applied sciences

MDPI

Article

Blood Typing at the Edge: A Hybrid Deep Learning Pipeline for Point-of-Care Blood Type Classification

Bruno Silva [1,*], Enmanuel Abilheira [1], Ljiljana Dukanovic [2], Afonso Pinheiro [2] and Vítor Carvalho [1,*]

[1] 2Ai, School of Technology, Polytechnic University of Cávado and Ave, 4750-810 Barcelos, Portugal
[2] CRIAM, 1600-514 Lisbon, Portugal; afonso.pinheiro@criamknowledge.com (A.P.)
* Correspondence: a20811@alunos.ipca.pt (B.S.); vcarvalho@ipca.pt (V.C.)

Abstract

Blood typing remains a manual, subjective procedure when not reliant on centralized laboratory infrastructure. This study presents an automated blood typing system for point-of-care deployment, developed in collaboration with CRIAM, whose portable device captures reaction images for in vitro diagnostics. The system integrates computer vision and artificial intelligence to classify these reactions automatically. Fourteen classification pipelines were trained and evaluated with a 3090-image dataset, encompassing fine-tuned convolutional neural networks, raw pixel-based classifiers, and hybrid architectures pairing pretrained embeddings from DINOv2 and EfficientNet-B4 with lightweight classifiers. Embedding-based approaches consistently outperformed alternatives in accuracy and cross-fold stability. The best pipeline, in terms of performance and suitability for low-power devices, combined DINOv2-small embeddings with logistic regression, achieving $99.87 \pm 0.12\%$ mean accuracy. After 8-bit integers (INT8) quantization and retraining with data augmentation, accuracy improved to $99.97 \pm 0.03\%$, surpassing the uncompressed baseline. All misclassifications were traced to borderline weak-positive Rh/D reactions, confirming errors are localized and explainable. Held-out validation on 856 images yielded 99.53% accuracy, with the single error attributed to a lighting artifact. While deployment on a legacy 32-bit CPU prototype processes four images in approximately 4.7 min, hardware benchmarking confirmed feasibility, from a Raspberry Pi Zero 2W to high-end mobile processors. These results establish quantized embedding-driven architectures as a solution for automated blood typing in point-of-care and resource-limited settings.

Keywords: blood typing; artificial intelligence; hemagglutination; deep learning; computer vision

Check for updates

Received: 1 May 2026
Revised: 5 June 2026
Accepted: 11 June 2026
Published: 16 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

This work aims to develop an intelligent system to analyze hemagglutination reactions (Figure 1), addressing the limitations of traditional blood typing methods, such as the need for specialized staff, laboratory infrastructure, and the risk of human error, since they are performed manually and analyzed with the naked eye. This was developed in collaboration with CRIAM [1], a company in the field of point-of-care (POC) in vitro diagnostics and focusing on the automation of agglutination assays using computer vision on a portable device.

This study involved the evaluation of various machine learning architectures, from convolutional neural networks (CNNs) and Vision Transformers (ViTs) to more conven-

tional machine learning approaches, and the optimization of these models for deployment on resource-constrained, battery-operated POC devices.

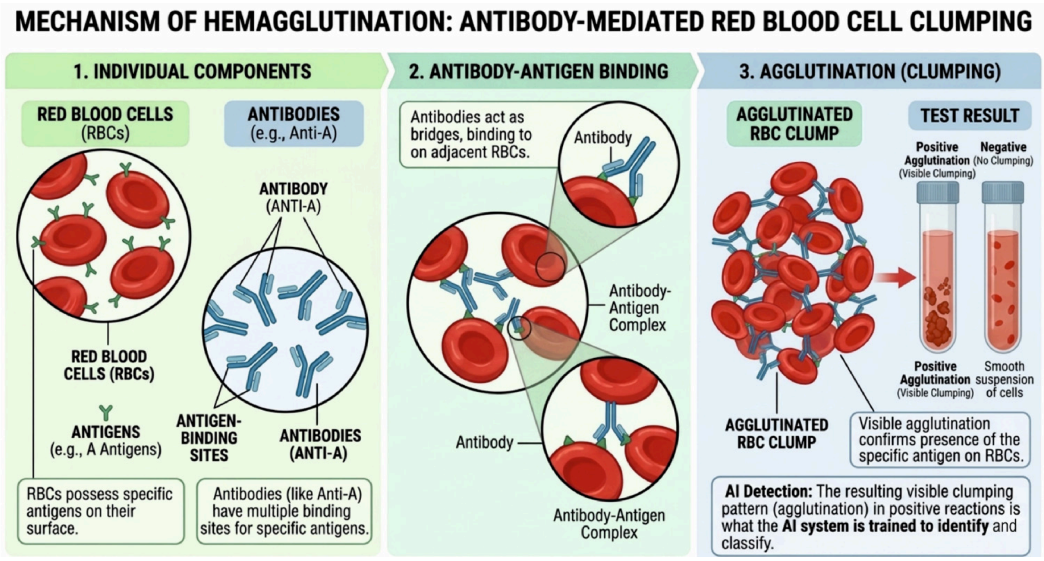


Figure 1. Hemagglutination mechanism between antigens in red blood cells and antibodies in serum or reagents.

The healthcare diagnostics sector faces a constant need for rapid and reliable results, especially in time-sensitive and POC settings. Integrating artificial intelligence (AI) offers a promising approach to overcome existing barriers, making tests more reliable and accessible in settings with limited resources.

Beyond operational efficiency, the clinical consequences of blood typing errors must motivate the design of any automated system. Incorrect blood group assignment can result in acute or delayed hemolytic transfusion reactions, which carry significant morbidity and mortality risk. These safety considerations justify the adoption of confidence-aware inference thresholds and human review pathways for borderline results.

1.1. Goals

The goal of this study was to develop an AI system that identifies hemagglutination reactions to accurately determine the blood type of a sample. By using AI, the aim is to improve the efficiency and reliability of blood typing in POC settings, be it laboratories or remote mobile settings, without the need for highly trained personnel. To achieve success in this work, several key objectives must be met:

- Explore current machine learning and image processing methods related to blood typing, reviewing the existing scientific literature on the topic.
- Develop a variety of AI models with diverse architectures and methods, and evaluate them in terms of accuracy, ensuring that it is equivalent to current automated blood typing systems.
- Optimize the best-performing model for deployment on resource-constrained devices, focusing on processing speed, model size, and efficiency, and test the model in a simulated and embedded environment.

1.2. Collaborative Framework

As previously referred to, this study was conducted in collaboration with CRIAM, a company specializing in POC diagnostics. CRIAM provided a labeled imagery dataset of

blood typing reactions captured on their device (Figure 2), which we were able to use to develop accurate models.



Figure 2. 2026, Criamknowledge S.A prototype device, adapted from [1].

1.3. Background

The recent literature demonstrates growing interest in applying computer vision and deep learning to automate hemagglutination-based blood typing [2–6].

Balaji et al. (2021) [2] proposed a lightweight CNN for ABO typing from slide images, achieving 96.7% validation accuracy while requiring fewer parameters and less training time than established architectures such as AlexNet and LeNet-5. Titus et al. (2023) [3] introduced SLHBGA, a schematic learning pipeline based on engineered features—white pixel density and object count—that consistently outperformed a CNN baseline in accuracy, sensitivity, and inference speed over ten days of hospital deployment. Reddy et al. (2025) [4] applied YOLOv8 to non-invasive ABO/Rh detection, reaching an F1-score of 96.5% and a mean average precision of 97.4%, with inference times of approximately 15 ms per image.

Beyond slide-based software systems, two hardware-integrated platforms were identified. Larpant et al. (2022) [5] coupled a paper-based analytical device with a YOLOv4-tiny and DenseNet-201 pipeline for simultaneous Rh phenotyping (D, C, E, c, e), achieving 99.19% agreement with the gold-standard tube method across 4692 samples, including correct resolution of all 48 ambiguous cases. Ding et al. (2023) [6] developed a centrifugal microfluidic disk paired with a Support Vector Machine (SVM) classifier trained on HOG features, obtaining 92.7% accuracy across 750 agglutination patterns, outperforming both KNN and a ResNet-based CNN on the same dataset.

Collectively, these works show that AI-assisted blood typing can approach the diagnostic performance of conventional laboratory methods while offering gains in speed, reproducibility, and suitability for point-of-care settings. Remaining challenges include the handling of weak or heterogeneous agglutination patterns and the limited size of available labeled datasets.

To broaden the field background, blood group determination should be positioned within the longer evolution of immunohematology methods, from conventional tube and slide agglutination to gel-column, solid-phase, paper-based, and microfluidic formats. Gel-column testing represented an important step toward more standardized antigen–antibody reaction reading, while recent reviews emphasize that modern blood group testing increasingly combines serology, automation, molecular methods, and miniaturized point-of-care platforms [7–11].

Several point-of-care strategies have attempted to reduce the dependence on centralized laboratory infrastructure. Disposable microfluidic chips, paper-based assays, forward/reverse ABO devices, passive microfluidic biochips, smartphone-readable paper devices, digital microfluidic assays, and three-dimensional origami paper devices have all

been proposed to simplify sample handling and accelerate result interpretation [12–19]. These studies demonstrate that the analytical format and the computational reader are increasingly co-designed; however, many approaches still depend on controlled illumination, careful handling, or visual interpretation by trained operators.

From the computational perspective, the literature has progressed from handcrafted image descriptors and template-matching methods to CNNs, object detectors, hybrid microfluidic image analysis, and neural second-opinion systems. Earlier methods such as ORB, SIFT, SURF, or HOG + SVM demonstrated the feasibility of automated agglutination interpretation but were more sensitive to brightness, texture, and acquisition conditions. More recent deep learning and embedding-based approaches seek to improve robustness, particularly for weak, heterogeneous, or ambiguous agglutination patterns [2–6,12,13].

Table 1 provides a comparative synthesis of representative image-based and AI-assisted blood typing studies, highlighting the task addressed, acquisition platform, computational method, scale of evaluation, reported performance, and main limitations. This comparative analysis clarifies that the present contribution has not only high classification accuracy but also the combination of foundation-model embeddings, a lightweight classifier, INT8 deployment, held-out validation, and explicit edge-device benchmarking.

Table 1. Comparative analysis of representative image-based and AI-assisted blood typing studies.

Study	Platform/Task	AI or Image-Analysis Method	Evidence and Reported Performance	Main Limitation/Relevance to This Work
Balaji et al. [2]	Slide-image ABO classification	Lightweight CNN	Reported validation accuracy of 96.7%	Proof-of-concept CNN approach; dataset scale and external validation remain limited.
Titus et al. [3]	Hospital workflow for blood-group identification	Engineered features including white-pixel density and object count	Reported improved accuracy, sensitivity, and inference speed over a CNN baseline during hospital deployment	Fast and interpretable, but dependent on handcrafted features and acquisition assumptions.
Reddy et al. [4]	Non-invasive ABO/Rh feature extraction and classification	YOLOv8-based detection/classification	Reported F1-score of 96.5%, mAP of 97.4%, and approximately 15 ms inference per image	Promising speed, but the task differs from direct hemagglutination interpretation.
Odeh et al. [20]	Automatic blood type determination from agglutination images	Image matching using ORB, SIFT, SURF, and related descriptors	Evaluated on 500 samples /1500 patches; optimized ORB achieved high accuracy with short processing time	Demonstrates value of computer vision, but handcrafted features remain sensitive to brightness and texture variability.

Table 1. *Cont.*

Study	Platform/Task	AI or Image-Analysis Method	Evidence and Reported Performance	Main Limitation/Relevance to This Work
Larpant et al. [5]	Paper-based simultaneous Rh phenotyping	YOLOv4-tiny and DenseNet-201 pipeline	4692 samples; 99.19% agreement with tube method, including ambiguous cases	Strong validation scale and clinical relevance; focused on Rh phenotyping rather than full ABO/Rh POC cartridge deployment.
Ding et al. [6]	Centrifugal microfluidic antigen typing	HOG features with SVM classifier	750 agglutination patterns; 92.7% accuracy; outperformed KNN and a ResNet-based CNN on the same dataset	Hardware-integrated and parallel, but lower accuracy than recent embedding-based approaches.
Korchagin et al. [21]	Second opinion for manual serological plate reading	Neural classifier/ensemble for well-image interpretation	3139 well images; reported mean quality of 98.4%	Useful decision-support paradigm, but targeted at serological plates rather than embedded POC cartridges.
Present study	CRIAM POC cartridge images for ABO/Rh classification	DINOv2 embeddings with logistic regression; ONNX INT8 deployment	3090-image development dataset; 99.97 +/− 0.03% cross-validation accuracy after augmentation; 99.53% final held-out test accuracy on 856 images	Single-device dataset; future work requires multi-device, multi-site, reagent-lot, and prospective validation.

