UCL CEGE0004 GROUP CHAOS PRESENTATION March 2024

Marc Johnston, Yuri Lai, Thitiwich Thummakul, Siwei Lu

25 years on: Is LeNet-5 still relevant?

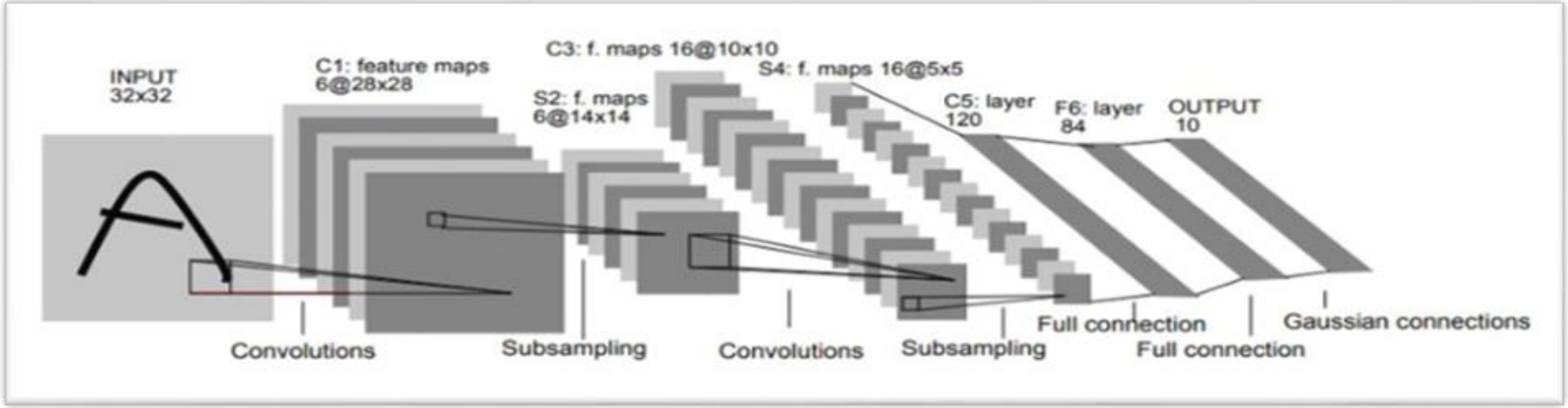
The assignment paper says to “Start with a title that captures the essence of your project” - this is only a suggestion!

Marc Johnston, Yuri Lai, Thitiwich Thummakul, Siwei Lu

Background

Yann LeCun, author of the paper and the father of modern AI

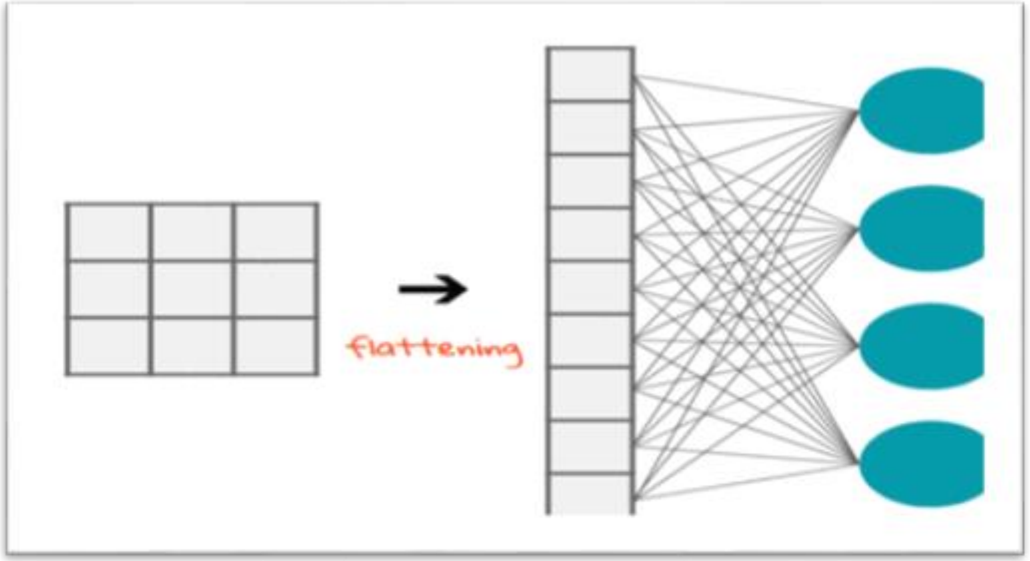
- Connectionist revival after the second "AI Winter "
- Automatic ML > Heuristics-based algorithms with hardwired logics
- A good balance between performance and computational efficiency
- Foundational to the modern AI boom



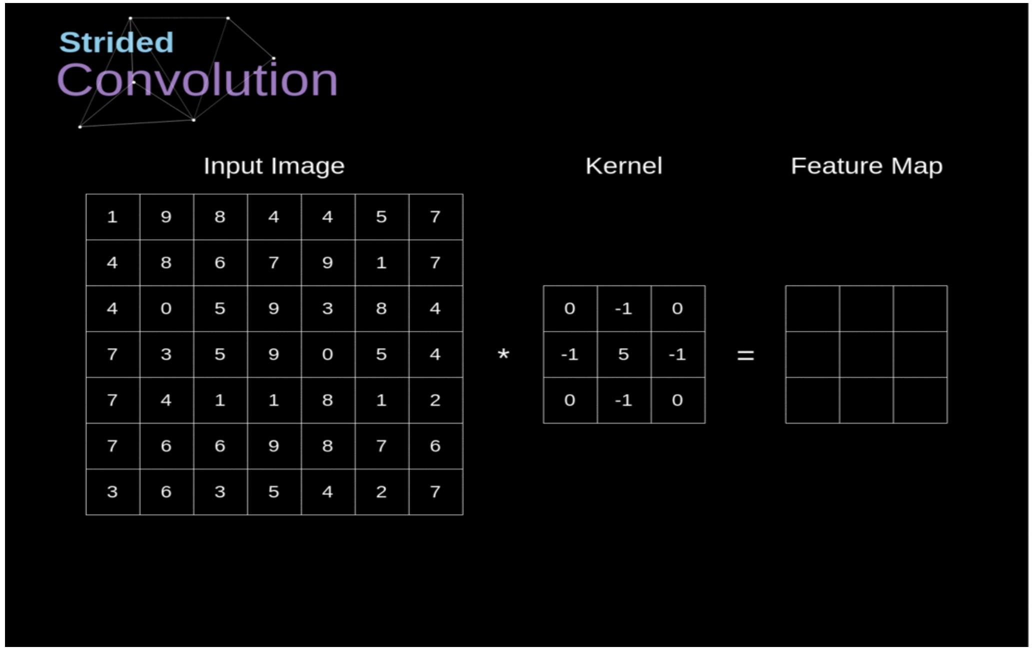
LeNet-5 model architecture

Theory

- Weight sharing means more efficient than fully connected NNs.
- Takes the context and correlations inside an image itself into consideration. Inherently 2D.
- More tolerant to variabilities in the data than fully connected NNs.



