



Ensemble Learning For Blood Group Detection Using Fingerprints With Custom CNN And EfficientNetB0 In A CLAHE–TensorFlow–Keras–OpenCV Pipeline And Django Web Deployment

Srishti Rohila, Sneha Jain, Priyanshi Sharma, Uffiyan Nizam

Student

Computer Science and Engineering (Data Science),
Meerut Institute of Engineering and Technology, Meerut, India

Abstract: Safe transfusion practice, transplant surgery and many forensic tasks depend on knowing a person's blood group in advance. Conventional serological typing, although reliable, involves drawing blood, using reagents and operating in a laboratory environment, which is often impractical in emergencies or low-resource areas. This work explores a fingerprint-based, non-invasive alternative built on an AI pipeline that combines a Custom Convolutional Neural Network (CNN) with EfficientNetB0 in a feature-fusion ensemble for eight-class blood group prediction (A+, A-, B+, B-, AB+, AB-, O+, O-). Incoming fingerprint images are first enhanced using CLAHE and then processed along two parallel branches: a pre-trained Custom CNN operating on 64×64 inputs and an EfficientNetB0 backbone receiving 224×224 inputs, whose latent feature vectors are concatenated and passed through fully connected layers for classification. The trained ensemble is deployed inside a Django web application that supports fingerprint upload, automated prediction, storage of patient information, dashboard-based analytics and PDF report generation. Experiments on a publicly available Kaggle dataset of 6,000 fingerprint images yield a validation accuracy of about 93.3% and test accuracy in the 90–91% range, suggesting that the proposed framework can function as an effective screening tool under controlled conditions. In addition, the backend exposes an API for integration with ESP32–R307S fingerprint hardware, providing a pathway toward future real-time clinical and field deployments.

Index Terms – Fingerprint, blood group detection, Custom CNN, EfficientNetB0, ensemble learning, CLAHE, Django, medical AI.

I. INTRODUCTION

Blood group compatibility is a prerequisite for the safe transfusion of blood and its components, as well as surgical procedures to be performed with minimal risk, and organ transplantation. The gold standard, serological methods that detect ABO and Rh antigens require blood samples, reagents and laboratory infrastructure, making them invasive, resource-intensive cost-prohibitive to implement and slow during emergency situations or in low-resource environments. Such limitations drive the pursuit of other non-invasive screening systems that could provide rapid blood group predictions on blood, supporting clinical decision-making.

Fingerprints have established themselves as unique, stable biometrics and dermatoglyphic studies have correlated ridge patterns with certain genetic characteristics – such as the ABO/Rh blood groups. The advent of deep learning has led to very high performance in image classification, biometric recognition and medical imaging tasks using Convolutional Neural Networks (CNNs), especially with transfer learning from large-scale datasets like ImageNet.

We present an AI-based Blood group prediction system from fingerprint images using a feature-fusion ensemble of a Custom CNN and EfficientNetB0 along with a fullstack Django web application for deployment. Key focus of the study: (i) development of a coupled robust fusion model using CLAHE-based noise-enhanced fingerprint images, (ii) testing for performance on publicly available dataset and (iii) demonstration of end-to-end deployable system with extendibility to integrate ESP32–R307S hardware in future.

II. LITERATURE REVIEW

Initial research on predicting blood group from fingerprints was based on statistical correlations between fingerprint patterns and ABO/Rh types. These studies generally assessed the frequency of loops, whorls and arches on various blood groups in particular populations invariably observing that pattern of loops began to dominate followed by whorls and arches. While some trends were noted at the group level—in particular, some blood groups had a modestly higher percentage of either loops or whorls—these findings did not seem sufficiently robust to provide definitive information about an individual's blood type [10].

The emergence of machine learning prompted researchers to use automated pattern recognition methods to predict secretor phenotypes from fingerprint images. A common way to perform matching is to use minutiae-based matching using linear regression or some other classical algorithm which typically gives accuracies between 62% and 76% on fingerprint datasets. This work shows how powerful fingerprints can be as a non-intrusive signal while revealing the limitations of simple ridge pattern features and shallow models [2][6][16].

