Deep Learning-Based Glaucoma Detection Using Convolutional Neural Networks

Submitted in partial fulfilment of the requirements of the degree of
Bachelor of Technology by

Chekka Leela Prasanthi – 21ECB0A11

Potnuri Sri Anjali Pravallika – 21ECB0A45

Supervisor (s):

Prof. K. V. Sridhar

Department of Electronics and Communication Engineering

NATIONAL INSTITUTE OF TECHNOLOGY, WARANGAL

2024-2025

Date: 28th November 2024

Table of Contents:

I. Abstract

This study develops a deep learning model to detect glaucoma using images of the eye. The model was trained with images focused on specific areas of the eye (regions of interest, or ROIs) and used data augmentation techniques, such as flipping the images, to improve performance.

The model has multiple layers that analyze the images, ending with a decision layer that classifies the images as either showing glaucoma or not. The implementation was tested on both CPU and GPU environments, and the time taken for training and inference was compared. The GPU significantly reduced the time required, making it more suitable for handling large datasets.

The model was trained for 30 epochs using the Adam optimizer and a special loss function called Focal Loss to help deal with imbalanced data. After training, the model showed results with a low loss value (0.0592) and accuracy (97.95%) on the training data. When tested on new data, it achieved a precision of 98.10%, recall of 96.88%, and an overall accuracy of 98.05%. These results demonstrate that the model can accurately detect glaucoma, making it a useful tool for automatic detection in medical imaging.

II. Introduction

Glaucoma is one of the leading causes of blindness globally, affecting millions of individuals, with many of them unaware of the condition due to its gradual and asymptomatic progression. Early diagnosis is crucial in preventing vision loss, but traditional diagnostic methods, such as measuring intraocular pressure (IOP) and examining the optic disc, are often costly, labour-intensive, and require specialized expertise. This presents a significant challenge in areas with limited access to healthcare professionals and adequate medical infrastructure.

Recent advancements in deep learning, particularly Convolutional Neural Networks (CNNs), have shown great promise in automating the detection of glaucoma from retinal fundus images. CNNs are well-suited for medical image analysis, as they can automatically learn and extract complex features from images, enabling accurate classification. This approach offers a potential solution to the problem of glaucoma detection by providing a more efficient, cost-effective, and accessible alternative to traditional methods.

This study focuses on the development of an automated glaucoma detection system using a CNN model to classify retinal fundus images as either indicating the presence or absence of glaucoma. The system is designed to assist in the early detection of glaucoma, especially in regions with limited access to healthcare professionals.

To evaluate the performance and feasibility of the proposed model, it was developed and tested in various computational environments. The model was trained and tested on a CPU, a GPU with CUDA support, and on cloud-based environments using GPUs available through Kaggle and Google Colab. The training and testing times for each environment were compared to assess the efficiency of the model under different hardware configurations. The results demonstrated the advantages of using GPUs, particularly in reducing the training time, compared to CPU-based execution.

The key contributions of this study include:
Development of a CNN-based model for the detection of glaucoma from retinal fundus images.
Testing and comparison of the model's performance across different computational environments (CPU, GPU with CUDA, Kaggle GPU, and Colab GPU), with a focus on training and testing time optimization.

Fig. 1. Glaucoma Picture

III. Literature Survey

A study used a Convolutional Neural Network (CNN) to detect glaucoma by analyzing features from retinal images. The system extracted features from raw image pixels using CNNs, selected important features with a Deep-Belief Network (DBN), and classified them using a softmax classifier. The system achieved 94% accuracy, 98.01% specificity, and 84.5% sensitivity. This approach demonstrated that CNNs are effective in detecting glaucoma in retinal images [1].

Another study used both logistic regression and CNNs to diagnose glaucoma using a dataset of 1,542 retinal images, including normal, early glaucoma, and advanced glaucoma patients. The dataset was split into training, validation, and test sets. The CNN model outperformed logistic regression, showing that deep learning methods are reliable for diagnosing glaucoma [2].

A novel method called M-LAP (Multi-Scale Localized Activation Pattern) was proposed to make CNNs more interpretable for glaucoma diagnosis. The method highlighted areas of retinal images that are affected by glaucoma, improving the interpretability of deep learning models. It performed well on a challenging dataset and localized glaucoma-affected areas, improving diagnosis accuracy [3].

