RESEARCH ARTICLE OPEN ACCESS

AI-Based Web System for Profile Summarization and Visual Recognition

Aishwariya Eswara V*, Poorana Senthilkumar S **
*(Department Of Computer Applications, Dr.N.G.P. Arts And Science College, Coimbatore, Tamil Nadu, India
Email: eswaravelumain@gmail.com)
** (Department Of Computer Applications, Dr.N.G.P. Arts And Science College, Coimbatore, Tamil Nadu, India
Email: pooranasenthilkumar@drngpasc.ac.in)

----------**********----------

Abstract:
Artificial Intelligence has enabled the development of intelligent web systems capable of performing automated data interpretation and visual analysis. This paper presents the design and implementation of an AI-Based Web System for Profile Summarization and Visual Recognition developed using Python (Flask framework), OpenCV, Dlib-based facial embeddings, Ultralytics YOLO model, and SQLite database. The system integrates four independent modules: Profile Summarization, Face Feed, Face Recognition, and Object Detection.The Profile Summarization module retrieves textual information from web sources and generates concise summaries using Natural Language Processing techniques. The Face Feed module captures and stores facial images along with metadata, while the Face Recognition module performs identity verification using 128-dimensional facial embeddings. The Object Detection module detects real-time objects using a pretrained YOLO model.The system demonstrates modular design, scalability, and practical implementation of AI techniques within a web-based environment.

Keywords — **Artificial Intelligence, Profile Summarization, Face Recognition, Object Detection, YOLO, Dlib, Flask Framework, OpenCV, NLP, Web-Based System.**

----------**********----------

I. INTRODUCTION

Artificial Intelligence has significantly influenced web application development by enabling automated content processing, image recognition, and intelligent decision-making. Modern web systems increasingly integrate multiple AI techniques to enhance functionality while maintaining architectural clarity and modular independence. The convergence of Natural Language Processing and Computer Vision has enabled applications to process textual and visual data efficiently within unified platforms.

With the growing availability of online information and real-time visual data, there is a need for systems that can intelligently summarize content and interpret visual inputs. Text summarization techniques assist in extracting meaningful insights from large textual sources, while visual recognition technologies enable automated identification and detection within image streams. Integrating these capabilities into a web-based system requires careful system design to ensure scalability, structured data flow, and controlled interaction between components.

The objective of this project is to design and implement a web-based AI system that combines profile summarization and visual recognition capabilities within a single platform. The system is developed using Python with the Flask framework for backend processing, OpenCV for image handling, Dlib-based facial embeddings through the face_recognition library, Ultralytics YOLO for object detection, and SQLite for user authentication management. This structured approach enhances maintainability, clarity, and scalability while

demonstrating practical implementation of AI techniques in a modular web system.

II. RELATED WORK

Several research studies have explored independent domains of text summarization, facial recognition, and object detection.

Text summarization techniques include extractive and abstractive approaches using Natural Language Processing. Earlier methods relied on statistical scoring, while recent approaches apply transformer-based models for semantic summarization.

Facial recognition systems commonly use Dlib's 128-dimensional facial embeddings for identity verification. These embeddings provide robust facial feature representation and are widely adopted in biometric authentication systems.

Object detection models have evolved from traditional Haar cascades to deep learning-based approaches such as YOLO (You Only Look Once), which provides real-time detection with high efficiency.

However, most existing systems focus on a single AI domain. Integrated web-based platforms that combine textual summarization and visual intelligence often lack architectural clarity or introduce unnecessary inter-module dependencies.

III. RESEARCH GAPS

Component Area	*Observed Limitation*	*Identified Gap*
Profile Summarization	Static systems	Dynamic web-based summarization
Face Recognition	Standalone apps	Integrated real-time recognition
Object Detection	Independent modules	Identity-aware detection
AI Web Systems	Single-module focus	Unified multi-module architecture

IV. SYSTEM OVERVIEW

The proposed AI-Based Web System consists of an authentication mechanism followed by a dashboard that provides access to four independent modules. The authentication module verifies user credentials using a SQLite Users Database. Upon successful login, users are redirected to the dashboard, which does not maintain any database connection.