2-layer FC network, 3x3 input (Up) One convolution layer (bottom))



Objectives

- Reproduce the results from Yann LeCun's paper "Gradient-Based Learning Applied to Document Recognition"
- Apply the model to two additional datasets to examine the performance of the model in both datasets
- Doing hyperparameter tuning to enhance the performance of the model

Dataset

MNIST Database

- A collection of handwritten digits from 0 to 9
- Contains 60,000 training images and 10,000 test images of handwritten digits
- Image format: 28x28 pixels
- Serves as a benchmark dataset for evaluating the performance of various machine learning algorithms, particularly for image classification tasks

Sample images from MNIST dataset

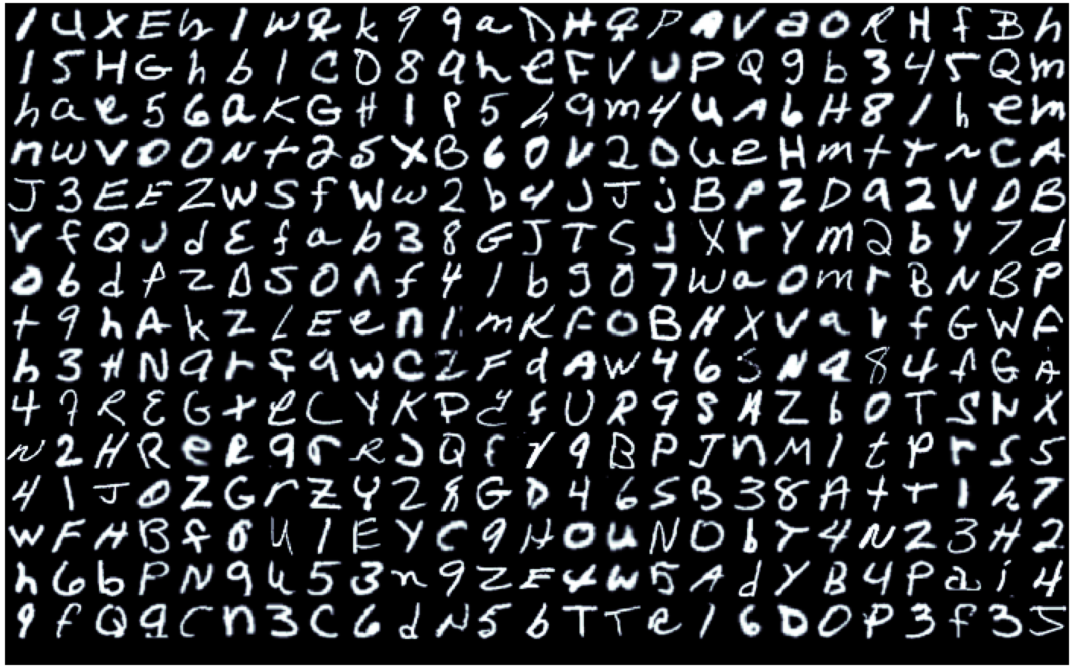
EMNIST Database

- An extension of the MNIST dataset that includes handwritten characters from both digits (0-9) and uppercase and lowercase letters (A-Z, a-z)

- Contains approximately 2,255,710 characters in total, with 6 different categories of variant

- Image format: 28x28 pixels

- Increases the diversity of data for testing versatility of the algorithm



Sample images from Extended MNIST dataset

FMNIST Database

- a dataset containing grayscale images of various clothing items, such as T-shirts, trousers, dresses, and shoes.
- consists of 60,000 training images and 10,000 test images
- Image format: 28x28 pixels
- One of a contemporary challenging dataset in modern days

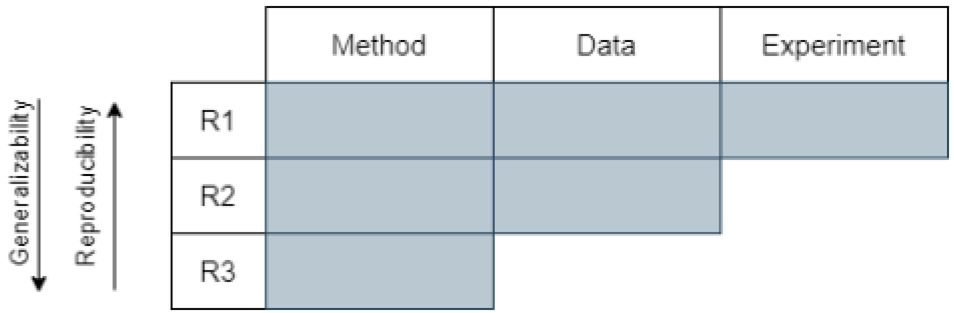
Sample images from Fashion MNIST dataset

Reproducibility

- Paper reproduced at ‘R2’ Level

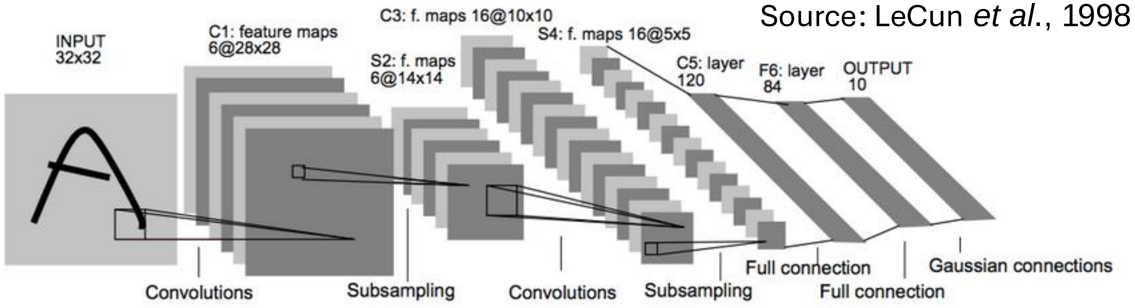
- Aim to reproduce the results using the same data
  - Many hyperparameters listed in original paper

- R1 method not possible as exact implementation details not available
  - i.e., software versions, code



