

Receipt Scanner for Receipt Validation and Data Extraction

Azan Ali (Lead)

Saad Ur Rehman

Shayan Ahmed

August 3, 2023

Abstract

The traditional manual receipt validation and data extraction processes are time-consuming and prone to errors, necessitating an automated solution to enhance efficiency and accuracy. Receipt Scanner aims to leverage modern technologies, such as visual feature extraction, machine learning, and OCR, to automate these processes. The project recognizes the increasing reliance on electronic transactions and digital receipts, driving the need for a robust and scalable solution to handle large volumes of receipts accurately.

1 Problem Statement

In today's digital era, traditional manual receipt validation and data extraction methods have proven to be slow and error-prone, posing significant challenges for businesses and individuals alike. To address these issues, the Receipt Scanner project aims to harness the power of cutting-edge technologies like visual feature extraction, machine learning, and Optical Character Recognition (OCR) to automate and streamline these processes. By automating receipt validation and data extraction, the project seeks to enhance efficiency, reduce errors, and save valuable time and resources.

As the digital landscape continues to evolve, the Receipt Scanner project aims to be at the forefront of automating receipt processing, revolutionizing the way businesses and individuals handle financial data.

2 PHASE-1: Receipt Validation and Comparison

2.1 Approaches

2.1.1 Autoencoder Approach

Autoencoders are a type of neural network architecture that belongs to the family of unsupervised learning models. In the context of receipt comparison and validation, autoencoders can be used to encode the input receipt image into a compressed representation (encoding) and then decode it back to reconstruct the original receipt image (decoding). The primary goal of autoencoders is to learn a compact representation of the input data, effectively reducing the dimensionality while preserving important features.

Reasons To Drop

1. **Lack of Semantic Understanding:** Autoencoders are unsupervised models and are trained to minimize the reconstruction error. However, they might not inherently understand the semantic meaning of the receipt components, such as transaction details or item names. This lack of semantic understanding can limit their ability to perform more advanced validation tasks.
2. **Limited Contextual Information:** Autoencoders encode the entire receipt into a compressed representation, treating all parts equally. In a receipt, different regions (e.g., header, item list, footer) have varying levels of importance. By using an autoencoder, important context-related information may not be effectively preserved, making it challenging to perform precise comparisons.

3. **Variability in Receipt Formats:** Receipts can come in various formats and layouts, with differing backgrounds, fonts, and structures. Autoencoders might struggle to capture all the variations present in receipt images, leading to reduced generalization and accuracy in comparing receipts with diverse layouts.

2.1.2 VGG16 Model Approach

VGG-16, short for Visual Geometry Group 16-layer model, is a deep convolutional neural network architecture that was proposed by the Visual Geometry Group at the University of Oxford in 2014. The primary objective of VGG-16 is image classification, which involves assigning a label to an input image from a predefined set of categories.

Reason To Drop While VGG-16 has demonstrated impressive accuracy in various image classification tasks, it does have some limitations. When applied to images with complex backgrounds, the model tends to extract features from both the foreground and the background. As a consequence, this indiscriminate feature extraction can lead to lower accuracy or issues in obtaining the desired results, particularly in scenarios where the background significantly influences the classification outcome.

