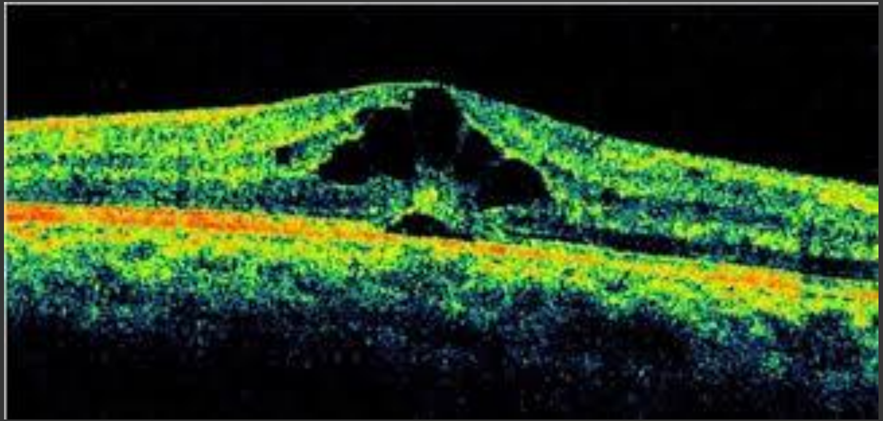
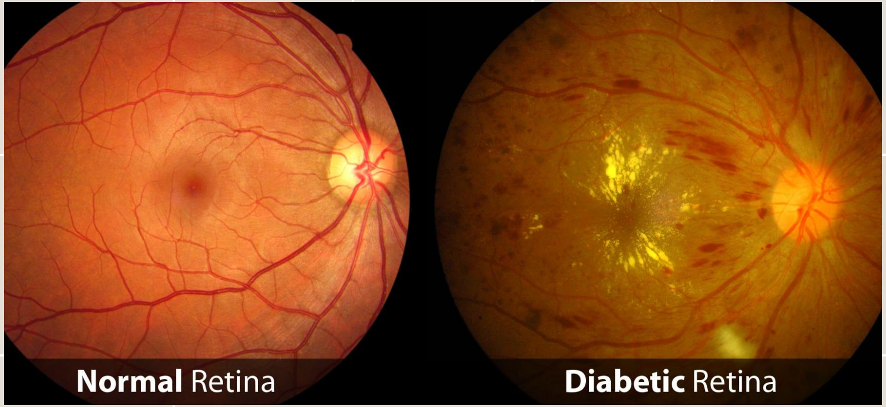
Brandon Gambrall
SAT5165
12/5/2025



Retinopathy Grade Detection

Using Convolutional Neural Network

What is retinopathy?



1. Microaneurysm damage to the blood vessels in the retina over time
2. Main cause is diabetes but also from hypertension/obesity
3. Can not be cured or reversed only slowing of progression
4. Causes blurred and distorted vision up to blindness

SAT5165

Issue:

Diabetic Retinopathy is a leading cause of preventable blindness

Assigning retinopathy grade can speed up medical diagnosis and interventions to slow progression.

The problem I tried to solve with this project is:

Build a deep learning model capable of classifying retinal images into appropriate retinopathy grades using a convolutional neural network.

The dataset I used:

Retinal Disease Detection – Macular Edema Risk Prediction by Mohamed Abdalkader

Each CSV file contains: Image Name (4000), retinopathy grade, macular edema risk, and visit caption

URL: https://www.kaggle.com/datasets/mohamedabdalkader/retinal-disease-detection

Exploratory Data Analysis

Sample Size Distribution

(roughly 70/15/15 split)

Split	Images	Notes
Train	~5,300	used for model learning
Valid	~1,200	used for tuning and early stopping
Test	~1,000	final performance measurement

