



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 10, October 2026

CivicAI: AI-Based Government Scheme Eligibility Checker

Sreejesh M P¹, Vishal Kumar M P¹, K. Usha Rani²

U.G. Students, Department of Computer Science and Engineering, K.L.N. College of Engineering, Pottapalayam,
Sivangai, Tamil Nadu, India¹

Assistant Professor, Department of Computer Science and Engineering, K.L.N. College of Engineering, Pottapalayam,
Sivangai, Tamil Nadu, India²

ABSTRACT: Government welfare schemes in India provide scholarships, agricultural income support, pensions and health cover, but many eligible citizens cannot claim them because scheme information is scattered across departmental portals, eligibility rules are complex, and supporting certificates are verified manually. This paper presents CivicAI, an explainable web platform that checks eligibility for government schemes and verifies the supporting certificates automatically. Uploaded certificates (income, community, land Patta/Chitta, bonafide and Aadhaar) are preprocessed with OpenCV, read with Tesseract OCR and parsed with regular expressions to obtain values such as annual income, date of birth, community and land area. A deterministic rule engine compares these values with the scheme criteria and returns a parameter-by-parameter checklist, a projected benefit, and alternative schemes when the citizen does not qualify. The platform provides separate portals for students, farmers and senior citizens and an administrator dashboard, and is built with an Angular front end, a Flask REST back end and a MongoDB database. On a test set of 250 sample certificates, the OCR pipeline reached an overall field-extraction accuracy of 92.2%, and the average end-to-end processing time was about 1.3 s per document. All seven functional test cases passed.

KEYWORDS: Government Scheme Eligibility, Optical Character Recognition, Tesseract, Explainable AI, Document Verification, Decision Support System, Flask, Angular, MongoDB.

I. INTRODUCTION

Governments run welfare programs that support students, farmers and senior citizens through scholarships, income support, pensions and health cover. In India, schemes such as PM-KISAN [1], the scholarships offered through the National Scholarship Portal [2] and Ayushman Bharat PM-JAY [3] reach a large number of beneficiaries and are an important source of financial and medical security for low-income families.

In practice, many eligible citizens do not receive these benefits. Information about schemes is spread over many ministry and departmental websites, eligibility conditions are written in dense legal language that combines income limits, community category, age, landholding and educational status, and the required certificates must be collected from different offices such as the Tahsildar, the Village Administrative Officer and the educational institution. The submitted certificates are then checked manually, which causes delays, clerical errors and inconsistent decisions. As a result, deserving applicants are sometimes excluded because of lack of awareness, minor discrepancies in a certificate or administrative backlogs, while intermediaries may exploit gaps in verification.

This paper presents CivicAI, an AI-based government scheme eligibility checker that combines automatic certificate verification with explainable eligibility decisions. A citizen enters basic profile details and uploads certificates; the system reads them with OpenCV and Tesseract OCR, extracts the values needed for eligibility, evaluates them against the scheme rules and shows a parameter-by-parameter checklist that explains the result. The main contributions are: (i) an OCR pipeline with adaptive preprocessing and regular-expression parsing for income, community, land, bonafide and Aadhaar documents; (ii) a deterministic rule engine with explainable output, a projected benefit and alternative scheme suggestions; (iii) separate portals for students, farmers and senior citizens together with an administrator control center; and (iv) an evaluation of OCR accuracy and processing time, together with functional, performance, security and usability testing.

CivicAI is a decision-support and screening aid. It does not replace the official verification and sanction process; the final approval of benefits remains with the designated officers, who can review the audit record that the system stores for every evaluation.

The rest of this paper is organized as follows. Section II reviews related work. Section III describes the methodology and system architecture. Section IV presents the implementation, experimental results and testing. Section V concludes the paper and outlines future enhancements.

II. LITERATURE SURVEY

Decision support systems (DSS) help users reach decisions by analyzing information and presenting a recommendation, but their value depends on how users interpret and rely on the output. Saad and Meyer [4] studied people who made a series of decisions while observing the output of a DSS. They found that decisions largely followed normative rules but were based on the perceived accuracy of the DSS rather than its actual accuracy, and that most participants made up their minds early. These findings show that a DSS has to be designed with the user's perception in mind. For a welfare eligibility checker, this is a reason to show the evidence behind every decision instead of a bare "approved" or "rejected" message.

