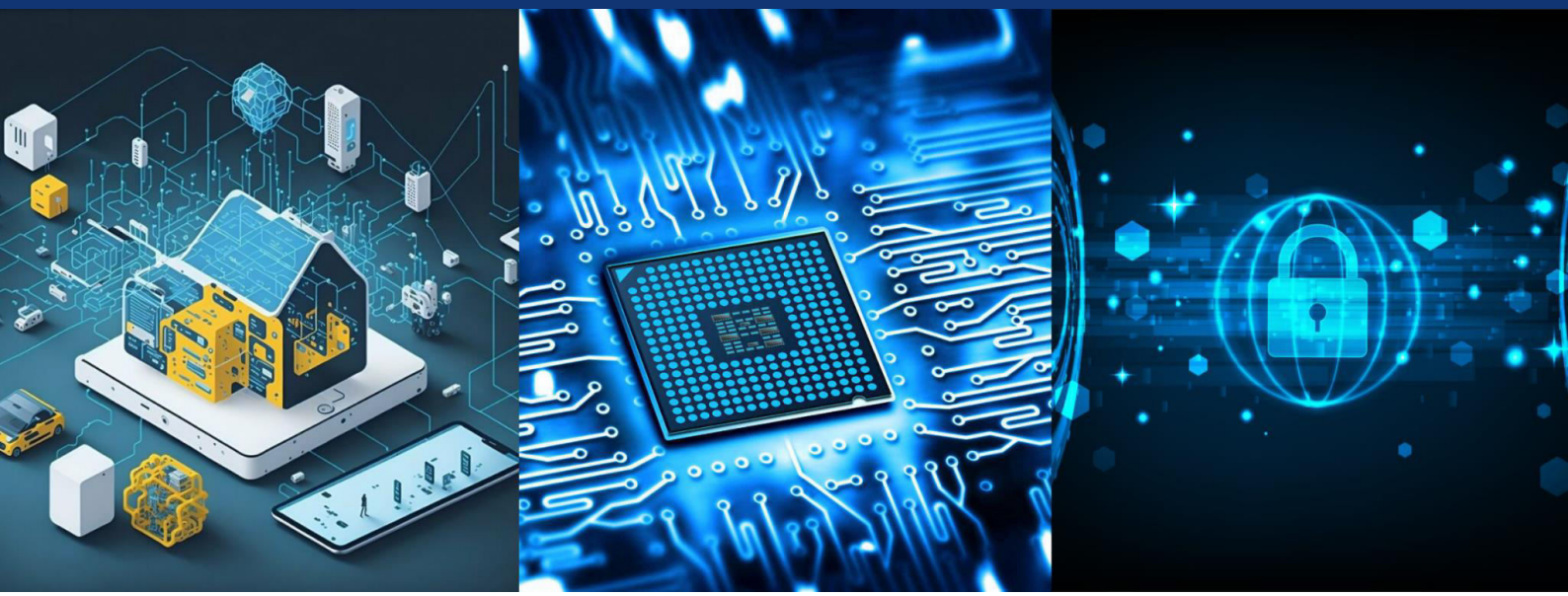
IJIRCCE

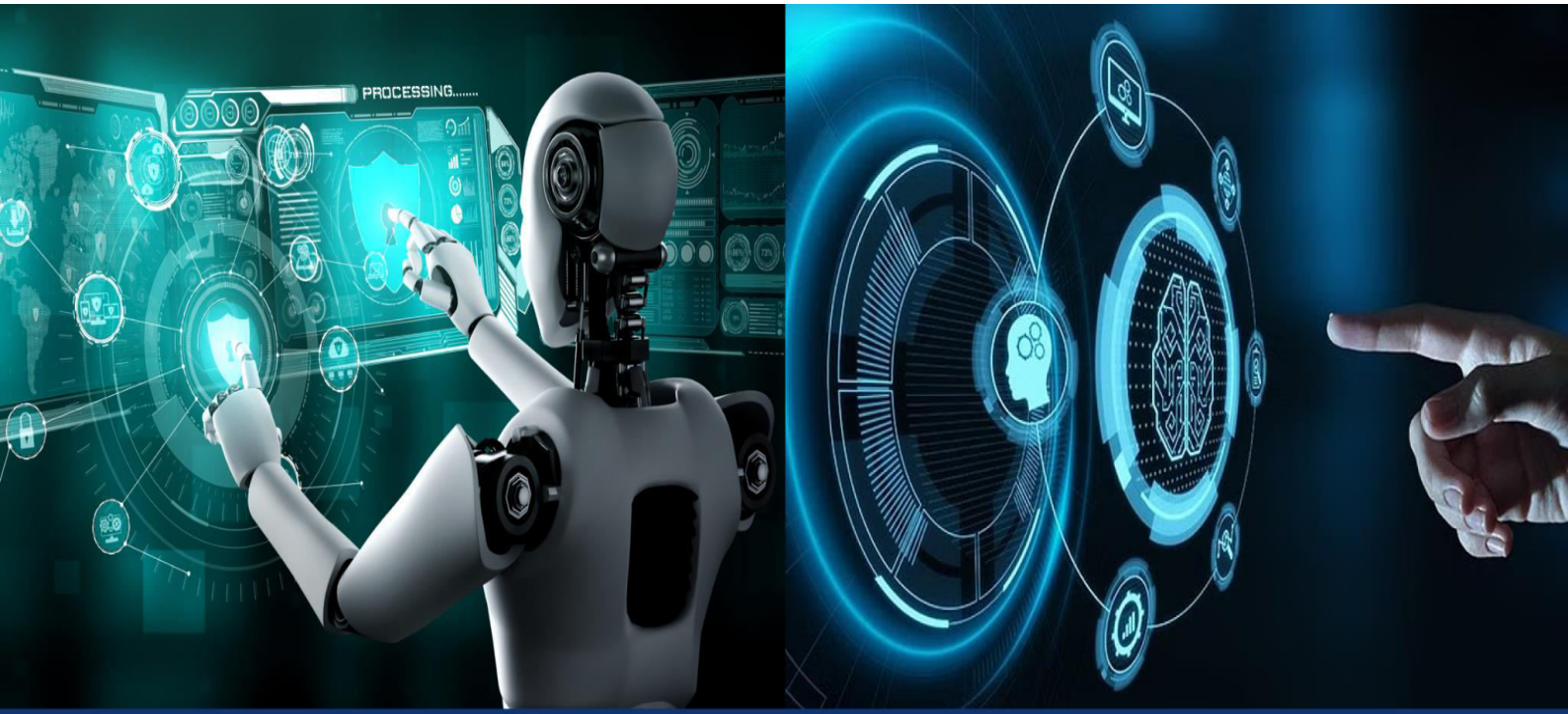
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

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



Impact Factor: 9.927

Volume 14, Issue 10, October 2026

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410022

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

Blood Group Detection Using Fingerprint Image

Bhoomika T R[1], Nisha K S[1], Pooja N E[1], Rani K J[1], Er.Ramesh Kumar H K[2]

Final-Year B.E. Students, Department of Computer Science and Engineering, STJ Institute of Technology, Ranebennur, Karnataka, India[1]

Assistant Professor, Department of Computer Science and Engineering, STJ Institute of Technology, Ranebennur, Karnataka, India[2]

ABSTRACT: Blood group identification is essential for transfusion, organ transplantation and emergency treatment. The standard serological method is accurate but invasive, slow, and dependent on trained staff and laboratory facilities. This paper presents a non-invasive desktop system that predicts a blood group label from a fingerprint image using a Convolutional Neural Network (CNN). Fingerprint images are converted to grayscale, resized to 128 × 128 pixels and passed to a three-block CNN (32, 64 and 128 filters) with global average pooling, dropout and a softmax output layer. The model is trained with the Adam optimizer and sparse categorical cross-entropy loss using an 80/20 stratified split and early stopping. The application is built in Python with TensorFlow/Keras, Pillow, NumPy, scikit-learn and a Tkinter graphical interface, and it reports the predicted class with a confidence score within seconds. The system was verified with 15 functional test cases covering dataset loading, training, prediction and error handling. The work is an experimental, dataset-based study and is not a medically validated blood group test.