From the dashboard, users can access Profile Summarization, Face Feed, Face Recognition, and Object Detection modules independently.

The Profile Summarization module accepts a name or keyword, retrieves relevant data from web sources, and applies NLP-based summarization techniques to generate concise textual output.

The Face Feed module captures facial images via webcam, collects metadata, and stores images along with associated data in a dedicated Face Database.

The Face Recognition module captures live facial input, extracts 128-dimensional embeddings using Dlib, and compares them with stored embeddings from the Face Database to identify or label the face as unknown.

The Object Detection module processes live camera frames using a pretrained Ultralytics YOLO model and displays detected object labels in real time.

The system strictly ensures that only Face Feed and Face Recognition interact through the Face Database. No other module connections are permitted.

V. EXPERIMENTAL METHODS

The system was implemented using Python with the Flask framework to manage routing and server-side processing. SQLite was used for managing user authentication data.

For facial recognition, OpenCV was used for image capture, and Dlib's facial landmark detection combined with the face_recognition library was used to generate 128-dimensional embeddings. Stored embeddings were compared using Euclidean distance to determine identity matches.

For object detection, a pretrained Ultralytics YOLO model was integrated to detect objects from real-time video streams. The model processes frames and returns bounding boxes with corresponding labels.

Profile summarization was implemented using Natural Language Processing techniques, including text cleaning, tokenization, and summarization algorithms to generate concise summaries from retrieved web content.

VI. WORKING PRINCIPLE

The proposed AI-Based Web System operates through a structured and modular workflow beginning with user authentication. When a user accesses the system, login credentials are verified through the SQLite Users Database. Upon successful authentication, the user is redirected to the dashboard interface. The dashboard functions as a navigation layer that allows users to independently access one of the four available modules without establishing interdependencies between them.

When the Profile Summarization module is selected, the system accepts a user-provided keyword or name. It retrieves relevant textual information from web sources and processes the content using Natural Language Processing techniques. The extracted content undergoes preprocessing and summarization before presenting the final concise output to the user.

When the Face Feed module is selected, the system activates the webcam to capture facial images and collect user metadata. The captured image and associated information are stored in the Face Database, and facial embeddings are generated for future comparison.

If the Face Recognition module is selected, the system captures a live facial image, extracts 128-dimensional embeddings using Dlib through the face_recognition library, and compares them against stored embeddings in the Face Database. Based on similarity measurement, the system identifies the individual or marks the face as unknown.

When the Object Detection module is accessed, live camera frames are processed using a pretrained Ultralytics YOLO model. The system detects objects within each frame and displays labeled bounding boxes in real time without interacting with any database component.

The overall system flow is illustrated in Figure 1.

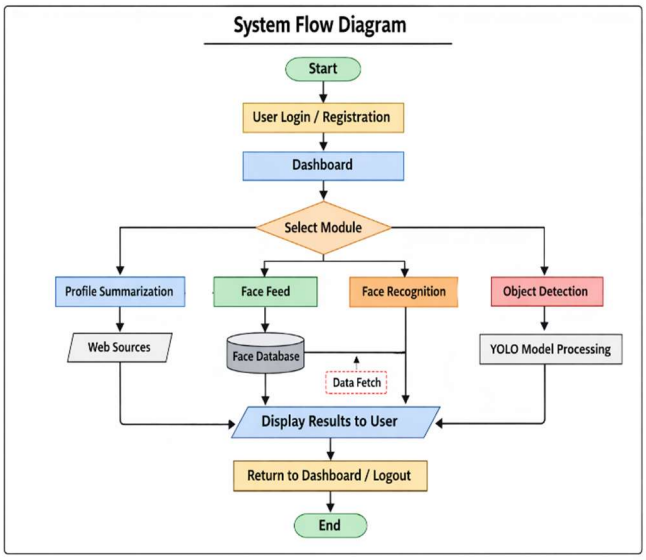

Fig 1. System Flow Diagram of the Proposed AI-Based Web System

VII. RESULTS AND DISCUSSION

The implemented system was tested module-wise to evaluate functional correctness, architectural separation, and processing performance. Each module operated independently as designed, and no unintended cross-module data exchange was observed. The authentication system successfully restricted unauthorized access. The dashboard effectively functioned as a control interface for accessing independent AI modules. The results obtained from each module are discussed below.