Optical character recognition (OCR) converts printed text in images into machine-readable text. Smith [5] describes the architecture of the Tesseract engine, including its line finding, feature classification and adaptive classifier. OCR accuracy depends strongly on image quality, and photographs of certificates taken with mobile phones often contain shadows, noise, uneven lighting and tilt. Preprocessing such as grayscale conversion, noise reduction and thresholding is therefore applied before recognition. Otsu's method [6] selects a global threshold automatically from the gray-level histogram and is widely used to separate text from the background, and the OpenCV library [7] provides these image-processing operations.

OCR output is only raw text; it does not show which number is an income, which date is a date of birth, or whether a document is of the expected type. Pattern-based extraction using regular expressions and document-type keywords can locate these fields even when the layout differs between issuing offices, but it can fail when the OCR misreads characters (for example, the letter "O" read as the digit "0") or when several values follow similar patterns. Extracted values therefore have to be validated before they are used in a decision.

Explainability and fairness are essential when automated systems affect access to public benefits. Doshi-Velez and Kim [8] discuss the need to define and evaluate interpretability rigorously, and Wachter et al. [9] show that counterfactual explanations, which tell a person what would have to change to obtain a different outcome, can be given without opening the black box of a model. Barocas et al. [10] discuss how decision systems learned from data can reproduce discrimination present in the data, which makes transparent, rule-based evaluation attractive for statutory eligibility checks. Table I summarizes how this work relates to CivicAI.

TABLE I. SUMMARY OF RELATED WORK AND ITS RELEVANCE TO CIVICAI

Ref.	Focus of the work	Relevance to CivicAI
[4] Saad and Meyer, 2025	Reliance on decision support output	Show the reasons behind each decision
[5] Smith, 2007	Architecture of the Tesseract OCR engine	Text extraction from certificates
[6] Otsu, 1979	Automatic global image thresholding	Binarization before OCR
[8] Doshi-Velez and Kim, 2017	Defining and evaluating interpretability	Design of the explainable checklist
[9] Wachter et al., 2018	Counterfactual explanations	"What to change" feedback after a rejection
[10] Barocas et al., 2023	Fairness in decision systems	Deterministic, auditable rule evaluation

The surveyed work treats decision support, OCR, explainability and fairness separately, and many welfare portals mainly digitize application forms while leaving the checking of certificates to officials. CivicAI brings these elements together for Indian welfare schemes in one platform: OCR-based certificate verification, a deterministic and auditable rule engine, counterfactual-style feedback with alternative scheme suggestions, and role-specific portals for the three main groups of beneficiaries.

III. METHODOLOGY

A. System Architecture

CivicAI follows a four-tier decoupled architecture, shown in Fig. 1. The presentation tier is a single-page application built with Angular 17 [11]; it provides the Student, Farmer, Senior Citizen and Administrator interfaces, validates forms in the browser and displays the results. The application tier is a Flask [12] REST gateway that handles authentication, file upload, scheme queries and audit logging. The AI and computer-vision tier performs image preprocessing, OCR, field extraction, rule evaluation and explanation generation using OpenCV 4.8, Tesseract OCR 5.3 (through pytesseract) and pypdf on Python 3.10. The persistence tier is a MongoDB [13] database that stores citizen profiles, extracted document data, the scheme repository and the evaluation logs. Because the tiers communicate only through defined interfaces, a component such as the OCR engine can be replaced without changing the others.