Source: Semmelrock *et al.*, 2023

Methodology - steps



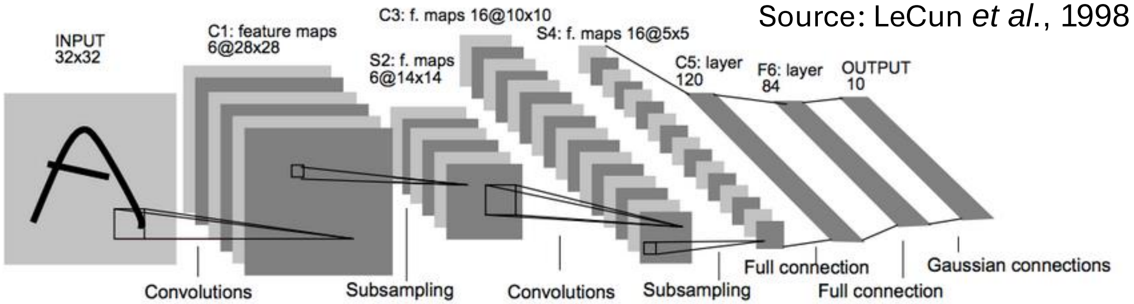
Source: LeCun *et al.*, 1998

- Original dataset still available
  - Separate files for train/test split

- Detailed structure of LeNet-5 described in original paper
  - Includes number and types of layers, input image recognition and hyperparameters

- Keras chosen as replication tool
  - Allows developers to build (and fine-tune) all LeNet-5 layers using simple built-in functions

Algorithm implementation



Source: LeCun *et al.*, 1998

LeNet-5 Architecture (from paper [1])	Functionality Implemented/Parameters
20 training set iterations	Keras set to run for 20 epochs
Convolution layer 1 (of 3) – **C1** • 6 feature maps (28x28 in size) • 5x5 kernel	Keras Conv2D Layer • Filters=6 • Kernal size=5x5
Subsampling Layer 1 (of 2) – **S2** • 2x2 kernel • Non overlapping	Keras AveragePooling2D Layer • pool_size=(2, 2) • strides=(2, 2)
Convolution layer 2 (of 3) – **C3** • 16 feature maps (28x28 in size) • 5x5 kernel	Keras Conv2D Layer • Filters=16 • Kernal size=5x5
Subsampling Layer 2 (of 2) – **S4** • 6 feature maps (5x5 in size) • 2x2 kernel	Keras AveragePooling2D Layer • pool_size=(2, 2) • strides=(2, 2)
Convolution layer 3 (of 3) – **C5** • 120 feature maps (1x1 in size)	Keras Dense Layer • units=120
Fully-Connected Layer – **F6** • 84 units	Keras Dense Layer • units=84
Output Layer – **OUTPUT** • 10	Keras Dense Layer • units=10
Layers C1 -> F6 • Sigmoid squashing function (hyperbolic tangent)	`def custom_activation(x):` `    return (K.tanh(2/3 * x) * 1.7159)`

Learning rate scheduler
Global learning rate was scheduled as:
• 0.0005 for the first two passes
• 0.0002 for the next three
• 0.0001 for the next three
• 0.00005 for the next 4
• 0.00001 thereafter.
(20 training epochs in total)
	`def lr_schedule(epoch, lr):`
	`    if epoch < 2:`
	`        return 0.0005`
	`    elif epoch < 5:`
	`        return 0.0002`
	`    elif epoch < 8:`
	`        return 0.0001`
	`    elif epoch < 12:`
	`        return 0.00005`
	`    else:`
	`        return 0.00001`

Methodology - challenges

- Original implementation code not available
  - Need to create own as close to original as possible
- Original hardware not available (Single 200MHz processor!)
- Input image size for LeNet-5 is 32x32
  - MNIST image size 28x28. Padding added so that the size matched

Result on MNIST

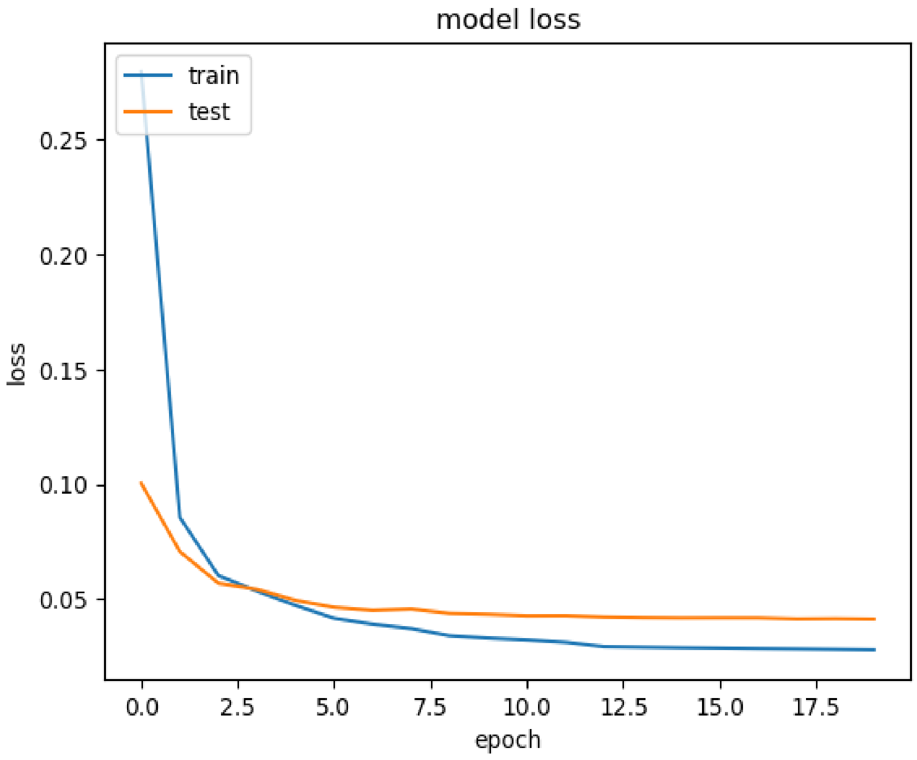


Figure.1. Training result on MNIST (20 epochs)

