A Comparative Study of Classification Algorithms with Varying Training Dataset Sizes on Cursive Hiragana Characters

Harshil Shah[1], Dr.Rajkumar Rajasekaran[2],Dr.Jolly Masih[3]
[1,2]Vellore Institute of Technology,Vellore
[3]Prestige Institute of Engineering Management and Research, Indore

Abstract. The quantity of data samples available for training a machine learning model For that purpose, we have decided to compare the performance of 4 commonly used classification algorithms at 14 different data availability points. The algorithms used were the Support Vector Machine (SVM), Multilayer Perceptron (MLP) and 2 different architectures of the Convolutional Neural Network (CNN). A standard 10 layer architecture and an 11 layer VGG-like architecture. To perform image classification can vary based on several circumstantial factors, and the changing performance of classification algorithms relative to this data availability holds significance in choosing the right algorithm. We have compared these algorithms specifically at the percentages 0.01, 0.1, 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 of 60000 training samples from the Kuzushiji-MNIST (KMNIST) dataset, a MNIST-like data set with cursive Hiragana characters in place of numerical digits. The classification accuracies tested on 10000 data samples from the KMNIST dataset showed that the 10 layer CNN outperformed all other algorithms at each data availability point. The SVM and MLP showed higher accuracies than the 11 layers CNN for data availability<=0.1% of the total training dataset, but the latter outperformed them when using larger training data sizes.

Keywords—Machine Learning, Classification, Support Vector Machine, Convolutional Neural Network, Kuzushiji-MNIST

1. Introduction

Character recognition has long been one of the key focuses on research in image processing and machine learning, and several state-of-the-art methods that perform very close to human recognition have been developed. This holds great significance in automating several systems using Natural Language Processing applications such as spelling and grammar checkers. While achieving the maximum accuracy has been the focus in developing these models, in several circumstances, the amount of data available can be constrained. This is especially true for smaller industrial applications, where the training data is limited, and knowing which model to choose based on the amount of data available can save a lot of time that would otherwise be spent on trial and error. To this end, we delve into the performance of 4 distinct algorithms in this work, namely the SVM, the MLP, and 2 architectures of the CNN, at different levels of available training data, ranging from 0.01% to a full 100% of the total training data (60000 training images). For the purpose of this analysis, we have used the Kuzushiji-MNIST (KMNIST) [11] dataset, which is similar to the popular MNIST dataset, in that it comprises images with dimensions 28x28 labelled into 10 categories. Each of these categories represents a row in the table of cursive Hiragana characters, a phonetic lettering system used in the Japanese language. The cursive style of Hiragana is not widely used in contemporary Japan, but holds highly valuable cultural significance, and several efforts have been made to ensure the preservation of texts utilizing it. Optical character recognition is one such method that could go a long way in aiding the digital preservation of these works. In this paper, the classification accuracies of each of the algorithms at each data availability point is compared and discussed, and to the best of our knowledge, an analysis of this kind of the above-mentioned algorithms has not been performed.

2. Literature Review

Ghosh et al. In [1] use a simple CNN model to achieve a state-of-the-art accuracy for models of similar simplicity on the KMNIST dataset. One of the CNN architectures analysed in our work showed a similar, but slightly lower accuracy compared to the CNN used in [1] for cases where at least 80% of the training data was available. In [2], Alcantara analyses the performance of deep neural networks with several different non-linear activation functions for the layers of the neural network. The results showed that the ReLu, Leaky ReLu, ELU and SLU all performed well on the classification task, with ELU outperforming the others for this task. Lamb et al. In [3] discuss the history of the Kuzushiji, or cursive Hiragana style of writing and its cultural significance and go on to propose a new model, KuroNet, which uses residual U-Nets with improved regularization to achieve a higher accuracy than previous models for the same task. It further discusses the model's excellent performance on texts written in Kuzushiji and problems that come with implementing it on printed texts with messy data. In [4], Kotsiantis reviews 6 different supervised machine learning classification techniques and how they compare with each other in several aspects of performance. In terms of simple classification accuracy, the paper ranks the SVM and Neural Networks higher than Decision Trees, k-Nearest-Neighbors, Rule-learners and Naive Bayes, but some of the lower accuracy classifiers can be trained much faster than the former. In [5], Keysers reviews 4 existing state-of-the-art techniques for the popular MNIST dataset including the CNN and compares their performance on the dataset. Further, the paper uses combinations of the techniques to propose a new hybrid technique that is shown to achieve better results and an error rate as low as 0.35%.Meshkini et al. In [6] compare the performance of 6 high performing deep Convolutional Neural Network architectures, namely the GoogleNet, VGG, ResNet, SqueezeNet, Alex Net and DenseNet on the Fashion MNIST dataset, a dataset similar to the MNIST dataset, but with clothing items in place of the digits. Abdul razzaq et al. In [7] compare 3 different classification algorithms on the NIST handwritten digits dataset, namely the Naive Bayes, K_Star and Multilayer Perceptron. The goal of this work was to use the lowest required number of selected features to achieve an acceptable accuracy and the paper showed the K_Star algorithm to outperform the others in this task.