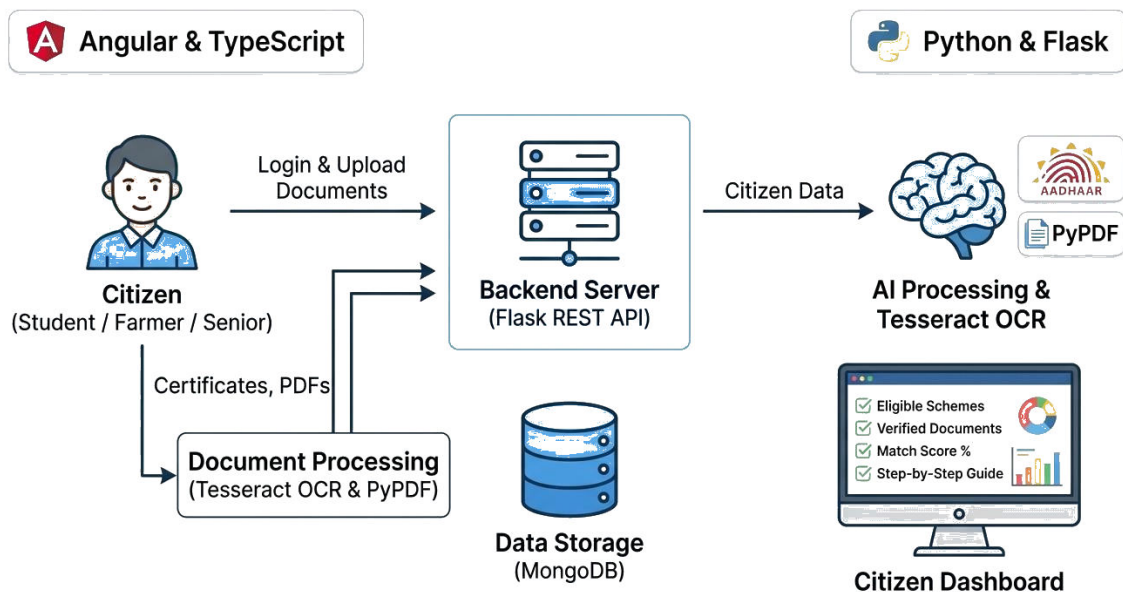


Fig. 1. System architecture of CivicAI.

B. Data Flow

Fig. 2 shows the data flow of the system. Citizens register and log in (process 1.0) and upload documents, which are processed by the OCR module (2.0). Administrators maintain the scheme catalog and its rules (3.0). The multi-criteria evaluation engine (4.0) combines the user profile, the verified OCR data and the scheme rules, and the result is returned to the citizen as eligible schemes and application guidance (5.0). Three data stores hold the user profiles (D1), the verified documents and OCR data (D2) and the government schemes repository (D3).

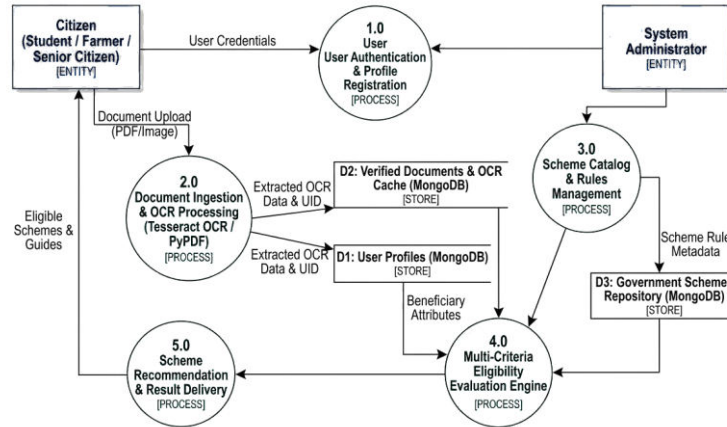


Fig. 2. Data flow diagram of CivicAI.

C. Document Preprocessing and OCR

Image files (PNG, JPG and JPEG) are first checked for quality: a heavily blurred photograph, detected by a low variance of the Laplacian (below 50), is rejected and the citizen is asked to upload it again. Accepted images are converted to grayscale, smoothed with a Gaussian filter and binarized with Otsu's threshold [6]; tilted photographs are deskewed with an affine transform. Tesseract [5] then reads the binarized image using page segmentation mode 6 for blocks of text and mode 11 for sparse, unstructured layouts. PDF files are read directly with the pypdf library.

D. Field Extraction and Document Validation

The OCR text is compared with keyword signatures defined for each document type (Aadhaar, income, community, bonafide, land Patta/Chitta, crop, age proof, pension and bank passbook) to confirm that the uploaded file is of the expected type. Regular expressions then extract the required values: annual income, date of birth or age, community, land area and certificate number. Each document receives a genuineness flag and an OCR confidence score; invalid or suspicious files are rejected with a clear message, and a certificate number that is already registered is flagged to prevent duplicate claims.