A deformable model was developed for detecting the optic disc and cup boundaries in retinal images, addressing challenges like blood vessel occlusion. The method improved the original snake model by adding clustering and smoothing updates. It achieved a success rate of 94% and showed good performance in detecting the cup-to-disc ratio, which is useful for clinical applications [4].

IV. Methodology

The methodology for developing the glaucoma detection model involves a structured approach aimed at accurately classifying eye images as either showing glaucoma or not. The process begins with preparing the dataset, including preprocessing images and extracting regions of interest (ROI) critical for glaucoma detection, such as the optic disc. This is followed by employing data augmentation techniques to enhance model robustness and diversity.

A deep learning model, specifically a Convolutional Neural Network (CNN), is then designed to analyze the images through multiple layers, capturing essential features for classification. The model is trained using advanced techniques, including the Adam optimizer and Focal Loss, to handle class imbalances effectively. Finally, the model's performance is evaluated using metrics such as precision, recall, and accuracy.

A) Data Collection and Organization Collection

For this project, over 13,000 images were collected from multiple datasets to build a reliable glaucoma detection model. These datasets include different types of images, such as Optical Coherence Tomography (OCT) and fundus images, which are important for diagnosing glaucoma. The datasets used are:

- Glaucoma OCT Scans (Origa) Dataset
  Includes OCT images that show the optic nerve area, which is key for detecting glaucoma.
- Glaucoma Fundus Imaging Datasets
  Provides clear images of the back of the eye (fundus), showing details of the optic disc and retina.
- Fundus Glaucoma Detection Data [PyTorch Format]
  A dataset already prepared for use in PyTorch, making it easier for training the model.
- Glaucoma Dataset: EyePACS AIROGS - Light
  Contains well-organized fundus images with labels to identify glaucoma cases.
- G1020 Final Glaucoma Dataset
  A large and detailed dataset with labeled images, useful for training and testing the model.

Organization

Images from all datasets were combined and standardized to make them consistent in quality and format. The images were divided into training, and testing sets to evaluate the model.

B) Preprocessing and Region of Interest Extraction

The preprocessing begins with loading the input image using OpenCV, followed by converting it to grayscale to simplify the data by focusing only on intensity values, reducing computational complexity. A Gaussian blur is then applied to the grayscale image using a kernel size of 65x65 to remove noise and smooth the image, making it easier to identify key features. The optic disc is focused on because it contains key indicators of glaucoma, such as changes in size, shape, or cupping, which are crucial for early detection.

To locate the Region of Interest (ROI), the pixel with the highest intensity value is identified in the blurred image, which typically corresponds to the optic disc. Using this pixel as the center, a circular ROI is defined with a radius of 100 pixels. A mask is created in the shape of this circle, and the original image is masked to extract the ROI. This process isolates the optic disc, ensuring that only the most relevant part of the image is retained for further analysis.

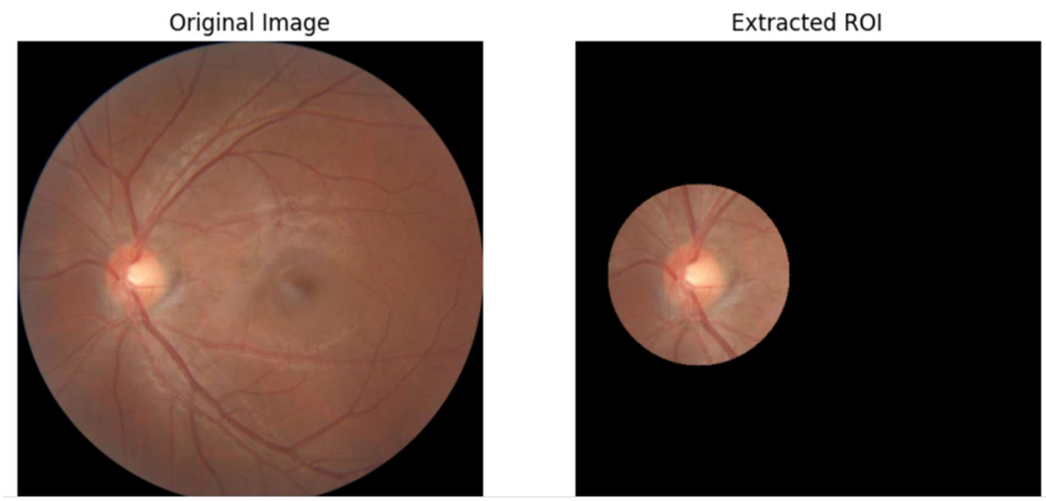

Fig. 2. Represents Original and Extracted ROI while Preprocessing

C) Data Augmentation

The ImageDataGenerator is used to augment the training images in the following ways:

