Project Progress Report: Multi-Label Anime Image Tag Classification

1. Problem Statement

The project aims to develop a deep learning model capable of automatically assigning relevant tags to anime-style illustrations. By using a multi-label classification approach with a ResNet50 architecture, the system learns to identify multiple characteristics, art styles, and content elements present in each image. This automation can significantly improve image organization and searchability in large anime art collections.

2. Data Preprocessing

I am working with the Danbooru2021 dataset, which contains anime-style illustrations with associated metadata and tags. The preprocessing pipeline yielded the following statistics:

- Total initial metadata entries: 2,963,928 images
- Processed samples (1:4 sampling rate): 740,982 images
- Initial unique tags: 206,412
- Filtered tags (min. occurrence 100): 8,266
- Final processed dataset size: 84,430 images

Preprocessing Methods:

1. **Data Sampling**: Implemented 1:4 sampling rate to manage dataset size while maintaining representation, and reduce training time for this initial version
2. **Tag Filtering**:
    a. Applied minimum occurrence threshold (100) to remove rare tags
    b. Reduced tag count from 206,412 to 8,266 tags
3. **Image Processing**:
    a. Verified image dimensions (512x512)
4. **Data Splits**:
    a. Train/Validation/Test split: 64/16/20

3. Machine Learning Model

a. Framework and Implementation

- **Primary Framework**: PyTorch
- **Model Architecture**: ResNet50 (pretrained on ImageNet)
- Modified for multi-label classification (8,266 outputs)
- Additional tools:
    - torchvision for image transformations and model architecture
    - sklearn for metrics and data splitting
    - CUDA for GPU acceleration
    - Automatic Mixed Precision (AMP) for efficient training

b. Training Decisions

- **Data Splits**: 64/16/20 (train/val/test)
- **Batch Size**: 16 (optimized for memory constraints)
- **Learning Rate**: 0.001
- **Optimization**:
    - Adam optimizer
    - Mixed precision training for performance
- **Regularization**:
    - Data augmentation (random horizontal flips, rotations)

c. Validation Methods

- Continuous monitoring of training/validation metrics
- Metrics tracked:
    - Loss
    - F1 score
    - Precision
    - Recall

d. Implementation Challenges

1. **Memory Management**:
    a. Initial GPU memory issues with larger batch sizes

b. Resolved by:
    i. Reducing batch size to 16
    ii. Implementing mixed precision training
    iii. Optimizing data loading
2. **Training Time**:
    a. Large dataset and model size led to long training times
    b. Solutions:
        i. Implemented checkpointing for training resumption
        ii. Added progress bars for better monitoring
        iii. Utilized mixed precision training for speed improvement

4. Preliminary Results

Our model's performance metrics show interesting patterns in its tag prediction capabilities:

Validation Set Performance:

- F1 Score: 0.3467
- Precision: 0.6832
- Recall: 0.2323

Test Set Performance:

- F1 Score: 0.3471
- Precision: 0.6820
- Recall: 0.2328

Analysis:

1. **High Precision, Low Recall Trade-off**:
    a. The model achieves strong precision (~68%) but relatively low recall (~23%)
    b. This indicates that when the model predicts a tag, it's usually correct, but it misses many tags that should be present
    c. This conservative prediction pattern is often preferable for tag recommendation systems
2. **Consistent Performance**:
    a. Very similar metrics between validation and test sets (difference < 0.1%)

    b. This consistency suggests the model generalizes well to unseen data
3. **Overall Performance**:
    a. F1 score of ~0.35 reflects the challenging nature of multi-label classification with 8,266 possible tags
    b. The model shows promising results considering the complexity of the task and large tag space

5. Next Steps

- **Reprocess the dataset to use more tags**
- **Retrain the model using the complete dataset**