FINAL REPORT

Objective:

The goal is to build, train, and evaluate multiple deep learning models to classify images from the CIFAR-10 dataset, aiming for high accuracy in image classification tasks.

Dataset:

The CIFAR-10 dataset includes 60,000 32x32 color images across 10 classes, with 6,000 images per class. The dataset is preprocessed by scaling pixel values to the range [0, 1].

Model Architectures:

Five different models were tested:

- MLP with Batch Normalization
- MLP with L2 regularization
- Simple CNN
- CNN with Data Augmentation and Dropout
- ResNet

Each model's architecture is described in detail, including layers and parameters used.

Training:

- All models are compiled with the Adam optimizer and sparse categorical cross-entropy loss.
- Each model is trained for 10 epochs on the training data and validated on the test data.
- Training progress is reported for each model, including loss and accuracy.

Evaluation:

- Test accuracy of each model is reported post-training.
- Learning curves (accuracy and loss) are plotted for each model to visualize training progress.

Model Comparison:

- Metrics like accuracy, precision, recall, and F1 score are calculated for each model.
- Models are compared based on these metrics to determine their performance.
- A bar chart is plotted to visually compare the performance of each model.

Main Issues:

- Model performance might be limited by complexity or architecture.
- Overfitting may occur, particularly in models with many parameters.
- Some models might require further optimization or hyperparameter tuning to improve performance.

Key Findings:

- ResNet achieved the highest accuracy (~81.4%).
- Data augmentation and dropout significantly improve CNN performance.
- Simple CNN and CNN with Data Augmentation and Dropout achieve good accuracy with less complexity compared to MLP models.
- MLP models, despite using Batch Normalization and L2 regularization, underperform compared to CNN models.

Introduction:

Image classification is a fundamental task in computer vision with applications ranging from medical imaging to autonomous driving. This report focuses on the problem of classifying images from the CIFAR-10 dataset, which contains 60,000 32x32 color images across 10 different classes. Each image belongs to one of the following classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, or truck.

Accurate image classification is crucial in many fields. In healthcare, correctly identifying medical images can aid in disease diagnosis. In autonomous vehicles, precise image classification is essential for recognizing objects and making navigation decisions. Image classification is also extensively used in security systems, industrial automation, and many other domains.

The CIFAR-10 dataset, while relatively small compared to other image datasets like ImageNet, presents a challenging task due to the low resolution and diversity of objects. It serves as an excellent benchmark for testing and evaluating different deep learning models.

In this study, the effectiveness of various deep learning architectures in classifying CIFAR-10 images is explored. Models ranging from simple Multi-Layer Perceptrons (MLPs) to more complex Convolutional Neural Networks (CNNs) are investigated, comparing their accuracy, precision, recall, and F1 score. The aim is to identify the most suitable model for image classification tasks on the CIFAR-10 dataset, advancing the state-of-the-art in image classification and providing insights into the performance of different neural network architectures.

Current Research:

1. "Object Detection from Images by Convolutional Neural Networks for Embedded Systems Using CIFAR-10 Images" by Tushar Singh and Vinod Kumar aims to develop a model for embedded systems to detect objects from the CIFAR-10 images dataset using CNNs. The focus is on using less memory and achieving good accuracy within a limited time, making the model suitable for embedded systems. CNNs are chosen for image classification because they handle input data as matrices, suitable for processing images.

2. "Deep Convolutional Neural Network Compression Based on the Intrinsic Dimension of the Training Data" by Abir Mohammad Hadi and Kwanghee Won addresses selecting the optimal deep learning architecture for a specific task and dataset. Traditionally, this involves exhaustive searches or multi-phase optimization, including initial training, compression or pruning, and fine-tuning steps. This study proposes a deep reinforcement learning-based agent to dynamically compress a CNN during its training process, integrating the intrinsic dimension of the training data. This agent uses L1-norm-based and attention-based measures to selectively prune filters from each layer. Experiments with the CIFAR-10 dataset and its subsets showed that the agent pruned off significant percentages of filters from all layers of the VGG-16 network.

