ExpensifyAI: The AI Expense Tracker and Predictor

Dr.Thomas Feldmen
Assistant Professor (Artificial Intelligence & Data Science)
All India Shri Shivaji Memorial Society's Institute of Information Technology
Pune – 411001, India

Diya Joshi
Undergraduate (Artificial Intelligence & Data Science)
All India Shri Shivaji Memorial Society's Institute of Information Technology
Pune-411001, India

Prerit Loharkar
Undergraduate (Artificial Intelligence & Data Science)
All India Shri Shivaji Memorial Society's Institute of Information Technology
Pune-411001, India

Namrah Latifi
Undergraduate (Artificial Intelligence & Data Science)
All India Shri Shivaji Memorial Institute of InformationTechnology
Pune-411001, India

Abstract— **This project is concerned with the creation of an automated expense management system that makes use of OCR and LLMs to enhance invoice processing. This system is capable of extracting necessary information embedded in documents of different formats on its own, which considerably lessens the amount of time allocated to data entry without compromising on precision. After extraction, data is categorized, and is subsequently pre-processed making it more orderly in relation to finances. Users can also manage and have a complete view about their expenses via an intuitive interface that makes it easy to configure filters and set tracking.**

Moreover, users are provided with additional assistance concerning trackable spending behaviour by the incorporation of predictive analytics that enables expense forecasting owing to previously recorded data. Eventually, the goal of this project is to increase spending accuracy and efficiency with relative spreadsheets regardless of the target user in question. It is also scalable incorporating into existing systems and thus ensuring relevancy and utility over time.

We aim to build a working prototype of the automated invoice processing and expense management systems that harnesses the power of Django, OCR, and OpenAI's language models for seamless financial processes. The system learns from a variety of templates for invoices and uses Tesseract OCR for text extraction, combining it with OpenAI's NLP models to classify and tag constructed invoice data into fields like dates, vendors, and itemized costs for invoices. The open-source web framework makes it easy to upload and process invoices and have the system organize expenses while estimating future costs using machine learning algorithms to analyze historical data. The system is efficient, accurate, and intuitive in nature, eliminating unnecessary intervention while providing deep financial insights through an interactive web interface. Detailed and personalized expense management can be achieved through the combinative use of OCR, Django, NLP, and predictive analysis, making the system easily configurable for every individual or organization.

Keywords—: OpenAI, OCR, Django

I. INTRODUCTION

This particular project deals with providing an automated invoice processing and expense tracking systems that utilize Optical Character Recognition, machine learning, and NLP to facilitate the smooth running of financial workflows. Using TesseractOCR technology, the system makes it easy to penetration and computerize invoices by taking the details, including the invoice number, date, and cost of items, and documented them in structured formats like JSON or CSV. The expenses data collected from invoices is stored and then organized with the aid of Open AI language models. In addition, Django allows the problem to be solved with web access interfaces, which enhances user interactivity and real time data processing. Furthermore, the mentioned expense data assists the machine learning algorithms to make accurate predictions The company benefits greatly from this value addition. Image preprocessing and error correction methods alongside other system functionalities greatly reduce the manual work and errors involved in spending management since the system can easily be adjusted to work with multiple invoice designs. Therefore, the system is very easy to scale and is ideal for business settings.

At the backend, I employ Django which provides a solid framework to handle, store and manage data. Using the Model-View-Template (MVT) architecture, Django allows the speedy and efficient processing of data. Like in the case of invoices and expenses, models take care of the database structure. Views build the data processing logic, while templates display the final output to the end users. The built-in ORM of Django comes in handy for storing structured JSON data from OCR extraction and expense prediction since it simplifies database operations. Additionally, powerful URL routing and middleware of Django allows the effortless workflow of submitting invoices and retrieving data stored in them for automated invoicing.

With Tailwind, styling makes use of the already existing utility classes which allows efficient and straightforward styling. Tailwind allows intricate user interfaces to be constructed at a high speed, without the need for custom CSS. This is achieved by Tailwinds utility classes, which are key in effective responsive web design because they allow the scaling of the interface for various screen sizes. Because of this, Tailwind makes it fast and simple to design bells and buttons gives the interface a professional look while keeping it user friendly.

To keep the user interface organized, HTML serves as the primary building block for all the information that needs to be presented to the user. HTML is used to create forms for users to easily and directly upload invoices to the system.

