

**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



## **BÁO CÁO ĐỒ ÁN CUỐI KỲ MÔN CS406 VÀ CS231**

**Đề tài: Diabetic Retinopathy Detection**

**Giáo viên hướng dẫn: Mai Tiến Dũng**

Sinh viên thực hiện:		
STT	Họ tên	MSSV
1	Trương Xuân Linh	19521759
2	Nguyễn Minh Trí	19522389

**TP. HỒ CHÍ MINH – 12/2021**

# MỤC LỤC

<b>I. MÔ TẢ ĐỀ TÀI .....</b>	<b>1</b>
1. Giới thiệu .....	1
2. Data.....	1
<b>II. XỬ LÝ ẢNH.....</b>	<b>4</b>
1. Crop .....	4
2. Btgraham-300 : .....	4
3. CLAHE.....	6
<b>III. HUẤN LUYỆN MÔ HÌNH.....</b>	<b>9</b>
1. Phương pháp đánh giá .....	9
2. Tinh chỉnh mô hình và các tham số.....	9
3. Hướng tiếp cận Imbalance data .....	10
4. Hướng tiếp cận Balance data:.....	11
5. Hướng tiếp cận Multi-class: .....	12
6. Hướng tiếp cận Imbalance multi-model : .....	13
7. Hướng tiếp cận Balance multi-model:.....	15
<b>IV. NHẬN XÉT .....</b>	<b>17</b>
<b>V. ỨNG DỤNG VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>19</b>
1. Ứng dụng .....	19
2. Hướng phát triển.....	19
<b>VI. BẢNG PHÂN CÔNG CÔNG VIỆC .....</b>	<b>21</b>
<b>VII. THAM KHẢO .....</b>	<b>22</b>

## I. MÔ TẢ ĐỀ TÀI

### 1. Giới thiệu

Võng mạc do tiểu đường là bệnh mắt có vấn đề và có thể gây mù. Điều này xảy ra khi lượng đường trong máu cao làm hỏng các mạch máu nhỏ ở phía sau mắt, được gọi là võng mạc. Tất cả những ai bị tiểu đường đều có nguy cơ mắc phải vấn đề này.

Võng mạc do tiểu đường ảnh hưởng đến cả hai mắt. Sẽ không có dấu hiệu nào báo trước. Chỉ khi bệnh nặng hơn, mạch máu yếu đi rồi chảy máu và dịch. Mạch máu mới khi phát triển cũng bị chảy máu và dịch. Tình trạng này khiến tầm nhìn của quý vị bị che khuất.

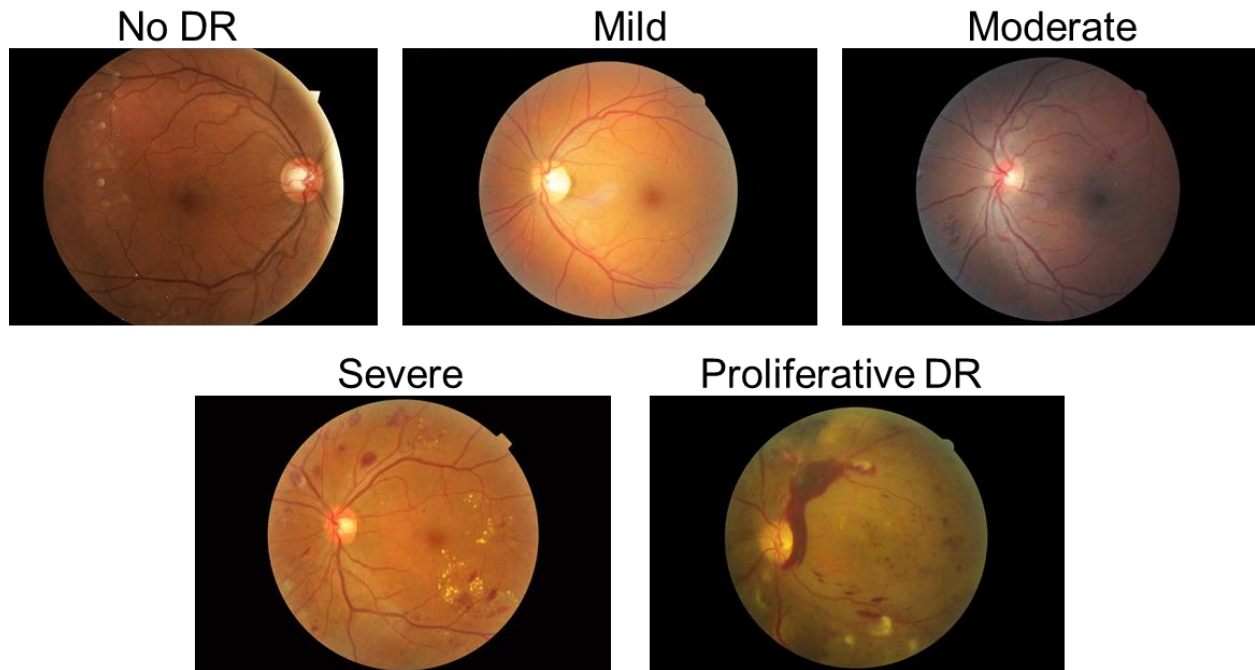
Theo thống kê đến năm 2015 của WHO, có khoảng 422 triệu người mắc bệnh tiểu đường. Trong đó, 148 triệu người chiếm 35% có những biến chứng về mắt với các mức độ khác nhau và 46 triệu người chiếm 11% có nguy cơ bị mù lòa do những tổn thương lâu ngày như bong võng mạc, xuất huyết võng mạc. Mỗi năm ước tính có hơn 12% bệnh nhân mới bị mù do tiểu đường, người bệnh tiểu đường có nguy cơ bị mù lòa cao gấp 25 lần so với người bình thường. Tuy nhiên, nếu được phát hiện và điều trị kịp thời có thể tránh được những nguy cơ trên.

### 2. Data

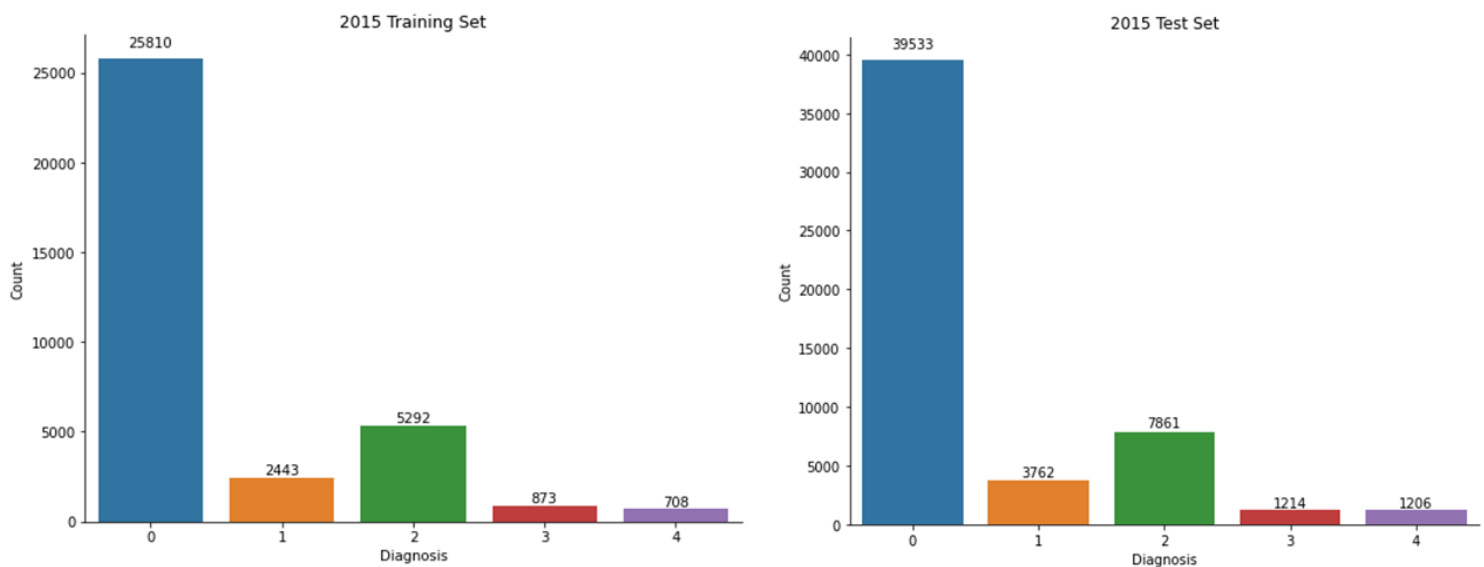
Để thực hiện đồ án trên, chúng em sử dụng bộ dữ liệu EyePACS2015, chi tiết bộ dữ liệu gồm:

