SSRG International Journal of Computer Science and Engineering
ISSN: 2348–8387 / https://doi.org/10.14445/23488387/IJCSE-V11I11P103

Volume 11 Issue 11, 20-24, November 2024

Original Article

A Deep Learning Augmented System for Automated Diagnosis of Hypertensive Retinopathy using Retinal Fundus Images

Sarthak Ahuja

Pathways School Gurgaon, India.

Corresponding Author : sarthak.ahuja231@gmail.com

Received: 28 September 2024 Revised: 27 October 2024 Accepted: 12 November 2024 Published: 30 November 2024

Abstract *- Hypertensive Retinopathy (HR) is a common, vision-threatening complication of high blood pressure. A 2023 study in India revealed a 49.33% prevalence of HR among hypertensive individuals, with 81% unaware of their condition. These figures highlight the urgent need for accessible diagnostic tools, especially since early HR signs are often invisible, making timely diagnosis difficult—particularly in resource-limited settings without specialist care. Fundus imaging is the gold standard for HR diagnosis but is often labor-intensive, inaccessible, and time-consuming. To address this, a dataset of 3,300 digitized fundus images (from both HR-diagnosed and HR-free patients) was used to train a Convolutional Neural Network (CNN) model based on YoloV8. The images were split into training (2,400), validation (450), and test (450) sets. Image processing techniques, including CLAHE and Otsu Thresholding, were applied to enhance accuracy. The CNN model achieved training, validation, and test accuracies of 95%-98%, with AUROC scores ranging from 0.96-0.99. An ML-based web app for real-time HR detection was also developed, and internal/external validation demonstrated 95%-98% accuracy. The current study demonstrated the potential of machine learning models in aiding early detection of hypertensive retinopathy in resource-limited settings, offering a valuable tool to support ophthalmologists and pathologists in clinical decision-making.*

Keywords *- Hypertensive Retinopathy, Fundus images, Machine Learning (ML)-based web app, CNN model.*

1. Introduction

Hypertensive Retinopathy (HR) is one of the most threatening complications of hypertension (high blood pressure). A 2023 regional study [1] in India found that 49.33% of hypertensive individuals suffered from HR, with 81% of cases previously undiagnosed, highlighting the urgent need for alleviating the burden of screening HR. HR is associated with poor outcomes due to a lack of methods for early detection, accurate prognostication, and personalized treatments. Fundus Imaging has a well-established role in the diagnosis and prognosis of HR. Identifying clinic features such as arteriolar narrowing, hemorrhages, cotton wool spots, and optic disc changes is a prerequisite for prognostication, individualised treatment, and early detection. Manual Fundus Analysis is the standard procedure for the diagnosis of HR. Manual Fundus Analysis of HR requires labor-intensive retina examinations with multiple consultations with other ophthalmologists and eye specialists. This is a highly strenuous, time-consuming, and cumbersome task. Given the high screening burden, there is an immediate need to adapt to methods of computational ophthalmology. Digitization of screening processes and Artificial Intelligence (AI)-based analysis can impart objectivity, increase accuracy, and reduce inter-observer variability, a crucial step toward personalized medicine. A study by B K Triwijoyo and Y D Pradipto [2] emphasized the efficacy of the application of deep learning models, particularly Convolutional Neural Networks (CNNs), for the diagnosis of diabetic retinopathy, which is pathophysiologically similar to hypertensive retinopathy in terms of retinal vascular changes. The scarce literature on AI modeling on HR ophthalmology images includes a proposal [3] for a model for determining HR presence. However, such a proposal lacks crucial image optimization techniques, resulting in inaccurate diagnostics.

