Detection of Diabetic and Hypertensive Retinopathy

19CSE305 Machine Learning

Case Study - Group 1

October 19, 2023

Abstract

Retinopathy in the context of coming from diabetes/hypertension is a complication that affects the eyes. This can affect people by causing vision problems which left unattended can cause reduced quality of an individual's life.
Thus, deep learning techniques are used for accurately detecting these diseases at an early stage and the proposed solution enables it to happen.

Introduction

Retinopathy is a condition that affects the retina, which is the part of the eye that helps you see. It is often caused by underlying health issues such as hypertension and diabetes. Symptoms generally include blurry vision, spots in vision, or vision loss, especially in advanced cases.

Hypertensive Retinopathy

- **Cause:** It is caused by high blood pressure, which puts extra pressure on the tiny blood vessels in the retina.
- **Effect on Blood Vessels:** High blood pressure can damage and weaken these blood vessels, making them narrower or causing them to leak.
- **Associated Condition:** Typically associated with high blood pressure, so controlling blood pressure is crucial to prevent or manage it.
- **Parameters:** Narrowing of the retinal arteries. White deposits may occur in the retina.

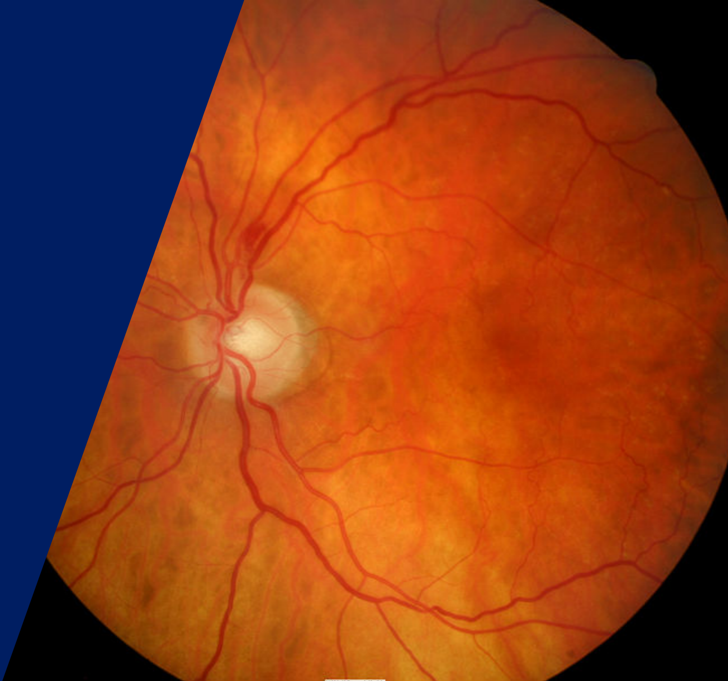
Diabetic Retinopathy

- **Cause:** It is caused by diabetes, a condition where the body has trouble regulating blood sugar levels. Over time, high blood sugar can damage the blood vessels in the retina.
- **Effect on Blood Vessels:** High blood sugar can cause the blood vessels in the retina to swell, leak, or even close off, affecting the flow of blood and nutrients to the eye.
- **Associated Condition:** Directly linked to diabetes, so managing blood sugar levels and regular eye check-ups are vital in preventing and controlling diabetic retinopathy.
- **Parameters:** Red Dots (micro aneurysms), Yellow ”flakes”, Increase in blood vessels.

Proposed Solution

- Patients with delayed detection face higher risks of coronary disease, peripheral vascular disease, and stroke. Timely identification and treatment of these conditions is crucial, but ophthalmologists encounter challenges in diagnosing them.
- Numerous deep learning solutions, including **CNN, hybrid CNN with ResNet, and hybrid CNN with DenseNet**, have been proposed to address this issue.
- These algorithms are primarily utilized for individual detection of hypertensive/diabetic retinopathy and further classification into their respective stages.
- The proposed algorithm detects both hypertensive and diabetic retinopathy based on the provided data and classifies them into their respective stages.
- We have chosen **EfficientNet** to solve our problem statement.