The API used in this project is designed to improve the organization of data by examining the extracted data from invoices and classifying them into relevant expense categories. This AI-driven categorization improves accuracy and reduces the amount of manual work needed in processing large volumes of unstructured financial data.

The Open AI API is implemented to comprehend the image's text in the invoice and derive the most valuable and pertinent information. First, the function checks all potential categorizations in a Django model called 'Category' and converts them into a string list. A dynamic prompt is then built looking for the invoice details like purchase date, an amount spent, prospective category from the list, and the product or service name. This prompt and invoice text are sent to OpenAI's language model (GPT-3.5-turbo-instruct), who processes and retrieves the sought information from the detail provided.

The system produces the output in a predetermined structure. In the occurrence of an error, it is stored, and None is returned. This procedure aids information extraction automation, which leads to enhancement in the figure-earning AI invoice processing category as it achieves better accuracy in information segmentation and structuring.

To derive the text from invoice photos, Optical Character Recognition (OCR) is used. Using 'cv2' and 'pytesseract' libraries, the process first imports and scales the pictures to enhance accuracy. Then the final image is made greyscale and binarized to filter out noise, increasing the text visibility.To improve text extraction, deskewing is conducted, which identifies and corrects any sort tilt or rotation in the text orientation. While preprocessing, the cleaned picture is sent to the 'pytesseract' OCR engine, which converts the visual text format into a readable string. This extracted text may then be analysed further for categorisation and better visualization purposes, making it appropriate for automated invoice processing that requires structural details from unstructured scanned documents.

II. LITERATURE SURVEY

The research papers that are referred mainly focus on using Tesseract OCR, recognized for its accuracy and ease in integration, mainly through Python's PyTesseract library. Each step is an essential image pre-processing (binarization, skew correction, resizing) to enhance OCR accuracy by isolating text components. Tesseract's integration with various pre-processing libraries, like OpenCV, is highlighted to increase ability of detection, segmentation, and extraction of structured data from invoices, especially in noisy, unstandardized or distorted images .

Invoice processing involves supervision of invoices from receipt to payment, and typically a very time-consuming task. This application uses Tesseract OCR to automate invoice information extraction, identifying data such as invoice numbers, dates, vendor names, and total expenditure from scanned images. It also incorporates support for multiple languages, also ensuring scalability to handle large number of invoices. Error-handling methods deal with issues like low-quality images, while reporting tools that provide analytics from the extracted data. Therefore, making it an efficient and scalable solution for various businesses.[1]

Optical Character Recognition (OCR) has become an critical technology that is being used to convert scanned images and various other visual data into text. This project applies OCR, specifically Tesseract, to digitize invoice information, converting it into JSON and CSV formats. Image preprocessing techniques, including grayscale and noise removal, are used to enhance image quality before processing. By integrating Tesseract OCR with Python libraries, the system provides a reliable method to convert complex invoice layouts into structured, usable data formats, supporting further applications in data analysis and automation.[2]

Although the second research does not explicitly incorporate machine learning, it argues that extracted data may be utilised for predictive analytics by categorising costs, allowing organisations to follow spending patterns. This is consistent with your project's objective of ML-based expenditure categorisation and visualisation. ML models might be taught to forecast future costs and assist with budget optimisation by using previous invoice data.

If a match is found between the text and any of the existing templates or similar templates, the server extracts the text according to the matched template's structure and converts it into JSON format.[1] The existing system contains data extraction and nothing more. In a paramount manner, image pre-processing techniques like black and white, inverted, noise removal, grayscale, thick font, and canny are applied to escalate the quality of the picture.[2]

OCR is a computationally fast algorithm. One of the most notable advantages of our system is its speed. It rapidly and accurately extracts essential information from invoices, significantly reducing manual data entry time and human error.[1] The main limitation is it only works on the format specified in the program and only restricted to English language.[2]

III. PROPOSED SYSTEM

Automated invoice processing has become very important in business environments where efficient expense management and quick access to financial information are vital. Traditional manual processing of invoices can be labour-intensive, error-prone, and inefficient, which limits a company's ability to answer to financial insights in real-time.

