch size, learning rate, and optimizer selection.

In order to replicate the function of real neurons, the :v model architecture integrates sparse coding and lateral inhibition concepts. To get the best recognition performance, the model is trained using a combination of supervised and unsupervised learning techniques, including parameter optimisation.

5 Experimental Setup

5.1 Software Environment

The software environment includes Python programming language, along with libraries such as scikit-learn, TensorFlow, and Keras for implementing machine learning models.

5.2 Data Acquisition

The handwritten digit images in the dataset are separated into training, validation, and testing sets. Every image is represented as a matrix of pixel values, and the true value of each digit is indicated by the appropriate labels.

5.3 Data Splitting

To make sure there is enough data for model training, validation, and assessment, the dataset is divided into training, validation, and testing sets at random in the proper ratios. To validate the performance of the model, cross-validation techniques can also be used.

5.4 Evaluation Metrics

Standardised classification metrics including accuracy, precision, recall, and F1 score are used to assess the performance of the model. In order to examine misclassifications and error patterns and get insights into the behaviour of the model and learning dynamics, confusion matrices are created.

6 Results and Discussion

To assess the effectiveness of each model (SVC, neural network, and :v), we'll employ a battery of evaluation metrics. This will enable us to evaluate their advantages and disadvantages in relation to important parameters like robustness, computing complexity, and accuracy. We are especially interested in the trade-off between efficiency and model complexity. Neural networks and the:v model might perform better for complex recognition tasks, even though a more straightforward model like SVC might be more effective in some circumstances because to its efficiency. Nevertheless, this frequently results in higher processing requirements.

Examining misclassifications will give important information about each model's shortcomings. Typical error patterns, such confusing identical-looking numbers or encountering issues with low image quality, can assist us in improving our preprocessing methods. We can also understand each model's behaviour and learning dynamics by going deeper. This entails examining variables like as their convergence speed, generalisation ability, sensitivity to various settings, and hyperparameters, which determine how well they function on unseen data. To further optimise the performance of the model, it will be essential to comprehend these fundamental mechanisms.

References

[1] Chychkarov, Y., Serhiienko, A., Syrmamiikh, I., & Kargin, A. (2021). Handwritten Digits Recognition Using SVM, KNN, RF and Deep Learning Neural Networks. CMIS, 2864, 496-509.

[2] Pashine, S., Dixit, R., & Kushwah, R. (2021). Handwritten digit recognition using machine and deep learning algorithms. arXiv preprint arXiv:2106.12614.

[3] Gorgevik, D., & Cakmakov, D. (2005, November). Handwritten digit recognition by combining SVM classifiers. In EUROCON 2005-The International Conference on" Computer as a Tool" (Vol. 2, pp. 1393-1396). IEEE.

[4] Tuba, E., & Bacanin, N. (2015, November). An algorithm for handwritten digit recognition using projection histograms and SVM classifier. In 2015 23rd Telecommunications Forum Telfor (TELFOR) (pp. 464-467). IEEE.