🧾 Receptix - Smart Receipt Scanner

A powerful Python application that uses **OCR (Optical Character Recognition)** to scan receipt images, extract expense data, and automatically categorize expenses into organized groups.

✨ Features

- 🖼️ **Image Upload**: Support for PNG, JPG, JPEG, GIF, BMP, and TIFF formats
- 🔍 **Advanced OCR**: Uses Tesseract OCR with image preprocessing for accurate text extraction
- 🏷️ **Smart Categorization**: Automatically categorizes expenses into:
  - Food & Dining
  - Transportation
  - Shopping & Retail
  - Healthcare & Pharmacy
  - Utilities & Bills
  - Entertainment & Recreation
  - Personal Care & Beauty
  - Office & Business
  - Education
  - Miscellaneous
- 📊 **Expense Analysis**: Real-time breakdown of expenses by category
- 💾 **Report Generation**: Save detailed reports in text format
- 📝 **Comprehensive Logging**: Track all scanning activities
- 🖥️ **User-Friendly GUI**: Clean, modern Tkinter interface

🚀 Installation

Prerequisites

1. **Python 3.7+** installed on your system
2. **Tesseract OCR** engine installed:

Windows

bash

```
# Download and install from: https://github.com/UB-Mannheim/tesseract/wiki
# Default installation path: C:\Program Files\Tesseract-OCR\tesseract.exe
```

macOS

bash

```
brew install tesseract
```

Ubuntu/Debian

bash

```
sudo apt update
sudo apt install tesseract-ocr
```

CentOS/RHEL

bash

```
sudo yum install tesseract
```

Python Dependencies

1. **Clone or download** the project files
2. **Install required packages**:

bash

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

Project Structure

```
receptix_gui/
├── main_gui.py          # Main GUI application
├── ocr_utils.py         # OCR processing functions
├── categorizer.py       # Expense categorization logic
├── requirements.txt     # Python dependencies
├── README.md           # This documentation
├── assets/             # Sample images (optional)
│   └── sample_receipt.jpg
├── logs/               # Application logs (auto-created)
│   └── receipts.log
└── reports/            # Generated reports (auto-created)
    └── receipt_report_YYYYMMDD_HHMMSS.txt
```

🎯 Usage

Starting the Application

```bash
python main_gui.py
```

Step-by-Step Guide

1. **Launch Receptix** by running `main_gui.py`
2. **Upload Image**: Click "📁 Browse Image" to select your receipt image
3. **Scan & Analyze**: Click "🔍 Scan & Categorize" to process the receipt
4. **Review Results**: Check the extracted text and expense breakdown
5. **Save Report** (Optional): Click "💾 Save Report" to export results
6. **Clear Data**: Use "🗑️ Clear" to reset for a new receipt

Supported Image Formats

- PNG (.png)
- JPEG (.jpg, .jpeg)
- GIF (.gif)
- BMP (.bmp)
- TIFF (.tiff)

🛠️ Configuration

Tesseract Path Configuration

If Tesseract is not in your system PATH, update the path in `ocr_utils.py`:

```python
# Windows example
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

# Linux/macOS example
pytesseract.pytesseract.tesseract_cmd = '/usr/bin/tesseract'
```

Custom Categories

You can add custom keywords to existing categories or create new ones by modifying the `CATEGORY_KEYWORDS` dictionary in `categorizer.py`:

```python
CATEGORY_KEYWORDS['Your Custom Category'] = [
    'keyword1', 'keyword2', 'keyword3'
]
```

📋 Example Output

Sample Receipt Analysis

```
===========================================
RECEPTIX - EXPENSE REPORT
===========================================
Generated: 2024-12-08 14:30:22
Source Image: grocery_receipt.jpg
-------------------------------------------

EXTRACTED TEXT:
SUPERMARKET XYZ
Milk 2%           $3.99
Bread             $2.50
Chicken Breast    $8.99
Coffee            $4.25
Total: $19.73

EXPENSE BREAKDOWN:
FOOD & DINING:
  Total: $19.73
  Items: Milk 2%, Bread, Chicken Breast, Coffee

GRAND TOTAL: $19.73
===========================================
```

🔧 Troubleshooting

Common Issues

1. "Tesseract not found" Error

- Ensure Tesseract is installed and in your system PATH
- Update the `tesseract_cmd` path in `ocr_utils.py`

2. Poor OCR Accuracy

- Ensure receipt image is clear and well-lit
- Try scanning at higher resolution
- Check if the image is properly oriented

3. No Amounts Detected

- Verify the receipt contains clear price information
- Check if currency symbols are visible

- Review the regex patterns in `categorizer.py`

4. Import Errors

- Run `pip install -r requirements.txt`
- Ensure all dependencies are installed correctly

Logging

Check the `logs/receipts.log` file for detailed processing information and error messages.

🤝 Contributing

Contributions are welcome! Here are ways you can help:

1. **Add Keywords**: Expand category keyword lists
2. **Improve OCR**: Enhance image preprocessing techniques
3. **UI Enhancements**: Improve the user interface
4. **Bug Fixes**: Report and fix issues
5. **Documentation**: Improve or translate documentation

📄 License

This project is open source and available under the MIT License.

🔮 Future Enhancements

□ Support for multiple languages
□ Receipt template recognition
□ Export to CSV/Excel formats
□ Cloud storage integration
□ Mobile app version
□ Machine learning-based categorization
□ Receipt validation and duplicate detection
□ Budget tracking and alerts

📞 Support

For support, questions, or feature requests:

1. Check the troubleshooting section above
2. Review the logs in `logs/receipts.log`

3. Create an issue in the project repository

🙏 Acknowledgments

- Tesseract OCR for OCR engine
- pytesseract for Python integration
- Pillow for image processing
- OpenCV for advanced image preprocessing

Made with ❤️ for better expense tracking