- 88702 ảnh chụp võng mạc với nhiều kích thước ảnh khác nhau được chia làm 2 tập dữ liệu:
  - Train set với 35126 ảnh. Trong đó, chúng em chia 85% cho phần training và 15% cho phần validation
  - Test set với 53576 ảnh
- Được chia làm 5 classes gồm:

○ No DR:	Không mắc bệnh
○ Mild:	Mắc bệnh nhẹ
○ Moderate:	Mắc bệnh vừa
○ Severe:	Mắc bệnh nặng
○ Proliferative DR:	Tăng sinh



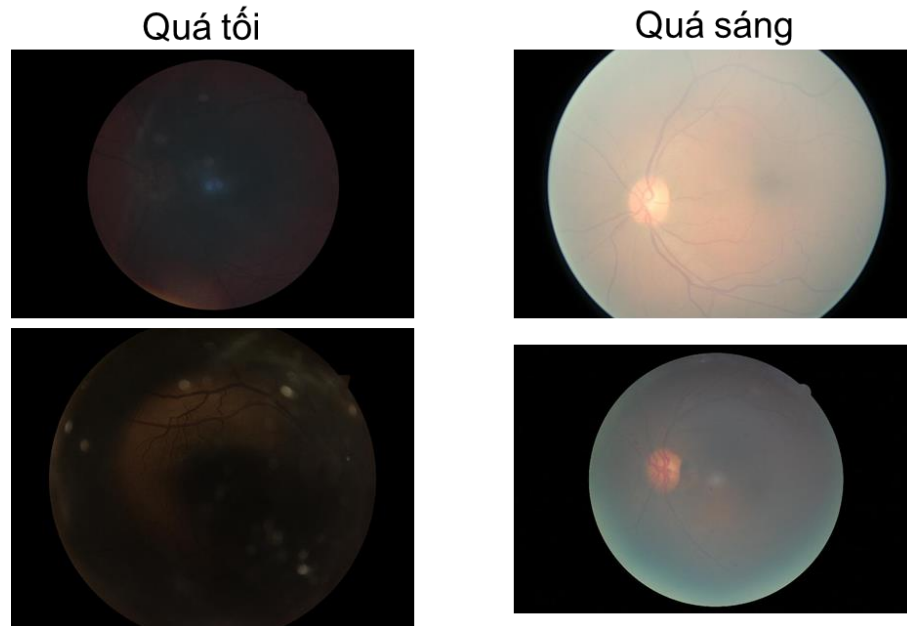
Hình 1.1: Một số ảnh trong tập dữ liệu



Hình 1.2: Biểu đồ phân phối dữ liệu training và testing

Qua 2 biểu đồ phân phối như hình 2 ở trên, ta có thể thấy được bộ dữ liệu đang bị mất cân bằng nghiêm trọng. Cụ thể số lượng ảnh ở class No DR - Không mắc bệnh chiếm số lượng lớn và chênh lệch nhiều so với các class còn lại

Bên cạnh đó, bộ data còn có một số hình ảnh do tác động của môi trường dẫn đến ảnh bị quá sáng hoặc quá tối khiến cho kẻ cả mắt thường cũng khó có khả năng nhìn rõ các mạch máu cũng như các chi tiết khác như hình dưới đây



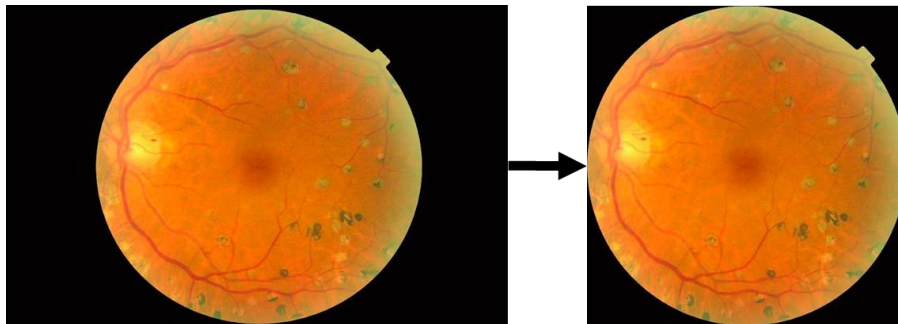
*Hình 1.3: Một số hình ảnh khó nhận diện trong bộ dữ liệu*

## II. XỬ LÝ ẢNH

### 1. Crop

Bởi vì dataset được tổng hợp ở nhiều phòng khám với nhiều điều kiện chụp khác nhau nên trước khi đưa vào training, chúng em quyết định cắt đi phần viền đen không đồng đều ở các ảnh khá thừa thãi và không mang thông tin giúp cho quá trình training cũng như là nhận diện ra bệnh.

Ý tưởng chính của phần crop đơn giản là duyệt từ trên xuống, từ dưới lên, từ trái qua phải và từ phải qua trái ở cả 3 kênh màu, cho đến khi gặp pixel có giá trị khác 0 thì thực hiện cắt tại đó.



Hình 2.1: Ảnh gốc(trái) sau khi được crop(phải)

### 2. Btgraham-300 :

Btgraham là một phương pháp xử lý ảnh được đề xuất trong một bài báo[1]. Ý tưởng của phương pháp này là làm mờ đi những chi tiết mạch máu và đốm bệnh ở ảnh gốc, sau đó lấy ảnh gốc trừ cho ảnh đã làm mờ thì những chi tiết đó sẽ được thể hiện rõ hơn.

Các bước thực hiện :

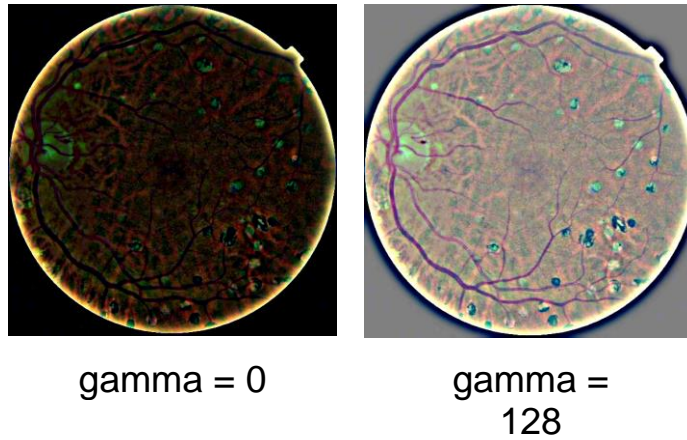
- Bước 1 : Sử dụng Gaussian Blur để làm mờ ảnh. Với kernel size = 10 ảnh sau khi blur sẽ mờ hầu hết các mạch máu và các đốm bệnh.
- Bước 2 : Trừ ảnh crop cho ảnh sau khi blur bằng công thức trộn hai ảnh

$$f = \alpha * a - \beta * b + \gamma$$

Trong đó:

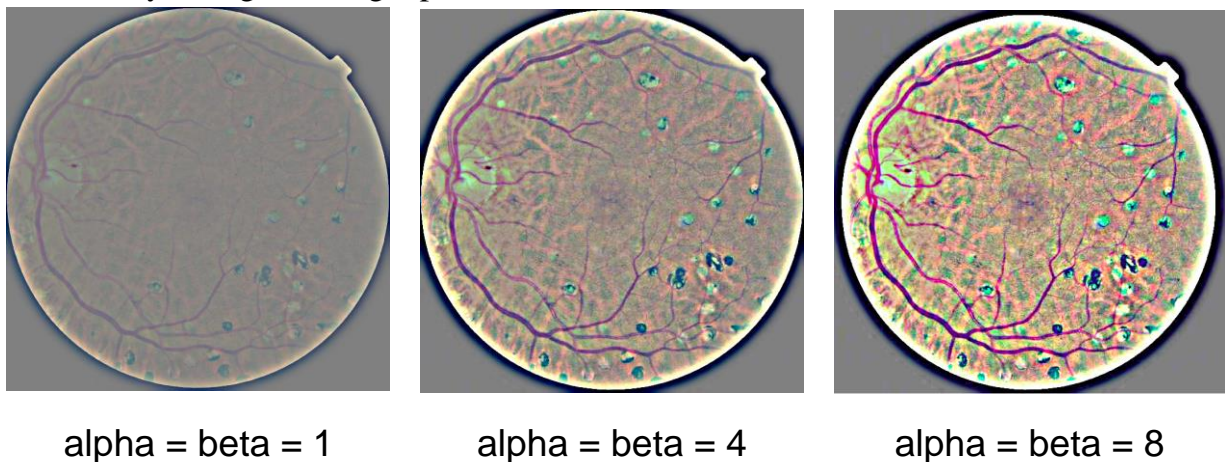
- + a : ảnh crop
- + b : ảnh crop sau khi blur
- + alpha, beta và gamma : hằng số

Mục đích sử dụng thêm hằng số gamma là để tăng sáng vì ảnh sau khi trộn khá tối và các chi tiết cần thiết vẫn chưa thể hiện rõ vì màu nền của võng mạc gần như trùng với màu các chi tiết đó.



