

# Codebase Overview

---

## Project Structure

```
remote-viewing-experiment/
├── src/
│   ├── config/                # Configuration files
│   │   └── experiment_config.py
│   ├── unique_objects_analysis_global.py # Main analysis code
│   ├── selection_process.py    # Image selection algorithms
│   └── cluster_diversity.py    # Cluster analysis tools
├── scripts/
│   ├── run_analysis.py        # Main execution script
│   └── test_real_data.py      # Testing with real images
├── documentation/
│   ├── analysis_outputs.md    # Output interpretation guide
│   └── codebase_overview.md   # This file
└── experiments/              # Experiment results
```

## Core Components

### 1. Analysis Pipeline (`unique_objects_analysis_global.py`)

- **Feature Analysis**

- `calculate_distance_metrics()`: Computes distance metrics between images
- `analyze_cluster_temporal_quality()`: Tracks cluster quality over time
- `find_closest_pairs()`: Identifies most similar image pairs

- **Visualization Functions**

- `plot_distance_matrix()`: Heatmap of image distances
- `plot_cluster_analysis()`: Cluster size and distance visualizations
- `plot_minimum_distances()`: Distribution of minimum distances
- `plot_cluster_temporal_analysis()`: Temporal quality trends

- **Statistical Analysis**

- Monte Carlo simulations
- Statistical significance testing
- Effect size calculations

### 2. Image Selection (`selection_process.py`)

- Selects diverse representative images
- Handles interpretability filtering
- Manages initial and final selection sizes

### 3. Cluster Analysis ([cluster\\_diversity.py](#))

- Analyzes diversity within image groups
- Computes inter/intra-cluster metrics
- Validates clustering quality

## Workflow

#### 1. Data Preparation

```
# Run feature extraction and initial analysis
python scripts/run_analysis.py --start_from features
```

- Loads images from ObjectsAll/OBJECTSALL/
- Extracts ResNet-50 features
- Creates initial clusters

#### 2. Analysis Execution

```
# Run analysis on existing clusters
python scripts/run_analysis.py --start_from analysis
```

- Performs Monte Carlo simulations
- Generates visualizations
- Computes statistical metrics

#### 3. Output Generation

- Creates experiment directory with timestamp
- Saves analysis results and visualizations
- Generates comprehensive metrics

## Key Visualizations

#### 1. Distance Matrix ([distance\\_matrix.png](#))

- Shows pairwise distances between images
- Uses custom colormap for better visualization
- Randomly samples if too many images

#### 2. Cluster Analysis ([cluster\\_analysis/](#))

- [cluster\\_sizes.png](#): Distribution of cluster sizes
- [intercluster\\_distances.png](#): Distances between cluster centroids
- [intracluster\\_distances.png](#): Within-cluster distance distributions

#### 3. Temporal Analysis ([temporal\\_quality.png](#))

- Shows how cluster quality changes over time
- Includes confidence intervals
- Highlights global average

## Configuration Options

Edit `src/config/experiment_config.py`:

```
PIPELINE_CONFIG = {  
    "start_from": "features", # Options: features, clusters, analysis  
    "save_baseline": False,  
    "use_baseline": False,  
    "baseline_dir": "baseline_clusters",  
    "baseline_name": "baseline_clusters_v1"  
}
```

## Testing

### 1. With Synthetic Data

```
python scripts/test_analysis.py
```

- Creates test data
- Validates analysis pipeline
- Checks visualization outputs

### 2. With Real Data

```
python scripts/test_real_data.py
```

- Uses actual images
- Full pipeline validation
- Memory-optimized processing

## Cleanup Tasks

### 1. Remove Duplicate Files

- Delete `src/unique-objects-analysis.py`
- Keep `unique_objects_analysis_global.py`

### 2. Update .gitignore

- Exclude large data files
- Ignore experiment outputs

- Skip model checkpoints

### 3. **Organize Outputs**

- Maintain consistent directory structure
- Clean up old experiment results
- Archive important baselines

## Future Improvements

### 1. **Code Organization**

- Split analysis functions into separate modules
- Create proper test suite
- Add type hints

### 2. **Features**

- Implement SBERT integration
- Add more visualization options
- Enhance cluster quality metrics

### 3. **Documentation**

- Add function docstrings
- Create API documentation
- Include usage examples