The clinical features in the images are not clear enough, leading to unreliable results. For instance, a study by Supriya Suman et al. [8] yielded an accuracy of 76.25%. Additionally, existing solutions have not addressed the imperative need for real-time feedback and comprehensive reporting, limiting their practical utility in busy clinical environments requiring rapid screening for many hypertensive patients. This research hypothesizes that a pattern-recognizing ML model, created using digital ophthalmology images and its usage through a web app, can assist ophthalmologists and non-specialists in detecting HR in large volumes of digitized images with high

accuracy and objectivity in a shorter time, thus helping in prognostication & timely administration of treatments. Therefore, the present work aims to diagnose an ocular complication of hypertension (hypertensive retinopathy) by ML-augmented digital ophthalmology image analysis. The main objectives were to develop a CNN model with a wide range of image optimization techniques and to create a mobile software that uses the ML model for real-time risk stratification of Hypertensive Retinopathy signs in digitized fundus images.

2. Methods

2.1. Data

Digital fundus images were obtained from a pool of hypertensive patients (n=665) [4], comprising both healthy individuals and those diagnosed with Hypertensive Retinopathy (HR). The images were sourced from clinical evaluations (A database from Shanghai Sixth People's Hospital), and patient details were maintained anonymously. All images were de-identified and randomly numbered for analysis.

The fundus images captured various clinical features of HR, including signs such as arteriolar narrowing, retinal hemorrhages, and cotton wool spots commonly associated with hypertensive damage to the retina. These images represent different stages of the disease, ranging from mild to severe manifestations. The dataset provides a comprehensive view of retinal health in hypertensive individuals, serving as a foundation for developing a CNN model to detect and classify HR (Figure 1).

2.2. Procedure

Initially, 665 digitized fundus images were collected. These images were then augmented to expand the dataset to 3,300 images. 'Augmentation' involves techniques such as rotation, flipping, zooming, and cropping to artificially increase the variety of the dataset, thereby improving the model's ability to generalize to new, unseen images [5]. The augmented dataset was bifurcated into two groups: 2,600 images of healthy eyes (eyes without HR) and 700 images of eyes diagnosed with Hypertensive Retinopathy (HR). Next, the images underwent preprocessing, where they were first resized, converted to grayscale, and then enhanced using Contrast Limited Adaptive Histogram Equalization (CLAHE) and Otsu's thresholding to improve image contrast and clarity. CLAHE is a technique that mainly increases the contrast in the image, while OTSU segments the foreground and background. A study underscores the efficacy of employing CLAHE in breast cancer detection [6]. The dataset was then split into training (70%) and testing (30%) sets, ensuring the model had ample data to learn from during training and enough data for accurate validation and performance testing. A web application was then developed with Streamlit, a Python framework. The software is divided into two major screens: Home and Diagnosis.

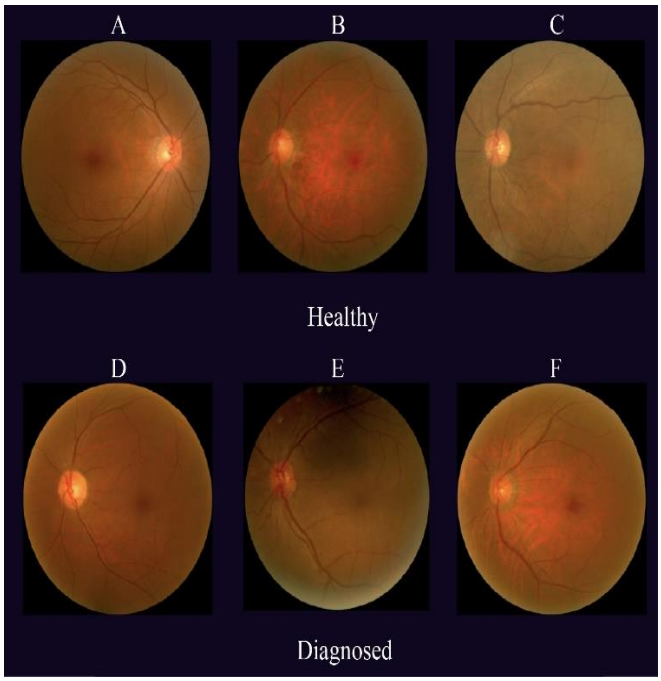

Fig. 1 Digital fundus images reminiscent of HR (D, E, F) patterns and non-HR patterns (A, B, C)

While the 'home' page imparts basic information about the disease, the 'diagnosis' page is where the algorithms take over. The user is expected to give an image from the gallery (the camera option was decided not to be given due to privacy laws) alongside other basic information as input, after which the model runs to give an optimized image output and a personalized risk-stratification report.

