Mobile Phone Side Classification using Deep Learning

Objective

This project aims to develop a robust image classification model capable of categorizing images into one of three distinct classes based on the presented view of a mobile phone:

- Front
- Back
- None (indicating that the image does not represent a valid front or back view of a mobile phone)

This classification system serves as a foundational component for subsequent applications such as product tagging, automated cataloging, and quality control within e-commerce and electronics sectors.

Frameworks & Tools

- **Programming Language:** Python
- **Deep Learning Framework:** PyTorch
- **Development Environment:** Jupyter Notebook

Models Utilized

To fully leverage the benefits of transfer learning, a strategic selection of pre-trained convolutional neural network (CNN) models was undertaken. Each chosen model offers distinct architectural advantages that contribute to robust image classification performance across varying computational and deployment environments.Employed Models:

- **ResNet50:**
    - Deep residual architecture with "skip connections" to address vanishing gradient problem.
    - Facilitates training of very deep networks.
    - Captures intricate and hierarchical visual patterns.
    - Crucial for distinguishing subtle differences and achieving high accuracy.
    - Ideal for transfer learning due to generalizable features learned from ImageNet.

- **MobileNetV2:**
    - Lightweight and highly efficient design.
    - Based on inverted residual blocks with linear bottlenecks, reducing computational complexity and parameters.
    - Suitable for real-time applications, embedded systems, and mobile deployment (resource-constrained environments).
    - Excellent balance between performance and resource utilization.
    - Optimized for deployment on edge devices, enabling on-device inference without constant cloud connectivity.

Task Overview

This initiative constitutes a multi-class image classification task involving three target categories: front, back, and none.

Data Preparation

Class Definitions

- **Front:** Characterized by a clear frontal perspective of a mobile device, typically showcasing the screen or user interface.
- **Back:** Images depicting the rear panel of the device, including elements such as cameras, logos, or the back casing.
- **None:** Encompasses objects that could potentially be mistaken for mobile phones (e.g., televisions, calculators, tablets, mirrors, game consoles, hands holding phones, laptops). This class contributes to the model's robustness by minimizing false positives.

Challenges Observed

- The model initially demonstrated proficiency in classifying 'front' images due to their highly distinguishable features (e.g., the screen).
- However, difficulties arose in differentiating between 'back' and 'none' classes, largely because objects like tablets or cased phones exhibited visual similarities.

Mitigation:

- Additional "hard" examples were collected, specifically for ambiguous cases (e.g., phones with covers, close-up images of tablets).

- Emphasis was placed on ensuring class diversity to enhance generalization capabilities.

Data Augmentation

Given the constrained dataset size, data augmentation played a pivotal role in mitigating overfitting and improving generalization. The following techniques were applied:

- Random rotations
- Horizontal flipping
- Brightness and contrast adjustments
- Cropping
- Hue and saturation shifts
- Normalization (based on ImageNet statistics for pre-trained models)

Dataset Overview

Category	Training Samples	Validation Samples	Train/Val Ratio
None	61	15	4.07
Back	96	25	3.84
Front	81	20	4.05
Total	238	60	—

Total Samples: 298

Model Architecture & Training Strategy

Transfer Learning

Both ResNet50 and MobileNetV2 were initialized with ImageNet-pretrained weights. This methodology facilitates:

- Accelerated convergence
- Improved generalization on smaller datasets
- Leveraging pre-learned low-level features (e.g., edges, shapes)

Fine-Tuning Strategy

- **Frozen Early Layers:** The initial layers were kept frozen to preserve the pretrained visual features.
- **Unfrozen Top Layers:** The upper layers were unfrozen to allow adaptation to the specific mobile classification task.

Custom Classifier Head

The final classification layer was replaced with a bespoke classifier head, comprising:

- AdaptiveAvgPool2d → Flatten
- Three Fully Connected (FC) Layers
  - FC → BatchNorm → PReLU → Dropout (0.5)
  - A final FC layer mapping to 3 output classes
- **PReLU Activation:** Enables the network to learn activation parameters, offering enhanced flexibility compared to traditional ReLU.
- **Dropout (p=0.5):** Implemented to prevent overfitting by randomly deactivating neurons during the training phase.
- **Batch Normalization:** Contributes to stable learning by normalizing feature distributions.

Class Imbalance Handling & Loss Function

Focal Loss

To address the observed class imbalance and enhance performance on challenging examples (particularly within the 'back' and 'none' classes), Focal Loss was employed:

{Focal Loss} = (1 - p_t)^\gamma \cdot \text{CE}$`

Where:

- p_t represents the model's estimated probability for the true class
- γ (typically set to 2) diminishes the loss contribution from easily classified examples
- CE denotes the cross-entropy loss

Rationale for Focal Loss:

- It effectively down-weights well-classified samples.
- It directs the training focus towards more difficult, misclassified examples.

Optimization Strategy

Learning Rate Scheduler

- Utilized to decay the learning rate over time or based on a plateau in validation loss.
- Contributes to preventing overshooting and aids in escaping local minima.

Early Stopping

- Validation loss was continuously monitored.
- Training was halted when no significant improvement was observed, thereby preventing overfitting.

Summary of Improvements & Rationale

Component	**Justification**
Data Augmentation	Enhances generalization; crucial given the limited dataset size.
Transfer Learning	Leverages rich features from ImageNet, reducing training time and improving accuracy.
Custom Classifier	Tailored layers enable adaptation to task-specific patterns.
Focal Loss	Addresses class imbalance and focuses on challenging samples.
PReLU & BatchNorm	Improves learning flexibility and stability.
Early Stopping	Mitigates the risk of overfitting.
Learning Rate Scheduler	Promotes smoother convergence and facilitates better minima discovery.

Evaluation Strategy

Given the imbalanced nature of the dataset and the classification task, the Confusion Matrix is employed to calculate precision, recall, **F1-score,** and accuracy. This allows for a comprehensive understanding of the model's overall performance across categories, identifying both successful and failing classifications. These metrics are crucial for determining which model performs optimally on testing or unseen data.

Resnet 50:

Accuracy: 0.9833

Confusion Matrix:

[[15 0 0]
[1 24 0]
[0 0 20]]

Classification Report:

```
Unset
              precision    recall  f1-score   support

     None       0.94      1.00      0.97        15
     back       1.00      0.96      0.98        25
    front       1.00      1.00      1.00        20

accuracy                           0.98        60
macro avg        0.98      0.99     0.98        60
weighted avg     0.98      0.98     0.98        60
```

Mobilenet v2:

Accuracy: 0.9333

Confusion Matrix:

[[15 0 0]
[2 23 0]
[0 2 18]]

Classification Report:

```
Unset
              precision    recall  f1-score   support

     None       0.88      1.00      0.94        15
     back       0.92      0.92      0.92        25
    front       1.00      0.90      0.95        20

accuracy                            0.93        60
macro avg        0.93      0.94     0.93        60
weighted avg     0.94      0.93     0.93        60
```