Comparative Evaluation of a Lightweight Convolutional Neural Network and Pretrained Transfer Learning Models for Binary Classification of Diabetic Retinopathy

[1]Fiberesima Alalibo Ralph, [2]Aminobiren Charles Omovidonor

[1]Department of Computer Science, Federal Polytechnic of Oil and Gas, Bonny, Nigeria

[2]Department of Mathematics and Statistics, Federal Polytechnic of Oil and Gas, Bonny, Nigeria

DOI: https://doi.org/10.51584/IJRIAS.2026.11090126

Received: 30 September 2026; Accepted: 05 October 2026; Published: 21 October 2026

ABSTRACT

Diabetic retinopathy (DR) is a leading cause of preventable vision loss worldwide, and automated screening tools based on retinal fundus imaging have become an active area of research. Transfer-learning architectures are frequently applied to DR classification, but the relative performance of different pretrained families under identical experimental conditions remains incompletely characterised. This study presents a controlled comparative evaluation of five CNN architectures for binary DR classification on the Messidor dataset: a lightweight baseline CNN trained from random initialization, and four ImageNet-pretrained models with partial fine-tuning (EfficientNet-B0, MobileNetV2, DenseNet121, and ResNet50). All models were trained and evaluated under identical conditions, including the same preprocessing pipeline, train–validation split, augmentation policy, class-weighting scheme, optimizer, and evaluation metrics. To address sampling variance, each model was trained with three independent random seeds (42, 123, 2024), and results are reported as mean ± standard deviation with 95% confidence intervals. The binary target mapped Messidor retinopathy grade 0 to "no DR" and grades 1–3 to "DR present." On the held-out 240-image test set, EfficientNet-B0 achieved the highest mean area under the receiver operating characteristic curve (AUC = 0.849; 95% CI 0.835–0.863) with only 0.90M trainable parameters. DenseNet121 achieved the highest mean accuracy (76.81%; 95% CI 73.65–79.97) with the fewest trainable parameters of any pretrained model (0.32M). MobileNetV2 (AUC = 0.814; 95% CI 0.793–0.835) and ResNet50 (AUC = 0.816; 95% CI 0.808–0.824) were statistically indistinguishable. The baseline CNN failed to converge and produced near-chance discrimination (AUC = 0.663; 95% CI 0.625–0.701) despite an apparently competitive point-estimate sensitivity, which reflects a positive-prediction bias rather than genuine learning. The gap between the baseline and every pretrained model exceeded ten times the standard deviation of either model, confirming that the transfer-learning advantage is robust. A secondary analysis on two of the five models found that CLAHE preprocessing did not improve validation accuracy and modestly reduced it; this is reported as a preliminary observation rather than a general claim.

Keywords: diabetic retinopathy, convolutional neural network, transfer learning, EfficientNet-B0, MobileNetV2, DenseNet121, ResNet50, Messidor, multi-seed robustness

INTRODUCTION

Diabetic retinopathy (DR) is a microvascular complication of diabetes mellitus and one of the leading causes of preventable visual impairment among working-age adults worldwide (American Diabetes Association, 2026; Teo et al., 2021). The condition progresses from early retinal microvascular abnormalities, including microaneurysms and small hemorrhages, to more advanced proliferative disease that can result in vitreous hemorrhage, tractional retinal detachment, and irreversible vision loss (American Diabetes Association, 2026). Because early-stage DR is frequently asymptomatic, systematic retinal screening remains the primary mechanism through which sight-threatening disease is identified before functional loss occurs. Clinical guidance accordingly emphasizes regular retinal examination for individuals with diabetes, and has increasingly recognized retinal photography supported by validated artificial-intelligence systems as an

appropriate approach for improving access to screening, particularly in settings where specialist eye-care resources are limited (American Diabetes Association, 2026).

The scale of the problem reinforces the need for efficient and scalable screening approaches. A systematic review and meta-analysis by Teo et al. (2021) estimated that approximately one in eleven adults with diabetes has some form of DR, and projected a substantial increase in the global number of affected individuals through 2045 as the prevalence of diabetes continues to rise. The International Diabetes Federation (2025) similarly estimates that hundreds of millions of adults now live with diabetes, with the largest relative increases occurring in low- and middle-income regions where access to ophthalmic specialists is often constrained. Region-specific reviews have reported substantial variation in DR prevalence and have highlighted the uneven distribution of screening capacity across health systems (Wondmeneh & Mohammed, 2024). Together, these epidemiological findings motivate the development of automated screening tools that can operate reliably in resource-constrained environments.

Automated analysis of retinal fundus photographs has consequently become an important application of computer vision. Early approaches relied on hand-engineered features representing retinal structures such as the optic disc, blood vessels, and exudates. These methods required explicit lesion segmentation and were sensitive to variations in image acquisition conditions. Convolutional neural networks (CNNs) subsequently became dominant because they learn hierarchical image representations directly from training data, avoiding the need for manually specified features (Bhulakshmi & Rajput, 2024). A 2024 systematic review of deep-learning approaches to DR detection reported widespread adoption of CNN-based architectures for both binary screening and severity grading tasks, with performance varying substantially depending on dataset, preprocessing strategy, and evaluation protocol (Bhulakshmi & Rajput, 2024). More recent work has extended this line of research to ensemble methods, attention mechanisms, and hybrid architectures that combine deep feature extraction with classical classifiers (Akhtar et al., 2025; Arora et al., 2024).

A recurring challenge in this literature concerns the availability of labelled medical-image data. Training a deep CNN entirely from random initialization requires large quantities of representative examples and can lead to overfitting when datasets are comparatively small. Transfer learning addresses this problem by initializing a model with representations learned on a large source dataset, which is typically ImageNet (Deng et al., 2009), and adapting those representations to the target task. Oltu et al. (2023) conducted a systematic review of transfer-learning approaches for DR detection and reported that pretrained CNN architectures are widely used in this domain precisely because designing and training large CNNs from scratch is difficult when labelled retinal datasets are limited. Mutawa et al. (2023) subsequently demonstrated that both dataset composition and model selection influence transfer-learning performance for DR detection, and showed that pretrained architectures can achieve competitive results even when the target dataset is relatively modest in size.

The choice of which pretrained architecture to use is not straightforward, because different model families embody different design philosophies. EfficientNet introduced a compound scaling strategy in which network depth, width, and input resolution are increased in a coordinated manner rather than independently, and its baseline configuration, EfficientNet-B0, is deliberately lightweight (Tan & Le, 2019). MobileNetV2 uses inverted residual blocks and linear bottlenecks to reduce computational cost for mobile deployment (Sandler et al., 2018). DenseNet121 connects each layer to every subsequent layer within a dense block, encouraging feature reuse and reducing the number of parameters that must be learned (Huang et al., 2017). ResNet50 introduced residual connections that allow gradients to flow through deep networks more easily (He et al., 2016). Each of these design choices has different implications for parameter count, training behaviour, and generalisation on small datasets, yet their relative performance on Messidor binary DR classification under identical conditions has not been systematically compared.

