

# ShortReport

August 17, 2024

## 1 Tóm tắt kết quả

Sau đây em xin tóm tắt qua về phần kết quả ạ.

### 1.1 Thông tin chung:

*Running Experiment Heffann3997: Retrain Heffann3997*

Starting validation on train set of dataver0...

About the dataset:

Number of samples: 1441

Number of classes: 3

Accuracy: 0.9771

Precision: 0.9780

Recall: 0.9771

F1 Score: 0.9772

Top-2 Accuracy: 1.0000

---

Starting validation on valid set of dataver0...

About the dataset:

Number of samples: 363

Number of classes: 3

**\*\*Accuracy: 0.9063\*\***

Precision: 0.9096

Recall: 0.9063

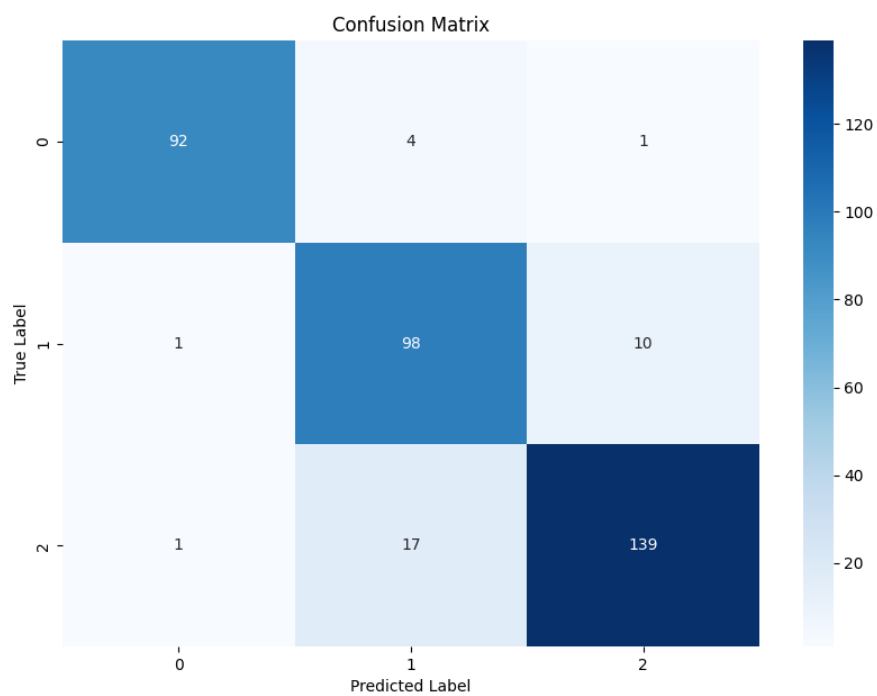
**\*\*F1 Score: 0.9072\*\***

**\*\*Top-2 Accuracy: 0.9835\*\***

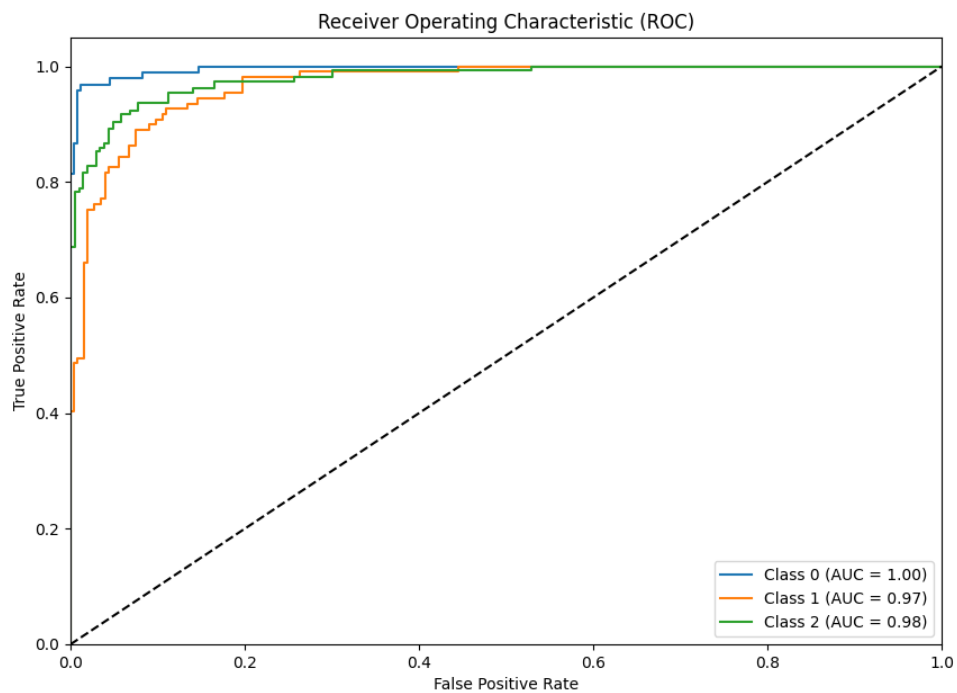
Experiment Heffann3997 completed.