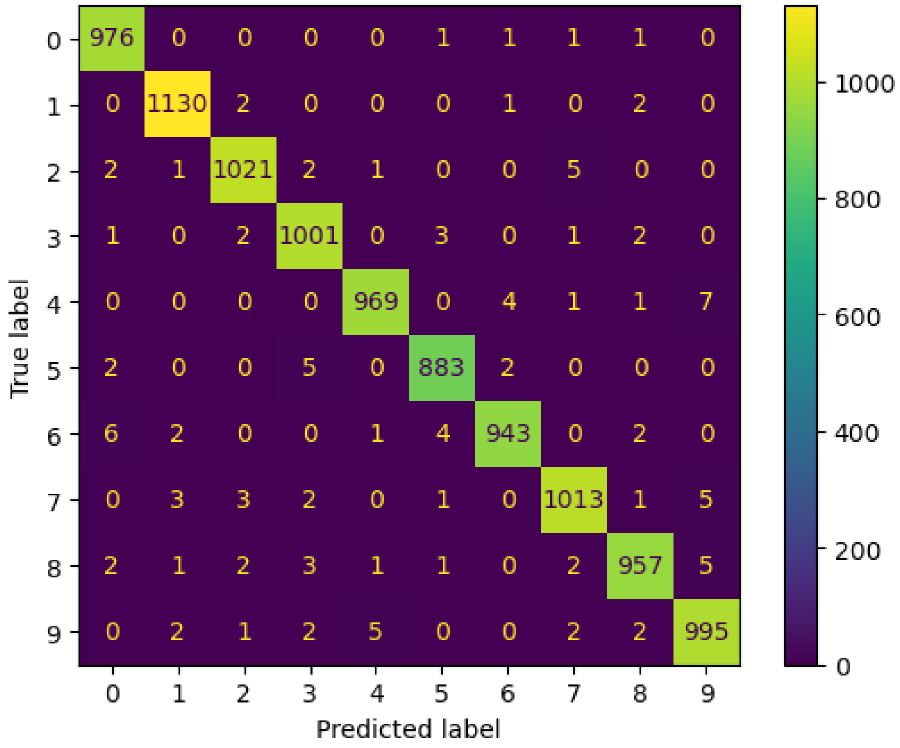


Figure.2. Confusion matrix on MNIST

Metrics	Value
Loss	0.0354
Accuracy	98.88%
Precision	98.88%
Recall	98.88%
F1	98.88%