3. Methodology

3.1 Process Flow

The flow of the processes performed for conducting the analyses is depicted in Figure 1 below.

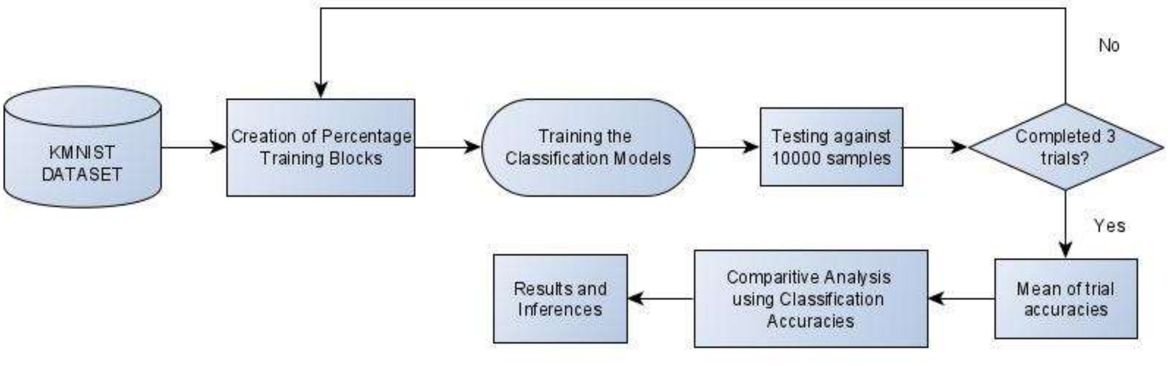


Figure 1: Process Flow Diagram

As shown in Figure 1, each of the 4 selected algorithms is trained separately on each subset of the KMNIST training dataset with 60000 labelled training images. These subsets are taken as a percentage of the total training data, and the specific percentages used were 0.01%, 0.1%, 1%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100%. The classification accuracy was determined by testing the models trained on the KMNIST testing data set, which comprises 10000 labelled images distinct from the training data set. This process was repeated for 3 trials and the mean of the classification accuracies from these trials were noted for further analyses. Based on these mean accuracies, the models were then compared with each other at the different percentages of data that they were trained on, which will be discussed in the Results section of this paper. All the experiments were done using Google Colaboratory [10] notebooks.

3.2 Algorithms Used:

3.2.1 Support Vector Machine (SVM):

The SVM is one of the most commonly used machine learning algorithms, mainly used for classification tasks. It works by determining hyper planes that divide the n-dimensional space, and can then be used to categorize examples. For this work, we have used the sockit library [8] to train the SVM model on the dataset and determine its classification accuracy. The model was trained using the RBF kernel and the gamma parameter set to scale.

3.2.2 Multi-layer Perceptron (MLP)

The MLP is one of the simplest architectures of an artificial neural network and is widely used for a variety of tasks. For this work, we have used the Keras library [9] to implement and train the model. The architecture of the MLP used is as depicted in Figure 2. The model was trained using the Adam optimizer and the Sparse Categorical Cross-entropy loss function.

