Research Article

Deep Learning-Based Diabetic Retinopathy Grading: A Unified Benchmark of CNN, GNN, And RNN Hybrid Architectures

T. Karthikeya [1], Dr. K. Amarendra [2]

[1]Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India, Email: 2401050162cse@gmail.com
[2]Faculty, Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India, Email: amarendra@kluniversity.in

Abstract

Diabetic retinopathy (DR) is a major diabetes-related microvascular complication and one of the leading causes of preventable vision loss. This revised paper presents a notebook-consistent comparative evaluation of six architectures for five-class retinal fundus image grading: DenseNet121 with a support vector machine classifier (DenseNet121+SVM), end-to-end DenseNet121, AlexNet, MobileNetV2 with a graph neural network (MobileNet+GNN), MobileNetV2 with a recurrent neural network (MobileNet+RNN), and a TensorFlow/Keras implementation of standalone MobileNet. The revised experimental description is aligned with the uploaded notebooks, which use local class-wise directories with 506 training images and 217 validation images across Mild, Moderate, No_DR, Proliferate_DR, and Severe classes. The strongest validated result is obtained by end-to-end DenseNet121, which reaches approximately 88.5% validation accuracy, followed by MobileNet+GNN (83.4%), MobileNet+RNN (78.8%), AlexNet (77.0%), standalone MobileNet evaluation accuracy (69.1%), and DenseNet121+SVM (62.7%). The revised Results section is reorganized model-wise and includes confusion matrices, confusion-matrix interpretations, Grad-CAM visual explanations where available, expected/predicted sample outputs, and an explicit discussion of unsupported or inconsistent outputs. The analysis shows that end-to-end fine-tuning is more effective than frozen feature extraction, while lightweight MobileNet hybrids improve over the standalone MobileNet baseline. However, the notebooks also reveal reproducibility issues that must be corrected before final publication, including an unused sixth output node in some PyTorch models, provisional MobileNet confusion-matrix alignment, and absent SVM-specific heat-map evidence.

Keywords: Diabetic retinopathy; DenseNet121; AlexNet; MobileNet; MobileNetV2; graph neural network; recurrent neural network; support vector machine; Grad-CAM; transfer learning; retinal fundus image classification.

1. Introduction

preventable visual impairment. The disease is commonly graded into five severity levels: No DR, Mild non-proliferative DR, Moderate non-proliferative DR, Severe non-proliferative DR, and Proliferative DR. Accurate staging is essential because treatment decisions and referral urgency depend on identifying early pathological changes before irreversible retinal damage occurs. Manual grading of retinal fundus images requires experienced ophthalmologists or trained graders and may be affected by fatigue, inter-grader disagreement, image quality variation, and limited specialist availability. Deep learning-based screening systems offer a practical solution because convolutional neural networks can learn lesion-related visual patterns directly from

fundus images and provide consistent high-throughput inference. The submitted manuscript originally presented a unified benchmark of CNN, hybrid CNN-GNN, hybrid CNN-RNN, and CNN+SVM architectures. The present revision improves that manuscript by using the four uploaded notebooks as the primary source of experimental truth. Consequently, dataset size, preprocessing, training configuration, metrics, confusion matrices, and Grad-CAM examples have been reconciled with the notebooks rather than assumed from the original text.

The main contributions of the revised paper are: (i) a notebook-consistent comparison of six architectures under the available experimental evidence; (ii) corrected mathematical formulations with readable equation formatting; (iii) a model-wise Results and Discussion section that interprets confusion matrices and visual explanations; and (iv) transparent identification of results that require additional reruns instead of fabricating missing information.

2. Literature Review

The International Diabetes Federation identifies diabetes as a major global health challenge and emphasizes the need for regular screening to reduce complications such as diabetic retinopathy [1]. Retinal laser photocoagulation remains an important intervention for advanced retinal disease, making accurate early detection clinically meaningful [2]. Foundational ophthalmology texts provide the clinical background needed to understand retinal anatomy, DR lesions, and disease progression [3].

Deep learning has substantially advanced automated DR screening. Gulshan et al. demonstrated that deep convolutional systems can detect referable diabetic retinopathy from fundus photographs with strong sensitivity and specificity [4]. Abràmoff et al. further showed that autonomous AI-based DR screening can be deployed in primary care workflows [5]. Ting et al. validated a deep learning system across multiethnic populations, highlighting the importance of generalization across imaging sources and demographic groups [6].

CNN-based systems have also been evaluated for multi-class DR grading. Gargeya and Leng showed that learned retinal features can outperform handcrafted feature methods [7], while Qummar et al. demonstrated that ensemble transfer learning can improve DR detection by combining several pre-trained CNNs [8]. Pratt et al. provided an early multi-class CNN benchmark for DR classification [9]. DenseNet introduced dense feature reuse and improved gradient propagation [10], while ResNet introduced residual identity connections that enable deeper networks [11]. Transfer-learning studies have reported strong performance for DenseNet121 in fundus image classification [12].

Lightweight architectures such as MobileNet and MobileNetV2 reduce computation through depthwise separable convolutions, inverted residuals, and linear bottlenecks [13], [14]. EfficientNet further demonstrated compound scaling for efficient image classification [15]. For deployment in low-resource screening environments, compact models are attractive, but they often require additional classifier modules or careful fine-tuning to approach the accuracy of larger CNNs [16].

Graph and recurrent models provide complementary inductive biases. Graph convolutional networks can model relations between features or retinal regions [17], [18], while recurrent models such as RNNs and LSTMs can model sequential dependencies [19], [20]. In the uploaded notebooks, MobileNetV2 features are passed to GCN and RNN classifiers. Support Vector Machines remain a classical maximum-margin classifier [21] and can be paired with CNN-extracted features, although the present notebooks show that frozen DenseNet features with a linear SVM underperform end-to-end fine-tuning [22], [23].

