Image Classification on CIFAR-100

Objective:

This assignment aims to provide hands-on experience with various machine learning and deep learning models for image classification tasks, specifically focusing on the CIFAR-100 dataset. You will be evaluating the performance of different models, analyzing their strengths and weaknesses, and gaining insights into the challenges and techniques specific to image data.

Dataset:

The **CIFAR-100 dataset** consists of 60,000 32x32 color images in 100 classes, with 600 images per class. There are 50,000 training images and 10,000 test images. The 100 classes are grouped into 20 superclasses. You can typically access this dataset easily through deep learning libraries like torchvision.datasets in PyTorch or tensorflow.keras.datasets in TensorFlow/Keras.

Models:

You will be working with a selection of the following models suitable for image classification:

1. **Custom Convolutional Neural Network (CNN):** Design and implement a basic CNN architecture from scratch (e.g., a few convolutional layers, pooling layers, and fully connected layers).
2. **ResNet (e.g., ResNet18, ResNet34, ResNet50):** Implement and train a Residual Network, either from scratch or by fine-tuning a pre-trained version.
3. **VGG (e.g., VGG16, VGG19):** Implement and train a VGG network, potentially using pre-trained weights for fine-tuning.
4. **DenseNet (e.g., DenseNet121):** Implement and train a Densely Connected Convolutional Network, known for feature reuse, potentially using pre-training.
5. **EfficientNet (e.g., EfficientNetB0 - B7):** Explore using a member of the EfficientNet family, known for balancing accuracy and computational cost, likely via fine-tuning.
6. **ConvNeXt:** Implement or fine-tune a ConvNeXt model, a modern CNN architecture inspired by Vision Transformers.
7. **Vision Transformer (ViT):** Implement or fine-tune a Vision Transformer model, applying the transformer architecture directly to image patches.
8. **Swin Transformer:** Implement or fine-tune a Swin Transformer, a hierarchical

vision transformer using shifted windows.

9. **(Optional) Support Vector Machine (SVM) with Feature Extraction:** As a baseline, extract features from images (e.g., HOG, SIFT, or features from a pre-trained CNN) and train an SVM classifier.

Tasks:

1. **Implementation:**
    - Implement the specified models using a deep learning library of your choice (e.g., PyTorch, TensorFlow/Keras).
    - Leverage existing library implementations, especially for pre-trained models.
    - Implement appropriate data loading, preprocessing (resizing, normalization), and data augmentation pipelines for the CIFAR-100 dataset.
2. **Evaluation:**
    - Train and evaluate each model on the CIFAR-100 dataset using the standard train/test split.
    - Use appropriate evaluation metrics for multi-class image classification, such as:
        - Accuracy (Top-1 and optionally Top-5)
        - Precision, Recall, F1-score (consider macro, micro, or weighted averages)
        - Confusion Matrix
3. **Comparison:**
    - Compare the performance of the different models on the CIFAR-100 test set.
    - Analyze their strengths and weaknesses regarding accuracy, training time, model complexity, and sensitivity to hyperparameters.
    - Discuss the impact of using pre-trained models versus training from scratch.
4. **Analysis:**
    - Investigate the impact of hyperparameter tuning (e.g., learning rate, batch size, optimizer choice, weight decay) on model performance.
    - Experiment with different data augmentation techniques (e.g., random flips, rotations, color jittering) and analyze their effect on generalization.
    - Visualize some results, such as misclassified images or filter activations (for CNNs).
5. **Report:**
    - Write a comprehensive report summarizing your findings. Include the following sections:
        - **Introduction:** Briefly describe the image classification task and the CIFAR-100 dataset and models.
        - **Results:** Present the performance of each model using tables and graphs. Include key metrics and visualizations.

- **Discussion:** Analyze the results, compare the models, discuss the impact of different techniques (pre-training, augmentation, hyperparameter tuning), and highlight challenges encountered.
- **Conclusion:** Summarize your key findings and draw conclusions about the effectiveness of different models for CIFAR-100 classification.

Submission:

Submit your code and your final report on MS Teams.

Good luck!