Hình 2.2. Kết quả với gamma bằng 0 và 128

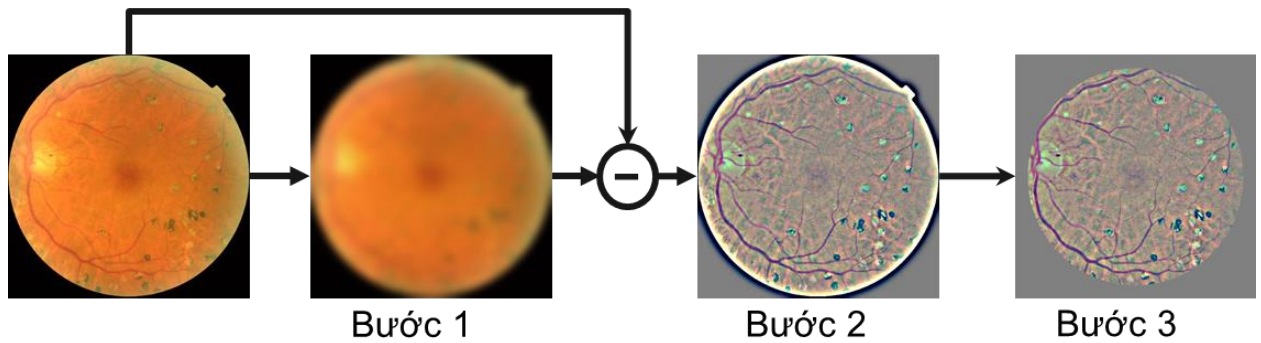
Hai hằng số alpha và beta có tác dụng giúp tăng sự khác biệt giữa 2 ảnh trước và sau khi blur. Nếu chọn alpha và beta nhỏ thì các chi tiết cần làm rõ sẽ không nổi bật hơn so với ảnh gốc. Còn nếu sử dụng số quá lớn sẽ làm xuất hiện thêm những chi tiết nhiễu ở vùng mờ để gây nhầm lẫn cho mô hình. Vì vậy chúng em dùng  $\alpha = \beta = 4$



Hình 2.3. Kết quả với từng cặp trọng số alpha, beta khác nhau

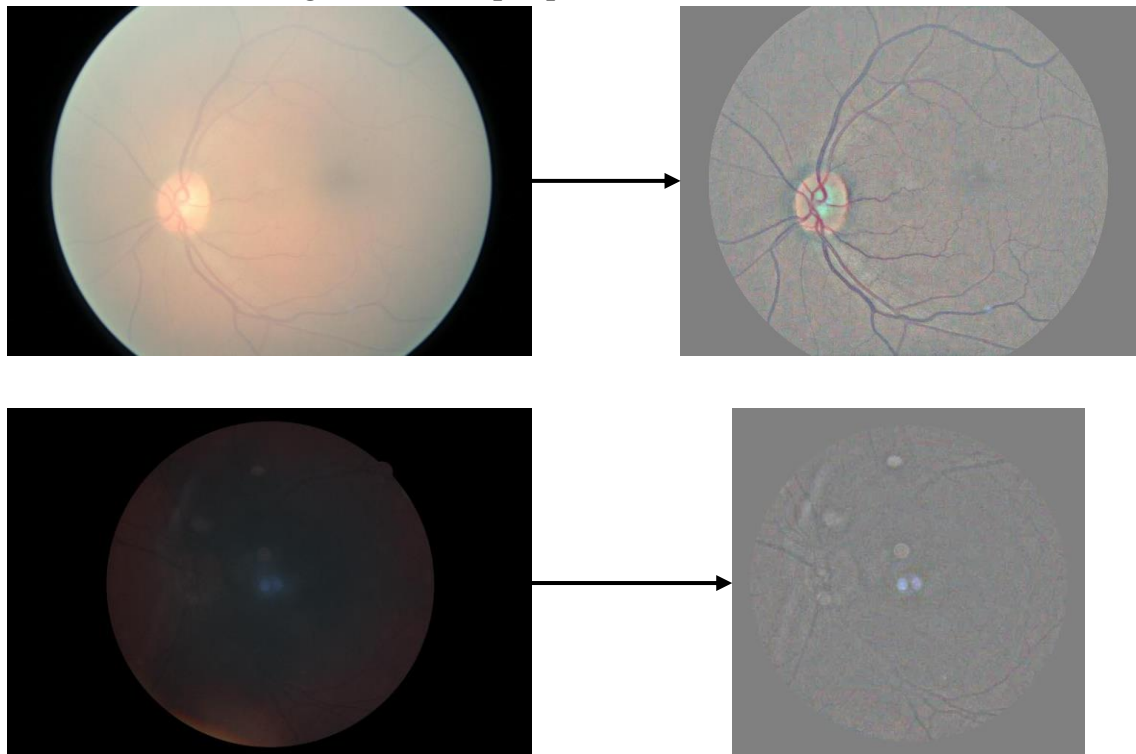
- Bước 3 : Cuối cùng cắt bỏ đi phần viền trắng và đen xuất hiện khi xử lý các bước trên vì chi tiết này không cần thiết cho việc huấn luyện mô hình.





Hình 2.4. Trình tự thực hiện phương pháp btgraham-300

Có thể dễ dàng nhận biết được bằng mắt thường những chi tiết về mạch máu và đốm bệnh ở võng mạc sau khi xử lý bằng phương pháp trên. Không những thế, phương pháp này còn giúp làm rõ được những chi tiết đó ở những ảnh xấu mà chúng em đã đề cập ở phần trước.



Hình 2.5. Ảnh gốc (bên trái) và ảnh btgraham-300 (bên phải)

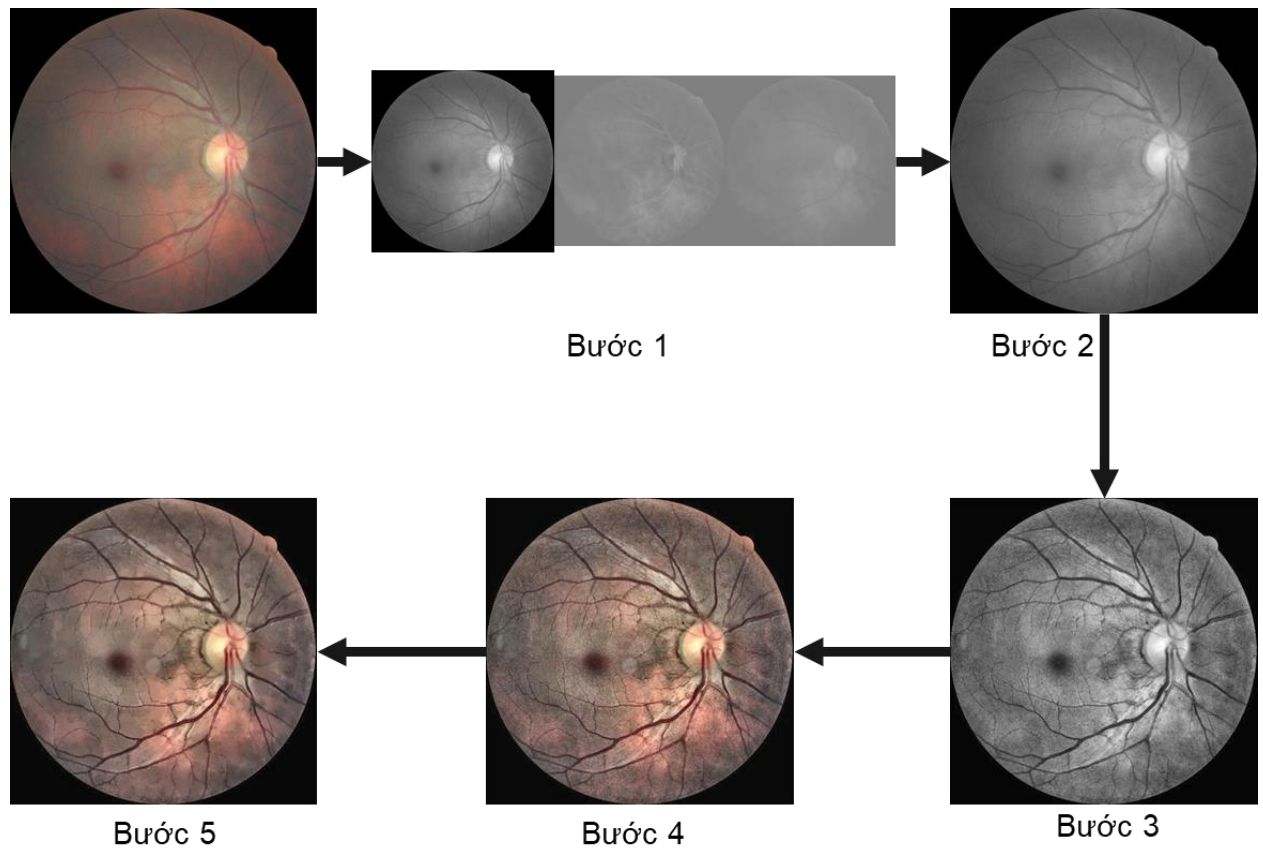
### 3. CLAHE

CLAHE là một phương pháp phổ biến sử dụng để làm rõ các chi tiết quan tâm trong ảnh [5]. Ý tưởng của phương pháp này là thực hiện cân bằng histogram trên từng phần của ảnh. Tuy nhiên, lưu ý rằng phương pháp này chỉ được sử dụng trên