It is important to note that the application does not directly allow users to open the camera due to patient privacy laws. Hence, it is expected to upload an image from the camera roll or gallery. Confusion matrices and AUROC curves were created based on test data predictions for the model.

Apart from that, internal validation was performed by using random images from the original dataset of images and evaluating the results given by the model. External validation was performed by renowned ophthalmologists and retina specialists who used their images to validate the efficacy and accuracy of the model.

3. Developing the Model

The first is a code that runs through a process of augmenting images using TensorFlow's Keras `ImageDataGenerator`. With various augmentation parameters set like rotation, shear, zoom, flip, channel shift, and brightness range, the script artificially increases the size of the dataset. Here, `ImageDataGenerator` applies defined transformations to images, generates new augmented images in batches 32, and saves them to a specified directory.

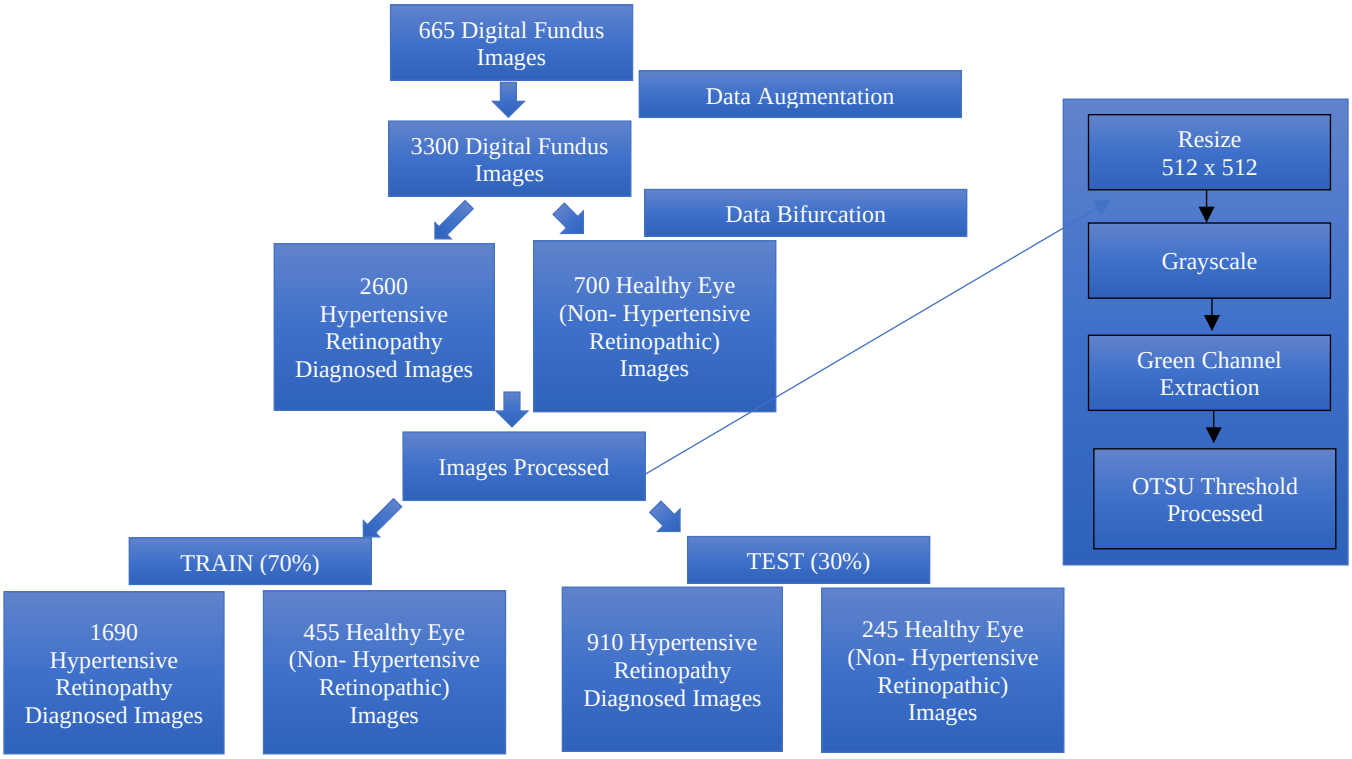

Fig. 2 Data split for training the model and optimizing images using image processing

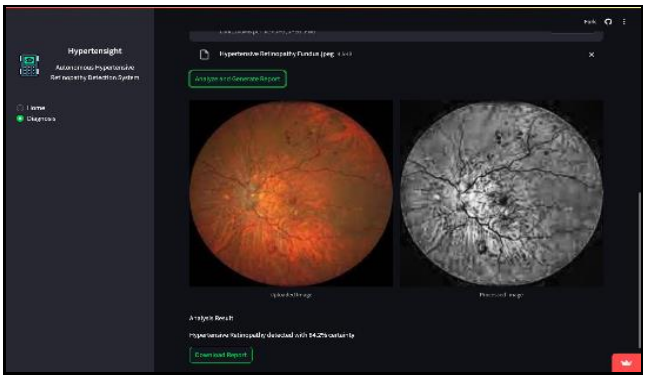

Fig. 3 Processed image on the web app

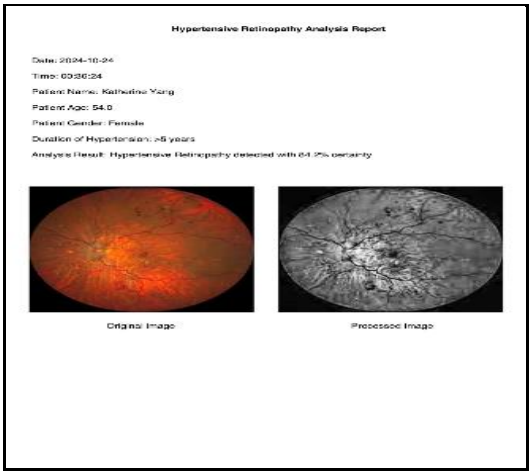

Fig. 4 Report generated via web app with diagnosis results

```
from keras import ImageDataGenerator
from skimage import io #(you will have to install this library with this command in the terminal:- pip install scikit-image)
Python

datagen = ImageDataGenerator(        # here we are creating an object for the TF-Keras class 'ImageDataGenerator' and giving
        rotation_range = 40,         #in all the trasnformation values (i am using this configuration you can alter it)