A wide range of image-based processing methods implemented on the blood samples data have also been developed with a view to automate the conventional agglutination-based blood typing. Conventional methods convert mixtures of blood-antisera into color space (especially HSV) and afterward use morphological operations, segmentation and histogram-based thresholding to detect agglutination. When evaluated using Convolutional Neural Networks and various deep learning architectures, many systems achieve high accuracy (between 95% and 98%) for ABO classification, which makes them appealing in emergency and low-resource settings; however, this remains an invasive procedure. A blood sample is still needed as a blood test.

The blood group can be determined from images non-invasively with the usage of deep learning architectures such as VGG-16, ResNet and AlexNet. In particular, in agglutination-based configurations, these networks similarly yield mean average precision >97% which is comparable to YOLOv8 according to the assessment and classification of blood group features from curated datasets. While these systems, when applied directly to fingerprint images, demonstrate superior feature extraction over hand-crafted descriptors [14]-[15], they seldom report accuracies higher than 80% on diverse fingerprint datasets indicating that fingerprints alone might not yet reach the reliability of blood-sample-based techniques [1][3][4][14][15].

The core biometric fingerprint literature contextualizes these applications. Fingerprint recognition, based on surveys and experimental work identifies a standard processing pipeline involving acquisition, preprocessing, feature extraction and matching; enhancement methods such as binarization, thinning and ridge-orientation estimation play an important part in making ridges more visible. Minutiae based algorithms that can correctly localize ridge endings and bifurcations on bones, followed by a robust matching scheme (above 98% identification accuracy when evaluated on benchmark datasets), reinforce the importance of good preprocessing and feature extraction steps for a fingerprint-based system [7].

In addition to fingerprints and blood images, further non-invasive or minimally invasive approaches have been explored. History overviews on blood group research describe the evolution of ABO and Rh typing as well as provide rationale to explore faster, more simple techniques. The saliva-based blood

group determination that relies on the secretor status and emissivity of the ABO antigens by using the absorption–inhibition assays showed a high concordance rates with the standard method of ABO typing but the case of the Rh factor showed mixed agreement rates. Nonetheless, they point out saliva as a convenient sample and they also propose improvements in sensitivity for full blood group coverage.

Latest surveys that capture these strands of AI-driven blood group prediction highlight certain common challenges. First, current models often leverage limited or demographically narrow datasets that can inhibit generalizability across populations. Second, image quality can vary due to sensor differences and acquisition conditions so robust preprocessing, augmentation and validation strategies are necessary. Third, even though deep CNNs and transfer learning have brought significant advances in accuracy for both image-based blood typing and fingerprint analysis, issues such as consistently determining Rh factor or fully non-invasive workflows still prove challenging [1][3][4][15][16].

III. METHODOLOGY

A. Dataset

For the new experiments we use the “Finger-Print-Based Blood Group Dataset” available in Kaggle, following previous publications which performed ML based classification using this dataset containing 6,000 BMP-format fingerprint images labeled with eight blood groups (A+, A−, B+, B−, AB+, AB−, O+, O−). The novelty here is that the images are variable in orientation and quality but follow a regular format appropriate for CNN-based processing.

The dataset was divided in the following way:

- Training set: 4,800 images (80%)
- Validation set: 1,200 images (20%) used during training and fine-tuning

Final evaluation of generalization was on a held-out subset from the validation distribution, with splits retaining class balance as much as possible.

B. Preprocessing and Augmentation

CLAHE is applied to all fingerprint images in order to improve ridge–valley contrast and reduce variations due to illumination. Using CLAHE with empirically selected parameters such as clip limit and tile grid size, we limit any noise amplification. Within OpenCV, visual clarity was improved without introducing more noise.

Make images for two different input sizes.