ảnh xám còn ảnh trong bộ dataset đang được sử dụng là ảnh màu, nên quá trình xử lý sẽ trải qua nhiều bước khác nhau như sau:

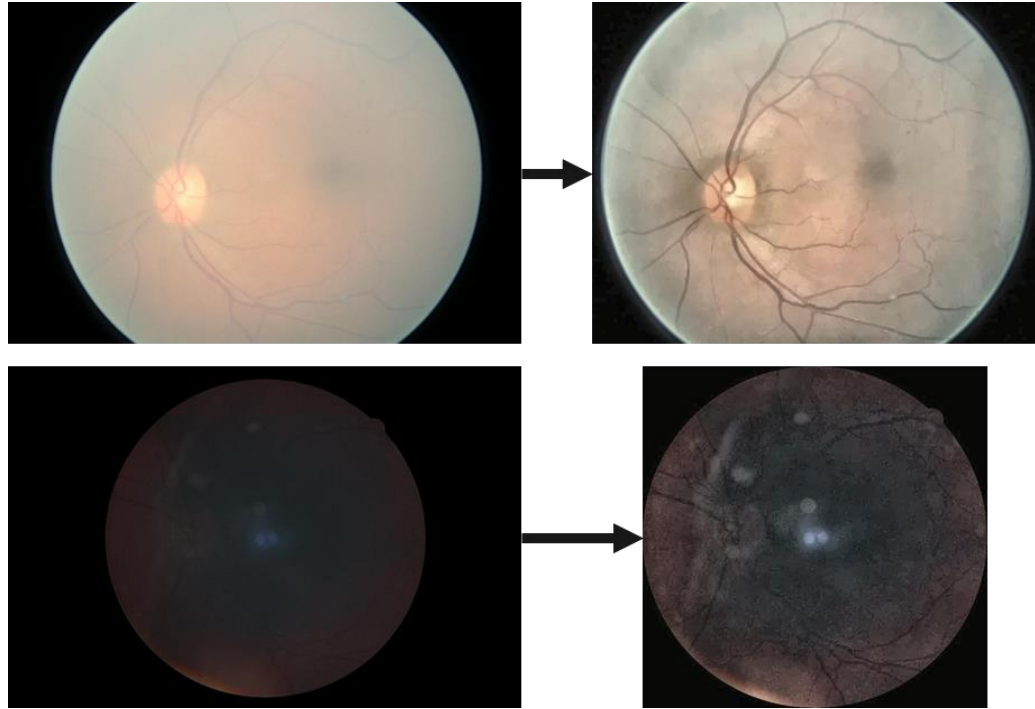
- Bước 1: Ảnh sau khi được crop sẽ được chuyển đổi sang hệ màu LAB. Hệ màu LAB là một hệ màu gồm 3 kênh:
  - L: Kênh chứa thông tin về độ sáng của ảnh
  - A: Kênh chứa các giá trị màu từ Green - Red
  - B: Kênh chứa các giá trị màu từ Blue - Yellow
- Bước 2: Tách lấy kênh L để thực hiện cân bằng histogram trên kênh ảnh này.
- Bước 3: Thực hiện cân bằng histogram trên từng phần ảnh vừa tách
- Bước 4: Merge ảnh đã cân bằng histogram với 2 kênh màu A và B đã tách ở bước 1 để khôi phục lại màu sắc cho ảnh
- Bước 5: Thực hiện khử nhiễu các đốm muối tiêu bằng hàm FastNlMeansDenoisingColored của OpenCV.



Hình 2.6. Trình tự các bước thực hiện xử lý ảnh CLAHE

Có thể thấy được, sau khi trải qua các bước, từ ảnh gốc ban đầu, ta đã thu được một ảnh có độ tương phản cao hơn, thể hiện rõ được các mạch máu cũng như là các

chi tiết bệnh hơn mà không làm mất đi màu sắc cũng như là các chi tiết khác của ảnh. Hơn nữa, phương pháp xử lý ảnh này còn thể hiện được tính nổi trội ở những ảnh chịu tác động của môi trường dẫn đến quá tối hoặc quá sáng được đề cập ở phần trước.



*Hình 2.7. Ảnh gốc (bên trái) và ảnh CLAHE (bên phải)*

### III. HUẤN LUYỆN MÔ HÌNH

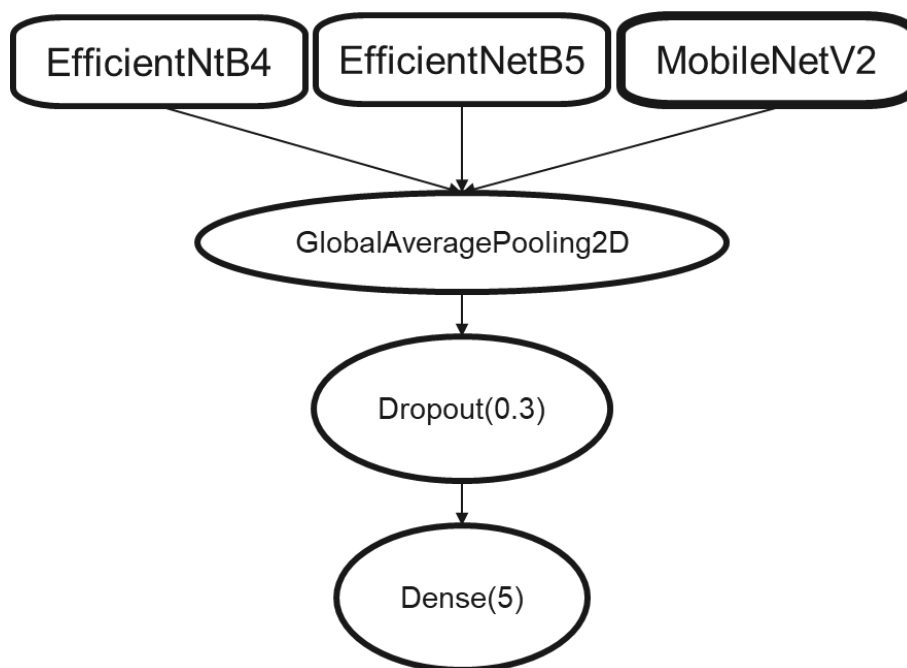
#### 1. Phương pháp đánh giá

Như đề cập ở phần trước, bộ dữ liệu mà chúng em đang sử dụng đang bị mất cân bằng ở cả tập train và tập test, vì vậy các phương pháp đánh giá thường được sử dụng đối với những bộ dữ liệu mất cân bằng này như: F1 score, Recall, Precision. Kappa...

Trong đồ án lần này, chúng em sử dụng phương pháp Kappa, cụ thể hơn là Quadratic Weighted Kappa để đánh giá kết quả. Hơn nữa, phương pháp này còn là phương pháp được sử dụng để đánh giá trong Competition liên quan đến bộ data chúng em sử dụng của Kaggle nên từ đó chúng em có thể đánh giá và so sánh với các nhóm thực hiện khác.

#### 2. Tinh chỉnh mô hình và các tham số

Trong quá trình thực hiện đồ án, chúng em sử dụng 3 mô hình CNN xuyên suốt đó là EfficientNetB4, EfficientNetB5 và MobileNetV2. Bên cạnh đó, chúng em sử dụng pretrained model trên tập ImageNet. Tuy nhiên vì bản chất của hai tập dữ liệu ImageNet và tập dữ liệu sử dụng trong đồ án là khác nhau, nên chúng em sẽ tiếp tục train dựa trên các trọng số pretrained mà không đóng băng bất cứ CNN nào lại cả. Hơn nữa, vì pretrained model dự đoán dựa trên 1000 class khác nhau nên chúng em sẽ thay đổi và tinh chỉnh một vài layer cuối để phù hợp hơn với bộ data mà chúng em sử dụng.



Qua đó, sau khi sử dụng pretrained ở các lớp CNN, chúng em thêm 1 lớp GlobalAveragePooling2D để làm phẳng các điểm dữ liệu cũng như trích xuất ra các đặc trưng quan trọng, tiếp đó là một lớp Dropout với rate = 0.3 để giúp cho mô hình tránh gặp phải overfitting, và sau cùng là lớp Dense với số unit = 5 và hàm kích hoạt Softmax để phù hợp với bộ dữ liệu 5 class đang sử dụng.

