Eye-Dentification: A Deep Learning-Based Ocular Disease Classification Tool Using Fundus Imaging

Aaditya Penmetsa*
Beaver Works Summer Institute
Massachusetts Institute of Technology
aadipenmetsa@gmail.com

Chenlu (Brigitta) Yu*
Beaver Works Summer Institute
Massachusetts Institute of Technology
chenluyu06@gmail.com

Yingqi (Cheech) Li*
Beaver Works Summer Institute
Massachusetts Institute of Technology
yingqili280@gmail.com

***Abstract*— Ocular diseases are responsible for a significant number of cases of vision impairment worldwide. Early intervention vital towards mitigating irreversible vision loss. This study presents Eye-Dentification, a deep learning-based image classification system designed to classify three categories of diseases: cataract, glaucoma, and diabetic retinopathy. The model utilizes convolutional neural network (CNN) architectures to analyze fundus images and accurately classify the condition. We evaluated our models—DenseNet201, EfficientNetB5, InceptionV3, and ResNet50—using accuracy, F1-score, ROC-AUC, and confusion matrix metrics. We found that DenseNet201 exhibited the highest glaucoma sensitivity (~94%), making it reliable for early-stage screening, while EfficientNetB5 provided a strong trade-off between performance and computational power. A streamlet website was developed to demonstrate user interaction and real-time prediction capabilities, with the support of DenseNet201 and InceptionV3. Our training process employed preprocessing, early stopping, and appropriate loss functions. In the future, we seek to integrate Grad-CAM for explainability, expand our model to classify more ocular diseases, and optimize the interface for mobile deployment. Our study highlights the benchmarks of convolutional neural networks on clinical applications.**

Keywords—Ocular disease classification, deep learning, convolutional neural network, EfficientNet, Streamlit, fundus

I. Introduction

Ocular diseases are the leading cause of blindness globally, yet diagnosis often depends on access to ophthalmologists and specialized equipment. Machine learning offers a promising approach to automating detection of retinal images. Our tool classifies medical fundus screenings. Our goals were to compare the performance of multiple CNN architectures. Identify the best model for multiclass fundus image classification. Deploy the trained model in an interactive web app. And visualize key training metrics including accuracies, loss curves, F1 scores, and confusion matrices.

II. Methods

A. Dataset

We used a preprocessed dataset consisting of over 4000 fundus images labeled for four target classes:

Figure 1. Distribution of classes in Training Data. Classes were evenly distributed to minimize the risk of bias in our model.

All images were resized to match model-specific input shapes (e.g., 224x224 for DenseNet201, 456x456 for EfficientNetB5). Then the images were normalized using model-specific preprocess input functions from TensorFlow.

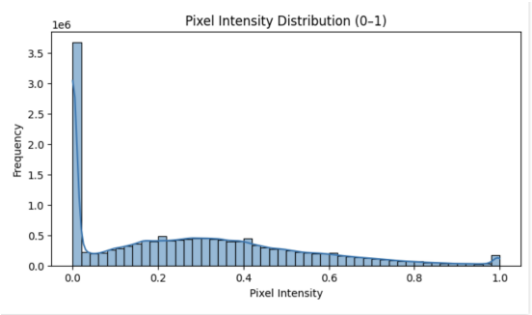

Figure 2. Pixel Intensity Distribution. Evenly split

Next, the images were converted into NumPy arrays for compatibility with our custom pipeline. Finally, the images are stratified into train, validation, and test splits.

B. Preprocessing Pipeline

* Authors contributed equally to this work

Figure 3. Data split

C. *Model Architectures*

We evaluated the following architectures:

1.EfficientNet

Model: "sequential"

Layer (type)	Output Shape	Param #
efficientnetb3 (Functional)	(None, 7, 7, 1536)	10,783,535
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1536)	0
dense (Dense)	(None, 256)	393,472
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 4)	516

Total params: 11,210,419 (42.76 MB)
Trainable params: 11,123,116 (42.43 MB)
Non-trainable params: 87,303 (341.03 KB)

Figure 4. EfficientNetB3 Model Architecture

2.InceptionV3

Layer (type)	Output Shape	Param #
inception_v3 (Functional)	(None, 5, 5, 2048)	21,802,784
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 2048)	0
dense_4 (Dense)	(None, 48)	98,352
dropout_2 (Dropout)	(None, 48)	0
dense_5 (Dense)	(None, 4)	196