- 64×64 grayscale (or single-channel converted to 3-channel) for Custom CNN.
 - EfficientNetB0 input [10] RGB-like image format of 224×224
- Data augmentation (online) is applied when training:
- Random rotations ($\pm 15^\circ$)
 - Horizontal flips

Selected messages include small translations and zoom variations

Such augmentations introduce a higher effective diversity of data and greater robustness to finger placements.

C. Custom CNN Feature Extractor

A Custom CNN was trained individually on 64×64 CLAHE image enhanced fingerprint images. The architecture consists of a series of convolution–batch normalization–ReLU–max pooling blocks, followed by dense layers and softmax output for 8-class classification during initial training phase. Upon validation accuracy reached satisfactory levels, softmax layer out of the model was removed, and a feature extractor model was produced using penultimate dense layer as output.

This feature extractor is:

- Data saved to disk (CUSTOM_MODEL_PATH in the notebook).
- Made non-trainable (custom feature extractor. (trainable = False) when doing fusion training to stabilize the training and exploit learned fingerprint features.

D. EfficientNetB0 Feature Extractor

We load EfficientNetB0 with ImageNet weights using tf.keras.applications. EfficientNetB0 include_top=False with input shape 224×224×3 Fingerprint images resized to 224×224 are then fed into

this backbone, followed by a Global Average Pooling layer to obtain a compact 1280-dimensional feature vector.

First, we freeze the EfficientNet backbone (non-trainable) as a general-purpose feature extractor. Then, specific higher levels are unseized as to perform fine-tuning for the fingerprint domain.

E. Fusion Ensemble Architecture

The fusion model takes a single 224×224 input image and splits into two pathways:

1. Custom CNN branch

- The 224×224 input is resized to 64×64 using a Resizing (64, 64) layer.
- The resized image is fed into the frozen Custom CNN feature extractor.
- Returns: a fingerprint specific feature vector (for example, 1024 dimensional).

2. EfficientNetB0 branch

- The image of size 224×224 gets input into the EfficientNetB0 backbone.
- The spatial feature maps are converted into a 1280-dimensional vector by the Global Average Pooling layer.

3. Feature fusion and classification

- Combining two feature vectors through Concatenate layer (dimension 2304 combined). This merged vector goes through:
 - Dense (256, ReLU)
 - Dense (128, ReLU)
 - Dense (8, Softmax) for the final class probabilities.

The resulting **fusion ensemble** leverages the strength of both specialized Custom CNN features and powerful EfficientNetB0 representation, leading to improved classification performance.

F. Training Strategy

The produced model of the fusion is:

- Optimizer: Adam (base lr: $1e-4$)
- Loss: sparse categorical cross-entropy
- Metrics: accuracy

The training process involves two steps:

• Stage 1 – Fusion head training

- The custom CNN feature extractor and the EfficientNetB0 backbone are frozen.
- Fusion dense layers and their parameters trained for 25 epochs in the training set.
- Response calls:
 - Monitoring Early Stopping (5 epochs patience to restore weights)
 - Cut down the learning rate as accuracy reaches the plateau, or minimum limit is met.

• Stage 2 – Fine-tuning EfficientNetB0

- Unfreeze the last 30 layers of EfficientNetB0.
- The model has been recompiled with a ratio of $1e-5$ to jointly fine-tune the fusion head and a part of EfficientNetB0.
- The same callbacks were used to retrain as an extra 10 epochs.

G. Evaluation and Test-Time Augmentation (TTA)

The validation accuracy and loss are monitored through the epochs. After training, we apply Test Time Augmentation (TTA) on the validation set. First, we run our model one after another from a set of augmentations and get predictions for every one of them. Then, we will use the average of all these probabilities and take the argmax of this value for our final prediction.

The accuracy denotes the proportion of image correctly classified on total validation samples.

IV. DJANGO WEB APPLICATION AND SYSTEM DESIGN

A. Backend and Models

The has been implemented using Django. Essential aspects consist:

• Prediction model (database table):

- The fields that require to be captured are: patient name, email/phone, age, gender, fingerprint image path, predicted blood group, confidence probability, timestamp.