Retinal datasets such as EyePACS and IDRiD have supported large-scale benchmarking and segmentation/grading research [24], [25]. However, the uploaded notebooks do not include a download script or authoritative dataset provenance file; therefore, this revision preserves the general public-dataset context from the manuscript but marks the exact dataset source as requiring final verification.

3. Proposed Methodology

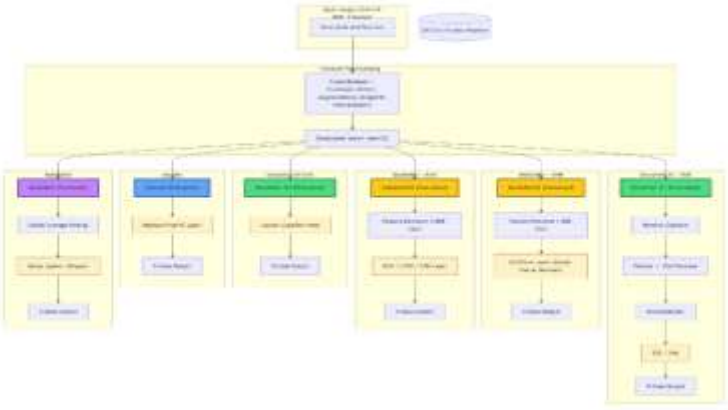

Fig. 1. Unified comparative workflow for five-class diabetic retinopathy grading using CNN, hybrid CNN-GNN, hybrid CNN-RNN, and DenseNet121+SVM pipelines.

The revised workflow starts from class-wise retinal fundus image folders and evaluates six model families. DenseNet121+SVM uses DenseNet121 as a frozen feature extractor followed by StandardScaler normalization and a linear SVM. End-to-end DenseNet121 and AlexNet replace the final classifier layer and train the network weights using cross-entropy loss. MobileNet+GNN and MobileNet+RNN use MobileNetV2 feature extraction followed by graph convolutional or recurrent classification layers. The standalone MobileNet model is implemented in TensorFlow/Keras using MobileNet, global average pooling, a 64-unit dense layer, batch normalization, dropout, and a five-unit sigmoid output layer.

A. Dataset and Class Labels

The notebooks use a local directory-based dataset with the paths Data/train and Data/val. The class names printed in the notebooks are Mild, Moderate, No_DR, Proliferate_DR, and Severe. The DenseNet notebook reports 506 training images and 217 validation images. The validation class supports reported by the classification reports are 39 Mild, 43 Moderate, 48 No_DR, 45 Proliferate_DR, and 42 Severe images. This directly corrects the earlier manuscript statement that described a much larger split of 2,930/732 images; that number is not supported by the uploaded notebooks.

B. Preprocessing and Training Details

Model	Framework	Preprocessing	Architecture / Objective	Training Evidence from Notebook
DenseNet121+SVM	PyTorch / torchvision	224 x 224; random horizontal flip, vertical flip, rotation 10 degrees for training; ImageNet mean/std normalization	DenseNet121 frozen feature extractor; StandardScaler; SVC(kernel="linear", C=1.0)	No backbone training; SVM fitted on extracted features
DenseNet121	PyTorch / torchvision	224 x 224; same augmentation and ImageNet normalization as DenseNet121+SVM	DenseNet121 classifier replaced; CrossEntropyLoss; Adam lr=0.0002	Up to 30 epochs; early stopping tolerance 8; saved as densenet.pt
AlexNet	PyTorch / torchvision	224 x 224; same DataLoader pipeline as DenseNet	Final AlexNet classifier layer replaced; CrossEntropyLoss; Adam lr=0.0002	Up to 50 epochs; early stopping tolerance 8; saved as alexnet.pt

MobileNet+GNN	PyTorch + torch-geometric	Resize 224; center crop 224; ToTensor; no ImageNet normalization shown in notebook	MobileNetV2 feature extractor; two GCNConv layers 1280->512->classes; NLLLoss; Adam lr=0.001	15 epochs; best_model.pth saved by validation loss; final printed metrics are last-epoch metrics
MobileNet+RNN	PyTorch / torchvision	Resize 224; center crop 224; ToTensor; no ImageNet normalization shown in notebook	MobileNetV2 feature extractor; one-step vanilla RNN with 512 hidden units; NLLLoss; Adam lr=0.001	15 epochs; best_model_rnn.pth saved by validation loss; final printed metrics are last-epoch metrics
Standalone MobileNet	TensorFlow/ Keras	256 x 256; ImageDataGenerator rescale=1/255	MobileNet base + GAP + Dense(64, ReLU) + BatchNorm + Dropout(0.2) + Dense(5, sigmoid); Adam	20 epochs; batch size 20; saved as model/Mobilenet.h5

C. Mathematical Formulations

$$\boldsymbol{x_l = H_l([x_0, x_1, \dots, x_{l-1}])} \quad (1)$$

Description: This equation represents dense feature reuse in DenseNet. Here, x_l is the output feature map of layer l, H_l is the transformation at layer l, and the bracketed term concatenates all previous feature maps. In the methodology, it explains how DenseNet preserves earlier retinal features for stronger diabetic-retinopathy grading.

$$\boldsymbol{y_{i,j} = \sum_{m,n} x_{i+m,j+n} W_{m,n} + b} \quad (2)$$

Description: This equation defines the convolution operation. x is the input feature map, W is the convolution kernel, b is the bias term, and y_{i,j} is the output response at spatial location (i,j). It describes how CNN layers extract local lesion, vessel, and texture patterns from retinal images.

