UNIVERSITY OF BRADFORD

MACHINE LEARNING

COS6026-B

Optical Character Recognition

Author
Bence HAROMI

Module leader
Prof. Rami QAHWAJI

May, 2021

Multiple convolutional neural network models with different architectures are presented in this paper, trained and evaluated in the MNIST dataset. The goal of the study is to compare the accuracies achieved by the different models. The paper confirmed that a dropout layer not only reduces overfitting but can also increase accuracy, and it also covered how well the estimation is, which was given by the 5-fold cross-validation.

Contents

List of Figures

List of Tables

1 Introduction

Optical character recognition (OCR) is a form of data entry used for digitising documents. It is widely used throughout the world. For example, the usage at post offices, academic institutions, bank statements and passport documents. Due to its wide usage, there are already many different machine learning models for this task. The following essay investigates the accuracy of several convolutional neural network (CNN) models with simple architectures.

2 Background

2.1 Convolutional Neural Networks

Convolutional neural networks are often used for image pattern recognition tasks because of their simplicity and precision (O'Shea & Nash 2015). Although this paper's goal is not to explain the fundamentals of CNN-s, a little background knowledge is required to understand the later discussed architectures of the models.

A CNN receives a raw image as input which is usually given to a convolutional layer. Convolutional layers help reduce the models' complexity because instead of using a neuron for each pixel, they use kernels that require a significantly lower number of neurons. The kernels are used to only look at a small part of the image and find connections there. The convolutional layer produces a so-called activation map. This map is added to a pooling layer which helps to reduce the map's dimensionality and complexity. The pooling layer can reduce the map to even a quarter of its original size.

A CNN can be so accurate that it could achieve near-human performance on the MNIST dataset and on a traffic sign recognition benchmark it outperformed humans. (Ciresan et al. 2012).

3 Methodology and Data

3.1 Data

The MNIST dataset is a database of handwritten digits with 70000 examples. The dataset is divided into two parts, training and test datasets. The training data contains 60000 and the test data 10000 samples. The samples are labelled images with 28x28 resolution. Each pixel with 0-255 grey levels where 0 (white) is the background and 255 (black) is the foreground. The label is an integer between 0 and 9, which is the digit shown on the image.

3.2 Preprocessing

Each class (digit) has around 6000 samples; the dataset is balanced. The images grayscale is in the range 0 to 255; with normalization, it is transformed to the range 0.0 to 1.0. Each image has 28x28 pixels and only has one colour channel; thus, they have been reshaped to 28x28x1. After the normalization, no further preprocessing is needed because there are no duplicates, no missing values, and the data is balanced.

4 Analysis and Discussions

4.1 Experiments

For experimentation purposes, the performance of multiple models was compared.
At first, three models have been created with a different number of convolution layers. The first one (*Model A*) has one, the second (*Model B*) two and the third (*Model C*) three.
Model A has the most straightforward architecture. Its first layer is the convolutional layer with the input shape of the images (28x28x1) and 32 filters where the size of the kernel is 3x3. The next layer is a max pooling layer which simplifies the output after the convolution. It is followed by a flattening layer which is required by the last layer, the classifier. The classifier uses a softmax activation, which was chosen because it is often used for multi-class classification tasks.
Model B has two extra layers between the max-pooling and flattening layers, a second convolutional layer with 64 filters and another max-pooling layer.
The third model's architecture is identical to the previous one, with the only difference that it has an additional convolutional layer with the same parameters as the second one.
Because the first three models do not have any post-processing after the flattening, their identifier is the *PLAIN* word in the figures.
The second three models are modified versions of the previous three. For each one, before the classifier, a layer with 100 neurons has been added; thus, the identifier word is *HUNDRED*.
Looking at the training charts, the first version of the models (*PLAIN)* showed very little sign of overfitting, but the second versions with the 100 unit layer (*HUNDRED*) are more clearly overfitted. To avoid this unwanted activity, the third version of the models, instead of the 100

unit layer, have a dropout layer. The identifier of this layer is *DROPOUT*. The dropout layer randomly turns off the given percentage of the input units by setting their weights to 0, by this reducing overfitting.

