BIRZEIT UNIVERSITY

Faculty of Engineering and Technology
Department of Electrical and Computer Engineering
ENCS5141—Intelligent Systems Laboratory

Assignment #2—Comparative Analysis of CNN and Patch-based LSTM Architectures for Image Classification

Prepared by: Lana Musaffer—1210455

Instructor: Yazan Abu Farha
Assistant: Ahmad Abbas
Date: June 1, 2025

Abstract

The aim of this study is to presents a comparative analysis of three deep learning architectures—CNN Baseline, AlexNet, and Patch-based LSTM—on MNIST and CIFAR-10 datasets. Metrics evaluated include classification accuracy, training time, inference time, and confusion matrices. Results show that AlexNet performs best on MNIST in accuracy, while Custom CNN shows better trade-offs on CIFAR-10. Patch-based LSTM is competitive in MNIST but shows slower performance in accuracy on CIFAR-10, though it remains efficient in time.

Contents

1 Introduction

1.1 Motivation

Image classification remains a core task in computer vision. This study aims to evaluate the performance of three distinct architectures: a simple CNN baseline, AlexNet, and Patch-based LSTM, across MNIST and CIFAR-10 datasets. The purpose is to understand their relative strengths in accuracy, training efficiency, and suitability for different types of image data.

1.2 Theoretical Background

1.2.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing grid-like data, such as images. They consist of multiple layers including convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply a set of learnable filters to extract local features, which are useful for recognizing edges, textures, and more complex shapes in deeper layers. Pooling layers reduce the spatial resolution of feature maps, thereby reducing computational cost and helping the model become invariant to small translations. CNNs are well-suited for image classification, object detection, and segmentation tasks due to their ability to capture spatial dependencies and hierarchical features [1].

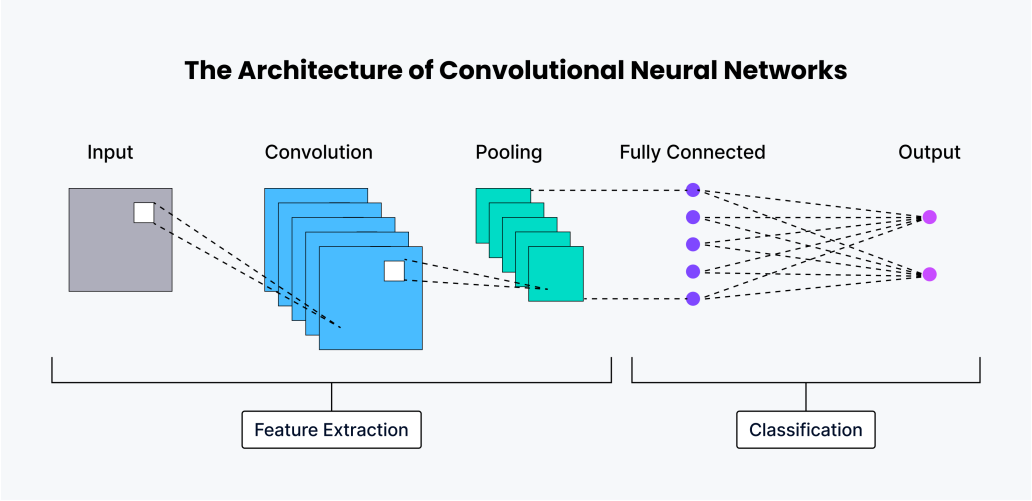

Figure 1.1: Typical CNN architecture. [2]

1.2.2 AlexNet

AlexNet is a pioneering deep CNN model that significantly advanced the field of computer vision by winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. It has eight layers: five convolutional layers followed by three fully connected layers. Innovations introduced by AlexNet include the use of Rectified Linear Units (ReLU) as activation functions, which allow faster convergence, and dropout, which reduces overfitting. The model also leverages GPU acceleration to train deeper networks more efficiently. Although originally trained on high-resolution color images, AlexNet can be adapted to smaller grayscale datasets by adjusting the input layer and number of output classes [3].

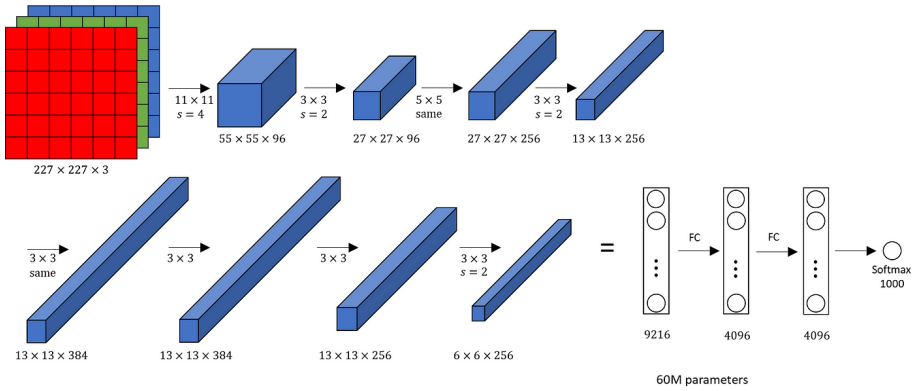

Figure 1.2: AlexNet architecture. [4]

1.2.3 Long Short-Term Memory (LSTM) Networks

LSTM networks are a specialized form of Recurrent Neural Networks (RNNs) designed to capture long-term dependencies in sequential data. They overcome the vanishing gradient problem of traditional RNNs through gated mechanisms, input, forget, and output gates which regulate the flow of information across time steps. While LSTMs are commonly used in natural language processing and time series forecasting, their sequential nature can be leveraged for visual data by converting spatial information into sequences. Patch-based LSTM models do this by dividing an image into non-overlapping patches and feeding them as a temporal sequence to the LSTM. This approach allows the model to learn relationships between distant spatial regions in an image, which is particularly beneficial for capturing global context [5, 6].