### 1.2 Một số hình ảnh minh họa

Confusion matrix mô hình thu được trên tập validation



AUC các nhãn B2, B5, B6 - tương ứng là 0, 1, 2



Ngoài ra 1 số hình ảnh mô phỏng kết quả khác, hay kết quả dự đoán cho từng record (file csv) thầy có thể xem tại đây ạ: [link](#)

## 2 Tóm tắt ý tưởng xây dựng mô hình

Mô hình được hình thành bởi 2 module: + Module 1 mô hình H0\_EfficientNetB0: đây là 1 mạng CNN và không có lớp dense ẩn giữa lớp GAP (shape [-1, 1280]) và lớp dense đầu ra (shape [-1, 3]). Mô hình có hơn 4 triệu tham số. + Module 2 mô hình H3997\_ANN: đây là 1 mạng ANN có đầu vào là vector shape [-1, 39] (tức vector 39 chiều - là tổng hợp của 13 output từ module 1 bao gồm 1 ảnh gốc và 12 ảnh cắt theo lưới từ ảnh gốc). Mạng này có 3 lớp dense size lần lượt là 39, 97, 3.

Như vậy 1 ảnh gốc của bệnh nhân sẽ đi qua mô hình theo luồng như sau: 1. Ảnh gốc được resize về 1048x768 được ảnh (A). 2. 12 ảnh size 256x256 (B) sẽ được cắt theo lưới từ ảnh (A). 3. Như vậy ta có 13 ảnh từ (A) + (B). Resize 13 ảnh này về 224x224. 4. 13 ảnh này cho qua module 1 (H0\_EfficientNetB0). Mỗi ảnh sẽ cho ra 1 vector 3 chiều [liên quan về B2, liên quan về B5, liên quan về B6]. 5. Như vậy ta thu được 1 vector 13\*3=39 chiều (C). 6. Cho vector (C) qua module 2 (H3997\_ANN) và cuối cùng thu được 1 vector 3 chiều (D). Cho (D) qua hàm softmax ta thu được xác suất thuộc về 3 nhãn B2, B5, B6 tương ứng.

## 3 Module 1

### 3.1 Ý tưởng ra đời

Trước đây em đã từng thử train ra nhiều model CNN - mỗi model có đầu vào input là ảnh được xử lý theo 1 cách thức đã xác định nào đó. Sau đó sử dụng module 2 để tổng hợp các kết quả thu được từ các model của module 1. Kết quả cao nhất thu được là 87% tuy nhiên cách thức này có nhược điểm là dùng quá nhiều model nên rất nhiều tham số. => Em đã **rút gọn nhiều model của module 1 xuống còn thành 1 model CNN duy nhất - là model H0\_EfficientNetB0 với chỉ hơn 4 triệu tham số**. Em cũng đã thử sử dụng ViT, hoặc kết hợp CNN để thu nhỏ đặc trưng trước khi dùng với ViT hoặc ngược lại, ... Nhưng nhìn chung các model như vậy không tốt - do quá nhiều tham số so với quy mô của tập dữ liệu nên hiệu quả không cao.

### 3.2 Dữ liệu cho vào train

Dữ liệu dùng để train module 1 - model H0\_EfficientNetB0.