The comparison shows that prior work has generally optimized only one dimension at a time: algorithmic classification, microfluidic integration, ambiguous-case handling, or decision support. In contrast, the proposed pipeline addresses several of these dimensions simultaneously by pairing robust self-supervised visual representations with a computationally lightweight classifier and a deployment-oriented quantization workflow. Nevertheless, the comparison also confirms that external clinical validation, multi-device data collection, and prospective performance evaluation remain essential before regulated clinical use.

1.4. Paper Organization

This paper is organized into five chapters. Section 2 presents the methodology developed for automated blood type classification, covering the dataset collection and preprocessing pipeline, the three families of architectures evaluated (fine-tuned CNNs, raw pixel-based classifiers, and hybrid embedding approaches), the training and validation strategies, the ONNX export and optimization workflow, the data augmentation pipeline, and the embedded deployment and cross-compilation procedures. Section 3 presents the

experimental results, including an overall performance comparison of all fourteen classification pipelines, a detailed analysis of the best-performing DINOv2 + logistic regression model, the quantization and optimization trade-offs across ONNX export formats, hardware benchmarking results across a range of embedded and mobile platforms, deployment tests on the CRIAM prototype device, and the final held-out validation on 856 reaction images drawn from 214 blood typing tests. Section 4 discusses the interpretation and practical significance of the findings, examining why embedding-based approaches outperform alternatives, the impact of lighting conditions and reaction sensitivity, the role of data augmentation, quantization trade-offs, deployment feasibility on embedded hardware, identified limitations, and recommendations for future improvement. Section 5 presents the conclusions and directions for future work.

2. Methodology

This chapter presents the methodology developed for automated blood type classification using artificial intelligence. It covers the dataset collection and preprocessing pipeline, the classification workflow, and the three types of architectures evaluated: fine-tuned CNNs, traditional machine learning with raw pixel features, and a hybrid approach combining deep learning embeddings with classical classifiers. Additionally, it details the training strategies, validation protocols, and model optimization techniques used for deployment.

2.1. Data Collection and Preprocessing

The dataset consisted of 3090 blood typing reaction images provided by CRIAM, acquired under controlled conditions by the camera inside their device (Figure 2). Blood samples are mixed and agitated for 2 min with the Anti-A (well 1), Anti-B (well 2), Anti-AB (well 3), and Anti-D (well 4) reagents on separate wells of a 3D-printed cartridge, with 773, 771, 773, and 773 images for each reagent respectively, following a slide test protocol, resulting in four reaction images per sample. Images were then labeled with the ground-truth results from the Neo Iris blood bank analyzer (Immucor Inc., Norcross, GA, USA.). This pre-labeling enabled a supervised learning approach, allowing the models to distinguish between agglutinated (positive) and non-agglutinated (negative) reactions.

To ensure robust model performance and mitigate overfitting, a common challenge in neural network training [22], data augmentation techniques were employed throughout the study in two distinct stages. In the first stage, applied across the training set during CNNs fine-tunes training, the augmentation pipeline included geometric transformations such as 90-degree rotations, horizontal and vertical flips, as well as photometric adjustments including brightness tuning (randomly scaling intensity between 0.7× and 1.3× of the original value), saturation adjustment to prevent the model from relying on specific reagent colorimetry, and Gaussian noise. In a second stage, specifically designed to recover the accuracy loss introduced by INT8 quantization, an augmentation pipeline was applied to generate three additional augmented versions of each original image, expanding the effective dataset size fourfold. Each augmented image was produced by applying a randomized sequence of four transformations drawn from the same pool described above.

2.2. Classification Workflow

The proposed system operates on a two-phase approach:

- Agglutination Detection: The core AI task, where the model analyzes the reaction images to detect the presence or absence of a hemagglutination reaction (Figure 3).
- Blood Type Classification: The system identifies which reagent (Anti-A, Anti-B, Anti-AB, or Anti-D) was used in each well via digit recognition or fixed positional logic. The results of the four separate reactions are then combined to determine the sample's

blood type, e.g., "A+" (A-positive) or "O−" (O-negative), as explicitly shown for every blood type in Figure 4.

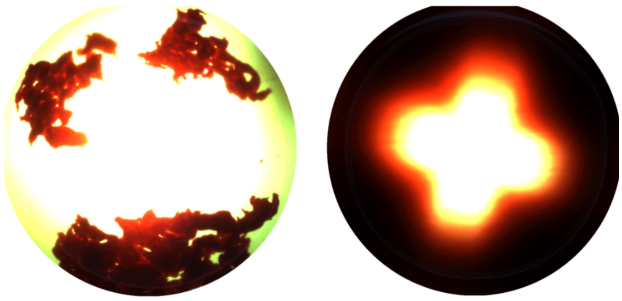


Figure 3. Positive hemagglutination reaction (**left**) and negative reaction (**right**) from CRIAM device.

Blood group	Anti-A	Anti-B	Anti-D	Control
O−	−	−	−	−
O+	−	−	+	−
A−	+	−	−	−
A+	+	−	+	−
B−	−	+	−	−
B+	−	+	+	−
AB−	+	+	−	−
AB+	+	+	+	−
Invalid	*	*	*	+

Figure 4. Reference interpretation scheme for ABO/Rh(D) blood typing reaction classification. The matrix defines the expected reaction pattern for each ABO/Rh(D) class using Anti-A, Anti-B, Anti-D, and control wells. Positive agglutination in a reagent well indicates detection of the corresponding antigen and supports the final blood group assignment. The control well is used to identify invalid reactions; if the control is positive, the image pattern should be classified as invalid rather than assigned to an ABO/Rh(D) class. Symbols: +, agglutination; −, no agglutination; *, invalid or non-interpretable pattern.

2.3. Model Architectures Evaluated

An evaluation of different AI architectures was conducted to find the optimal balance between accuracy and computational load:

- Fine-Tuned CNNs: Pretrained CNN models (Figure 5) were fine-tuned to specifically detect hemagglutination. This method applied transfer learning [23,24], allowing the model to use knowledge from a large dataset in a new domain. The models used were EfficientNet B4, EfficientNet V2, Inception V3, NFNet, and ResNet 50.
- Raw Pixels with Machine Learning: Algorithms such as logistic regression (LR) [25], SVM [26], and Extreme Gradient Boosting (XGBoost), which build upon decision tree

and random forest foundations [27], were tested as lightweight alternatives using pixel values as input features (Figure 6).

- Hybrid Approach: This method tested a pretrained ViT called DINOv2 [28] and the CNN EfficientNet B4 combined with traditional machine learning classifiers (Figure 7). In this architecture, the deep learning models extracted generalized image embeddings, which are then classified by a more lightweight machine learning model.

All models were validated using stratified K-fold cross-validation to ensure the results were consistent and reliable across different subsets of data [29].

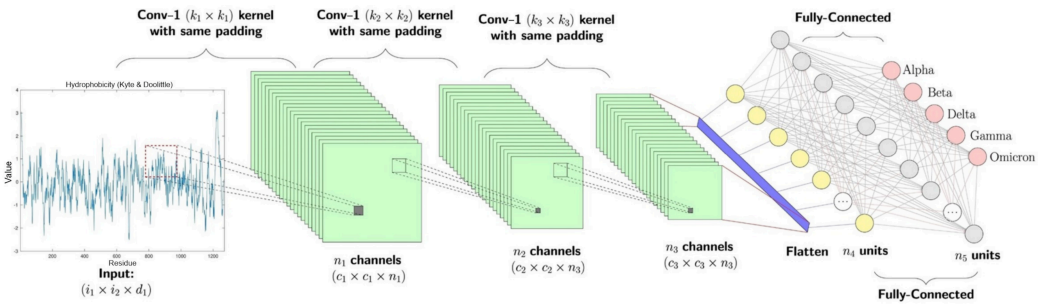


Figure 5. Convolutional neural network (CNN) architecture.

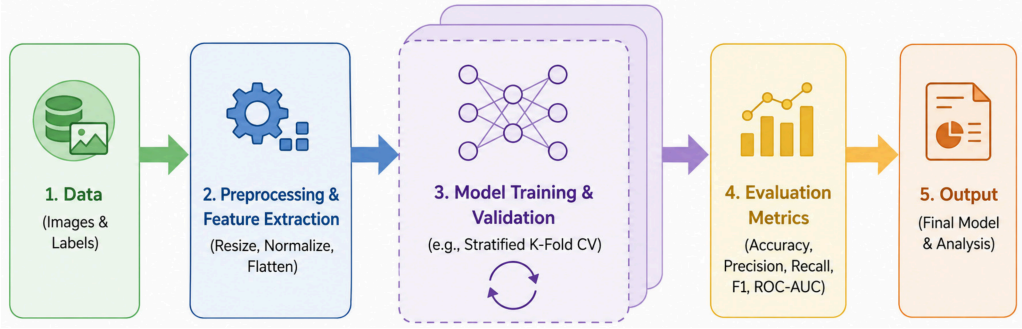


Figure 6. Raw pixel workflow, adapted from [30].

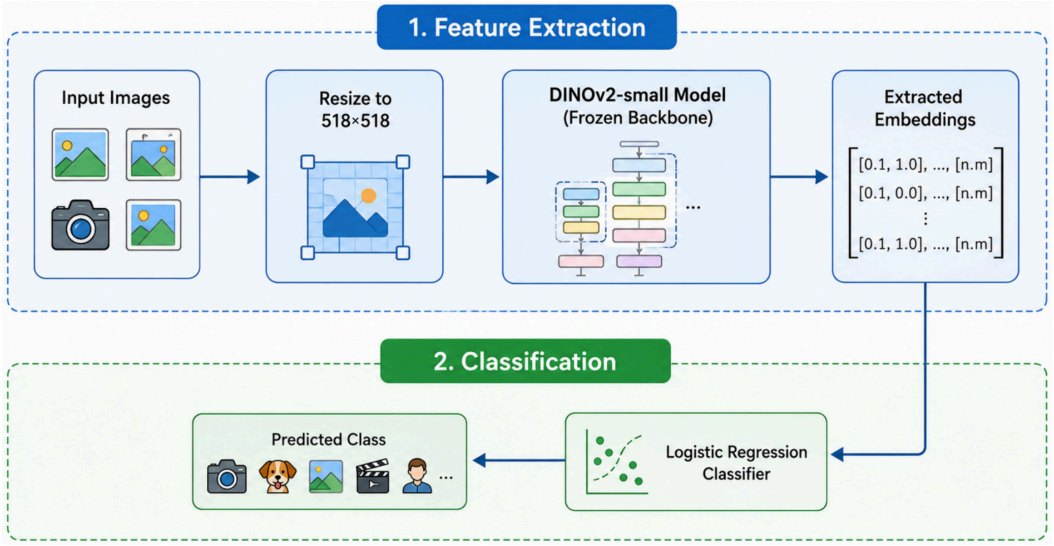


Figure 7. Hybrid workflow with DINOv2 and logistic regression, adapted from [30].

2.4. CNN Fine-Tune Approach

Instead of training a model from scratch, this approach utilizes transfer learning, using the pre-existing knowledge of the models trained on large imagery datasets.

- Feature Extractor: The backbone consists of the pretrained CNN layers, which are already trained for identifying shapes, textures, and objects in general.
- Custom Classification Head: The original 1000-class output layer was removed and replaced with a fully connected binary classification layer (Positive/Negative).

The training followed a "progressive unfreezing" strategy to maintain the integrity of the pretrained weights while gradually allowing the model to adapt to the imagery dataset.

- Initial Phase (Epochs 0–2): All backbone layers were frozen; only the new custom classification head was trained to align with the backbone's feature extraction.
- Partial Unfreezing (Epoch 3): The final four layers of the model were unfrozen, allowing the model to begin learning high-level visual features specific to hemagglutination.
- Full Fine-Tuning (Epoch 7): All model parameters were unfrozen, enabling the entire network to undergo subtle adjustments.

To handle the "noisy" or borderline cases common in medical imaging, the training loop was enhanced with specific optimizations.

- Online Hard Example Mining (OHEM): Rather than calculating loss for every image equally, the system implemented a 30% mining ratio. In each batch, the model identified the 30% of images where its predictions were least accurate and prioritized those for weight updates, forcing the network to "study" the hardest cases.
- Numerical Precision: Training utilized 16-bit Mixed Precision (FP16). This allowed for faster computation and lower memory usage while maintaining high numerical stability for critical gradients.
- Optimization Algorithm: The AdamW optimizer was used for the gradient update process.

To ensure the fine-tuning was generalizable and not just "memorizing" the training set, an evaluation framework was applied.