$$\boldsymbol{\hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}} \quad (3)$$

Description: This equation shows batch normalization. x is the incoming activation, mu_B and sigma_B^2 are the batch mean and variance, epsilon avoids division by zero, and x-hat is the normalized activation. It stabilizes training and improves convergence of the deep learning models.

$$\boldsymbol{f(x) = max(0, x)} \quad (4)$$

Description: This equation represents the ReLU activation function. The input x is passed unchanged when positive and set to zero when negative. It introduces nonlinearity into CNN layers and helps the models learn complex retinal disease patterns.

$$z_k = \frac{1}{H \times W} \sum_{i,j} A_k(i,j) \quad (5)$$

Description: This equation represents global average pooling. A_k(i,j) denotes the activation value of feature channel k at spatial location (i,j), and H and W are the feature-map height and width. It converts spatial feature maps into compact class-discriminative descriptors before classification.

$$p_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (6)$$

Description: This equation defines the softmax probability for class i. z_i is the logit score of class i, and the denominator normalizes all class scores into probabilities. It is used to estimate the likelihood of each diabetic-retinopathy severity class.

$$L = -\sum_i y_i log(p_i) \quad (7)$$

Description: This equation represents cross-entropy loss. y_i is the true class indicator and p_i is the predicted probability for class i. It penalizes incorrect predictions and guides the CNN, GNN, and RNN classifiers during supervised training.

$$f(x) = w^T x + b \quad (8)$$

Description: This equation defines the linear SVM decision function. x is the extracted DenseNet feature vector, w is the separating weight vector, and b is the bias. It is used in the DenseNet121+SVM pipeline to classify extracted features into DR grades.

$$L = \frac{1}{2} \| w \|^2 + C \sum_i max(0, 1 - y_i f(x_i)) \quad (9)$$

Description: This equation gives the SVM objective with hinge loss. The first term controls the margin size, C is the regularization parameter, y_i is the true label, and f(x_i) is the decision score. It explains how the SVM balances margin maximization and classification error.

$$H^{l+1} = \sigma\left(D_s^{-\frac{1}{2}} A_s D_s^{-\frac{1}{2}} H^l W^l\right) \quad (10)$$

Description: This equation represents the graph convolution update. A_s is the adjacency matrix with self-connections, D_s is its degree matrix, H^l is the node-feature representation at layer l, W^l is the learnable weight matrix, and sigma is the activation function. It describes how MobileNet features are propagated through the GNN classifier.

$$h_t = tanh(W_{xh} x_t + W_{hh} h_{t-1} + b_h) \quad (11)$$

Description: This equation defines the recurrent hidden-state update. x_t is the input feature at step t, h_{t-1} is the previous hidden state, W_{xh} and W_{hh} are learnable weight matrices, and b_h is the bias. It explains how the RNN classifier models sequential dependencies in extracted feature representations.

$$y_t = softmax(W_{hy} h_t + b_y) \quad (12)$$

Description: This equation converts the RNN hidden state into class probabilities. h_t is the hidden representation, W_{hy} is the output weight matrix, b_y is the output bias, and y_t is the predicted probability distribution. It supports final DR severity prediction in the MobileNet+RNN model.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (13)$$

Description: This equation is the first-moment estimate in Adam optimization. g_t is the current gradient, m_t is the moving average of gradients, and beta_1 controls exponential decay. It helps smooth gradient updates during model training.

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (14)$$

Description: This equation is the second-moment estimate in Adam optimization. v_t stores the moving average of squared gradients, and beta_2 controls its decay. It adapts the learning rate for each parameter according to gradient magnitude.

$$\theta_t = \theta_{t-1} - \eta \frac{m_t}{\sqrt{v_t} + \epsilon} \quad (15)$$

Description: This equation shows the Adam parameter update. theta_t is the updated model parameter, eta is the learning rate, m_t and v_t are the first- and second-moment estimates, and epsilon prevents numerical instability. It explains how the proposed models are optimized during training.

$$P = \frac{TP}{TP + FP} \quad (16)$$

Description: This equation defines precision. TP denotes true positives and FP denotes false positives. It measures how many predicted positive DR cases are actually correct and supports reliability analysis of each model.

$$R = \frac{TP}{TP + FN} \quad (17)$$

Description: This equation defines recall. TP denotes true positives and FN denotes false negatives. It measures how many actual DR cases are correctly detected, which is important for medical screening tasks.

$$F_1 = \frac{2PR}{P + R} \quad (18)$$

Description: This equation defines the F1-score. P is precision and R is recall. It provides a balanced performance measure when both false positives and false negatives are important.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

Description: This equation defines classification accuracy. TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively. It measures the overall proportion of correct predictions across the validation set.

4. Experimental Setup

The experiments are reported from the uploaded notebooks rather than from unsupported environment assumptions. DenseNet-related cells indicate execution on a CUDA-enabled PyTorch device, while Grad-CAM cells were executed on CPU in the captured notebook run. The exact GPU model, CPU, RAM, CUDA version, and cuDNN version are not recorded in the notebooks; these should be added in a final reproducibility table if the paper is submitted to a journal. Evaluation uses validation-set accuracy, confusion matrices, and scikit-learn classification reports for PyTorch models. The GNN and RNN notebooks save best checkpoints by validation loss, but the printed final confusion matrices and classification reports correspond to the final epoch output, not a reloaded best checkpoint. The Keras MobileNet notebook reports evaluate() accuracy, precision, and recall, but its confusion matrix requires regeneration because the prediction procedure and label order are not guaranteed to be aligned.