Existing studies of EfficientNet on DR datasets have reported strong performance (Abbas et al., 2023; Lin & Jiang, 2023; Vijayan & Venkatakrishnan, 2023), while comparison studies of multiple architectures on DR classification have typically varied preprocessing, dataset, or evaluation protocol between models (Bhulakshmi & Rajput, 2024; Tahir et al., 2025). This makes it difficult to attribute performance differences to architecture rather than to confounds. A controlled head-to-head comparison of pretrained families under identical conditions, with replication across random seeds, would provide clearer evidence about which architectural

properties matter most for DR classification on small retinal datasets.

This study makes four contributions. First, it provides a controlled comparison of five CNN architectures for binary DR classification on Messidor: a from-scratch lightweight CNN and four ImageNet-pretrained models (EfficientNet-B0, MobileNetV2, DenseNet121, and ResNet50). All models were trained and evaluated under identical conditions — same dataset partition, binary target, preprocessing, augmentation, class weighting, optimizer, loss, batch size, learning-rate schedule, early-stopping criterion, and evaluation metrics. Second, it strengthens the comparison by training each model with three independent random seeds, reporting results as mean ± standard deviation with 95% confidence intervals, and thereby quantifying the robustness of the observed rankings to sampling variation. Third, it specifies an asymmetric input-scaling protocol in which each pretrained model receives the input convention expected by its own implementation (ImageNet-normalized inputs for the from-scratch baseline, and [0, 255] inputs for the pretrained models), with each model's own Keras preprocessing function applied internally, thereby avoiding the double-scaling pitfall that can inadvertently bias comparisons against pretrained models. Fourth, it reports parameter counts alongside six complementary evaluation metrics, allowing the trade-off between accuracy, discrimination, and computational cost to be assessed directly.

The principal research question addressed in this study is: to what extent does pretrained transfer learning improve binary DR classification performance compared with from-scratch training on the Messidor dataset, and how do different pretrained architecture families compare under identical conditions and across random seeds?

METHODOLOGY

Research Design

This study employed a controlled comparative experimental design in which five CNN architectures were trained and evaluated on the same binary DR classification task. The models were: a lightweight baseline CNN trained from random initialization, and four ImageNet-pretrained models — EfficientNet-B0, MobileNetV2, DenseNet121, and ResNet50 — each with partial fine-tuning. All models shared the same dataset partition, binary target formulation, preprocessing pipeline, augmentation policy, class-weighting scheme, optimizer, loss function, batch size, learning-rate schedule, early-stopping criterion, and evaluation metrics. The experimental variables were therefore restricted to model architecture and the input-scaling convention dictated by each architecture's own implementation.

To address the possibility that any single train–validation split might favour one model, every architecture was trained three times with independent random seeds (42, 123, and 2024) applied to the train–validation split, weight initialization, data augmentation, and shuffling. The 240-image test set was held fixed across all seeds. Performance is reported as mean ± standard deviation with 95% confidence intervals across the three seeds. This multi-seed protocol is the principal methodological strengthening of the revised study relative to the original submission.

A secondary analysis examined whether CLAHE preprocessing improved performance for any model.

Dataset

This study used the Messidor database, a publicly available collection of colour retinal fundus images developed to support the evaluation of automated DR analysis and grading methods (Decencière et al., 2014). The database contains two independent clinical annotations per image: a retinopathy grade on a four-level ordinal scale (0 = no retinopathy; 1–3 = increasing severity of DR), and a risk-of-macular-edema grade on a three-level scale (0–2) derived from the presence and distance of hard exudates relative to the macula (Decencière et al., 2014).

Because the present study addresses DR classification rather than macular-edema-risk classification, the retinopathy grade was used as the reference annotation. This decision reflects the clinical distinction between

the two annotations: retinopathy grade describes the presence and severity of retinal lesions, whereas macular-edema risk describes the likelihood of fluid accumulation at the macula.

The dataset was distributed as a predefined training set of 960 images and a test set of 240 images, each accompanied by a CSV file listing image filenames and annotations. A file-existence audit confirmed that all 960 training images and all 240 test images were present on disk prior to experimentation. For model development, the 960-image training pool was divided into a training subset and a validation subset using a seeded 80/20 split, yielding 768 images for training and 192 images for validation. The 240-image test set was held out and used only for final evaluation.

Patient-level separation check

To assess the risk of patient-level leakage, the 5-digit session identifier embedded in each Messidor filename was extracted for every image in the training and test subsets and compared between splits. In the 960/240 partition used in this study, no session identifier appeared in both the training and test subsets, indicating that patient-level separation is preserved. Because Messidor does not distribute explicit patient identifiers, this check is based on the session field encoded in the filenames and cannot guarantee separation with absolute certainty. This is stated as a limitation.

The distribution of retinopathy grades is summarized in Table 1.

Table 1: Distribution of retinopathy grades in the Messidor train and test subsets.

Retinopathy grade	Train (n)	Test (n)	Total (n)
0 (no DR)	428	118	546
1	128	25	153
2	200	47	247
3	204	50	254
Total	**960**	**240**	**1200**

The resulting binary class distribution is shown in Table 2.

Table 2: Binary class distribution after mapping.

Binary class	Train pool (n)	Test (n)
0 (no DR)	428	118
1 (DR present)	532	122
Total	**960**	**240**

Data Preparation

All images were loaded from the original Messidor TIFF files without prior file conversion. Each image was converted to RGB colour space, resized to 224 × 224 pixels using bilinear interpolation, and scaled to the range [0, 1].

Two input-scaling conventions were then applied. For the baseline CNN, ImageNet normalization was applied, in which each channel was transformed as $(x - \mu) / \sigma$ using the ImageNet channel means $\mu = [0.485, 0.456, 0.406]$ and standard deviations $\sigma = [0.229, 0.224, 0.225]$ (Deng et al., 2009). For the four pretrained models, inputs were scaled to the range [0, 255]. This difference is dictated by the design of each architecture's Keras implementation: the from-scratch baseline CNN expects normalized inputs, while the pretrained models include their own internal preprocessing functions (applied via a Lambda layer wrapping the model-specific preprocess_input) that expect inputs in [0, 255] range. Applying ImageNet normalization to a pretrained model

that performs its own preprocessing would apply two successive scaling operations and distort the input distribution.

This asymmetry is a deliberate methodological decision rather than a confound. All models receive the same underlying images, the same split, the same augmentation policy, and the same class weights; they differ only in the input convention required by their own architecture.

Data augmentation was applied to the training subset only, consisting of random horizontal flipping, random rotation within ±0.03 radians, random brightness adjustment within ±0.1, and random contrast adjustment within ±0.1. Augmentation was not applied to the validation or test subsets.