- Stratified 5-Fold Cross-Validation: The dataset was divided into five stratified folds, each preserving the original ratio of positive to negative reaction images. At each iteration, three folds were used for training, one for validation, and one for testing. Both the validation, test, and train folds were rotated independently across fold fine-tune, meaning no two fine-tunes during the 5-Fold Cross-Validation shared the same validation, test, or train. Over the full procedure, every sample appeared in the training, validation, and test sets at least once.
- Internal Validation Split: Within each fold, 20% of the training data was reserved for real-time validation to trigger "early stopping" if the model's accuracy on unseen data began to plateau.
- Hold-out Testing: A dedicated test set for each fold was used exclusively for final metrics, ensuring reported accuracy reflects performance on entirely unseen images.

2.5. Raw Pixel Approach

This method could also be referred to as a "pixel-as-feature" approach, where the visual information is represented as a high-dimensional vector (Figure 6).

- Standardization and Resizing: All images are resized to a fixed resolution of 128×128 pixels to ensure a uniform feature vector size across the dataset.
- Dimensionality: For RGB images, this results in a feature vector of 49,152 dimensions ($128 \times 128 \times 3$ color channels).
- Normalization: Pixel intensities are normalized from their raw 8-bit range (0–255) to a floating-point range from 0 to 1. This prevents features with higher variance from dominating the model's objective function.

To ensure the model's reliability, the script implements a robust validation framework combined with feature importance analysis.

- Stratified 5-Fold Cross-Validation: Performance is averaged across five different data splits, ensuring that the results (Accuracy, Precision, Recall, F1-score, and ROC-AUC) are stable and not dependent on a specific training subset.
- Stability Assessment: The standard deviation of accuracy across folds is monitored to categorize the model's performance as "Stable" or "Unstable," providing a confidence metric for the results.

2.6. Hybrid Approach

Instead of training a CNN model, this method uses a pretrained CNN, EfficientNet B4, or the Vision Transformer, DINOv2, to create embeddings of the images and then uses those embeddings in a machine learning classification model (Figure 7).

The workflow illustrated in Figure 7 operates in two main stages, each composed of sequential steps.

1. Input: Raw reaction images are fed into the pipeline as the starting point of the process.
2. Preprocessing: Each image is resized to a fixed resolution of 518 × 518 pixels for DINOv2, or 380 × 380 pixels for EfficientNet B4, to ensure compatibility with the model's expected input format. Standard ImageNet normalization (mean and standard deviation) is also applied so that the input data aligns with the model's pre-training distribution.
3. Embedding Generation: The resized images are passed through the DINOv2 or EfficientNet B4, used as a frozen backbone. The classification head is removed, transforming the network into a pure feature extractor. Each image is converted into a high-dimensional numerical vector (embedding) that captures visual information.
4. Extracted Embeddings: The output is a matrix of embeddings, one per image, representing the dataset in a fixed-size numerical space.
5. Classifier: The extracted embeddings are passed to a classification model used as the final decision engine. The classification models evaluated include logistic regression (C = 1.0, L2 penalty, lbfgs solver, max_iter = 100, and random_state = 42), SVM (random_state = 42, kernel = 'rbf', C = 1.0), and XGBoost (n_estimators = 500, learning_rate = 0.1, max_depth = 6, random_state = 42, eval_metric = 'logloss').
6. Predicted Class: The classifier outputs a predicted class label for each input image (positive or negative).

The hybrid pipeline includes a validation framework to measure both performance and reliability.

- Stratified 5-Fold Cross-Validation: The data is split into five folds, ensuring the ratio of positive to negative reactions remains consistent. Performance metrics (Accuracy, Precision, Recall, and F1-score) are reported as an average across these folds to ensure the results are representative of the entire dataset.
- Stability Assessment: The system calculates the standard deviation of accuracy across all folds, providing a confidence metric for the model's consistency.

The workflow includes an "overfitting detector" that compares the training accuracy against the validation accuracy. A difference of less than 0.05 is flagged as "Good Generalization," indicating that the embeddings and the classification model are capturing universal features of the hemagglutination reactions rather than just memorizing the training images.

The system also tracks specific misclassified paths, allowing for a post-review of the images that the model classified incorrectly.

2.7. ONNX Export and Optimization

The first step in the optimization workflow is converting the model in PyTorch 2.7 to the Open Neural Network Exchange (ONNX) format [31] (Figure 8).

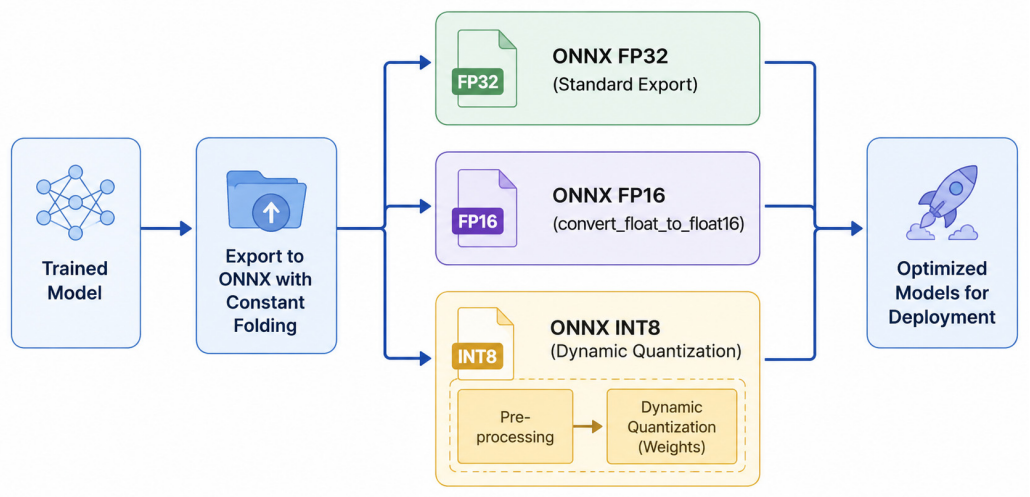


Figure 8. Model export and optimization workflow from Torch to ONNX, adapted from [7].

- Standardization: The DINOv2 backbone is exported to a standardized computational graph (Opset version 17), which allows the model to be run across different hardware backends (CPU, GPU) using the ONNX Runtime 1.23.2.
- Constant Folding: During the export process, "do_constant_folding" is enabled. This optimization pre-calculates constant expressions in the model graph (such as fixed weights or normalization constants), replacing them with their result to reduce the number of operations required during inference.

The following optimization step reduces the numerical precision of the exported ONNX model from the standard 32-bit floating-point (FP32) format to 16-bit (FP16), using the ONNX Runtime quantization tooling.

- Precision Reduction: The standard FP32 weights and activations are converted to FP16.
- Benefits: This conversion can improve processing speed and reduce the model's memory footprint by 50% while maintaining high numerical precision.

To support an even lighter deployment, a dynamic INT8 quantization is implemented:

- Weight Quantization: The model weights are converted from FP32 to 8-bit integers (INT8). This is done using a per-channel approach, which provides higher accuracy for ViT architectures by allowing different scaling factors for each weight channel.
- Memory Efficiency: This optimization can result in an approximate 75% reduction in memory usage, making the large DINOv2 model more adequate for resource-constrained computing.

The script performs a dummy inference pass to ensure that the embeddings generated by the optimized ONNX models match the original PyTorch model outputs, guaranteeing that the optimization process has not corrupted the feature extraction logic.

2.8. Image Data Augmentation

To recover accuracy lost during the quantization process, we implemented a stochastic data augmentation pipeline specifically targeting the quantized hybrid pipeline. For each original image in the 3090-image dataset, three new augmented versions were generated, expanding the training set to 12,360 images in total. Each version was produced by applying a randomized sequence of four transformations drawn from a predefined pool of

geometric and photometric operations, as exemplified in Figure 9. These augmented images were then passed through the quantized ONNX DINOv2-small backbone to extract their corresponding embeddings, and the resulting expanded embedding set was used to retrain the logistic regression classifier. This approach ensured that the classifier was exposed to a broader distribution of visual representations, compensating for the representational precision lost during INT8 compression; we also compared all the ONNX exports with this augmented dataset.

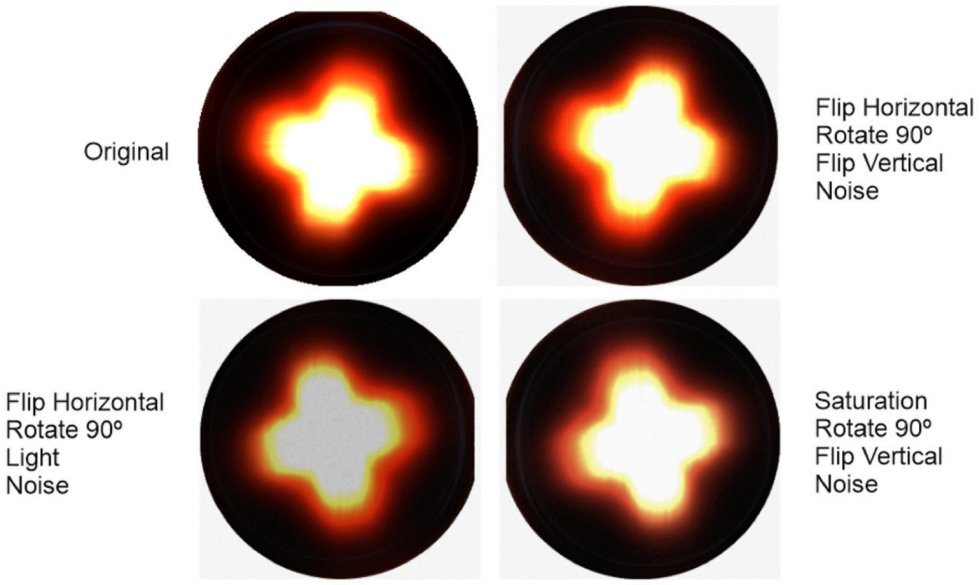


Figure 9. Example of augmented image transformations, adapted from [7].

Rather than applying fixed transformations, the script utilizes an approach where each augmented image is the result of a unique, randomized sequence of operations.

- Augmentation Factor: For every original image in the dataset, the system generates three additional augmented versions.
- Randomized Chains: Each augmented image is produced by applying a sequence of four randomly selected transformations from a predefined pool.

The augmentation is designed to try to simulate the physical and optical variations encountered during blood typing procedures.

- Geometric Transformations: To account for variations in reaction orientation or camera alignment, the pipeline applies 90-degree rotations, horizontal flips, and vertical flips.
- Brightness Tuning: Randomly adjusting brightness levels between 0.7× and 1.3× of the original intensity.
- Saturation Adjustment: Modifying color vividness to ensure the model focuses on the texture of the agglutination rather than specific reaction colors.
- Noise Injection: Gaussian noise is added to simulate electronic interference from the camera sensor, forcing the model to learn noise-invariant features. The noise is generated from a normal distribution with a mean of 0 and a standard deviation of 10, scaled to match the dimensions of the input image. After adding noise, pixel values are clipped to the standard [0, 255] range.

2.9. Embedded Deployment and Cross-Compilation

To ensure the blood typing system is viable for real-world POC applications, the software was designed to run on resource-constrained hardware. Specifically, the pipeline

was optimized and cross-compiled for a legacy ARM Cortex-A9 32-bit architecture, a processor that requires specific handling due to its 32-bit instruction set and limited floating-point performance compared to modern ×86_64 development environments.

The development was conducted on an ×86_64 host running Linux, requiring a cross-compilation strategy. The GNU Toolchain for ARM-A (arm-linux-gnueabihf) was utilized to generate binaries compatible with the target device.

The project relies on a complex stack of libraries that require individual cross-compilation for the ARMv7-A architecture:

- OpenCV: Compiled with NEON SIMD instructions enabled to accelerate image preprocessing tasks, such as the normalization and resizing required for the DINOv2 input.
- ONNX Runtime: A custom 32-bit build of the ONNX Runtime (v1.x) was produced for the target. Given the legacy nature of the ARM-A9, the execution provider was restricted to the CPU, utilizing MLAS (Microsoft Linear Algebra Subprograms) optimized for ARMv7 to handle the transformer-based inference of the DINOv2 model.
- mlpack and Armadillo: These libraries provide the logistic regression backend. They were linked against OpenBLAS (specifically the ARMv7 target) to provide linear algebra routines for the final classification stage.
- cnpy: A lightweight library used to bridge the gap between Python-based training and C++ inference by loading .npy weight and metadata files.

The deployment phase addressed several challenges inherent to the ARM-A9:

- Memory Management: Given the 32-bit address space, memory-mapped I/O and large buffer allocations for the ViT model were monitored to prevent exhaustion.
- Instruction Tuning: Compiler flags such as "-march=armv7-a", "-mfloat-abi=hard", and "-mfpu=neon" were enforced during the build process to use the hardware's floating-point unit.
- Inference Latency: While the primary model (DINOv2) is computationally intensive, the use of a C++ implementation for the entire pipeline, from image processing via OpenCV to classification via mlpack, minimized the overhead typically associated with interpreted languages like Python.