Total params: 21,901,332 (83.55 MB)
Trainable params: 98,548 (384.95 KB)
Non-trainable params: 21,802,784 (83.17 MB)

Figure 5. InceptionV3 Model Architecture

3.DenseNet

Model: "sequential"

Layer (type)	Output Shape	Param #
densenet201 (Functional)	(None, 7, 7, 1920)	18,321,984
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1920)	0
dense (Dense)	(None, 256)	491,776
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 4)	516

Total params: 18,847,172 (71.90 MB)
Trainable params: 18,618,116 (71.02 MB)
Non-trainable params: 229,056 (894.75 KB)

Figure 6. DenseNet201 Model Architecture

4.Custom CNN

Layer (type)	Output Shape	Param #
conv2d_94 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_4 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_95 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_5 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_96 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_6 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense_33 (Dense)	(None, 128)	11,075,712
dropout_20 (Dropout)	(None, 128)	0
dense_34 (Dense)	(None, 4)	516

Total params: 11,169,476 (42.61 MB)
Trainable params: 11,169,476 (42.61 MB)
Non-trainable params: 0 (0.00 B)

Figure 7. Custom CNN Model Architecture

All our models were trained using transfer learning with ImageNet weights. Classification head included batch normalization, ReLU activation, dropout, and a dense SoftMax layer.

D. *Training Details*

1. **Optimization & Convergence**: Utilized the Adam optimizer (learning rate = 1e-4, β_1=0.9, β_2=0.999) across all models. Early stopping (patience=5) was enabled to accommodate overfitting.
2. **Batch Size Considerations**: EfficientNetB5 was trained on a batch size of 8. Whereas DenseNet201 and InceptionV3 were trained on 64 epochs.
3. **Class Weights**: No class weighting was applied because of the relatively balanced dataset.
4. **Deployment-Specific Modifications:** EfficientNetB5 was trained on sparse categorical cross entropy whereas DenseNet201 and InceptiopnV3 were trained on Categorical cross entropy.
5. **Training Stability and Regularization**: All models were trained using a dropout layer in a range of 0.3-0.4. Batch normalization layers followed global average pooling.