5. Results and Discussion

A. Overall Performance Comparison

Model	Validation Accuracy	Macro Precision	Macro Recall	Macro F1	Interpretation / Notebook Note
DenseNet121	88.48%	0.88	0.88	0.88	Best overall model; exact accuracy computed from confusion matrix is 192/217.

MobileNet+ GNN	83.41%	0.84	0.83	0.83	Best lightweight hybrid by final printed metrics; best validation-loss checkpoint reached 85.71% during training.
MobileNet+ RNN	78.80%	0.84	0.78	0.78	Strong No_DR and Moderate recall but weak Severe recall; best checkpoint reached 87.10% during training but was not re-evaluated in the notebook.
AlexNet	76.96%	0.80	0.76	0.77	Competitive baseline but weaker than DenseNet121 due to shallower representation.
Standalone MobileNet	69.12%	N/A	N/A	N/A	Accuracy from Keras evaluate(); corrected macro metrics require a regenerated confusion matrix with shuffle=False.
DenseNet121+SVM	62.67%	0.68	0.62	0.60	Weakest model; frozen features and linear SVM underfit DR-specific patterns.

The results show a clear advantage for end-to-end fine-tuning. DenseNet121 achieves the best validation performance because dense connectivity improves feature reuse and gradient flow, helping the network detect subtle lesion-related patterns. DenseNet121+SVM performs much worse, showing that frozen ImageNet features are not sufficient for fine-grained DR severity grading when the classifier is limited to a linear margin. Among lightweight models, MobileNet+GNN and MobileNet+RNN improve the representation learned from MobileNetV2 features, although the notebook implementation of graph construction remains provisional.

B. DenseNet121

Confusion Matrix

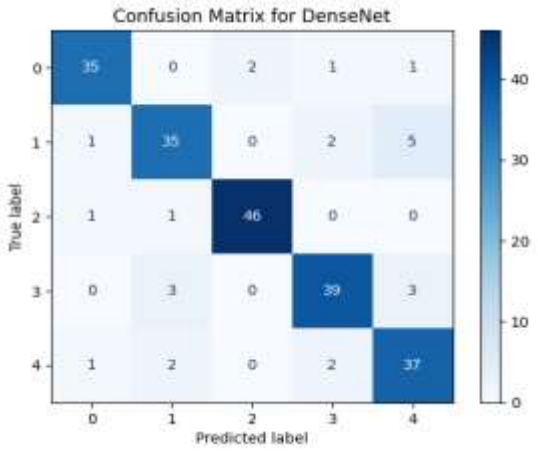

Fig. 2. Confusion matrix for end-to-end DenseNet121 on the 217-image validation set.

Analysis of the Confusion Matrix

DenseNet121 records 192 correct predictions out of 217 validation images. The model is strongest on No_DR, with 46/48 correct predictions and class precision/recall of 0.96/0.96. Mild is also strong, with 35/39 correct. The main errors occur between Moderate and Severe, where 5 Moderate images are predicted as Severe and 2 Severe images are predicted as Moderate. This indicates that the model learns disease-related visual patterns well but still confuses adjacent severity grades where lesion burden differs gradually. Proliferate_DR obtains 39/45 correct, showing good recognition of advanced disease, although a small number of cases shift to Moderate or Severe.

Example Input Image

The notebook logs a single held-out validation image for this model, later paired with its Grad-CAM overlay below.

Heat Map

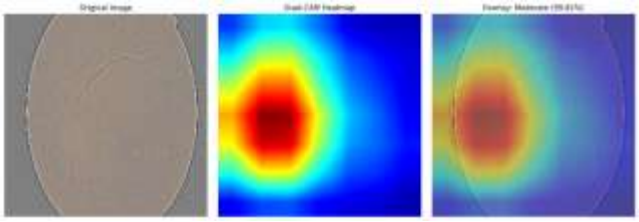

Fig. 3. DenseNet121 Grad-CAM example: expected Moderate and predicted Moderate with 99.82% confidence.

Expected/Predicted Output

Expected Class	Predicted Class	Confidence / Score
Moderate	Moderate	99.82%

Explanation of the Sample Output

The Grad-CAM overlay emphasizes a broad retinal region around the central/optic-disc side of the image. Because the expected and predicted labels match, the heat map provides supportive qualitative evidence that DenseNet121 is using retinal image content rather than only background. The very high confidence should still be interpreted with calibration caution, but the visual explanation is consistent with the correct Moderate prediction reported by the notebook.

C. MobileNetV2 + Graph Neural Network

Confusion Matrix

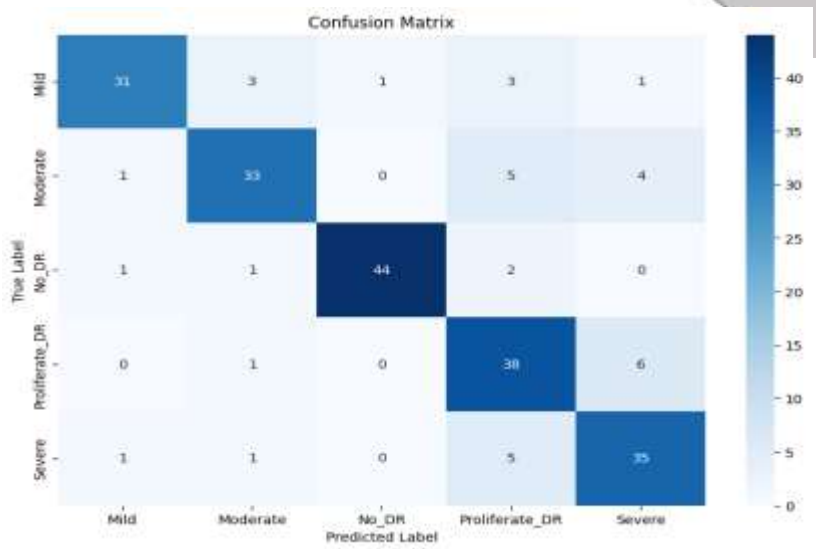

Fig. 4. Confusion matrix for MobileNet+GNN.