2.10. Embedded Performance Tests

The transition from a high-performance development environment to the resource-constrained nature of POC devices necessitates an evaluation of the model's efficiency. This section details the benchmarking of the optimized blood typing pipeline, specifically the DINOv2 feature extractor coupled with a logistic regression classifier, across a diverse array of hardware architectures. The objective was to determine the feasibility of the system in real-world scenarios, ranging from state-of-the-art mobile processors to the legacy single-core hardware found in the primary target device.

To ensure the reproducibility of results and to simulate production-ready deployment, each hardware platform was configured with a specific software environment designed to minimize background overhead and maximize the efficiency of the ONNX Runtime and C++ implementation.

- Standardized Linux Environments: For the QEMU Virtual Machine (VM) and the CRIAM device, a headless Linux-based distribution was utilized. By removing the Graphical User Interface (GUI), the system resources were dedicated entirely to the inference task, providing a clear baseline for the raw performance of the CPU and memory controller. Similarly, the models were simulated on a headless Raspberry Pi OS, which is optimized for the ARMv8 architecture while maintaining a minimal footprint.

- Android Containerization via Termux: The evaluation on Android devices (Snapdragon 808 and Snapdragon 8s Gen 4) presented a unique challenge due to the Android OS's management of background processes. To create a consistent testing environment comparable to the other Linux-based targets, Termux was used as a terminal emulator. Inside Termux, proot-distro was employed to install and run an Ubuntu environment. This setup allowed the execution of the same C++ binaries and linked libraries used on the other devices.

The benchmarking process was structured to analyze two primary dimensions: the inference speed of embedding and the cross-validation of the logistic regression. Each hardware tier was selected to test specific technical constraints:

- Virtualization and Emulation (QEMU VM): The QEMU environment was utilized to test the impact of virtualization overhead.
- High-End Mobile Architectures (Snapdragon 8s Gen 4): The analysis in this System on a Chip (SoC) was to evaluate the performance of a new high-end mobile SoC in this task.
- Mid-Tier Mobile (Snapdragon 808): The Snapdragon 808 was selected to evaluate the performance on an old mobile SoC, to see how the solution behaves in 10-year-old hardware.
- Ultra-Low Power IoT (Raspberry Pi Zero): The Raspberry Pi Zero served as a stress test for the system's memory footprint and an SoC for extremely low power consumption (2 Watts).
- Target Embedded Hardware (CRIAM Device): The single-core ARM Cortex-A9 architecture of CRIAM's prototype device represented the core target environment.

2.11. Criam Device Deployment

A challenge of deploying the proposed pipeline on the CRIAM prototype device, powered by a Texas Instruments AM4378 SoC running an older Linux distribution, was the incompatibility between the system's native C and C++ runtime libraries and those required by the classifier binary, which was compiled on a more recent toolchain. To overcome this without modifying the device's operating system or installing additional packages, a self-contained deployment strategy was adopted.

The classifier was shipped alongside a portable libs directory, some of them extracted from the QEMU VM, located alongside the binary, containing all necessary shared libraries, including the ARM hard-float dynamic linker "ld-linux-armhf.so.3", the Armadillo linear algebra library, and any other runtime dependencies required by the ONNX inference engine and the logistic regression model. At execution time, the binary is explicitly invoked through a dynamic linker using the "--library-path flag", which instructs the Linux loader to resolve all shared library dependencies from the local libs directory rather than from the system paths. This approach effectively creates a userspace-portable execution environment without requiring root privileges, package managers, or any modification to the host system.

To integrate this mechanism into the existing C++ codebase of the CRIAM device, a lightweight wrapper module was developed. The module exposes a single function, "classifyImage()", which accepts the path to an image and the path to the classifier binary, validates their existence, constructs and executes the appropriate shell command via "popen()", captures the standard output, and returns a structured result code: 1 for a positive reaction, 0 for a negative reaction. This design was intentionally kept dependency-free, relying only on the C and C++ standard libraries, so that it could be compiled and linked into the existing device code without introducing any new requirements on the target platform.

To complement the cross-validation results obtained on the training dataset, a final evaluation was conducted on a fully held-out set of 214 blood typing tests, using 856 individual reaction images (four per test), none of which had ever been used during training, validation, or any prior stage of model development.

3. Results

This chapter presents the performance evaluation of fourteen classification pipelines using stratified five-fold cross-validation, which was applied to all methods, on a binary dataset using 3090 images. For each model, accuracy, precision, recall, and F1-score were computed on the training, validation, and test partitions, and results are reported as mean values across folds, accompanied by standard deviations and observed ranges where applicable. Following the comparative analysis, the top-performing pipeline is examined in detail, covering its confusion matrix, misclassification profile, low-confidence prediction behavior, and cross-fold stability.

The chapter then extends this analysis to deployment considerations, evaluating the pipeline across four export formats to check the computational feasibility across a range of hardware platforms, from modern mobile chipsets to embedded systems. Inference and on-device refitting latency are reported for each platform, including the target CRIAM prototype device.

Finally, a final validation is conducted on a test set of 856 reaction images, never used before during the model's development, drawn from 214 complete blood typing tests, covering all eight blood types, to assess generalization beyond the cross-validation framework.

3.1. Overall Performance Summary

Across all evaluated models, high classification performance was observed, with several approaches achieving mean test accuracy above 99% (Table 2). Models using pretrained feature embeddings consistently outperformed those trained directly on raw pixel representations. Fine-tuned deep neural networks exhibited very good validation performance, with slightly reduced but still strong test results.

Table 2. Models' performance metrics ordered by F1-score.

Method	Model	Accuracy	Precision	Recall	F1 Score
Hybrid	DINOv2 + LR	99.87 ± 0.12%	100.00 ± 0.00%	99.75 ± 0.24%	99.87 ± 0.12%
Hybrid	DINOv2 + XGB	99.77 ± 0.08%	99.94 ± 0.13%	99.62 ± 0.13%	99.78 ± 0.08%
Hybrid	EfficientNetB4 + SVC	99.77 ± 0.08%	99.94 ± 0.13%	99.62 ± 0.13%	99.78 ± 0.08%
Hybrid	EfficientNetB4 + LR	99.77 ± 0.17%	100.00 ± 0.00%	99.56 ± 0.32%	99.78 ± 0.16%
Hybrid	EfficientNetB4 + XGB	99.77 ± 0.19%	99.87 ± 0.15%	99.68 ± 0.28%	99.78 ± 0.19%
Fine-Tune	NFNet	99.77 ± 0.08%	99.77 ± 0.08%	99.77 ± 0.08%	99.77 ± 0.08%
Fine-Tune	ResNet50	99.74 ± 0.08%	99.74 ± 0.08%	99.74 ± 0.08%	99.74 ± 0.08%
Hybrid	DINOv2 + SVC	99.71 ± 0.12%	99.94 ± 0.13%	99.49 ± 0.15%	99.71 ± 0.12%
Fine-Tune	EfficientNet B4	99.64 ± 0.16%	99.65 ± 0.16%	99.64 ± 0.16%	99.64 ± 0.16%
Fine-Tune	Inception V3	99.64 ± 0.19%	99.65 ± 0.19%	99.64 ± 0.19%	99.64 ± 0.19%
Fine-Tune	EfficientNet V2	99.48 ± 0.31%	99.49 ± 0.30%	99.48 ± 0.31%	99.48 ± 0.31%
Raw Pixel	Logistic Regression	97.96 ± 0.73%	98.97 ± 0.47%	97.03 ± 1.24%	97.98 ± 0.73%
Raw Pixel	XGB	97.73 ± 0.42%	97.85 ± 0.59%	97.72 ± 0.81%	97.78 ± 0.42%
Raw Pixel	SVC	95.02 ± 0.53%	95.96 ± 0.96%	94.24 ± 1.32%	95.08 ± 0.54%

The highest overall test performance was obtained by embedding-based models, particularly those using DINOv2 representations, which demonstrated both excellent accuracy and minimal variability across folds.

Embedding-based models demonstrated the lowest variability across folds, with standard deviations for test accuracy typically below 0.02%. Fine-tuned models exhibited

slightly higher variability, particularly in test performance, though still within a narrow range. Raw pixel-based models showed the highest variability, reflecting greater sensitivity to fold composition.

Three classifiers, logistic regression (LR), Support Vector Classifier (SVC), and XGBoost, were trained using DINOv2 embeddings. All DINOv2-based models achieved high performance across validation and test sets.

The DINOv2 embeddings combined with LR achieved the strongest overall results, with a mean test accuracy and an F1-score of 99.87%. Precision and recall were similarly high, with recall exceeding 99.7% on average. Performance variability across folds was minimal, with a standard deviation of approximately 0.12% for test accuracy.

The DINOv2 with SVC and XGBoost configurations also demonstrated robust performance, with mean test accuracies exceeding 99.7% and consistently high precision–recall balance. Across all three classifiers, validation and test metrics were closely aligned, indicating stable generalization across folds.

Models trained on EfficientNet-B4 embeddings showed performance comparable to the DINOv2-based approaches, though with slightly lower mean scores.

LR, SVC, and XGBoost classifiers trained on EfficientNet-B4 embeddings achieved a mean test accuracy of approximately 99.77%. F1-scores consistently exceeded 99.7%, and precision and recall remained well balanced across all folds.

Standard deviations across folds were low for all EfficientNet-B4 embedding models, indicating stable performance. Differences among classifiers within this embedding family were marginal, with no single classifier showing a statistically or practically dominant advantage over the others.

Five deep neural network architectures were fine-tuned: EfficientNet-B4, EfficientNet-V2, Inception-V3, NFNet, and ResNet50.

All fine-tuned models achieved very good validation results. In particular, the fine-tuned ResNet50 achieved a mean validation accuracy of 100% across folds, indicating perfect classification on validation splits. However, test performance was slightly lower, with a mean test accuracy of approximately 99.74%.

The fine-tuned NFNet exhibited the most consistent performance, with validation and test accuracy closely aligned and a mean test accuracy exceeding 99.7%. EfficientNet-B4 and EfficientNet-V2 fine-tuned models showed similar trends, achieving validation accuracy above 99.8% and test accuracy marginally below validation levels.

The fine-tuned Inception-V3 model, while still achieving high performance, exhibited slightly greater variability across folds relative to the other fine-tuned architectures.

Three classifiers, LR, SVC, and XGBoost, were trained using flattened raw pixel inputs. Among these, XGBoost trained on raw pixels achieved the highest performance, with a mean validation accuracy of approximately 97.73%. Test accuracy was lower than that of embedding-based and fine-tuned models but remained competitive relative to classical baselines.

The pixel-based logistic regression and pixel-based SVC models achieved lower overall performance, with validation accuracies ranging between 95 and 97%, and exhibited higher variability across folds compared to embedding-based approaches.

Overall, raw pixel-based models consistently underperformed relative to models using learned feature representations.

3.2. Statistical Significance of Performance Differences

To determine whether the accuracy differences observed across pipelines are statistically meaningful rather than attributable to chance, pairwise Nadeau-Bengio corrected paired *t*-tests were conducted across all model combinations, using the per-fold accuracy

values obtained from the stratified five-fold cross-validation. This correction accounts for the correlation between folds that share training data.

The results provide statistical support for three conclusions. First, all embedding-based pipelines, regardless of backbone (DINOv2 or EfficientNet-B4) or downstream classifier, significantly outperformed all raw pixel-based approaches ($p < 0.002$ in every case), confirming that learned feature representations provide a meaningful and reliable advantage over direct pixel inputs for hemagglutination classification.

Second, no statistically significant difference was found between DINOv2 and EfficientNet-B4 as embedding backbones under any classifier pairing ($p \geq 0.059$ in all cases), indicating that the performance advantage of embedding-based methods is attributable to the paradigm itself rather than to any specific backbone architecture. The ~0.1% accuracy gap between the two families should therefore not be interpreted as evidence of backbone superiority.

Third, DINOv2 + LR significantly outperformed four of the five fine-tuned CNN architectures evaluated: EfficientNet-B4 ($p = 0.034$), EfficientNet-V2 ($p = 0.026$), Inception-V3 ($p = 0.034$), and ResNet50 ($p = 0.003$). The sole exception was NFNet ($p = 0.059$), which represents the strongest fine-tuned baseline and cannot be statistically distinguished from the best embedding pipeline. Within the DINOv2 embedding family, LR was significantly superior to SVC ($p = 0.036$) but statistically equivalent to XGBoost ($p = 0.059$). The selection of logistic regression as the recommended downstream classifier therefore rests primarily on its substantially lower inference latency and its suitability for on-device refitting under the computational constraints of the target hardware, rather than on statistical accuracy superiority alone, and the selection of the DINOv2 backbone, instead of EfficientNet-B4, was because the embedding of the former have a 1792 dimensional vector, against a much smaller one from DINOv2-small, with a 384 dimensional vector. The smaller vector makes the RAM and storage footprint smaller, gives better fitting speed and inference with the logistic regression.