4.2 Evaluation

Although the MNIST dataset has dedicated test data, for the training it has not been used.
The evaluation of the models is done by 5-fold cross-validation. The data is shuffled and split into five groups. For five iterations, in each iteration, one of the groups has been used for test data and the other groups as training data which means that 20 percent of the data is used for testing. At the end of each iteration, the scores are measured. After all of the iterations, the means of the scores are calculated; these are the final scores. All of the samples were used either as part of test or train data. Using this technique, a more accurate estimate can be measured than just a simple train-test split measurement.
The accuracies of the cross-validations can be found below, in Table 1.

model	PLAIN	HUNDRED	DROPOUT
Model A	98.10%	98.48%	98.02%
Model B	98.73%	98.78%	98.91%
Model C	98.90%	98.86%	99.04%

Table 1: 5-fold cross-validation mean accuracies

After the training, a blind evaluation has been done. As mentioned before, the dataset's original test data has not been used during the cross-validation by the models, neither for training nor for evaluation. To see how well the models would work on never seen data, in this blind evaluation, the original test data (10000 samples) has been used for evaluation and the whole training data (60000 samples) for the training of the models.
The results of the blind evaluation can be found below, in Table 2.

model	PLAIN	HUNDRED	DROPOUT
Model A	98.15%	98.59%	98.19%
Model B	98.75%	98.88%	98.96%
Model C	99.22%	98.98%	99.32%

Table 2: Blind evaluation accuracies with the original test data

A summary table can be found below to compare the estimated and the actual blind evaluation accuracies (Table 3).

On the following pages the accuracy versus number of epochs and loss versus number of epochs plots of the models can be found. In all 4 figures the blue and red lines represent the scores achieved during the training and validation respectively. (Figures 1-4). Additionally,

model	5-fold	blind
Model A PLAIN	98.10%	98.15%
Model A HUNDRED	98.48%	98.59%
Model A DROPOUT	98.02%	98.19%
Model B PLAIN	98.73%	98.75%
Model B HUNDRED	98.78%	98.88%
Model B DROPOUT	98.91%	98.96%
Model C PLAIN	98.90%	99.22%
Model C HUNDRED	98.86%	98.98%
Model C DROPOUT	99.04%	99.32%

Table 3: Comparison of the 5-fold and the blind accuracies

in each figure, the first columns represent the PLAIN models, second column the HUNDRED models and the last column the DROPOUT models. Moreover the first row corresponds to *Model A*, second row to *Model B* and the last row to *Model C*.