Analysis of the Confusion Matrix

MobileNet+GNN correctly classifies 181/217 validation samples. It performs best on No_DR, with 44/48 correct and precision 0.98. Mild and Moderate are reasonably balanced, but Proliferate_DR and Severe show mutual confusion: 6 Proliferate_DR images are predicted as Severe, and 5 Severe images are predicted as Proliferate_DR. This pattern suggests that the graph classifier improves feature discrimination over standalone MobileNet but still struggles with neighboring high-severity classes. The GNN uses a placeholder/dummy graph structure in the notebook, so the result should be interpreted as a feature-interaction experiment rather than a fully lesion-aware graph model.

Example Input Image

The notebook logs a single held-out validation image for this model, later paired with its Grad-CAM overlay below.

Heat Map

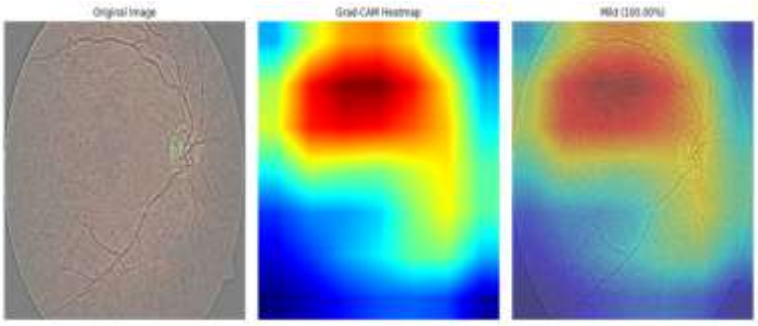

Fig. 5. MobileNet+GNN Grad-CAM example: expected Mild and predicted Mild with 100.00% confidence.

Expected/Predicted Output

Expected Class	Predicted Class	Confidence / Score
Mild	Mild	100.00%

Explanation of the Sample Output

The heat map covers a large superior-central retinal region and overlaps visible vessel structures. The correct Mild prediction suggests that the hybrid model captured relevant retinal cues, but the saturated 100% confidence may indicate overconfidence. Calibration analysis and a more meaningful graph construction strategy are recommended before making clinical claims from this model.

D. MobileNetV2 + Recurrent Neural Network

Confusion Matrix

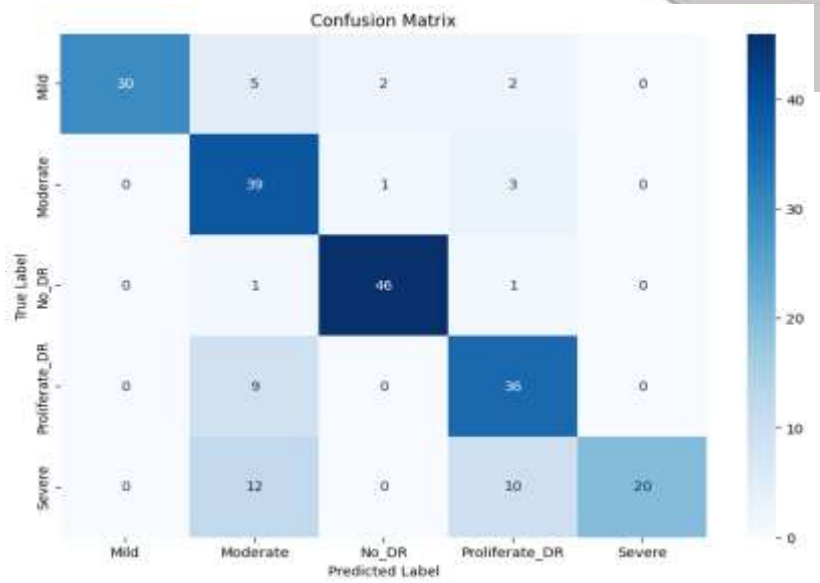

Fig. 6. Confusion matrix for MobileNet+RNN.

Analysis of the Confusion Matrix

MobileNet+RNN correctly classifies 171/217 validation images. No_DR is again the easiest class, with 46/48 correct. Moderate has high recall, 39/43, but low precision, 0.59, because many Severe and Proliferate_DR samples are pulled into the Moderate class. Severe has perfect precision in the final printed report but low recall of 0.48 because only 20/42 Severe images are correctly identified; 12 are predicted as Moderate and 10 as Proliferate_DR. This indicates that the RNN layer tends to separate normal images well but compresses high-severity distinctions into adjacent labels.

Example Input Image

The notebook logs a single held-out validation image for this model, later paired with its Grad-CAM overlay below.