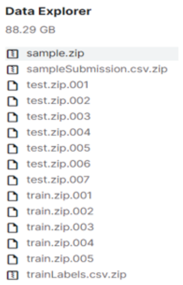
Datasets



Datasets

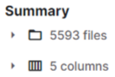
- The availability of comprehensive and diverse datasets plays a pivotal role in the development and refinement of machine learning (ML) models for the detection and management of hypertensive and diabetic retinopathy. These datasets serve as valuable resources for training, validating, and testing ML algorithms, enabling the creation of robust and accurate diagnostic tools.
- Collected from various clinical settings and research studies, these datasets typically include a wide range of retinal images capturing different stages and manifestations of hypertensive and diabetic retinopathy.
- We have collected these datasets from Kaggle.

Datasets

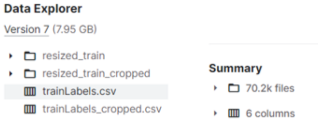


Data Explorer
88.29 GB

- sample.zip
- sampleSubmission.csv.zip
- test.zip.001
- test.zip.002
- test.zip.003
- test.zip.004
- test.zip.005
- test.zip.006
- test.zip.007
- train.zip.001
- train.zip.002
- train.zip.003
- train.zip.004
- train.zip.005
- trainLabels.csv.zip



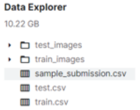
Summary
- 5593 files
- 5 columns



Data Explorer
Version 7 (7.95 GB)

- resized_train
- resized_train_cropped
- trainLabels.csv
- trainLabels_cropped.csv

Summary
- 70.2k files
- 6 columns



Data Explorer
10.22 GB

- test_images
- train_images
- sample_submission.csv
- test.csv
- train.csv

Datasets

Diabetic Retinopathy Detection 2015 Kaggle

We have a large set of high-resolution retina images taken under a variety of imaging conditions. A left and right field is provided for every subject. Images are labelled with a subject id as well as either left or right (e.g. 1_left.jpeg is the left eye of patient id 1).

- train.zip - the training set (5 files total)
- test.zip - the test set (7 files total)
- sample.zip - a small set of images to preview the full dataset
- sampleSubmission.csv - a sample submission file in the correct format
- trainLabels.csv - contains the scores for the training set

Link: https://www.kaggle.com/competitions/diabetic-retinopathy-detection/data

Datasets

APTOS 2019 Blindness Detection Kaggle

The images in the dataset come from different models and types of cameras, which can affect the visual appearance of left vs. right. Some images are shown as one would see the retina anatomically (macula on the left, optic nerve on the right for the right eye). Others are shown as one would see through a microscope condensing lens (i.e. inverted, as one sees in a typical live eye exam).

- It is inverted if the macula (the small dark central area) is slightly higher than the midline through the optic nerve. If the macula is lower than the midline of the optic nerve, it's not inverted.
- If there is a notch on the side of the image (square, triangle, or circle) then it's not inverted. If there is no notch, it's inverted.

Datasets

APTOS 2019 Blindness Detection Kaggle

- train.csv - the training labels
- test.csv - the test set (you must predict the diagnosis value for these variables)
- sample_submission.csv - a sample submission file in the correct format
- train.zip - the training set images
- test.zip - the public test set images

Link: https://www.kaggle.com/competitions/aptos2019-blindness-detection/data

Data Preprocessing

Diabetic Retinopathy (resized):

- The primary purpose of the code is to resize images to a specified width - new_width, while maintaining their aspect ratios and save the resized images to a different directory. This can be useful for preparing images for use in machine learning or deep learning models, where standardizing image sizes is often required.
- The code checks if the width of the image is greater than a specified new_width (1024 in this case). If the width is larger, it resizes the image to the specified width while maintaining the aspect ratio using the resize function. If the width is not greater than the new width, the image remains the same, and no resizing is performed. The resized or original image is then saved with the same filename in a different directory.

