■ Teeth Image Classification using CNN from Scratch

1. Introduction

The goal of this project was to build a deep learning model from scratch to classify dental images into 7 categories: CaS, CoS, Gum, MC, OC, OLP, OT. Unlike pretrained transfer learning, we constructed our own custom CNN and trained it directly on the dataset.

2. Dataset Organization

The dataset (Teeth_Dataset) was split into:
- Training (7 class folders, imbalanced sample counts)
- Validation (7 class folders)
- Testing (7 class folders, plus extra folders 'out', 'output', 'outputs' which were filtered out).

Training distribution (imbalanced):
CaS: 480, CoS: 450, Gum: 360, MC: 540, OC: 324, OLP: 540, OT: 393.

3. Preprocessing & Augmentation

Before training, we ensured the data was in the right form:
1. Resize all images to 224×224 pixels.
2. Normalization: dataset-specific mean=[0.736, 0.502, 0.478], std=[0.181, 0.204, 0.198].
3. Data augmentation (torchvision.transforms):
- Random horizontal & vertical flips
- Random brightness/contrast adjustments (ColorJitter)
- Random rotations

We visualized class distribution and showed before/after augmentation samples.

4. Model Architecture

We built a custom CNN (SmallCNN) from scratch:
- ConvBlock: Conv2d → BatchNorm → ReLU
- Stacked layers with filters: 32 → 64 → 128 → 256
- Pooling & dropout at each stage
- Adaptive Average Pooling before classifier
- Classifier: Flatten → Linear(256→128) → ReLU → Dropout → Linear(128→7)

Also provided TinyCNN for quick tests, but main experiments used SmallCNN.

5. Handling Imbalance

The dataset was imbalanced. To address this:
- Computed class weights and applied them in the loss function.
- Used WeightedRandomSampler to oversample minority classes and ensure balanced

batches.

6. Training Setup

Training configuration:
- Loss: CrossEntropyLoss (with class weights)
- Optimizer: AdamW (lr=1e-4, weight_decay=1e-4)
- Scheduler: ReduceLROnPlateau
- Epochs: 30, Batch size: 32

We logged training and validation accuracy/loss. Best model checkpoints were saved automatically.

7. Results

Results summary:
- Validation accuracy reached ~66%
- Test accuracy: ~66%
- Classification report: high recall for CoS (>0.90), lower for MC and OLP (~0.40–0.55)
- Confusion matrix revealed detailed error patterns.

8. Interpretability (Grad-CAM)

To interpret predictions, we implemented Grad-CAM:
- Used the last convolutional layer (features.13.conv)
- Generated heatmaps showing regions most important for classification
- Confirmed that the model focuses on relevant areas of the teeth and gums.

9. Conclusion & Future Work

We successfully built a CNN classifier from scratch for dental image classification.

Achievements:
✓ Handled imbalanced dataset
✓ Implemented normalization & augmentation
✓ Designed and trained CNN without pretrained weights
✓ Achieved ~66% test accuracy
✓ Added Grad-CAM interpretability

Future work:
- Deeper architectures (ResNet-style)
- Advanced augmentations (CutMix, MixUp)
- Hyperparameter tuning
- Class imbalance handling with focal loss or SMOTE