10/28/2025

Retinal Disease Detection using Multi-Task Deep Learning

By: Armaan Patel

Table of Contents:

Folders and files:

- Retinal Disease Detection using Multi-Task Deep Learning.pdf - contains full report
- test - folder containing test files
    - images - folder containing test retinal images
    - annotations.csv - CSV file that contains information about the retinal images
- train - folder containing training files
    - images - folder containing training retinal images
    - annotations.csv - CSV file that contains information about the retinal images
- valid - folder containing validation files
    - images - folder containing validation retinal images
    - annotations.csv - CSV file that contains information about the retinal images
- Retinal_Disease_Detection.ipynb - JupyterLab notebook that contains all code
- retina_multitask_best.pth - best model
    - generated after running notebook
- retina_multitask_final.pth - final model
    - generated after running notebook
- test_predictions.csv - CSV file that contains the predictions that came from the test

Import test, train, and valid folders from the Kaggle link into the same folder as Retinal_Disease_Detection.ipynb and run the notebook to create the model files and test_predictions.csv file.

Deliverables:
- Jupyter Notebook containing preprocessing, model training, and evaluation
- Trained model files (`retina_multitask_best.pth`, `retina_multitask_final.pth`)
- CSV of test predictions (`test_predictions.csv`)
- Grad-CAM visualizations and performance metrics
- Final written report summarizing methods, results, and discussion

Project Goals:
- Develop a multi-task deep learning model to detect diabetic retinopathy severity and macular edema risk from retinal fundus images.
- Evaluate model performance using training, validation, and test sets with metrics such as accuracy, F1-score, and confusion matrices.
- Visualize model attention and interpretability using Grad-CAM to ensure clinical relevance.
- Save and deploy the best-performing model for reproducibility and future experimentation.

Tools:
JupyterLab, Python

Libraries:
PyTorch, Torchvision, Scikit-learn, Pandas / NumPy, Pillow (PIL), Matplotlib, OpenCV

Acknowledgments:
Dataset provided by Mohamed Abdalkader via Kaggle.
Special thanks to open-source contributors whose work made this project possible.

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

Project Summary:

This project develops an explainable deep learning model for automated retinal disease detection using fundus images from the Retinal Disease Detection dataset on Kaggle. The model is built with PyTorch and employs a multi-task convolutional neural network (CNN) that simultaneously predicts both diabetic retinopathy severity and risk of macular edema. A shared ResNet-50 backbone extracts visual features, while two specialized heads classify disease grade and edema risk. To enhance interpretability, Grad-CAM visualizations highlight the retinal regions most influential to each prediction, demonstrating that the network focuses on clinically relevant features such as hemorrhages, exudates, and macular changes. The result is a reliable and transparent diagnostic aid that blends accuracy with explainability, supporting future applications in ophthalmic screening and medical AI research.

Key Features:

Dataset:

- Kaggle Retinal Disease Detection dataset containing annotated fundus images divided into train, validation, and test sets, with detailed clinical captions.
- Columns: Image name, Retinopathy grade, Risk of macular edema, Caption.
- Images sourced from diverse patient cases representing multiple disease stages.

Multi-Task Deep Learning Model:

- Built with PyTorch using a ResNet-50 backbone pretrained on ImageNet.
- Two classification heads:
    - Head 1 – Retinopathy Grade: Multi-class classification (e.g., grades 0 – 4).
    - Head 2 – Macular Edema Risk: Binary classification (0 = No risk, 1 = At risk).
- Trained jointly with weighted cross-entropy losses for balanced learning.

Training Pipeline:

- Data augmentation (resize, rotation, color jitter, flips) for generalization.
- Weighted sampling to address class imbalance.
- Adaptive learning-rate scheduling and checkpoint saving for best validation performance.
- Implemented end-to-end in JupyterLab for reproducibility.

Evaluation Metrics:

- Accuracy, precision, recall, and F1-score for both tasks.
- Confusion matrices and classification reports for detailed performance breakdown.
- Separate metrics reported for training, validation, and test sets.

Explainability & Visualization:

- Grad-CAM used to generate class activation maps for both heads.
- Highlights lesion areas (microaneurysms, hemorrhages, exudates) for retinopathy and macular region for edema.
- Confirms the model's attention aligns with clinically relevant features.

Experiment Tracking:

- Best model artifacts automatically saved and versioned.

Outcome:

- Achieved interpretable, high-accuracy classification of retinal diseases.
- Demonstrated potential for AI-assisted diabetic retinopathy and edema screening tools.

Results:

After training the multi-task CNN on the Retinal Disease Detection dataset, the model achieved strong performance on both disease grading and macular edema risk prediction tasks.

Task	Metric	Training	Validation	Test
Retinopathy Grade	Accuracy	0.867	0.693	0.719
Macular Edema Risk	Accuracy	0.990	0.941	0.959

(Note: Results vary slightly based on random seed, augmentations, and fine-tuning strategy.)

