VLG Recruitment Challenge ‘24 Report

-**Suhani Rajangaonkar (24119054) (7803060594)**

Introduction

This project focuses on implementing an image classification model using a dataset of images stored in folders. Each folder represents a specific class, and the dataset contains 40 distinct classes. The model is based on a pre-trained EfficientNet architecture fine-tuned for this task.

Development Challenges and Insights

Challenges Faced

1. **Dataset Handling**: Loading the dataset required careful preprocessing, including resizing, normalization, and augmentations for better generalization.
2. **Hardware Constraints**: The availability of a GPU was essential for training. The fallback to CPU would significantly slow down the training process.
3. **Data Augmentation**: Balancing transformations to improve generalization without losing the core characteristics of the images.

Solutions

- Used PyTorch's `ImageFolder` and transformations (`Resize`, `ToTensor`, etc.) to handle dataset preprocessing.
- Leveraged the `torch.device` to ensure the model utilized GPU if available.
- Implemented data augmentation techniques, including random flips, rotations, and crops, to improve model robustness.

Model Architecture

The model is based on the EfficientNet-V2 small architecture pre-trained on ImageNet. Key modifications include:

1. **Input Layer:** Images resized to match the model's expected input dimensions (512x512 during visualization and 224x224 during training).

2.**Output Layer**: The classifier is modified to output predictions for 40 classes, achieved by replacing the last fully connected layer with:

```
nn.Sequential(
   nn.Dropout(0.2),
   nn.Linear(num_features, 40)
```

Explainability Report

The model processes the input images as follows:

1. **Preprocessing:** Images are resized, normalized, and augmented before being fed into the model.
2. **Feature Extraction**: The convolutional layers of EfficientNet extract hierarchical features from images, identifying edges, textures, shapes, and patterns.
3. **Classification:** The modified fully connected layers assign probabilities to each of the 40 classes.

Observations:

- The convolutional layers focus on spatial relationships, enabling the model to identify key patterns in images.
- Augmentations like random crops and flips enhance the model's ability to generalize to unseen data.

Results and Observations

Training and Validation Performance:

The model was trained for 30 epochs using the Lion optimizer, a cross-entropy loss function, and a step learning rate scheduler. Metrics observed during training include:

- **Training Loss and Accuracy:** Gradual improvement across epochs.
- **Validation Loss and Accuracy:** Stabilized after a few epochs, indicating successful generalization.

Visual Data Relationships

The visualizations of the dataset provided insights into:

1. **Class Distribution**: The presence of balanced or imbalanced classes can impact model performance.
2. **Sample Diversity**: The variety in the dataset influences the need for data augmentation.

Sample images for each class were displayed, confirming the correctness of labels and transformations. Additionally, the visualization of the first dataset image highlighted the preprocessing pipeline's correctness.

Conclusion

This project successfully implemented a fine-tuned EfficientNet model for image classification. Key takeaways include:

- Data augmentation and preprocessing are critical for enhancing model robustness.
- EfficientNet's pre-trained features significantly reduce training time and improve accuracy.
- Visualizations provide valuable insights into data quality and preprocessing effectiveness.

Future work could involve:

- Exploring additional architectures or ensemble methods.
- Fine-tuning hyperparameters for further performance improvement.
- Addressing class imbalances using techniques like oversampling or weighted loss functions.