AutoCull End-User Documentation

Group 11

October 14, 2025

Contents

1 Introduction

AutoCull is an intelligent image management application that automatically identifies and removes near-duplicate or low-quality photos from a given collection. The program leverages AI/ML techniques to both streamline large-scale photo curation, and improve the user's photographic abilities.

This document serves as a comprehensive guide for end users, covering installation, usage, configuration, and troubleshooting.

2 System Requirements

- Operating System: Windows 10 or later / macOS / Linux
- Python 3.10 or higher
- PostgreSQL 14 or higher
- Recommended RAM: 8 GB+
- Disk space: Minimum 1 GB free

3 Installation Guide

3.1 Step 1: Clone or Download the Repository

```
git clone https://github.com/Manwithacape/COSC345---Group_11.git
cd COSC345---Group_11
```

3.2 Step 2: Create a Virtual Environment

```
python -m venv venv
source venv/bin/activate        # macOS/Linux
venv\Scripts\activate           # Windows
```

3.3 Step 3: Install Dependencies

```
pip install -r requirements.txt
```

3.4 Step 4: Set Up PostgreSQL Database

AutoCull uses PostgreSQL to store image metadata, similarity metrics, and user preferences.

1. Install PostgreSQL 14 or higher from https://www.postgresql.org/download/.
2. During installation, note your chosen **username**, **password**, and **port number** (default is 5432).

3. Open the PostgreSQL interactive terminal:

```
psql -U postgres
```

4. Create a new database and user for AutoCull:

```
CREATE DATABASE autocull_db;
CREATE USER autocull_user WITH PASSWORD 'yourpassword';
GRANT ALL PRIVILEGES ON DATABASE autocull_db TO autocull_user;
```

Example contents:

```
{
    "dbname": "autocull_db",
    "user": "autocull_user",
    "password": "yourpassword",
    "host": "localhost",
    "port": 5432
}
```

5. Initialize the schema using file contents from `schema.db`: You should see a confirmation message indicating that tables were created successfully.

3.5 Step 5: Launch AutoCull

```
python app.py
```

If the program launches successfully, you should see the AutoCull interface.

4 Configuration

AutoCull requires a database configuration file named `db_config.json` or environment variables defined in a `.env` file in the project root.

```
{
    "dbname": "autocull_db",
    "user": "autocull_user",
    "password": "yourpassword",
    "host": "localhost",
    "port": 5432
}
```

Ensure this file is present before running the application.

5 Dependencies

AutoCull relies on several Python libraries for image processing, database interaction, and machine learning. All dependencies can be installed automatically using the `requirements.txt` file included with the project.

- **ImageHash** – Perceptual hashing for duplicate detection
- **numpy** – Numerical operations and data manipulation

- **opencv_python** – Image analysis and computer vision tools
- **piexif** – EXIF metadata reading and writing
- **Pillow** – Image loading, display, and format conversion
- **python-dotenv** – Environment variable and configuration management
- **scikit_learn** – Machine learning and clustering algorithms
- **scikit-image** – Advanced image processing utilities
- **ttkbootstrap** – Modern themed interface for Tkinter
- **rawpy** – RAW image file decoding
- **psycopg2-binary** – PostgreSQL database connectivity
- **google-genai** – Integration for Google Gemini-based text generation
- **torch** – Core PyTorch deep learning framework
- **torchvision** – Vision utilities for PyTorch models
- **transformers** – Access to pretrained deep learning models
- **face_recognition** – Face detection and recognition functionality
- **exifread** – Lightweight EXIF metadata extraction
- **CLIP (OpenAI)** – Image-text embedding model for similarity and semantic scoring

To install all dependencies:

```
pip install -r requirements.txt
```

6 Key Features

- Intelligent duplicate detection using perceptual hashing
- Photo scoring based on exposure, focus, and composition
- EXIF metadata viewer
- Collection management interface
- Integrated PostgreSQL backend for image data
- Lightweight, dark-themed Tkinter GUI

7 Using AutoCull

7.1 Basic Workflow

1. Launch AutoCull.
2. Import images by selecting "Import" from the sidebar.
3. AutoCull scans for near duplicates and low-quality images.
4. Review detected clusters of similar images.
5. Confirm deletion or move duplicates to a separate folder.

7.2 GUI

The GUI provides options for:

- Photo importing
- Viewing detected duplicates
- Marking photos for deletion
- Previewing detected duplicates
- Batch delete/move operations
- Viewing collections
- Creating collections
- Viewing exif data
- Viewing image metrics

8 Troubleshooting

- **Program doesn't start:** Ensure Python 3.10+ is installed and dependencies are correctly installed.
- **Database connection error:** Verify that PostgreSQL is running and credentials in `db_config.json` are correct.
- **No duplicates found:** Try lowering the `threshold` value.
- **Slow performance:** Use smaller batches or disable subfolder recursion.
- **Permission errors:** Run as administrator or check write permissions on output directories.

9 Uninstallation

To remove AutoCull, delete the application folder and its virtual environment. If desired, remove the PostgreSQL database using:

```
DROP DATABASE autocull_db;
DROP USER autocull_user;
```

10 About

This application was developed as part of a university project for COSC345. For more information, visit the GitHub repository: https://github.com/Manwithacape/COSC345---Group_11

Please do not attempt to contact anyone regarding bugs, issues, or feature requests. These will likely not be addressed.

11 Credits and License

Developed by **Group 11, COSC345 2025 S2** as part of the AutoCull project.