**XÂY DỰNG ỨNG DỤNG HỖ TRỢ CHẨN ĐOÁN BỆNH**

**VÕNG MẠC ĐÁI THÁO ĐƯỜNG**

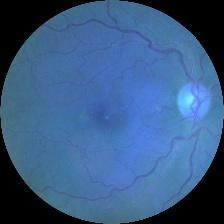
## I. Những việc đã làm được

### 1. Xử lý dữ liệu

* Chuyển hình ảnh sang kênh màu RGB
* Cắt ảnh sát viền võng mạc, loại bỏ phần nền đen dư thừa
* Giảm kích thước hình ảnh 224x224 pixels.



*Hình ảnh tiền xử lý*



*Hình ảnh sau khi đã được xử lý*

### 2. Lựa chọn kiến trúc mô hình

#### 2.1. Mô hình với dữ liệu cân bằng

Thực hiện đánh giá trên các kiến trúc mô hình DenseNet169, EfficientNetB5 và ResNet50 trên 2 nhãn: 0 và 2.

***2.1.1. Tiến hành***

* Dữ liệu :

|  |  |  |
| --- | --- | --- |
| Nhãn | Train | Test |
| 0 | 2489 | 489 |
| 2 | 2511 | 511 |

* Tham số:
  + Batch size: 8
  + Hàm tối ưu: Adam
  + Hàm mất mát: categorical\_crossentropy
* Kiến trúc: sử dụng pre-trained weights.
  + ResNet50: resnet50\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop
  + EfficientNetB5: efficientnet-b5\_imagenet\_1000\_notop
  + DenseNet169: densenet169\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop
* Huấn luyện:
  + Warm up step với learning rate 1e-3: 2 epochs để tránh early overfitting.
  + Huấn luyện mô hình 30 epochs. Tuy nhiên mô hình sẽ dừng huấn luyện (early stopping) nếu như validation loss có giá trị không tốt trong 5 epoch liên tiếp (giảm overfitting).

#### 2.1.2. Đánh giá

|  |  |  |  |
| --- | --- | --- | --- |
|  | ResNet50 | EfficientNetB5 | DenseNet169 |
| Best train accuracy | 0.8653 | 0.8556 | 0.8252 |
| Best train loss | 0.3501 | 0.3303 | 0.3831 |
| Best validation accuracy | 0.7238 | 0.7749 | 0.7419 |
| Best validation loss | 0.5195 | 0.5473 | 0.4901 |
| Test accuracy | 0.7336 | 0.7440 | 0.7440 |

* Có thể thấy, cả 3 mô hình đều mang lại kết quả khả quan và xấp xỉ nhau. Em lựa chọn EfficientNetB5 để tiếp tục xử lý với trường hợp dữ liệu không cân bằng (Em đã thử dùng DenseNet train với dữ liệu mất cân bằng, nhưng cho kết quả không tốt)

#### 2.2. Mô hình với dữ liệu không cân bằng

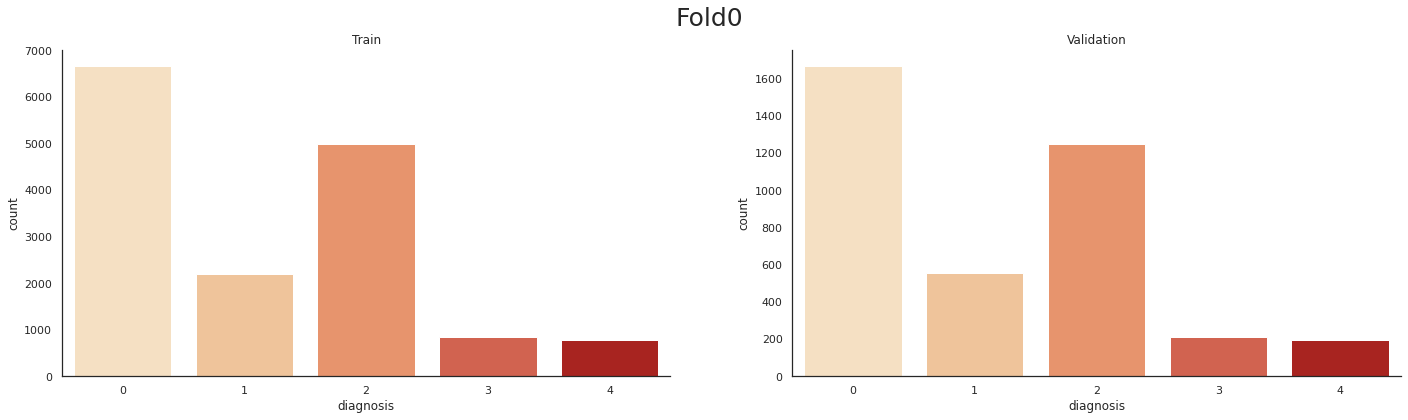
Thực hiện đánh giá mô hình EfficientNetB5 trên 5 nhãn dữ liệu mất cân bằng 0, 1, 2, 3, 4.

