iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.70034

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

ExpensifyAI: The AI Expense Tracker and Predictor

Swati Kadu[1], Diya Joshi[2], Prerit Loharkar[4], Namrah Latifi[3]
[1]*Assistant Professor (ArtificialIntelligence & Data Science), All India Shri Shivaji Memorial Society's Institute of Information Technology, Pune – 411001, India*
[2, 3, 4]*Undergraduate (Artificial Intelligence& Data Science), All India Shri Shivaji Memorial Society's Institute of Information Technology, Pune-411001, India*

Abstract: ***This project focuses on developing an automated expense processing system which utilizes OCR and LLMs for invoice enhancement. Such system processes the necessary information contained in documents of various types and formats automatically. This greatly decreases the time spent on data entry, without losing accuracy. Following extraction, data is classified, and further processed to be more orderly with regard to finances. Users are also able to view and manage their expenditure through an intuitive interface which allows setting filters or defining tracking criteria. One of the main features of the system is the ability to incorporate analytics self-service – where users can analyze their spending habits based on historical data and make predictive models, enabling them to spend less in the future. In particular, structured historical expense data and spending for the upcoming month were evaluated using XGBoost, which also served as a financial prevision tool due to its advanced machine learning capabilities. XGBoost does not rely on explicitly defined relationships, rather it thrives on the existence of complex, non-linear relationships between features such as amount of expenses charged previously, transaction frequency, category, seasonality, and trend indicators. It constructs an ensemble of decision trees where each subsequent tree is built to increase precision, hence a greater ensemble will be more accurate. Tesseract OCR, Django, OpenAI's language models, and XGBoost were incorporated into a single cohesive and scalable framework that serves the organizational and individual users personalized financial goals.***

Ultimately, the result of this project is to improve spending accuracy and efficiency with respect to related spreadsheets irrespective of the intended user. Moreover, it is scalable incorporating into pre-existing systems and thus maintaining relevance and value over time.

Our aim is to develop a functional prototype for automated invoice processing and expense management using Django, OCR, and OpenAI language models to streamline financial workflows. The system is designed to learn from various invoice templates, applying Tesseract OCR for text extraction and NLP models for classifying key fields such as dates, vendors, and itemized costs. Leveraging Django's web framework, users can easily upload, organize, and manage expenses, while historical data is used to estimate future costs through machine learning. The intuitive web interface delivers actionable financial insights, and the combination of OCR, NLP, and predictive analytics allows for detailed, customizable expense tracking suited to both individuals and organizations—making the system efficient, accurate, and user-friendly.

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.

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 backend of ExpensifyAI will integrate XGBoost (Extreme Gradient Boosting) to project upcoming costs with greater accuracy and flexibility. XGBoost, as a machine learning model, is known for its speed and precision. Unlike traditional models in a time series, XGBoost does not assume the presence of a seasonality or trend which needs to be decomposed; rather, it learns directly from the data using gradient-boosted decision trees. After an expense record is generated through OCR, and later categorized using NLP along with its advanced categorization features, the data is structured into a supervised learning format which uses records as individual pieces of data that come with specific value indicators like prior spending, temporal indicators (month, day of the week), spending frequency, and behaviors associated with various categories. XGBoost builds an array of decision trees which use complex non-linear relations with variable interactions and deal with many interdependent relationships competently. With this setup, accurate short-term forecasting is made possible leveraging spending noise or erratic spending patterns alongside numerous other noise-inducing factors. The model uses historical data to train its predictive system for estimating monthly expenditures, preverbal data includes expected values alongside importance scores for features explaining which elements most greatly affect spending are metrics with further justification for relevance.

Users benefit from XGBoost's capability to handle complex, high-dimensional data and its robustness against outliers and irregular spending patterns. For this project, the system predicts the next one month's expense trajectory by leveraging engineered features such as prior spending amounts, temporal indicators, and category-specific behaviors. These predictions help users and businesses anticipate upcoming costs and make proactive, data-driven budgeting decisions. As new invoice data is continuously added to the system, the model can be retrained or incrementally updated to maintain accuracy and relevance over time. While XGBoost does not natively provide confidence intervals like traditional time series models, its use of feature importance metrics offers interpretability into which variables drive predictions, thus supporting transparency in financial forecasting. By embedding XGBoost into the system pipeline, ExpensifyAI shifts from being a reactive expense tracker to a **predictive financial assistant**, delivering timely foresight that enhances planning and resource allocation.