3.3. DINOv2 with Logistic Regression Performance Review

Among all fourteen classification pipelines evaluated, the hybrid architecture combining DINOv2 embeddings with a logistic regression classifier is, among the top solutions, the most suitable for low-power devices. This section provides a detailed analysis of the model's behavior, decomposing its performance across folds, examining the confusion matrix, characterizing the misclassified cases, and discussing the confidence profile of the classifier's predictions.

The confusion matrix presented in Figure 10 provides a view of the model's classification outcomes across the full 3090-image dataset by pooling predictions from all five folds. The matrix reveals four distinct outcome categories:

- Out of 1509 true negative images (non-agglutinated reactions), the model correctly classified all of them, with zero false positive predictions. This is reflected in the perfect precision score and confirms that the model does not mistakenly assign a positive agglutination.
- Out of 1581 true positive images (agglutinated reactions), the model correctly classified 1577 and misclassified 4, resulting in four false negative predictions. These represent cases where a genuine agglutination reaction was labeled by the model as negative. This outcome is the sole source of error in the entire evaluation and exclusively affects recall, registering at 99.75%.

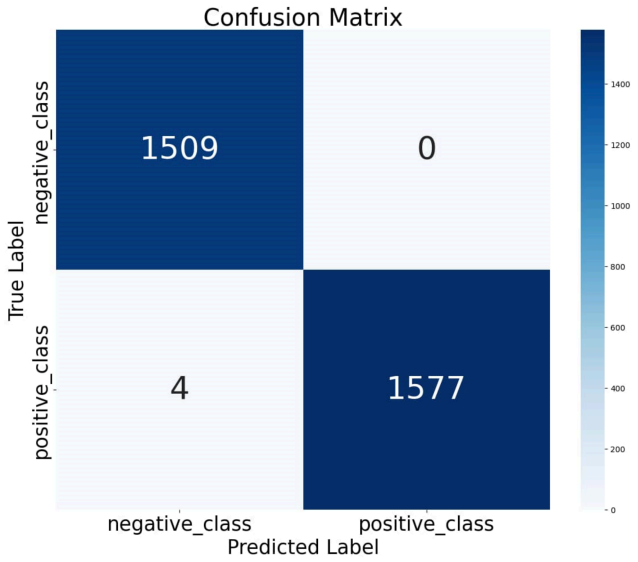


Figure 10. Confusion matrix of the DINOv2 + LR model.

Collectively, the model achieved 3086 correct classifications out of 3090 images, equating to 99.87% accuracy. The overall error rate across the dataset is therefore 0.13%, or approximately one error per 772 predictions.

Visual inspection of the misclassified images, presented in Figure 11, reveals that all four cases share a common characteristic: they are borderline "Weak Positive" reactions in which the degree of agglutination is substantially lower than that observed in clear, high-confidence positive reactions.

In standard hemagglutination reactions, a strongly positive result manifests as a distinctly irregular, fragmented pattern with clearly visible spots of red cells surrounded by transparent zones (Figure 3). A strongly negative result presents a smooth, uniform disk of red cells with no visible disruption to the surface. The misclassified images occupy an intermediate space, displaying a subtle textural irregularity that is not sufficiently pronounced to be unambiguously classified as positive by the linear decision boundary learned by the logistic regression classifier atop the DINOv2 embedding space.

The fact that this system's error set is limited exclusively to these borderline reactions, and does not include any clear, high-intensity reactions, confirms that the DINOv2 embedding successfully distinguishes unambiguous cases and that its residual error is confined to a challenging subset.

Further analysis of the four misclassified images revealed that all originate exclusively from well 4 of the four-well cartridges used in the CRIAM device, corresponding to the Anti-Rh/D reagent, which is known to be a less reactive antigen in certain people. Wells 1, 2, and 3 present zero classification errors across the entire dataset, corresponding respectively to the Anti-A, -B, and -AB reagents.

Considering that well 4 accounts for 773 images, its individual well-level accuracy is 769 out of 773 correct classifications, equivalent to 99.48%. This concentration of errors is clinically and technically meaningful for two reasons. First, from a diagnostic standpoint, a complete blood type determination requires the correct classification of all four wells simultaneously; the fact that the system's entire residual error is confined to a single reagent position means that the reliability of wells 1, 2, and 3 is perfect, and that targeted improvements to well 4 alone would be sufficient to achieve full system accuracy. However, we should also consider the accuracy of well 4 on its own to determine the true performance of the model.

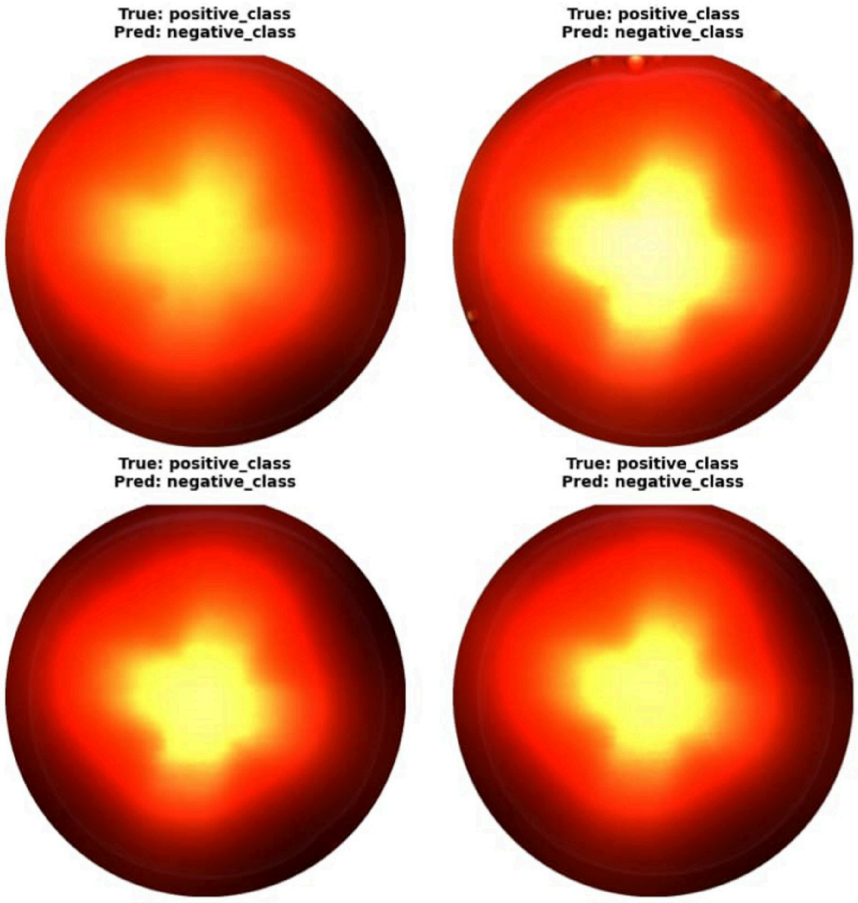


Figure 11. Weak positive reactions misclassified by DINOv2 + LR.

Beyond the binary classification of correct and incorrect predictions, Figure 12 illustrates images of reactions that, while correctly classified, were assigned relatively low confidence scores by the LR classifier. These low-confidence predictions are defined as cases where the classifier's predicted class probability falls below approximately 70%, indicating that the image lies close to the decision boundary in the DINOv2 embedding space. The visual characteristics of these low-confidence correct predictions are consistent with those of the misclassified images: they typically present either as weak positive reactions with diffuse, low-intensity agglutination patterns, or as negative reactions whose well surface exhibits a slight unevenness potentially attributable to minor lighting artifacts or air bubbles.

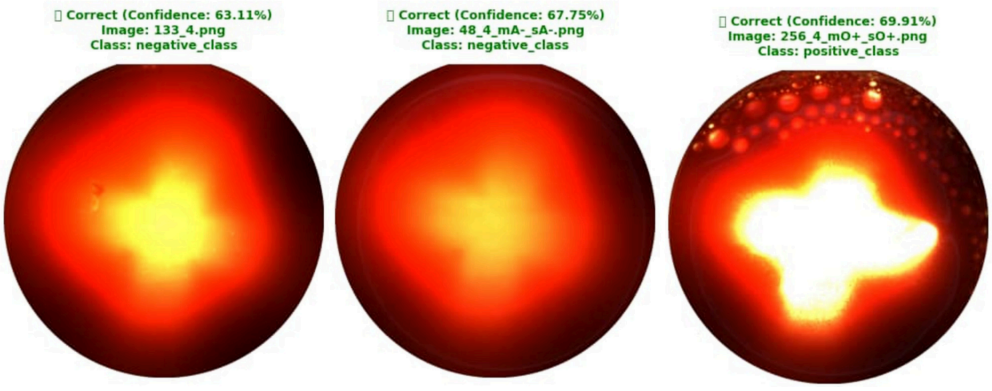


Figure 12. Low confidence classifications of DINOv2 + LR.

This suggests that the DINOv2 embedding maps the most ambiguous biological signals to a region of the feature space that is close to the linear decision boundary, a configuration that is consistent with the known difficulty of these borderline cases.

The existence of this low-confidence cluster has a practical implication for clinical deployment. A confidence-aware inference system, one that flags predictions below a defined threshold for secondary review by a trained technician, could be implemented without compromising the high-throughput performance of the pipeline for unambiguous cases. This would directly address the precision–safety trade-off inherent in fully automated blood typing and align with current regulatory frameworks in medical diagnostics, which typically require a human review pathway for borderline results.

The standard deviation of the test accuracy across the five folds was 0.12%; this low variance indicates that the model's performance is not a product of a fortunate data partition but rather reflects a genuinely consistent generalization capability across different subsets of the 3090-image dataset. Similarly, the standard deviation of recall across folds was 0.24%, slightly higher than that of accuracy and precision, which is expected given that recall is sensitive to the four false-negative images and their distribution across folds. In each fold where two or more of the four misclassified images are present in the test partition, recall will be marginally lower. The fact that precision maintained a standard deviation of 0.00% across all folds is a direct consequence of the model generating no false positive predictions in any fold, confirming that this result is structurally robust and not fold-dependent.

Within the full ranking of fourteen pipelines presented in Table 2, the DINOv2 + LR model occupies the top position by mean test F1-score. Its second closest competitor, DINOv2 + XGBoost, achieved a mean test accuracy of 99.77 $\pm$ 0.08% and a mean test F1-score of 99.78 $\pm$ 0.08%, indicating that the choice of downstream classifier has a measurable but modest impact on final performance when DINOv2 embeddings are used. The LR classifier appears to be an advantage rather than a limitation in this context: the DINOv2 embedding space produces sufficiently separable representations of positive and negative reactions that a non-parametric linear boundary is adequate, while simultaneously being computationally lightweight enough to be fitted and evaluated within milliseconds on embedded hardware.

3.4. Optimization Performance Analysis

The experimental analysis evaluated the DINOv2-small + LR pipeline across four ONNX exports. Performance was measured by classification accuracy, model file size in megabytes (MB), and mean inference latency in milliseconds (ms). As shown in Table 3, the baseline PyTorch FP32 model achieved an accuracy of 99.87 $\pm$ 0.12%. Exporting to ONNX FP32 yielded a 10% improvement in inference speed and a minor 2% reduction in storage size while maintaining identical accuracy. The ONNX INT8 variant proved the most efficient for CPU deployment, achieving a significant 63% reduction in size (down to 32.7 MB) and a 36% reduction in latency (343.77 ms). The accuracy trade-off was minimal, with a final score of 99.81 $\pm$ 0.12%, representing a difference of only 0.06% from the baseline. Conversely, while ONNX FP16 halved the model size, it resulted in a 90% increase in latency on the CPU (1019.22 ms), confirming that half-precision may not be suitable for non-specialized CPU hardware.

The ONNX exports, combined with the augmented dataset, emerged as superior models. While the standard INT8 quantization typically introduces a slight performance penalty, the inclusion of data augmentation successfully mitigated this, elevating the test accuracy to 99.97 $\pm$ 0.03% and the test recall to 99.94 $\pm$ 0.06%, as shown in Table 4. The results were also obtained through a 5-fold cross-validation. When evaluated with the augmented dataset, the ONNX INT8 model achieved a higher accuracy of 99.97%, while the

FP32 and FP16 models achieved 99.94% and 99.95% respectively. Although the INT8 model obtained a slightly higher value, the difference is marginal and should not be interpreted as evidence of the intrinsic superiority of INT8 quantization. Instead, the results indicate that quantization preserved the classification performance under the evaluated conditions. Such small variations may be attributed to numerical effects introduced by quantization, the specific distribution of the augmented/test samples, and the limited error margin at near-perfect accuracy levels.

Table 3. Quantization performance metrics ordered by accuracy.