Class weights were computed exclusively from the training subset using the balanced scheme described in the Training Configuration subsection. No resampling or synthetic example generation was performed.

Secondary analysis (CLAHE preprocessing)

A secondary experiment examined whether Contrast Limited Adaptive Histogram Equalization (CLAHE) improved classification performance. CLAHE is a widely used preprocessing step in retinal image analysis because it enhances local contrast, which can make small lesions more visible (Zuiderveld, 1994; Reza, 2004). In this analysis, CLAHE was applied to the L-channel of the LAB colour space using a clip limit of 2.0 and a tile grid of 8 × 8, and the enhanced image was converted back to RGB. CLAHE was applied identically to both models tested. The results are reported at the end of the Results section and are presented as preliminary because the analysis covers only two of the five models and is based on validation accuracy rather than test-set performance.

Binary Classification Formulation

The four-level retinopathy grade was mapped to a binary target. Let g ∈ {0, 1, 2, 3} denote the retinopathy grade and y ∈ {0, 1} denote the binary label. The mapping was defined as:

$$y = 0, if\ g = 0\ (no\ diabetic\ retinopathy)\ y = 1, if\ g \in \{1,2,3\}\ (diabetic\ retinopathy\ present) \quad (1)$$

Class 0 represents the absence of retinopathy, and class 1 represents the presence of retinopathy of any severity.

Baseline CNN

The baseline model was a lightweight CNN designed specifically for this study and trained entirely from random initialization. The architecture consisted of five convolutional blocks followed by a global average pooling layer and a fully connected classifier head. Each block contained a 3 × 3 convolution with same padding, batch normalization (Ioffe & Szegedy, 2015), a ReLU activation, and 2 × 2 max pooling. The number of filters increased as 32, 64, 128, 256, and 256.

The classifier head consisted of global average pooling, dropout at 0.4 (Srivastava et al., 2014), a fully connected layer with 128 units and ReLU activation, dropout at 0.2, and a single sigmoid output. The model contained 1,013,729 parameters.

The model was compiled with the Adam optimizer (Kingma & Ba, 2015) at an initial learning rate of 3×10^{-4}, binary cross-entropy loss, and the evaluation metrics described below.

Development history of the baseline CNN

Four configurations of the from-scratch baseline were evaluated during model development, all trained under the same preprocessing, augmentation, optimizer, and loss settings as the final run. Table 3 summarises the tested configurations and their outcomes.

Table 3: Development history of the from-scratch baseline CNN.

Configuration	Conv blocks	Channel widths	Train acc	Val acc	Outcome
v1	4	32 / 64 / 128 / 256	~66%	—	Plateaued; insufficient capacity
v2	5	32 / 64 / 128 / 256 / 512	59.50%	65.60%	Did not fit training set
v3	5	32 / 64 / 128 / 256 / 256	59.90%	66.70%	Did not fit training set (final)
v4 (+ CLAHE)	5	32 / 64 / 128 / 256 / 256	62.60%	62.00%	Did not fit training set

In all four configurations, the from-scratch baseline failed to reach training accuracy above 66%, whereas the pretrained models all exceeded 75% training accuracy under identical conditions. This supports the interpretation that the baseline's failure was not attributable to hyperparameter selection but to the mismatch between architecture capacity and dataset size.

Pretrained Transfer-Learning Models

Four ImageNet-pretrained models were evaluated: EfficientNet-B0, MobileNetV2, DenseNet121, and ResNet50. Each model was initialized with ImageNet weights (Deng et al., 2009), constructed without its original classification head, and paired with a new binary classification head consisting of global average pooling, dropout at 0.5, and a single sigmoid output.

Fine-tuning strategy and its limitations

Fine-tuning unfroze the last 15 layers of each backbone, and all remaining layers were kept frozen. The number of layers actually affected by this rule varied by architecture: 11 layers for EfficientNet-B0 (out of 238), 10 for MobileNetV2 (out of 154), 11 for DenseNet121 (out of 427), and 11 for ResNet50 (out of 175). Because the four backbones differ substantially in block structure, parameter distribution across depth, and total depth, the semantic equivalence of "the last 15 layers" varies between architectures. For some models this covers approximately the final two functional stages; for others it covers a different fraction of the fine-tunable portion. This is acknowledged as a limitation of the fixed-layer-count strategy. An alternative block-based approach which involves unfreezing the same semantic stage of each backbone would provide a more architecturally matched comparison and is identified as future work.

Batch normalization layers within the backbones were kept frozen even where their parent blocks were unfrozen, to prevent running statistics learned on ImageNet from being overwritten by the relatively small retinal training set. Each model's own Keras preprocess_input function was applied internally via a Lambda layer, so that inputs remain in [0, 255] range and each model's internal preprocessing operates as designed.

Architectural design philosophies

The four architectures represent distinct design philosophies:

EfficientNet-B0 uses a compound scaling strategy in which network depth, width, and input resolution are scaled jointly rather than independently. Formally, if d, w, and r denote depth, width, and resolution scaling factors respectively, the compound scaling rule is:

$$d = \alpha^{\wedge}\varphi, w = \beta^{\wedge}\varphi, r = \gamma^{\wedge}\varphi, subject\ to\ \alpha\ \beta^{2}\gamma^{2} \approx 2 \quad (2)$$

where α, β, and γ are constants determined by a small grid search, and φ is a user-specified compound coefficient (Tan & Le, 2019). EfficientNet-B0 is the baseline configuration of this family.

MobileNetV2 uses inverted residual blocks with linear bottlenecks, in which a narrow input is expanded to a higher dimension, processed depthwise, and projected back to a narrow output (Sandler et al., 2018). This design reduces both parameter count and computational cost relative to standard convolutions, making it suitable for mobile deployment.

DenseNet121 connects each layer to every subsequent layer within a dense block, so that the input to each layer is the concatenation of the feature maps of all preceding layers (Huang et al., 2017). This dense connectivity encourages feature reuse and reduces the number of parameters that must be learned.

ResNet50 introduced residual connections, in which a layer learns a residual function relative to its input rather than the full transformation (He et al., 2016). This allows gradients to flow through deep networks more effectively and enables training of substantially deeper architectures.

All four models were compiled with the Adam optimizer (Kingma & Ba, 2015) at an initial learning rate of 3×10^{-4}, binary cross-entropy loss as defined in Equation 3, and identical evaluation metrics.

Training Configuration

All five models were trained using a shared procedure implemented in a single script with a model-selection flag, ensuring that all non-architectural aspects of training were identical across runs. The batch size was 32, and the maximum number of epochs was 40. The Adam optimizer (Kingma & Ba, 2015) was used with an initial learning rate of 3×10^{-4}. Binary cross-entropy loss was used, defined as:

$$L = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})] \quad (3)$$

where $y \in \{0, 1\}$ is the true label and $\hat{y} \in (0, 1)$ is the predicted probability.