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.

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 module that utilizes libraries such as Matplotlib, Seaborn, and Plotly to generate interactive and insightful visual summaries of expense data. These visualizations include bar graphs, pie charts, and line charts, which provide users with an intuitive understanding of their spending behavior over time. A Django-powered interactive dashboard enables self-service exploration, allowing users to filter and analyze expenses by category, vendor, and specific time periods.

In addition to historical data visualization, the dashboard also incorporates predictive insights generated by XGBoost model. This includes line graphs that depict future one-month expense forecasts, along with upper and lower confidence intervals that help users understand the potential range of variation in their predicted spending. These forecasts are automatically refreshed as new data is added, and users can visually compare historical patterns with projected trends. This comprehensive visualization approach not only aids in identifying anomalies and patterns but also empowers users to make informed financial decisions by anticipating future expenses in an interactive, accessible format.

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.

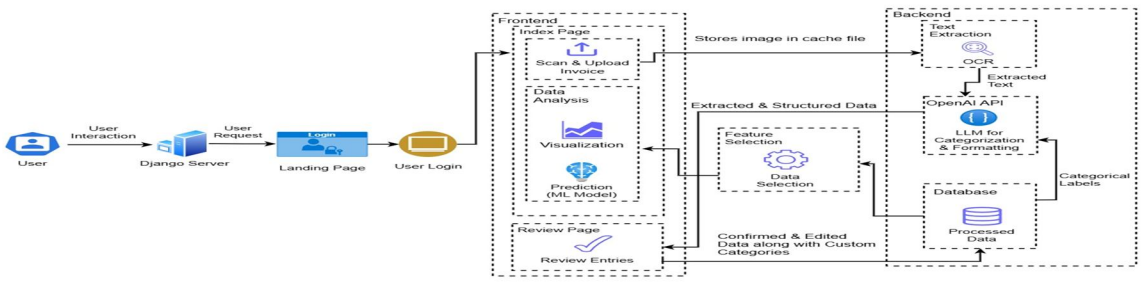


Fig 1. System Architecture

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

To maintain performance and ease of integration with cloud infrastructure, this system can be deployed on key cloud service providers like AWS, or Heroku. These platforms offer varying levels of automation for spiral model processes, sophisticated protective measures, and good command of resources which are important in sustaining a responsive application. These platforms provide flexible deployment pipelines, robust security, and resource allocation, which are very important for maintaining a responsive application. The deployment architecture makes use of Docker containers for environment consistency and AWS Elastic Beanstalk, EC2 for orchestrating containerized applications. Databases such as AWS RDS and Heroku Postgres store structured data, while AWS S3 can be used for managing invoice images.

Integration of the distributed streaming platform Apache Kafka greatly improves this deployment strategy. Kafka enables real time data flow between different pieces of the system for seamless processing. It works as a message broker with high throughput between numerous microservices to OCR processing, NLP categorization, prediction and visualization enabling them to work independently and asynchronously. This typical design has an impact on the responsiveness of the application and fault tolerance, particularly in instances with heavy inflow of invoiced data includes incoming data streams. An important enhancement in this deployment strategy is the integration of Apache Kafka, a distributed streaming platform that facilitates real-time data processing across multiple components of the system. Apache Kafka acts as a high-throughput message broker between the various microservices—OCR processing, NLP categorization, prediction, and visualization—allowing each to operate independently and asynchronously. This distributed architecture significantly improves fault tolerance and system responsiveness, especially when handling large volumes of incoming invoice data. For example, once a user uploads an invoice, a Kafka producer publishes the event to a topic, and multiple consumer services (e.g., OCR module, categorization engine, forecasting module) independently consume and process the event, all without blocking one another. This decoupling ensures that high-volume workloads can be processed in parallel, thus enhancing the system's scalability and efficiency.