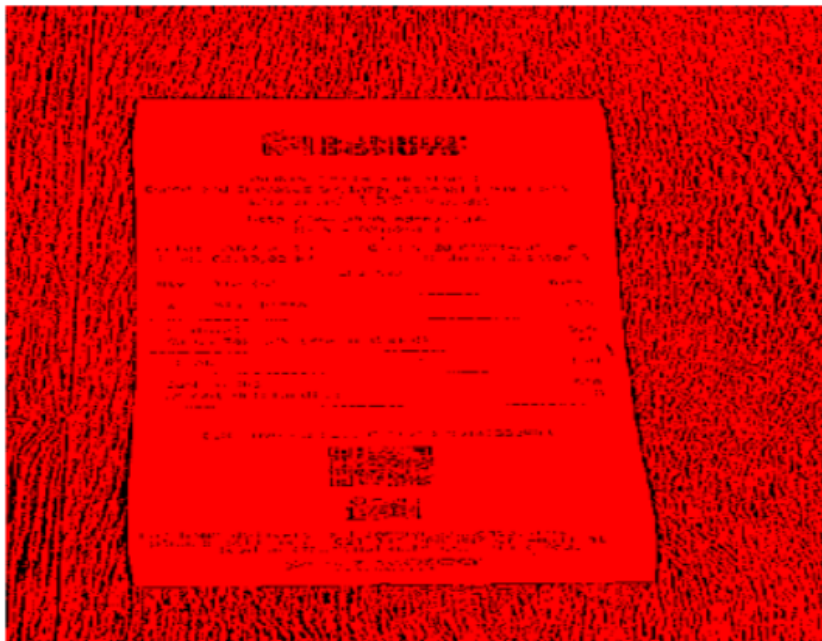


Figure 1: VGG16 - Image features Detection

2.1.3 BRISK Algorithm Approach

BRISK, short for Binary Robust Invariant Scalable Keypoints, is a feature detection and description algorithm used in computer vision and image processing. BRISK is designed to detect and describe distinctive key points in images.

Reason To Drop

- **Narrowed Matching Precision:** Despite its advantage of faster feature detection and matching times compared to the SIFT algorithm, BRISK falls short in terms of

accuracy. It often struggles to achieve correct matching results, leading to unclear and unreliable outputs.

- **Homogeneous Regions:** In images with large homogeneous regions or repetitive patterns, BRISK might struggle to find distinctive keypoints and descriptors, which can affect its matching accuracy.

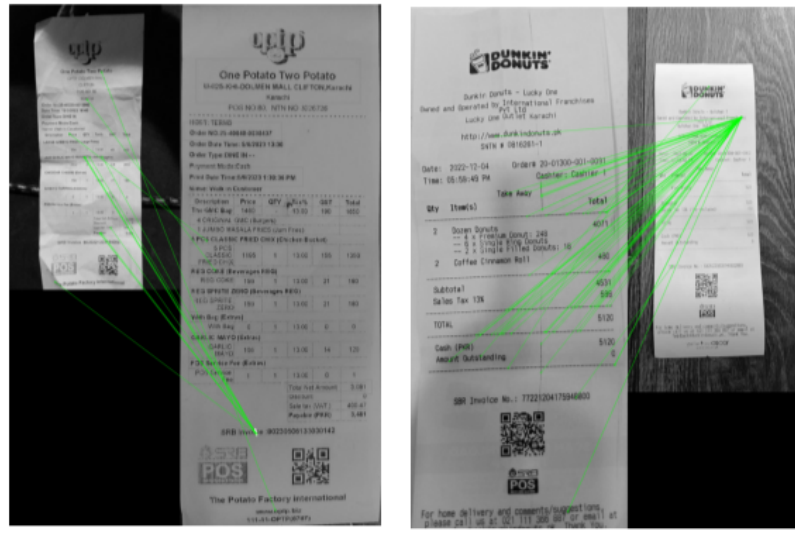


Figure 2: Brisk - Receipt Visual Features Comparison

2.1.4 ORB Algorithm Approach

ORB (Oriented FAST and Rotated BRIEF) is a feature detection and description algorithm used in computer vision and image processing. ORB is designed to efficiently detect and describe keypoints in images, making it suitable for tasks like image matching.

Reason To Drop

- **Limited Matching Accuracy:** ORB is a binary feature descriptor, which means it may not provide the same level of matching accuracy as more advanced algorithms like SIFT (Scale-Invariant Feature Transform) or SURF (Speeded-Up Robust Features). ORB might struggle with complex scenes or images with significant viewpoint changes.
- **Lack of Distinctiveness:** ORB features might not be as distinctive as those generated by other algorithms like SIFT, potentially leading to a higher rate of false positives and false negatives in the matching process.
- **Limited Robustness to Occlusions and Noise:** ORB might struggle with matching images that contain occlusions (objects partially hidden) or noise. It may produce false matches or miss matches in such scenarios.