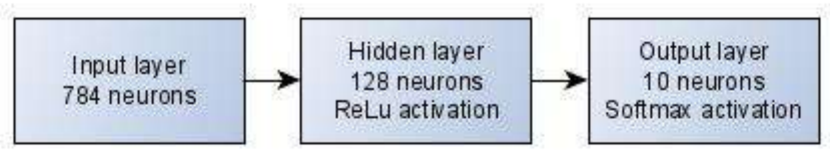


Figure 2: Architecture of the MLP

3.2.3 Convolutional Neural Network (CNN) - 1

The CNN is a deep neural network architecture that is widely used for image classification and has produced state-of-the-art results in several image classification tasks. We have implemented and trained the CNN models using the Keras library. In this work, we have used 2 different CNN architectures, the first of which has a standard 10 layer architecture as depicted in Figure 3 below. Both CNNs were trained using the Adam optimizer and the Sparse Categorical Cross-entropy loss function.

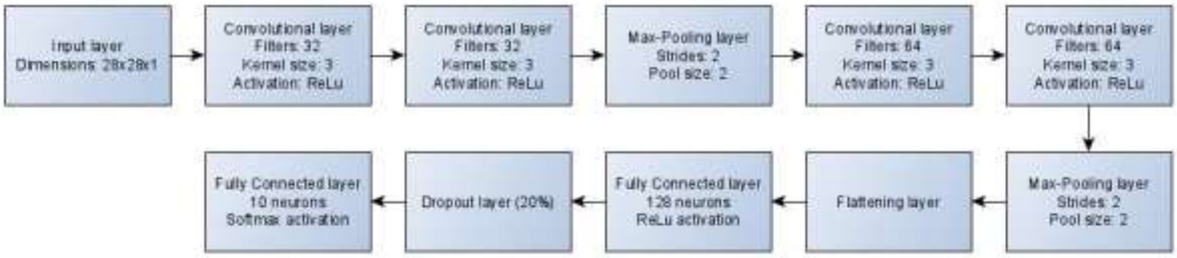


Figure 3: Architecture of first CNN

3.2.4 Convolutional Neural Network (CNN) - 2

In this second CNN model, we have used a VGG-like architecture, in that it follows the sequence of a Convolutional layer, followed by a Max-Pooling layer as depicted below in Figure 4.

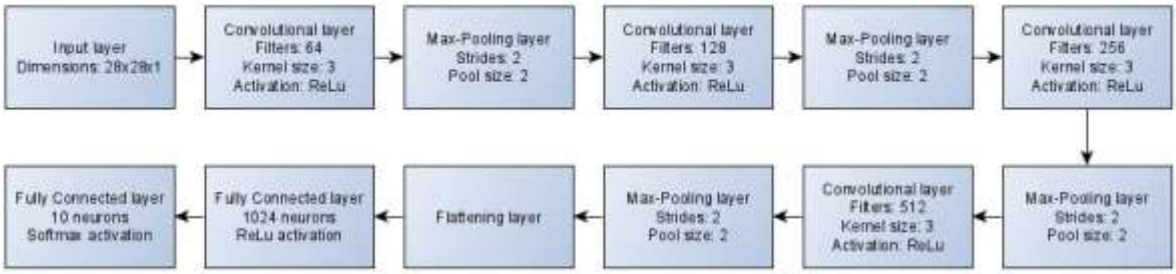


Figure 4: Architecture of second CNN

3.3 Dataset Used:

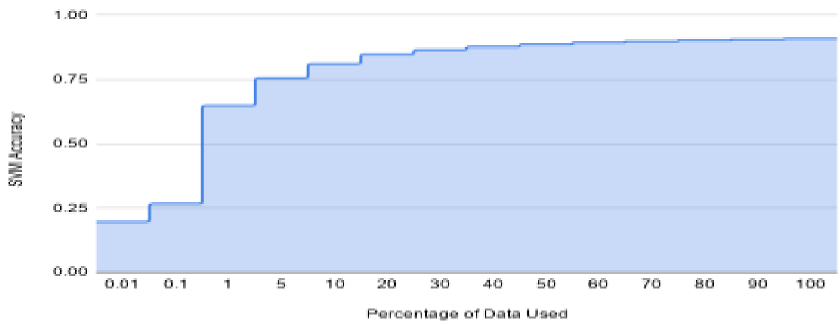
For the purpose of this work, we have used the Kuzushiji-MNIST dataset. This dataset comprises 60000 labelled training images and 10000 labelled testing images of cursive Hiragana characters. Hiragana is one of the lettering systems used in the Japanese language, and Kuzushiji, or cursive Hiragana was a style of writing hiragana characters that were in use for over a millennium, but unfortunately, is no longer commonly used. The images in the dataset are greyscale and of dimensions 28x28, each image depicting a particular Hiragana character. The labels are 10 in number, each of which represent a row of characters in the Hiragana script. The classification is based on these rows and each character is classified as belonging to its corresponding row.