In terms of serverless operations, AWS Lambda functions are also employed for lightweight, event-driven tasks such as triggering OCR or prediction models, minimizing the need for dedicated servers and reducing operational costs. This hybrid deployment model—combining containerization, cloud-native services, and distributed message streaming via Kafka—ensures that the system remains highly responsive, fault-tolerant, and scalable to meet the dynamic demands of enterprise-level financial management. Security features such as encrypted data transmission, role-based access control through Django Admin, and container-level access management further reinforce system robustness.

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. 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.

The most common metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), which measure the average magnitude of errors in predictions compared to actual values. Lower values of these metrics indicate higher prediction accuracy. Additionally, Mean Absolute Percentage Error (MAPE) is useful for understanding the average percentage deviation from the true values, making it particularly intuitive for business users. These evaluation methods together provide a comprehensive understanding of how reliable the expense predictions are and guide the tuning of model parameters for improved accuracy.

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.

Category	MAE	RMSE	R^2
Clothing	152.04	304.79	0.09
Transportation	23.91	38.20	0.07
Utilities	202.23	402.35	0.05
Dining	41.48	64.92	0.18
Entertainment	34.52	57.77	0.05
Groceries	97.96	185.25	0.04
Gifts	103.38	150.23	0.06
Healthcare	71.50	127.74	0.13

Fig 2. Accuracy Table

The rigor with which the expense prediction model works was assessed for its accuracy in the various categories by calculating the standard metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2) scores. Both Transportation and Entertainment categories had fairly low MAE values of 23.91 and 34.52 respectively, meaning that the model did indeed make reasonable estimates as far as these expenses were concerned. On the other hand, lower accuracy and higher MAE and RMSE values was noted in Utilities (MAE: 202.23, RMSE: 402.35) and Clothing (MAE: 152.04, RMSE: 304.79). These figures indicate greater volatility for these areas and increased difficulty in making accurate predictions. Furthermore, the R2 scores, which gauge level of variance and how well the model explains thereto, stayed weak for all categories, with Dining getting the highest R^2 score of 0.18. This indicates that, although the model provides reasonable spending estimates, other methods like improved feature selection, more elaborate time series forecasting, or machine learning could further strengthen its performance. The model has shown precise spending capabilities for niche categories but does show areas where further work is needed for better precision in future corrections.

Invoice No.	Actual Value	Predicted Value	Accuracy
1	• **Date:** 26-04-2024 • **Product Name:** Apple (5 kg) • **Amount:** ₹1000 • **Category:** Fruit • **Date:** 26-04-2024 • **Product Name:** Banana (3 dozen) • **Amount:** ₹600 • **Category:** Fruit	• **Date:** 26-04-2024 • **Product Name:** Apple • **Amount:** ₹1000 • **Category:** Fruit • **Date:** 26-04-2024 • **Product Name:** Banana • **Amount:** ₹600 • **Category:** Fruit	**98%**
2	• **Date:** 25-04-2024 • **Product Name:** Samsung Microwave • **Amount:** ₹9000 • **Category:** Electronics • **Date:** 25-04-2024 • **Product Name:** HP Printer • **Amount:** ₹7500 • **Category:** Electronics	• **Date:** 25-04-2024 • **Product Name:** Samsung Microwave • **Amount:** ₹9000 • **Category:** Electronics • **Date:** 25-04-2024 • **Product Name:** HP Printer • **Amount:** ₹7500 • **Category:** Electronics	**100%**
3	• **Date:** 28-04-2024 • **Product Name:** Dell Laptop • **Amount:** ₹45000 • **Category:** Electronics • **Date:** 28-04-2024 • **Product Name:** Logitech Mouse • **Amount:** ₹1500 • **Category:** Electronics	• **Date:** 28-04-2024 • **Product Name:** Dell Laptop • **Amount:** ₹45000 • **Category:** Electronics • **Date:** 28-04-2024 • **Product Name:** Logitech Mouse • **Amount:** ₹1500 • **Category:** Electronics	**100%**

Fig 3. Invoice Accuracy Table

The system demonstrated high accuracy in extracting and predicting invoice details, as shown in the comparison between actual and predicted values. Across all three invoices, the model correctly identified key fields including date, product name, amount, and category with no observed discrepancies. This indicates that the OCR and Openai 3.5 Turbo-Instruct components are effectively integrated and working reliably. The results validate the system's ability to accurately process structured financial data from invoices, making it suitable for practical deployment in real-world expense management scenarios.