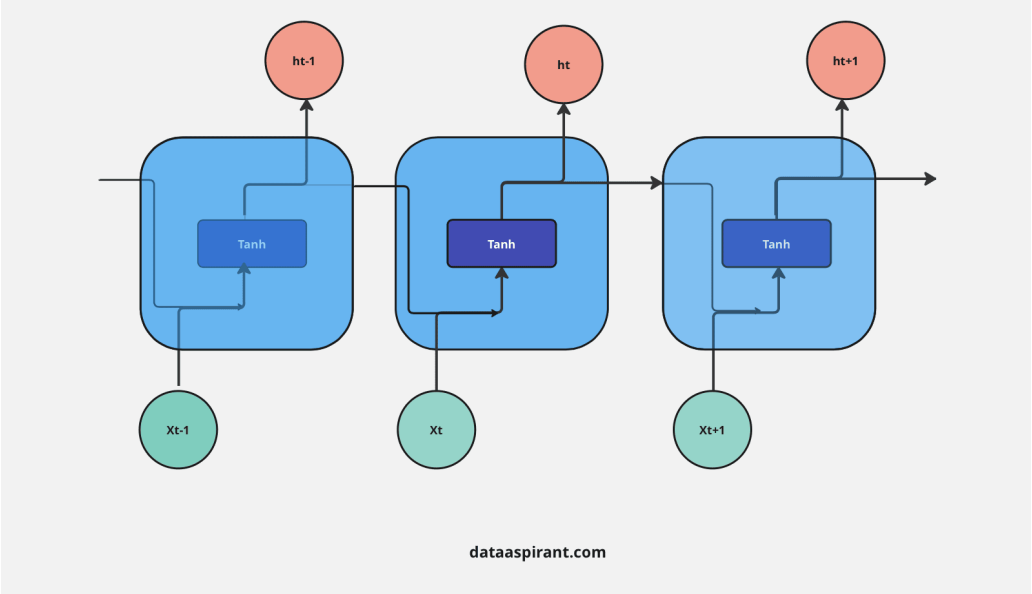

Figure 1.3: LSTM architecture. [7]

1.3 Objective

The objective of this experiment is to conduct a comparative analysis of CNN, AlexNet, and Patch-based LSTM models for image classification. The models will be trained and tested on MNIST and CIFAR-10 datasets. Performance will be evaluated based on multiple metrics including classification accuracy, training time, inference time, and confusion matrices. This study aims to provide insights into the practical advantages and limitations of each model architecture when applied to datasets of different complexity and visual characteristics.

2 Procedure and Discussion

2.1 Datasets Overview

2.1.1 MNIST

The MNIST dataset contains 60,000 grayscale images of handwritten digits from 0 to 9, each sized at 28×28 pixels. It is a widely used benchmark for image classification tasks due to its simplicity and clear digit boundaries.

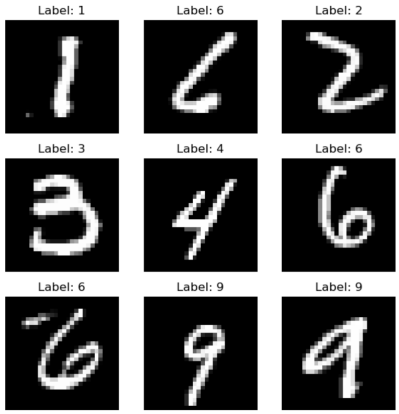

Figure 2.1: Sample images from the MNIST dataset.

2.1.2 CIFAR-10

CIFAR-10 consists of 60,000 colored images with 10 object classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck). Each image is of size 32×32 pixels and contains varying backgrounds and object shapes, making the dataset more complex.

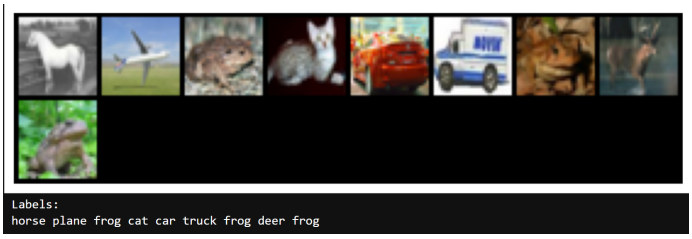

Figure 2.2: Sample images from the CIFAR-10 dataset.

2.2 Data Preprocessing

The data preprocessing phase included several important steps to prepare the MNIST and CIFAR-10 datasets for model training. These steps were tailored to accommodate the input requirements of the three architectures under evaluation: a baseline CNN, AlexNet, and a patch-based LSTM.

For models requiring RGB input, such as AlexNet, grayscale MNIST images were converted to three-channel RGB format. Additionally, all images used with AlexNet were resized to 224×224 pixels to match the input dimensions expected by the pretrained architecture. In contrast, for models like the baseline CNN and LSTM, the original dimensions of MNIST (28×28) and CIFAR-10 (32×32) were retained unless otherwise needed.

