

Fake Invoice Detector using Computer Vision

Invoice no: 60178633

Date of issue:

12/25/2013

Seller:

Mendoza Group
40007 Brown Creek
South Meredithburgh, ND 48117

Tax Id: 900-78-2621
IBAN: GB36JYKL45599075785785

Client:

Elliott, Kelly and Warren
70134 Bridges Plains
Priceton, CO 07903

Tax Id: 939-81-7319

ITEMS

No.	Description	Qty	UM	Net price	Net worth	VAT [%]	Gross worth
1.	Confetti Stemless Wine Drink Glass Multi Green & Blue TAG FREE SHIPPING	5,00	each	23,99	119,95	10%	131,94

SUMMARY

	VAT [%]	Net worth	VAT	Gross worth
	10%	119,95	11,99	131,94
Total		\$ 119,95	\$ 11,99	\$ 131,94

By: Sofia Raheel

PROJECT OVERVIEW

The **Fake Invoice Detector** is a mini web application that uses computer vision and OCR to verify the authenticity of uploaded invoices. By comparing the visual layout and extracted key fields (like invoice number, date, and amount) with known authentic templates, the system flags invoices as *Original* or *Fake* based on detected anomalies. This project combines OpenCV, OCR (Tesseract/EasyOCR), and basic template matching to automate invoice verification.

CODE BASE

The presented codebase implements an **Invoice Verification System** that leverages **image processing**, **template matching**, and **optical character recognition (OCR)** to determine the authenticity of invoice documents. The system is built using Python, OpenCV, Tesseract OCR, and Flask (for the web interface). It distinguishes between **original** and **fake** invoices based on visual and textual similarities to known templates.

FUNCTIONAL COMPONENTS

1. **File Handling:** Automatically creates necessary directories for training, testing, templates, and output to organize image data efficiently.
2. **Preprocessing:** Converts images to grayscale, applies thresholding and noise removal to improve OCR and template matching accuracy.
3. **Image Hashing:** Generates a SHA-256 hash for each image to enable fast and exact duplicate detection.
4. **Template Creation:** Saves training images as standardized visual templates for comparison with uploaded invoices.
5. **Template Matching:** Uses OpenCV's normalized cross-correlation to identify visual similarities between invoices and templates.
6. **OCR Extraction:** Extracts text from images using Tesseract, along with bounding boxes for spatial comparison.
7. **Field Highlighting:** Annotates detected text fields on the image for visualization and result interpretation.

INVOICE VERIFICATION

- The `verify_invoice` function preprocesses the image, checks for exact hash matches, performs template matching, extracts and compares text, and finally classifies the invoice as **"Original"** or **"Fake"** based on similarity thresholds

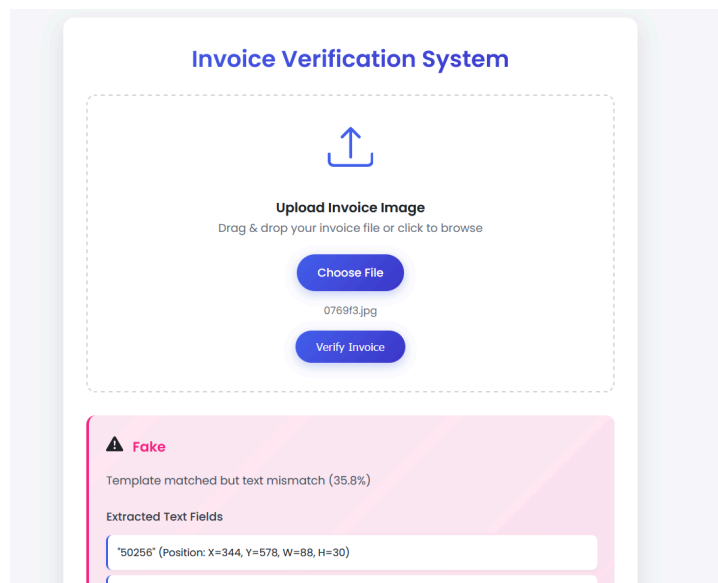
WEB APPLICATION (FLASK)

- The Flask app provides two routes: `/` for the upload interface and `/verify` for processing images and returning JSON results. It ensures file validation and secure handling.

KEY TECHNOLOGIES

OpenCV, **Tesseract OCR**, **Flask**, and **Hashlib** are used for image processing, text recognition, web interfacing, and file verification, respectively.

SAMPLE OUTPUT



Context: A sample fake image was provided where small changes in the name and amounts were made. Hence, detecting it as fake. Along with this, it displays the

places/positions where these changes are detected.

CONCLUSION

This project demonstrates how basic computer vision and OCR techniques can be used for document authenticity detection. The modular structure of the code allows for further enhancement, such as using machine learning models or expanding to multi-page PDFs. The web interface offers a simple way for end users to interact with the model and test different invoices for validation.