Images Size and Quality

All images resized to 224x224x3

Moderate variation in background lighting

Moderate amount of image artifact and distortion

Class Distribution Histogram

Class	Count	Percent
No DR	~1800	≈ 40%
Mild	~600	≈ 13%
Moderate	~1500	≈ 33%
Severe	~300	≈ 7%
Proliferative DR	~200	≈ 4%

Preprocessing

1. File Renaming and Column Cleaning

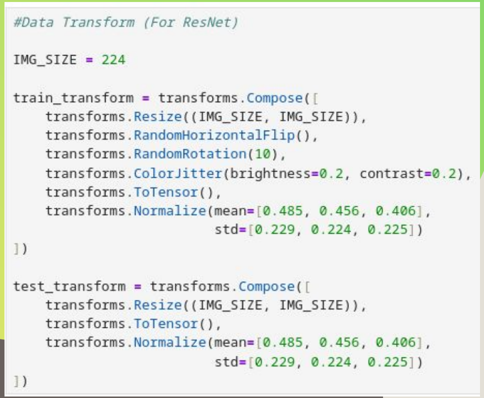
- “Image name” file was changed to just “image”
- “Retinopathy grade” column was changed to just “label”
- Structure that is best for working with PyTorch, TensorFlow, etc.

2.Dataset Merge & Conversion

- Read with Spark
- Converted to Pandas
- Converted to PyTorch dataset class

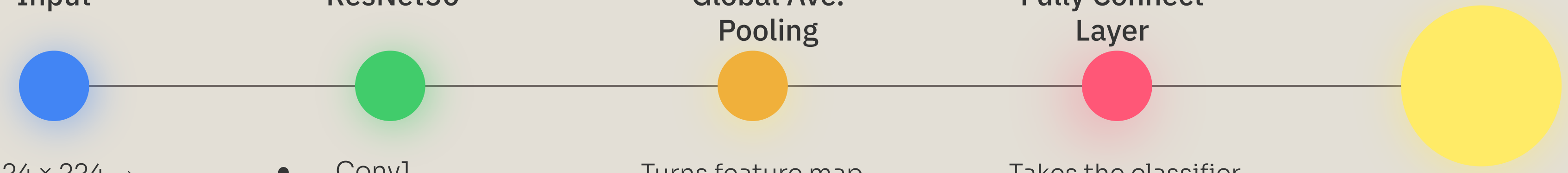
3. Image Preprocessing

- ResNet transformations and train augmentations (reduces overfitting and improve training accuracy)



```
#Data Transform (For ResNet)

IMG_SIZE = 224

train_transform = transforms.Compose([
    transforms.Resize((IMG_SIZE, IMG_SIZE)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(10),
    transforms.ColorJitter(brightness=0.2, contrast=0.2),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406],
                         std=[0.229, 0.224, 0.225])
])

test_transform = transforms.Compose([
    transforms.Resize((IMG_SIZE, IMG_SIZE)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406],
                         std=[0.229, 0.224, 0.225])
])
```

CNN Model Structure



Input

224 × 224 → standard resolution

3 channels → RGB color images

ResNet50

- Conv1
- Batchnorm1
- ReLu
- MaxPool
- Layer1
- Layer2
- Layer3
- Layer4

Layer 1-4 is why ResNet works well on Medical Images

Global Ave. Pooling

Turns feature map into a vector and output becomes classifier

Fully Connect Layer

Takes the classifier and maps to one of the retinopathy grades (class).

Softmax

Converts each class into a probability equal to one.