III. Results

A. *Model Comparison*

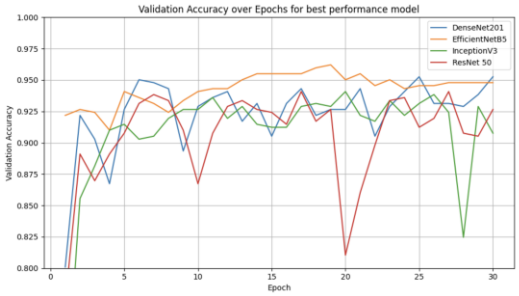

Figure 8. Validation Accuracy Comparison Between Models

B. Confusion Matrices

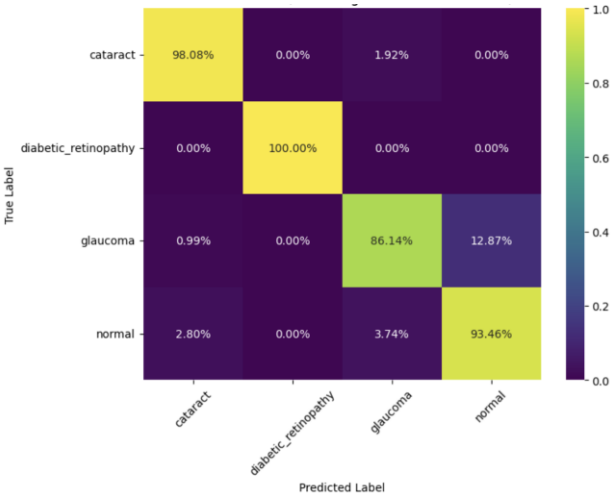

Figure 8. ResNet50 Confusion Matrix

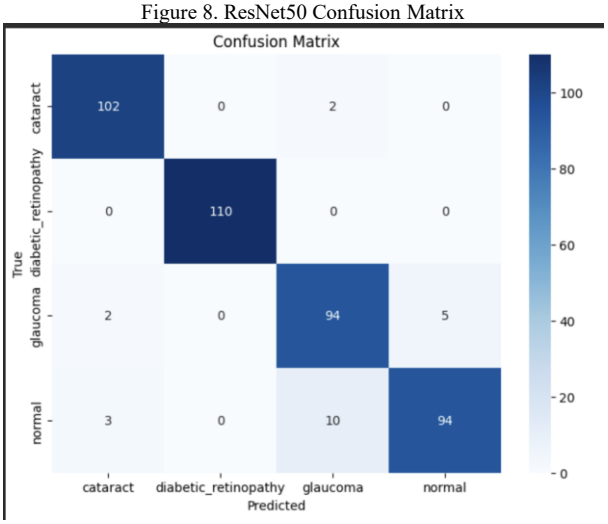

Figure 9. EfficientNetB5 Confusion Matrix

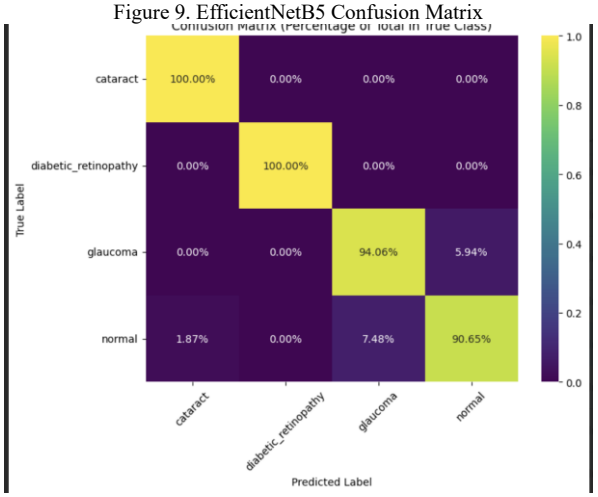

Figure 10. DenseNet201 Confusion Matrix

We analyze how each model balanced sensitivity and specificity, with a particular focus on distinguishing glaucoma from normal. Glaucoma detection sensitivity emerged as a key differentiator between models. DenseNet201 and EfficientNetB5 both correctly identified approximately 90-94% of glaucoma cases. While ResNet50 detected around 86%. Therefore, DenseNet201 and EfficientNetB5's success rate suggests they may be better suited for early detection. On the other hand, specificity showed the opposite trend. ResNet50 correctly labeled approximately 93% of normal cases, outperforming DenseNet201 with a 91% and EfficientNetB5 with an 88%. EfficientNetB5 misclassified 13 out of 107 normal images, whereas ResNet50 misclassified 7. DenseNet201's count fell in between. These results indicate that ResNet50 is the most conservative model, with fewer errors on healthy patients. This trade-off reflects a tension in diagnostic AI. While high sensitivity reduces unnecessary follow-ups, high specificity is more important in screening contexts as failing to identify a disease poses risks. Previous research supports the idea that screening tools are designed to favor sensitivity even at the expense of some specificity [7][8]. DenseNet201 and EfficientNetB5's moderate drop in specificity may be acceptable. In contrast, ResNet50's lower sensitivity may pose a significant clinical limitation for early-stage diagnosis.

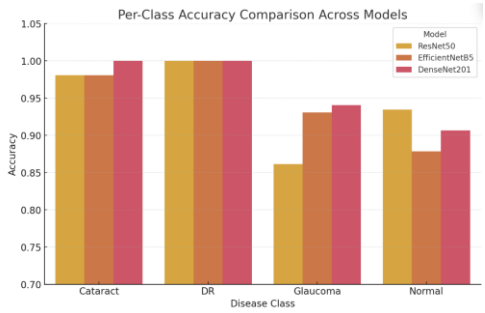

Figure 12. Comparison Between Class Accuracy

C. ROC Curves and AUC Values Per Class

Model	Cataract	Diabetic Retinopathy	Glaucoma	Normal	Micro-Average
DenseNet 201	0.9993	1.0000	0.9921	0.9925	0.9964
Efficient NetB5	0.9986	1.0000	0.9900	0.9903	0.9951
ResNet50	0.9986	1.0000	0.9861	0.9883	0.9951