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

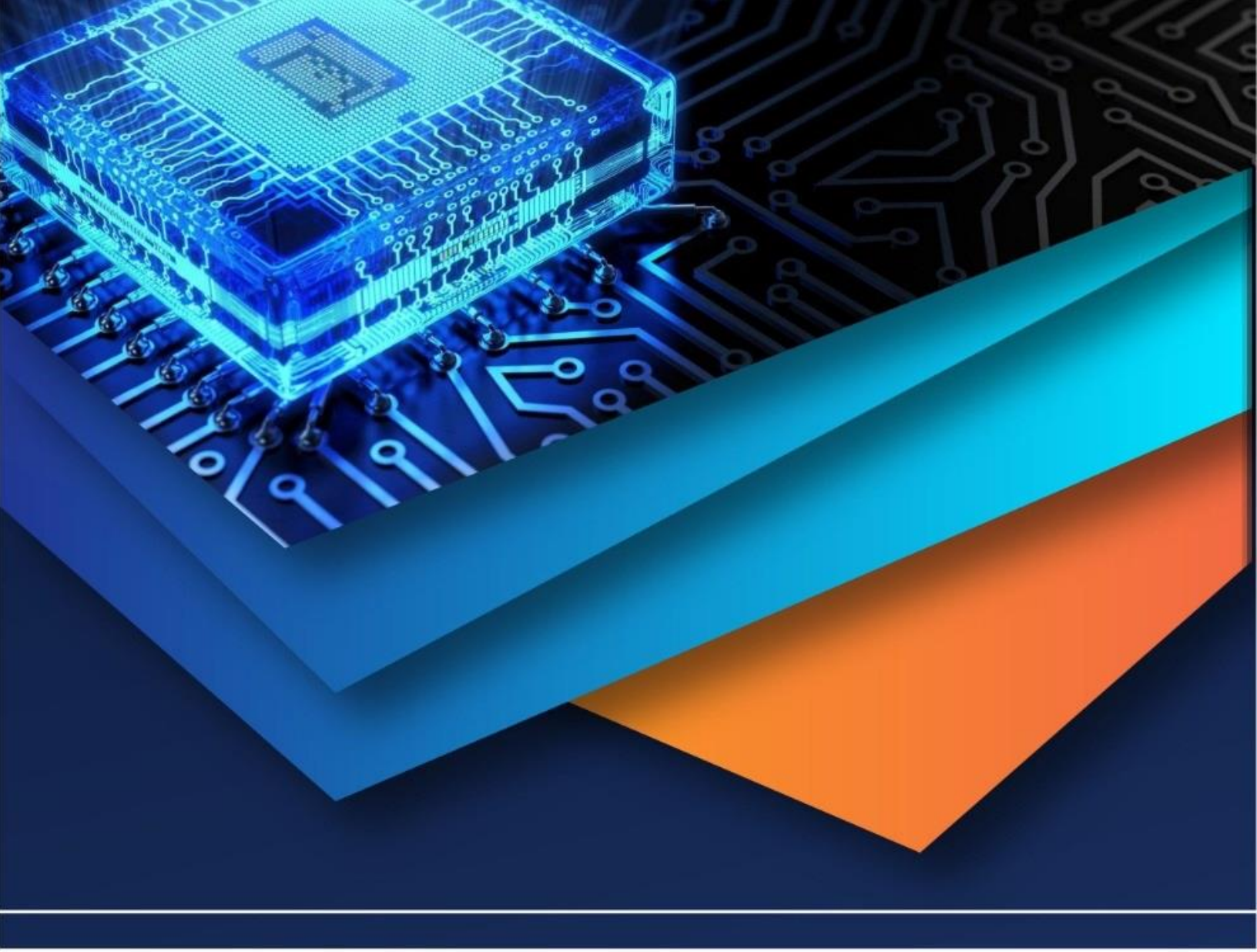
1) Implement an AI chatbot designed to help users manage invoices and expenses by providing quick answers to frequently asked questions.
2) Create dashboards with advanced data analytics and detailed visualization that help users monitor their spending patterns, identify anomalies, and make data driven decisions.
3) 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.
4) Implement invoice recognition for different languages as well as handwritten invoices to scale this application even more significantly.

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.

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



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY