**Report on Image Classification Using Various CNN Architectures on MNIST, FMNIST, and CIFAR-10 Datasets**

**1. Introduction**

* **Objective**: This project aims to compare the performance of different CNN architectures on various datasets. Specifically, we will evaluate LeNet-5, AlexNet, GoogLeNet, VGGNet, ResNet, Xception, and SENet on MNIST, FMNIST, and CIFAR-10 datasets.
* **Goal**: To compare the results across different architectures on each dataset in terms of accuracy, precision, recall, F1-score, and loss curves.

**2. Datasets**

* **MNIST**: Handwritten digits, 28x28 grayscale images.
* **FMNIST**: Fashion items like shirts, shoes, etc., also 28x28 grayscale images.
* **CIFAR-10**: 10 classes of general images (e.g., cars, animals), 32x32 RGB images.
* **Preprocessing Steps**:
  + Resizing (if needed) and normalizing images as per requirement of dataset.

**3. Model Architectures**

* Brief descriptions of each architecture:
  + **LeNet**: A simple CNN, suitable for smaller datasets like MNIST.
  + **AlexNet**: Deeper architecture that introduces ReLU activation, dropout, and data augmentation.
  + **GoogLeNet**: Uses Inception modules, allowing it to learn at multiple scales.
  + **ResNet**: Introduces residual connections to tackle vanishing gradient problems in deep networks.
  + **VGG**: Uses very deep layers with 3x3 convolutions for feature extraction.
  + **Xception**: An extension of Inception with depth wise separable convolutions.
  + **SENet**: Introduces "Squeeze and Excitation" blocks to adaptively recalibrate feature maps.
* **Model Parameters**:
  + Number of layers, filter sizes, and unique features for each model.
  + Parameter counts for each model as complexity indicators.

**4. Training Process**

* **Loss Function:**

The loss function used is Cross-Entropy Loss (torch.nn.CrossEntropyLoss()), which is well-suited for multi-class classification tasks.

* **Optimizer:**
  + Adam Optimizer (optim.Adam): The Adam optimizer was chosen for its adaptive learning rate capabilities, making it suitable for deep networks. It combines the advantages of both RMSProp and SGD with momentum, allowing faster convergence.
  + Learning Rate: The learning rate for Adam was set to 0.001, which balances learning speed and stability during training.
* **Learning Rate Scheduler:** A Step Learning Rate Scheduler (torch.optim.lr\_scheduler.StepLR) was used to decay the learning rate at specific intervals. The scheduler reduces the learning rate by a factor of 0.1 every 2 epochs (step size=2, gamma=0.1), allowing the model to make finer adjustments as training progresses and reducing the likelihood of overshooting the optimal weights.
* **Batch Size and Number of Epochs:**
  + Batch Size: A batch size of 32 was used to balance computational efficiency and model generalization.
  + Epochs: The model was trained for a total of 5 epochs for MNIST and Fashion-MNIST dataset but trained for 10 epochs for CIFAR-10 dataset for better results as CIFAR-10 is a complex dataset. Number of epochs chosen based on validation performance to prevent overfitting while ensuring adequate training.
* **Training Configuration:**
  + Transfer Learning: I utilized pre-trained versions of AlexNet, GoogLeNet, and ResNet18, which were initially trained on ImageNet. This pre-training allows these networks to start with meaningful feature representations rather than learning everything from scratch.
  + Output Layer Modification: To adapt the pre-trained models for a 10-class classification problem, the final fully connected layers were replaced to produce outputs compatible with the target dataset.
  + Fine-Tuning: I chose to fine-tune the entire network, allowing the models to adjust all layers to better fit the characteristics of datasets.

**5. Evaluation Metrics**

* **Loss Curves**: Visualize loss curves for each model-dataset combination to observe overfitting or underfitting.
* **Metrics**:
  + **Accuracy**: Overall correctness of predictions.
  + **Precision**: Ability to avoid false positives.
  + **Recall**: Ability to avoid false negatives.
  + **F1-Score**: Harmonic mean of precision and recall, providing a balanced measure.

**6. Results and Analysis**

* **Performance Summary**:

**MNIST**:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Metrics/Models** | **LeNet** | **AlexNet** | **GoogLeNet** | **ResNet** | **VGGNet** | **Xception** | **SENet** |
| Accuracy (%) | 98.7 | 99.3 | 99.7 | 99.6 | 99.4 | 99.67 | 99.75 |
| Precision | 0.98 | 0.993 | 0.997 | 0.996 | 0.994 | 0.997 | 0.998 |
| Recall | 0.98 | 0.993 | 0.997 | 0.996 | 0.994 | 0.997 | 0.998 |
| F1-Score | 0.98 | 0.993 | 0.997 | 0.996 | 0.994 | 0.997 | 0.998 |

**FMNIST**:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Metrics/Models** | **LeNet** | **AlexNet** | **GoogLeNet** | **ResNet** | **VGGNet** | **Xception** | **SENet** |
| Accuracy (%) | 89.13 | 90.94 | 94.62 | 94.52 | 91.94 | 95.14 | 94.86 |
| Precision | 0.89 | 0.908 | 0.946 | 0.945 | 0.92 | 0.95 | 0.948 |
| Recall | 0.89 | 0.908 | 0.946 | 0.945 | 0.92 | 0.95 | 0.948 |
| F1-Score | 0.89 | 0.908 | 0.946 | 0.945 | 0.92 | 0.95 | 0.948 |