Class	Softmax Probability
No DR	0.72
Mild	0.10
Moderate	0.08
Severe	0.05
Proliferative DR	0.05

Hyperparameters

Tuned Hyperparameters by Monitoring Validation Accuracy and loss curves

- Learning rate: 1e-3 was unstable compared to 1e-4
- Data Augmentation
- Adam knows best- stable training for medical images and adaptive learning rate
- Dropout 0.3 was decided as a good fit
- Tested Batch Sizes: 16 vs 32

Hyperparameter	Value
Architecture	Frozen Layers
Image Size	224 × 224
Optimizer	Adam
Learning Rate	1e-4
Loss Function	CrossEntropyLoss
Epochs	20
Weight Decay	1e-5
Dropout	0.3 on classifier
Augmentations	flip, rotate, color jitter

Model Performance

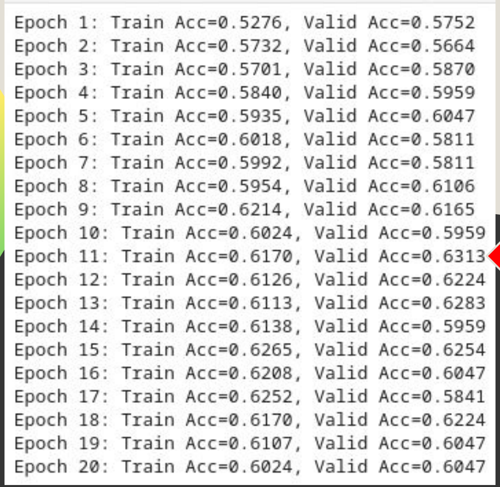
ResNet superior model

First model built was a simple CNN model

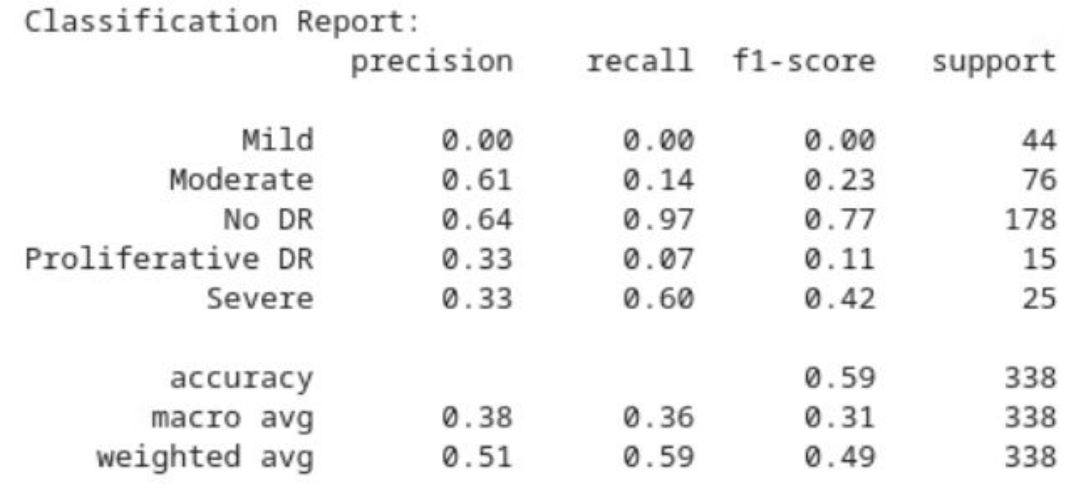
- Peak Accuracy was 55%

ResNet Model performed better

- Peak Accuracy is 63%
- Best performance on No DR class
- Has extreme difficulty with minority classes
- Minority class size from data set hurt the model