        shear_range = 0.2,          #this is the reference link for the same:- https://www.tensorflow.org/api_docs/python/tf/keras/preproces
        zoom_range = 0.2,
        horizontal_flip = True,
        vertical_flip = True,
        channel_shift_range=0.2,
        brightness_range = (0.5, 1.5))
import numpy as np  #importing the libraries
import os
from PIL import Image
image_directory = '/home/tech_lab/dataset/train/hr/' #this is the path which you will have to set for your malignant image folder.
#As you are in the windows computer it gives the path in '\' so please use double backslash- '\\' or put an r before the left qoute. only
SIZE = 224  #setting up the image size
dataset = [] #creating an empty list to store all the preprocessed images
my_images = os.listdir(image_directory) #reading the images from the folder
for i, image_name in enumerate(my_images):    #loading up the images in the loop one by one
    if (image_name.split('.')[1] == 'jpg'):
        image = io.imread(image_directory + image_name)
        image = Image.fromarray(image, 'RGB')
        image = image.resize((SIZE,SIZE))  # setting the size of the image
        dataset.append(np.array(image))
x = np.array(dataset) #final dataset storing as a numpy array
i = 0
for batch in datagen.flow(x, batch_size=32,     #applying the image augmentation changes to the images
                          save_to_dir= r'/home/tech_lab/dataset/augmented/', #here you have to write the path for you final destination fo
                          #first create a new empty folder for the augmented images. and then paste its path above.
                          save_prefix='dr',
                          save_format='jpg'):
    i += 1
    if i > 100:
        break
Python
```