Compared with the original paper

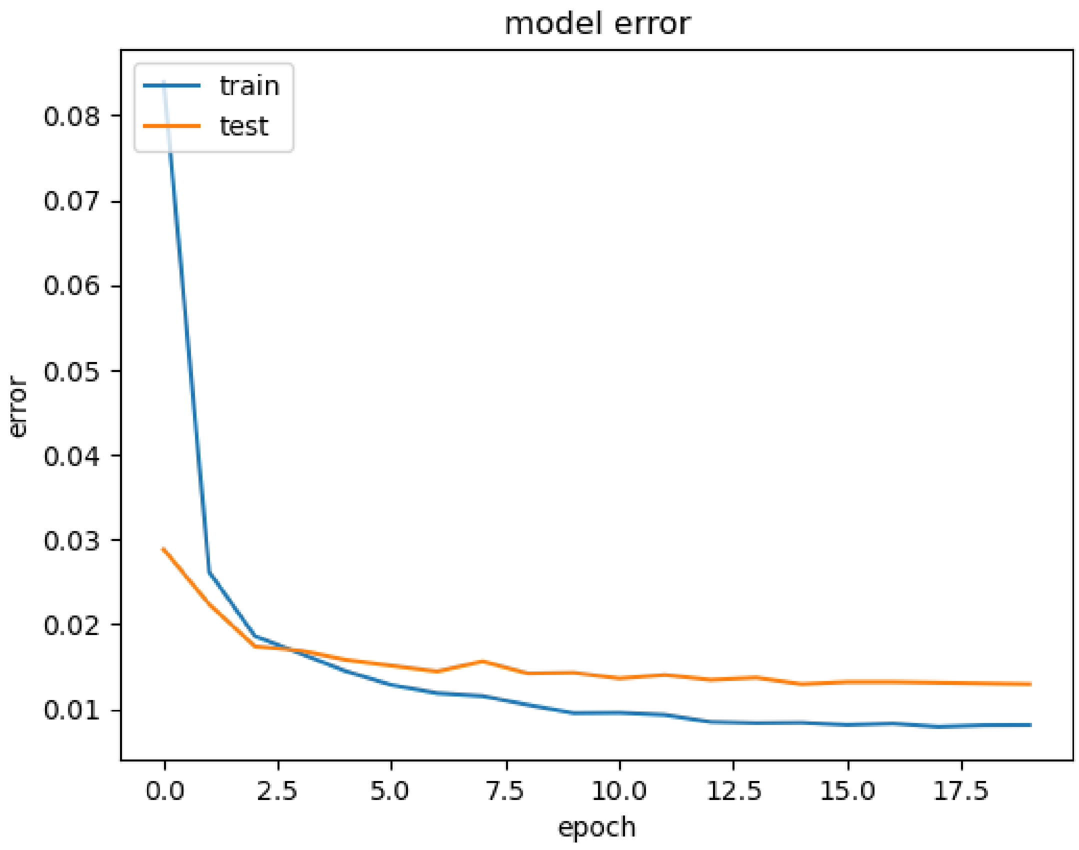


Figure.3. Training and test error from reproduced model

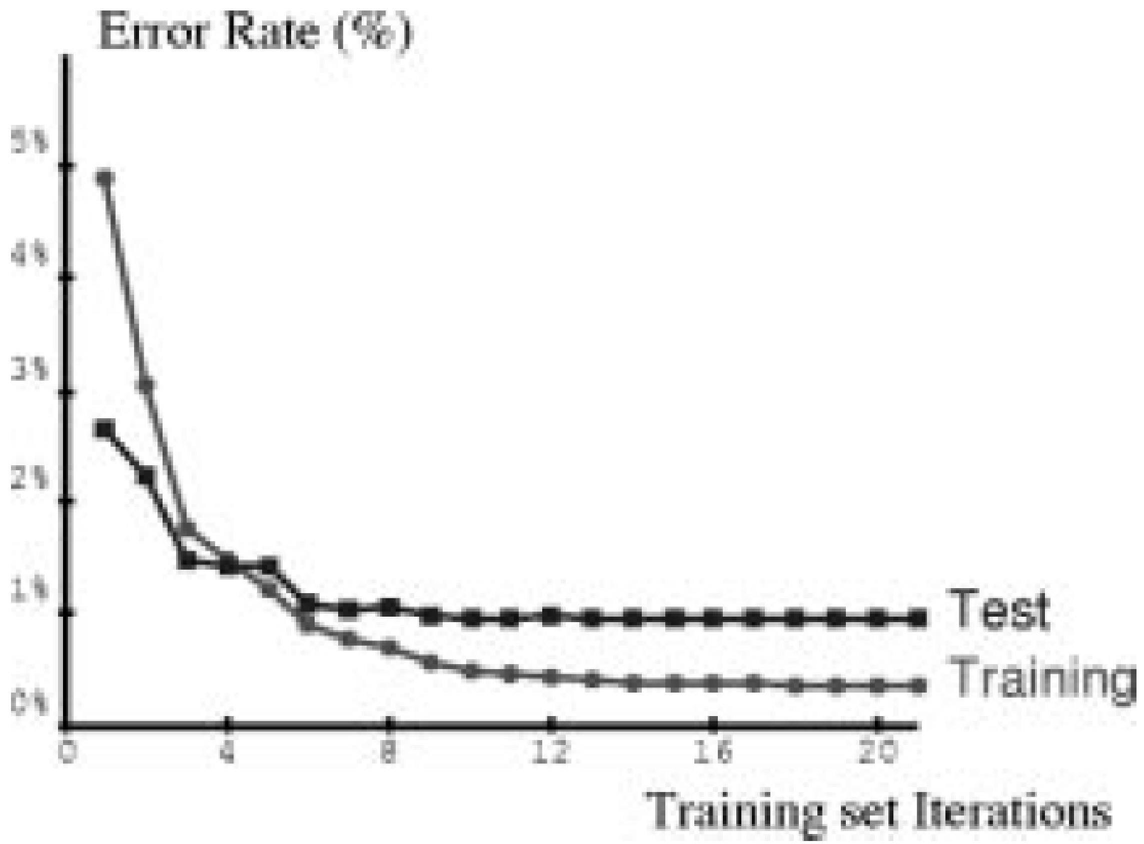


Figure.4. Training and test error from original paper

Miss-classified examples

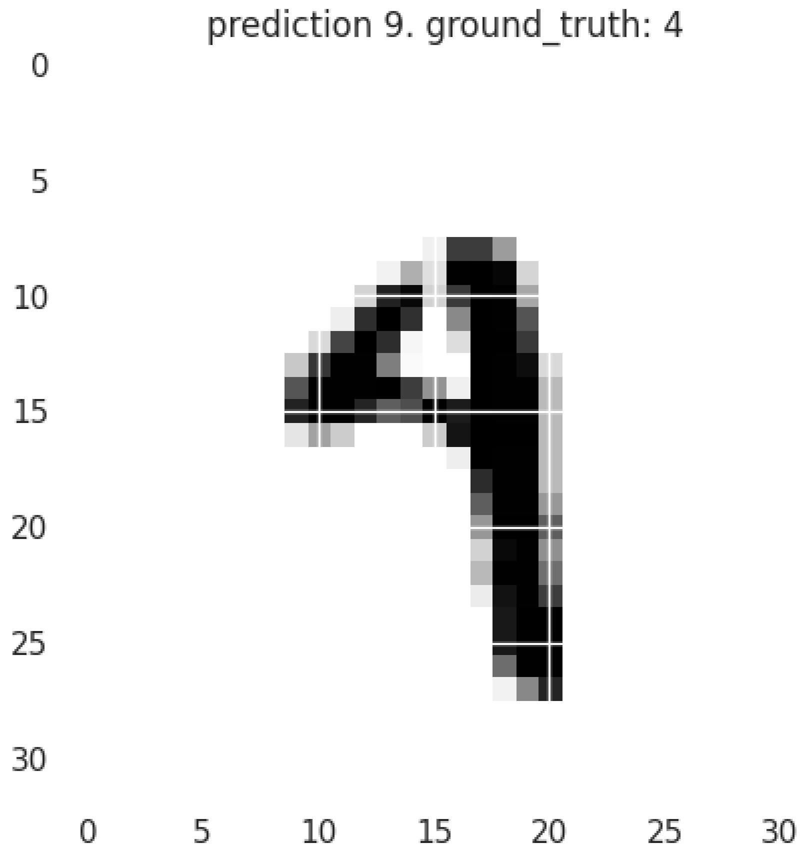


Figure.5. miss-classified example one

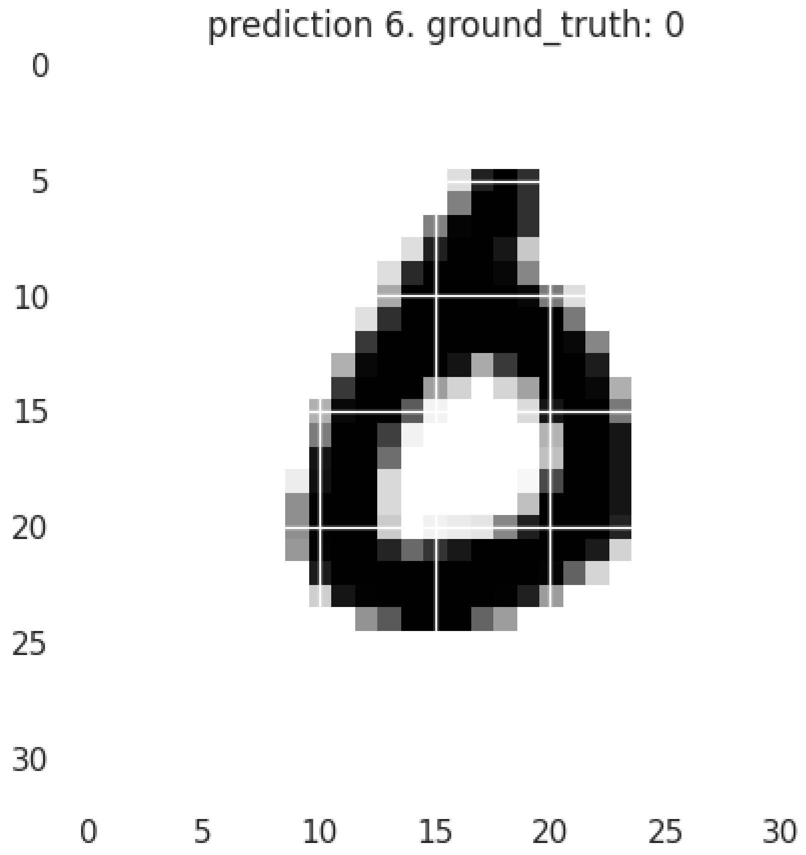