4. Results and Discussion

Upon training and testing the models using the methods discussed so far, the classification accuracies for each of the trials were noted and their means were as depicted below in Table 1. The graphs of the mean accuracies relative to the percentage of total training data available were also plotted as shown in Figures 5, 6, 7 and 8.

Percentage of Data Used	SVM	CNN-1	CNN-2	MLP
0.01	0.1966	0.2047	0.1785	0.1539
0.1	0.2672	0.3082	0.2114	0.2602
1	0.6481	0.7171	0.6994	0.5324
5	0.7552	0.8453	0.8204	0.6292
10	0.8108	0.8851	0.8833	0.6205
20	0.8478	0.9154	0.8966	0.6671
30	0.8651	0.9342	0.9235	0.7059
40	0.8762	0.9424	0.9310	0.7208
50	0.8849	0.9457	0.9386	0.7357
60	0.8924	0.9529	0.9345	0.7442
70	0.8973	0.9546	0.9406	0.7442
80	0.9014	0.9580	0.9382	0.7509
90	0.9045	0.9551	0.9464	0.7463
100	0.9070	0.9590	0.9460	0.7528

Table 1: Mean Classification Accuracies for the algorithms



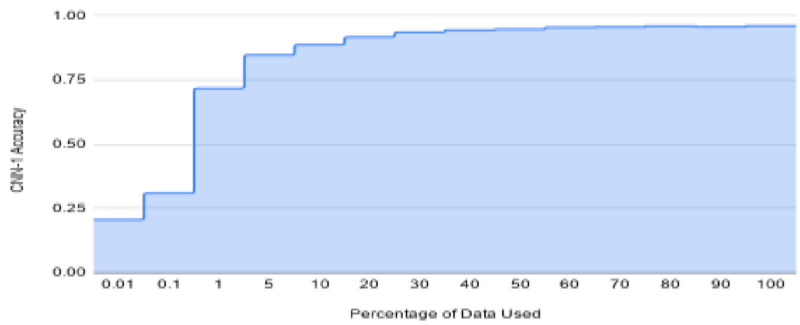


Figure 5: Mean SVM Accuracies

Figure 6: Mean CNN-1 Accuracies

Figure 7: Mean CNN-2 Accuracies

Figure 8: Mean MLP Accuracies

The combined mean accuracies for all the algorithms were plotted in a line chart as depicted below in Figure 9 for visual comparison.