Normalization was applied to the CIFAR-10 dataset to standardize the pixel intensity values. Each RGB channel was normalized to have a mean and standard deviation of $(0.5, 0.5, 0.5)$, ensuring that the data values fell approximately within the range $[-1, 1]$.

For the patch-based LSTM architecture, additional preprocessing was performed to restructure each image into a sequential format. Images were divided into non-overlapping square patches, each of which was flattened into a vector. These vectors were then treated as ordered sequences for input to the LSTM.

Patch Configuration:

- **MNIST:** Each 28×28 grayscale image was split into 4×4 patches, resulting in $7 \times 7 = 49$ patches per image. Each patch was flattened into a vector of size 16 ($1 \times 4 \times 4$).

- **CIFAR-10:** Each $32 \times 32 \times 3$ image was divided into 4×4 patches, yielding $8 \times 8 = 64$ patches per image. Each patch became a vector of size 48 ($3 \times 4 \times 4$).

This transformation enabled the LSTM model to interpret spatial features as temporal sequences, thereby leveraging its ability to capture long-range dependencies across image regions.

2.3 Hyperparameters and Model Architectures

The following hyperparameters were used for training all three architectures (CNN, AlexNet, and Patch-based LSTM). These values were selected experimentally based on preliminary training results and adjusted to balance accuracy and training time:

- **Batch Size = 64:** A moderate batch size that provides a trade-off between training speed and stability.
- **Number of Epochs = 10:** Sufficient for observing model convergence without overfitting on MNIST or CIFAR-10 datasets.
- **Learning Rate = 0.001:** A common starting value for the Adam optimizer, allowing stable convergence for all models.
- **Optimizer = Adam:** This optimizer was chosen due to its robustness and fast convergence.

These hyperparameters were not optimized exhaustively but were selected through iterative testing to ensure fair and stable comparison between the models.

The custom CNN model consisted of the following layers: 2 convolutional layers, each followed by a ReLU activation and a MaxPooling layer. The extracted features are then passed through 2 fully connected layers. This results in a total of 6 layers (excluding activations and flattening).

The pretrained AlexNet was utilized, with its weights initialized from training on the ImageNet dataset. To adapt it to the MNIST and CIFAR-10 datasets, the final fully connected layer was modified to match the number of target classes (10). Additionally, feature extraction layers of the pretrained model were frozen to reduce training time and focus learning on the final classification layer.

The implemented Patch-based LSTM consists of a single-layer LSTM with a hidden size of 128. Each input image is split into flattened non-overlapping patches which are fed as a temporal sequence. The final hidden state is passed through a fully connected layer for classification.

2.4 Training and Evaluation Methodology

To ensure a fair and consistent comparison among the three architectures—Custom CNN, AlexNet, and Patch-based LSTM—all models were trained and evaluated using the same experimental setup. All datasets were loaded using `DataLoader` objects with a batch size of 64. The training data was shuffled at every epoch, and the test data was kept static to evaluate the model performance consistently. The number of training epochs was fixed at 10 for all experiments, which was sufficient for all models to converge on both MNIST and CIFAR-10 datasets.

2.4.1 Training Accuracy and Loss Curves

To monitor the learning progress of each model, the training loss and test accuracy were recorded for each epoch. These plots help visualize how well the models converge during training and whether they suffer from underfitting or overfitting.

- **CNN on MNIST:**

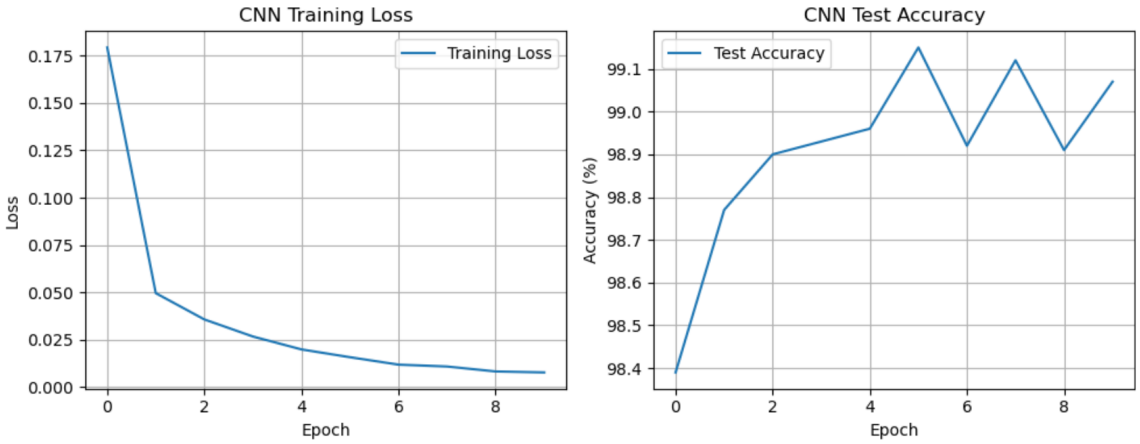

Figure 2.3: CNN Training Loss and Accuracy on MNIST

Observation: The CNN model shows rapid convergence with a steady drop in training loss, with test accuracy exceeds 99% within a few epochs, which indicates effective learning with no overfitting.