E. Eligibility Rule Engine and Explainable Output

Eligibility rules are stored as documents in the MongoDB scheme repository, so an administrator can change an income ceiling or add a scheme without changing the source code. For the selected scheme, the engine evaluates each criterion in turn, such as the income ceiling, community category, age, landholding, course level and required document. For every criterion, the result records the official threshold, the value verified from the certificate, a pass or fail status and a plain-language reason. Three kinds of explanation are produced: why the citizen qualified, shown as the verified attributes that satisfied the criteria; why the citizen was rejected, shown as the exact difference from the requirement (for example, an annual income of ₹2,80,000 exceeds the limit of ₹2,50,000 by ₹30,000); and, for a missing or illegible certificate, the authority from which it should be obtained. Because the rules are deterministic, two citizens with the same verified values always receive the same result. The complete workflow from login to result is given in Fig. 3.

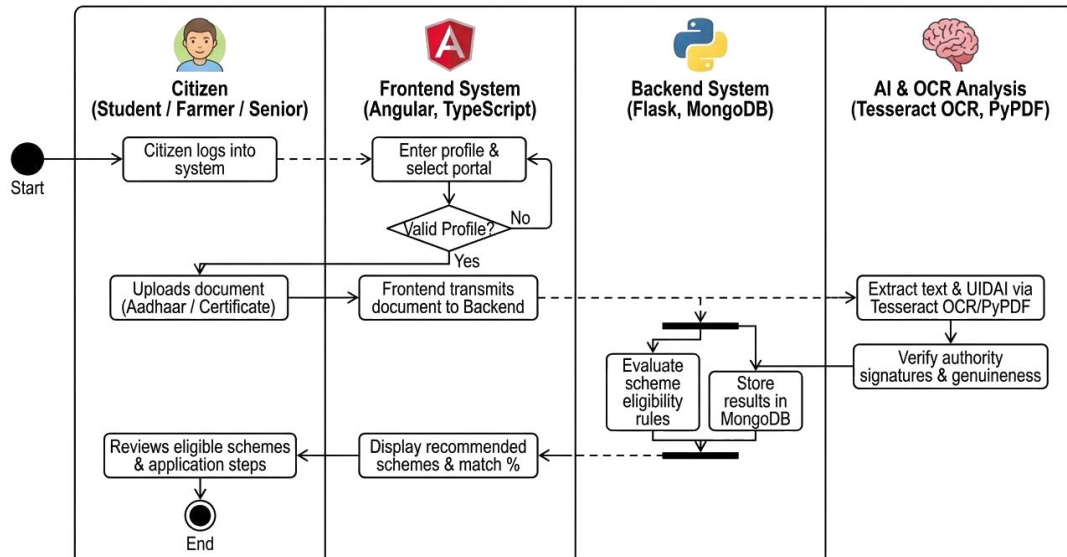


Fig. 3. Activity diagram of the document upload and scheme evaluation workflow.

F. Benefit Projection and Alternative Schemes

When a citizen is eligible, the payout rules stored with the scheme are used to project the benefit over several years; for example, PM-KISAN pays ₹6,000 per year in three installments of ₹2,000 [1]. When the citizen does not qualify, the engine evaluates the other schemes of the same portal, ranks those that the citizen satisfies by an eligibility match percentage, and lists schemes that were narrowly missed together with the condition that is not met. A rejection is therefore followed by a useful next option.

G. Security and Privacy

Aadhaar numbers are masked so that only the last four digits appear in the interface and the logs. Uploaded file names are sanitized with secure_filename to prevent directory traversal, cross-origin requests are restricted to authorized origins, and Angular's built-in sanitization protects the interface against script injection. Role-based access control separates citizen functions from administrative ones. Data is collected only for eligibility evaluation, following the consent and purpose-limitation principles of the Digital Personal Data Protection Act, 2023 [14].

IV. EXPERIMENTAL RESULTS

A. User Interfaces

The student portal (Fig. 4) lists the required certificates and shows the OCR verification status of every upload: genuine documents are marked as verified together with the OCR confidence, while unreadable or wrong-type documents are rejected with a reason. The eligibility result (Fig. 5) shows, for each scheme, the required condition, the value from the citizen's profile and a pass or fail mark, followed by the evaluation reason and any missing documents. The administrator control center (Fig. 6) shows the number of registered users by category, the number of active schemes and the stored user records, with search and category filters.