A. Profile Summarization Module

The Profile Summarization module successfully retrieved relevant textual data from web sources based on user input. The NLP-based summarization process reduced large textual content into concise summaries while maintaining contextual meaning. The preprocessing steps ensured removal of redundant information and irrelevant data. During testing, the module consistently generated readable summaries that preserved essential facts related to the input keyword. Since this module does not interact with the Face Database or other visual modules, architectural independence was maintained throughout execution.

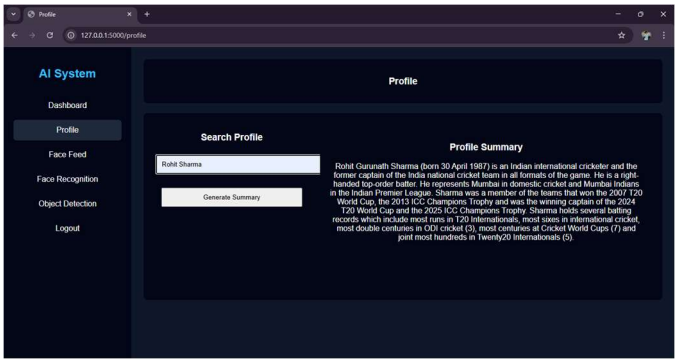

Fig 2. Output of Profile Summarization Module

B. Face Feed Module

The Face Feed module effectively captured facial images using a live webcam interface. The system collected user metadata and stored both the image files and associated information in the Face Database. Facial embeddings were generated at the time of storage to enable accurate future comparisons. The module demonstrated stable performance under normal lighting conditions. Since this module only writes data to the Face Database and does not access web sources or other modules, the system architecture remained consistent with the proposed design.

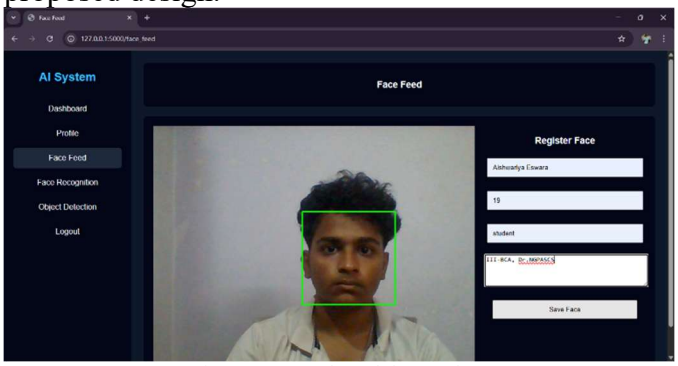

Fig 3: Face Feed Module Interface

C. Face Recognition Module

The Face Recognition module captured live facial input and extracted 128-dimensional embeddings using Dlib-based feature encoding. The extracted embeddings were compared with stored vectors from the Face Database using similarity measurement techniques. When a match was found within the defined threshold, the system displayed the recognized user details; otherwise, it labeled the face as unknown. Testing indicated reliable recognition performance under moderate lighting variations. The module strictly accessed only the Face Database and did not connect with web sources or object detection components.

