Nazif Celik 090200712

Question 1 - Learning from Few Data

I've created a comprehensive solution for ImageNet's datasubset (700 class version) classification challenge with the constraint of using only 50 images per class and building models from scratch . ImageNet's training subset is too poor for classification. In this challenge, the main focus was using only a small bit of data to learn. Here's a breakdown of what each artifact provides:

training data augmented before training process

Augmentation	Description
Grayscale	BGR → GRAY → BGR for consistency
Left-right flip	Horizontal flip
Rotation	±15° random angle
Undersampling	Keeps only a random subset
Zooming	Central crop and resize

The my artifact offers significant improvements in training process:

- More sophisticated CNN architecture with deeper layers and regularization
- Test-time augmentation to improve prediction accuracy
- An option to create an ensemble of models for better performance
- Custom dataset augmentation to increase the number of training samples
- K-fold (5-fold) cross-validation to maximize use of limited data
- Image augmentation performed during training for each fold
- Performance visualization across folds
- Custom image loading and processing with OpenCV for efficiency

Key Strategies Used

1. **Data augmentation**: I've incorporated multiple techniques to artificially expand your limited dataset:
    a. Standard transformations (rotation, flip, shift)
    b. More advanced transformations (grayscale, zoom, brightness/contrast)
    c. Different implementations showing both on-the-fly and pre-computed augmentation

2. **Model architecture**: Built custom CNNs with:
    a. Multiple convolutional blocks with increasing filter sizes
    b. Batch normalization for training stability
    c. Dropout for regularization
    d. L2 regularization to prevent overfitting on small dataset

3. **Training optimization**:
    a. Early stopping to prevent overfitting
    b. Learning rate scheduling
    c. Class weighting to handle potential imbalance

Question 2 - Grounded SAM

Grounded SAM2 can flexibly detect and segment almost any object or region described in natural language.

I performed segmentation with Grounded SAM2 on a data subset in the Imagenet dataset where I sampled one photo from each class and 700 images with encrypted names (such as n01443537, n01484850, n01491361, n01494475, n01496331...).

Grounded SAM2 integrates Grounding DINO3, an open-set object detector, with the Segment Anything Model (SAM)4 to enable open-vocabulary detection and segmentation. SAM2.1-hiera-large model was used for image segmentation process in this challange.

ALGORITHMIC LOGIC

my own algorithmic logic for classify the examples in the dataset:

I compared the csv file to measure how successful the segmentation process was. I gave 700 natural language class names as TEXT_PROMPT and it matched them with the encrypted class names. Encrypted names that correspond natural language class names were founded in:

[https://gist.github.com/aaronpolhamus/964a4411c0906315deb9f4a3723aac57]

1000 class into 700 class seperated by using a few CSV file manipulation process

In a loop, it wrote the name of the object identified in each photograph, along with the name of the photograph, line by line into the csv file.

Then I gave another csv file. In the csv file I gave, the first column is the encrypted class names, the second column is the real class names. With this, I measured how accurate my classification was.

Finally, I saved the final csv file with the encrypted class names in the first column, the real class names with natural language in the second column, and the predicted class names in the third column.