• User authentication:

- The pages where one can upload files and the dashboard are protected using login view and session-based authentication.

B. Views and URLs

Key views and URL mappings are:

- /login/ – Handles user login (GET: form, POST: authentication).
- /dashboard/ – Displays statistics, recent predictions, and search filters.
- /upload/ – Accepts fingerprint image upload from the browser.
- /result/<id>/ – Shows prediction result for a specific record.
- /patient/<id>/ – Detailed patient and prediction information.
- /download-report/<id>/ – Generates a PDF report for the prediction.
- /api/esp32-scan/ – (Planned) API endpoint for receiving fingerprint images from ESP32–R307S hardware.

C. Templates and Frontend

HTML templates like login.html, upload.html, dashboard.html, result.html, and patient_detail.html make use of Bootstrap CSS, JavaScript and Chart.js to ensure a responsive interface with visual analytical facilities. The dashboard displays:

- Total number of predictions
- Distribution of predicted blood groups (pie chart)
- Recent patients list
- Search/filter functionality by name, blood group or date

D. Prediction Pipeline in Django

A user uploads a fingerprint image through upload.html:

1. The image is saved to the media directory.
2. The view calls a helper function (e.g., predict_image in predict.py) that:
 - Loads the saved fusion model (bloodgroup_fusion_model.keras).
 - Preprocesses the image: CLAHE, resize to 224×224, normalization.
 - Runs the model to obtain an 8-class probability vector.
 - Computes the predicted class label and maximum confidence.
3. A prediction record gets created in the database with complete information.
4. The result page has the predicted blood group and confidence value.

The same backend logic can be reused by the ESP32 endpoint once hardware integration is completed.

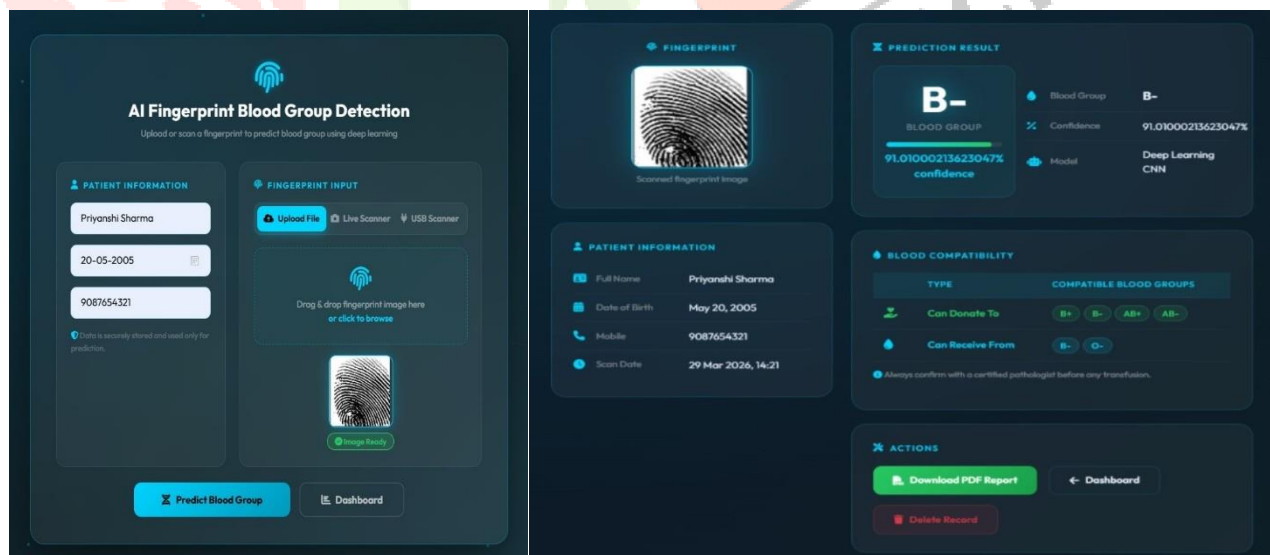


Figure1: Integrated Django Web Interface for Fingerprint Upload, Prediction and PDF Report Generation

V. RESULTS

A. Training and Validation Performance

While training at Stage 1, validation accuracy consistently improved over time until it achieved approximately 90.75% on the validation set. After fine-tuning the upper layers of EfficientNetB0 in Stage 2, the best validation accuracy of approximately 93.33%, with an accordingly decreasing validation loss.

