AI-Driven OCR and Facial Landmark-Based Liveness Detection for Aadhaar Verification for Government Sector

S L Vedha Vidhya[#1], S Sharmila[#2], S M Baseetha[#3], J S SathishKumar[#4], Geetha G[5]
[1-5]*UG Student, Department of AI&DS, Dhirajlal Gandhi College of Technology, Salem, Tamilnadu, India*

***Abstract*—This project introduces an intelligent, automated system designed to streamline the processing of loan applications by integrating Optical Character Recognition (OCR), Natural Language Processing (NLP), and Flask-based web development. Users can complete a digital loan application form and upload their Aadhaar card as a supporting document. Upon submission, Tesseract OCR is employed to extract key information such as the applicant's name and address from the uploaded Aadhaar card. Using NLP techniques and regular expressions, the extracted data is cleaned and refined to accurately isolate relevant details, which are then cross-verified against the information provided by the user to ensure authenticity. The verified data is systematically organized and compiled into a downloadable PDF report for documentation purposes. Additionally, geolocation services are incorporated to validate address details using GPS coordinates, adding an extra layer of verification. The entire system enables faster processing, minimizes human error, and strengthens data verification through fuzzy matching and location-based intelligence. This project showcases how AI-driven document automation can significantly optimize financial workflows and enhance the reliability of loan application procedures. The platform is built to be scalable, secure, and adaptable, making it highly suitable for a wide range of digital verification applications, especially in fintech and e-governance domains.**

Keywords-OCR, NLP, Fuzzy Matching, GPS based address verification, Flask web application

I.INTRODUCTION

In today's rapidly evolving digital landscape, financial institutions are increasingly turning to automated technologies to enhance efficiency and deliver better user experiences [1]-[3]. A crucial aspect of this transformation is the loan application process, which traditionally involves heavy documentation and manual checks. This project presents an automated system for loan applications and Aadhaar card verification, leveraging Optical Character Recognition (OCR), Natural Language Processing (NLP), and modern web technologies to streamline and speed up the workflow. Applicants can complete an online form and upload their Aadhaar card as proof of identity and address [4]-[6]. The system utilizes the Tesseract OCR engine to extract essential details such as name, date of birth, and address from the uploaded Aadhaar image [7]-[10]. Extracted information is then refined and validated using NLP techniques and regular expressions, ensuring accurate data matching with the user's submitted form. Fuzzy matching algorithms further improve verification by accommodating minor discrepancies [11]-[13]. Each application generates a downloadable PDF report, and the applicant's GPS location is captured to verify the provided address. Built with Flask, this solution highlights the power of AI-driven automation in financial services by reducing manual workload, minimizing errors, and delivering faster, more secure loan processing [14]-[15].

II. ALGORITHM

- Tesseract OCR: Used to extract textual information from Aadhaar card images or PDF documents.
- Regular Expressions (Regex): Applied to clean and extract specific address components from the OCR-extracted text.
- Geopy (Nominatim API): Utilized to translate GPS coordinates into a readable address format.
- Set Intersection Matching: Implemented to compare the Aadhaar address with the GPS-derived address by identifying common area names.

III.PROPOSED SYSTEM

The proposed system leverages cutting-edge deep learning methods, particularly transformer-based Large Language Models (LLMs) such as BERT, T5, and GPT, to automatically create precise and well-structured abstracts and titles for research papers. Moving beyond traditional extractive summarization, this system adopts an abstractive approach, enabling it to grasp the semantic meaning and context of a document and generate summaries and titles that resemble human writing.

The model is trained on an extensive collection of research papers from multiple fields, allowing it to master domain-specific terminology, writing styles, and key idea extraction. Users can easily upload their research documents in PDF or text formats through an intuitive interface. The system then processes the content and delivers high-quality abstract and title suggestions in seconds, greatly minimizing manual work and enhancing the quality and presentation of academic writing.

Benefits of the Proposed System:

- Delivers context-aware, meaningful abstracts and title suggestions.
- Reduces the time and effort needed for academic writing.
- Adapts to domain-specific vocabulary for greater accuracy.
- Utilizes abstractive summarization for more natural and coherent outputs.
- Offers a user-friendly platform with quick and reliable results.

IV. SYSTEM REQUIREMENTS

4.1. Hardware Requirements:

- Processor: Intel Core i3 or better
- RAM: At least 4 GB
- Storage: Minimum of 2 GB free disk space
- GPU: (Optional, but enhances performance) NVIDIA GPU with CUDA support

4.2. Software Requirements:

- Operating System: Windows 10, Ubuntu 18.04, or newer versions
- Programming Language: Python 3.8 or later

4.3. Libraries and Frameworks:

- Transformers Library: (e.g., HuggingFace Transformers)
- Deep Learning Framework: PyTorch or TensorFlow (depending on model preference)
- Machine Learning Tools: Scikit-learn
- Natural Language Processing: NLTK or SpaCy (for text preprocessing)

4.4. PDF/Text Parsing Tools:

- PyMuPDF or pdfplumber (for content extraction from PDF files)

4.5. Development Environment:

- Visual Studio Code, Jupyter Notebook, or PyCharm

4.6. Internet Requirement:

- Needed for downloading pre-trained models and related dependencies

V.OUTPUT

This section includes fields for Email ID, Address, Aadhaar Card upload, Loan Amount Requested, and Purpose of Loan. A prominent "Submit" button allows the applicant to submit the form. Below this, there is an additional "Verify Address" section with a "Check Address" button, which provides an extra step for validating the entered address, ensuring enhanced data accuracy and reliability.

