

CS5330: Pattern Recognition and Computer Vision

Remo Project Milestone 1- Proof of Concept

Project Overview

Project chosen – Project 1

The goal of this project is to develop a system that can scan book covers and extract relevant metadata for database entry. The system should capture both front and back covers and extract key bibliographic information including title, creators, copyright date, summary, series information, genres, form, format, ISBN, page count, and book type.

Authors:

- Karthik Varunn Saseendran
- Paul Adaimi

Implementation Components

1. Image Capture System (`final_project.py`)

Components:

- OpenCV integration with phone camera via IP connection.
- Real-time frame processing and book detection.
- Rectangle overlay for guided image capture.
- Automated frame saving when book is detected.
- Dynamic rectangle sizing based on frame dimensions provides a clear visual guidance for user.
- Text overlay for instructions.

The processing steps are:

- Grayscale Conversion -Reduces processing complexity and improves edge detection reliability
- Gaussian Blur - Kernel size: 5x5. Reduces image noise and improves contour detection
- Edge Detection - Uses Canny algorithm with threshold values: 50, 150
- Detection Criteria - Four-sided contour shape detection. Size and position validation using the rectangle positions(margin of error: 5 pixels outside box, 70 pixels inside box)

Implementation Details:

Key functions:

```
detect_book_in_rectangle(frame, x1, y1, x2, y2)  # Detects book presence
crop_to_rectangle(frame, x1, y1, x2, y2)        # Crops frame to region
```

The system uses contour detection to identify when a book is properly positioned within the defined rectangle. A *countdown timer* ensures stable image capture, saving frames as frame1.jpg (front cover) and frame2.jpg (back cover) in a folder named scanned.

2. Text Extraction and Processing (test.ipynb)

Components:

- EasyOCR for text extraction
- OpenAI ChatGPT API for metadata generation
- Google Books API for supplementary data

Implementation Flow:

1. Text Extraction:

```
reader = easyocr.Reader(['en'])
results = reader.readtext('frame1.jpg')
```

2. Text Processing:

- Concatenation of extracted text segments
- Formation of structured query for APIs

3. Metadata Generation:

- ChatGPT API call for comprehensive metadata
- JSON output formatting with required fields

Results Analysis

What's Working

1. Image Capture System:

- Successful phone camera integration
- Accurate book detection within defined regions
- Reliable frame saving mechanism

2. Text Extraction:

- EasyOCR effectively extracts visible text
- Basic bibliographic information retrieval

3. Metadata Generation:

- ChatGPT API successfully generates structured JSON output
- Sometimes the google API returns the right ISBN.

Current Limitations

1. Image Quality Dependencies:

- Results heavily dependent on lighting conditions.
- Text extraction accuracy varies with cover design.
- ISBN extraction is not that effective and relies solely on API call for now.

2. **API Integration:**
 - Google Books API matches not always precise
 - Potential for metadata discrepancies

Technical Challenges

1. **OCR Reliability:**
 - Variable text recognition accuracy
 - Font style and cover design impact
2. **Data Validation:**
 - Ensuring metadata completeness
 - Verifying information accuracy

Future Improvements

1. **Convert the python notebook(s) into functioning python modules (.py) files for a more streamlined and production level solution.**
2. **Image Processing and Text extraction:**
 - Implement image enhancement preprocessing.
 - Add dynamic lighting adjustment.
 - Combine the above two to get better images for easier ISBN extraction.
 - Try PaddleOCR, KerasOCR and Googlevision.
3. **API Integration:**
 - Add multiple metadata source cross-referencing.
 - Implement confidence scoring for matches.
4. **Database Integration:**
 - Develop database schema for insertion using json information if detected book is not already present.
5. **User Interface: (if possible)**
 - Add metadata review/edit capabilities

Conclusion

The current implementation demonstrates a functional proof of concept for automated book metadata extraction. While core functionalities are working, there are clear areas for improvement in reliability and accuracy. The combination of computer vision, OCR, and AI-powered metadata generation shows promise for creating a robust book cataloging system.

Next Steps

1. Implement image preprocessing for improved OCR
2. Develop metadata validation framework
3. Create database integration layer
4. Build user interface for metadata review
5. Add support for multiple metadata sources if possible