Model Variation	Size MB	Time	Accuracy
CPU Torch FP32	88.3	536.07	0.9987 ± 0.0012
CPU ONNX FP32	86.3	482.55	0.9987 ± 0.0012
CPU ONNX FP16	43.3	1019.22	0.9987 ± 0.0012
CPU ONNX INT8	32.7	343.77	0.9981 ± 0.0012

Table 4. Detailed performance metrics, with augmented dataset (AD) and original dataset (OD), ordered by F1-score.

Model	Test Accuracy	Test Precision	Test Recall	Test F1-Score
ONNX INT8 AD	99.97 ± 0.03%	100.00 ± 0.00%	99.94 ± 0.06%	99.97 ± 0.03%
ONNX FP16 AD	99.95 ± 0.03%	100.00 ± 0.00%	99.92 ± 0.24%	99.95 ± 0.03%
ONNX FP32 AD	99.94 ± 0.02%	100.00 ± 0.00%	99.91 ± 0.24%	99.94 ± 0.02%
ONNX FP32 OD	99.87 ± 0.12%	100.00 ± 0.00%	99.75 ± 0.24%	99.87 ± 0.12%
ONNX FP16 OD	99.87 ± 0.12%	100.00 ± 0.00%	99.75 ± 0.24%	99.87 ± 0.12%
ONNX INT8 OD	99.81 ± 0.12%	100.00 ± 0.00%	99.62 ± 0.24%	99.81 ± 0.12%

The comparative analysis highlights two trends regarding the use of image data augmentation:

- Error Margin Reduction: The models trained with augmented images exhibited a significantly lower standard deviation across all test metrics. For instance, the test F1-score variance dropped from 0.12% in the baseline to 0.03% for INT8 and FP16, and 0.02% for FP32, suggesting that the augmented dataset improved the model's generalization and stability.
- Mitigating Quantization Loss: Although the standalone ONNX INT8 model saw a marginal dip in test recall (99.62%), the Augmentation variant compensated for the loss and surpassed all metrics prior to augmentation. This demonstrates that data-centric optimizations can effectively offset the precision loss inherent to integer quantization.

Further analysis of the four misclassified images in the INT8 + Augmented dataset also revealed that all originate exclusively from well 4. Considering that well 4 accounts for 3092 images in the augmented dataset, its individual well-level accuracy is 3088 out of 3092 correct classifications, equivalent to 99.87%.

The following metrics highlight the computational efficiency and scalability of the proposed pipeline for deployment on resource-constrained hardware:

- Low Random Access Memory (RAM) Requirement: The solution requires as little as 1 GB of RAM, making it feasible for edge devices.
- Minimal Storage: The embeddings for the entire 12,360 images from the augmented dataset occupy only 18.4 MB of storage; a single image embedding takes around 1.5 kB of storage.
- With this solution, it is feasible to re-fit the model with a new set of images using only hardware available to a portable device.

- This efficiency is critical for battery-operated devices, as it minimizes power consumption and allows for operation independent of the electrical grid.

3.5. Hardware Simulation Results

To evaluate the practical viability of the proposed pipeline in point-of-care settings, the computational time was measured across a diverse range of SoCs, from modern mobile chipsets to resource-constrained embedded boards. The benchmarks focused on two primary tasks: the generation of a single-image embedding and the execution of a 5-fold cross-validation and a final fit with 100% of the dataset for the LR classifier.

The results reveal a significant performance gap between high-end mobile processors and low-power embedded systems (Table 5). The Snapdragon 8S Gen 4 demonstrated very good speed, processing an image in just 0.49 s. In contrast, the AM4378 from CRIAM's prototype device took 70.08 s per image, considerably slower than the Raspberry Pi (RPI) Zero 2W.

Table 5. Hardware speed comparison ordered by generation of embedding.

Device	Embedding Generation (s/img)	Logistic Regression 5-Fold Cross-Validation + Final Fit (s)
SD 8S Gen 4	0.49	0.96
SD 808	6.8	6.66
RPI Zero 2W	13.68	13.91
QEMU VM	34.66	22.03
AM4378	70.08	58.79

An important finding is the efficiency of the LR fitting process. The least powerful hardware tested, the 5-fold cross-validation, plus the final fit, was completed in 58.79 s. This confirms that once an embedding is generated, the system can be updated or recalibrated with new data directly on the device without requiring cloud connectivity or specialized high-performance computing. This "learning" at the edge is a significant advantage for adaptive point-of-care diagnostics.

While embedding generation remains the primary bottleneck on lower-end hardware, the overall pipeline is feasible for non-instantaneous diagnostic workflows. For high-throughput environments, modern SoC hardware like the Snapdragon 8S series is recommended to maintain a fluid user experience.

3.6. Criam Device Deployment Tests

To validate the pipeline under the target device, the full blood typing workflow was executed directly on CRIAM's prototype device, which is powered by a Texas Instruments AM4378 SoC (ARM Cortex-A9, single-core). This represents the most constrained hardware target in this study and reflects a testing environment on a legacy processor.

The primary computational bottleneck remains the DINOv2 embedding generation stage. On the AM4378, a single image embedding required approximately 70.08 s. With four wells requiring sequential processing, the total embedding generation time per blood typing run was approximately 4.7 min. The LR classifier 5-fold cross-validation on this hardware completed in 58.79 s. However, single-pass inference of the whole dataset is only 17.99 s, confirming the low cost of the classification stage across all hardware tiers. Additionally, the LR fit is saved for later use, so it does not need to be fitted every time the machine is used.

3.7. Final Validation Tests

The optimized ONNX INT8 DINOv2 pipeline, retrained with the augmented dataset and paired with the logistic regression classifier, was subjected to a final evaluation. The evaluation was conducted on a fully held-out set of 214 complete blood typing tests, using 856 individual well images (four per test), none of which had ever been used during any prior stage of model development, covering all eight clinically relevant blood types (A+, A−, AB+, AB−, B+, B−, O+, O−). The pipeline achieved an overall accuracy of 99.53% (Table 6), correctly classifying 213 out of 214 complete tests, with a corresponding F1-score of 99.54%, precision of 99.58%, and recall of 99.53% (Table 7). Seven of the eight blood types, namely A+, AB+, AB−, B+, B−, O+, and O−, were classified with perfect 100% accuracy, with the sole source of error confined to the A− group, where one out of 41 samples was misclassified. These results confirm that the performance levels observed during cross-validation on the original dataset generalize to unseen samples, validating the pipeline's suitability.

Table 6. Class distribution of the held-out test set and accuracy of the INT8-quantized DINOv2 + LR model with data augmentation.

Blood Type	Total Samples	Correct Predictions	Accuracy
A+	52	52	100.00%
A−	41	40	97.56%
AB+	9	9	100.00%
AB-	1	1	100.00%
B+	24	24	100.00%
B-	4	4	100.00%
O+	50	50	100.00%
O−	33	33	100.00%

Table 7. Performance metrics of final validation tests.

Metric	Value	95% CI
Accuracy	99.53%	[97.54%, 99.93%]
Precision	99.58%	[97.72%, 99.95%]
Recall	99.53%	[97.54%, 99.93%]
F1-Score	99.54%	[97.58%, 99.93%]

Confidence intervals were computed using the Wilson score method at the 95% level (n = 214). Despite the small number of errors, the lower bounds confirm that accuracy exceeds 97.5% with high confidence.

Per-reagent classification accuracy further reinforces the strength of the pipeline: Anti-A and Anti-AB reagents (wells 1 and 3) achieved a perfect 100% across all 214 tests, while Anti-B and Anti-D (wells 2 and 4) reached 99.53%, with a single image-level error each.

The only misclassified test in this evaluation was a true "A−" sample that was predicted as "AB+" (Figure 13). Inspection of the per-reagent predictions for this test revealed that both the Anti-B (well 2) and Anti-D (well 4) wells were incorrectly classified as positive when the ground truth is negative for both (Figure 4). This simultaneous false positive across two independent wells is inconsistent with the behavior observed in every other test previously and is not attributable to a borderline or weak agglutination signal of the type identified as the residual error source in Section 3.2.

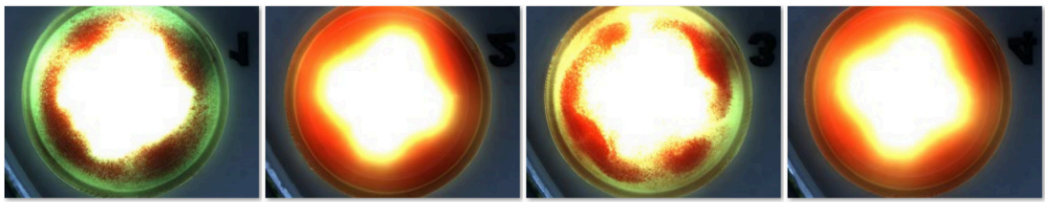


Figure 13. Misclassified test images of the final validation dataset.

Visual examination of the four reaction images from this test identified the root cause as an abnormally strong lighting affecting the well surface, possibly due to low sample or reagent volume. The overexposed regions within the images altered the texture and intensity distribution to a degree sufficient to push the DINOv2 embeddings across the linear decision boundary of the LR classifier. This finding is meaningfully distinct from the weak reaction errors described in Section 3.2. Whereas the training set misclassifications arose from ambiguity, the single error observed here is a consequence of an atypical acquisition condition that falls outside the photometric range represented in the training data. This distinction reinforces the diagnostic reliability of the pipeline under normal operating conditions and simultaneously identifies a concrete and addressable failure mode: sensitivity to abnormally strong illumination.

4. Discussion

This chapter presents an interpretation of the experimental results obtained throughout this study, discussing the practical significance of the proposed hybrid pipeline, the factors underlying its performance, and the trade-offs encountered during deployment. The chapter also examines the impact of data augmentation and INT8 quantization, analyses the computational feasibility of the system across embedded hardware targets, and addresses limitations that must be resolved.

The hybrid DINOv2 + LR pipeline emerged as the best all-rounder approach, among the top models in accuracy and the most suitable for low-power devices. achieving a mean test accuracy of 99.97 $\pm$ 0.03% and a mean test recall of 99.94 $\pm$ 0.06% when the model was trained with image data augmentation and exported as ONNX INT8. Embedding-based methods (DINOv2, EfficientNet-B4) consistently outperformed raw-pixel models and fine-tuned CNNs in both accuracy and cross-fold stability. Quantization to INT8 produced gains in storage footprint and CPU latency with negligible accuracy loss, and data augmentation successfully mitigated the small recall drop typically introduced by quantization.

4.1. Interpretation and Practical Significance

These results indicate that a data-centric hybrid architecture (pretrained embedding + lightweight classifier) is especially well suited for POC hemagglutination detection where compute, memory, and power are constrained. The combination of strong pretrained visual representations, compact linear or tree-based classifiers, and quantization-aware deployment produces very good results while remaining feasible on low-power hardware with 1 GB RAM. The low variability across folds also suggests the model can be reliable across routine variations in the acquisition process. The combination of DINOv2 (a self-supervised Visual Transformer) with a lightweight logistic regression classifier is genuinely underexplored. Most prior work used CNNs or YOLO variants for the task. The discovery that a frozen ViT backbone produces embeddings linearly separable enough for blood typing is an interesting finding. The quantization + augmentation is also meaningful, showing that INT8 compression losses can be compensated through data augmentation.

4.2. Why Embeddings Outperform Raw Pixels and Fine-Tuning

Pretrained embeddings encode high-level, generalizable features (shapes, textures, pattern co-occurrence). These features appear to capture the salient visual cues of agglutination better than pixel vectors and enable simpler classifiers to separate classes. Fine-tuning deep CNNs yielded strong results, but with slightly higher variance, and they need to be trained on fast GPUs. This makes them less attractive for resource-constrained deployment compared to shallow classifiers on top of pretrained embeddings, and it is impractical to retrain the classifier within a low-power device.

4.3. Role and Value of Data Augmentation

Image data augmentation proved critical in two ways:

- Performance recovery after quantization: Data augmentation compensated for representational information lost during INT8 compression, improving recall and reducing metric variance.
- Regularization and stability: Augmented data reduced overfitting, producing smaller standard deviations across folds and stabilizing model behavior.

The augmentation strategy used here was deliberately simple (rotations, flips, brightness, saturation, Gaussian noise). The positive impact suggests room to explore richer augmentation approaches and even generative AI to create synthetic images.

4.4. Quantization Optimization and Trade-Offs

The ONNX INT8 conversion provided a substantial reduction in storage (≈75%) and latency, with a minimal accuracy penalty (~0.06% absolute). Conversely, FP16 on CPU increased latency considerably and is therefore unsuitable unless GPU acceleration is available. These findings suggest that quantization targets should match the intended runtime hardware.