**Ảnh gốc (A)** -> Mô hình YOLO ver10 để detect cụm tế bào quan tâm trong ảnh -> **Vẽ bounding box của các cụm tế bào quan tâm vào (A) thì ta thu được ảnh (B)**. Bên cạnh đó ta cắt ra **8 cụm tế bào nổi bật nhất (C)** (theo diện tích) - nếu ảnh A không detect đủ 8 miếng thì sẽ lấy ngẫu nhiên từ ảnh cắt (6 miếng size 512x512 và 2 miếng size 768x768 - lý do chọn các size này vì nó tương thích với size 1048x768 đã resize từ (A) để đảm bảo cách lấy ảnh đồng nhất). Tiếp tục theo đó ta cắt **từ ảnh (B) size 1048x768 - đây là ảnh đã vẽ lên bounding box cụm tế bào quan trọng của ảnh**, ta cắt ra **12 miếng theo lưới size 256x256 - (D)**.

Bằng cách này ta đã thu được nhiều ảnh hơn. Lấy (A + B + C + D), ta thu được tổng thể gấp 1 + 1 + 8 + 12 = 22 lần số lượng ảnh có ban đầu. Đây có thể xem là 1 cách tăng cường bộ dữ liệu mà tập trung xoay quanh các vùng ảnh có cụm tế bào quan tâm, quá đó giảm vùng nhiễu, giúp model tập trung cập nhật trọng số hiệu quả hơn.

**Kết quả validation thu được của model H0\_EfficientNet này trên tập validation tập ảnh (A) - tức ảnh gốc chưa biến đổi gì lên ảnh là 87%. Có thể thấy model đã được dạy chỉ tập trung vào vùng quan trọng rất tốt và trên nhiều quy mô size ảnh 224x224 hay 256x256 hay 768x768, ...**

