Unique Objects Analysis Workflow

This document explains how the unique objects feature extraction and analysis pipeline works, step by step.

Overview

The pipeline consists of two main scripts:

1. `unique-objects-feature-extraction.py`: Extracts visual features from images
2. `unique-objects-analysis.py`: Analyzes these features to select and cluster diverse images

Part 1: Feature Extraction (`unique-objects-feature-extraction.py`)

Setup Phase

1. Initializes AlexNet pre-trained model
    - Uses the model in evaluation mode
    - Sets up image preprocessing transformations (resize, normalize)
2. Creates FeatureExtractor class
    - Hooks into AlexNet's last maxpool layer (layer 12)
    - Captures feature representations of images

Extraction Process

1. Scans the `ObjectsAll/OBJECTSALL` directory
    - Identifies all images with extensions: .jpg, .jpeg, .png, .thl
    - Limits to first 2400 images (configurable via max_images)
2. For each image:
    - Loads and preprocesses the image
    - Runs it through AlexNet
    - Captures the feature vector
    - Saves feature vector as .pt file
    - Maps original filename to feature filename
3. Creates mapping file
    - Saves feature_mapping.json containing:

```
{
    "image_base_name": {
        "image_file": "original_image.thl",
        "feature_file": "feature_vector.pt"
    }
}
```

Part 2: Analysis (`unique-objects-analysis.py`)

1. Loading Phase

- Loads feature vectors from .pt files
- Loads feature_mapping.json
- Flattens feature vectors for analysis

2. Diverse Image Selection

Uses Max-Min Distance Selection algorithm:

1. Starts with random image
2. Iteratively selects images that are most different from already selected ones
3. Continues until 100 images are selected
4. Uses cosine similarity as distance metric

3. Statistical Validation

1. Performs Monte Carlo simulation
    - Creates 1000 random selections of 100 images
    - Compares average distances in random vs. diverse selections
    - Calculates p-value to validate selection method

4. Clustering

1. Clusters 100 selected images into 20 groups of 5
    - Uses Max-Min approach within each cluster
    - Ensures intra-cluster diversity

5. Analysis Visualization

Creates three types of visualizations:

1. Main dendrogram

    - Shows relationships between all 100 selected images
    - Saved as 'main_dendrogram.png'

2. Twenty-way clustering visualization

    - Shows how images split into 20 clusters
    - Saved as 'twenty_way_clustering.png'

3. Per-cluster visualizations

    - Individual dendrograms for each cluster of 5
    - Distance matrices and heatmaps
    - Saved in respective cluster folders

6. Output Organization

1. Creates directory structure:

```
src/selected_clusters/
├── main_dendrogram.png
├── twenty_way_clustering.png
├── 1/
│   ├── cluster_dendrogram.png
│   ├── distance_matrix.csv
│   ├── distance_heatmap.png
│   └── [5 selected images]
├── 2/
│   └── [same structure]
└── [folders 3-20]
```

2. Saves selected image list
    - Creates selected_unique_objects.txt
    - Lists all 100 selected image identifiers

Distance Metrics and Clustering Details

Cosine Similarity

- Measures similarity between feature vectors
- Range: [-1, 1]
    - 1: identical features
    - 0: orthogonal features
    - -1: opposite features

Max-Min Selection Strategy

1. Initial selection:
    - Randomly selects first image
2. Subsequent selections:
    - Computes similarities to all selected images
    - Finds minimum similarity for each candidate
    - Selects candidate with lowest maximum similarity
3. Benefits:
    - Ensures maximum diversity
    - Avoids redundant selections
    - Provides good coverage of feature space

Ward's Hierarchical Clustering

- Used for creating dendrograms
- Minimizes variance within clusters
- Provides interpretable hierarchical structure

File Formats

Feature Files (.pt)

- PyTorch tensor format
- Contains flattened feature vectors
- Extracted from AlexNet's last maxpool layer

Distance Matrix (CSV)

- Pairwise cosine similarities
- Row/column headers are image identifiers
- Values are formatted to 3 decimal places

Visualization Files

1. Dendrograms (.png)
    - Show hierarchical relationships
    - Include image labels
    - Use consistent color scheme
2. Heatmaps (.png)
    - Visualize pairwise distances
    - Include numerical annotations
    - Use seaborn's coolwarm colormap

Usage Notes

1. Run feature extraction first:
```
python src/unique-objects-feature-extraction.py
```
2. Then run analysis:
```
python src/unique-objects-analysis.py
```
3. Check output:
    - Verify all clusters have 5 images
    - Review dendrograms for logical groupings
    - Examine distance matrices for diversity

Error Handling

The scripts include robust error handling for:

- Missing directories
- File not found errors
- Image loading failures
- Feature extraction errors
- Invalid file formats

Each error is logged with specific information to aid debugging.