1. **Rescaling**: The pixel values of the images are rescaled by dividing them by 255 (using rescale=1./255) to normalize the image data to a range of [0, 1].
2. **Horizontal Flip**: Random horizontal flipping of images is applied to make the model more robust to variations in image orientation.
3. **Vertical Flip**: Random vertical flipping is also applied, which helps the model generalize better to different image orientations.
4. **Fill Mode**: The fill mode is used to handle any gaps caused by the flip transformations by filling the empty regions with the nearest pixel value.

   These augmentations help the model generalize better, by creating more varied images for the model to learn from.

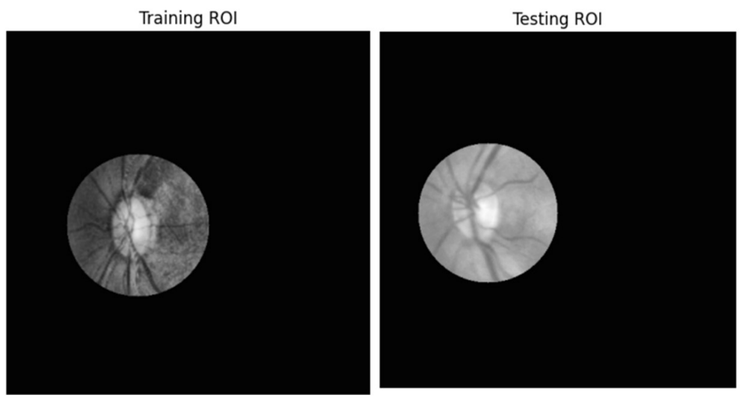

Fig. 3. Represents Training and Testing ROI

D) Model Architecture and Training Process

The architecture of the model is a Convolutional Neural Network (CNN) designed for binary classification (glaucoma vs non-glaucoma). The model consists of the following layers and is shown in figure 1.

1. Conv2D Layers:
   - Three Conv2D layers are used, each with increasing filters (64, 128, 128) and a kernel size of (3, 3). These layers help extract important features from the images.
   - Activation function: ReLU is used in all convolutional layers, allowing the network to model complex patterns by introducing non-linearity.
2. MaxPooling2D Layers:
   - After each convolutional layer, MaxPooling2D layers with a pool size of (2, 2) are applied. This reduces the spatial dimensions of the feature maps and helps in downsampling the image while preserving important features.
3. Flatten Layer:
   - The output from the final convolutional layer is flattened into a 1D array using the Flatten layer, which prepares the data for the fully connected layers.
4. Dense Layers:
   - A Dense layer with 512 neurons and ReLU activation is used to further process the features extracted by the convolutional layers.
   - The final Dense layer has 1 neuron with a sigmoid activation function, as the task is binary classification (glaucoma or not). The sigmoid activation outputs a probability between 0 and 1, which is then used to classify the image.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 254, 254, 64)      1792

 max_pooling2d (MaxPooling2D  (None, 127, 127, 64)     0
 )

 conv2d_1 (Conv2D)           (None, 125, 125, 128)     73856

 max_pooling2d_1 (MaxPooling  (None, 62, 62, 128)      0
 2D)

 conv2d_2 (Conv2D)           (None, 60, 60, 128)       147584

 max_pooling2d_2 (MaxPooling  (None, 30, 30, 128)      0
 2D)