- **CNN on CIFAR-10:**

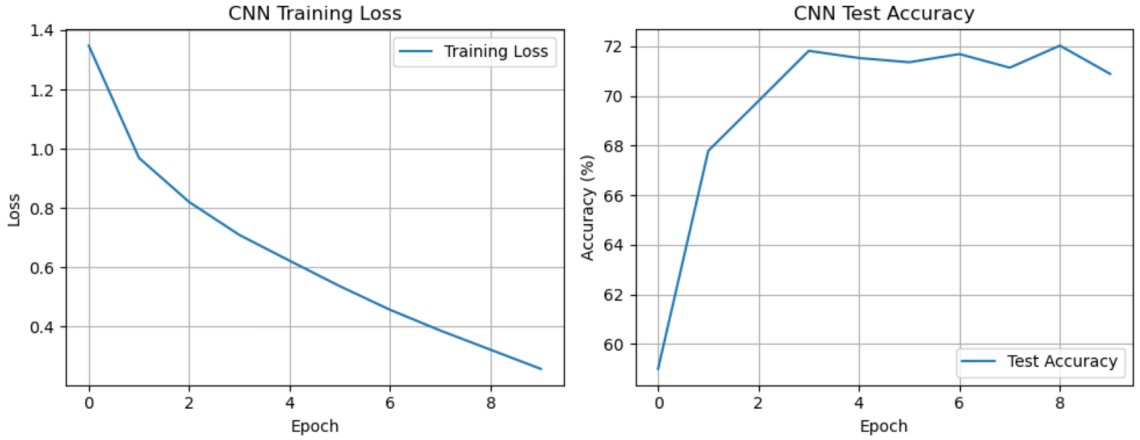

Figure 2.4: CNN Training Loss and Accuracy on CIFAR-10

Observation: The CNN model shows gradual improvement on CIFAR-10, with a steady decline in training loss and test accuracy stabilizing around 71–72%. Compared to MNIST, the performance is lower, reflecting the higher complexity and variability of the CIFAR-10 dataset.

- **AlexNet on MNIST:**

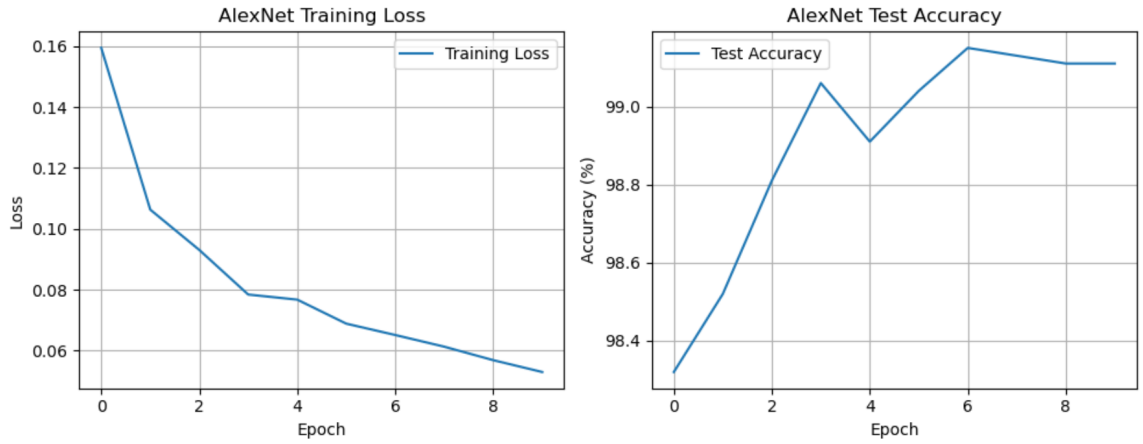

Figure 2.5: AlexNet Training Loss and Accuracy on MNIST

Observation: AlexNet on MNIST shows a steady decrease in training loss and achieves high test accuracy, exceeding 99%. The model converges smoothly, demonstrating its effectiveness even on simpler datasets like MNIST.

- **AlexNet on CIFAR-10:**

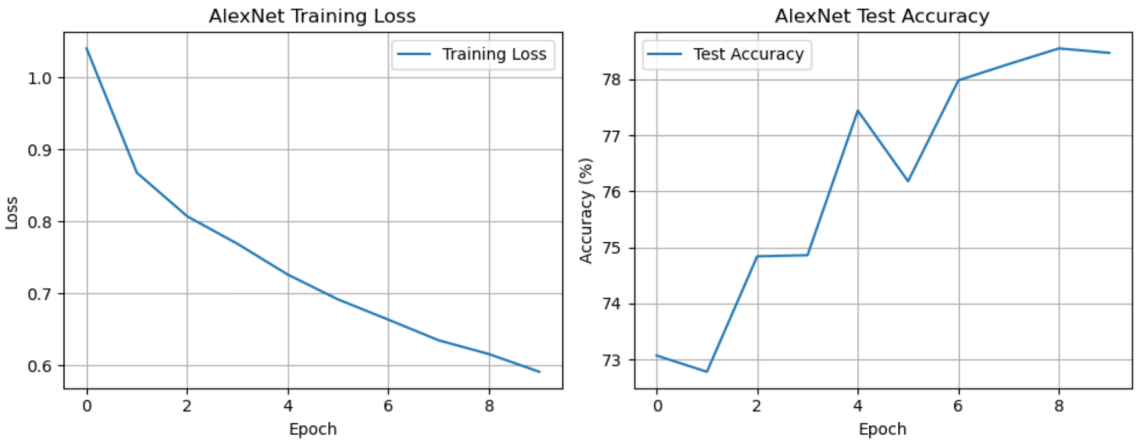

Figure 2.6: AlexNet Training Loss and Accuracy on CIFAR-10

Observation: On CIFAR-10, AlexNet shows a steady decline in training loss and a gradual improvement in accuracy, reaching around 78%. While the accuracy fluctuates slightly, the model demonstrates consistent learning and performs reasonably well on this more complex dataset.