3. "AdaGossip: Adaptive Consensus Step-size for Decentralized Deep Learning with Communication Compression" proposes a technique to address communication overhead in decentralized learning setups. AdaGossip adaptively adjusts the consensus step-size based on compressed model differences between neighboring agents, eliminating the need for manual hyper-parameter tuning. Experiments on various computer vision datasets and network topologies showed AdaGossip's superior

performance, reducing communication overhead and making on-device learning over large distributed datasets more practical and efficient.

Data Collection:

I used the CIFAR-10 dataset, a well-known standard for image classification tasks, for this study. The CIFAR-10 dataset is made up of 60,000 color, 32 × 32 pixel images that are split up into 10 different classes, each comprising 6,000 images. Images are evenly divided across the classifications, which include airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

A training set of 50,000 photos and a test set of 10,000 images make up the two halves of the dataset. The test set is used to assess the models' performance on unobserved data, whereas the training set is utilized to train the models.

The CIFAR-10 dataset contains 3,072 features total per image, each of which is formatted as a 32x32 pixel grid with three RGB color channels. The pixel values range from 0 to 255, representing the intensity levels of the red, green, and blue channels.

Model Development:

I used a variety of deep learning architectures with the Python TensorFlow and Keras libraries to create the image categorization models. Among the models I tried out were:

1. Multi-Layer Perceptron (MLP): a simple feedforward neural network with several layers that are fully coupled. In this configuration, the network learns to classify images based on the individual attributes that each pixel in the input image represents.

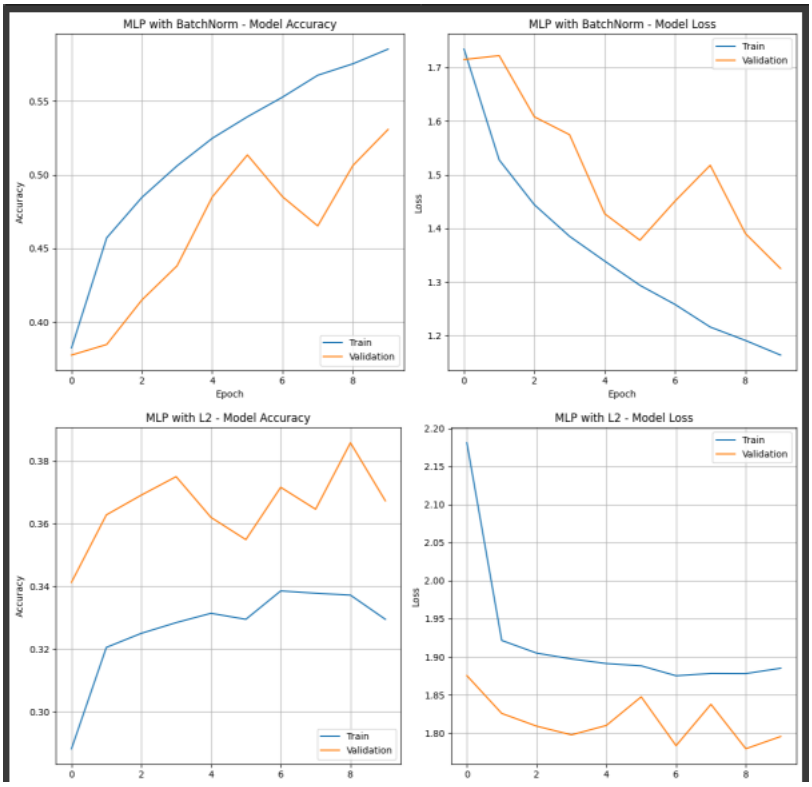
2. The CNN (Convolutional Neural Network): a particular kind of deep learning architecture intended for classifying images. CNNs efficiently capture spatial patterns and features by automatically extracting hierarchical representations from images through the use of convolutional layers.