 flatten (Flatten)           (None, 115200)            0

 dense (Dense)               (None, 512)               58982912

 dense_1 (Dense)             (None, 1)                 513

=================================================================
Total params: 59,206,657
Trainable params: 59,206,657
Non-trainable params: 0
```

Fig. 4. Model Architecture

The training process for the glaucoma detection model uses the Adam optimizer with a learning rate of $1\times10^{(-4)}$, which is suitable for image data due to its adaptive learning rate. The binary crossentropy loss function is used, as it is ideal for binary classification tasks like distinguishing between glaucoma and non-glaucoma.

To evaluate the model, several metrics are used, including binary accuracy to measure the overall correctness of predictions, AUC (Area Under Curve) to assess the model's ability to distinguish between classes, precision and recall to evaluate the model's performance in identifying true positives. To prevent overfitting, EarlyStopping is used to halt training if validation loss does not improve for 3 consecutive epochs, and ReduceLROnPlateau reduces the learning rate by a factor of 0.1 if the validation loss plateaus for 2 epochs, aiding in better convergence.

The model is trained for 30 epochs using data generators that provide the training and validation data, with data augmentation techniques applied. After training, the model is evaluated on the test set to determine its generalization capability, with the final performance measured by the accuracy score.

The model training and testing process utilizes the NVIDIA MX350 GPU in a CUDA-enabled environment to accelerate the computations. The CUDA (Compute Unified Device Architecture) framework allows the model to take advantage of the GPU's parallel processing capabilities, significantly speeding up the training process compared to using a CPU. The GPU handles the intensive operations such as matrix multiplications, convolutions, and other deep learning tasks, allowing the model to process the image data more efficiently and reducing training time.

V. Results

The model demonstrates strong performance, achieving a low training loss of 0.0592 and a high training accuracy of 97.95%. It also shows excellent ability to distinguish between classes, with an AUC of 0.9972. Precision and recall are both impressive at 98.23% and 97.75%, respectively, indicating the model's effectiveness in identifying glaucoma cases with minimal false positives and negatives.

While the validation loss (0.0699) is slightly higher than the training loss, the validation accuracy remains high at 97.74% as shown in figure 2, suggesting that the model generalizes well to unseen data.

```
loss: 0.0592 - binary_accuracy: 0.9795 - auc_1: 0.9972 - precision_1: 0.9823 - recall_1: 0.9775
- val_loss: 0.0699 - val_binary_accuracy: 0.9774 - val_auc_1: 0.9664 - val_precision_1: 0.9824
- val_recall_1: 0.9583 - lr: 1.0000e-04
```

Fig. 5. Model Performance Results

```
(py310) C:\Users\prane\Downloads\Glaucoma_Detection>python prediction_glaucoma.py
Num GPUs Available:  1
2024-11-05 21:49:24.251085: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep
Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-11-05 21:49:24.745702: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1616] Created device
/job:localhost/replica:0/task:0/device:GPU:0 with 1314 MB memory:  -> device: 0, name: NVIDIA GeForce MX350, pci bus id: 0000:01:00.0,
compute capability: 6.1
Model is loaded
2024-11-05 21:49:25.768652: I tensorflow/stream_executor/cuda/cuda_dnn.cc:384] Loaded cuDNN version 8907
2024-11-05 21:49:25.989614: W tensorflow/core/common_runtime/bfc_allocator.cc:290] Allocator (GPU_0_bfc) ran out of memory trying to
allocate 1.04GiB with freed_by_count=0. The caller indicates that this is not a failure, but this may mean that there could be performance
gains if more memory were available.
```

```
1/1 [==============================]
0.0
Prediction: No Glaucoma
```

Fig. 6. Glaucoma Detection

	CPU	NVIDIA GeForce MX350
Preprocessing of Images	35min	15min
Training of Model	7.5 min for each epoch	5min (time decreased for each epoch)
Evaluation	4min	2min

Table I: Timing Comparison of various tasks of different hardware platforms

VI. Conclusion

This project shows that using Convolutional Neural Networks (CNN) for glaucoma detection in medical images can improve diagnosis accuracy. By applying a custom preprocessing and data augmentation, the model was able to handle issues like class imbalance and image noise. The model performed well in identifying glaucoma, with strong results in accuracy, AUC, precision, and recall. Using a GPU for training made the process faster. With more improvements, it could reduce the need for repeated tests and support better decision-making in healthcare.

VII. References

[1] Abbas, Q., 2017. Glaucoma-deep: detection of glaucoma eye disease on retinal fundus images using deep learning. Int J Adv Comput Sci Appl 8, 41–5.
[2] Ahn, J.M., Kim, S., Ahn, K.S., Cho, S.H., Lee, K.B., Kim, U.S., 2018. A deep learning model for the detection of both advanced and early glaucoma using fundus photography. PloS one 13, e0207982.
[3] Gupta, K., Goldbaum, M., Yousefi, S., 2021. Glaucoma precognition based on confocal scanning laser ophthal moscopy images of the optic disc using convolutional neural network, in: Proceedings of the IEEE/CVF Con ference on Computer Vision and Pattern Recognition, pp. 2259–2267.
[4] Garc´ ıa, G., Colomer, A., Naranjo, V., 2021. Glaucoma detection from raw sd-oct volumes: A novel approach focused on spatial dependencies. Computer Methods and Programs in Biomedicine 200, 105855.