- **Patch-LSTM on MNIST:**

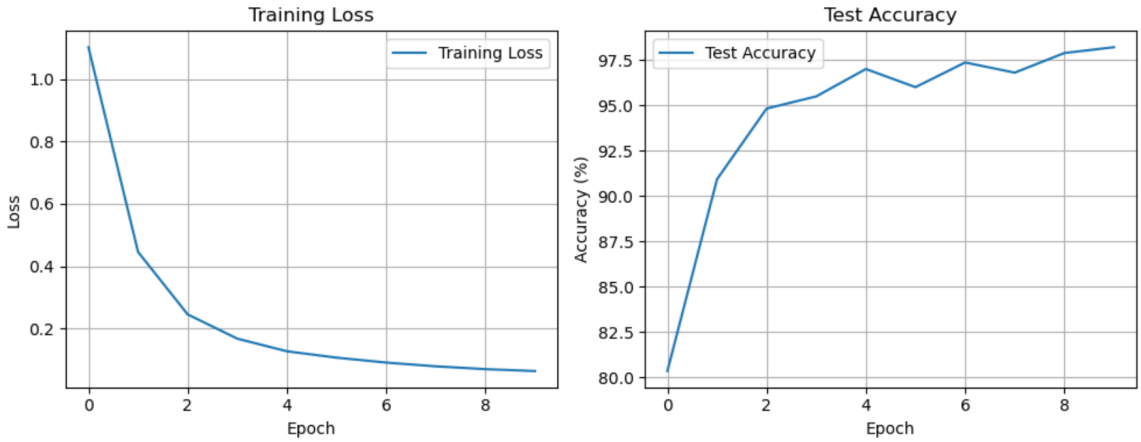

Figure 2.7: LSTM Training Loss and Accuracy on MNIST

Observation: The Patch-LSTM model on MNIST shows a strong performance, with a rapid decrease in training loss and a steady rise in test accuracy, reaching above 97%. This suggests that the model effectively learns from the sequential patch representation of the digits.

- **Patch-LSTM on CIFAR-10:**

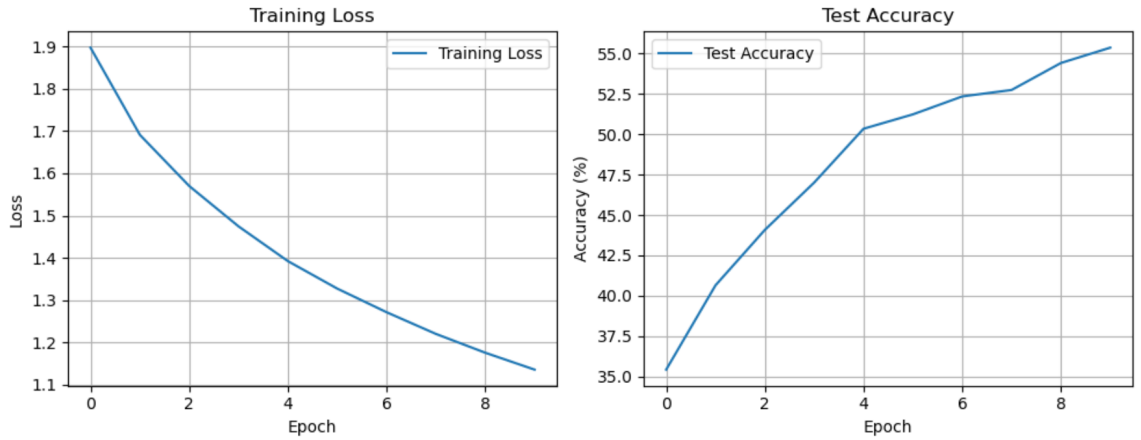

Figure 2.8: LSTM Training Loss and Accuracy on CIFAR-10

Observation: The Patch-LSTM model on CIFAR-10 shows a consistent decrease in training loss and a gradual improvement in accuracy, reaching around 55%. However, the overall accuracy remains relatively low, suggesting that the model struggles to capture complex patterns in the colored and diverse CIFAR-10 dataset using sequential patch processing.

2.4.2 Confusion Matrix Analysis

The confusion matrix provides detailed insights into the classification performance of each model by showing the number of correct and incorrect predictions per class. It is especially useful for identifying classes that are frequently misclassified.