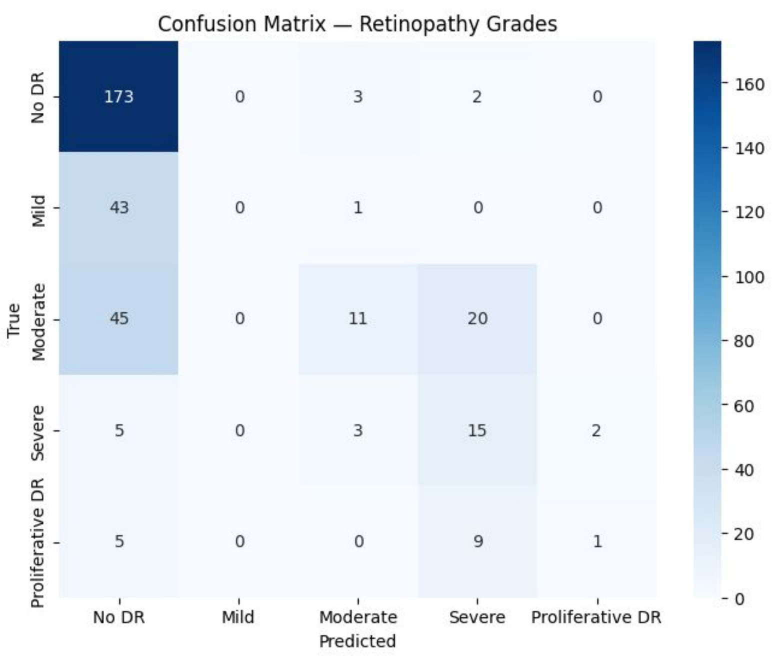


```
Epoch 1: Train Acc=0.5276, Valid Acc=0.5752
Epoch 2: Train Acc=0.5732, Valid Acc=0.5664
Epoch 3: Train Acc=0.5701, Valid Acc=0.5870
Epoch 4: Train Acc=0.5840, Valid Acc=0.5959
Epoch 5: Train Acc=0.5935, Valid Acc=0.6047
Epoch 6: Train Acc=0.6018, Valid Acc=0.5811
Epoch 7: Train Acc=0.5992, Valid Acc=0.5811
Epoch 8: Train Acc=0.5954, Valid Acc=0.6106
Epoch 9: Train Acc=0.6214, Valid Acc=0.6165
Epoch 10: Train Acc=0.6024, Valid Acc=0.5959
Epoch 11: Train Acc=0.6170, Valid Acc=0.6313
Epoch 12: Train Acc=0.6126, Valid Acc=0.6224
Epoch 13: Train Acc=0.6113, Valid Acc=0.6283
Epoch 14: Train Acc=0.6138, Valid Acc=0.5959
Epoch 15: Train Acc=0.6265, Valid Acc=0.6254
Epoch 16: Train Acc=0.6208, Valid Acc=0.6047
Epoch 17: Train Acc=0.6252, Valid Acc=0.5841
Epoch 18: Train Acc=0.6170, Valid Acc=0.6224
Epoch 19: Train Acc=0.6107, Valid Acc=0.6047
Epoch 20: Train Acc=0.6024, Valid Acc=0.6047
```



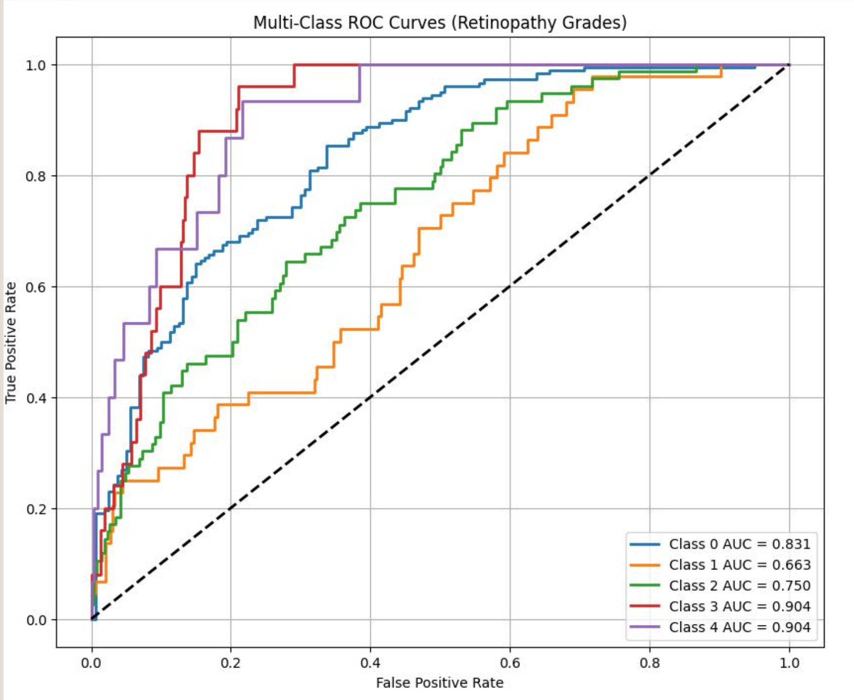
```
Classification Report:
                  precision    recall  f1-score   support

            Mild       0.00      0.00      0.00        44
        Moderate       0.61      0.14      0.23        76
           No DR       0.64      0.97      0.77       178
Proliferative DR       0.33      0.07      0.11        15
          Severe       0.33      0.60      0.42        25

        accuracy                           0.59       338
       macro avg       0.38      0.36      0.31       338
    weighted avg       0.51      0.59      0.49       338
```