Một số các siêu tham số và kỹ thuật khác chúng em sử dụng xuyên suốt trong quá trình training như:

- Loss Function: CrossEntropy
- Optimizer: Adam
- Learning Rate: 0.001
- Batch Size: 16

### 3. Hướng tiếp cận Imbalance data

Sau khi xử lý ảnh, chúng em thực hiện huấn luyện mô hình với 3 data : Crop, btgraham-300 và CLAHE (với số lượng ảnh bằng với số lượng ảnh gốc). Chúng em sử dụng 3 mô hình EfficientNetB4 , EfficientNetB5 và MobileNetV2. Đồng thời sử dụng pretrained model với data ImageNet để giảm thiểu thời gian huấn luyện.

Sau khi thực hiện huấn luyện, chúng em thu được kết quả sau:

Model	Preprocessing	Accuracy	Kappa
MobileNetv2	Crop	<u>0.80</u>	0.42
	Btgraham-300	0.79	<u>0.59</u>
	CLAHE	0.79	0.52
EfficientNet B4	Crop	0.79	0.57
	Btgraham-300	0.75	0.08
	CLAHE	0.77	0.42
EfficientNet B5	Crop	0.78	0.45
	Btgraham-300	0.77	0.58
	CLAHE	0.75	0.29

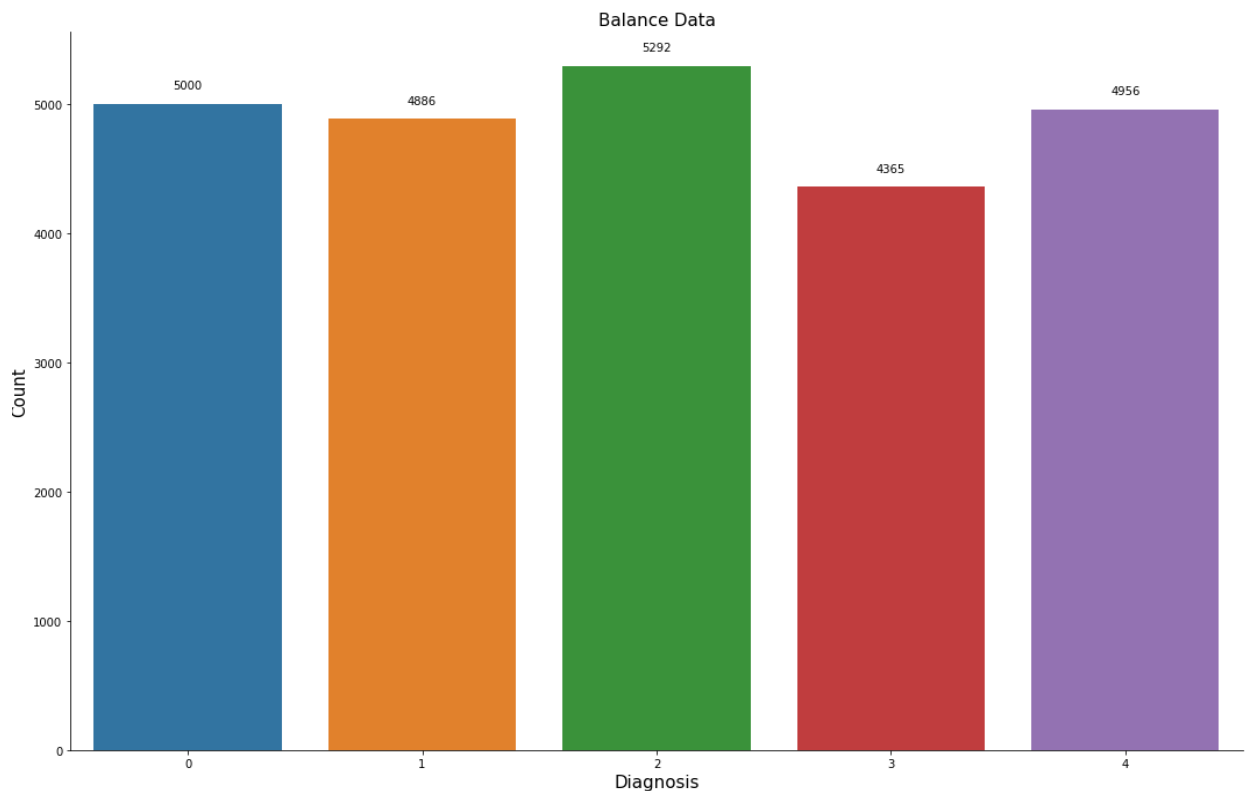
Bảng 2-1: Bảng kết quả với hướng tiếp cận Imbalance data

Các mô hình đều cho kết quả accuracy khá cao ( $>0.75$ ), tuy nhiên ở đây mô hình dự đoán lệch khá nhiều về class 0 do sự mất cân bằng khá nặng của data (số lượng ảnh class 0 chiếm hơn 70% tổng số lượng ảnh). Vì thế nên kết quả kappa không cao được như vậy. Việc nhầm lẫn này khá nguy hiểm vì nếu chẩn đoán nhầm class 1, 2, 3, 4 (có bệnh) sang class 0 (không có bệnh) sẽ rất nguy hiểm, điều đó khiến bệnh không được phát hiện và điều trị kịp thời gây ảnh hưởng xấu đến vòng mạng bệnh nhân.

Vì vậy, chúng em quyết định cân bằng lại data.

#### 4. Hướng tiếp cận Balance data:

Chúng em thực hiện cân bằng data bằng cách giảm số lượng ảnh class 0 từ 25810 ảnh xuống còn 5000 ảnh. Các class khác thực hiện tăng cường dữ liệu bằng cách tăng giảm độ sáng ngẫu nhiên trong khoảng  $[-10, 10]$ , lật ảnh theo chiều ngang và chiều dọc.



Hình 2.8. Thống kê số lượng ảnh từng class sau khi cân bằng.

Sau khi cân bằng, chúng em thực hiện huấn luyện với 3 mô hình tương tự như đã huấn luyện với Imbalance data.

Sau khi huấn luyện thu được kết quả :

Model	Preprocessing	Accuracy	Kappa
MobileNetv2	Crop	0.33	0.28
	Btgraham-300	0.37	0.33
	CLAHE	0.35	0.28
EfficientNet B4	Crop	0.30	0.25
	Btgraham-300	<u>0.55</u>	0.33
	CLAHE	0.35	0.32
EfficientNet B5	Crop	0.29	0.14
	Btgraham-300	0.52	<u>0.48</u>
	CLAHE	0.31	0.09

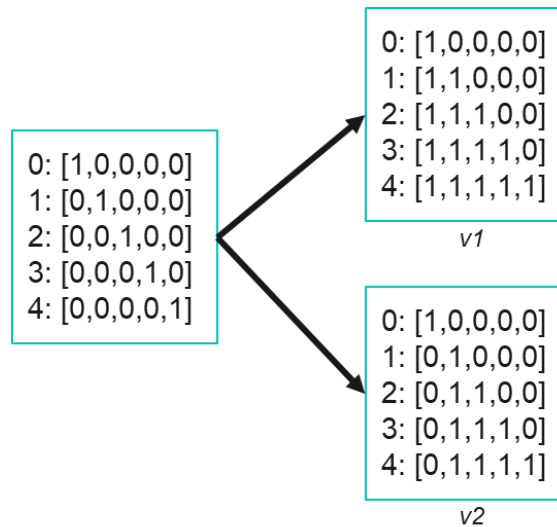
Bảng 2-2: Bảng kết quả với hướng tiếp cận Balance data

Sau khi cân bằng, tuy đã giảm bớt hiện tượng dự đoán lệch về class 0, nhưng kết quả vẫn không cải thiện hơn. Cụ thể là mô hình EfficientNetB5 với phương pháp xử lý ảnh btgraham-300 có kết quả kappa cao nhất (0.48), thấp hơn so với một vài kết quả kappa ở Imbalance data.

## 5. Hướng tiếp cận Multi-class:

Ý tưởng chính của hướng tiếp cận này được chúng em tìm hiểu và tham khảo trong phần discussion của kaggle [13]. Nội dung chính của phần discussion này xoay quanh vấn đề bài toán đang giải quyết nên là một bài toán classification thông thường với output dạng one-hot vector hay là một bài toán regression với output dạng ordinary regression.