Figure.6. miss-classified example two

Tuning Hyper-parameter

Figure. 7. MNIST model accuracy vs learning rate

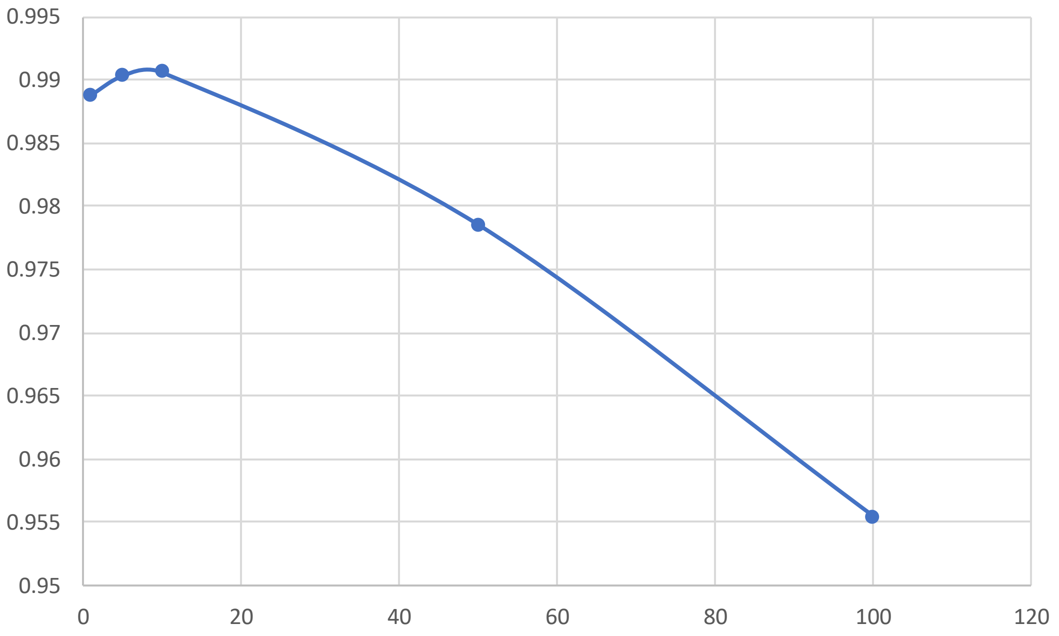
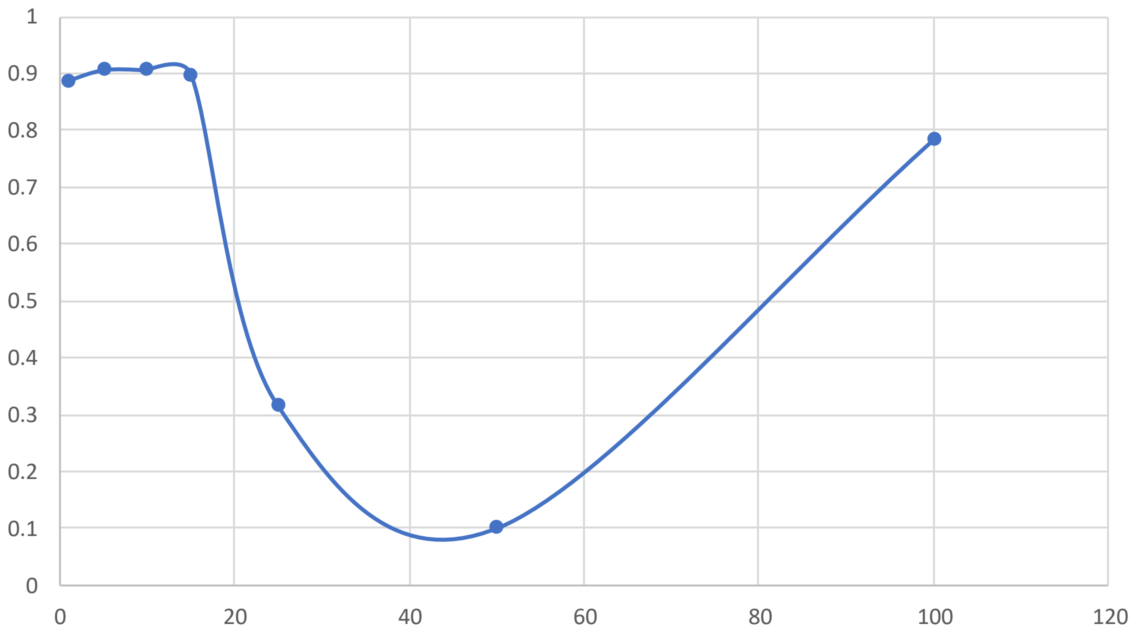


Figure. 8. FMNIST model accuracy vs learning rate



Tuning Hyper-parameter

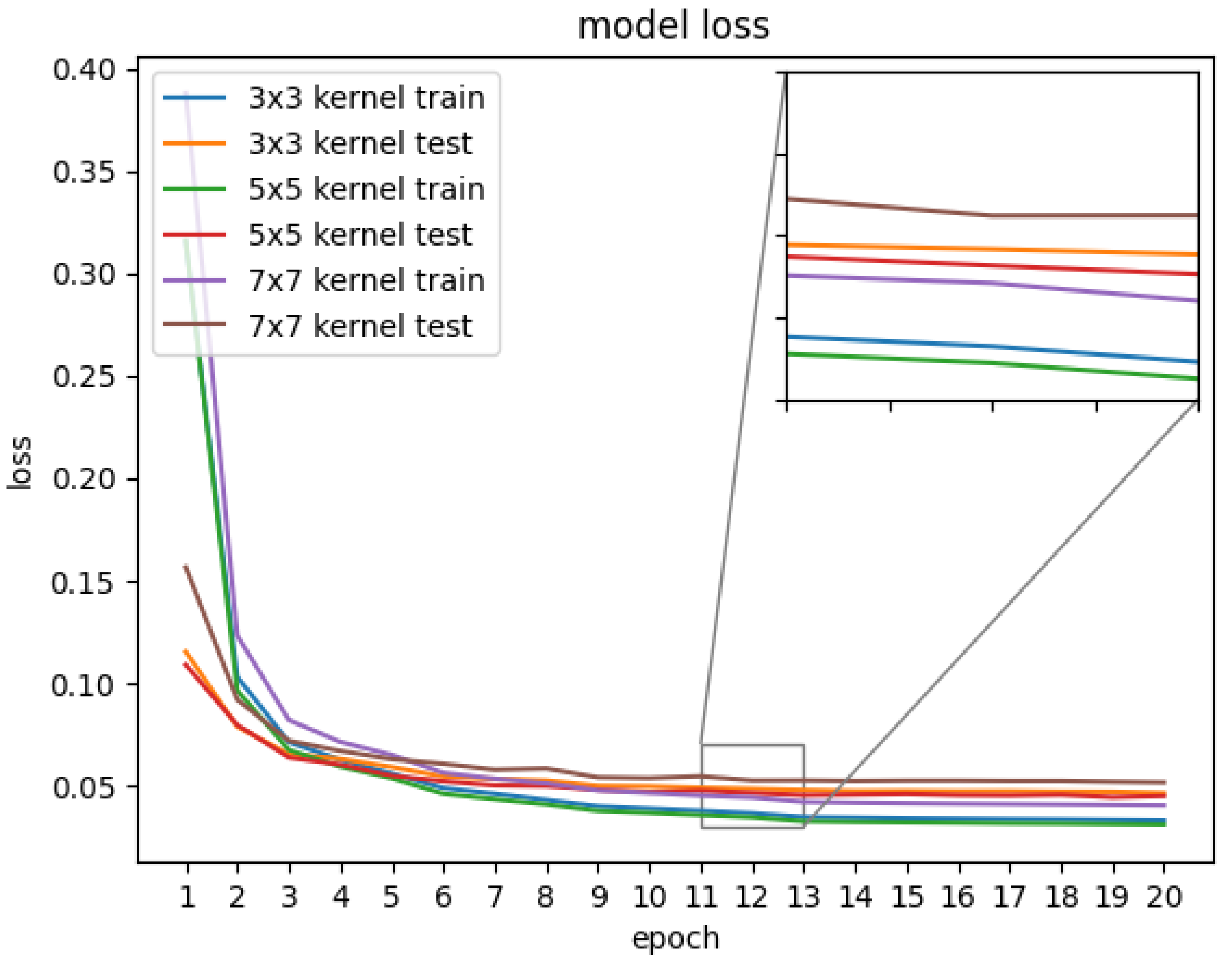


Figure.9. Model loss trained on MNIST with varying kernel sizes.