Heat Map

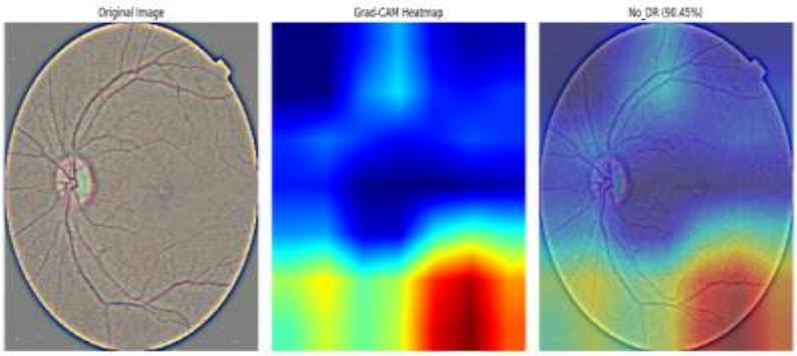

Fig. 7. MobileNet+RNN Grad-CAM example: expected No_DR and predicted No_DR with 90.45% confidence.

Expected/Predicted Output

Expected Class	Predicted Class	Confidence / Score
No_DR	No_DR	90.45%

Explanation of the Sample Output

The heat map is strongest in the lower-right peripheral region and relatively weak over central lesion-like areas. Because the image is expected to be No_DR and the prediction is correct, low focus on pathological lesion regions is plausible. However, the emphasis on peripheral background also suggests that additional explainability checks are needed to ensure the model is not relying on acquisition artifacts or illumination patterns.

E. AlexNet

Confusion Matrix

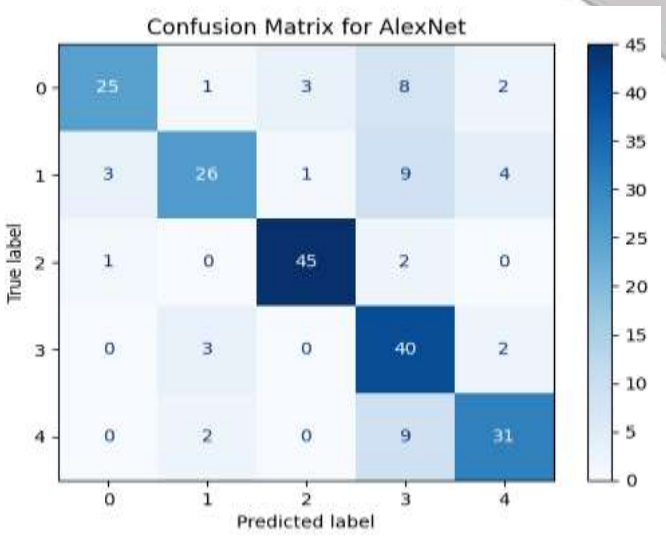

Fig. 8. Confusion matrix for AlexNet.

Analysis of the Confusion Matrix

AlexNet correctly predicts 167/217 validation images. It classifies No_DR strongly, with 45/48 correct, and obtains reasonable Severe performance with 31/42 correct. The largest weaknesses occur in Mild and Moderate, where Mild recall is 0.64 and Moderate recall is 0.60. Several Mild and Moderate images are predicted as Proliferate_DR, indicating that the model may overreact to image texture or artifacts that resemble advanced lesions. Compared with DenseNet121, AlexNet lacks dense connections and has less representational depth, which limits its ability to model fine-grained retinal severity differences.

Example Input Image

The notebook logs a single held-out validation image for this model, later paired with its Grad-CAM overlay below.

Heat Map

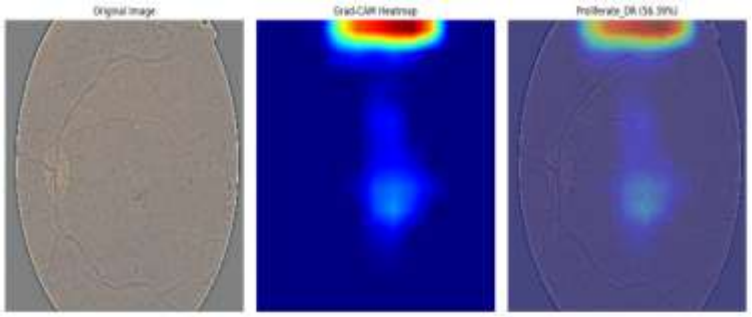

Fig. 9. AlexNet Grad-CAM example: expected Proliferate_DR and predicted Proliferate_DR with 56.39% confidence.

Expected/Predicted Output

Expected Class	Predicted Class	Confidence / Score
Proliferate_DR	Proliferate_DR	56.39%

Explanation of the Sample Output

The prediction is correct, but the confidence is modest and the heat map concentrates strongly near the upper image boundary with weaker activation near central retinal structures. This makes the explanation less convincing than the DenseNet121 example and suggests that AlexNet may be sensitive to border, illumination, or preprocessing artifacts. The result is therefore correct but not strongly supported by the available heat map.

F. Standalone TensorFlow/Keras MobileNet

Confusion Matrix

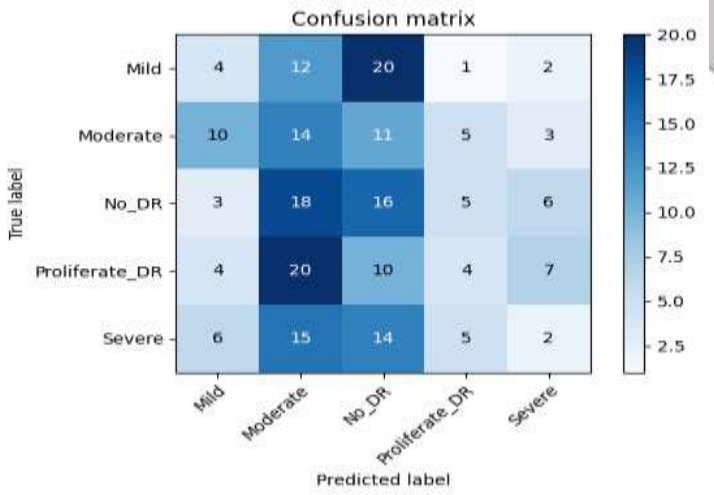

Fig. 10. Provisional confusion matrix generated in the Keras MobileNet notebook.