Vì thế, chúng em quyết định thử đổi góc nhìn sang hướng regression và thực nghiệm trên chúng ở 2 version khác nhau, version 1 là ý tưởng trích từ discussion, còn version 2 được chúng em đề xuất thêm



Lấy kinh nghiệm từ những lần thử nghiệm trước, chúng em chỉ train trên cách xử lý ảnh btgraham-300 và model EfficientNetB5 và thu được kết quả như sau:

Model	Accuracy	Kappa
Multi-classes v1	0.51	0.24
Multi-classes v2	<u>0.54</u>	<u>0.24</u>

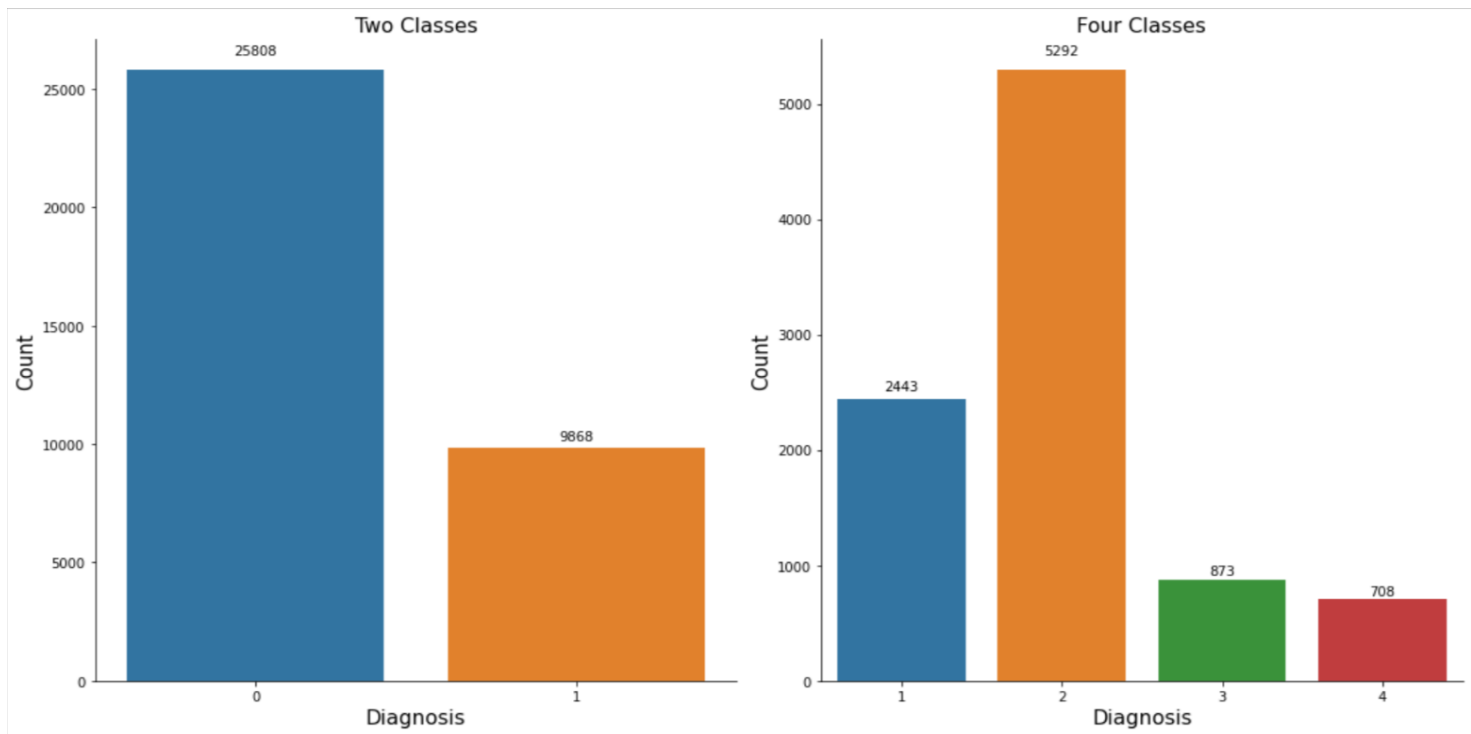
Bảng 2-3: Bảng kết quả với hướng tiếp cận Multi-Class

Dựa vào bảng, ta có thể thấy được cách tiếp cận này cho ra kết quả không tốt, tệ hơn cả những cách tiếp cận trước. Hơn nữa, cách tiếp cận này còn có xu hướng “lạc quan hóa”. Có nghĩa là những ảnh mắt bệnh nặng, model có xu hướng dự đoán nhẹ hơn, còn những ảnh mang bệnh nhẹ thì model có xu hướng dự đoán thành không có bệnh. Đây là một điều không tốt, nhất là trong y học.

## 6. Hướng tiếp cận Imbalance multi-model :

Ý tưởng chính của hướng tiếp cận này là chia bài toán thành hai bài toán nhỏ hơn : Phân biệt có bệnh hoặc không có bệnh, nhận diện mức độ ảnh có bệnh. Sau khi huấn luyện riêng biệt hai bài toán nhỏ trên, sẽ dự đoán bằng cách đưa ảnh vào mô hình để dự đoán xem có bệnh hay không, nếu không thì gán ảnh đó cho class 0, nếu có thì tiếp tục đưa ảnh qua mô hình thứ 2 để tiếp tục dự đoán mức độ bệnh. Để thực hiện phương pháp này thì chúng em cần phải gán nhãn lại data.





Hình 2.9. Thống kê số lượng ảnh sau khi re-class

Tương tự Multi-class, ở hướng tiếp cận này chúng em tiếp tục sử dụng phương pháp xử lý ảnh btgraham-300 và huấn luyện bằng EfficientNetB5.

Thêm vào đó, chúng em có thực hiện phương pháp chọn lại ngưỡng đánh giá. Trong trường hợp kết quả xác suất sau khi dự đoán cho hai class 0 và 1 là  $[0.5656, 0.4344]$ , thông thường với kết quả trên thì mô hình sẽ dự đoán đó là class 0 vì ngưỡng mặc định là 0.5. Nếu như thay đổi ngưỡng đánh giá thành 0.4 thì mô hình này sẽ dự đoán là class 1. Điều này khiến cho mô hình ưu tiên lựa chọn những class khác hơn là class 0. Vì nếu dự đoán nhầm thành có bệnh sẽ không ảnh hưởng nhiều đến bệnh nhân so với việc dự đoán nhầm thành không có bệnh.

Với hướng tiếp cận trên cho ra kết quả :

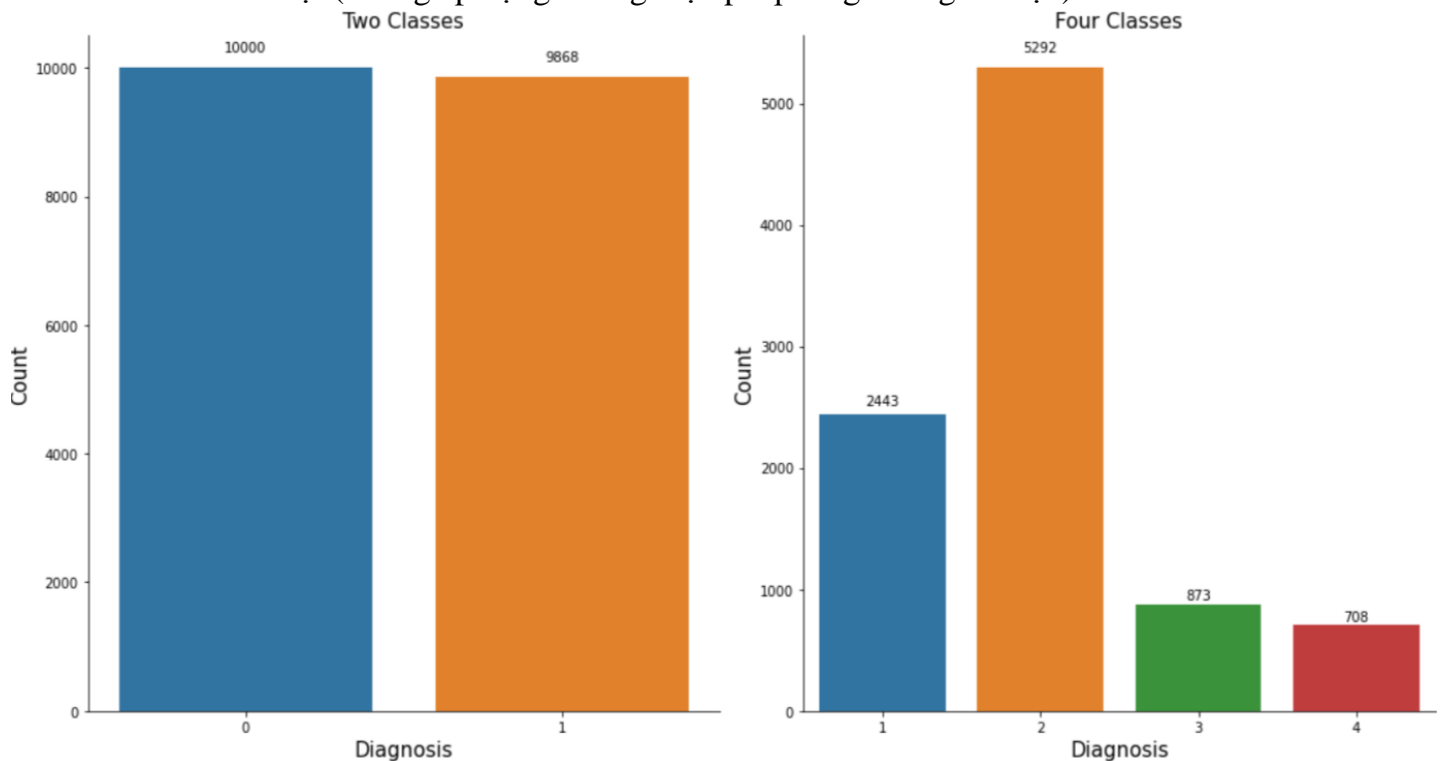
Data	Threshold	Accuracy	Kappa
Imbalance	0.5	<u>0.81</u>	0.69
	0.4	0.80	<u>0.70</u>
	0.3	0.79	0.70