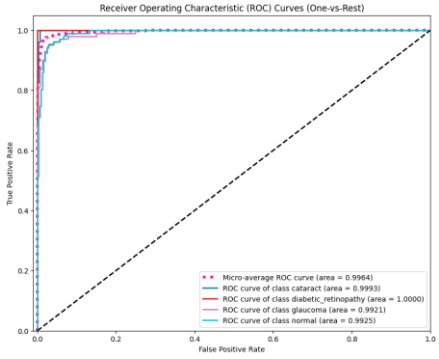

Figure 13. DenseNet201 ROC Curve

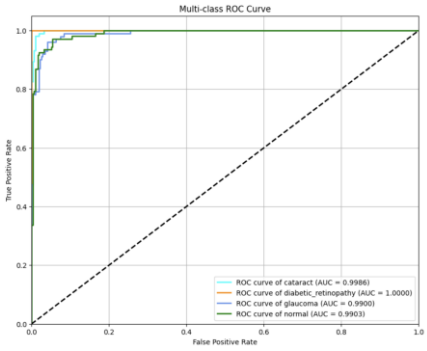

Figure 14. EfficientNetB5 ROC Curve

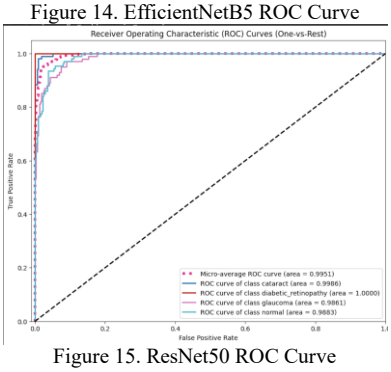

Figure 15. ResNet50 ROC Curve

D. F1 Scores

	precision	recall	f1-score	support
cataract	0.95	0.98	0.97	104
diabetic_retinopathy	1.00	1.00	1.00	110
glaucoma	0.89	0.93	0.91	101
normal	0.95	0.88	0.91	107
accuracy			0.95	422
macro avg	0.95	0.95	0.95	422
weighted avg	0.95	0.95	0.95	422

Macro F1 Score: 0.9469146410398844

Figure 16. EfficientNetB5 F1 Scores

7/7 ━━━━━━━━━━━━━━━━━━━━ 6s 877ms/step

	precision	recall	f1-score	support
cataract	0.98	1.00	0.99	104
diabetic_retinopathy	1.00	1.00	1.00	110
glaucoma	0.92	0.94	0.93	101
normal	0.94	0.91	0.92	107
accuracy			0.96	422
macro avg	0.96	0.96	0.96	422
weighted avg	0.96	0.96	0.96	422

Figure 17. DenseNet201 F1 Scores

	precision	recall	f1-score	support
cataract	0.96	0.98	0.97	104
diabetic_retinopathy	1.00	1.00	1.00	110
glaucoma	0.94	0.86	0.90	101
normal	0.88	0.93	0.91	107
accuracy			0.95	422
macro avg	0.95	0.94	0.94	422
weighted avg	0.95	0.95	0.95	422

Figure 18. ResNet50 F1 Scores

These CNN models achieve high overall accuracy on a multi-class eye disease dataset encompassing cataract, diabetic retinopathy, glaucoma, and normal classes. Their respective ROC curves were near-perfect, ranging from 0.995 to 0.996. DenseNet201 and EfficientNetB5 achieved the highest glaucoma-specific AUC's (~0.990-0.992), compared to ResNet50 (~0.986), suggesting slightly better discrimination between glaucoma and other conditions. While all models demonstrated strong ranking capabilities, the best performing models reached near perfect scores for diabetic retinopathy (1.00) and cataract (~0.999). These results suggest that the models are unlikely to confuse diseased and healthy eyes when using threshold-based decision rules.

Despite comparable metrics, performance varied. DenseNet201 achieved higher macro-averaged F1-scores (~0.96) than ResNet50 and EfficientNetB5 (~0.95), indicating more consistent classification performance across all disease classes. ResNet50's macro-average F1 was notably affected by its lower performance on glaucoma (F1 = 0.90), whereas its micro-average F1 remained high (~0.95) because of strong performance on more separable classes like diabetic retinopathy. In contrast, DenseNet201 demonstrated nearly equal macro and micro F1 scores (~0.96), suggesting robust and balanced generalization across all four categories.

Looking at the architecture, these performance differences may stem from DenseNet's feature reuse through dense connectivity [1], and EfficientNet's scaling strategy that optimally balances network depth, width, and resolution [2]. In contrast, ResNet50's fixed-depth residual structure [3] may be less effective at capturing class distinctions. These architectural distinctions are critical in medical screening applications, where consistent per-class performance outweighs aggregate accuracy. Glaucoma is frequently underdiagnosed due to its subtle visual features, making high recall and precision in minority classes especially important [5][6].