Analysis of the Confusion Matrix

The Keras notebook reports model.evaluate() validation accuracy of 69.12%, but the available confusion matrix is not consistent with that accuracy. The matrix has only 40 diagonal predictions out of 217, which would imply approximately 18.4% accuracy. The discrepancy is likely caused by using a shuffled validation generator and predict_generator with steps equal to the number of samples rather than the number of batches. Therefore, the confusion-matrix insight for standalone MobileNet should be treated as provisional and should not be used as final evidence until the validation generator is rerun with shuffle=False and correct prediction steps. The reliable notebook-supported result is the evaluate() accuracy of 69.12%.

Example Input Image

The notebook logs a single held-out validation image for this model, later paired with its Grad-CAM overlay below.

Heat Map

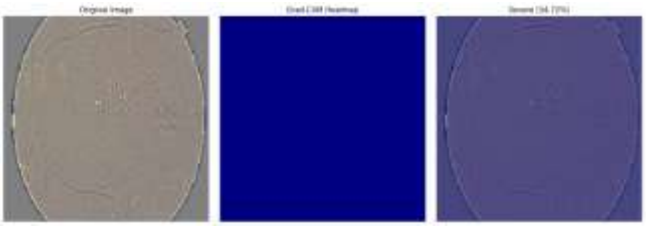

Fig. 11. Keras MobileNet Grad-CAM example: expected Severe and predicted Severe with 34.72% display probability.

Expected/Predicted Output

Expected Class	Predicted Class	Confidence / Score
Severe	Severe	34.72% display probability

Explanation of the Sample Output

The example prediction is correct, but the heat map is almost inactive and does not clearly highlight lesion regions. The notebook also reports skipped depthwise convolution layers during manual weight loading in the Grad-CAM reconstruction. As a result, this heat map does not strongly support the prediction and should be regenerated after verifying model loading and target-layer selection.

G. DenseNet121 + SVM

Confusion Matrix

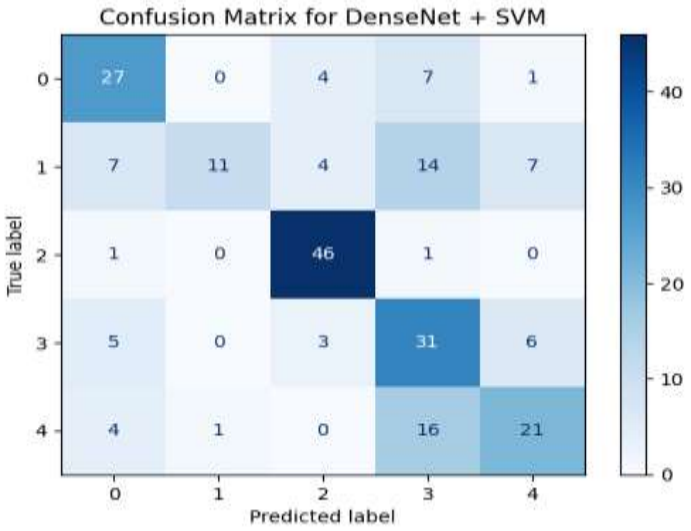

Fig. 12. Confusion matrix for DenseNet121+SVM.

Analysis of the Confusion Matrix

DenseNet121+SVM correctly predicts 136/217 validation images and is the weakest evaluated pipeline. It performs well on No_DR, with 46/48 correct, but has severe difficulty with Moderate and high-severity classes. Moderate recall is only 0.26 because many Moderate images are predicted as Proliferate_DR or Severe. Proliferate_DR and Severe are also confused with each other. These errors show that frozen DenseNet features extracted from ImageNet are not sufficiently adapted to DR morphology, and the linear SVM cannot recover missing disease-specific features.

Explanation of the Sample Output

No DenseNet121+SVM-specific example input, heat map, expected/predicted output, or attribution visualization is provided in the notebooks. Grad-CAM is naturally designed for differentiable CNN classifiers and is not directly available for the downstream SVM decision function unless a separate feature-attribution or occlusion experiment is implemented. Additional work is required to generate an SVM-compatible explanation, such as occlusion sensitivity over the DenseNet feature extractor, class activation from the CNN backbone, or SHAP/LIME on the SVM feature vector.

H. Consolidated Interpretation

Across the experiments, No_DR is consistently the easiest class for most models. DenseNet121, MobileNet+GNN, MobileNet+RNN, and AlexNet all obtain strong No_DR recall, indicating that the absence of disease-related lesions is visually separable from diseased classes. The most frequent mistakes occur between adjacent DR severity levels, especially Moderate, Proliferate_DR, and Severe. This is expected because these classes differ by lesion number, location, and progression stage rather than by entirely distinct visual patterns.

The results also demonstrate the importance of model adaptation. DenseNet121+SVM uses a powerful backbone but freezes its representation, leading to poor Moderate recall and high confusion among diseased categories. End-to-end DenseNet121 fine-tuning adapts the backbone to fundus-specific morphology and produces the best result. Lightweight MobileNetV2 hybrids show that feature-level post-processing can improve compact backbones, but graph construction and recurrent feature interpretation must be made clinically meaningful before deployment claims are made.