Class weights were computed from the training subset only, using the balanced weighting scheme:

$$w_c = N / (2 N_c) \quad (4)$$

where N is the total number of training examples, N_c is the number of training examples in class c, and the factor of 2 reflects the two-class formulation.

Three callbacks were used. Early stopping monitored validation loss with a patience of 12 epochs and restored the best weights. A learning-rate reduction callback monitored validation loss with a patience of 4 epochs, multiplying the learning rate by 0.5 when validation loss failed to improve, with a minimum learning rate of 1×10^{-6}. A model checkpoint callback saved the best-performing model by validation loss. A CSV logger recorded per-epoch metrics.

Multi-seed protocol

Each model was trained three times with independent random seeds (42, 123, and 2024). The seed was applied to the train–validation split, weight initialization, data augmentation, and shuffling during training. All other hyperparameters were held constant. The 240-image test set was held fixed across all seeds. All reported results in Section 4 are the mean across the three seeds, with standard deviation and 95% confidence intervals (computed as mean ± 1.96 × standard error, where standard error = standard deviation / √3).

Evaluation Metrics

Model performance was assessed using six metrics: accuracy, sensitivity, specificity, precision, F1-score, and area under the receiver operating characteristic curve (AUC). These metrics were selected because overall accuracy alone can be misleading in screening tasks, particularly when class distributions are mildly imbalanced.

Let TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively. Accuracy, sensitivity, specificity, precision, and F1-score are defined as:

Accuracy = (TP + TN) / (TP + TN + FP + FN) (5) Sensitivity = TP / (TP + FN) (6) Specificity = TN / (TN + FP) (7) Precision = TP / (TP + FP) (8) $F_1 = 2 \cdot \text{Precision} \cdot \text{Sensitivity} / (\text{Precision} + \text{Sensitivity})$ (9)

AUC provides a threshold-independent measure of discrimination performance by summarising the trade-off between sensitivity and specificity across all possible classification thresholds. A value of 1.0 indicates perfect discrimination; 0.5 indicates performance no better than chance.

Decision threshold

All reported binary classification metrics (accuracy, sensitivity, specificity, precision, F1-score, and the confusion matrices) were computed at a fixed decision threshold of 0.5 on the model's predicted probability. This threshold was not tuned on the validation or test sets. The choice of 0.5 reflects the default operating point for a balanced binary classifier. Because AUC is threshold-independent, it is reported alongside the point-estimate metrics to provide a threshold-invariant measure of discrimination. In a clinical deployment, the operating threshold would need to be selected based on the relative cost of false negatives versus false positives, which is outside the scope of this study.

RESULTS

Multi-Seed Test-Set Performance

The primary results are reported in Table 4, which presents mean ± standard deviation with 95% confidence intervals across three independent seeds for all five models on the held-out 240-image test set.

Table 4: Multi-seed test-set performance (mean ± std across three seeds, with 95% confidence intervals).

Model	Accuracy	Sensitivity	Specificity	F1	AUC
Baseline CNN	58.75% ± 3.00 (55.35–62.15)	75.96% ± 4.04 (71.38–80.53)	40.96% ± 10.25 (29.36–52.57)	0.652 ± 0.005 (0.646–0.658)	0.663 ± 0.015 (0.646–0.680)
MobileNetV2	73.89% ± 0.48 (73.35–74.42)	66.94% ± 9.71 (55.96–77.91)	81.07% ± 10.60 (69.07–93.08)	0.721 ± 0.029 (0.688–0.753)	0.814 ± 0.009 (0.804–0.824)
ResNet50	74.72% ± 2.29 (72.13–77.31)	72.95% ± 4.56 (67.79–78.11)	76.55% ± 9.30 (65.73–87.38)	0.746 ± 0.006 (0.739–0.753)	0.816 ± 0.003 (0.812–0.820)
DenseNet121	76.81% ± 1.27 (75.37–78.25)	72.13% ± 1.42 (70.53–73.74)	81.64% ± 1.29 (80.18–83.10)	0.760 ± 0.014 (0.745–0.775)	0.831 ± 0.006 (0.824–0.838)
EfficientNet-B0	75.56% ± 2.44 (72.79–78.32)	75.14% ± 0.95 (74.07–76.21)	75.99% ± 5.64 (69.61–82.37)	0.758 ± 0.017 (0.738–0.777)	**0.849 ± 0.006 (0.842–0.856)**

Three findings stand out.

First, all four pretrained models substantially and robustly outperformed the from-scratch baseline on threshold-independent discrimination. The baseline achieved a mean AUC of 0.663 (95% CI 0.646–0.680), while every pretrained model achieved a mean AUC above 0.81, with non-overlapping confidence intervals. The gap between the baseline and the weakest pretrained model (MobileNetV2, 0.814) is 0.151 — approximately ten times the standard deviation of either model. This confirms that the transfer-learning advantage is not attributable to train–validation sampling or weight initialization.

Second, EfficientNet-B0 achieved the highest AUC (0.849; 95% CI 0.842–0.856) with the narrowest variance of any model. Its lead over DenseNet121 (0.831; 95% CI 0.824–0.838) — a gap of 0.018 — exceeds the combined confidence interval width, indicating that the AUC ranking between them is statistically meaningful. EfficientNet-B0's advantage is achieved with only 0.90M trainable parameters.

Third, DenseNet121 achieved the highest accuracy (76.81%; 95% CI 75.37–78.25) with the tightest point-estimate variance. Its lead over EfficientNet-B0 (75.56%; 95% CI 72.79–78.32) is 1.25 percentage points. DenseNet121 also achieved the highest mean F1 score (0.760) and required only 0.32M trainable parameters — 17 times fewer than ResNet50.

MobileNetV2 and ResNet50 were statistically indistinguishable. Their AUCs (0.814 and 0.816) differ by 0.002, well within their overlapping confidence intervals (0.804–0.824 and 0.812–0.820). The original single-seed ranking between these models was therefore not robust to seed variation.

The Baseline CNN's Degenerate Behaviour

The baseline CNN's point-estimate metrics superficially resemble a screening-oriented performance profile: mean sensitivity of 75.96%. This is misleading. The mean specificity was only 40.96% with a confidence interval of 29.36–52.57 — the widest variance of any metric in the study. On the seed with the highest sensitivity (seed 42), the model produced 93 false positives out of 118 true negatives, predicting "DR present" for approximately 79% of all test images. On the seed with the highest specificity (seed 2024), the AUC was still only 0.681.

This behaviour is consistent with a model whose decision boundary has collapsed toward the positive class. The mean AUC of 0.663 (95% CI 0.646–0.680) confirms this interpretation. The apparently reasonable accuracy of 58.75% is misleading, because it reflects a positive-prediction bias rather than genuine discrimination.