* **EfficientNetB5 - 1**: Được giữ về kiến trúc như ở mục 2.1, output của layer cuối cùng của là 5, activation function (hàm kích hoạt) và loss function (hàm mất mát) vẫn là soft max và categorical\_crossentropy.
* **EfficientNetB5 - 2**: Được giữ như kiến trúc ở mục 2.1, tuy nhiên điểm khác biệt ở đây là output của layer cuối cùng là 1, activation function được thay đổi thành linear, hàm mất mát (loss function) là mean\_squared\_error. Kết quả dự đoán sẽ được so sánh với các ngưỡng cho trước để phân bố về đúng nhãn (mở rộng ngưỡng phân bố nhãn, thay vì 0 chỉ xếp vào lớp 0 thì có thể mở rộng ra < 0.5 là xếp vào lớp 0).

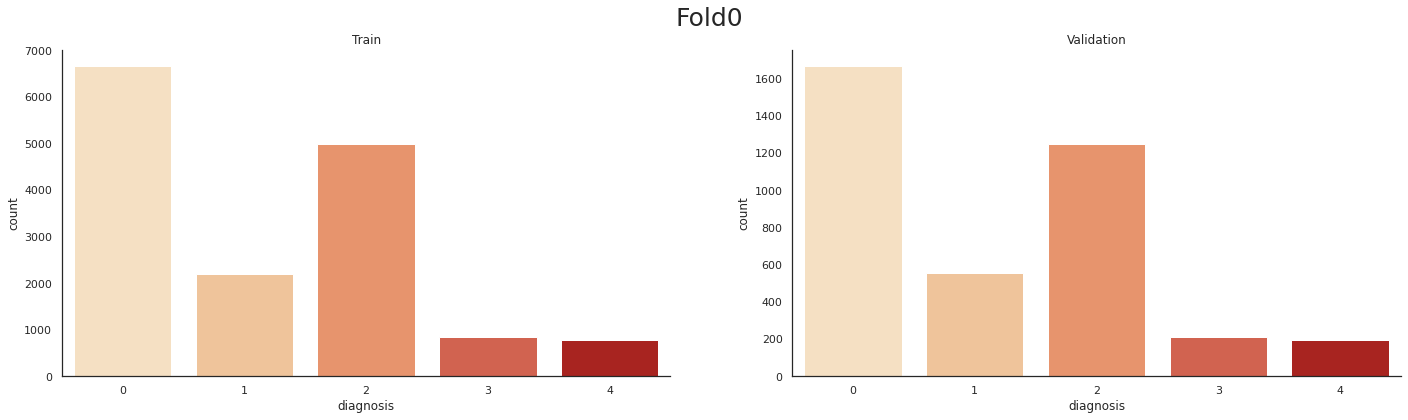
|  |  |
| --- | --- |
| Nhãn | Ngưỡng |
| 0 | < 0.5 |
| 1 | < 1.5 |
| 2 | < 2.5 |
| 3 | < 3.5 |

#### 2.2.1. Tiến hành

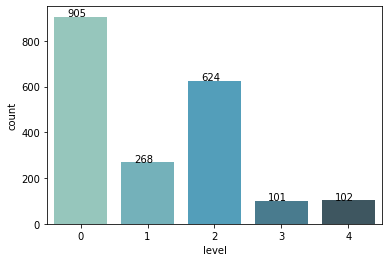
* Dữ liệu



*Dữ liệu huấn luyện (train)*

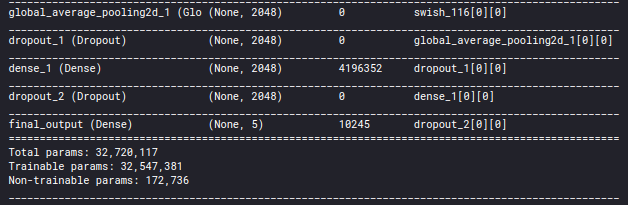


*Dữ liệu xác thực (validation)*

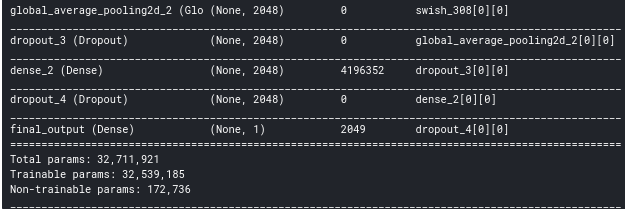


*Dữ liệu kiểm thử (test)*

* Tham số:
  + Batch size: 8
  + Hàm tối ưu: Adam
* Kiến trúc: sử dụng imagenet pretrained weight.
  + EfficientNetB5 - 1: một vài layers cuối