Bảng 2-4: Bảng kết quả với hướng tiếp cận Imbalance Multi-Model

Với hướng tiếp cận này, mô hình đã cho kết quả kappa cao hơn hẳn những hướng tiếp cận trước. Và vì train với data mất cân bằng nên hiện tượng dự đoán lệch về class 0 vẫn xảy ra mặc dù đã sử dụng phương pháp chọn lại ngưỡng đánh giá. Vì thế chúng em quyết định tiếp tục sử dụng hướng tiếp cận này và train sau khi cân bằng lại data.

### 7. Hướng tiếp cận Balance multi-model:

Ở hướng tiếp cận này, chúng em chỉ cân bằng data chỉ bằng cách giảm số lượng ảnh class 0 xuống 10000 ảnh để cân bằng số lượng đối với tổng số lượng ảnh các class còn lại (không áp dụng những biện pháp tăng cường dữ liệu).



Hình 2.10. Thống kê số lượng ảnh re-class sau khi cân bằng

Kết quả thu được :

Data	Threshold	Accuracy	Kappa
Balance	0.5	0.73	0.65
	0.4	<u>0.78</u>	<u>0.74</u>
	0.3	0.64	0.59

*Bảng 2-5: Bảng kết quả với hướng tiếp cận Balance Multi-Class*

Với hướng tiếp cận này, đã thu được kết quả kappa cao nhất là 0.74, cao hơn so với Imbalance multi-model và những hướng tiếp cận cũ

#### IV. NHẬN XÉT

Dưới đây là bảng tổng hợp đầy đủ tất cả các kết quả trong quá trình training, cũng như kết quả submit kaggle.

Model	Accuracy	Kappa
EfficientNetB4-ImbalanceData-Crop	0.8	0.42
EfficientNetB5-ImbalanceData-Crop	0.78	0.45
MobileNetV2-ImbalanceData-Crop	0.8	0.42
EfficientNetB4-ImbalanceData-CLAHE	0.77	0.42
EfficientNetB5-ImbalanceData-CLAHE	0.75	0.29
MobileNetV2-ImbalanceData-CLAHE	0.79	0.52
EfficientNetB4-ImbalanceData-Btgraham	0.75	0.08
EfficientNetB5-ImbalanceData-Btgraham	0.77	0.58
MobileNetV2-ImbalanceData-Btgraham	0.79	0.59
EfficientNetB4-BalanceData-Crop	0.30	0.25
EfficientNetB5-BalanceData-Crop	0.29	0.14
MobileNetV2-BalanceData-Crop	0.33	0.28
EfficientNetB4-BalanceData-CLAHE	0.35	0.32
EfficientNetB5-BalanceData-CLAHE	0.31	0.09
MobileNetV2-BalanceData-CLAHE	0.35	0.28
EfficientNetB4-BalanceData-Btgraham	0.55	0.33
EfficientNetB5-BalanceData-Btgraham	0.52	0.48
MobileNetV2-BalanceData-Btgraham	0.37	0.33
EfficientNetB5-MultiClass-v1	0.51	0.24

<b>EfficientNetB5-MultiClass-v2</b>	<b>0.54</b>	<b>0.24</b>
<b>EfficientNetB5-MultiModel-Imbalance0.5</b>	<b>0.81</b>	<b>0.69</b>
<b>EfficientNetB5-MultiModel-Imbalance0.4</b>	<b>0.80</b>	<b>0.70</b>
<b>EfficientNetB5-MultiModel-Imbalance0.3</b>	<b>0.79</b>	<b>0.70</b>
<b>EfficientNetB5-MultiModel-Balance0.5</b>	<b>0.73</b>	<b>0.65</b>
<b>EfficientNetB5-MultiModel-Balance0.4</b>	<b>0.78</b>	<b><u>0.74</u></b>
<b>EfficientNetB5-MultiModel-Balance0.3</b>	<b>0.64</b>	<b>0.59</b>

Bảng 3-1: Bảng kết quả với tất cả các hướng tiếp cận được sử dụng

All	Successful	Selected	
Submission and Description	Private Score	Public Score	Use for Final Score
<a href="#">submitfinal.csv</a> 20 days ago by <a href="#">Linh trương</a> submitfinal	0.74480	0.75046	<input type="checkbox"/>

Hình 3.1: Private Score và Public Score khi submit kết quả tốt nhất lên Kaggle

**Xử lý ảnh:** Sau quá trình training và thử nghiệm, chúng em nhận thấy rằng: trong 2 cách xử lý ảnh là CLAHE và Btgraham-300 thì Btgraham-300 có cách xử lý cho kết quả tốt hơn.

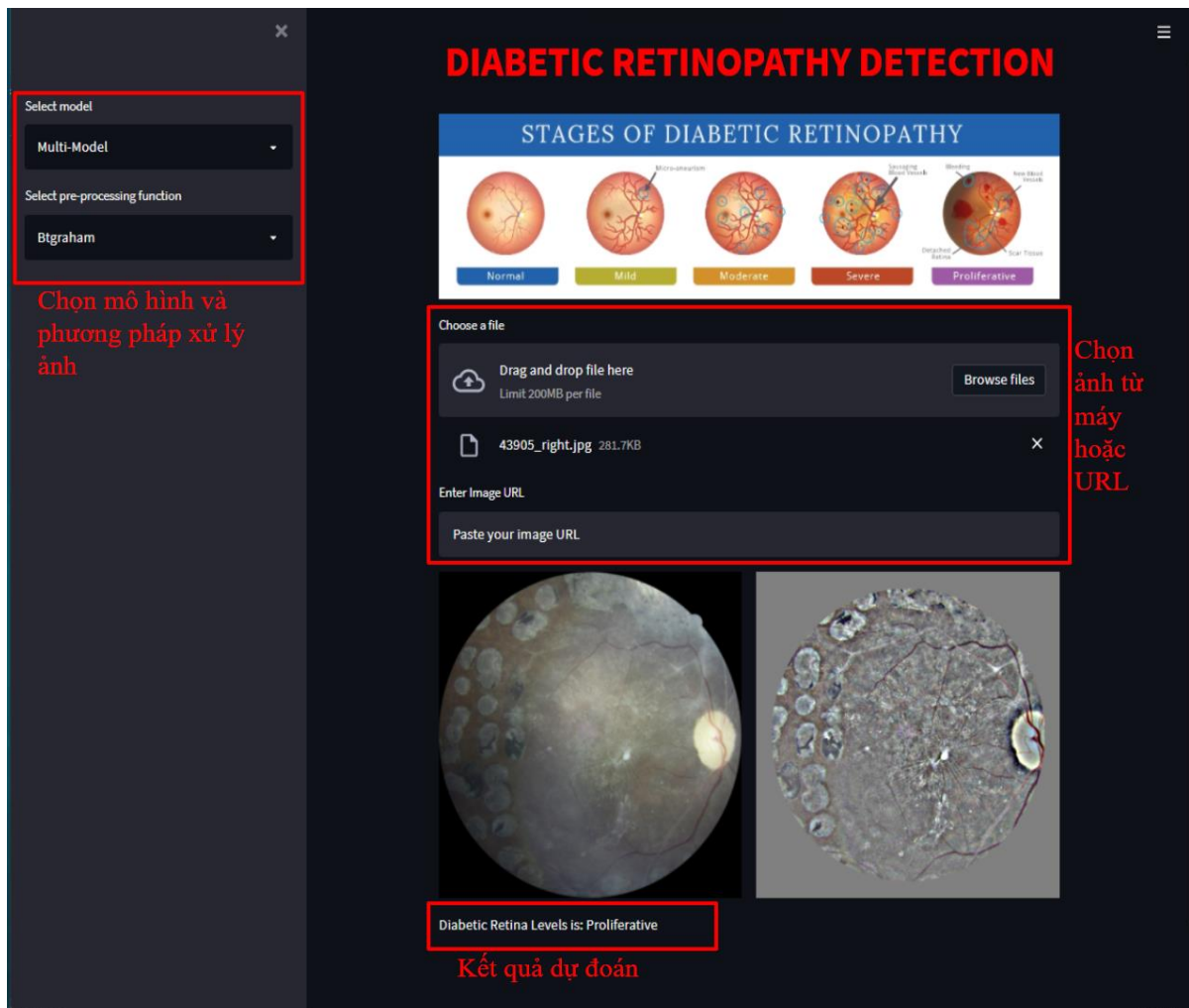
**Model:** Trong 3 model được sử dụng xuyên suốt là MobileNetV2, EfficientNetB4 và EfficientNetB5 thì EfficientNetB5 cho ra mô hình có kết quả tốt hơn.