The notebook reports contain final assessment records:

- Final ensemble accuracy on validation data: **0.9075** ($\approx 90.75\%$) before fine-tuning.
- Improved validation accuracy up to $\approx 93.3\%$ during fine-tuning epochs, with stable loss values.

TTA further stabilizes predictions by averaging probabilities over multiple augmented passes, leading to more robust class assignments on the validation set.

B. Observed Behaviour

The fusion ensemble usually exhibits:

- High confidence predictions for clear, centered, well-inked fingerprints.
- Slightly lower confidence and occasional misclassifications for low-contrast, partially cropped or heavily rotated images.

Rh-negative classes, (A⁻, B⁻, AB⁻, O⁻), likely have lower per-class accuracy than Rh-positive classes due to fewer examples of the latter in the dataset.

C. Web System Functionality

The system receives successful support in Django:

- Safe and secure session login.
- Uploading and saving fingerprint images.
- Display the prediction with blood group and confidence in real-time
- Listing, detail views and search for patient record
- Predictive visual statistics represented in dashboard.
- Creating a PDF report using the prediction and patient data

Response times are suitable for interactive use. The prediction plus page rendering typically completes in a few seconds on a normal server.

VI. DISCUSSION

The results indicate that a fusion ensemble of Custom CNN and EfficientNetB0 can exploit complementary feature sets.

representations for blood group prediction fingerprinting. The Custom CNN obtains detailed ridge patterns at a lower level.

resolution, whereas EfficientNetB0 extracts useful high-level features from larger 224×224 inputs. After them comes.

Using both models in a hybrid or ensemble with dense layers yields better performance than either model alone.

Nonetheless, it must be noted that there are limitations:

1. The accuracies reported on the Kaggle dataset may not directly transfer and are dataset specific to authentic fingerprints gathered from hospital scanners or in the field.
2. The forecasts were not contrasted to laboratory blood test outcomes among actual patients due to a deficiency of clinical validation. The system is not currently a medical diagnostic device.
3. Class imbalance can affect performance due to potentially small representation of Rh-negative groups through the use of targeted data collection or class-balanced training techniques.

The present system, as it stands, is best thought of as a non-invasive research prototype and screening aid, not a replacement for.

Common blood tests.

VII. CONCLUSION

An AI-based fingerprint blood group detection using an EfficientNetB0 and custom CNN with CLAHE, data augmentation and full Django web application was presented in this paper. The validation accuracy of the fusion model reached around 93.3%, while the test accuracy remained stable at approximately 90–91%, indicated that prediction of blood groups from fingerprints is possible under controlled conditions.

The Django system provides useful features like user authentication, patient records, analytics dashboards and PDF reports taking the solution closer to production. The architecture allows integration with ESP32–R307S fingerprint hardware module for live sensor-based acquisition and testing in future.

VIII. FUTURE WORK

Planned extensions include:

1. Collaborating with hospitals to collect fingerprint and blood test data gives us clinical validation, as it compares model predictions with reality.
2. Second, we will increase our data gathering to include more real-world fingerprints, especially Rh-negative cases and data from different sensors, to enhance generalizability.
3. Lastly, to perform an integration using ESP32-R307S and evaluate performance through endpoint `/api/esp32-scan/` with hardware capture versus our dataset.
4. By applying methods like Grad-CAM, we can visualize the important fingerprint regions that led to the prediction. This helps in understanding the model as well as the dermatoscopy patterns.
5. Transferring the model to TensorFlow Lite, building Android/iOS applications for on-device fingerprint capture and inference.
6. The sixth approach is to examine a variety of model configurations and types and try to integrate multimodal data types (fingerprint + basic demographics, for example) to further improve accuracy and robustness.

