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

**VÕNG MẠC ĐÁI THÁO ĐƯỜNG**

### 1. Đánh giá ResNet với dữ liệu không cân bằng

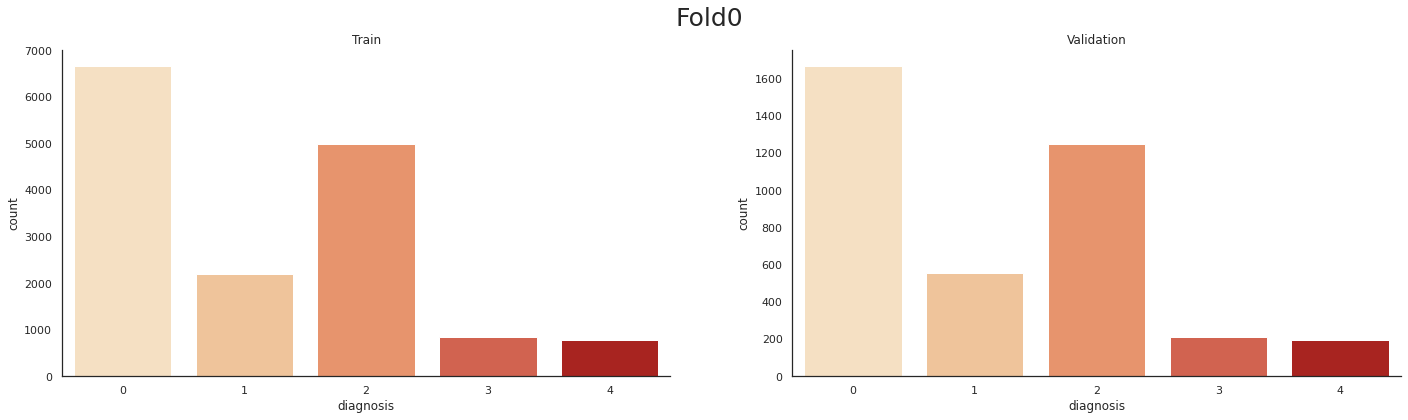
Thực hiện đánh giá mô hình **ResNet50** trên 5 nhãn dữ liệu mất cân bằng 0, 1, 2, 3, 4.

* **ResNet50 - 1**: Được giữ về kiến trúc như mục 2.1 bài báo cáo trước ([link](https://drive.google.com/file/d/14sTSJKCdtKvH34zRqJOkr1TN-ppbjaRG/view?usp=sharing)), 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.
* **ResNet50 -2**: Được giữ về kiến trúc như mục 2.1 bài báo cáo trước ([link](https://drive.google.com/file/d/14sTSJKCdtKvH34zRqJOkr1TN-ppbjaRG/view?usp=sharing)), 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).

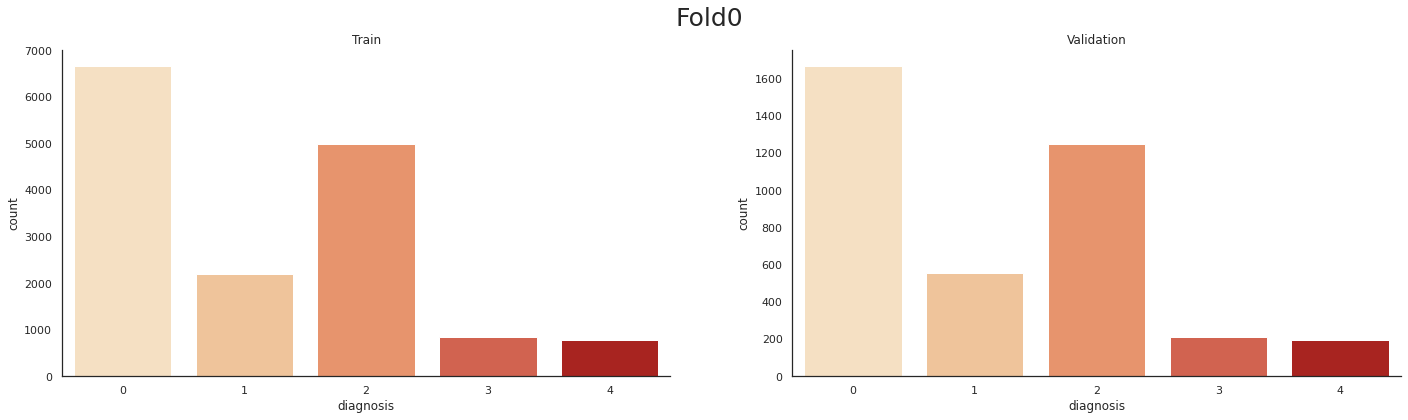
|  |  |
| --- | --- |
| Nhãn | Ngưỡng |
| 0 | < 0.5 |
| 1 | < 1.5 |
| 2 | < 2.5 |
| 3 | < 3.5 |