- **CNN on MNIST:**

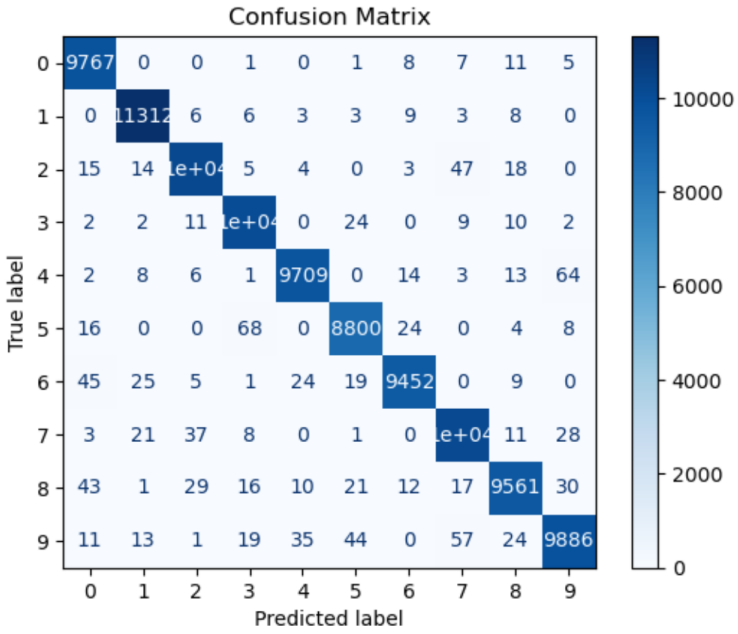

Figure 2.9: Confusion Matrix for CNN on MNIST

Observation: It shows highly accurate classification, with most predictions concentrated along the diagonal. Misclassifications are minimal, with slight confusion between similar digits such as 4 and 9 or 3 and 5, indicating excellent overall performance on this dataset.

- **CNN on CIFAR-10:**

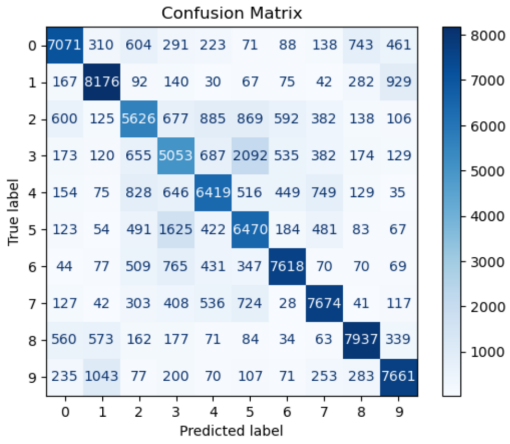

Figure 2.10: Confusion Matrix for CNN on CIFAR-10

Observation: It reveals noticeable misclassifications across several classes, while some classes like '1' and '6' are predicted relatively well, others such as '2', '3', and '8' show frequent confusion with similar-looking or overlapping categories. This highlights the increased challenge of CIFAR-10's diverse, colored images compared to MNIST.

- **AlexNet on MNIST:**

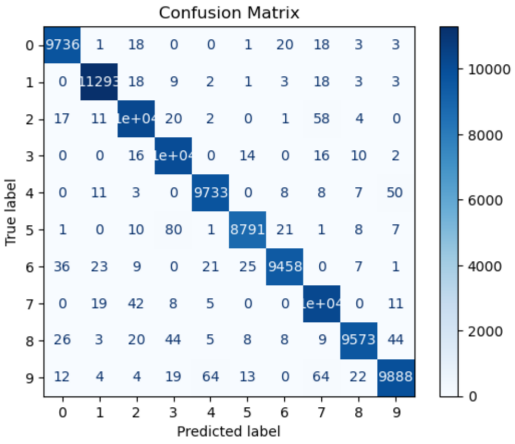

Figure 2.11: Confusion Matrix for AlexNet on MNIST

Observation: It shows high classification accuracy, with most values tightly clustered along the diagonal. Few misclassifications are present such as confusions between digits 3 and 5, or 8 and 9, but overall, the model demonstrates excellent performance in recognizing handwritten digits.

- **AlexNet on CIFAR-10:**

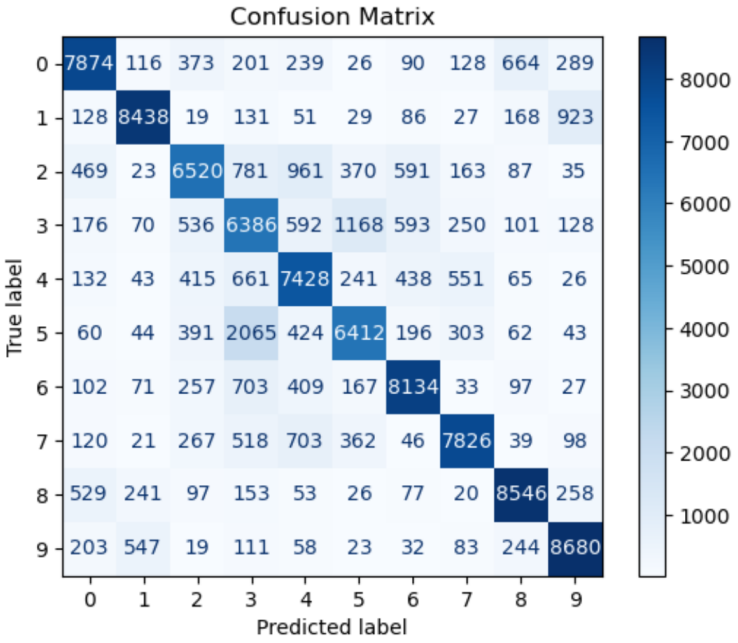

Figure 2.12: Confusion Matrix for AlexNet on CIFAR-10

Observation: It reveals noticeable misclassifications across several classes. While the diagonal elements dominate, indicating correct predictions, there is frequent confusion between visually similar categories—such as cats, dogs, and deer. This highlights the challenge of classifying more complex, color-rich images, despite AlexNet's deep architecture.