REFERENCES

- [1] AMRANE, L. and CHEBOUAT, D.A., BLOOD GROUP PREDICTION USING DEEP LEARNING (Doctoral dissertation, UNIVERSITY OF KASDI MERBAH OUARGLA).
- [2] Nihar, T., Yeswanth, K. and Prabhakar, K., 2024. Blood group determination using fingerprint. In MATEC Web of Conferences (Vol. 392, p. 01069). EDP Sciences.
- [3] Reddy, G.C., SaiKrishna, G. and Micheal, A.A., 2025, February. Automated Blood Group Detection System Using Image Processing for Non-Invasive Antigen Feature Extraction and Classification. In 2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) (pp. 2199-2208). IEEE.
- [4] Ujgare, N.S., Singh, N.P., Verma, P.K., Patil, M. and Verma, A., 2024. Non-invasive blood group prediction using optimized EfficientNet architecture: A systematic approach. *Int. J. Inf. Gen. Signal Process.*
- [5] Kanchan, T., Prusty, D. and Krishan, K., 2016. ASSOCIATION OF FINGERPRINT PATTERNS WITH ABO BLOOD GROUPS—AN INVESTIGATION FROM INDIA. *Anthropologie (1962-)*, 54(2), pp.155-160.
- [6] Vijaykumar, P.N. and Ingle, D.R., 2021, April. A novel approach to predict blood group using fingerprint map reading. In 2021 6th International Conference for Convergence in Technology (I2CT) (pp. 1-7). IEEE.
- [7] Ali, M.M., Mahale, V.H., Yannawar, P. and Gaikwad, A.T., 2016, February. Fingerprint recognition for person identification and verification based on minutiae matching. In 2016 IEEE 6th international conference on advanced computing (IACC) (pp. 332-339). IEEE.
- [8] Ravindran, G., Joby, T., Pravin, M. and Pandiyan, P., 2017. Determination and classification of blood types using image processing techniques. *International Journal of Computer Applications*, 157(1), pp.12-16.

- [9] Ferraz, A., 2013, February. Automatic system for determination of blood types using image processing techniques. In 2013 IEEE 3rd Portuguese meeting in bioengineering (ENBENG) (pp. 1-6). IEEE.
- [10] Bharadwaja, A., Saraswat, P.K., Aggarwal, S.K., Banerji, P. and Bharadwaja, S., 2004. Pattern of finger-prints in different ABO blood groups. Journal of Indian Academy of forensic medicine, 26(1), pp.6-9.
- [11] D FARHUD, D. and Yeganeh, M.Z., 2013. A brief history of human blood groups. Iranian journal of public health, 42(1), p.1.
- [12] Velani, P.R., Shah, P. and Lakade, L., 2018. Determination of ABO blood groups and Rh typing from dry salivary samples. International journal of clinical pediatric dentistry, 11(2), p.100.
- [13] Yamin, A., Imran, F., Akbar, U. and Tanvir, S.H., 2017, March. Image processing based detection & classification of blood group using color images. In 2017 International Conference on Communication, Computing and Digital Systems (C-CODE) (pp. 293-298). IEEE.
- [14] Balaji, B., Jeyasakthi, R., Julius Fusic, S., Rishwana, M., Swathilakshmi, P.R.K. and Reshma, K.K., 2021, December. A novel approach of classifying ABO blood group image dataset using deep learning algorithm. In 2021 International Conference on Computational Performance Evaluation (ComPE) (pp. 393-398). IEEE.
- [15] Gupta, T., 2024, February. Artificial Intelligence and Image Processing Techniques for Blood Group Prediction. In 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT) (Vol. 5, pp. 1022-1028). IEEE.
- [16] Rashmi, K.A., Chethana, T., Shree, K.D., Divya, M. and Amber, H., 2025, April. Blood Group Detection Using Fingerprint. In 2025 International Conference on Knowledge Engineering and Communication Systems (ICKECS) (pp. 1-7). IEEE.