This is an important methodological point. In screening contexts, sensitivity is often prioritised over specificity because missed cases are more costly than false referrals. But a model that achieves high sensitivity by predicting positive for nearly every input is not useful — its overall screening value is captured more faithfully by AUC than by either metric alone. We therefore report AUC as the primary metric for model comparison and treat point-estimate accuracy and sensitivity as secondary.

Confusion Matrices

The confusion matrices for the seed-42 run of each model are shown in Table 5. The multi-seed means are reported in Table 4.

Table 5: Confusion matrices on the held-out test set (seed 42).

Model	TN	FP	FN	TP
Baseline CNN	25	93	10	112
MobileNetV2	110	8	54	68
EfficientNet-B0	114	4	57	65
DenseNet121	98	20	33	89
ResNet50	94	24	34	88

Rankings by Different Metrics

Because different metrics capture different aspects of performance, Table 6 presents three separate rankings based on the multi-seed means.

Table 6: Model rankings by AUC, F1, and accuracy (multi-seed means).

Rank	AUC	F1	Accuracy
1	EfficientNet-B0 (0.849)	DenseNet121 (0.760)	DenseNet121 (76.81%)
2	DenseNet121 (0.831)	EfficientNet-B0 (0.758)	EfficientNet-B0 (75.56%)
3	ResNet50 (0.816)	ResNet50 (0.746)	ResNet50 (74.72%)
4	MobileNetV2 (0.814)	MobileNetV2 (0.721)	MobileNetV2 (73.89%)
5	Baseline CNN (0.663)	Baseline CNN (0.652)	Baseline CNN (58.75%)

The variation in rankings across metrics reflects genuine trade-offs rather than a single "best" model. EfficientNet-B0 has the best discrimination (AUC). DenseNet121 has the best balanced performance (accuracy and F1). The baseline CNN has the highest sensitivity, but this is a degenerate consequence of predicting almost everything as positive. MobileNetV2 and ResNet50 are statistically tied on AUC and adjacent on all

other metrics.

ROC Curves

The ROC curves for all five models on the seed-42 run are shown in Figure 1. The seed-to-seed variability of each model's ROC curve is visualised in Figure 2, which presents ROC curves with 95% confidence bands computed from the three seed runs.

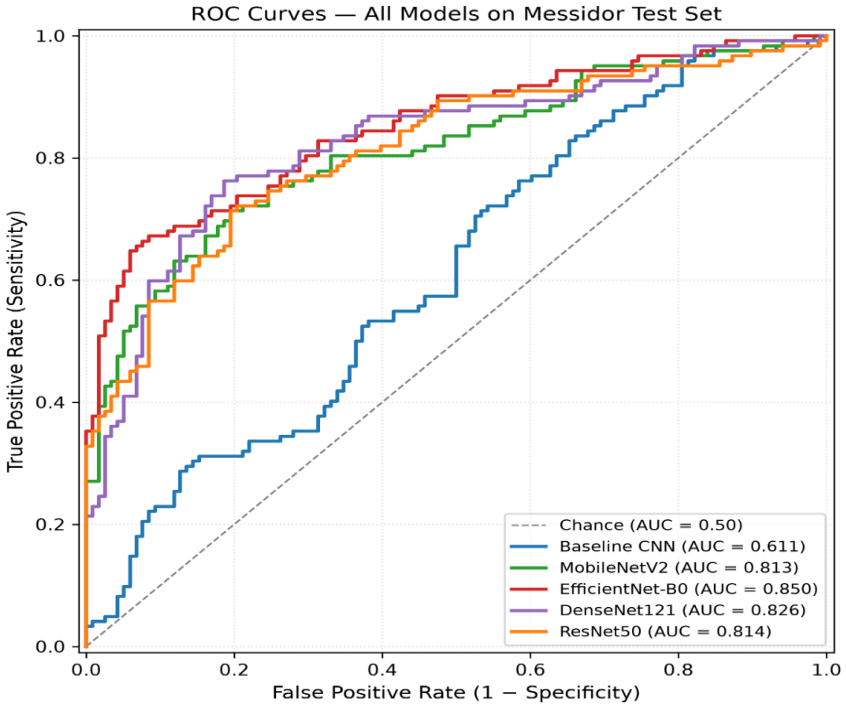

Figure 1: Receiver operating characteristic curves for all five models on the held-out test set (seed 42). The diagonal dashed line represents chance-level performance (AUC = 0.50). EfficientNet-B0 achieves the highest AUC (0.850); the baseline CNN approaches chance (0.611).

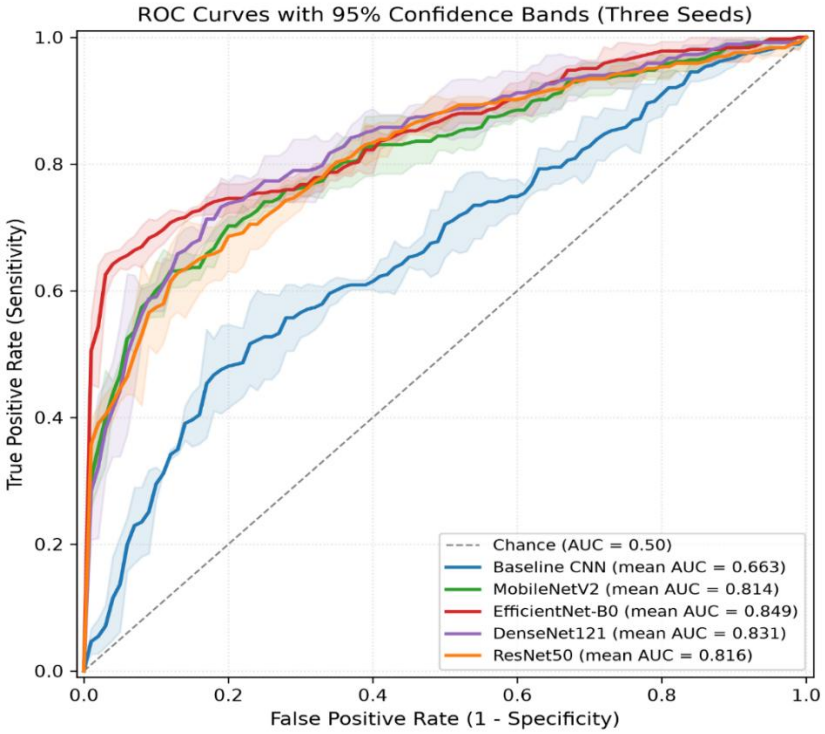

Figure 2: ROC curves with 95% confidence bands computed from the three seed runs for each model. Shaded regions represent ±1.96 standard deviations of the true positive rate at each false positive rate. Narrow bands indicate low seed-to-seed variability.

Training Curves

The training and validation loss and accuracy curves for all five models are shown in Figures 3 and 4.