Model Performance



Confusion Matrix — Retinopathy Grades

	No DR	Mild	Moderate	Severe	Proliferative DR
No DR	173	0	3	2	0
Mild	43	0	1	0	0
Moderate	45	0	11	20	0
Severe	5	0	3	15	2
Proliferative DR	5	0	0	9	1



Multi-Class ROC Curves (Retinopathy Grades)

Why the results matter

The model performed poorly due circumstances given.

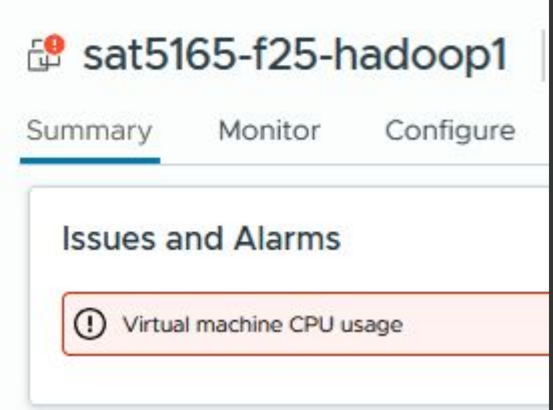
The model currently over classifies No DR which is problematic.

1. Misdiagnosis is highly likely
2. Undermines the goal of this model

Class	Count	Percent
No DR	~1800	≈ 40%
Mild	~600	≈ 13%
Moderate	~1500	≈ 33%
Severe	~300	≈ 7%
Proliferative DR	~200	≈ 4%

Ways to improve the model

1. Most important is address the class imbalance at the data set level
2. Increase Augmentation Strength
3. Continue to fine tune model
4. Use a better computational setup to run model to increase performance



sat5165-f25-hadoop1

Issues and Alarms

Virtual machine CPU usage

Any questions?
Ask away!

Work Cited

Abdalkader, Mohamed. “Retinal Disease Detection.” *Kaggle*, 27 Aug. 2025, www.kaggle.com/datasets/mohamedabdalkader/retinal-disease-detection.

“Build Your First Convolutional Neural Network to Recognize Images | by Intuitive Deep Learning (by Joseph) | Intuitive Deep Learning | Medium.” *Build Your First Convolutional Neural Network to Recognize Images*, Medium, 21 Mar. 2019, medium.com/intuitive-deep-learning/build-your-first-convolutional-neural-network-to-recognize-images-84b9c78fe0ce.

Diabetic Retinopathy, Mayo Foundation for Medical Education and Research, www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/symptoms-causes/syc-20371611. Accessed 20 Nov. 2025.

“Image Classification Using ResNet.” *GeeksforGeeks*, GeeksforGeeks, 23 July 2025, www.geeksforgeeks.org/computer-vision/image-classification-using-resnet/.

Thank you

Special thanks to Andrea Wroblewski and Olivia Gette!