The Grad-CAM results provide qualitative but uneven support. DenseNet121 and MobileNet+GNN display broad activation over retinal content for correctly predicted examples. MobileNet+RNN correctly predicts a No_DR example, but its attention is partly peripheral. AlexNet predicts a Proliferate_DR case correctly but emphasizes the image boundary, which weakens interpretability. Keras MobileNet predicts a Severe sample correctly, but the heat map is almost inactive, so the explanation requires regeneration. Therefore, Grad-CAM should be treated as a diagnostic

visualization rather than conclusive clinical evidence.

6. Conclusion

This revised paper presents a notebook-grounded evaluation of six models for five-class diabetic retinopathy grading. End-to-end DenseNet121 achieves the strongest validation performance, confirming the benefit of dense feature reuse and fine-tuned domain adaptation. MobileNet+GNN and MobileNet+RNN improve the lightweight MobileNetV2 representation and show that hybrid classifiers can provide competitive performance under reduced backbone complexity. AlexNet remains a reasonable baseline but is less reliable for subtle severity separation. DenseNet121+SVM performs worst, showing that frozen features and a linear classifier are insufficient for robust DR grading on this dataset. The revised Results section adds model-wise confusion-matrix analysis and Grad-CAM-based sample interpretation, making the experimental discussion more meaningful than numerical reporting alone. At the same time, the revision identifies important gaps that must be addressed before final publication, including dataset provenance, corrected MobileNet confusion-matrix generation, five-class output alignment, best-checkpoint evaluation, and SVM-specific explainability.

7. Future Scope

Future work should first focus on reproducibility. The dataset source, preparation procedure, train/validation/test split strategy, class counts, hardware environment, and exact software versions should be recorded. The PyTorch DenseNet121 and AlexNet models should be retrained with exactly five output neurons to match the five DR classes. The standalone MobileNet confusion matrix should be regenerated using an unshuffled validation generator. MobileNet+GNN should be extended with a clinically meaningful graph, such as lesion-region nodes, vessel-region nodes, or learned feature-similarity edges, instead of the placeholder edge structure used in the notebook. Additional explainability methods such as occlusion sensitivity, SHAP, LIME, and lesion-aware Grad-CAM should be applied across all models. External validation on multi-centre datasets and model calibration analysis are also required before any clinical decision-support claims are made. Finally, lightweight deployment can be explored using pruning, quantization, and knowledge distillation, especially for MobileNet-based pipelines intended for low-resource screening settings.

References

1. International Diabetes Federation, IDF Diabetes Atlas, 10th ed. Brussels, Belgium: IDF, 2021.
2. M. S. Ip et al., "Retinal laser photocoagulation," Survey of Ophthalmology, vol. 43, suppl. 1, pp. S55-S83, 1998.
3. R. P. Crick and P. T. Khaw, A Textbook of Clinical Ophthalmology, 3rd ed. Singapore: World Scientific, 2003.
4. V. Gulshan et al., "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," JAMA, vol. 316, no. 22, pp. 2402-2410, 2016.
5. M. D. Abramoff et al., "Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices," NPJ Digital Medicine, vol. 1, no. 1, p. 39, 2018.
6. D. S. W. Ting et al., "Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes," JAMA, vol. 318, no. 22, pp. 2211-2223, 2017.
7. R. Gargeya and T. Leng, "Automated identification of diabetic retinopathy using deep learning," Ophthalmology, vol. 124, no. 7, pp. 962-969, 2017.
8. S. Qummar et al., "A deep learning ensemble approach for diabetic retinopathy detection," IEEE Access, vol. 7, pp. 150530-150539, 2019.
9. H. Pratt, F. Coenen, D. M. Broadbent, S. P. Harding, and Y. Zheng, "Convolutional neural networks for diabetic retinopathy," Procedia Computer Science, vol. 90, pp. 200-205, 2016.
10. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2017, pp. 4700-4708.
11. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2016, pp. 770-778.
12. X. Li, T. Pang, B. Xiong, W. Liu, P. Liang, and T. Wang, "Convolutional neural networks based transfer learning for diabetic retinopathy fundus image classification," in Proc. 10th Int. Conf. Information Science and Control Engineering, 2017, pp. 1262-1266.
13. A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv

preprint arXiv:1704.04861, 2017.
14. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2018, pp. 4510-4520.
15. M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. Int. Conf. Machine Learning, 2019, pp. 6105-6114.
16. C. Qian, X. Chen, and A. Bhatt, "Lightweight deep learning framework for diabetic retinopathy grading on mobile devices," Journal of Biomedical Informatics, vol. 112, p. 103602, 2020.
17. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in Proc. Int. Conf. Learning Representations, 2017.
18. J. Zhao et al., "Graph-based retinal image analysis for diabetic retinopathy lesion detection," IEEE Transactions on Medical Imaging, vol. 39, no. 5, pp. 1635-1648, 2020.
19. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.
20. J. L. Elman, "Finding structure in time," Cognitive Science, vol. 14, no. 2, pp. 179-211, 1990.
21. C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.
22. J. I. Orlando, E. Prokofyeva, and M. B. Blaschko, "A discriminatively trained fully connected conditional random field model for blood vessel segmentation in fundus images," IEEE Transactions on Biomedical Engineering, vol. 64, no. 1, pp. 16-27, 2017.
23. K. Raza, N. Singh, and L. Chauhan, "Comparative analysis of deep learning and support vector machine for diabetic retinopathy detection," International Journal of Advanced Computer Science and Applications, vol. 11, no. 3, pp. 388-396, 2020.
24. EyePACS, "Diabetic Retinopathy Detection," Kaggle Competition Dataset, 2015.
25. P. Porwal et al., "IDRiD: Diabetic retinopathy--segmentation and grading challenge," Medical Image Analysis, vol. 59, p. 101561, 2020.