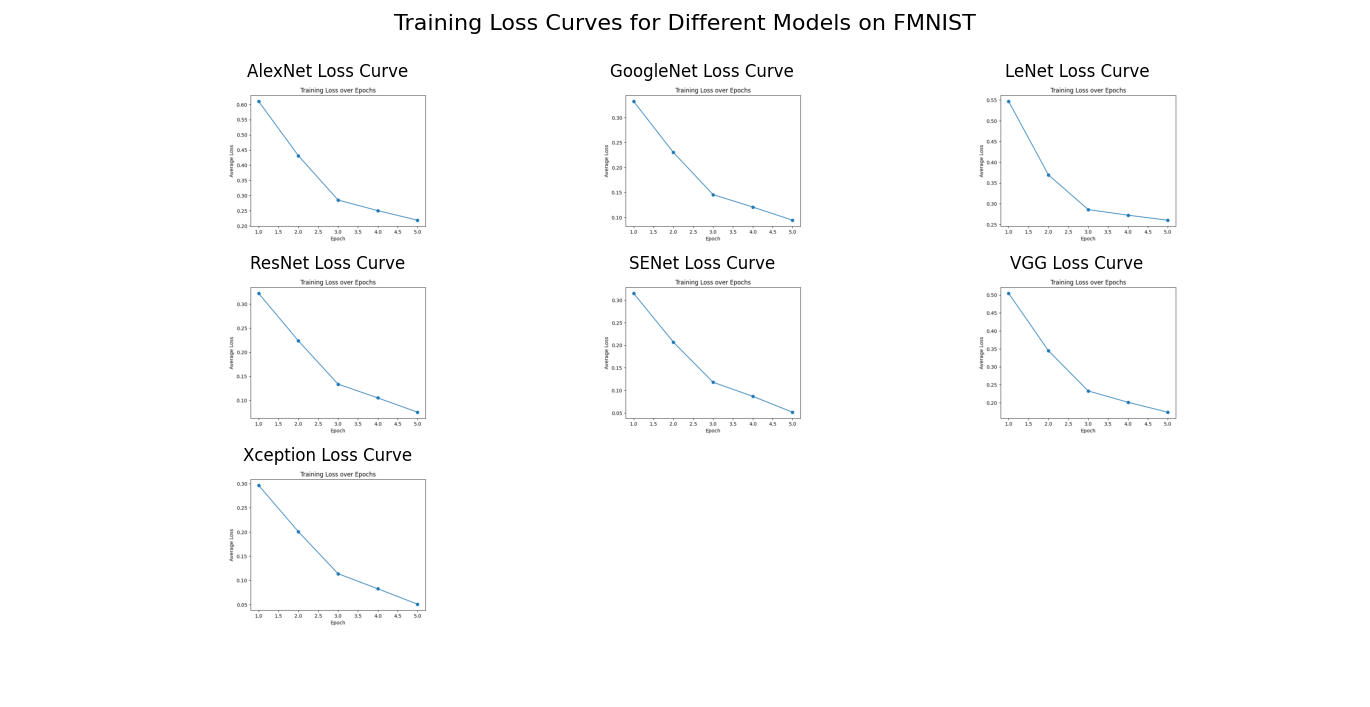
**CIFAR-10:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Metrics/Models** | **LeNet** | **AlexNet** | **GoogLeNet** | **ResNet** | **VGGNet** | **Xception** | **SENet** |
| Accuracy (%) | 66.2 | 10 | 92.74 | 90.33 | 75.61 | 94.03 | 93.75 |
| Precision | 0.669 | 0.01 | 0.92 | 0.9 | 0.75 | 0.94 | 0.939 |
| Recall | 0.662 | 0.1 | 0.92 | 0.9 | 0.75 | 0.94 | 0.937 |
| F1-Score | 0.664 | 0.01 | 0.92 | 0.9 | 0.75 | 0.94 | 0.937 |

* **Loss Curve Comparisons**:
* Loss curve comparisons is conducted for each model across all datasets to analyse performance

A graph of a training loss

Description automatically generated with medium confidence



A graph of training loss curves

Description automatically generated

* **Metric Comparison**:
* **MNIST Dataset: SENet** achieves the highest accuracy (99.75%) and F1-score (0.99), making it the best architecture for this dataset.
* **FMNIST Dataset: Xception** achieves the highest accuracy (95.14%) and F1-score (0.95), making it the best architecture for this dataset**.**
* **CIFAR-10: Xception** achieves the highest performance on CIFAR-10, with an accuracy of 94.03% and F1-scores of 0.94, indicating its robustness in handling complex, color images.

**7. Discussion**

* **Best Overall Architectures: Xception** and **SENet** consistently perform well across all datasets, demonstrating adaptability and robustness in handling both simple (**MNIST**) and complex (**CIFAR-10**) datasets.
* **Suitability of Simple Architectures**: While **LeNet** performs reasonably on **MNIST**, it struggles with **FMNIST** and **CIFAR-10**, highlighting the need for more complex architectures in challenging datasets.
* **Performance of AlexNet**: **AlexNet** shows limitations, especially on **CIFAR-10**, where it falls significantly behind. This suggests it may not be ideal for datasets requiring intricate feature extraction.

**8. Conclusion**

* **Summary of Findings**: **Xception** and **SENet** emerge as the top-performing models, with the highest adaptability across diverse datasets, while **LeNet** and **AlexNet** are better suited for simpler tasks.