KEYWORDS: Blood group detection, fingerprint, Convolutional Neural Network, deep learning, biometrics, non-invasive diagnosis.

I. INTRODUCTION

Blood group identification is a basic step in blood transfusion, organ transplantation and emergency care. Traditionally, the blood group is found by serological testing, in which a blood sample is mixed with anti-A, anti-B and anti-D antibodies and the agglutination reaction is observed. Although reliable, this method requires blood collection, laboratory equipment, reagents and trained technicians, and it can delay results when time is critical.

Recent progress in artificial intelligence and biometrics has encouraged non-invasive alternatives. Fingerprints are unique to each person and contain ridge patterns such as loops, whorls and arches. Several studies have explored whether these patterns can be linked with blood group, and CNNs are well suited to such image tasks because they learn ridge flow, texture and minutiae-like features automatically [1]–[4].

This paper describes a system that predicts the blood group (A, B, AB and O with Rh factor) from a fingerprint image using a CNN. The main objectives are:

- to predict blood group from fingerprint images without collecting blood samples;
- to apply a CNN for analysis of fingerprint patterns;
- to reduce the time and cost of conventional blood group testing; and
- to provide a simple, user-friendly interface for non-expert users.

II. LITERATURE SURVEY

Devi et al. [1] proposed a non-invasive system that captures fingerprints with a high-quality scanner and applies binarization, thinning and Gabor filtering to extract ridge patterns and minutiae points. The features are classified by a CNN, and photoplethysmogram (PPG) signals are added as a second biometric input, which improved performance over fingerprint data alone.

Alekhya et al. [2] used CNNs directly on fingerprint image datasets after preprocessing. They relate loops, whorls and arches to blood group and highlight the usefulness of the approach in emergencies and in remote areas with limited medical facilities.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

Sivamurugan et al. [3] improved prediction accuracy by optimizing the model architecture and training techniques, and suggested applications in blood donation camps and rural healthcare.

Rashmi et al. [4] combined fingerprint analysis with deep learning and used grayscale conversion, normalization and data augmentation. Performance was measured with accuracy, precision, recall and F1-score, and a web-based interface gave real-time prediction. The authors note that noisy images and limited datasets remain challenges.

In summary, the existing work agrees that CNN-based fingerprint analysis is fast, low-cost and non-invasive, while dataset size, image quality and clinical validation remain open issues. The present work contributes a complete, easy-to-use desktop implementation with a documented design, test plan and training pipeline.

III. EXISTING AND PROPOSED SYSTEM

A. Existing System

The existing system relies on serological laboratory tests and automated blood typing equipment. Its main disadvantages are:

- it is invasive and can cause pain and fear, especially in children and the elderly;
- it is time-consuming because of sample collection, preparation, testing and observation;
- it requires skilled personnel, as incorrect handling can give wrong results;
- it needs a laboratory setup, which is difficult in rural or remote areas;
- it is costly because of chemicals, equipment and maintenance; and
- it carries a risk of infection and sample contamination.

B. Proposed System

The proposed system takes a fingerprint image as input instead of a blood sample. Images are preprocessed (resizing, normalization, noise removal) and fed to a CNN trained on fingerprint images labelled with blood groups. After training, a new fingerprint image is classified into one of the blood group classes. Its advantages are that it is non-invasive, gives results within seconds, is cost-effective (only a scanner and a computer are needed), is easy to use, reduces manual error, and can process many inputs quickly in camps or hospitals.

IV. SYSTEM REQUIREMENTS

The functional requirements cover user input (JPG/PNG upload and optional scanner capture), image preprocessing (noise removal, enhancement, resizing, normalization, grayscale conversion), feature extraction, model processing with a trained CNN, and clear display of the predicted blood group. The non-functional requirements are performance (results within a few seconds), accuracy, reliability, usability and scalability.