Data Preprocessing

2015 and 2019 dataset data preprocessing:

- A function is designed to prepare an image for use in machine learning or deep learning tasks.
- **Input Parameters:**
  - **path:** This parameter is used to specify the file path to the input image that you want to preprocess.
  - **sigmaX:** It's a parameter used for Gaussian blur. In this code, it's set to a default value of 10.
  - **do_random_crop:** This parameter, when set to True, enables random cropping of the image. By default, it's set to False.
- **Image Import and Color Conversion:** OpenCV (cv2) converts the image from BGR color space to RGB color spacer, ensuring that the image is in the correct color format for further processing.

Data Preprocessing

- **Cropping:**
  - The code applies a ”smart crop” operation to the image using the crop_black function with a tolerance (tol) of 7. This operation likely removes black borders or background around the image. If do_random_crop is set to True, it performs random cropping using the random_crop function. The size of the random crop is controlled by the argument (0.9, 1), which suggests that the cropped area will be between 90 to 100 percent of the original image’s size.
- **Resizing and Color Adjustment:**
  - The image is resized to a square shape with a side length of image_size. This step ensures that all images have the same dimensions. A color adjustment is applied by adding a weighted combination of the original image and a Gaussian-blurred version of the image. This adjustment can enhance the image’s visual features and is a common preprocessing step in image classification tasks.

Data Preprocessing

- **Circular Crop:**
  - The function applies a circular crop operation using the circle_crop function. This may be used to crop the image to a circular shape.
- **Data Type Conversion:**
  - The Image is converted to a PyTorch tensor, which is a common data format for deep learning frameworks like PyTorch. The tensor's dimensions are permuted to match the expected input format.
- **Output:**
  - The The preprocessed image is returned as the output of the function.

Data Loading and Transformation

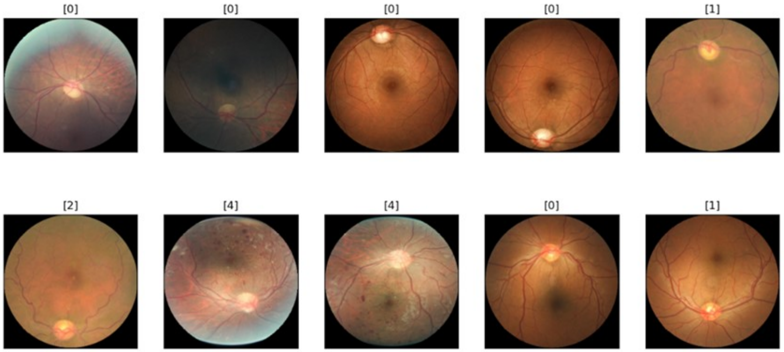
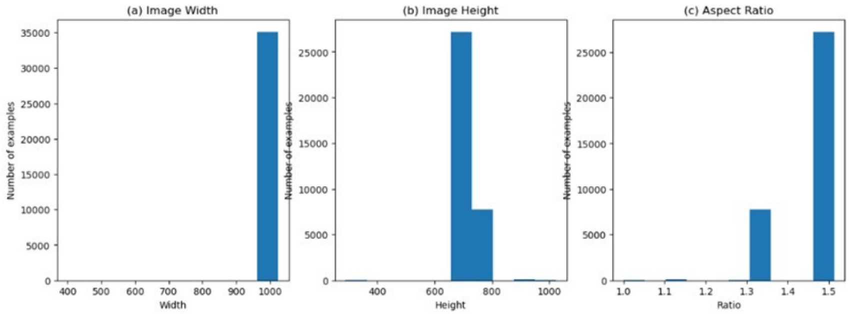
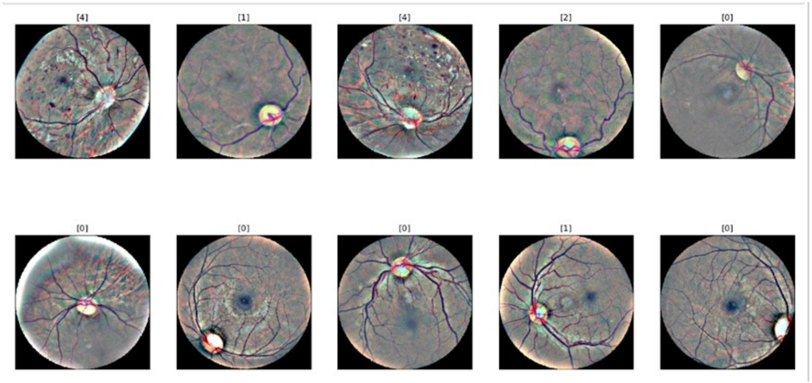