4.5. Deployment in the Embedded Device

The data reveals that the feature extraction stage (DINOv2 embedding generation) is the primary computational bottleneck of the pipeline. On the high-end Snapdragon 8S Gen 4, this process takes less than half a second (0.49 s), enabling a near-instantaneous response. However, as we move to resource-constrained devices like the Raspberry Pi Zero 2W, the latency increases to 13.68 s per image. While this ~28 times increase in latency makes the RPi Zero 2W unsuitable for real-time video applications, it remains functional for static diagnostic workflows.

In contrast to the embedding generation, the LR classifier shows good efficiency across all platforms. Even on the RPi Zero 2W, a full 5-fold cross-validation (which involves multiple training and evaluation passes) completes in just 13.91 s. This finding is critical for the development of adaptive diagnostics. It implies that the system can be "field-personalized", trained on a small set of local samples or recalibrated for different conditions, directly on the device without needing a cloud server. This "Learning at the Edge" capability is a direct result of using fixed foundation model embeddings, which compress high-dimensional image data into a compact, linearly separable feature space.

The AM4378 (TI Sitara, ARM Cortex-A9) in CRIAM's prototype device represents the most restrictive deployment target evaluated in this study. Operating as a single-core processor with a constrained 32-bit address space, the device required approximately 70.08 s to generate a single DINOv2 embedding, placing it well below even the Raspberry Pi Zero 2W. For a complete four-well blood typing run, this translates to a total time-to-result of approximately 4.7 min, as the four embedding passes must be executed sequentially. The logistic regression stage continued to contribute negligible overhead, completing the

full 5-fold cross-validation in 58.79 s. These results confirm that the proposed pipeline is functionally viable on the target CRIAM hardware and that the primary path to improved user experience lies in hardware modernization.

4.6. Impact of Lighting and Reaction Sensitivity

The misclassification of very weak reactions shows a critical hurdle in automated blood typing: the detection of low-titer or weak reactions under inconsistent illumination. While the system reliably identifies robust positive signals, "weak positive" reactions constitute a borderline case difficult to address, and the linear brightness augmentation was also insufficient to address this limitation. In transfusion medicine, misclassifying these weak positives as false negatives can result in potentially fatal hemolytic transfusion reactions, underscoring the need to adjust clinical safety thresholds for the confidence level of the prediction. This diagnostic error suggests that the current computer vision pipeline maintains a sensitivity threshold that barely exceeds the baseline. Because the true negative reactions and weak positive both exhibit a similar "smooth" morphology, the model lacks the discriminative power required to differentiate subtle agglutination from a negative result. To address these limitations in future iterations, we propose the following improvements:

- Multi-Exposure Image Capture: Implementing a multi-exposure strategy will ensure that the fine texture of weak reactions is preserved, preventing the device's internal light source from "washing out" subtle indicators.
- Temporal Differential Learning: By capturing images both before and after the reaction, the model can explicitly learn the temporal transitions associated with agglutination, rather than relying solely on a static endpoint.

4.7. Limitations

The following limitations should be considered:

- Single-device data source: A limitation of this study is that the dataset was collected using a single blood-typing device under controlled conditions. As a result, the model may not fully capture variability arising from differences in hardware, image acquisition settings, or device calibration. Collecting data from multiple devices and environments would likely improve the robustness and generalization capability of the proposed approach, particularly for real-world point-of-care deployment
- Data augmentation heuristics: The current data augmentation pipeline is heuristic, as the set of transformations applied to the images is expected to vary and may not fully capture the variability of real-world data.
- Independent multi-center clinical validation absent: Although the pipeline was assessed using stratified cross-validation and an additional held-out validation set, it has not yet been evaluated prospectively across multiple clinical sites, operators, reagent lots, and independent device units. Such validation will be required before clinical deployment.

4.8. Implications for Dataset Expansion and Multi-Device Testing

The results of the final evaluation strengthen the evidence base for the proposed pipeline and confirm that the performance levels observed under cross-validation on the original 3090-image dataset generalize to entirely new samples. However, the lighting-induced misclassification provides additional empirical support for the limitation already discussed in Section 4.6, namely that the current dataset originates exclusively from a single prototype device operated under controlled laboratory conditions.

The fact that a single anomalous lighting condition was sufficient to produce this error shows how important it is to expand the training dataset to images acquired across multiple units of the CRIAM device and across a variety of real-world operating environments. Different device units may exhibit subtle differences in internal illumination intensity and spatial uniformity arising from manufacturing tolerances and component aging. Beyond illumination, other environmental and device-related factors, such as ambient temperature, dust accumulation, or device age and wear, may affect the quality of the reactions or the captured images. Testing across a wider range of conditions and devices would be essential to ensure the model's robustness in real-world deployment. This multi-device, multi-environment data collection strategy is therefore identified as an action for advancing the system towards clinical deployment, complementing the augmentation-based and architecture-level improvements outlined in Section 4.9.

4.9. Suggestions to Mitigate Limitations

The following recommendations address the identified limitations and outline a roadmap for strengthening the system's reliability, generalizability, and on-device performance:

- Data collection expansion: Prioritize collection across multiple devices, reagent lots, and clinical sites, and the inclusion of more sample edge cases.
- Evaluating real-world field testing: Continuous testing and validation of on-device generalization against a broader spectrum of clinical blood samples and different devices.
- Augmentation ablation studies: Quantify the marginal benefit of each augmentation type and the augmentation factor (1× vs. 3× vs. more). Explore generative models (GANs, diffusion) to produce higher-fidelity synthetic images.
- Continuous improvement: Implemented by integrating new image embeddings during device operation and dynamically refitting the logistic regression model to adapt to new data.
- Quantization-aware training (QAT): Instead of post-training quantization only, adopt QAT to further reduce accuracy loss on INT8 and allow more aggressive optimizations.
- Model compression strategies: Evaluate pruning, knowledge distillation (student models trained to mimic large models), and per-layer quantization granularity to improve latency/memory further.

Beyond immediate deployment, the framework can be expanded through several methodological advancements to enhance robustness and capability:

- Classification precision could be refined by analyzing paired images (captured both before and after the reaction) to allow the model to explicitly learn temporal differences associated with agglutination.
- Alternative backbone architecture, including emerging self-supervised models such as DINOv3, can be explored.
- Exploration of federated learning in which multiple devices, belonging to different users, collaboratively train a unified model to improve overall performance, without sending any sensitive data.

4.10. Operationalization and Edge Deployment

The following practical considerations should guide the transition from a validated prototype to a production-ready edge deployment:

- Runtime optimizations: Use ONNX Runtime with optimization, preferably with INT8 on the CPU.
- Update strategy: Implement secure over-the-air updates with versioning, rollback capability, and post-update validation.
- Human factors: Design the user interface to show raw images and model heatmaps.

4.11. Translational, Regulatory, and Reporting Considerations

The literature review highlighted that performance claims for AI-enabled in vitro diagnostic systems should not rely solely on internal cross-validation. Diagnostic AI studies should be reported transparently and evaluated in relation to accepted diagnostic-accuracy and prediction-model reporting frameworks, including STARD, STARD-AI, TRIPOD + AI, and PROBAST + AI [32,33]. These frameworks are particularly relevant for image-based blood typing because model performance may be affected by acquisition hardware, illumination, reagent lot, operator handling, and the prevalence of weak or borderline agglutination patterns.

From a translational perspective, future development of the proposed system should also align with the regulatory logic applicable to in vitro diagnostic medical devices. Under the European IVD framework, performance evaluation is expected to consider scientific validity, analytical performance, and clinical performance, while clinical performance studies with human specimens should follow appropriate IVD performance-study principles [34–38]. Therefore, the current results should be interpreted as strong technical and analytical feasibility evidence for the proposed edge-AI pipeline, but not yet as sufficient clinical evidence for regulated use.

A practical validation roadmap should include the following: (i) locked-model testing on prospectively collected samples; (ii) multi-device and multi-site image acquisition; (iii) controlled stress testing for illumination, reagent volume, cartridge variation, and weak-positive reactions; (iv) comparison with a recognized reference method; (v) predefined acceptance criteria for sensitivity, specificity, invalid-result rate, and false-negative risk; and (vi) usability and operator-error assessment for point-of-care workflows. This roadmap would directly address the main limitations identified in this study and would strengthen the evidence required for eventual clinical translation.

5. Conclusions and Future Work

This study demonstrates that a hybrid inference pipeline based on pretrained visual embeddings and lightweight classifiers achieves good performance for hemagglutination detection while remaining compatible with strict computational and memory constraints. Across fourteen evaluated pipelines, embedding-based methods consistently outperformed both raw pixel representations and fine-tuned, pretrained convolutional architectures.

The DINOv2-small + LR classifier configuration emerged as the most effective solution, achieving a mean test accuracy of $99.97 \pm 0.03\%$ and a mean test recall of $99.94 \pm 0.06\%$ when combined with stochastic data augmentation and INT8 quantization. The results provide strong empirical evidence that representational quality dominates classifier complexity in this domain. Self-supervised ViT embeddings captured agglutination patterns with sufficient generality to allow linear decision boundaries to separate reactions reliably, even under aggressive model compression.

Furthermore, post-training INT8 quantization of the embedding extractor achieved significant reductions in model size and latency with negligible performance degradation. Notably, the introduction of data augmentation not only mitigated potential recall loss from quantization but also resulted in a net improvement over the FP32 baseline. From a methodological standpoint, these hybrid pipelines decouple feature learning from decision modeling, which is advantageous for point-of-care systems. The compact embedding space ($\approx$1.5 kB per image) enables rapid refitting and incremental updates without retraining the backbone, allowing the system to adapt to distributional shifts while preserving deterministic inference behavior. It is important to distinguish between technical feasibility and clinical readiness in this context. While the 4.7 min processing time on the AM4378 demonstrates that the pipeline is functionally deployable on the target hardware, this latency may

not meet the throughput requirements of time-critical transfusion workflows. The result should therefore be interpreted as proof of computational viability on legacy hardware, not as evidence of clinical suitability. Hardware modernization, as demonstrated by the sub-5 s performance on contemporary ARM64 processors, remains a recommendation for deployment in urgent diagnostic scenarios.

Final evaluation on a test set of 214 complete blood typing tests, with 856 individual reaction images covering all eight blood types and never used at any prior stage of model development, confirmed that the cross-validation performance generalizes to entirely new samples. The system achieved an overall accuracy of 99.53% (213 out of 214 tests correctly classified) and an F1-score of 99.54%. The sole misclassified test was attributed not to borderline agglutination ambiguity, but to an abnormally strong and non-uniform lighting condition that pushed the DINOv2 embeddings across the classifier's decision boundary, an atypical acquisition artifact outside the photometric range represented in training data. This finding both confirms the diagnostic reliability of the pipeline under normal operating conditions and identifies a concrete, addressable failure mode that multi-device and multi-environment data collection strategies can directly target.

Ultimately, this study establishes that quantization-aware, embedding-driven architectures offer a scalable, robust, and efficient blueprint for low-resource medical imaging applications.

While the current results provide a solution for automated blood typing, several approaches for further research and development can be done to improve the technology:

- External and Multi-Center Clinical Validation: the priority should be the expansion of the dataset to include multi-device and multi-site validation. Since the current study relied on data from a single device, future studies must perform model generalization across diverse hardware iterations, varying ambient lighting conditions, and different reagent lots.
- Generative AI: The success of simple data augmentation suggests that more sophisticated data generation techniques could further enhance model robustness. Future research should explore the use of generative AI to synthesize more reaction images, especially weak agglutination reactions, which are often underrepresented in clinical datasets. By creating synthetic images of these "edge cases," the model can be better trained to recognize reactions that are currently difficult to capture.
- Temporal Differential Learning: Instead of relying on a static endpoint image, the system could analyze paired images, captured both before and after the reaction period. By learning the temporal transition (the "process" of agglutination), the model can more effectively differentiate between true negative results and subtle changes.
- Architectural Evolution and Federated Learning: As the field of self-supervised learning advances, the pipeline should be tested with emerging backbones like DINOv3 to see if further gains in representational quality can be achieved. Additionally, the exploration of Federated Learning offers a path for decentralized updates. This would allow multiple CRIAM POC devices to collaboratively train a global model by sharing local model updates (weights) rather than sensitive patient data, ensuring privacy while continuously improving the system's collective intelligence.
- Optimization and Interpretability: To further reduce latency on the lowest-end hardware, such as the Raspberry Pi Zero 2W, future iterations should evaluate quantization-aware training (QAT), model pruning, and knowledge distillation. Also, the system's "explainability" must be improved. Integrating Heatmaps into the user interface would allow medical professionals to see exactly which areas of the image the AI is focusing on, providing a "second opinion" that is transparent and verifiable.

- Broadening the Diagnostic Scope: The framework developed here for ABO and Rh typing can be tested in other agglutination assays, expanding the system's capabilities to multiple tests or diseases.
- Upgrade the PoC device hardware to accelerate inference. Testing demonstrates that a modern ARM64 SoC can process the four images in under 5 s, which enhances the user experience and increases effectiveness in urgent scenarios.
- An energy efficiency evaluation must be performed to guarantee the absence of thermal issues and to assess system autonomy during battery-powered operation.
- Investigate strategies to improve robustness of the pipeline with weak reactions misclassifications.
- Conducting an ablation study to quantify the independent contribution of each pipeline component will provide insights on how to improve the overall pipeline.