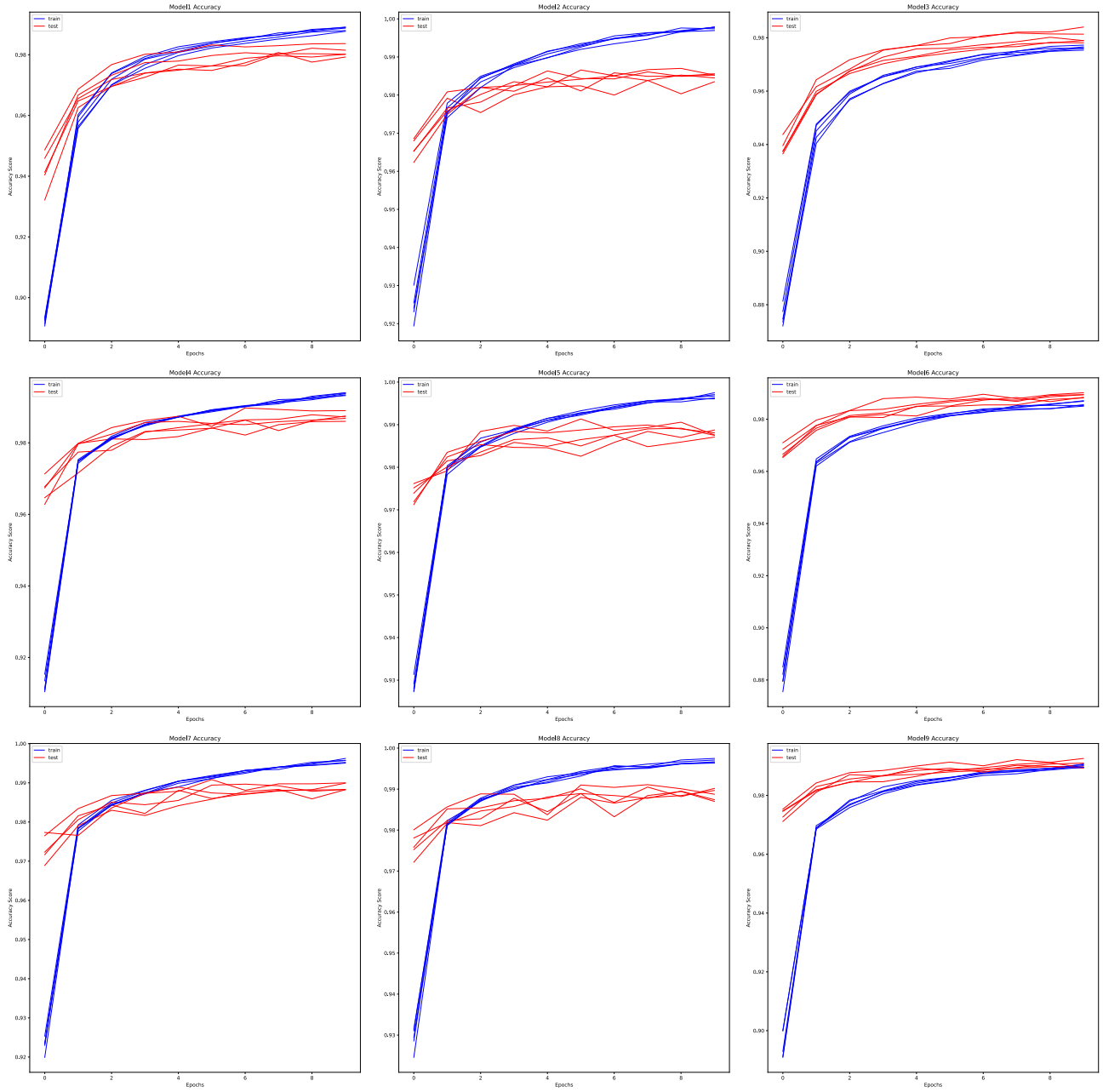

Figure 1: The 5-fold cross validation training's accuracy plots

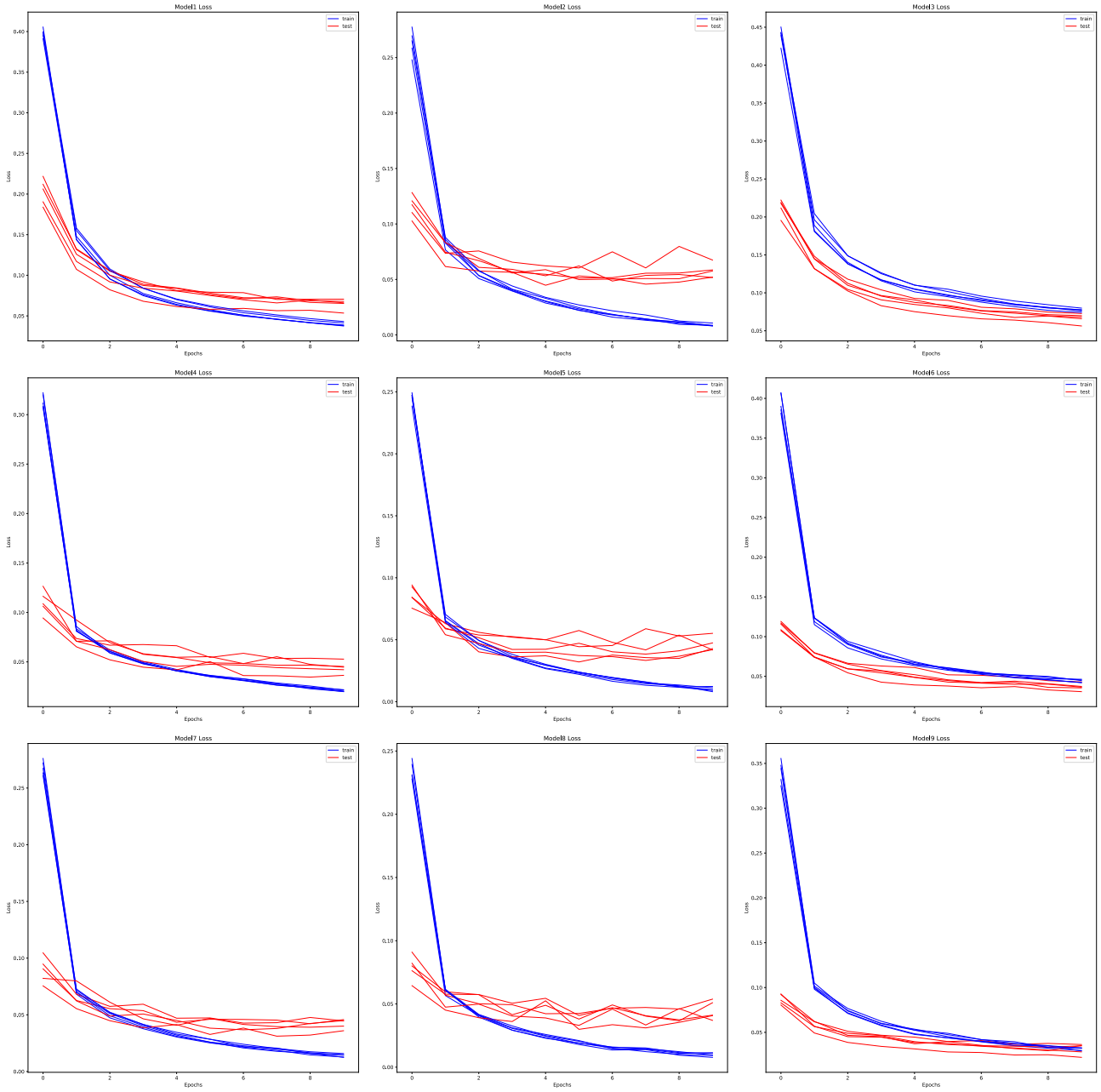

Figure 2: The 5-fold cross validation training's loss plots

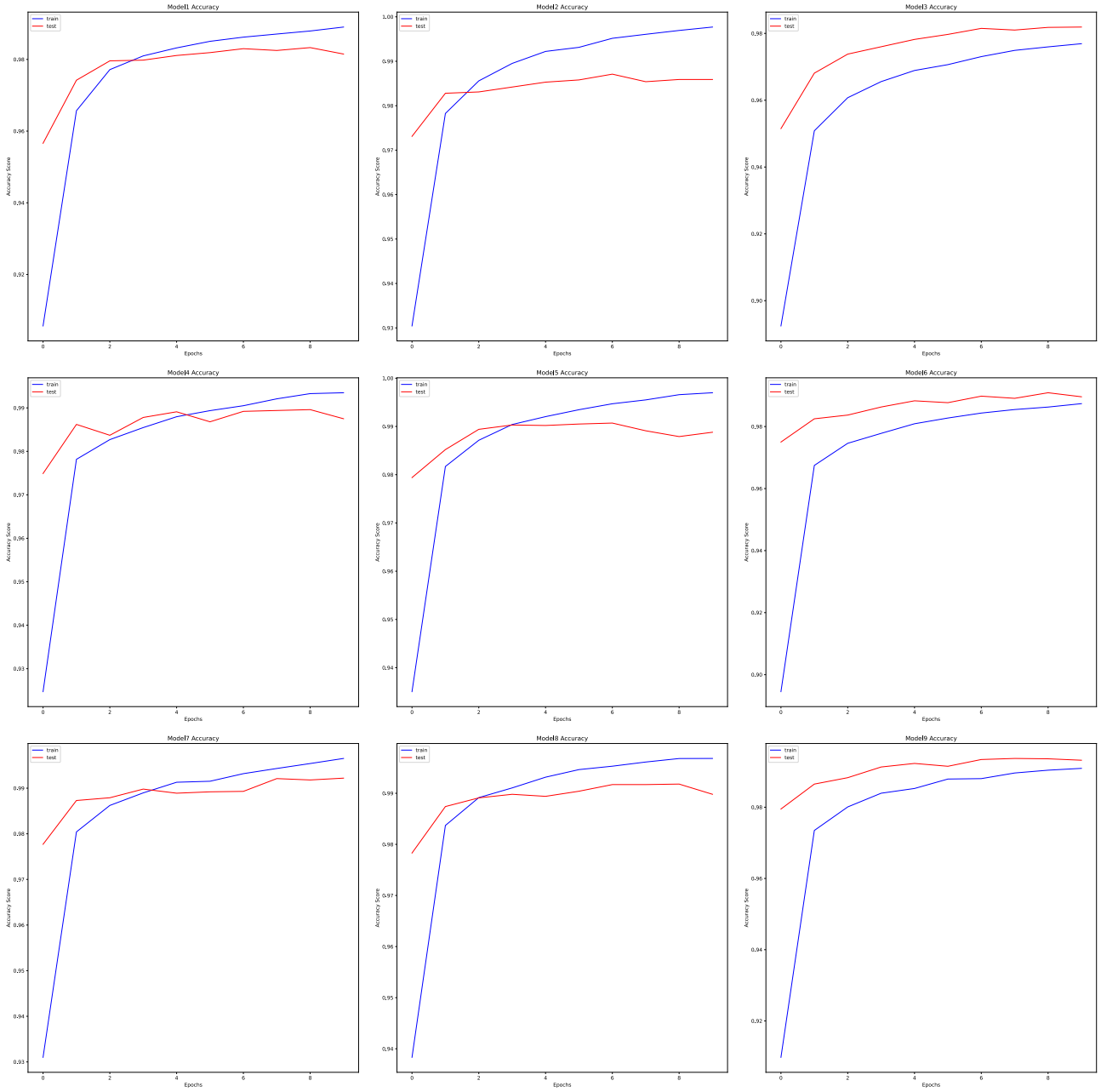

Figure 3: The blind validation training's accuracy plots

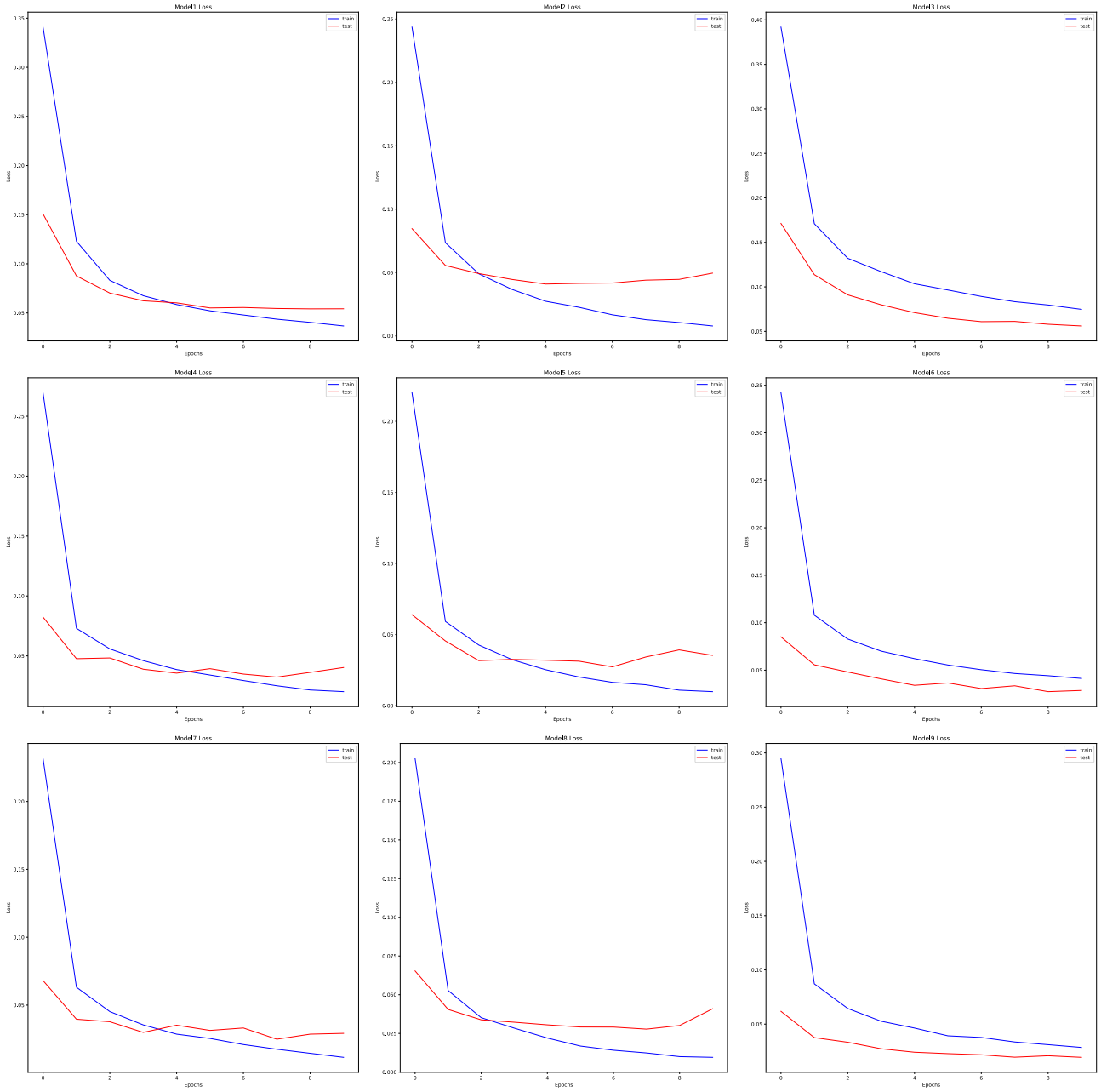

Figure 4: The blind validation training's loss plots

5 Conclusions and suggestions for future work

In this paper, multiple CNN models with different architectures have been presented, including a model which achieved 99.04% accuracy on the MNIST dataset. It was also shown that the 5-fold cross-validation gives an excellent estimate of the accuracy of the models. From the comparison, it turned out that using multiple layers of convolutional layers help to achieve more accurate models. With a dropout layer, not only overfitting can be reduced, but also accuracy can be increased.

For future work, the comparison could be extended by adding more models to it. Further experimentation could be done with other CNN architectures but also with different types of machine learning models.

References

Ciresan, D. C., Meier, U. & Schmidhuber, J. (2012), 'Multi-column deep neural networks for image classification', *CoRR* **abs/1202.2745**.
URL: *http://arxiv.org/abs/1202.2745*

O'Shea, K. & Nash, R. (2015), 'An introduction to convolutional neural networks', *CoRR* **abs/1511.08458**.
URL: *http://arxiv.org/abs/1511.08458*