Image size distribution:

```
Out[25]: Text(0.5, 0, 'Ratio')
```



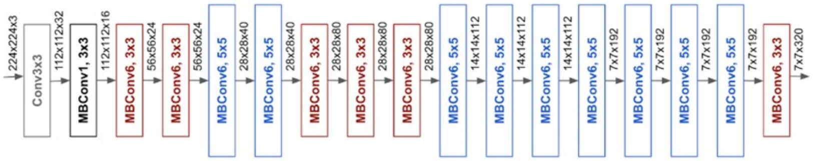
Examining first batch of training dataset:



MODEL

EfficientNet is a family of neural network architectures that have achieved state-of-the-art performance on a variety of computer vision tasks while being more computationally efficient compared to some other architectures. When using EfficientNet for transfer learning, you typically leverage a pre-trained version of the model and fine-tune it for your specific task. Pre-training the model with 2015 Dataset and using the same model to further train and test on 2019 dataset.

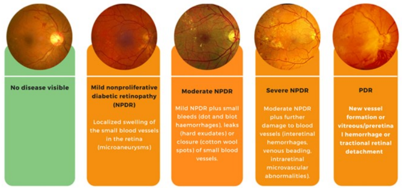
Using EFfienctNet for Transfer Learning



MODEL

Train model on Diabetic Retinopathy 2015 Data by implementing the training and evaluation loop for a deep learning model, a convolutional neural network (CNN), using PyTorch. It includes various components for training, evaluating, and potentially early stopping. The code can be designed to train a model for a classification task, monitoring its performance on a validation set, and implementing early stopping to prevent overfitting. The model is based on the EfficientNet B7 architecture and is fine-tuned for the specific task at hand.

Training on Diabetic Retinopathy 2019 data



No disease visible

Mild nonproliferative diabetic retinopathy (NPDR)

Localized swelling of the small blood vessels in the retina (microaneurysms)

Moderate NPDR

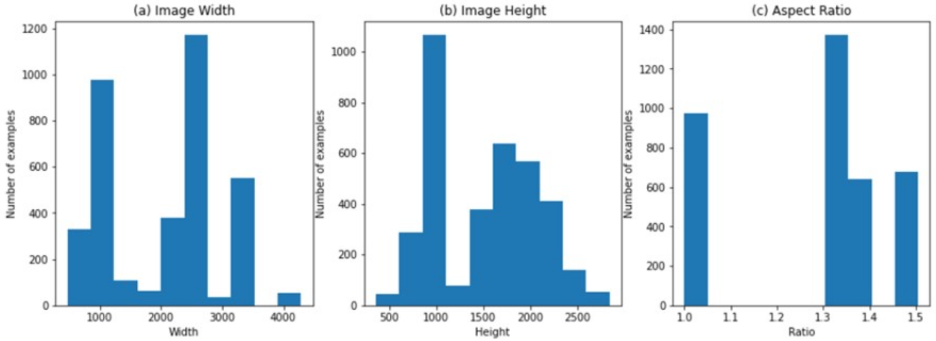
Mild NPDR plus small bleeds (dot and blot haemorrhages), leaks (hard exudates) or closure (cotton wool spots) of small blood vessels.