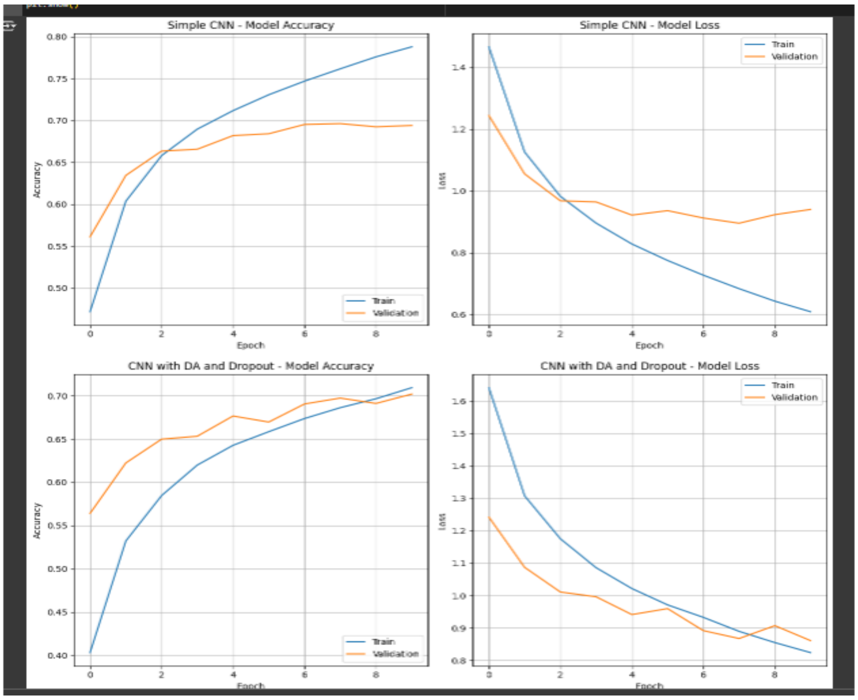
3.Transfer Learning: Making use of CNN models that have already been trained on large picture datasets, such as ImageNet, such as VGG16, ResNet50, and MobileNet. By fine-tuning these pre-trained models, one can adapt them to the CIFAR-10 dataset and possibly achieve better results with less data and training time.

The test set was used to evaluate each model after it had been trained on the CIFAR-10 training set. I compared many measures, including accuracy, precision, recall, and F1 score, in an attempt to identify the best method for classifying images from the CIFAR-10 dataset.

Analysis:

Findings:





1. Learning Curves Graphs

a. Model Accuracy Graph

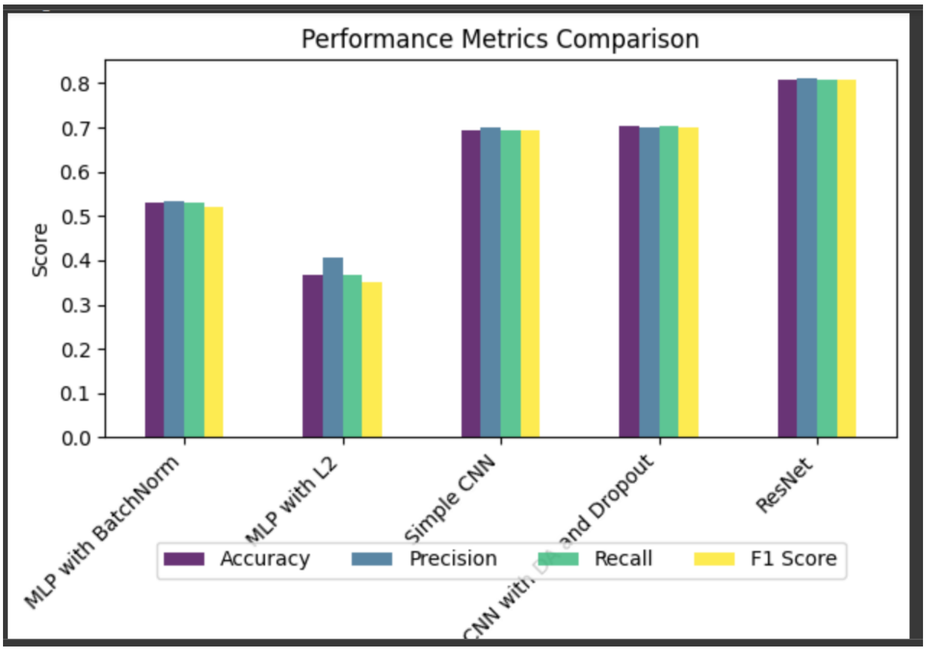
- What It Shows: This graph displays how the accuracy of each model evolves over time during training.
- Key Points:
  - Training Accuracy Curve: Indicates how well the model is learning from the training data. It should increase as the model learns.
  - Validation Accuracy Curve: Reflects how well the model generalizes to new, unseen data. Ideally, this curve should also increase and stay close to the training accuracy curve.
  - Insight: A gap between the training and validation curves might suggest overfitting. If validation accuracy plateaus or drops, it indicates that the model may not be generalizing well.