A. Data acquisition and pre-processing

Initially, data acquisition and preprocessing are important for ensuring that the invoices are all set for OCR and processing. This phase involves the collection and secure storage of invoice images, which is achieved by user uploads. Once collected, invoices undergo preprocessing to increase OCR accuracy, addressing abnormality in image. This preprocessing involves conversion of images to grayscale, resizing to standard dimension, applying noise reduction techniques to refine text and reduce background noise. By standardizing the quality of input images, the system can achieve improved text recognition results, setting a stable foundation for later steps.

B. OCR using PyTesseract

During this stage of the process, text extraction is performed with the use of Tesseract OCR, which is an image file text extraction application. In the beginning Tesseract performs quite well with the invoices once they have been supplied with format-specific parameters like the setting of page segmentation mode to an appropriate value for document structure analysis. This method of OCR can indeed read the text on the image of an invoice but the produced output is derived with minimal effort and is quite crude. There are further adders that can be defined which focus aid the output refinement and reduction of errors from OCR and tokenization over post-processed raw data which define data structure.

This corroborates that the OCCR output is clean and classifies as correct data for further parsing thus reduces the fault of misinterpreting the data in later processes.

C. Data extraction and structuring

After processing raw documents through OCR, the system extracts and identifies relevant information from the documents. This extracting can include important invoice particulars like date, vendor name, total amount and item descriptions, which is possible through a combination of regular expressions and NLP techniques. Once these entities are recognized, they are transformed into a structured format like JSON or a relational database schema so that easy access is ensured. The structured information can then be stored in the structured information can then be stored in a database so that retrieval, searching and filtering is made easier. At this stage, a strong and accurate with tracing can support analytics and machine learning tasks even within a model of the data is created.

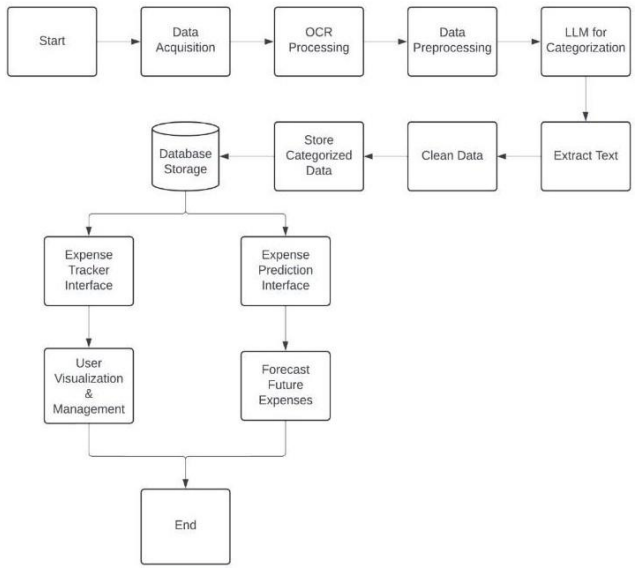

Fig 1. Flowchart

D. Category prediction using GPT model

To better understand the analysis, expenditure should be split into certain predetermined categories, like travel, office supplies, or utilities. For this aim, a version of the system

includes GPT technology that utilizes NLP techniques to classify an item's or expenditure's description. With regards to specialized understanding of these advanced business spending categories, fine-tuning is indeed possible using labelled data. In cases with a certain degree of uncertainty, additional, rule-based tests or labelled datasets can be employed that should enhance accuracy. Moreover, by classifying expenditure into groups, one is able to gain analytical complexity while also drawing useful snapshots into spending behaviour.

E. Data visualization

The system includes a data visualization tool that depends on libraries like Matplotlib, Seaborn, or Plotly. These instruments enable the creation of visual summaries in the form of bar graph, pie charts, and line graphs to provide an overview of the expenditures and unusual patterns. Also, these visualizations are incorporated in a Django powered interactive data dashboard that enable self-service exploration and filtering of data at users' convenience. This makes users competent to research specific categories of expenditures, periods, or vendors which is an essential tool for a business's intelligence.