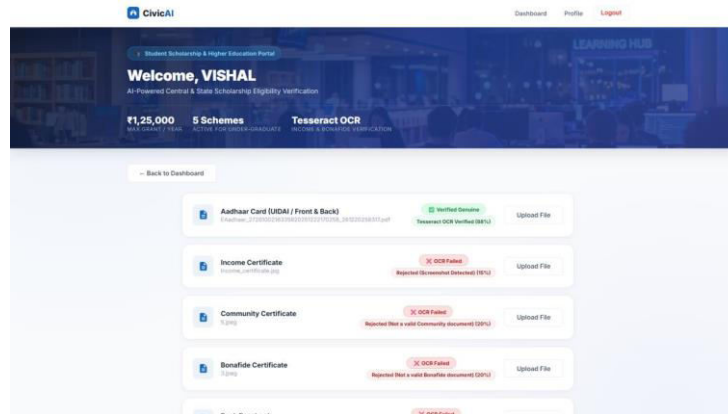


Fig. 4. Document verification page of the student portal (OCR analysis).

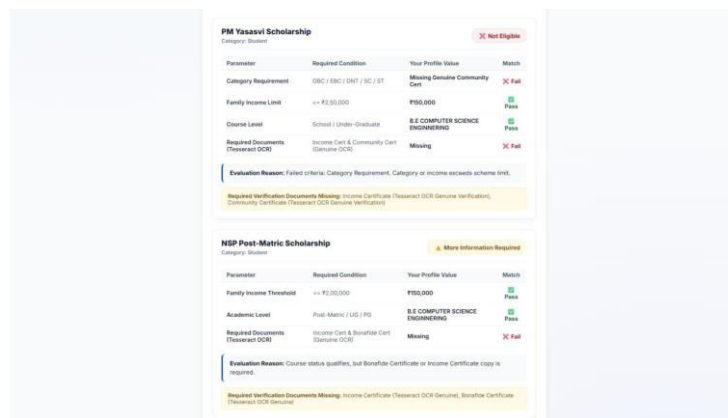


Fig. 5. Explainable AI rule reasoning and parameter checklist.

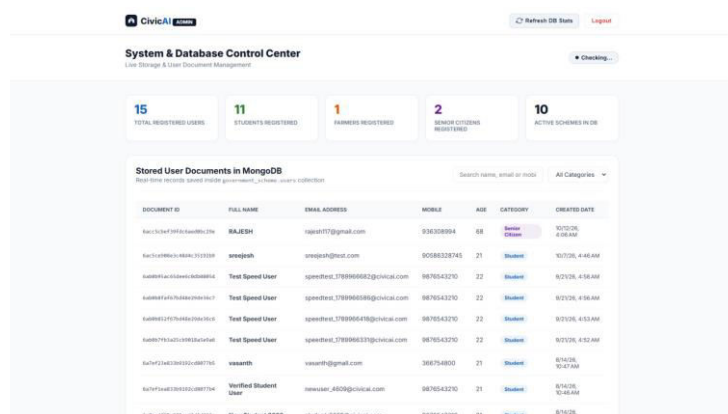


Fig. 6. Administrator system and database control center.

B. OCR Accuracy

The OCR pipeline was evaluated on 250 sample certificates of five types, based on standard revenue and institutional document formats from Tamil Nadu and the central ministries. Accuracy is the percentage of target fields extracted correctly. The results are given in Table II. The overall accuracy was 92.2%, with the highest value for the Aadhaar card (94.2%) and the lowest for the land Patta/Chitta (89.5%).

TABLE II. OCR FIELD-EXTRACTION ACCURACY BY DOCUMENT TYPE

Document category	Test samples	Target field extracted	Accuracy (%)
Income certificate	60	Annual income (₹)	91.8
Aadhaar identity card	50	Date of birth / age / gender	94.2
Land Patta / Chitta	50	Landholding (acres)	89.5
Bonafide certificate	45	College / course	93.1
Community certificate	45	Caste / sub-caste	92.6
Overall	250	All key parameters	92.2

C. Processing Time

The execution time of each pipeline stage was measured over 100 automated runs on standard server hardware (Table III). The average end-to-end time was 1,305 ms (about 1.3 s). Tesseract recognition took about 68% of this time, while the matching of scheme rules took only 22 ms.