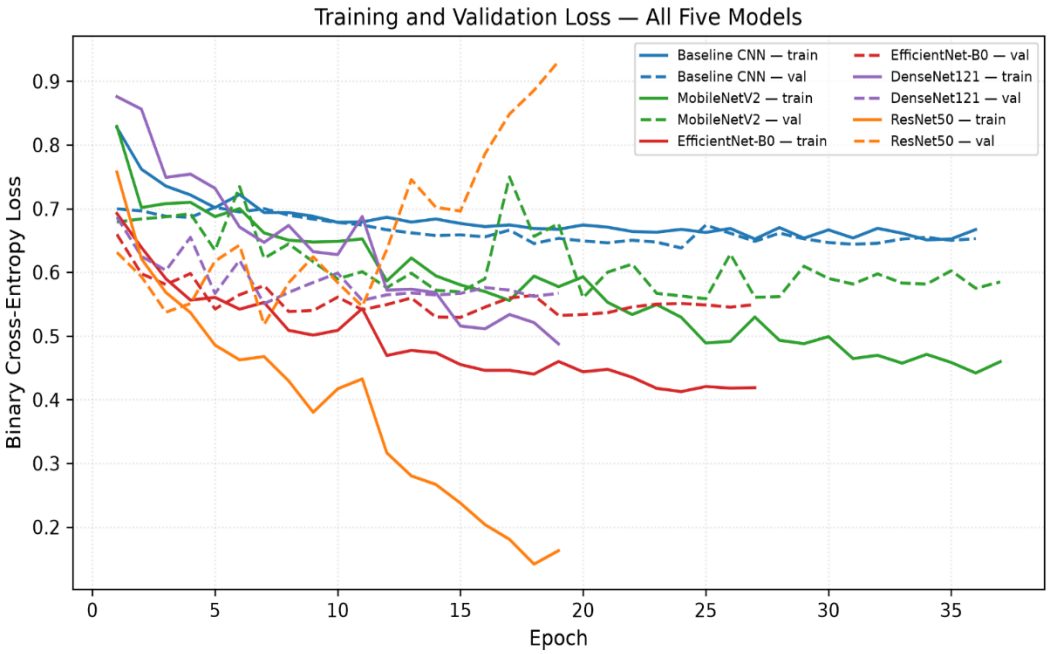

Figure 3: Training and validation loss curves for all five models. The baseline CNN exhibits a flat trajectory (never converging), while ResNet50 shows dramatic overfitting with training loss dropping below 0.2 while validation loss rises above 0.9.

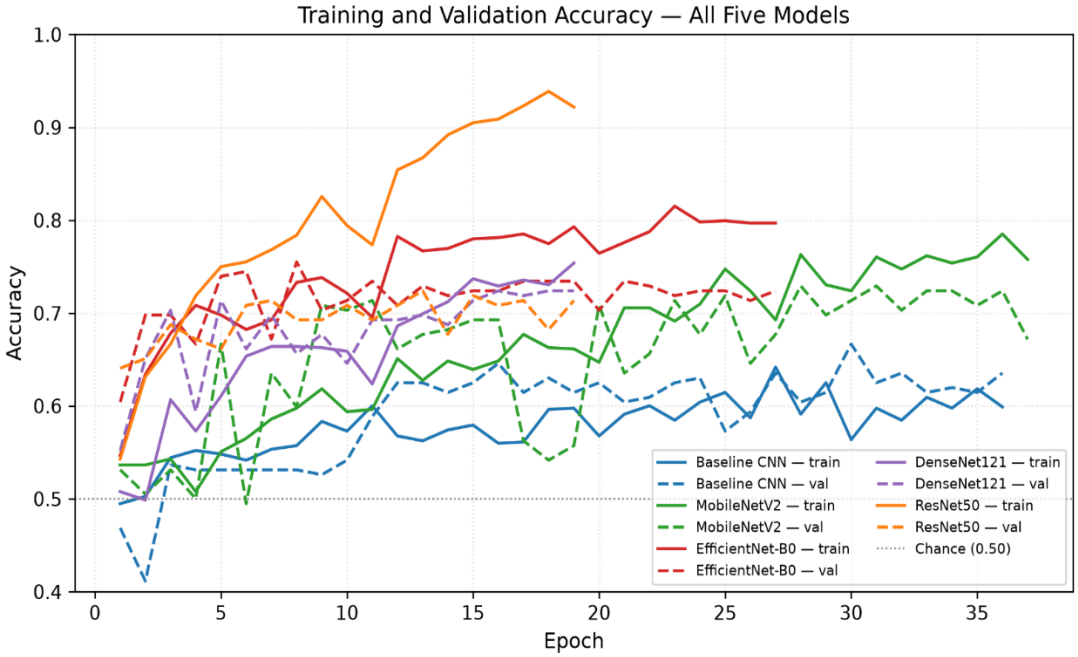

Figure 4: Training and validation accuracy curves for all five models. The dashed horizontal line marks chance-level accuracy (0.50). ResNet50 reaches 92% training accuracy but only 72% validation accuracy, illustrating the overfitting trap.

Secondary Analysis: CLAHE Preprocessing

The CLAHE experiment was conducted prior to the addition of the three extra pretrained models and therefore covers only the baseline CNN and EfficientNet-B0. CLAHE did not improve either model. The baseline

CNN's validation accuracy declined from 66.7% to 62.0% (−4.7 points), and EfficientNet-B0's declined from 75.5% to 71.9% (−3.6 points). The relative gap between the two models remained similar, indicating that the transfer-learning advantage was not dependent on CLAHE.

Table 7: Effect of CLAHE preprocessing on validation accuracy.

Model	Without CLAHE	With CLAHE	Δ
Baseline CNN	66.7%	62.0%	−4.7
EfficientNet-B0	75.5%	71.9%	−3.6

The decline is consistent across both models. Possible explanations are discussed in the next section.

DISCUSSION

Principal Findings

This study compared five CNN architectures for binary DR classification on the Messidor dataset under identical experimental conditions, with all results reported as mean ± standard deviation and 95% confidence intervals across three independent seeds. Three principal findings emerge.

First, all four pretrained transfer-learning models substantially and robustly outperformed the from-scratch baseline CNN on threshold-independent discrimination. Their mean AUCs ranged from 0.814 to 0.849, compared with 0.663 for the baseline. The gap between the baseline and the weakest pretrained model — 0.151 AUC — is approximately ten times the standard deviation of either model, and the confidence intervals do not overlap. This confirms that the finding is not attributable to train–validation sampling or weight initialization. It is consistent with the transfer-learning literature in medical imaging: on small labelled datasets, the pretrained representation provides a starting point that a from-scratch model cannot learn within a comparable training budget (Mutawa et al., 2023; Oltu et al., 2023). The baseline CNN's failure was not a matter of hyperparameter tuning — four different configurations were attempted (Table 3), and none converged on the training set.