Table I. Hardware and software requirements

Item	Specification
Processor / RAM / Storage	Intel i5 or higher / 8 GB (16 GB recommended) / 256 GB SSD
Operating system	Windows 10/11, macOS or Linux (Ubuntu)
Language	Python 3.11
Deep learning	TensorFlow / Keras
GUI	Tkinter
Libraries	Pillow, NumPy, scikit-learn, JSON
IDE	Visual Studio Code / PyCharm

V. SYSTEM DESIGN AND METHODOLOGY

A. System Architecture

The system is a desktop application written in Python that works in two phases: training and prediction. The Tkinter interface runs on the main thread and provides options to load the dataset, train the CNN, upload a fingerprint, preview

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

it and show the result. Training runs in a background thread so that the interface stays responsive. The trained model is saved as *fingerprint_cnn.keras* and the class names as *fingerprint_classes.json*.

B. Image Preprocessing

Every image, whether from the dataset or uploaded by the user, is converted to grayscale and resized to 128 × 128 pixels using Pillow. Pixel values are normalized by a rescaling layer (1/255) inside the model, so training and prediction use identical preprocessing. Supported formats are JPG, JPEG, PNG, BMP, TIF and TIFF; unreadable images are skipped and counted.

C. CNN Architecture

The CNN is trained from scratch and learns ridge and texture features automatically, without manual feature extraction. The layer configuration is given in Table II.

Table II. CNN layer configuration

Layer	Details
Input	128 × 128 × 1 grayscale image
Rescaling	1/255 normalization
Conv2D + MaxPooling	32 filters, 3 × 3, ReLU, same padding
Conv2D + MaxPooling	64 filters, 3 × 3, ReLU, same padding
Conv2D + MaxPooling	128 filters, 3 × 3, ReLU, same padding
Global Average Pooling	Reduces feature maps to a vector
Dropout	Rate 0.35
Dense	64 units, ReLU
Dropout	Rate 0.25
Dense (output)	Number of classes, Softmax

The softmax layer gives the probability of each blood group class, and the class with the highest probability is selected using argmax.

D. Training Procedure

The user selects a dataset folder that contains one sub-folder per blood group. Folder names such as A+, A-, B+, B-, AB+, AB-, O+ and O- (or A_positive, and so on) are mapped to class labels. The system checks that at least eight readable images and at least two classes exist, and that each class has at least two images. The data are split 80 % for training and 20 % for validation with a stratified split (random state 42) so that every class appears in both sets.

Table III. Training parameters

Parameter	Value
Image size	128 × 128
Optimizer	Adam
Loss	Sparse categorical cross-entropy
Batch size	16
Maximum epochs	12
Early stopping	Monitor val_loss, patience 3, restore best weights
Train / validation split	80 % / 20 % (stratified)

Adam [5] adapts the learning rate for each parameter, and early stopping reduces overfitting. After training, the validation accuracy is displayed, and the model and class names are saved for later use.

E. Prediction Phase

The user uploads a fingerprint image through the interface. The image is converted to grayscale, resized to 128 × 128 and given to the trained model held in memory. The predicted blood group and its confidence score are displayed in the Prediction Result area. If no model has been trained, a warning asks the user to load and train a dataset first.

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410022

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

F. Software Methodology

The project follows the Prototyping Model with six stages: requirement analysis, quick design, prototype building, user evaluation, refinement and final product. A CNN prototype with a simple interface was built first, tested, and improved repeatedly until the objectives were met. The design was documented with use case, sequence, class, data flow and activity diagrams.

VI. IMPLEMENTATION

The application is implemented in Python 3.11. TensorFlow/Keras builds and trains the CNN, Pillow reads and resizes images, NumPy converts images to arrays, scikit-learn provides *train_test_split*, and Tkinter provides buttons, file dialogs, previews and message boxes. The main modules are:

- *build_cnn()* creates the sequential CNN and compiles it;
- *load_images_from_folder()* reads class sub-folders, maps folder names to labels and ignores unreadable files;
- the training routine splits data, fits the model with early stopping, evaluates validation accuracy and saves the model and class names; and
- the prediction routine preprocesses the uploaded image, calls *model.predict()* and decodes the result with argmax.

The object design uses a main application class that controls the GUI and coordinates the CNNModel, DatasetLoader, ImageUtils and PredictionResult classes.