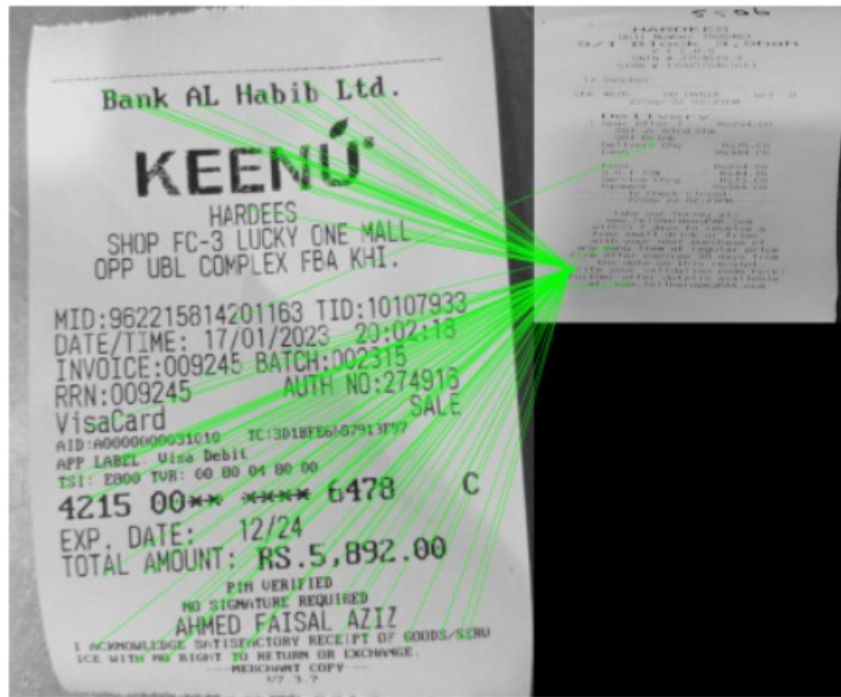


Figure 3: ORB - Receipt Visual Features Comparison

2.1.5 Sift Algorithm Approach

The SIFT (Scale-Invariant Feature Transform) algorithm is a widely used computer vision technique for image feature extraction and matching. It can be applied to receipt layout comparison and detection to find common features or key points in different receipts and determine their similarities or differences. Here's an explanation of how the SIFT algorithm works in this context:

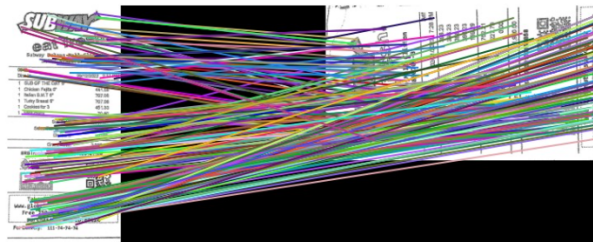


Figure 4: SIFT - Receipt Visual Features Comparison

1. Feature Extraction: Extract distinctive and scale-invariant keypoints from receipts. Keypoints are invariant to rotation, scale, and illumination changes.
2. Keypoint Detection: Identify unique regions (corners, edges, etc.) in the receipts. Use Difference of Gaussians (DoG) method for scale-space keypoint detection.
3. Keypoint Description: Describe each keypoint's local neighborhood robustly. Create descriptor vectors based on gradient orientations.
4. Receipt Matching: Match keypoints between two receipts. Consider descriptor similarity, spatial proximity, and scale difference.

5. Layout Comparison & Detection: Identify common regions or layouts in both receipts. Higher keypoint matches indicate similar layouts.
6. Robustness to Changes: Handle rotations, scaling, translation, and partial occlusion. Ensures accurate layout comparison in various receipt images.

3 PHASE-2: Receipt Extraction and Parsing

3.1 Receipt-Parser APIs Comparative Analysis

3.1.1 Introduction

The report provides a comparison of the top Receipt-Parser APIs available based on price model, accuracy, and limitations. The APIs analyzed include Amazon Textract, Google Cloud Vision, Eden AI, TapScanner, Mindee, Aspire, and Veryfi. Receipt-Parser APIs allow for the automated extraction of key data fields from receipts such as merchant name, date, total amount, and line items. This data can then be used for accounting, expense reporting, and other business applications.

3.1.2 Table


API	Pricing	Accuracy	Limitations
Amazon Textract	Pay-per-page tiers from \$1.50 to \$0.60 per 1,000 pages	90% accuracy for clear receipt images. Accuracy drops to 60-70% for blurry or distorted images.	Can struggle with low contrast receipts. Limited handwriting recognition.
TapScanner	Free for individuals, \$9/month per user for teams	85-90% accuracy for clean receipt images. 70-80% for lower quality images.	Struggles with faded text or warped receipts. Minimal ability to handle complex layouts.
Mindee	Free up to 100 docs/month, \$0.004 per doc after	93-95% accuracy for crisp, well-lit receipt images. 90% for blurry images.	Accuracy declines below 85% for poorly lit or distorted receipts. Limited handwriting recognition.
Aspire	Free up to 1,000 pages/month, \$0.002 per page after	90% accuracy for high-quality receipt photos. Around 80% for lower quality images.	Has trouble with warped or distorted receipts. Minimal support for faded text.
Veryfi	Free up to 200 pages/month, \$0.005 per page after	90-95% accuracy for well-composed receipt images. 80-85% for blurry images.	Struggles with distorted or warped receipts. Limited handwriting recognition.

