CSE4261 Assignment1
ID: 2012376105

Mst. Shimla Sinthia

25th May 2025

1. Introduction

Convolutional Neural Networks (CNNs) have demonstrated remarkable performance on image classification tasks. This assignment aims to compare the performance of 10 popular CNN architectures, pre-trained on the ImageNet dataset, when fine-tuned on a subset of 20 classes from the CIFAR-100 dataset. The goal is to analyze their accuracy, loss, model size, and inference time to identify which architectures generalize better to new image domains.

2. Dataset and Preprocessing

We used 20 selected fine labels from the CIFAR-100 dataset. The images were resized from 32×32 to 224×224 to match the input requirement of the pre-trained models. All pixel values were normalized to the range [0, 1]. The dataset was split into training and test sets provided by the dataset, and labels were one-hot encoded.

3. Models Used

The following CNN architectures were selected:

- VGG16
- VGG19
- ResNet50
- ResNet101
- NASNetMobile
- InceptionV3
- Xception
- EfficientNetB0
- DenseNet121
- MobileNetV2

4. Experimental Setup

Each model was loaded with pre-trained ImageNet weights (excluding the top layer), and a new classifier head was added:

- Global Average Pooling Layer
- Dense layer with 128 ReLU units

- Output Dense layer with 20 softmax units

Training was conducted for 3 epochs using the Adam optimizer and categorical cross-entropy loss. The base layers were frozen during training to utilize the learned features.

5. Evaluation Metrics

The models were compared based on:

- Test Accuracy
- Test Loss
- Model Size (in MB)
- Inference Time per Image (in seconds)

6. Results

Table 1: Comparison of CNN Models on 20 Classes of CIFAR-100

Model	Accuracy	Loss	Model Size (MB)	Inference Time (s)
VGG16	0.4315	1.8905	56.39	0.1235
VGG19	0.3480	2.1530	76.65	0.1835
ResNet50	0.0500	2.9957	90.99	0.0804
ResNet101	0.1080	2.8885	163.74	0.1609
NASNetMobile	0.8155	0.5814	16.81	0.0764
InceptionV3	0.7930	0.6849	84.18	0.0782
Xception	0.8065	0.6342	80.59	0.1041
EfficientNetB0	0.0500	2.9957	16.08	0.0638
DenseNet121	0.8075	0.6157	27.36	0.0838
MobileNetV2	0.7840	0.6790	9.25	0.0477

7. Discussion

The results reveal that lightweight models such as **NASNetMobile**, **DenseNet121**, and **Xception** offer an excellent trade-off between accuracy and model size. **NASNetMobile** achieved the highest accuracy (81.55%) with a small model size of just 16.81 MB and low inference time.

Classical models like **VGG16** and **VGG19** performed moderately but are quite large and computationally expensive. Deeper architectures such as **ResNet50**, **ResNet101**, and **EfficientNetB0** underperformed, likely due to overfitting or mismatches in feature learning for this specific 20-class subset.

MobileNetV2, the smallest model at 9.25 MB, also performed very well (78.4% accuracy), making it ideal for deployment in mobile or embedded systems.

InceptionV3, though relatively larger (84.18 MB), achieved a strong performance of 79.3% accuracy, making it a powerful choice when moderate resources are available and high accuracy is required.

8. Conclusion

This experiment demonstrates that pre-trained CNNs can effectively adapt to new datasets through transfer learning. However, performance depends on the architecture's design, size, and ability to generalize. For this 20-class CIFAR-100 subset, **NASNetMobile**, **DenseNet121**, and **Xception** emerged as the most balanced in terms of performance, efficiency, and size.

7. Code Link

The complete source code used for training and evaluating the CNN models is available on Google Colab. You can access it using the following hyperlink:

Colab Link

Alternatively, you can directly use the following URL:
`https://colab.research.google.com/drive/1pIR9sd43ITxnnVRx0ul2ZnF-mkem0tMz?usp=sharing`