Fig. 5 Data Augmentation

```
import cv2
import os

def process_images_from_folder(folder):
    clahe = cv2.createCLAHE(clipLimit=5.0, tileGridSize=(8, 8))


    for filename in os.listdir(folder):
        img_path = os.path.join(folder, filename)


        img = cv2.imread(img_path)
        img_resized = cv2.resize(img, (512, 512))
        (_, green_channel, _) = cv2.split(img_resized)
        equalized_green_channel = clahe.apply(green_channel)
        _, otsu_thresh = cv2.threshold(equalized_green_channel, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
        output_path = os.path.join(folder, f"processed_opt")
        cv2.imwrite(output_path, otsu_thresh)

        print(f"Processed and saved: /Users/sarthakahuja/Downloads")

process_images_from_folder('/Users/sarthakahuja/Downloads/newhealthy')
```

Fig. 6 Processing images

```python
from ultralytics import YOLO
import os

def load_model(model_path):
        model = YOLO(model_path)
        return model


def set_training_params(model, data_path, epochs, batch_size, img_size):
        model.train(data=data_path, epochs=30, batch=64, imgsz=224)

def initialize_and_train():

    model_path = os.path.join('/Users', 'sarthakahuja', 'Downloads', 'yolov8l-cls.pt')
    data_path = os.path.join('/Users', 'sarthakahuja', 'Downloads', 'final')

    epochs = 30
    batch_size = 64
    img_size = 224


    print("Initializing model...")
    model = load_model(model_path)

    print("Starting training process...")
    set_training_params(model, data_path, epochs, batch_size, img_size)

if __name__ == "__main__":
    initialize_and_train()
```

Fig. 7 Training the model

All the images undergo image enhancement and thresholding. The function `process_images_from_folder` reads all images in a specified folder using OpenCV (`cv2`). The function resizes every image to 512 pixels x 512 pixels resolution and then separates the color channels from it. The green channel is extracted and further processed through Contrast Limited Adaptive Histogram Equalization (CLAHE), which enhances the image's contrast so that the eye's clinical features can be clearly visible. The enhanced green channel then undergoes Otsu's thresholding, where the image is converted automatically into binary format. Lastly, each processed image is saved in the designated folder. The YOLO model was loaded and trained using the Ultralytics YOLO library. The set_training_params function configured the training parameters, such as the dataset path, number of epochs, batch size, and image size. The training process was initiated by calling the .train() method on the model. The script loaded the model at this stage and began the training process.

4. Results and Discussion

4.1. Performance Evaluation of Model:

The model demonstrated excellent performance, with an Area Under the Receiver Operating Characteristic (AUROC) score of 0.991, indicating near-perfect discrimination between HR and non-HR images. The AUROC score signifies the model's ability to accurately differentiate between healthy and diseased eyes. Additionally, the model achieved an F1 score accuracy of 0.95, highlighting a strong balance between precision and recall and further confirming its reliability in clinical settings. Further external validation was performed by an ophthalmologist with a random group of over 45. Results showed that all the participants diagnosed with HR had already been put through the system and were also diagnosed positively. Hence, this underscores the efficacy of the model. Compared to previous models, such as the one by Supriya Suman et al. [8] that yielded an accuracy of 76.25%, this model has a 95% accuracy. The confusion matrix evaluates the cases of 'false positives' and 'false negatives' that are 7 and 3, respectively. This represents a significant milestone since only 7/325 cases of images predicted as 'diagnosed' were incorrect, and only 3/87 of the images predicted as 'healthy' were wrong.