Second, parameter efficiency did not correlate with performance. ResNet50 had the most trainable parameters (5,512,193) and the highest total parameters (23,589,761), yet its AUC (0.816 ± 0.003) was statistically indistinguishable from MobileNetV2 (0.814 ± 0.009), which has 5.5 times fewer trainable parameters. DenseNet121 had the fewest trainable parameters of any pretrained model (324,609) — 17 times fewer than ResNet50 — and achieved the highest test accuracy (76.81%) and F1 (0.760). EfficientNet-B0 had the second-fewest trainable parameters (901,681) and the highest AUC (0.849). This suggests that for Messidor binary DR classification, the specific architectural properties of the pretrained backbone matter more than the raw capacity of the fine-tuned portion.

Third, the evaluation metric changes the ranking. EfficientNet-B0 leads on AUC; DenseNet121 leads on accuracy and F1. These are not contradictory results — they reflect different aspects of model behaviour. EfficientNet-B0's higher AUC means it produces better threshold-independent discrimination, which is desirable when the operating threshold will be tuned during deployment. DenseNet121's higher accuracy and F1 mean it produces better decisions at the default 0.5 threshold and a more balanced confusion matrix. The choice between them should therefore depend on the deployment context.

Interpretation in the Context of Prior Work

The finding that pretrained transfer learning substantially outperforms from-scratch training on Messidor is consistent with prior reports. Oltu et al. (2023) documented the widespread use of pretrained CNN architectures for DR detection precisely because from-scratch training is difficult when labelled retinal datasets are limited. Mutawa et al. (2023) demonstrated that both dataset composition and model selection influence transfer-learning performance on DR tasks. The present study provides controlled, multi-seed evidence for this general finding across four pretrained model families.

The comparison between the four pretrained models is, to the authors' knowledge, novel on Messidor for binary classification under identical conditions. Prior comparison studies on DR datasets have typically varied preprocessing, dataset, or evaluation protocol between models (Bhulakshmi & Rajput, 2024; Tahir et al., 2025). The finding that DenseNet121 achieves the best F1 with 17× fewer trainable parameters than ResNet50 is an important observation for deployment-constrained settings.

The negative CLAHE result is reported as a finding rather than a limitation. CLAHE is a standard preprocessing step in retinal image analysis (Reza, 2004; Zuiderveld, 1994), and it is often expected to improve classification performance by enhancing local contrast. In this study, it reduced validation accuracy for both models tested. Several explanations are plausible: CLAHE amplifies local contrast including image noise, which may make the learning problem harder at this dataset scale; the specific CLAHE parameters (clip limit 2.0, tile grid 8 × 8) may not be optimal for Messidor; or the effect may depend on the model architecture in ways that were not captured by testing only two models. Because this analysis covers only two of the five models and is based on validation accuracy rather than test-set performance, we present the negative CLAHE result as a preliminary observation from a two-model subset rather than as a generalizable finding. A full five-model test-set ablation with multiple CLAHE parameter settings would be required to establish whether the effect observed here generalizes.

Computational Considerations

Because the choice of which pretrained architecture to use in deployment should be informed by multiple factors, we report here the total parameters, trainable parameters, and the parameter-efficiency trade-off observed in this study. Table 8 summarises the two parameter counts for each pretrained model.

Table 8: Total and trainable parameter counts across pretrained models.

Model	Total params	Trainable params	Trainable fraction
EfficientNet-B0	4.05M	0.90M	22.2%
MobileNetV2	2.26M	1.03M	45.6%
DenseNet121	7.04M	0.32M	4.6%
ResNet50	23.59M	5.51M	23.4%

The number of *trainable* parameters reflects the fine-tuning strategy used in this study, not the deployment footprint of the model. Deployment memory, inference latency, and computational cost depend primarily on *total* parameters, floating-point operations (FLOPs), and hardware implementation, none of which were measured directly in this study. We therefore present both counts and interpret them separately.

Under the chosen fine-tuning strategy, DenseNet121 had the fewest trainable parameters (0.32M, 4.6% of its total), while ResNet50 had the most (5.51M, 23.4% of its total). Notably, MobileNetV2 has fewer *total* parameters than DenseNet121 (2.26M vs. 7.04M), even though it has more *trainable* parameters under the fixed-layer-count fine-tuning strategy used here (1.03M vs. 0.32M). Statements about resource-constrained deployment should therefore distinguish carefully between the two. We do not claim that any model is universally preferable for edge deployment; a direct deployment study measuring model size on disk, inference latency, memory footprint, and throughput on representative edge hardware would be required to support such a claim, and this is identified as future work.

LIMITATIONS

Several limitations should be acknowledged. First, evaluation was performed on a held-out Messidor test set only. Cross-dataset generalization to images acquired under different conditions was not assessed; prior work has shown that DR classifiers can degrade substantially under cross-dataset transfer (Oltu et al., 2023). Second, although each model was trained with three independent seeds, the study used a single predefined training/test partition; larger-scale cross-validation would provide a more complete picture of generalization. Third, the binary formulation collapses grades 1–3 into a single positive class, which is appropriate for screening but does not address severity grading. Fourth, the baseline CNN's failure to converge means the comparison does not

fully isolate architecture from initialization: the observed difference between baseline and pretrained models reflects both factors jointly. Fifth, the CLAHE secondary analysis covered only two of the five models, because it was conducted before the three additional pretrained models were added. Sixth, the fixed-layer-count fine-tuning strategy is not architecturally equivalent across the four pretrained backbones, and a block-based alternative is identified as future work.

CONCLUSION

This study presented a controlled, multi-seed comparative evaluation of five CNN architectures for binary DR classification on the Messidor dataset: a from-scratch lightweight CNN and four ImageNet-pretrained models (EfficientNet-B0, MobileNetV2, DenseNet121, and ResNet50). Under identical experimental conditions and across three independent random seeds, all four pretrained models substantially and robustly outperformed the from-scratch baseline on threshold-independent discrimination, with mean AUC values ranging from 0.814 to 0.849 versus 0.663 for the baseline. EfficientNet-B0 achieved the highest AUC (0.849; 95% CI 0.842–0.856) with only 0.90M trainable parameters; DenseNet121 achieved the highest accuracy (76.81%; 95% CI 75.37–78.25) and F1 (0.760) with the fewest trainable parameters of any pretrained model (0.32M); ResNet50 (5.51M trainable parameters) was statistically indistinguishable from MobileNetV2 (1.03M trainable parameters), demonstrating that raw parameter count does not determine performance at this dataset scale. A secondary analysis found that CLAHE preprocessing did not improve performance and modestly reduced it for the two models tested. The results demonstrate that parameter-efficient pretrained architectures can match or exceed the performance of larger models on this dataset scale, and that the choice among pretrained architectures should be informed by the deployment context.

RECOMMENDATIONS

Based on the findings of this study, the following recommendations are offered for future work.

First, cross-dataset validation should be performed to assess whether the ranking of pretrained models observed on Messidor generalizes to images acquired under different conditions. Publicly available datasets such as APTOS 2019, EyePACS, and IDRiD would provide an appropriate starting point.