VII. TESTING

Unit testing checked functions such as *build_cnn()*, *load_images_from_folder()* and the prediction routine. Integration, functional, validation, performance, GUI and error-handling tests were also carried out. Selected test cases are shown in Table IV.

Table IV. Selected test cases

ID	Test scenario	Expected result
TC01	Start the application	Main window opens
TC02	Select a valid dataset folder	Dataset loaded and training starts
TC03	Select an empty dataset folder	Error message displayed
TC04	Dataset with only one class	Training rejected
TC05	Class with fewer than two images	Validation error displayed
TC07	Train the CNN model	Training completes; validation metrics shown
TC08/09	Save model and class names	.keras and .json files created
TC11	Predict an uploaded fingerprint	Predicted class and score displayed
TC12	Upload an unreadable image	Prediction error handled
TC13	Upload an image before training	Warning asks user to train first
TC14/15	Unsupported or unreadable files in dataset	Ignored; unreadable count reported

VIII. RESULTS AND DISCUSSION

The interface in its initial state shows the *Load Dataset and Train CNN* and *Upload Fingerprint* buttons, an empty fingerprint preview and the message "Awaiting input…". After a dataset of 66 fingerprint images covering the blood group classes A+, A-, AB+, AB-, B+, B-, O+ and O- was loaded and trained, the interface reported the number of images, the available groups and zero unreadable images. The validation accuracy obtained on the 20 % hold-out set was [XX.XX] %.

When a fingerprint image was uploaded, it was shown in the preview panel and the trained CNN displayed its prediction in the Prediction Result section. In the demonstrated example, the predicted blood group was A+. Prediction takes only a few seconds, compared with the multi-step procedure of laboratory testing. All 15 test cases produced the expected behaviour, including correct handling of empty datasets, single-class datasets, unreadable images and prediction before training.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

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

The validation accuracy shown by the application is an experimental indicator only. With a small dataset, a CNN trained from scratch may overfit, and results may not carry over to unseen persons. Larger and more varied datasets, data augmentation and subject-level train/test separation are therefore needed before the model can be considered reliable.

IX. LIMITATIONS

The system gives an experimental, dataset-based prediction and is **not a medically validated blood group test**. It must not be used for transfusion or any clinical decision. The reported performance depends on the size and quality of the dataset and on how the labels were obtained. Validation should be done on fingerprints collected from volunteers whose blood groups are confirmed by laboratory tests, with separate persons in the training and test sets.

X. CONCLUSION AND FUTURE SCOPE

This paper presented a non-invasive, low-cost desktop system that predicts blood group from a fingerprint image using a CNN. The system covers dataset loading, preprocessing, training, saving, prediction and a simple GUI, and it handles errors such as invalid datasets and unreadable images. It shows how deep learning can be used to build quick, user-friendly screening tools.

Future work includes collecting a larger laboratory-verified dataset, adding data augmentation and transfer learning, using additional biometric signals such as PPG [1], reporting precision, recall and F1-score [4], providing a web or mobile interface, and capturing fingerprints in real time from a scanner.

REFERENCES

[1] R. D. Devi, K. Likhitha, M. Varsha, M. Sathwika and K. Balamani, "Blood Group Detecting Using Fingerprint," *IEEE International Conference on Sustainability, Innovation & Technology (ICSIT)*, 2025.
[2] J. L. Alekhya, B. Choppara, E. Jaswanth and N. Jaswanth, "Blood Group Detection Using CNN from Fingerprint Images," *2026 IEEE International Students' Conference on Electrical, Electronics and Computer Science*, 2026.
[3] C. Sivamurugan, V. K. Murthi, B. Perumal, Y. N. Sankar, Y. Siddarthha and Y. Varun, "Enhanced Blood Group Prediction with Fingerprint Images using Deep Learning," *IEEE*.
[4] Rashmi K. A., Divya M., Chethana T., Hadiya Amber and Divya Shree K. S., "Blood Group Detection Using Fingerprint," *3rd International Conference on Knowledge Engineering and Communication Systems (ICKECS)*, 2025.
[5] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.
[6] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.



INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details