Result on FMNIST

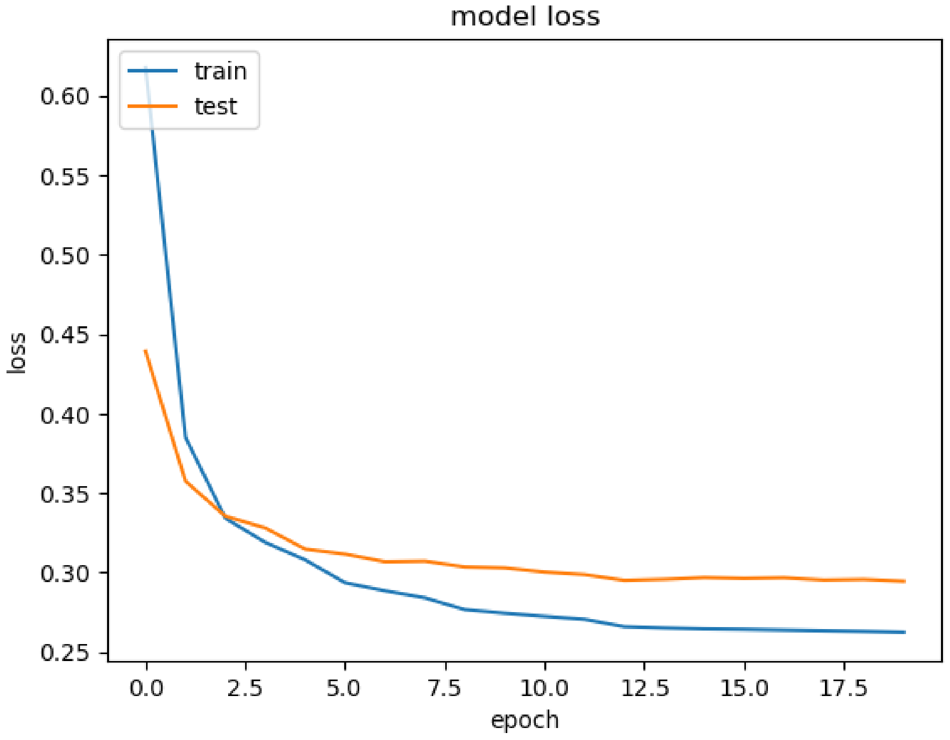


Figure.10. Training result on FMNIST (20 epochs)

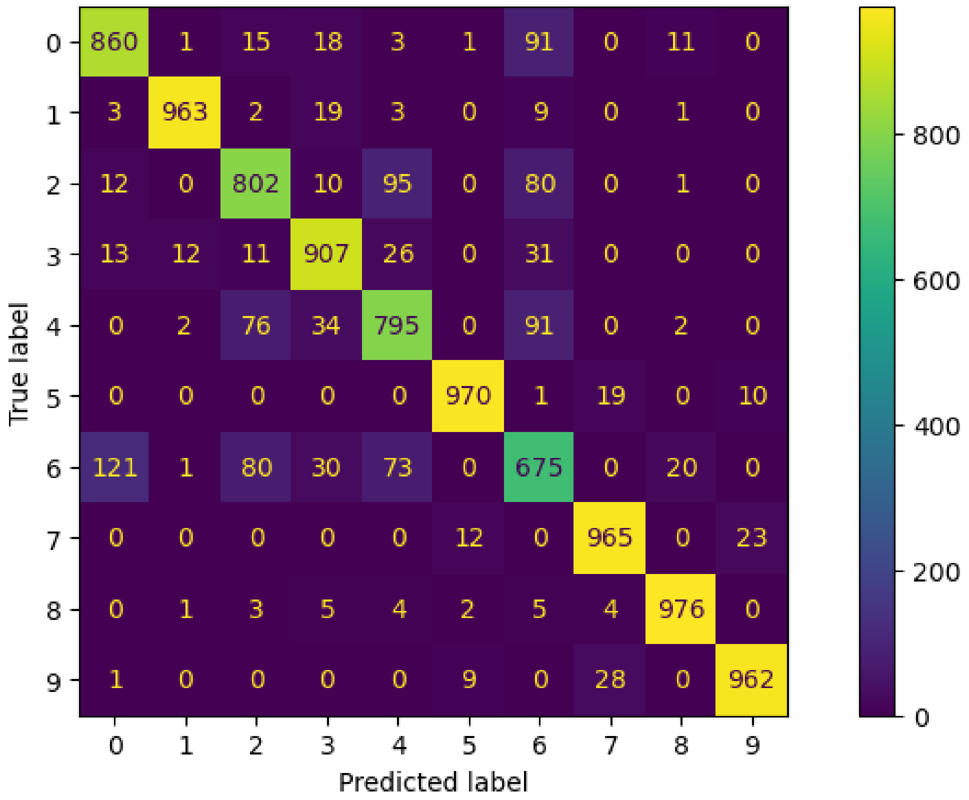


Figure.11. Confusion matrix on FMNIST

	Label		**Label**
0	T-shirt/top	5	Sandal
1	Trouser	6	Shirt
2	Pullover	7	Sneaker
3	Dress	8	Bag
4	Coat	9	Ankle boot

Metrics	**Value**
Loss	0.3093
Accuracy	88.75%
Precision	88.75%
Recall	88.75%
F1	88.75%