Eden AI and Mindee offer the most accurate Receipt-Parser APIs overall based on third-party evaluations. For companies needing multi-language support, Google Cloud Vision has the broadest language capability. Amazon Textract provides the lowest cost per document for high-volume use cases. When selecting a Receipt-Parser API, businesses should evaluate accuracy, language needs, document types, and monthly processing volumes.

3.2 Approaches

3.2.1 OCR & OpenAI for Receipt Extraction & Parsing

Optical character recognition (OCR) techniques, combined with OpenAI's language processing capabilities, are utilized for data extraction and parsing from validated receipts. The OCR engine identifies text elements such as store name, date, total amount, individual item names, and prices. OpenAI's language model aids in understanding and parsing the extracted text, enabling structured data output.



```

1 {
2   "brand_name": "KFC",
3   "outlet": "Town Bahria Kfc 0181",
4   "receipt_id": "2215",
5   "date": "08/10/2021",
6   "time": "9:40 PM",
7   "items_detail": [
8     {
9       "name": "MIGHTY ZINGER BURC",
10      "price": "Rs. 596",
11      "quantity": "1"
12    },
13    {
14      "name": "CHEESE 40",
15      "price": "Rs. 200",
16      "quantity": "1"
17    },
18    {
19      "name": "MIGHTY BURGER COM",
20      "price": "Rs. 765",
21      "quantity": "1"
22    },
23    {
24      "name": "ZINGER COMB ? 66C",
25      "price": "Rs. 660",
26      "quantity": "1"
27    },
28    {
29      "name": "ZINGER BURGER @ 47",
30      "price": "Rs. 940",
31      "quantity": "1"
32    },
33    {
34      "name": "CHICKEN 43",
35      "price": "Rs. 435",
36      "quantity": "3"
37    }
38  ],
39  "total_amount": "Rs. 3,615"
40 }

```

Figure 5: OpenAI - Response

Advantages

- OpenAI’s language model improves parsing accuracy and context understanding.
- OCR combined with OpenAI’s language processing provides a comprehensive solution for data extraction.
- Scalable and adaptable to different receipt formats, handling complex layouts effectively.

Limitations

- Accuracy can be affected by low-quality images or text with distortions.
- Language models may not handle rare or highly specialized store names accurately.

3.2.2 LayoutLMV3

PADDLE OCR Paddle OCR (Optical Character Recognition), a cutting-edge technology developed by PaddlePaddle, offers a powerful solution for processing receipt images. By harnessing the capabilities of deep learning and computer vision, Paddle OCR excels in efficiently extracting and interpreting text from receipt images, optimizing the entire data extraction workflow.

After processing an image, Paddle OCR return the extracted text and associated information as a JSON object, making it easy to parse and utilize the recognized text and its attributes in your application.

ANNOTATION USING LABEL STUDIO Our data annotation workflow seamlessly combined Paddle OCR’s JSON output with Label Studio’s annotation capabilities, resulting in accurately labeled receipt images. Here’s a concise overview of how we achieved this synergy:

- **Paddle OCR Integration:** We utilized Paddle OCR’s advanced capabilities to process receipt images and extract relevant information. Paddle OCR generated JSON files containing detected text and bounding box coordinates for each element in the image.
- **Label Studio Setup:** Using Label Studio, we set up annotation projects and configured tasks tailored to our receipt image analysis. We defined annotation labels, guidelines, and interface layouts to ensure clear instructions for annotators.
- **JSON Import:** Leveraging Label Studio’s JSON import feature, we seamlessly integrated Paddle OCR’s JSON output into the platform. This allowed annotators to work with pre-existing bounding box data generated by Paddle OCR.
- **Annotator Collaboration:** Annotators interacted with the labeled bounding boxes in Label Studio’s user-friendly interface. They identified and verified the content within each bounding box, associating meaningful labels with the detected elements.

- **Review and Quality Control:** As annotations progressed, our team monitored the annotation process for accuracy and consistency. Label Studio’s collaborative environment facilitated ongoing communication and feedback among annotators and reviewers.
- **Iterative Refinement:** In cases where uncertainty or ambiguity arose, annotators and reviewers engaged in an iterative refinement process. This ensured that the labeled bounding boxes accurately represented the content within the receipt images.
- **Exporting Annotated Data:** Upon completion of annotations, Label Studio allowed us to export the annotated data, now enriched with accurate labels. This prepared dataset formed the foundation for training and fine-tuning our machine learning models.