* + EfficientNetB5 - 2: một vài layers cuối

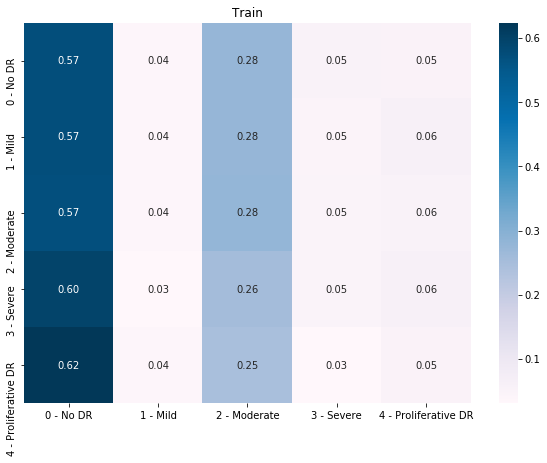


* Huấn luyện:
  + Warm up step với learning rate 1e-3: 5 epochs để tránh early overfitting.
  + Huấn luyện mô hình 30 epochs, tuy nhiên mô hình sẽ dừng huấn luyện (early stopping) nếu như validation loss có giá trị không tốt trong 5 epoch liên tiếp (giảm overfitting).

#### 2.2.2. Đánh giá

|  |  |  |
| --- | --- | --- |
|  | EfficientNetB5 - 1 | EfficientNetB5 - 2 |
| Best train accuracy | 0.7126 | 0.7179 |
| Best train loss | 0.5879 | 0.2876 |
| Best validation accuracy | 0.6523 | 0.6291 |
| Best validation loss | 0.8751 | 0.5442 |
| Test accuracy | 0.6555 | 0.6294 |

* Đánh giá lại trên bộ dữ liệu huấn luyện:
  + **EfficientNetB5 - 1**:



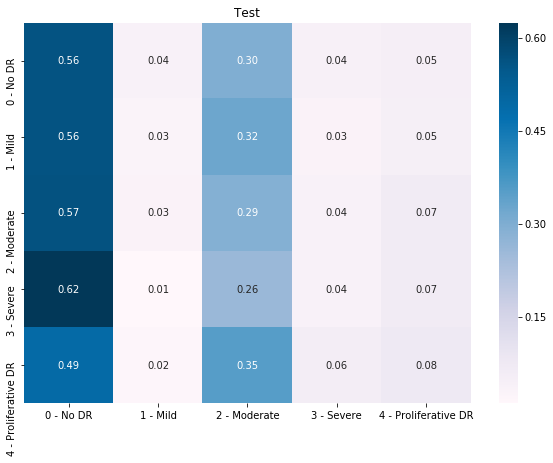
*Ma trận nhầm lẫn áp dụng tập test của EfficientNet B5 -1*

* + **EfficientNetB5 - 2**:



*Ma trận nhầm lẫn áp dụng tập train của EfficientNet B5 -2*

* Đánh giá trên dữ liệu kiểm thử:
  + **EfficientNetB5 - 1:**



*Ma trận nhầm lẫn áp dụng tập test của EfficientNet B5 -1*

* + **EfficientNetB5 - 2**:



*Ma trận nhầm lẫn áp dụng tập test của EfficientNet B5 -2*

## 3. Kết luận

Sau khi thực hiện huấn luyện và đánh giá trên nhiều kiến trúc mô hình với lượng dữ liệu cân bằng, em đã chọn được kiến trúc mô hình phù hợp: Efficient Net B5.

Việc giải quyết bài toán mất cân bằng theo 2 mô hình ở mục 2, có thể thấy cả hai: EfficientNetB5-1 và EfficientNetB5-2 đều cho ra độ chính xác trong quá trình huấn luyện lẫn kiểm thử (test) xấp xỉ nhau. Tuy nhiên, nếu đánh giá dựa trên ma trận nhầm lẫn, có thể thấy EfficientNetB5-2 đã cho được kết quả tốt hơn, EfficientNetB5-1 chỉ dự đoán được các hình ảnh thuộc 2 nhãn có lượng dữ liệu lớn (0 và 2) trong khi EfficientNetB5-2 đã có thể phân bố hình ảnh về hầu hết các nhãn. Em sẽ tiếp tục nghiên cứu và cải thiện độ chính xác bằng cách dùng k-cross validation ở lần báo cáo trước vào mô hình EfficientNetB5-2 và báo cáo với thầy trong thời gian tới.

## II. Kế hoạch tiếp theo

* Tiếp tục cải thiện mô hình, nâng cao độ chính xác.
* Xây dựng giao diện
* Viết báo cáo