Severe NPDR

Moderate NPDR plus further damage to blood vessels (interetinal hemorrhages, venous beading, intraretinal microvascular abnormalities).

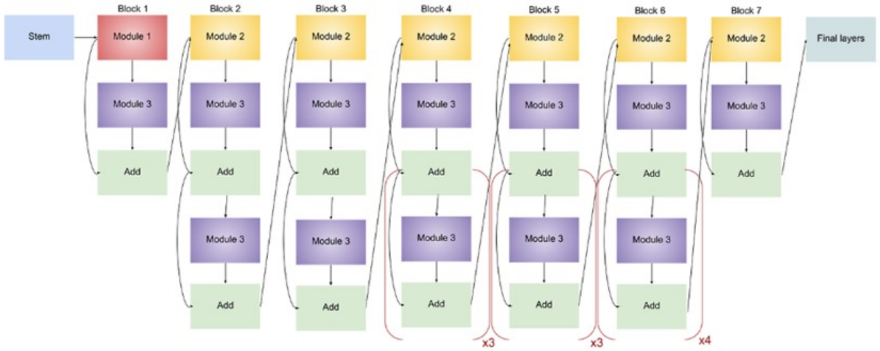
PDR

New vessel formation or vitreous/preretina l hemorrhage or tractional retinal detachment



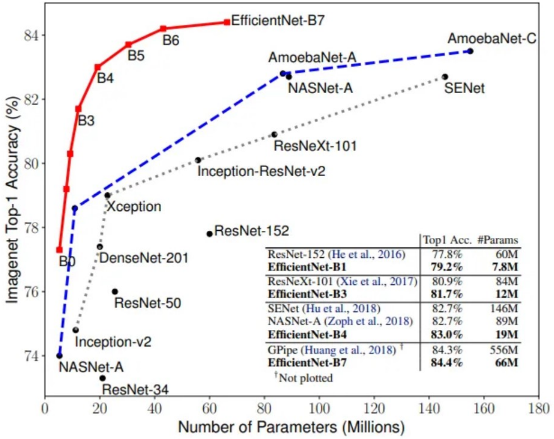
(a) Image Width
(b) Image Height
(c) Aspect Ratio
Number of examples
Width
Height
Ratio
0
200
400
600
800
1000
1200
1400
1000
2000
3000
4000
500
1500
2500
1.0
1.1
1.2
1.3
1.4
1.5

Load Pre-Trained EFfienctNetB7 trained on 2015 data



- **EfficientNetB7 has the following key characteristics:**
  - **Depth and Width:** It is very deep and wide compared to smaller variants. The number of layers and channels is significantly larger, which allows it to capture complex patterns and features in images.
  - **Parameter Size:** EfficientNetB7 has more parameters than the smaller variants. This means it has a greater capacity to learn from a large amount of data.
  - **Image Size:** EfficientNetB7 is typically trained on larger input images compared to smaller variants. This enables it to capture more fine-grained details in images.
  - **Model Scaling:** EfficientNetB7 follows a similar scaling factor as other variants, which involves increasing the depth, width, and resolution of the model while keeping the computational cost reasonable.

- **Performance:** Due to its depth and capacity, EfficientNetB7 is capable of achieving high accuracy and robust performance on a wide range of computer vision tasks, including image classification, object detection, and segmentation.
- **Transfer Learning:** EfficientNetB7 is often used as a powerful feature extractor for transfer learning. Researchers and practitioners fine-tune this model on specific tasks, leveraging the pre-trained weights and representations learned from large-scale datasets.
- **Reason:** It can be applied to various computer vision tasks, including medical image analysis, autonomous driving, and natural image classification. EfficientNetB7 requires loading a pre-trained version of the model and fine-tuning it for a specific task. The model’s strong feature extraction capabilities help in adapting itself to a unique dataset and problem.



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.2%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.7%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**83.0%**	**19M**
GPipe (Huang et al., 2018) [†]	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

[†]Not plotted

Thank you!