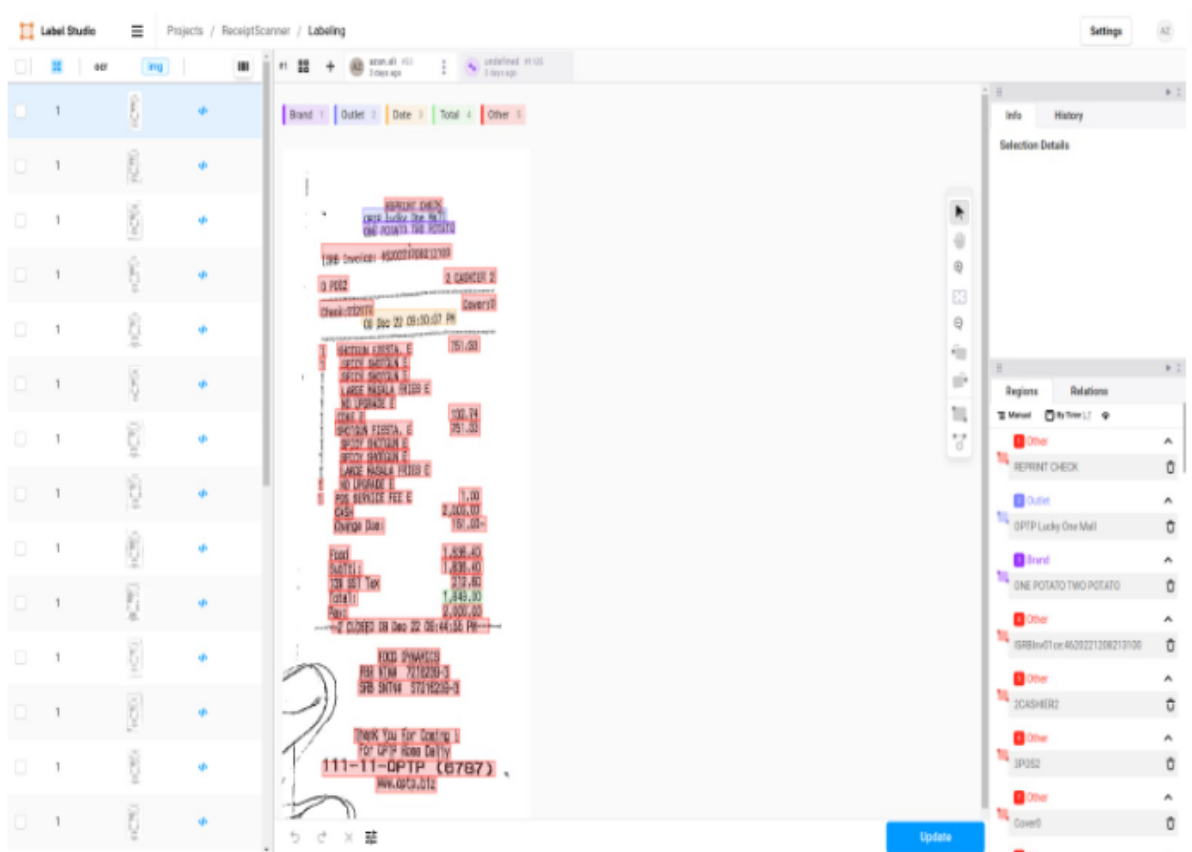


Figure 6: Annotation using Label Studio

```

{
  "ocr": "http:\\\\localhost:8082\\12220-638c9096021c5.png",
  "id": 1,
  "bbox": [ ...
  ],
  "transcription": [ ...
  ],
  "label": [ ...
  ],
  "annotator": 1,
  "annotation_id": 43,
  "created_at": "2023-08-02T15:20:03.620687Z",
  "updated_at": "2023-08-02T15:20:03.620709Z",
  "lead_time": 13.571
},
{

```

Figure 7: Model Training and Evaluation

4 Final Proposed Solution

- Phase 1: SIFT Algorithm for receipt comparison
- Phase 2: LayoutLMv3/OpenAi for data extraction & parsing

5 Conclusion

The Receipt Scanner for Receipt Validation and Data Extraction project addresses the challenges of traditional manual receipt management processes by leveraging modern technologies. The combination of SIFT algorithm for receipt layout comparison and OCR with OpenAI or LayoutLMV3 for data extraction provides an accurate and efficient solution. The project's multi-phase approach ensures the validation and extraction processes are handled systematically, resulting in a user-friendly and scalable application. While there are limitations to each approach, they are outweighed by the significant benefits they bring to the overall project.