Table 1. Performance evaluation table

	precision	recall	f1-score	support
0	0.92	0.92	0.91	566
1	0.93	0.92	0.93	677
accuracy	(AUROC) 0.991		0.95	1243
macro avg	0.94	0.93	0.97	1243
weighted avg	0.93	0.94	0.94	1243

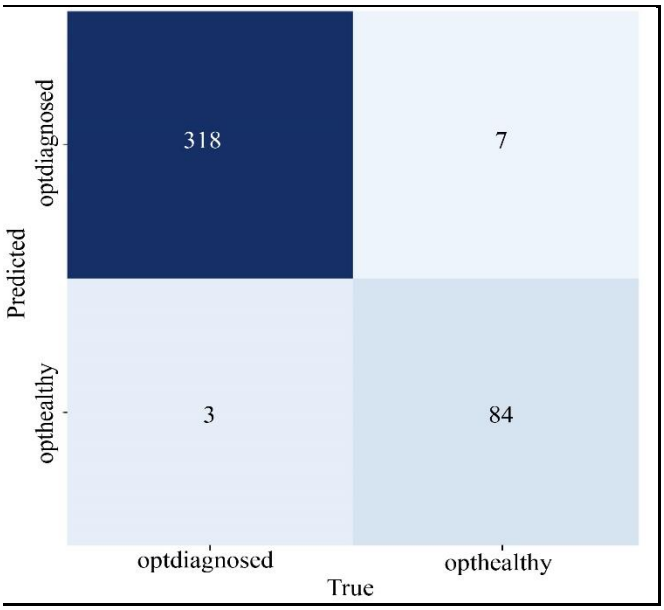

Fig. 8 Confusion matrix

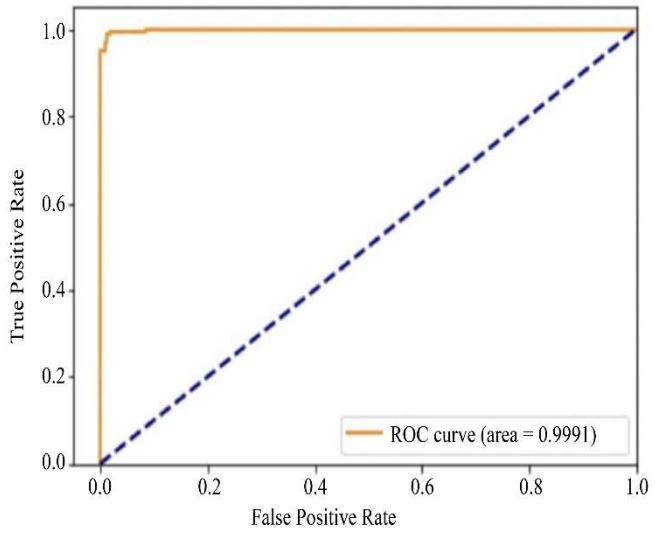

Fig. 9 AUROC Score

5. Discussion

In this study, the ML model and a mobile app were developed to identify (detect) hypertensive retinopathy (HR) from digital fundus images with high accuracy and efficiency. This model has great significance given the high global prevalence of hypertension, restricted ophthalmologist

availability, and the time intensive nature of HR diagnosis. After this, because patients are flagged as HR, the ML based detection model can be used to qualify the load of the clinicians and speed up diagnosis. We aim to combine computational ophthalmology with artificial intelligence to be useful to both ophthalmologists and pathologists, especially in resource-limited settings. This study was able to reach significantly better results owing to the image optimization techniques largely. Arteriolar narrowing, retinal hemorrhage, and cotton wool spots are characteristic morphological features of hypertensive retinopathy. Early identification of these features is important for preventing vision threatening complications.