IV. Web App Deployment

A. Tech Stack

1. Frontend: Streamlit
2. Backend: Python, TensorFlow
3. Deployment: Streamlit Cloud

B. User Interface Features

1. Model selection dropdown
2. Image upload widget
3. Predicted label and confidence scores display
4. User feedback

C. Clinical Relevance and Model Selection

DenseNet201 emerges as the most clinically favorable model, especially in the case of early and accurate disease detection. In the context of glaucoma, DenseNet201's sensitivity (~94%) stands out. It missed only ~6% of glaucoma cases, outperforming both EfficientNetB5 (~7%) and ResNet50 (~14%). In an environment where diagnosing diseases early is crucial, DenseNet201 is perfect. DenseNet201's trade-off is optimal. It can ensure that nearly all patients with glaucoma are correctly identified.

While EfficientNetB5 also performed well, its slightly lower specificity and higher false-positive rate on normal cases may lead to unnecessary issues. Furthermore, although EfficientNet models are optimized for compute efficiency and

scalability [11][12], the accuracy-to-compute advantage is less critical in clinical environments with access to GPUs. DenseNet201's architecture is efficient and has a relative 20 million parameters. Making it a feasible option for deployment in hospital or clinical settings.

DenseNet201, offers the most favorable clinical performance, balancing excellent multi-class accuracy, superior glaucoma sensitivity, and reasonable computing demands. It is well-suited for robust clinical setups where patient safety and diagnostic accuracy are the main concerns. EfficientNetB5 remains a strong secondary option, especially for mobile deployment. ResNet50, is aligned with the needs of a high-sensitivity medical screening tool.

V. Discussion

Eye-Dentification highlights the feasibility of building a low-cost tool for ocular disease screening using pre-trained CNNs and open-source tools. While DenseNet201 proved to be most effective, differences in performance across disease classes indicate the need for further data augmentation and potential use of attention mechanisms or focal loss to better handle class imbalance.

VI. Future Improvements

The current version of Eye-Dentification demonstrates high diagnostic performance across all ocular conditions. Future work will address three key development axes: explainability, disease coverage, and deployment optimization.

1. Integration of Model Explainability via Grad-CAM: To enhance clinical interpretability, we will integrate Grad-CAM (Gradient-weighted Class Activation Mapping) into the interface pipeline. This generates class-specific heatmaps superimposed on the input fundus image. Such action is critical for clinical adoption and for identifying failure modes in misclassified cases.
2. Extension to Additional Ocular Pathologies and Datasets: Expanding the model's diagnostic scope by incorporating additional disease classes, including age-related macular degeneration (AMD), hypertensive retinopathy, and retinal vascular occlusions. This requires retraining on more comprehensive datasets.
3. Model Compression and Edge Deployment Optimization: To facilitate real-time use on mobile devices. We plan to investigate model compression techniques to tackle this issue. Target platforms include Android tablets and ARM-based devices. Conversion to TensorFlow Lite or ONNX formats will be benchmarked for latency, accuracy, and energy efficiency in clinical settings.

Acknowledgements

We would like to thank our instructor, teaching assistants, fellow program attendees, and program directors at the Beaver Works Summer Institute for their support in this project. As always, like BWSI, stay transformative!

References

[1] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708, 2017.

[2] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," International Conference on Machine Learning, pp. 6105–6114, 2019.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.

[4] D. M. W. Powers, "Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation," Journal of Machine Learning Technologies, vol. 2, no. 1, pp. 37–63, 2011.

[5] E. Beede, E. Baylor, F. Hersch, A. Iurchenko, L. Wilcox, P. Ruamviboonsuk, and L. P. Vardoulakis, "A Human-Centered Evaluation of a Deep Learning System Deployed in Clinics for the Detection of Diabetic Retinopathy," Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–12, 2020.

[6] V. Gulshan, L. Peng, M. Coram, 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.

[7] G. Huang, Z. Liu, L. van der Maaten and K. Q. Weinberger,
"Densely Connected Convolutional Networks,"
in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261–2269.
https://doi.org/10.1109/CVPR.2017.243

[8] M. Tan and Q. Le,
"EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,"
in Proceedings of the 36th International Conference on Machine Learning (ICML), 2019, pp. 6105–6114.
https://arxiv.org/abs/1905.11946

[9] K. He, X. Zhang, S. Ren, and J. Sun,
"Deep Residual Learning for Image Recognition,"
in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
https://doi.org/10.1109/CVPR.2016.90