F. Django framework and Admin Page

Django Framework serves as a backbone of the system and it provides a complementary backend and UI support. The devement cycle of the software is improven through the integration of OCR, data processing, machine learning, and even visualization into one cohesive infrastructure using the Model-View-Template (MVT) design architecture. In addition, Django Object-Relational Mapping (ORM) facilitates the storage of structured data in databases which increases the efficiency of query processing and database management. A critical part of this system is the Django Admin page, which facilitates effective management and supervision of data through a robust inbuilt interface. With the admin page, administrators can monitor incoming invoices, manually classify costs and add new categories as appropriate, and rectify any issues with OCR output or problems. In addition, the Django Admin site introduces role-based access control, enhancing restriction of sensitive financial information and enabling administrators to set parameters for machine learning models, categorization rules, and visualizations. This powerful framework allows for easy integration of other components and scaling of the application to fit the business requirements.

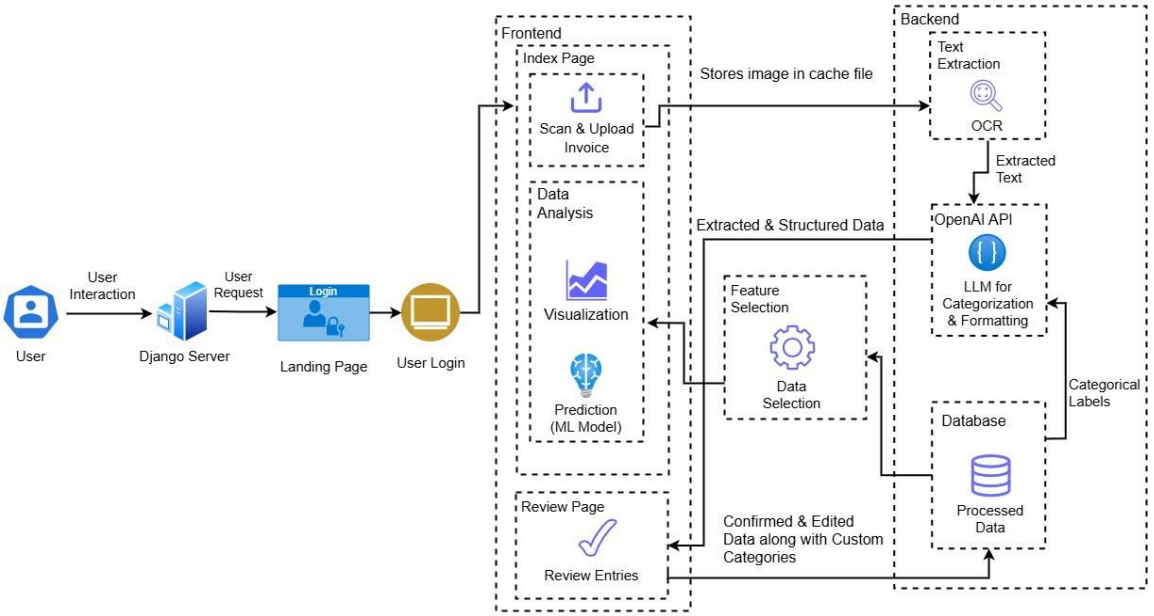

Fig 2. System Architecture

G. Machine learning expenditure predictions model

Beside fitting existing data, the model applies a machine learning approach to estimate expenditures given the previous spending patterns. This involves feature engineering where relevant features are extracted from the structured invoice data, for example, averages of monthly expenses, frequencies of vendors, or modifications in spending over seasons so that they can be used to train the model. Different approaches, such as random forest, linear regression, or neural networks, can be applied first to determine which one is best for the task of expense estimation. These models are refined and tested through using historical invoice data, and their effectiveness is evaluated using metrics like accuracy, mean squared error, and others. This prediction offers valuable trends to look forward when deciding budget and spending.

H. System Deployment

The construction of this system aims to make use of cloud services such as AWS or Heroku which offer scalability, ease of use, and high resource security. The application can be deployed on AWS with more flexibility through Elastic BeanStalk, EC2s or even Docker containers on ECS to gain better control on environmental variables and resource limits. In order to securely control the invoices dataset, we can store and manage it in AWS RDS or S3. Also, a set of AWS Lambda functions can be created and combined to execute specific processes like turning on the OCR, classifying expenses, and executing them, without the use of a dedicated server.

This method is however simpler with Heroku as the deployment can be done with an automatic procedure which makes this option more suitable for low maintenance projects or expanding ones as it is more effortless to make use of servidores.

