🎨

Aesthify - Setup and Operation

📚 Project Overview

Aesthify is an AI-driven system that evaluates the aesthetic quality of interior layouts through object detection and design theory metrics.

It combines:

- YOLOv8 object detection for identifying key elements
- Roboflow integration for enhanced domain-specific models
- Quantitative scoring based on design principles like Balance, Proportion, Symmetry, Simplicity, Unity, Contrast, and Harmony

Applications:

- Academic research on aesthetic perception
- Survey-based comparative studies (human vs AI)
- Adaptable to domains like bus cabins, office spaces, and product displays
- Fully operational offline and online

🖥️ System Requirements

- Python 3.10 or higher installed
- pip (Python package manager)
- (Recommended) Virtual Environment setup (`venv`)
- Access to YOLO model file (`models/yolov8n.pt`)
- (Optional) Roboflow API Key if using multi-model detection
- A working webcam if using capture mode

🛠️ Setup Instructions

1. Clone the Repository

- Open your terminal (Command Prompt, Terminal, etc.)
- Run:

```
git clone https://github.com/yourusername/aesthify.git
cd aesthify
```

2. Create a Virtual Environment

- For Windows:
  - Run: `python -m venv venv`
  - Then activate: `venv\Scripts\activate`
- For Mac/Linux:
  - Run: `python3 -m venv venv`
  - Then activate: `source venv/bin/activate`

3. Install Required Packages

- After activating the virtual environment, run:

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

This will install Flask, OpenCV, scikit-learn, and all necessary packages.

⚙️ Configuring Environment Variables

- In the root folder, create a file named `.env`
- Add the following line inside:

```
ROBOFLOW_API_KEY=your_optional_key_here
```

- If you don't plan to use Roboflow models, you can leave it empty.

🚀 Running the Application Locally

- Activate your virtual environment
- Then run:

```
python app.py
```

- Open your browser and go to: http://127.0.0.1:5000/

You should see the Aesthify web app running.

📸 Using Aesthify

- Upload an image using the **Upload Image** button
- Or use your **Webcam** by clicking **Start Camera** and then **Capture**
- Click **Evaluate** to run the aesthetic analysis
- View:
  - Aesthetic scores (Balance, Symmetry, Simplicity, etc.)
  - Labeled image with detected objects

📚 Survey Mode Setup (Optional)

1. Prepare Evaluation Results

- Run a few images through the normal evaluation process.
- Open `evaluation_results_dump.xlsx` (generated automatically).
- Copy all aesthetic scores from it.
- Paste them into `evaluation_results.xlsx`, linking correctly to Image IDs.

2. Organize Survey Images

- Save all survey images into a folder named `survey_images`
- Filenames must match the Image IDs used in your Google Form and evaluations.

3. Update Image Mapping

- Open `survey_analysis.py`
- Update the `img_map` dictionary so each Survey ID maps to the correct new image name.

Example:

```
img_map = {
    'Q1': 'image1.jpg',
    'Q2': 'image2.jpg',
    ...
}
```

4. Import Survey Results

- Collect your Google Form responses.
- Download the results as an Excel file.
- Save it as `survey_results.xlsx` in the project directory.

Make sure it contains:

- User IDs
- Image ratings
- Style preference answers
- Emotional tags

(Format should match what `survey_analysis.py` expects.)

Example Format:

📊 Running Survey Analysis

- After everything is set, simply run:

```
python survey_analysis.py
```

This will:

- Analyze user preferences
- Perform clustering
- Learn optimal weightings
- Generate graphs and evaluation plots

🎯 Important Notes

- For offline-only use, Roboflow API can be disabled (only YOLOv8 model used).
- Keep evaluation images and survey images correctly named and organized.
- If camera capture is not available (e.g., on desktop PCs without webcam), only upload mode can be used.
- Make sure `.env`, YOLO model, and images are properly in place before running.

🛡️ Few Shortcuts/Tips

Developer Task	Shortcut / Quick Tip
Restart server after code change	Stop terminal → Run `python app.py` again
Check detected labels	Printed automatically in console or view inside returned JSON
Modify scoring weights	Edit weights inside `survey_analysis.py` if custom survey weight learning
Update models	Replace YOLO model file inside `/models/`
Customize evaluation pipeline	Modify `main_pipeline.py` if adding more aesthetic metrics