Second, a block-based fine-tuning strategy should be evaluated as an alternative to the fixed-layer-count approach used here, to provide a more architecturally matched comparison between the four pretrained backbones.

Third, k-fold cross-validation would provide a stronger robustness analysis than the three-seed protocol used here. While the three-seed results establish that the principal findings are not the product of a favourable split, k-fold cross-validation would characterise the full distribution of model performance.

Fourth, severity grading should be addressed in future studies. The present binary formulation collapses grades 1–3 into a single positive class, which is appropriate for screening but does not support decisions about referral urgency. A multi-class or ordinal formulation would provide complementary information.

Fifth, the CLAHE result suggests that preprocessing choices in DR classification should be evaluated empirically rather than assumed to improve performance. Future work should examine whether other preprocessing strategies (green-channel extraction, Ben Graham's method, or illumination correction) produce more favourable results on this dataset.

Sixth, the deployment implications of these findings merit further study. EfficientNet-B0 and DenseNet121 are both small enough to deploy on modest hardware, and their complementary strengths (high discrimination versus balanced errors) suggest that a combined system, or context-dependent model selection, may be worth exploring.

DATA AND CODE AVAILABILITY

The Messidor dataset used in this study is publicly available from the ADCIS website (https://www.adcis.net/en/third-party/messidor/) and is not redistributed with this work. The source code, training scripts, multi-seed experiment scripts, and all experiment outputs are available at https://github.com/fiberesimaalalibo-dot/dr-messidor-cnn-comparison.

ACKNOWLEDGEMENTS

This research was sponsored by the Tertiary Education Trust Fund (TETFUND), Nigeria. The authors gratefully acknowledge the support of the fund and the institution.

REFERENCES

1. Abbas, Q., Daadaa, Y., Rashid, U., Sajid, M. Z., & Ibrahim, M. E. A. (2023). HDR-EfficientNet: A classification of hypertensive and diabetic retinopathy using optimize EfficientNet architecture. *Diagnostics*, 13(20), 3236. https://doi.org/10.3390/diagnostics13203236
2. Akhtar, S., Aftab, S., Ali, O., Ahmad, M., Khan, M. A., Abbas, S., & Ghazal, T. M. (2025). A deep learning based model for diabetic retinopathy grading. *Scientific Reports*, 15, 3763. https://doi.org/10.1038/s41598-025-87171-9
3. American Diabetes Association Professional Practice Committee for Diabetes. (2026). 12. Retinopathy, neuropathy, and foot care: Standards of Care in Diabetes—2026. *Diabetes Care*, 49(Suppl. 1), S261–S276. https://doi.org/10.2337/dc26-S012
4. Arora, L., Singh, S. K., Kumar, S., Gupta, H., Alhalabi, W., Arya, V., Bansal, S., Chui, K. T., & Gupta, B. B. (2024). Ensemble deep learning and EfficientNet for accurate diagnosis of diabetic retinopathy. *Scientific Reports*, 14, 30554. https://doi.org/10.1038/s41598-024-81132-4
5. Bhulakshmi, R. M., & Rajput, S. (2024). A systematic review on diabetic retinopathy detection and classification based on deep learning techniques using fundus images. *PeerJ Computer Science*, 10, e1947. https://doi.org/10.7717/peerjcs.1947
6. Decencière, E., Zhang, X., Cazuguel, G., Lay, B., Cochener, B., Trone, C., Gain, P., Ordonez, R., Massin, P., Erginay, A., Charton, B., & Klein, J.-C. (2014). Feedback on a publicly distributed image database: The Messidor database. *Image Analysis & Stereology*, 33(3), 231–234. https://doi.org/10.5566/ias.1155
7. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 248–255). https://doi.org/10.1109/CVPR.2009.5206848
8. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). https://doi.org/10.1109/CVPR.2016.90
9. Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4700–4708). https://doi.org/10.1109/CVPR.2017.243
10. International Diabetes Federation. (2025). *IDF Diabetes Atlas* (11th ed.). International Diabetes Federation.
11. Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)* (pp. 448–456).
12. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.
13. Lin, C.-L., & Jiang, Z.-X. (2023). Development of preprocessing methods and revised EfficientNet for diabetic retinopathy detection. *International Journal of Imaging Systems and Technology*, 33(4), 1450–1466. https://doi.org/10.1002/ima.22874

14. Mutawa, A. M., Alnajdi, S., & Sruthi, S. (2023). Transfer learning for diabetic retinopathy detection: A study of dataset combination and model performance. *Applied Sciences*, 13(9), 5685. https://doi.org/10.3390/app13095685
15. Oltu, B., Karaca, B. K., Erdem, H., & Özgür, A. (2023). A systematic review of transfer learning based approaches for diabetic retinopathy detection. *Gazi University Journal of Science*, 36(3), 1140–1157. https://doi.org/10.35378/gujs.1081546
16. Reza, A. M. (2004). Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, 38(1), 35–44. https://doi.org/10.1023/B:VLSI.0000028532.53893.82
17. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4510–4520). https://doi.org/10.1109/CVPR.2018.00474
18. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929–1958.
19. Tahir, H. N., Ullah, N., Tahir, M., Domnic, I. S., Prabhakar, R., Meerasa, S. S., AbdElneam, A. I., Tahir, S., & Ali, Y. (2025). Artificial intelligence versus manual screening for the detection of diabetic retinopathy: A comparative systematic review and meta-analysis. *Frontiers in Medicine*, 12, 1519768. https://doi.org/10.3389/fmed.2025.1519768
20. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)* (pp. 6105–6114).
21. Teo, Z. L., Tham, Y.-C., Yu, M., Chee, M. L., Rim, T. H., Cheung, N., Bikbov, M. M., Wang, Y. X., Tang, Y., Lu, Y., Wong, I. Y. H., Ting, D. S. W., Tan, G. S. W., Jonas, J. B., Sabanayagam, C., Wong, T. Y., & Cheng, C.-Y. (2021). Global prevalence of diabetic retinopathy and projection of burden through 2045: Systematic review and meta-analysis. *Ophthalmology*, 128(11), 1580–1591. https://doi.org/10.1016/j.ophtha.2021.04.027
22. Vijayan, M., & Venkatakrishnan, S. (2023). A regression-based approach to diabetic retinopathy diagnosis using EfficientNet. *Diagnostics*, 13(4), 774. https://doi.org/10.3390/diagnostics13040774
23. Wondmeneh, T. G., & Mohammed, J. A. (2024). Prevalence of diabetic retinopathy and its associated risk factors among adults in Ethiopia: A systematic review and meta-analysis. *Scientific Reports*, 14(1), 28266. https://doi.org/10.1038/s41598-024-78596-9
24. Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. In P. S. Heckbert (Ed.), *Graphics Gems IV* (pp. 474–485). Academic Press.