#### 1.1. Tiến hành

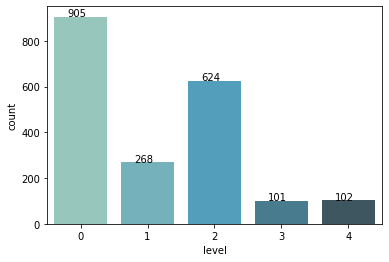
* Dữ liệu



*Dữ liệu huấn luyện (train)*

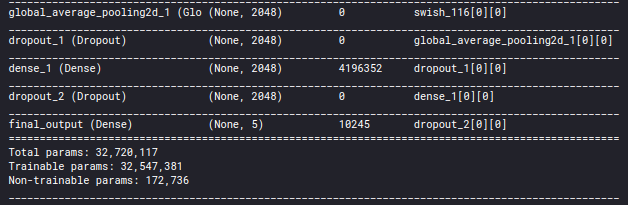


*Dữ liệu xác thực (validation)*

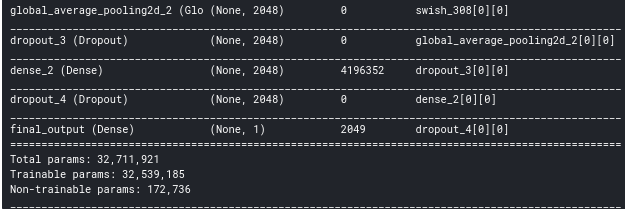


*Dữ liệu kiểm thử (test)*

* Tham số:
  + Batch size: 8
  + Hàm tối ưu: Adam
* Kiến trúc: sử dụng retrained weight: resnet50 weights tf dim ordering tf kernels notop.
  + **ResNet50** - 1: một vài layers cuối



* + **ResNet50** - 2: một vài layers cuối

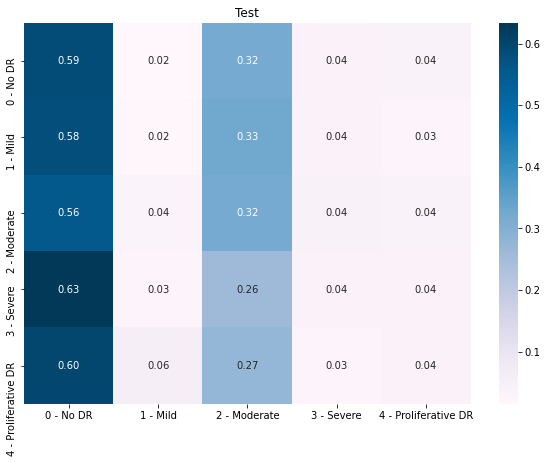


* Huấn luyện:
  + Warm up step với learning rate 1e-3: 5 epochs để tránh early overfitting.
  + Huấn luyện mô hình 30 epochs, tuy nhiên mô hình sẽ dừng huấn luyện (early stopping) nếu như validation loss có giá trị không tốt trong 5 epoch liên tiếp (giảm overfitting).

#### 1.2. Đánh giá

|  |  |  |
| --- | --- | --- |
|  | **ResNet50** - 1 | **ResNet50** - 2 |
| Best train accuracy | 0.6957 | 0.4839 |
| Best train loss | 0.7411 | 0.2926 |
| Best validation accuracy | 0.6530 | 0.4263 |
| Best validation loss | 0.8462 | 0.5761 |
| Test accuracy | 0.6560 | 0.5877 |

* Đánh giá lại trên bộ dữ liệu huấn luyện:
  + **ResNet50 - 1**:



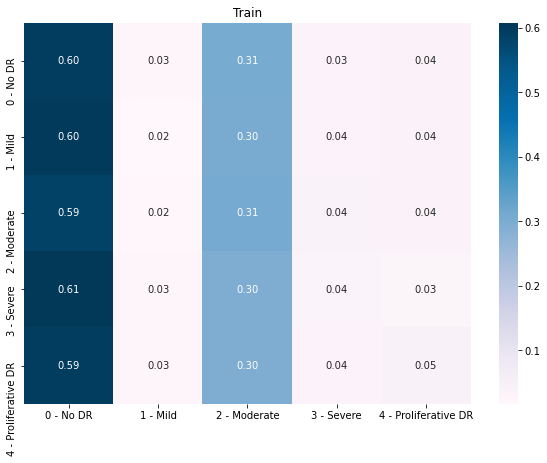
*Ma trận nhầm lẫn áp dụng tập train của ResNet50 -1*

* + **ResNet50 - 2**:



*Ma trận nhầm lẫn áp dụng tập train của ResNet50 -2*

* Đánh giá trên dữ liệu kiểm thử:
  + **ResNet50 - 1:**



*Ma trận nhầm lẫn áp dụng tập test của ResNet50 -1*

* + **ResNet50 - 2**:



*Ma trận nhầm lẫn áp dụng tập test của ResNet50 -2*

Có thể thấy cả hai: **ResNet50 - 1** và **ResNet50 - 2** đều cho ra độ chính xác trong quá trình huấn luyện lẫn kiểm thử (test) chênh lệch nhau không nhiều. Tuy nhiên, nếu đánh giá dựa trên ma trận nhầm lẫn, có thể thấy **ResNet50 -2,** tuy độ chính xác thấp hơn nhưng lại cho được kết quả tốt hơn. **ResNet50 - 1** chỉ dự đoán được các hình ảnh thuộc 2 nhãn có lượng dữ liệu lớn (0 và 2) (lý do tại sao dữ liệu không đối xứng qua đường chéo chính) trong khi **ResNet50 - 2** đã có thể phân bố hình ảnh về hầu hết các nhãn. Ngoài ra, các chỉ số train loss và validation loss của **ResNet50 - 2** tốt hơn **ResNet50 - 1** rất nhiều.

## 2. Kết luận

* Có thể thấy, được huấn luyện trên cùng một tập dữ liệu và tham số EfficientNetB5 ở bài báo cáo trước cho ra kết quả với độ chính xác, phân bố điểm dữ liệu về các nhãn cao hơn ResNet50 trên cả hai tập train và test.

#### 

|  |  |  |
| --- | --- | --- |
|  | EfficientNetB5 - 2 | ResNet50 - 2 |
| Best train accuracy | 0.7179 | 0.4839 |
| Best train loss | 0.2876 | 0.2926 |
| Best validation accuracy | 0.6291 | 0.4263 |
| Best validation loss | 0.5442 | 0.5761 |
| Test accuracy | 0.6294 | 0.5877 |