Interpretation of Results:

- Retinopathy Grade Head:
    - The model successfully distinguishes between normal, mild, moderate, and severe diabetic retinopathy cases. Misclassifications primarily occur between adjacent grades (e.g., moderate vs. severe), which aligns with clinical ambiguity between borderline stages.
- Macular Edema Head:
    - The binary classifier performs robustly, often detecting edema risk accurately when the model's attention focuses near the macular region. High recall indicates strong sensitivity to early edema indicators.

Confusion Matrices:
Both heads show clear diagonal dominance, confirming consistent predictions across classes. Small off-diagonal entries suggest occasional confusion between visually similar disease levels.

Grad-CAM Insights; Grad-CAM visualizations revealed that:

- The Retinopathy Head activates strongly around microaneurysms, hemorrhages, and exudates scattered throughout the retina.
- The Macular Edema Head consistently attends to the central macula, where fluid accumulation typically appears.
- These activation patterns confirm that the network is focusing on clinically relevant retinal features, not on background noise or artifacts.
- The explainability component provides transparency into the model's decision process, an essential factor for medical AI trustworthiness.

Discussion & Future Work

- Strengths:
    - Multi-task learning improved overall feature sharing and reduced overfitting compared to training separate single-task models.
    - Integration of Grad-CAM provided interpretability, aligning predictions with ophthalmic knowledge.
    - High test accuracy suggests strong potential for real-world screening applications.
- Limitations:
    - Dataset size and label quality limit clinical generalization.
    - Slight class imbalance across retinopathy grades affects edge-case performance.
    - Model assumes consistent image quality and lighting across datasets.
- Future Improvements:
    - Fine-tune the full ResNet-50 backbone at a lower learning rate for enhanced feature extraction.
    - Train with higher-resolution inputs (512×512 or 768×768) for better lesion-level detail.
    - Integrate the Caption column using a text encoder (e.g., BERT) for multimodal learning.
    - Explore deployment with ONNX or TorchScript for edge/clinical environments.

Conclusion:

This project demonstrates that explainable multi-task deep learning can accurately detect retinal diseases and macular edema risk from fundus photographs.
By combining performance with transparency through Grad-CAM, this approach bridges the gap between AI automation and clinical interpretability, showcasing the promise of trustworthy medical imaging AI.

Documentation:

JupyterLabs Notebook
- Imported libraries
- Configured base settings like seed
- Data transforms
    - Define data augmentation for training and deterministic transforms for validation/testing
    - Resize to a fixed square to match backbone input and stabilize batch shapes
    - Light augmentations (flip, small rotation, color jitter) to improve robustness
    - Normalize to ImageNet mean/std
- Robust Image Path Resolver
    - This is for datasets that might use other image formats
- Dataset Class
    - Reads annotations.csv
    - Ensures integer dtypes for labels
    - Maps retinopathy grades to contiguous indices {0..K-1} internally (keeps a mapping to the original values for reporting)
    - Returns a tuple: (image_tensor, dr_label_idx, me_label, filename, caption)
- Handling Class Imbalance and Dataloaders
    - Computes class weights for DR head
    - Optionally uses a WeightedRandomSampler so rare grades appear more often in training batches
    - Builds DataLoaders for train/valid/test with appropriate shuffling/sampling
- Model Architecture: Shared Backbone + Two Heads
    - Backbone: ResNet-50 pretrained on ImageNet (final FC removed → features)
    - DR Head: MLP → logits over num_dr_classes
    - ME Head: MLP → logits over 2 (No-ME / ME)
    - We initially keep early backbone layers partially frozen, then fine-tune deeper layers.
    - Two separate heads allow each task to specialize from the same shared retinal features.
    - Loss & Optim:
        - DR: CrossEntropy with class weights
        - ME: CrossEntropy (2 classes)
        - Total loss: w1 * DR_loss + w2 * ME_loss (default 1.0 each)
        - Optimizer: Adam, different LRs for backbone vs heads
        - LR scheduler: Reduce on plateau
- Training and Validation
    - Per epoch:
        - Train pass: forward → compute both losses → backprop → optimizer step
        - Validation pass: compute losses & accuracies (no gradient)
        - Track history; save checkpoint when combined val score improves
            - Combined score to keep balanced progress from both tasks

- Learning Curves and Diagnostics
    - Plots loss/accuracy curves for each head
    - Generates confusion matrices and classification reports (precision/recall/F1) for validation
- Test Set Evaluation
    - Loads best checkpoint
    - Runs inference on the test split
    - Reports accuracies for both tasks
    - Saves a CSV with per-image predictions (both heads) mapped back to original grade values
- Inference on a Single Image
    - Applies validation transforms
    - Returns original DR grade value and ME risk (0/1)
    - For demo purposes
- Save Final Model
- Grad-CAM for the Two-Head Model
    - Visualize where the model looks for each head
    - Confirms whether or not CNN is focusing on retinal lesions
        - Essential for medical AI explainability and debugging