Mini Project 1: Dataset Curation & Preprocessing

Low-Light Image Enhancement Dataset Preparation

Benjamin Tran
CS 5190 Computer Vision

November 4th, 2025

1 Objective

This mini project focuses on collecting, preprocessing, and organizing low-light image datasets to support the final group project on Hybrid Low-Light Image Enhancement. The goal was to create a ready-to-use dataset with proper train/test splits, preprocessing (resizing and normalization), and data augmentation (synthetic darkening) for both classical and machine learning approaches.

2 Dataset Collection

2.1 Dataset Source

The **LOL (Low-Light) Dataset** was selected as the primary dataset for this project. The dataset structure includes:

- **Training Set (`our485/`)**: 485 image pairs (low-light and corresponding high-light ground truth)
- **Evaluation Set (`eval15/`)**: 15 image pairs for final evaluation

The dataset contains real-world low-light images captured under various lighting conditions, making it suitable for testing enhancement algorithms.

2.2 Dataset Organization

The dataset was organized with the following structure:

```
lol_dataset/
          our485/
                low/   (485 low-light images)
                high/  (485 ground truth images)
          eval15/
              low/   (15 low-light images)
              high/  (15 ground truth images)
```

3 Preprocessing Pipeline

A comprehensive preprocessing pipeline was implemented using Python with OpenCV, NumPy, and scikit-learn. The pipeline consists of three main stages:

3.1 Image Resizing

All images were resized to a uniform size of **512×512 pixels** using bilinear interpolation. This standardization ensures:

- Consistent input dimensions for all models
- Reduced computational requirements
- Maintained aspect ratio through proper interpolation

3.2 Normalization

Images were normalized using **standard normalization** (zero mean, unit variance):

- Formula: $(x - \mu)/\sigma$
- Results scaled to [0, 255] range for compatibility with standard image processing libraries
- This normalization improves model convergence and stabilizes training

3.3 Data Augmentation

Synthetic darkening augmentation was applied to increase dataset diversity and robustness. The augmentation process:

- Creates **9 additional variations** per original image
- Uses multiple darkening factors (0.2, 0.3, 0.4)
- Applies gamma correction with values (1.8, 2.0, 2.2)
- Adds controlled noise to simulate real-world low-light conditions

This augmentation strategy significantly expands the training dataset while maintaining realistic low-light characteristics.

3.4 Train/Test Split

The dataset was split using an **80/20 ratio**:

- **Training set**: 80% of images (with augmentation applied)
- **Validation set**: 20% of images (no augmentation, for unbiased evaluation)
- **Test set**: 15 images from `eval15` (held-out evaluation set)

The split was performed using scikit-learn's `train_test_split` with a fixed random seed (42) for reproducibility.

4 Results

4.1 Dataset Statistics

Final Processed Dataset:

- **Original Training Images**: 776 images (from `our485/low`)
- **Processed Training Images**: 7,760 images (with augmentation)

  - Each original image generates 10 versions (1 original + 9 augmented)

- **Validation Images**: 194 images (no augmentation)
- **Test Images**: 15 images (from `eval15`)
- **Total Processed Images**: 7,969 images

Image Specifications:

- Resolution: 512 × 512 pixels
- Format: JPEG
- Color space: RGB (BGR in OpenCV)
- Normalization: Standard normalization applied

4.2 Output Structure

The processed dataset is organized as follows:

```
processed_dataset/
          train/              # 7,760 training images
                image1.jpg
                image1_aug1.jpg
                image1_aug2.jpg
                ...
          test/               # 209 test images
                validation images (194)
                eval_*.png (15)
          visualizations/     # Preprocessing examples
                preprocessing_examples_train.png
                preprocessing_examples_test.png
          dataset_stats.json  # Dataset statistics
```

4.3 Visual Deliverables

The following visualizations were generated:

1. **Preprocessing Examples**: Side-by-side comparisons showing original, resized, normalized, and synthetically darkened images
2. **Dataset Statistics**: Statistical analysis including:
   - Brightness distribution histograms
   - RGB channel statistics
   - Image size distribution
   - Train/test split visualization

5 Implementation Details

5.1 Tools and Libraries

- **Python 3.11**
- **OpenCV**: Image processing and manipulation

- **NumPy**: Numerical operations
- **scikit-learn**: Train/test splitting
- **Matplotlib**: Visualization
- **tqdm**: Progress tracking

5.2 Key Features Implemented

1. Automated dataset discovery (recursive image search)
2. Configurable preprocessing parameters (size, normalization method)
3. Flexible augmentation strategies
4. Progress tracking and error handling
5. Comprehensive visualization generation

6 Deliverables

The following deliverables have been completed:

Preprocessed Dataset: Ready-to-use dataset with train/test splits

Visual Examples: Preprocessing step demonstrations

Statistics Report: JSON file with dataset metrics

Visualization Tools: Scripts for dataset analysis

Documentation: Code documentation and usage guides

7 Future Integration

This preprocessed dataset is now ready for:

1. **Classical Methods**: Histogram equalization and Retinex-based enhancement
2. **Machine Learning Approaches**: Neural network training and evaluation
3. **Hybrid Pipeline**: Combining classical and ML methods for comparison

The consistent preprocessing ensures fair comparison across different enhancement approaches in the final project.

8 Conclusion

The dataset curation and preprocessing mini project has been successfully completed. The LOL dataset has been processed, augmented, and split into appropriate train/test sets. The preprocessing pipeline includes resizing, normalization, and synthetic darkening augmentation, resulting in a dataset of 7,969 processed images ready for the final group project on Hybrid Low-Light Image Enhancement.

The dataset is properly organized, documented, and includes visual examples demonstrating each preprocessing step. All code is modular and reusable, allowing for easy extension and modification as needed for the final project implementation.

Project Files:

- `dataset_preprocessing.py`: Main preprocessing script
- `process_lol_dataset.py`: LOL-specific processing script
- `visualize_dataset.py`: Statistical visualization tool
- `README.md`: Project documentation
- `requirements.txt`: Python dependencies
- `Mini_Project_1_Report.tex`: This report