Model tốt nhất hiện tại là mô hình **EfficientNetB5** với phương pháp xử lý ảnh **Btgraham-300**, hướng tiếp cận **MuiltiModel** dựa trên bộ data được **cân bằng dữ liệu** với **ngưỡng được chọn là 0.4** đã phần nào xử lý tốt trên bộ dữ liệu bị mất cân bằng, tuy nhiên vẫn còn nhiều trường hợp mô hình dự đoán sai. Bên cạnh đó, hướng tiếp cận cho ra model tốt nhất là Multi-model cho thời gian dự đoán khá lâu vì quá trình dự đoán phải đưa qua 2 model liên tiếp nhau.

## V. ỨNG DỤNG VÀ HƯỚNG PHÁT TRIỂN

### 1. Ứng dụng

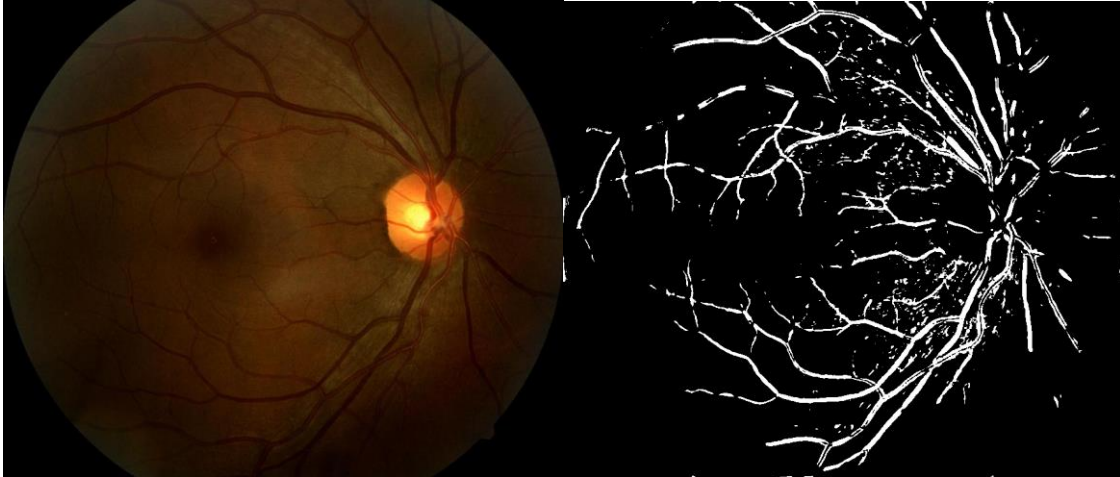
Chúng em đã xây dựng một Web App với sự hỗ trợ của thư viện Streamlit với chức năng chính là lựa chọn mô hình, phương pháp xử lý ảnh và chọn ảnh cần dự đoán. Sau đó hệ thống sẽ hiển thị ảnh sau khi được xử lý cũng như kết quả dự đoán của mô hình



### 2. Hướng phát triển

Về *model*, ta có thể sử dụng thêm nhiều mô hình khác khác để cải thiện kết quả, ví dụ như: EfficientNet V2, ResNet-101, Inception v3,... theo như discussion trên kaggle[15].

Về *xử lý ảnh*, ta có thể thử nghiệm thêm phương pháp xử lý ảnh khác để có thể tách rõ các mạch máu và các đốm bệnh ra khỏi ảnh như theo hình dưới đây



*Hình 3.2: Ảnh gốc (ảnh trái) sau khi xử lý tách kênh màu Green và thực hiện lấy ngưỡng và thu được hình kết quả (ảnh phải)*



## VI. BẢNG PHÂN CÔNG CÔNG VIỆC

STT	Họ và tên	MSSV	Nhiệm vụ	Mức độ hoàn thành
1	Trương Xuân Linh	19521759	<ul style="list-style-type: none"> <li>• Tìm hiểu và thực hiện phương pháp tiền xử lý ảnh Crop và phương pháp xử lý ảnh CLAHE</li> <li>• Thực hiện train 3 model sau khi tiền xử lý với Crop và CLAHE với các hướng tiếp cận: Imbalance data, Balance Data.</li> <li>• Tìm hiểu và thực hiện training theo hướng tiếp cận Multi-class, Imbalance multi-model và Balance multi-model.</li> <li>• Thực hiện các phương pháp cân bằng dữ liệu</li> <li>• Tìm hiểu phương pháp đánh giá Kappa</li> <li>• Viết báo cáo và làm slide</li> </ul>	100%
2	Nguyễn Minh Trí	19522389	<ul style="list-style-type: none"> <li>• Tìm hiểu và thực hiện phương pháp xử lý ảnh Btgraham-300</li> <li>• Thực hiện train 3 model sau khi tiền xử lý với Btgraham-300 với các hướng tiếp cận: Imbalance data, Balance data</li> <li>• Thực hiện training theo hướng tiếp cận Multi-class và Imbalance multi- model</li> <li>• Thực hiện các phương pháp cân bằng dữ liệu</li> <li>• Tìm hiểu phương pháp đánh giá Kappa</li> <li>• Viết báo cáo và làm slide</li> </ul>	100%

## VII. THAM KHẢO

- [1] Kaggle Diabetic Retinopathy Detection competition report  
<https://storage.googleapis.com/kaggle-forum-message-attachments/88655/2795/competitionreport.pdf>
- [2] Tổng quan về Chụp OCT bán phần sau nhãn cầu  
<https://www.vinmec.com/vi/ky-thuat-chan-doan/chup-oct-ban-phan-sau-nhan-cau-34/>
- [3] Bệnh võng mạc đái tháo đường  
<https://www.vinmec.com/vi/tin-tuc/thong-tin-suc-khoe/benh-vong-mac-dai-thao-duong/>
- [4] Diabetes  
<https://www.who.int/en/news-room/fact-sheets/detail/diabetes>
- [5] CLAHE Histogram Equalization – OpenCV  
<https://www.geeksforgeeks.org/clahe-histogram-equalization-opencv/>
- [6] EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks  
<https://arxiv.org/pdf/1905.11946v5.pdf>
- [7] MobileNetV2: Inverted Residuals and Linear Bottlenecks  
<https://arxiv.org/pdf/1801.04381v4.pdf>
- [8] Diabetic Retinopathy Detection - Leaderboard  
<https://www.kaggle.com/c/diabetic-retinopathy-detection/leaderboard>
- [9] Diabetic Retinopathy Detection – Competition  
<https://www.kaggle.com/c/diabetic-retinopathy-detection>
- [10] Global average pooling 2D  
<https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/blocks/global-average-pooling-2d>
- [11] 5 Perspectives to Why Dropout Works So Well  
<https://towardsdatascience.com/5-perspectives-to-why-dropout-works-so-well-1c10617b8028>
- [12] [Deep Learning] Chẩn đoán bệnh võng mạc đái tháo đường với Deep Learning - Diabetic Retinopathy Kaggle Competition  
[https://viblo.asia/p/deep-learning-chan-doan-benh-vong-mac-dai-thao-duong-voi-deep-learning-diabetic-retinopathy-kaggle-competition-gDVK2G6jZLj#\\_metric-evaluation-5](https://viblo.asia/p/deep-learning-chan-doan-benh-vong-mac-dai-thao-duong-voi-deep-learning-diabetic-retinopathy-kaggle-competition-gDVK2G6jZLj#_metric-evaluation-5)
- [13] APTOS 2019 Blindness Detection - Regression vs Classification ?  
<https://www.kaggle.com/c/aptos2019-blindness-detection/discussion/98239#latest-573709>
- [14] Quadratic Kappa Metric explained in 5 simple steps  
<https://www.kaggle.com/aroraaman/quadratic-kappa-metric-explained-in-5-simple-steps>

**[15]** Why do everybody use resnet?

<https://www.kaggle.com/c/aptos2019-blindness-detection/discussion/99550>