TABLE III. AVERAGE LATENCY OF EACH PIPELINE STAGE

Pipeline stage	Average latency (ms)	Share of total time (%)
Image ingestion and OpenCV normalization	310	23.8
Tesseract OCR text extraction	890	68.2
Regex entity parsing and tokenization	45	3.4
Deterministic scheme rule matching	22	1.7
XAI reasoning and database persistence	38	2.9
Total end-to-end	1,305	100.0

D. System Testing

Unit tests covered the image preprocessing function, the regular-expression tokenizer (currency symbols, comma separators and the confusion between “O” and “0”), the boundary values of the income rule (₹2,50,000 against ₹2,50,001) and the Angular validators for the Aadhaar number, telephone number and e-mail address. Seven functional test cases covered scholarship eligibility of a student, PM-KISAN for a farmer with 2.8 acres, an IGNOAPS pension for a senior citizen aged 64, rejection with alternative schemes when the income exceeded the limit, rejection of a blurred document, an administrator update of an income ceiling, and prevention of a duplicate certificate upload. All seven test cases passed. Integration tests verified the multipart upload from Angular to Flask, the call from Flask to the Tesseract process, MongoDB connection pooling and consistent JSON error responses.

In a load test with Apache JMeter that simulated 200 concurrent sessions, the back end sustained about 48 requests per second with an average response time of 1.4 s and memory use below 450 MB. The security checks confirmed Aadhaar masking, file-name sanitization, restriction of cross-origin requests and protection against script injection. In usability testing, students, farmers and senior citizens using desktop browsers and low-cost smartphones could carry out the main tasks; they suggested clearer upload-progress indicators and simpler wording in some explanations, which are planned as refinements.

E. Discussion and Limitations

The results are based on sample certificates and a controlled test environment, so the accuracy on large collections of real photographs from different offices still has to be established. OCR remains the slowest and most error-prone stage, and the rule engine is only as correct as the rules stored in the repository, which must be updated whenever a scheme changes. CivicAI therefore works as a screening and decision-support aid, and the final sanction stays with the competent authority.

V. CONCLUSION AND FUTURE ENHANCEMENT

This paper presented CivicAI, an explainable system that checks eligibility for government schemes and verifies supporting certificates automatically. By combining OpenCV preprocessing, Tesseract OCR, regular-expression parsing and a deterministic rule engine, the system extracted key fields from 250 sample certificates with an overall accuracy of 92.2% and processed a document in about 1.3 s on average, and all functional test cases passed. The parameter-by-parameter checklist, the counterfactual-style feedback and the alternative scheme suggestions help citizens understand a decision and find other benefits they may qualify for, which can reduce exclusion caused by lack of awareness and incomplete documents. Further evaluation with real-world certificates and a pilot with a welfare department are needed before wider use.

In future work, CivicAI will be integrated with DigiLocker and the UIDAI Aadhaar e-KYC service so that digitally signed certificates can be fetched directly, which reduces the dependence on scanned images. A multilingual voice assistant for Tamil, Hindi, Telugu and Kannada will help citizens with limited literacy, and robotic process automation can pre-fill and submit applications on the official portals with the citizen's consent.

REFERENCES

- [1] Ministry of Agriculture & Farmers Welfare, Government of India, "PM-KISAN Samman Nidhi." [Online]. Available: <https://pmkisan.gov.in>
- [2] Government of India, "National Scholarship Portal." [Online]. Available: <https://scholarships.gov.in>
- [3] National Health Authority, Government of India, "Ayushman Bharat Pradhan Mantri Jan Arogya Yojana (PM-JAY)." [Online]. Available: <https://pmjay.gov.in>
- [4] Y. Saad and J. Meyer, "Aided decision processes in dynamic events: Measuring decision support systems' influence and human responsibility," *IEEE Trans. Human-Mach. Syst.*, vol. 55, no. 3, pp. 408–417, 2025.
- [5] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, Curitiba, Brazil, 2007, pp. 629–633.
- [6] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.
- [7] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [8] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," arXiv:1702.08608, 2017.
- [9] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harvard J. Law & Technol.*, vol. 31, no. 2, pp. 841–887, 2018.
- [10] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning: Limitations and Opportunities*. Cambridge, MA, USA: MIT Press, 2023.
- [11] Google, "Angular documentation." [Online]. Available: <https://angular.dev>
- [12] Pallets Projects, "Flask documentation." [Online]. Available: <https://flask.palletsprojects.com>
- [13] MongoDB, Inc., "MongoDB manual." [Online]. Available: <https://www.mongodb.com/docs/manual/>
- [14] Government of India, "The Digital Personal Data Protection Act, 2023," Act No. 22 of 2023.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details