Result on EMNIST

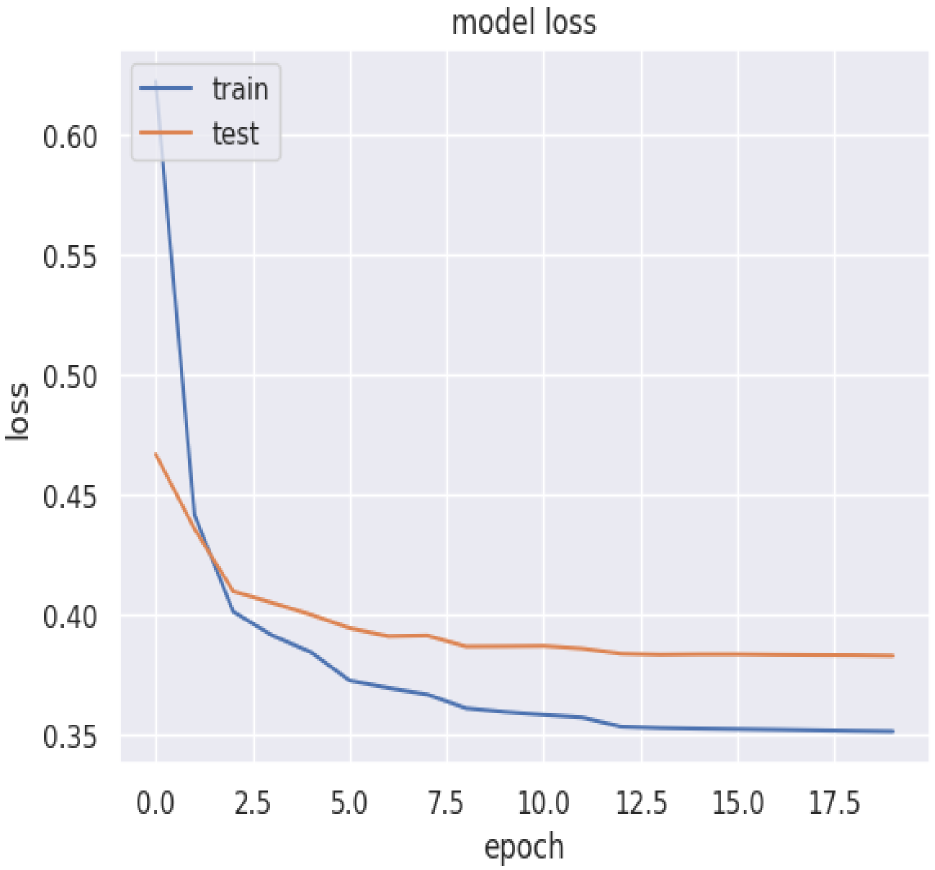


Figure.12. Training result on EMNIST (20 epochs)

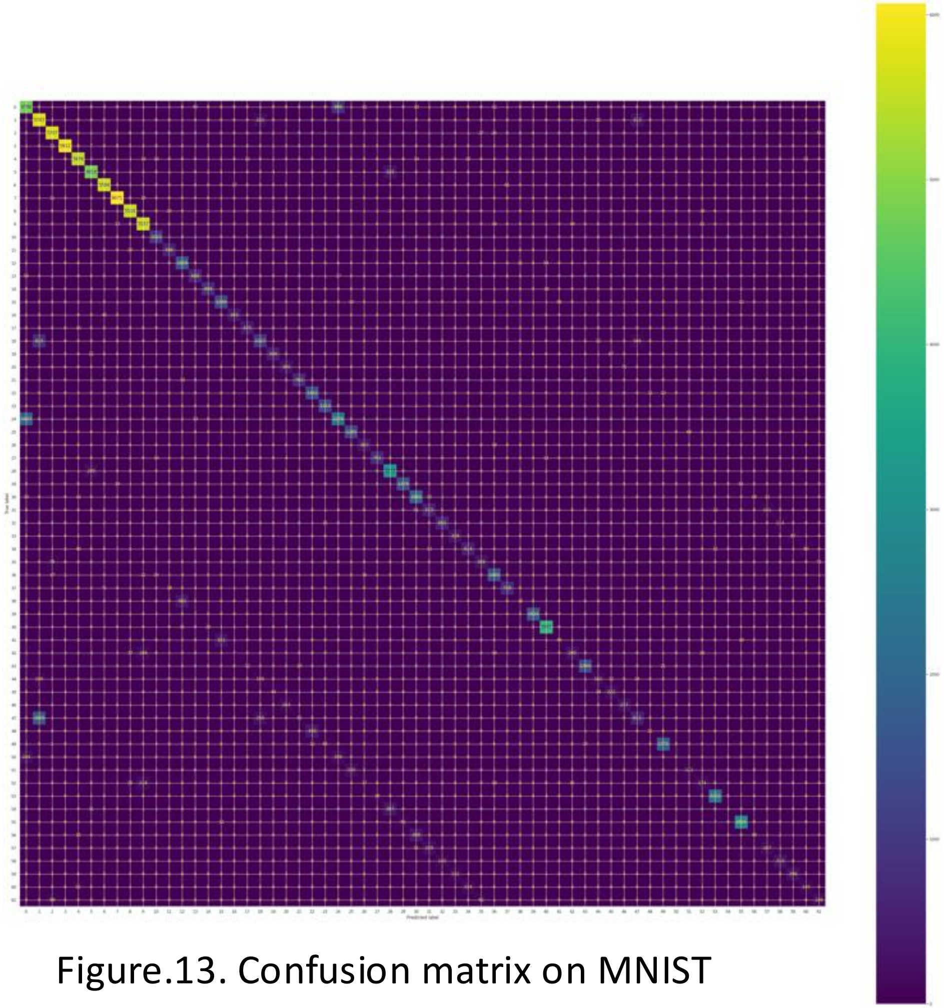


Figure.13. Confusion matrix on MNIST

Metrics	Value
Loss	0.3768
Accuracy	86.32%
Precision	86.32%
Recall	86.32%
F1	86.32%

Limitations

- Some group members had access to GPUs while others did not. This could lead to inconsistent results.
- Hyperparameters were tuned independently due to limited time and resources.
- We do not have the original codes from 1998. The modern machine learning/ data science ecosystem did not exist back then.

- The paper is highly reproducible despite original codes missing.
- We reproduced nearly identical accuracy, error, speed of convergence as the original paper.
- The model performed less well on extra datasets due to increased complexities. (Mnist:98% ,Fmnist: 88%, Emnist: 86%)
- Increase learning rate by tenfold has a general positive impact on model performance, and 5x5 kernel size is optimal.
- If we had more time and computational resources, we would have benchmarked the performance of LeNet-5 against other models and increased the variabilities in hyper parameter tuning. And running everything with a GPU.

References

- Yann LeCun, Leon Bottou, Yoshua Bengio and Patrick Haner. Gradient-Based Learning Applied to Document Recognition. 1998
- Harald Semmelrock, Simone Kopeinik, Dieter Theiler, Tony Ross-Hellauer, and Dominik Kowald. Reproducibility in machine learning-driven research. 2023.