Recently, there has been a shift toward early and accurate diagnosis because of a relationship between HR and other cardiovascular risks. A Convolutional Neural Network (CNN) was developed to classify these HR patterns as accurately as possible in this study. Further CNN architectures would be explored to improve performance, and this would be followed up with testing and validation of the model on a larger cohort of fundus images from different centers.

Moreover, a study by Pranav Modi [7] highlights the different stages of HR, which expands the scope of the study by potentially making the algorithm segregate the severity of these stages in the future. Additionally, a fundus imaging kit compatible with a smartphone would be viable in a medically underserved community where practicing physicians and non-specialists can use the software to diagnose the patient.

6. Conclusion

A CNN model is presented, a promising and accessible method for early detection of Hypertensive Retinopathy (HR) and one of the first studies to apply machine learning models to detect this ocular disease. Although the model achieves high accuracy, around 95%, and has the potential to reduce the burden of undiagnosed HR, several critical challenges remain. One problem is that the store currently contains about 5,000 fundus images. However, it is not enough, given the millions of people worldwide suffering from HR, making it very hard to train a model capable of properly serving so many people. Additionally, this limited dataset also means that more patients may have false negatives, which is a serious concern for HR, given how it can be a condition leading to irreversible blindness. However, it is important to expand the dataset to generate the model's performance. Another challenge is the variability of images, particularly with new fundus images provided by ophthalmologists globally. Scanning Laser Ophthalmoscopy (SLO) images [9], which are rectangular and differ from the typical circular images, are not represented in the current training set, occasionally leading to inaccuracies. To overcome the identified challenges, future efforts should focus on expanding the dataset and incorporating additional image formats. This enhancement will improve the model's robustness and accuracy, contributing to more reliable Hypertensive Retinopathy (HR) detection. Such improvements are critical for reducing the risk of vision loss associated with undiagnosed HR, particularly in clinical settings where early detection plays a crucial role in patient outcomes.

References

[1] Chethana Warad et al., “Incidence and Determinants of Hypertensive Retinopathy in Hypertension Patients at a Teaching Hospital in North Western Karnataka,” *Indian Journal of Clinical and Experimental Ophthalmology*, vol. 9, no. 4, pp. 634-640, 2023. [CrossRef] [Publisher Link]

[2] B.K. Triwijoyo, and Y.D. Pradipto, “Detection of Hypertension Retinopathy Using Deep Learning and Boltzmann Machines,” *Journal of Physics: Conference Series*, vol. 801, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[3] Miguel Alberto Urina-Triana et al., “Machine Learning and AI Approaches for Analyzing Diabetic and Hypertensive Retinopathy in Ocular Images: A Literature Review,” *IEEE Access*, vol. 12, pp. 54590-54607, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Pranav Modi, and Tasneem Arsiwalla, *Hypertensive Retinopathy*, StatPearls - NCBI Bookshelf, 2023. [Publisher Link]

[5] CGI-Hypertensive Retinopathy Diagnosis Challenge (CGI-HRDC2023), 2023. [Online]. Available: https://www.cgs-network.org/cgi23/cgi-hrdc2023/

[6] Jyotsana, Image Augmentation Techniques, Medium, 2023. [Online]. Available: https://medium.com/@jyotsana.cg/image-augmentation-techniques-798243f6afdf

[7] Avani Manesh et al., “Breast Cancer Detection Using CLAHE-CNN Architecture,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 11, no. 1, 2023. [CrossRef] [Publisher Link]

[8] Jörg Fischer et al., “Scanning Laser ophthalmoscopy (SLO),” *High Resolution Imaging in Microscopy and Ophthalmology*, pp. 35–57, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Supriya Suman et al., “Automated Detection of Hypertensive Retinopathy Using Few-Shot Learning,” *Biomedical Signal Processing and Control*, vol. 86, 2023. [CrossRef] [Google Scholar] [Publisher Link]