In addition, Heroku Postgres and Redis, which are services for Heroku, can provide the system's database needs, and ability to change the scale of the application allows it to respond to increased activity. Both providers offer a protected and elastic infrastructure which makes the deployment of Django-based applications easy and enables users to interact hassle-free with developing application's OCR, machine learning, and OD features.

IV. RESULTS AND DISCUSSION

As a result of the project, a functional system that automates the processing of invoices should be achieved. This system should automatically retrieve information from a number of invoice types, sort them according to appropriate categories, and save them where instructed. The use of Tesseract OCR to scan invoices and employing various pre-processing methodologies should ensure that the invoices are scanned and the details of the invoice, such as vendor information, item descriptions, amounts, and dates, should be converted into a machine-readable format such as JSON or CSV. In addition, this containing the machine learning component should allow for continuous expense prediction based on the historical data making the system more efficient and effective for expense control and the decision making on the expenses. The web front of the system should be developed using Django, HTML, Tailwind, and CSS, and it should enable users to seamlessly estimate the predictions, and upload invoices and categorized expenses. It should also ensure that users can intuitively and interactively use the interface. The functionality of the system can be tested under varying constraints to assess the effectiveness of the system. To test the accuracy of the OCR system, literature recommends that the extracted texts be compared against the original invoices as well as other relevant details, dates, items, and amounts, to ensure that the information is properly captured. Furthermore, OCR effectiveness can be tested and quantified using various measures like the Character Accuracy Rate (CAR) or the Word

Accuracy Rate (WAR) that determines how many of the characters and words identified were correct. Precision, Recall, and F1 Score are metrics that measure the consistency of classification and forecasting within a system's classification expenses. Likewise, the usable test will check that the online application is functional and bug-free on different devices.

Moreover, response time and system scalability really say something about system performance, particularly when we are handling a high load of invoices and performing a number of ML predictions concurrently. Through the measuring of the set parameters, the project works towards confirming its goal of: high accuracy in data extraction, stable classification and predictions, and user-friendly interface for practical application in financial management.

V. CONCLUSION

The entire process of invoice management within a company is a big chore on its own. This project aims to have OCR, alongside machine learning, integrated into a user-friendly web interface in order to promote productivity and accuracy while automating invoice processing as well as expense tracking. The use of data extraction and forecasting allows for specially designed financial management systems to be implemented, making it easier to scale. With the utilization of Tesseract OCR to extract text and a GPT model that categorizes expenses while predicting and analyzing a company's finances through machine learning, the expenses are streamlined alongside the accuracy.

VI. FUTURE SCOPE

1. Improving security and tailoring user experience by introducing a login page that expands user accessibility, making it convenient to control access to select authorities.

2. Attempt to integrate sophisticated machine learning algorithms in the expense forecasting in order to improve its accuracy by taking into account trends, vendor behavior, and business patterns.

3. Implement an AI chatbot designed to help users manage invoices and expenses by providing quick answers to frequently asked questions.

4. Create dashboards with advanced data analytics and detailed visualization that help users monitor their spending patterns, identify anomalies, and make data driven decisions.

5. Aim at enabling support for more file types such as multi-page invoice PDFs in order to make the system more flexible to complex invoice documents.

VII. ACKNOWLEDGMENT

AISSMS's IOIT, Pune AI and Data Science department warmly deserves our heartfelt gratitude for their contribution and assistance during the project "ExpensifyAI: The AI Expense Tracker and Predictor." We are incredibly thankful to the mentors and professors involved with this project for their time and effort.Their expertise and knowledge played a key role in making this project successful.

VIII. REFERENCES

1. An Intelligent Invoice Processing System using Tesseract OCR,2024, Ashlin Deepa R N, Suhas Chinta, Nikhil Kumar Ashili, B Sankara Babu, Revanth Reddy Vydugula, Raj Sripada VSL.

2. Digitization of Data from Invoice using OCR, 2022, Venkata Naga Sai Rakesh Kamisetty, Bodapati Sohan Chidvilas, S. Revathy, P. Jeyanthi, V. Maria Anu, L. Mary Gladence.

3. An Empirical Analysis of Topic Categorization using PaLM, GPT and BERT Models, 2023, Dhanvanth Reddy Yerramreddy, Jayasurya Marasani, Ponnuru Sathwik Venkata Gowtham, S Abhishek, Anjali