Sardar Vallabhbhai National Institute of Technology

Surat - 395007

Department of Artificial Intelligence

Deep Learning (AI302)

Lab Practical - 4

Problem Statement: Design and Implementation of CNN Architectures for **Imbalanced Image Classification** using Multiple Benchmark Datasets.

DATASETS FOR EXPERIMENTATION

Students must work with at least TWO of the following imbalanced datasets:

#	Dataset	Description	Imbalance Characteristics
1	**Flower Recognition**	5 flower classes: daisy, dandelion, rose, sunflower, tulip	Create imbalance by sampling (e.g., 100:500:200:50:150)
2	**CIFAR-10 (Imbalanced)**	10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck	Create long-tailed distribution with imbalance ratio 100:1
3	**Skin Cancer MNIST (HAM10000)**	7 skin lesion classes including melanoma, nevus, etc.	Naturally imbalanced - melanoma is minority class (~10%)
4	**Chest X-Ray (Pneumonia)**	Binary classification: Normal vs Pneumonia chest X-rays	Imbalanced with ~3:1 ratio (Pneumonia:Normal)
5	**Intel Image Classification**	6 scene classes: buildings, forest, glacier, mountain, sea, street	Create imbalance by undersampling specific classes
6	**PlantVillage Disease**	38 classes of plant diseases across multiple crop types	Naturally imbalanced across disease categories
7	**Blood Cell Images**	4 blood cell types: Eosinophil, Lymphocyte, Monocyte, Neutrophil	Can create imbalance to simulate rare cell detection
8	**Diabetic Retinopathy Detection**	5 severity levels: No DR, Mild, Moderate, Severe, Proliferative	Highly imbalanced - severe cases are rare

Problem Statement 1: Architecture Design Focus

Design a **custom CNN architecture** or modify any standard CNN model to perform **multi-class image classification** on your selected imbalanced datasets.

- Justify the choice of architecture based on dataset characteristics
- Decide the number of layers, filter sizes, kernel sizes, and activation functions
- Implement appropriate regularization techniques (Dropout, Batch Normalization, L2)
- Analyze the impact of dataset imbalance on classification performance

Problem Statement 2: Imbalanced Dataset Handling

The selected datasets are **class-imbalanced**. Design and implement strategies to handle imbalance:

Data-Level Techniques:

- Random Oversampling of minority classes
- Random Undersampling of majority classes
- SMOTE (Synthetic Minority Oversampling Technique) for image features
- Data augmentation specifically for minority classes (rotation, flip, zoom, color jitter)

Algorithm-Level Techniques:

- Class weighting in loss function
- Cost-sensitive learning
- Threshold adjustment during inference

Evaluate how these strategies affect:

- Training convergence and stability
- Class-wise accuracy, especially for minority classes
- Overall model performance (accuracy, macro/micro F1)

Problem Statement 3: Comparative Architecture Analysis

Implement **at least two different CNN architectures** (e.g., EfficientNet vs ResNet, MobileNet vs VGG, or DenseNet vs Inception) on your selected imbalanced datasets.

Compare the models based on:

- Overall Accuracy and Top-k Accuracy
- Precision, Recall, F1-score (class-wise and macro-averaged)
- Confusion Matrix analysis
- Computational cost (FLOPs, parameters, inference time)
- ROC-AUC and PR-AUC curves
- Robustness to class imbalance (G-Mean, Balanced Accuracy)

Problem Statement 4: Loss Function & Optimization Challenge

Investigate the effect of **different loss functions and optimizers** on imbalanced datasets.

Loss Functions to Experiment:

- Cross-Entropy Loss (baseline)
- Weighted Cross-Entropy Loss
- Focal Loss (with different gamma values: 0.5, 1, 2, 5)
- Class-Balanced Loss
- Label Smoothing Cross-Entropy

Optimizers to Experiment:

- SGD (with and without momentum)
- Adam
- AdamW (Adam with weight decay)
- RMSProp

Analyze:

- Convergence speed and training curves
- Minority class recognition improvement
- Overfitting behavior and generalization
- Impact of learning rate scheduling

Problem Statement 5: Feature Representation & Visualization

Extract deep features from the trained CNN model and visualize feature distributions.

Visualization Techniques:

- t-SNE (t-distributed Stochastic Neighbor Embedding)
- PCA (Principal Component Analysis)
- UMAP (Uniform Manifold Approximation and Projection)
- Grad-CAM or Class Activation Maps

Study how imbalance affects:

- Feature clustering quality
- Inter-class separation and intra-class compactness
- Minority class representation in feature space
- Decision boundary visualization

Problem Statement 6: Generalization & Transfer Learning Test

Train the CNN model on one imbalanced dataset and test its **transferability** on another related dataset.

Suggested Transfer Learning Experiments:

- Flower dataset → Intel Image Classification (natural images)
- Skin Cancer → Blood Cell Images (medical imaging domain)
- CIFAR-10 → Flowers or Intel Images
- Pre-trained ImageNet weights → Any selected dataset

Analyze:

- Transferability of learned features
- Performance degradation across domains
- Fine-tuning vs feature extraction approaches
- Impact of dataset complexity and domain shift

Problem Statement 7: Error Analysis & Improvement Proposals

For incorrectly classified samples, perform detailed failure analysis:

- Identify which classes fail most frequently
- Analyze confusion patterns between similar classes
- Correlate errors with class imbalance ratios
- Visualize misclassified samples

- Propose architectural or data-level improvements based on analysis

DATASET LINKS

1. Flower Recognition Dataset:

https://www.kaggle.com/datasets/alxmamaev/flowers-recognition

2. CIFAR-10 Dataset:

https://www.cs.toronto.edu/~kriz/cifar.html

Also available via: tensorflow.keras.datasets.cifar10 or torchvision.datasets.CIFAR10

3. Skin Cancer MNIST (HAM10000):

https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000

4. Chest X-Ray Pneumonia:

https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

5. Intel Image Classification:

https://www.kaggle.com/datasets/puneet6060/intel-image-classification

6. PlantVillage Disease Dataset:

https://www.kaggle.com/datasets/emmarex/plantdisease

7. Blood Cell Images:

https://www.kaggle.com/datasets/paultimothymooney/blood-cells

8. Diabetic Retinopathy Detection:

https://www.kaggle.com/c/diabetic-retinopathy-detection