Author Contributions: Conceptualization, B.S., L.D., A.P. and V.C.; methodology, B.S., L.D., A.P. and V.C.; software, B.S.; validation, B.S., L.D., A.P. and V.C.; formal analysis, B.S., L.D., A.P. and V.C.; investigation, B.S.; resources, B.S., L.D. and A.P.; data curation, B.S.; writing—original draft preparation, B.S.; writing—review and editing, B.S., E.A., L.D., A.P. and V.C.; visualization, B.S.; supervision, L.D., A.P. and V.C.; project administration, L.D., A.P. and V.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by National Funds and FCT/MCTES (PIDDAC), through the Foundation for Science and Technology, I.P. (FCT) under the project UID/05549/2025 (https://doi.org/10.54499/UID/05549/2025) and LASI-LA/P/0104/2020 (https://doi.org/10.54499/LA/P/0050/2020).

Data Availability Statement: The datasets presented in this article are not readily available because they are intellectual property of CRIAM, provided under a legal agreement. Requests to access the datasets should be directed to Criamknowledge, S.A.

Conflicts of Interest: Two of the authors are affiliated with CRIAM (A.F.; L.D.), the company whose prototype device was used in the present study. This affiliation is disclosed for transparency purposes. The authors confirm that this relationship did not influence the study design, data collection, data analysis, interpretation of results, manuscript preparation, or final conclusions. The authors declare no other conflicts of interest.

Abbreviations

ABO	ABO Blood Group System (A, B, O Antigens)
AI	Artificial Intelligence
ARM	Advanced RISC Machines (processor architecture)
AUC	Area Under the Curve
CNN	Convolutional Neural Network
CPU	Central Processing Unit
FP16	16-bit Floating Point (half precision)
FP32	32-bit Floating Point (single precision)
GPU	Graphics Processing Unit
HOG	Histogram of Oriented Gradients
HSV	Hue, Saturation, Value (color space)
INT8	8-bit Integer (quantization format)
KNN	K-Nearest Neighbors
ONNX	Open Neural Network Exchange
POC	Point-of-Care
QAT	Quantization-Aware Training
RAM	Random Access Memory
ReLU	Rectified Linear Unit (activation function)

ResNet	Residual Network
RhD	Rhesus D Antigen
ROC	Receiver Operating Characteristic
SLHBGA	Schematic Learning Hybrid Blood Group Algorithm
SoC	System on a Chip
SVC	Support Vector Classifier
SVM	Support Vector Machine
ViT	Vision Transformer
VM	Virtual Machine
XGBoost	Extreme Gradient Boosting

References

1. Criamknowledge, S.A. CRIAM Portable Point-of-Care Diagnostic Platform. 2026. Available online: https://www.criamknowledge.com (accessed on 10 January 2026).
2. Balaji, B.; Jeyasakthi, R.; Julius Fusic, S.; Rishwana, M.; Swathilakshmi, P.R.K.; Reshma, K.K. A novel approach of classifying ABO blood group image dataset using deep learning algorithm. In Proceedings of the 2021 International Conference on Computational Performance Evaluation (ComPE), Shillong, India, 1–3 December 2021; pp. 393–398. [CrossRef]
3. Titus, A.; Devi, K.R.; Divya, G.; Nimmagadda, P.; Shankar, S.V. A Systematic Procedure to Identify Human Blood Groups by using Image Processing Assisted Learning Principle. In Proceedings of the 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 14–15 December 2023; pp. 1–6.
4. Reddy, G.C.; SaiKrishna, G.; Micheal, A.A. Automated Blood Group Detection System Using Image Processing for Non- Invasive Antigen Feature Extraction and Classification. In Proceedings of the 2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 5–7 February 2025; pp. 2199–2208.
5. Larpant, N.; Niamsi, W.; Noiphung, J.; Chanakiat, W.; Sakuldamrongpanich, T.; Kittichai, V.; Tongloy, T.; Chuwongin, S.; Boonsang, S.; Laiwattanapaisal, W. Simultaneous phenotyping of five Rh red blood cell antigens on a paper-based analytical device combined with deep learning for rapid and accurate interpretation. *Anal. Chim. Acta* **2022**, *1207*, 339807. [CrossRef] [PubMed]
6. Ding, S.; Duan, S.; Chen, Y.; Xie, J.; Tian, J.; Li, Y.; Wang, H. Centrifugal microfluidic platform with digital image analysis for parallel red cell antigen typing. *Talanta* **2023**, *252*, 123856. [CrossRef] [PubMed]
7. Silva, B.; Abilheira, E.; Dukanovic, L.; Pinheiro, A.; Carvalho, V. Blood Type Classification Using DINOv2 Embeddings and Logistic Regression: A Preliminary Study with Quantization Analysis. In Proceedings of the 10th International Young Engineers Forum on Electrical and Computer Engineering (YEF-ECE 2026), Lisbon, Portugal, 19 June 2026.
8. Lapierre, Y.; Rigal, D.; Adam, J.; Josef, D.; Meyer, F.; Greber, S.; Drot, C. The gel test: A new way to detect red cell anti-gen-antibody reactions. *Transfusion* **1990**, *30*, 109–113. [CrossRef] [PubMed]
9. Malomgre, W.; Neumeister, B. Recent and future trends in blood group typing. *Anal. Bioanal. Chem.* **2009**, *393*, 1443–1451. [PubMed]
10. Mujahid, A.; Dickert, F.L. Blood group typing: From classical strategies to the application of synthetic antibodies generated by molecular imprinting. *Sensors* **2015**, *16*, 51. [CrossRef] [PubMed]
11. Li, H.-Y.; Guo, K. Blood group testing. *Front. Med.* **2022**, *9*, 827619. [CrossRef] [PubMed]
12. Kim, D.S.; Lee, S.H.; Ahn, C.H.; Lee, J.Y.; Kwon, T.H. Disposable integrated microfluidic biochip for blood typing by plastic microinjection moulding. *Lab A Chip* **2006**, *6*, 794–802. [CrossRef] [PubMed]
13. Al-Tamimi, M.; Shen, W.; Zeineddine, R.; Tran, H.; Garnier, G. Validation of paper-based assay for rapid blood typing. *Anal. Chem.* **2012**, *84*, 1661–1668. [CrossRef] [PubMed]
14. Noiphung, J.; Talalak, K.; Hongwarittorrn, I.; Pupinyo, N.; Thirabowonkitphithan, P.; Laiwattanapaisal, W. A novel paper-based assay for the simultaneous determination of Rh typing and forward and reverse ABO blood groups. *Biosens. Bioelectron.* **2015**, *67*, 485–489. [CrossRef] [PubMed]
15. Songjaroen, T.; Laiwattanapaisal, W. Simultaneous forward and reverse ABO blood group typing using a paper-based device and barcode-like interpretation. *Anal. Chim. Acta* **2016**, *921*, 67–76. [CrossRef] [PubMed]
16. Huet, M.; Cubizolles, M.; Buhot, A. Red blood cell agglutination for blood typing within passive microfluidic biochips. *High-Throughput* **2018**, *7*, 10. [CrossRef] [PubMed]
17. Chomean, S.; Ingkananth, S.; Kiatchaipar, M.; Kaset, C. Portable paper-based device for ABO and RhD typing using smartphone interpretation. *Anal. Chim. Acta* **2021**, *1180*, 338884. [CrossRef] [PubMed]

18. Sklavounos, A.A.; Lamanna, J.; Modi, D.; Gupta, S.; Mariakakis, A.; Callum, J.; Wheeler, A.R. Digital microfluidic hemagglutination assays for blood typing, donor compatibility testing, and hematocrit analysis. *Clin. Chem.* **2021**, *67*, 1699–1708. [CrossRef] [PubMed]
19. Ardakani, F.; Shojaeifard, Z.; Hemmateenejad, B. Point-of-need determination of blood typing using a three-dimensional origami microfluidic paper-based analytical device. *Microchem. J.* **2022**, *181*, 107796. [CrossRef]
20. Odeh, N.; Toma, A.; Mohammed, F.; Dama, Y.; Oshaibi, F.; Shaar, M. An efficient system for automatic blood type determination based on image matching techniques. *Appl. Sci.* **2021**, *11*, 5225. [CrossRef]
21. Korchagin, S.; Zaychenkova, E.; Ershov, E.; Pishchev, P.; Vengerov, Y. Image-based second opinion for blood typing. *Health Inf. Sci. Syst.* **2024**, *12*, 28. [CrossRef] [PubMed]
22. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
23. Torrey, L.; Shavlik, J. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*; Klinger, K., Snavely, J., Brehm, M., Eds.; IGI Global: Hershey, PA, USA, 2009; pp. 242–264.
24. Cai, Z.; Peng, C. A study on training fine-tuning of Convolutional Neural Networks. In Proceedings of the 2021 13th International Conference on Knowledge and Smart Technology (KST), Chonburi, Thailand, 21–24 January 2021; pp. 84–89. [CrossRef]
25. Peng, C.J.; Lee, K.; Ingersoll, G.M. An introduction to logistic regression analysis and reporting. *J. Educ. Res.* **2002**, *96*, 14–23. [CrossRef]
26. Evgeniou, T.; Pontil, M. Support Vector Machines: Theory and applications. In *Lecture Notes in Computer Science*; Goos, G., Hartmanis, J., Eds.; Springer: Berlin/Heidelberg, Germany, 1978; pp. 249–257. [CrossRef]
27. Rokach, L.; Maimon, O. Decision Trees. In *Springer eBooks*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 165–192. [CrossRef] [PubMed]
28. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A. DINOv2: Learning Robust Visual Features without Supervision. *arXiv* **2023**, arXiv:2304.07193.
29. Hastie, T.J.; Tibshirani, R.; Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer Series in Statistics; Springer: Berlin/Heidelberg, Germany, 2009.
30. Silva, B.; Abilheira, E.; Dukanovic, L.; Pinheiro, A.; Carvalho, V. Blood typing with computer vision and machine learning: A preliminary approach. In Proceedings of the 19th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2026), Marbella, Spain, 2–4 March 2026.
31. Bai, J.; Lu, F.; Zhang, K. ONNX: Open Neural Network Exchange. GitHub Repository. Available online: https://github.com/onnx/onnx (accessed on 10 January 2026).
32. Cohen, J.F.; Korevaar, D.A.; Altman, D.G.; Bruns, D.E.; Gatsonis, C.A.; Hooft, L.; Irwig, L.; Levine, D.; Reitsma, J.B.; de Vet, H.C.W.; et al. STARD 2015 guidelines for reporting diagnostic accuracy studies: Explanation and elaboration. *BMJ Open* **2016**, *6*, e012799. [CrossRef] [PubMed]
33. Sounderajah, V.; Guni, A.; Liu, X.; Collins, G.S.; Karthikesalingam, A.; Markar, S.R.; Golub, R.M.; Denniston, A.K.; Shetty, S.; Moher, D.; et al. The STARD-AI reporting guideline for diagnostic accuracy studies using artificial intelligence. *Nat. Med.* **2025**, *31*, 3283–3289. [CrossRef] [PubMed]
34. Collins, G.S.; Moons, K.G.; Dhiman, P.; Riley, R.D.; Beam, A.L.; Van Calster, B.; Ghassemi, M.; Liu, X.; Reitsma, J.B.; Van Smeden, M.; et al. TRIPOD + AI statement: Updated guidance for reporting clinical prediction models that use regression or machine learning methods. *BMJ* **2024**, *385*, e078378. [CrossRef] [PubMed]
35. Moons, K.G.M.; Damen, J.A.; Kaul, T.; Hooft, L.; Navarro, C.A.; Dhiman, P.; Beam, A.L.; Van Calster, B.; Celi, L.A.; Denaxas, S. PROBAST + AI: A risk-of-bias and applicability assessment tool for prediction model studies using artificial intelligence. *BMJ* **2025**, *24*, 388. [CrossRef] [PubMed]
36. European Parliament and Council. *Regulation (EU) 2017/746 of 5 April 2017 on In Vitro Diagnostic Medical Devices*; Official Journal of the European Union 2017; European Parliament and Council: Brussels, Belgium, 2017.
37. Medical Device Coordination Group. *MDCG 2022-2: Guidance on General Principles of Clinical Evidence for In Vitro Diagnostic Medical Devices (IVDs)*; European Commission: Brussels, Belgium, 2022.
38. *ISO 20916:2019*; In Vitro Diagnostic Medical Devices—Clinical Performance Studies Using Specimens from Human Sub-jects—Good Study Practice. International Organization for Standardization: Geneva, Switzerland, 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.