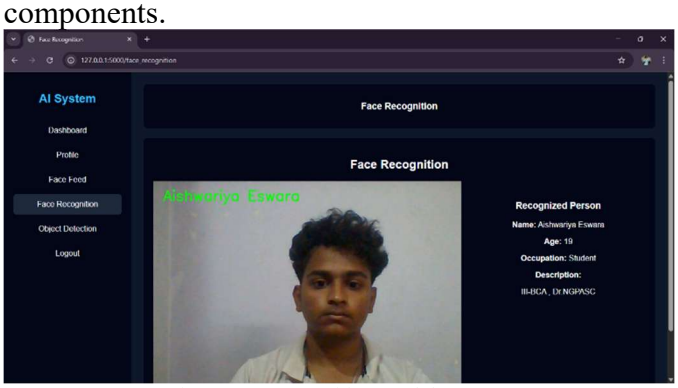

Fig 4: Face Recognition Module Output

D. Object Detection Module

The Object Detection module processed real-time camera frames using the pretrained Ultralytics YOLO model. The system accurately detected common objects and displayed labeled bounding boxes on the video stream. The detection process operated independently without any database interaction. During evaluation, the module maintained real-time responsiveness and produced consistent detection outputs. The complete independence of this module ensured adherence to the defined system architecture.

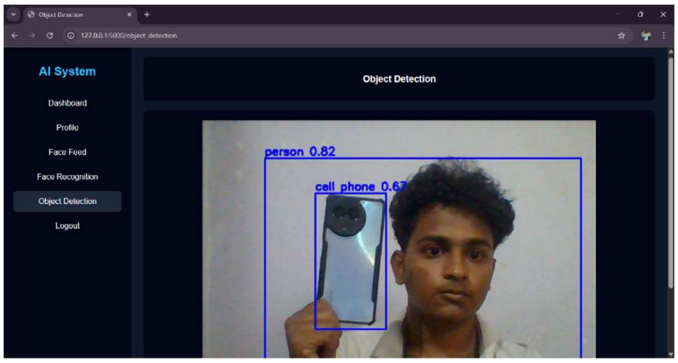

Fig 5: Object Detection Module Output

VIII. CONCLUSIONS

This paper presented the design and implementation of an AI-Based Web System for Profile Summarization and Visual Recognition developed using Flask, OpenCV, Dlib-based embeddings, YOLO, and SQLite. The system successfully

integrates textual and visual intelligence modules within a unified web interface while strictly maintaining architectural independence between components. The only permitted interaction between Face Feed and Face Recognition modules through the Face Database was implemented accurately. Experimental evaluation demonstrated functional correctness, modular clarity, and real-time performance. The proposed structure ensures maintainability, scalability, and structured AI integration suitable for academic and research applications.

IX. FUTURE ENHANCEMENTS

Future improvements to the system may include:

- Integration of transformer-based deep learning models for advanced text summarization
- Implementation of multi-face simultaneous recognition
- Deployment on cloud infrastructure for scalability
- Mobile-responsive interface development

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," *Advances in Neural Information Processing Systems*, 2017.
[2] R. Nallapati, B. Zhou, C. Gulcehre, and B. Xiang, "Abstractive Text Summarization using Sequence-to-Sequence RNNs and Beyond," *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016.
[3] A. See, P. J. Liu, and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," *Proceedings of ACL*, 2017.
[4] H. P. Luhn, "The Automatic Creation of Literature Abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.
[5] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.
[6] D. E. King, "Dlib-ml: A Machine Learning Toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.
[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, 2012.
[8] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.
[9] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," *British Machine Vision Conference (BMVC)*, 2015.
[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of IEEE CVPR*, 2016.
[11] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.
[12] G. Jocher et al., "Ultralytics YOLOv5," Ultralytics, 2020.
[13] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
[14] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2011.
[15] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, O'Reilly Media, 2018.
[16] A. B. Downey, *Think Python: How to Think Like a Computer Scientist*, O'Reilly Media, 2015.
[17] J. Hipp, D. Kennedy, and M. Mistachkin, "SQLite: A Serverless SQL Database Engine," *ACM SIGMOD Record*, vol. 35, no. 1, pp. 6–14, 2006.
[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[19] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed., Pearson Education, 2010.
[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Proceedings of ICLR*, 2013.