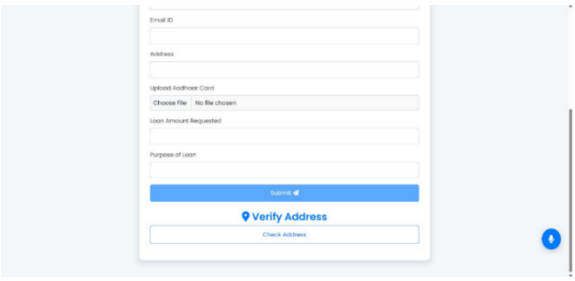

Fig.1.

It features the title and icon at the top, followed by input fields for Full Name, Date of Birth, Aadhaar Number, Mobile Number, Email ID, and Address. This section is designed to collect essential personal and identification information from the applicant in a clear and structured manner.

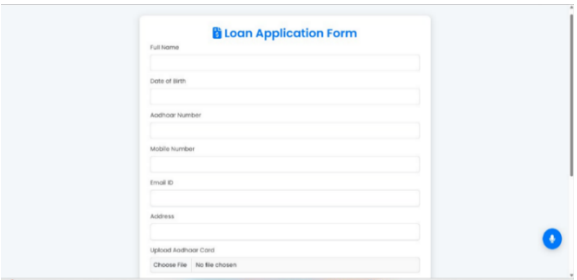

Fig.2.

VI.CONCLUSION

The creation of a Legal Chatbot based on the Indian Penal Code (IPC) using Natural Language Processing (NLP) and deep learning represents a major step forward in legal technology. This project simplifies access to complex legal information by offering a user-friendly platform capable of understanding and responding to queries in natural language. By integrating powerful tools such as PyMuPDF for extracting text, HuggingFace Transformers for contextual understanding, and multilingual functionality, the system delivers quick, accurate, and relevant legal guidance. The chatbot automates the process of retrieving and explaining IPC sections, associated punishments, and legal interpretations, making it a valuable resource for the general public, legal practitioners, students, and law enforcement personnel. It provides users with timely legal information without requiring expertise in legal terminology or extended consultation periods. Its multilingual features further promote inclusivity, catering to India's diverse linguistic population. This project highlights the transformative potential of AI and NLP in modernizing the legal sector by making it more transparent, efficient, and accessible. In addition to easing the burden on legal professionals, it fosters greater legal awareness and literacy among citizens. With ongoing development and integration, such systems could be expanded to cover broader areas of law and support activities like legal documentation and judicial assistance.

REFERENCES

[1] Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). "LEGAL-BERT: The Muppets straight out of Law School," arXiv preprint arXiv:2010.02559.

[2] Nguyen, H.-T., Phi, M.-K., Ngo, X.-B., Tran, V., Nguyen, L.-M., & Tu, M.-P. (2022). "Attentive Deep Neural Networks for Legal Document Retrieval," arXiv preprint arXiv:2212.13899.

[3] Wang, Q., Zhao, K., Amor, R., Liu, B., & Wang, R. (2022). "D2GCLF: Document-to-Graph Classifier for Legal Document Classification," Findings of the Association for Computational Linguistics: NAACL 2022, pp. 2208–2221.

[4] Nguyen, H.-T., Nguyen, M.-P., Vuong, T.-H.-Y., Bui, M.-Q., Nguyen, M.-C., Dang, T.-B., Tran, V., Nguyen, L.-M., & Satoh, K. (2022). "Transformer-based Approaches for Legal Text Processing," arXiv preprint arXiv:2202.06397.

[5] Shaheen, Z., Wohlgenannt, G., & Filtz, E. (2020). "Large Scale Legal Text Classification Using Transformer Models," arXiv preprint arXiv:2010.12871.

[6] Kore, R. C., Ray, P., Lade, P., & Nerurkar, A. (2020). "Legal Document Summarization Using NLP and ML Techniques," International Journal of Engineering and Computer Science, vol. 9, no. 05, pp. 25039–25046.

[7] Dimlioglu, T., Wang, J., Bisla, D., & et al. (2023). "Automatic document classification via transformers for regulations compliance management in large utility companies," Neural Computing and Applications, vol. 35, pp. 17167–17185.

[8] Wan, L., Papageorgiou, G., Seddon, M., & Bernardoni, M. (2019). "Long-length Legal Document Classification," arXiv preprint arXiv:1912.06905.

[9] Tuteja, M., & González Juclà, D. (2023). "Long Text Classification using Transformers with Paragraph Selection Strategies," Proceedings of the Natural Legal Language Processing Workshop 2023, pp. 17–24.

[10] Sharma, S., & Sharma, S. (2023). "A Comprehensive Analysis of Indian Legal Documents Summarization Techniques," SN Computer Science.

[11] Smith, R. (2007). *An Overview of the Tesseract OCR Engine*. In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007).

[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention Is All You Need.* arXiv preprint arXiv:1706.03762. (For NLP, Transformers background)

[13] Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment.* Computing in Science & Engineering.

[14] "Flask Documentation." Flask.palletsprojects.com

[15] Unique Identification Authority of India (UIDAI).