- **Patch-LSTM on MNIST:**

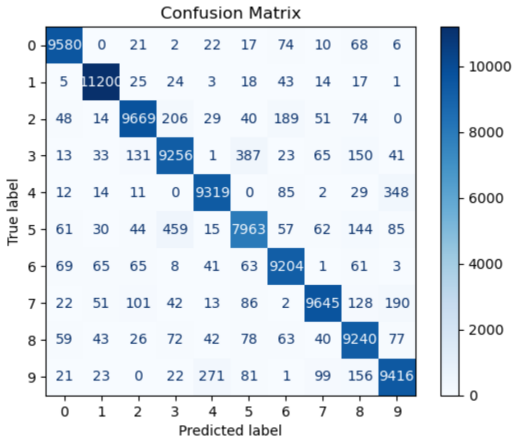

Figure 2.13: Confusion Matrix for LSTM on MNIST

Observation: It achieves solid classification performance, with most predictions correctly aligned along the diagonal. Some confusion is evident between classes with similar shapes, such as 3 and 5 or 4 and 9, but overall, the matrix reflects strong model accuracy and reliable digit recognition.

- **Patch-LSTM on CIFAR-10:**

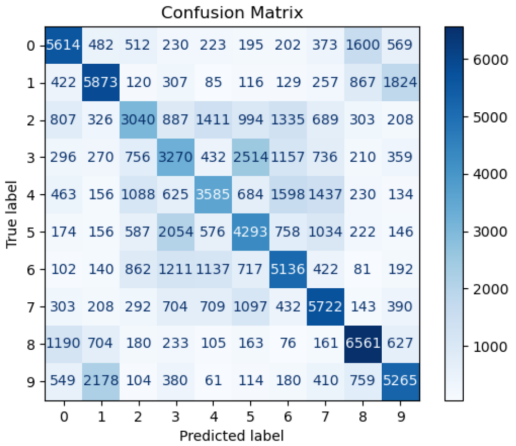

Figure 2.14: Confusion Matrix for LSTM on CIFAR-10

Observation: It struggles with classification accuracy, as evident from the widespread off-diagonal values in the confusion matrix. Significant confusion occurs between several classes, indicating the model's difficulty in distinguishing complex visual patterns in colored, high-variability images.

2.5 Results Comparative Analysis

The results presented in the below table and figures provide a comprehensive comparison of the three architectures across two datasets.

Table 2.1: Comparison of model performance across datasets

Model	Dataset	Accuracy (%)	Train Time (s)	Inference Time (s)
Custom CNN	MNIST	99.07	94.58	39.74
Custom CNN	CIFAR-10	70.89	202.71	85.10
AlexNet	MNIST	99.11	1031.89	139.29
AlexNet	CIFAR-10	78.47	926.53	201.66
Patch-LSTM	MNIST	98.21	89.21	38.79
Patch-LSTM	CIFAR-10	55.36	199.66	81.44

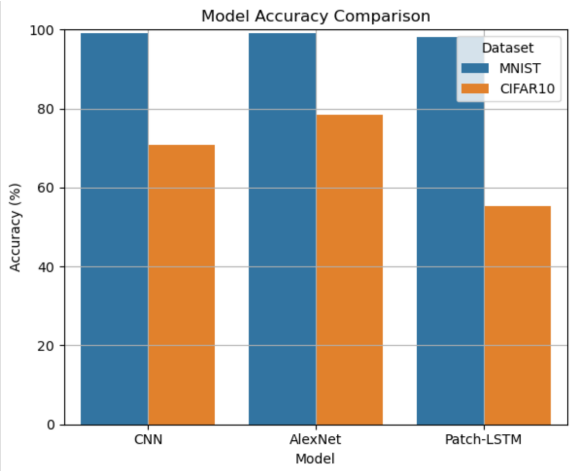

Figure 2.15: Accuracy Comparison Across Models and Datasets

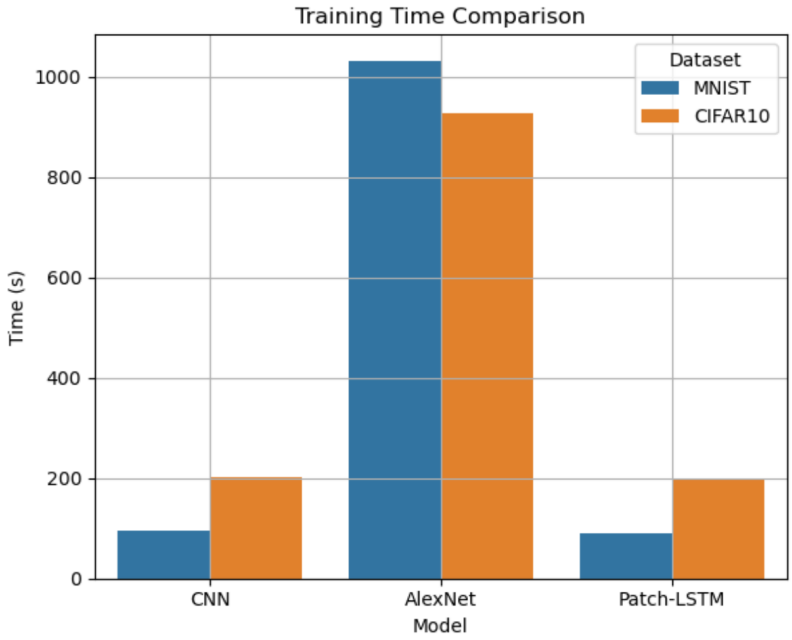

Figure 2.16: Training Time Comparison Across Models and Datasets

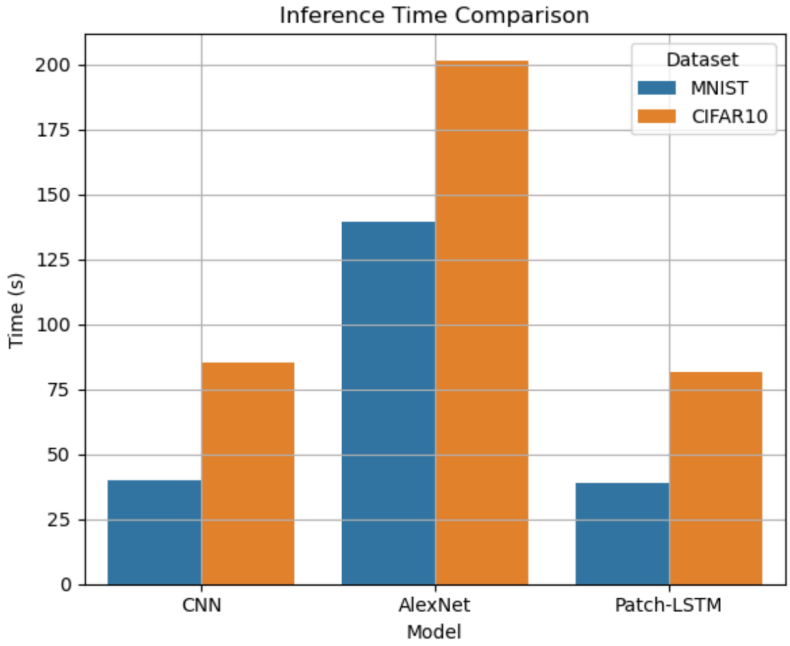

Figure 2.17: Inference Time Comparison Across Models and Datasets

Observations:

- **Accuracy:**
  - On the MNIST dataset:
    - AlexNet achieved the highest accuracy (99.11%), followed closely by CNN (99.07%).
    - Patch-LSTM also performed competitively (98.21%) using its sequential patch-based approach.
  - On the CIFAR-10 dataset:
    - AlexNet again outperformed other models with 78.47% accuracy.
    - Custom CNN showed moderate performance (70.89%), while Patch-LSTM performed the worst (55.36%).
- **Training Time:**
  - AlexNet required the longest training time on both datasets, exceeding 1000 seconds for MNIST.
  - Custom CNN and Patch-LSTM trained significantly faster, with Patch-LSTM being the fastest on MNIST (89.21s).
- **Inference Time:**
  - Patch-LSTM and custom CNN had faster inference times compared to AlexNet.
  - On MNIST, Patch-LSTM was the fastest (38.79s), while AlexNet was the slowest (139.29s).
  - On CIFAR-10, Patch-LSTM and custom CNN both maintained moderate inference speeds, faster than AlexNet.

2.6 Discussion of Advantages and Disadvantages of Each Architecture

2.6.1 Custom CNN Architecture:

- **Advantages:**
  - High accuracy on MNIST with fast training and inference times.
  - Lightweight and simple architecture, easy to train from scratch.
- **Disadvantages:**
  - Limited representational power for complex datasets like CIFAR-10.
  - Struggles with color and texture variations.

2.6.2 AlexNet Architecture:

- **Advantages:**
  - Strong performance on both datasets, especially CIFAR-10.
  - Deep and wide layers help capture more complex features.
- **Disadvantages:**
  - Very high training and inference times.
  - Requires more computational resources and memory.

2.6.3 Patch-Based LSTM Architecture:

- **Advantages:**
  - Competitive accuracy on MNIST with low training and inference times.
  - Captures sequential spatial features using patches.

- **Disadvantages:**
  - Poor performance on CIFAR-10 due to high visual complexity.
  - Lacks spatial inductive bias like convolution layers.

Each architecture exhibits unique strengths. Custom CNN offers a fast and efficient baseline, AlexNet achieves the highest accuracy, and Patch-based LSTM explores sequence modeling but requires more optimization for complex datasets.

3 Conclusion

This study evaluated the performance of three deep learning architectures—Custom CNN, AlexNet, and Patch-based LSTM—on two widely used image classification datasets: MNIST and CIFAR-10. The analysis covered training dynamics, accuracy, inference speed, and resource efficiency.

On the simpler MNIST dataset, all models achieved high accuracy, with AlexNet leading slightly at 99.11%, followed closely by custom CNN at 99.07% and Patch-LSTM at 98.21%. These results demonstrate that even lightweight models like custom CNN and Patch-LSTM can perform competitively on structured grayscale data. Additionally, the Patch-LSTM had the shortest training time (89.21s) and inference time (38.79s), highlighting its efficiency.

However, performance diverged more significantly on the more challenging CIFAR-10 dataset. AlexNet achieved the highest accuracy at 78.47%, significantly outperforming custom CNN (70.89%) and Patch-LSTM (55.36%). Despite its higher accuracy, AlexNet required substantially more training time (926.53s) and inference time (201.66s), suggesting a trade-off between accuracy and computational cost.

Overall, custom CNN offered the best balance between speed and performance for MNIST, while AlexNet was more suitable for complex datasets like CIFAR-10 despite its computational demands. Patch-LSTM proved to be an efficient alternative for sequential input processing but struggled with colored and diverse images.

References

[1] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] https://zilliz.com/glossary/convolutional-neural-network

[3] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[4] https://medium.com/@siddheshb008/alexnet-architecture-explained-b6240c528bd5

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] S. Bai, J. Z. Kolter and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[7] https://dataaspirant.com/lstm-long-short-term-memory/