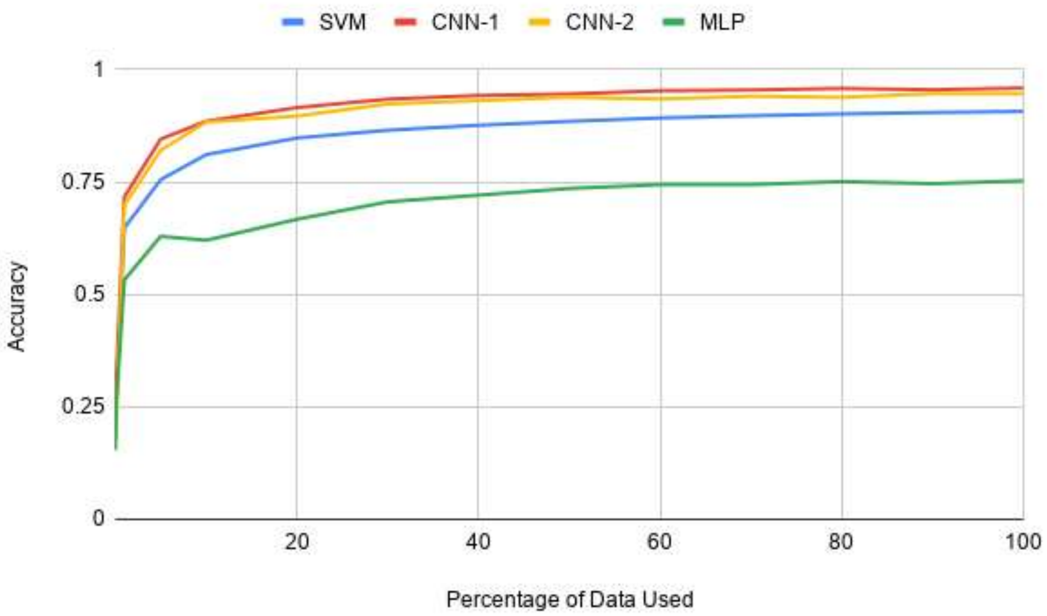


Figure 9: Combined graph of all models

As we can see from the data in Table 1 and Figure 9, the CNN-1 model outperformed every other model at every single data percentage. While the SVM and MLP showed better results than the CNN-2 when less than 1% of the total training data (<600 training samples) was available, the CNN-2 outperformed both of them for all data percentages >=1. Between the SVM and MLP, the SVM proved to be better for every single data percentage, and continued to scale to an accuracy >90%, but the MLP failed to achieve an accuracy within 10% of any of the others for data percentages 1 and higher.

Overall, from the results shown, it can be inferred that the CNN-1 model, which achieved an accuracy of 95.90% when the complete training data was available, is the best choice among the models discussed in this experiment for an image classification task similar to this one. However, depending on the dataset, the architecture of this model can be tweaked and fine-tuned further to achieve a higher accuracy, and for images of larger sizes, deeper Convolutional Neural Network architectures like the VGG-16 or ResNet can produce very high accuracies, but at the cost of a larger training time.

5. Conclusion

The Kuzushiji script holds high cultural value and developments in optical character recognition for the same furthers this cause. In using this KMNIST dataset, we experimentally compare the accuracies of 4 widely used classification algorithms, the SVM, MLP and 2 architectures of the CNN at different data percentage points ranging from 0.01% (6 training samples) to 100% (60000 training samples). The results showed that the CNN-1 architecture with 10 layers outperformed all other models at each data percentage point and in terms of the classification accuracy, is the best model of the four. While the 11 layer CNN-2 architecture and the SVM both achieved accuracies higher than 90%, the MLP proved to be unsuitable for this image classification task with a maximum accuracy of 75.28%. In future developments of this work, a stronger focus on the lower end of the data percentage could yield new insights. State-of-the-art deep neural networks could also be compared with both each other and simpler ones such as the CNN-1 discussed in this paper, to determine the amount of data required for a very deep neural network to produce promising results.

References

[1] Ghosh, A., Mukherjee, A., & Ghosh, C. (2019, September). Simplistic Deep Learning for Japanese Handwritten Digit Recognition. In *International Conference on Innovation in Modern Science and Technology* (pp. 87-93). Springer, Cham.

[2] Alcantara, G. (2017). Empirical analysis of non-linear activation functions for Deep Neural Networks in classification tasks. *arXiv preprint arXiv:1710.11272*.

[3] Lamb, A., Clanuwat, T., & Kitamoto, A. (2020). KuroNet: Regularized Residual U-Nets for End-to-End Kuzushiji Character Recognition. *SN Computer Science*, *1*(3), 1-15.

[4] Learning, S. M. (2007). A Review of Classification Techniques, SB Kotsiantis. *Informatica*, *31*, 249-268.

[5] Keysers, D. (2007). Comparison and combination of state-of-the-art techniques for handwritten character recognition: topping the mnist benchmark. *arXiv preprint arXiv:0710.2231*.

[6] Meshkini, K., Platos, J., & Ghassemain, H. (2019, December). An Analysis of Convolutional Neural Network for Fashion Images Classification (Fashion-MNIST). In *International Conference on Intelligent Information Technologies for Industry* (pp. 85-95). Springer, Cham.

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

[8] Chollet, F. (2018). Keras; 2015. *XH Lu et al./Application of Machine Learning and Grocery Transaction Data*, *252*.

[9] Bisong, E. (2019). Google Colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform* (pp. 59-64). Apress, Berkeley, CA.

[10] Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., & Ha, D. (2018). Deep learning for classical Japanese literature. *arXiv preprint arXiv:1812.01718*.