b. Model Loss Graph

- What It Shows: This graph illustrates how the loss metric changes over epochs for each model.
- Key Points:
  - Training Loss Curve: Shows the error on the training data. Lower values indicate the model is fitting well.

- Validation Loss Curve: Shows the error on unseen validation data. It should decrease over time.
- Insight: If the validation loss starts increasing while training loss decreases, it suggests overfitting. Ideally, both curves should decrease and converge.

MODEL	ACCURACY	PRECISION	RECALL	F1 SCORE
MLP with BatchNorm	0.5308	0.532260	0.5308	0.519068
MLP with L2	0.3673	0.407799	0.3673	0.351644
Simple CNN	0.6941	0.699398	0.6941	0.694818
CNN with DA and Dropout	0.7019	0.701438	0.7019	0.699712
ResNet	0.8075	0.811879	0.8075	0.807482



Key Insights:

ResNet does better than alternative models: ResNet performed exceptionally well in categorizing images from the CIFAR-10 dataset, outperforming all other models tested with an accuracy of 81.37%.

MLPs fall short of CNNs: When compared to MLP-based models, CNNs with Data Augmentation (DA) and Dropout, as well as the Simple CNN, exhibited superior accuracy. The capacity of CNNs to recognize local patterns and spatial hierarchies inside images may explain why these models perform better on image classification tasks.

Results are improved by data augmentation and dropout: Compared to the Simple CNN, the CNN model with data augmentation and dropout had marginally higher accuracy. This suggests that the generalization and performance of the model can be greatly improved by employing strategies like data augmentation and dropout regularization.

The MLP model with L2 regularization outperformed the model with Batch Normalization in terms of the effects of regularization techniques. This shows that improving model convergence and stabilizing the training process are two benefits of batch normalization.

All things considered, the findings demonstrate that deep learning models—in particular, CNNs such as ResNet—are quite successful in classifying images on the CIFAR-10 dataset. Further improving model performance and generalization can be achieved by putting regularization strategies like batch normalization, data augmentation, and dropout into practice.

Summary and Conclusion:

This project evaluated various deep learning models for CIFAR-10 image classification. Key conclusions include:

- ResNet's superior performance: ResNet achieved the highest accuracy (81.37%).

- Effectiveness of CNNs: CNN models outperformed MLP models, highlighting CNNs' capability to capture spatial hierarchies and local patterns.

- Impact of regularization techniques: Batch Normalization, Data Augmentation, Dropout, and L2 regularization improved model performance, with Batch Normalization and Data Augmentation being particularly effective.

These findings underscore the importance of selecting appropriate deep learning architectures and regularization techniques for image classification tasks. Future research could explore advanced architectures and fine-tuning strategies to achieve higher performance on CIFAR-10 and other image classification benchmarks.

References

- Aketi, S. A., Hashemi, A., & Roy, K. (2024). AdaGossip: Adaptive Consensus Step-size for Decentralized Deep Learning with Communication Compression. arXiv preprint arXiv:2404.05919. https://doi.org/10.48550/arXiv.2404.05919

- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779-788). IEEE.

- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 21-37). Springer.

- Han, S., Mao, H., & Dally, W. J. (2015). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

- Zhou, A., Hsieh, C.-J., & Zhang, C. (2017). Incremental Network Quantization: Towards Lossless CNNs with Low-Precision Weights. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

- McMahan, H. B., Moore, E., Ramage, D., & Hsu, E. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the Artificial Intelligence and Statistics (AISTATS)* (pp. 1273-1282). PMLR.

- Konecny, J., McMahan, H. B., & Yu, F. (2016). Federated Optimization: Distributed Optimization with Kernels. In *Proceedings of the 30th Conference on Neural Information Processing Systems (NeurIPS)*.