## **4 Module 2**

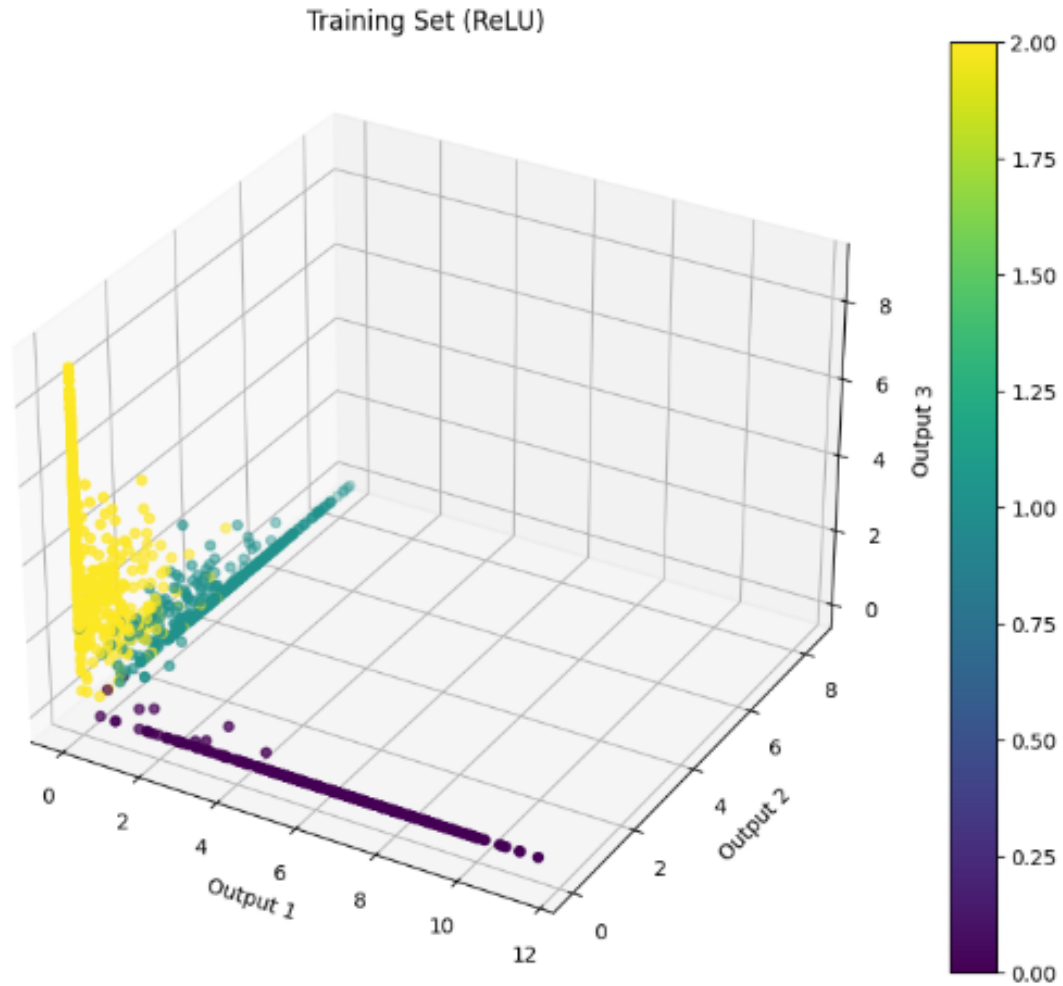
### **4.1 Ý tưởng ra đời**

Theo kinh nghiệm trước đó về việc ghép module 1 (nhiều model CNN) với module 2 (1 bộ kết hợp) cho kết quả cao hơn việc sử dụng từng model riêng lẻ. Bởi vậy em đã quyết định dùng thêm model H3997\_ANN như là 1 bộ tổng hợp các kết quả của model 2.

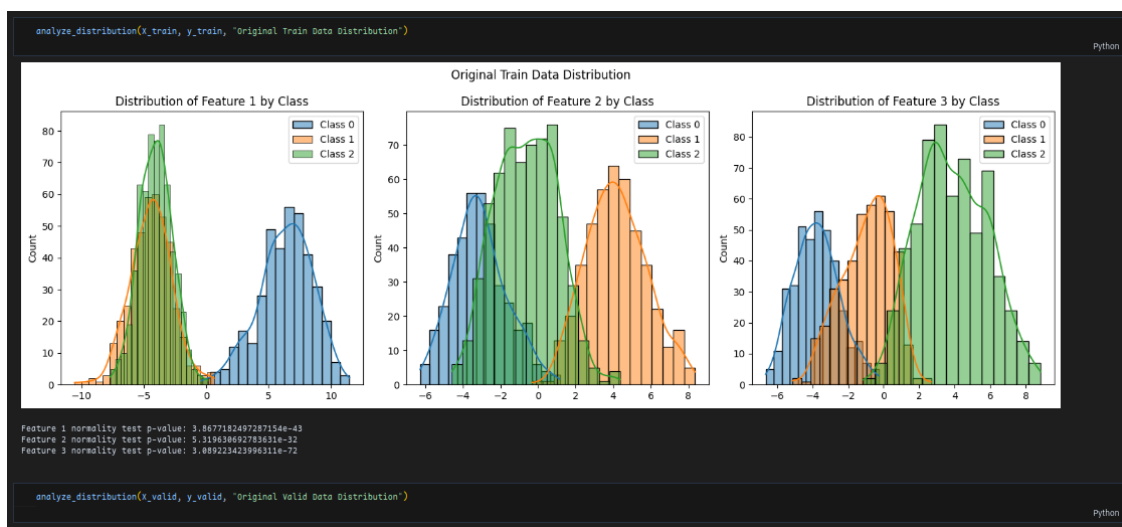
### **4.2 Dữ liệu cho vào train**

Trước hết em vẫn chỉ muốn dùng tối thiểu số input có thể để giảm khối lượng tính toán. Bởi vậy em sử dụng 3 output từ module 1. Tức ảnh (A) qua mạng cho ra 3 kết quả tương ứng với 3 nhãn B2, B5, B6. Kết quả không như kỳ vọng khi chỉ tăng thêm gần 1%, lên hơn 88%. Lúc này module 2 bao gồm các lớp dense (3, 97, 3)

Hình ảnh đầu ra của module 1 với tập train của ảnh gốc (A) - chưa biến đổi gì trên ảnh.



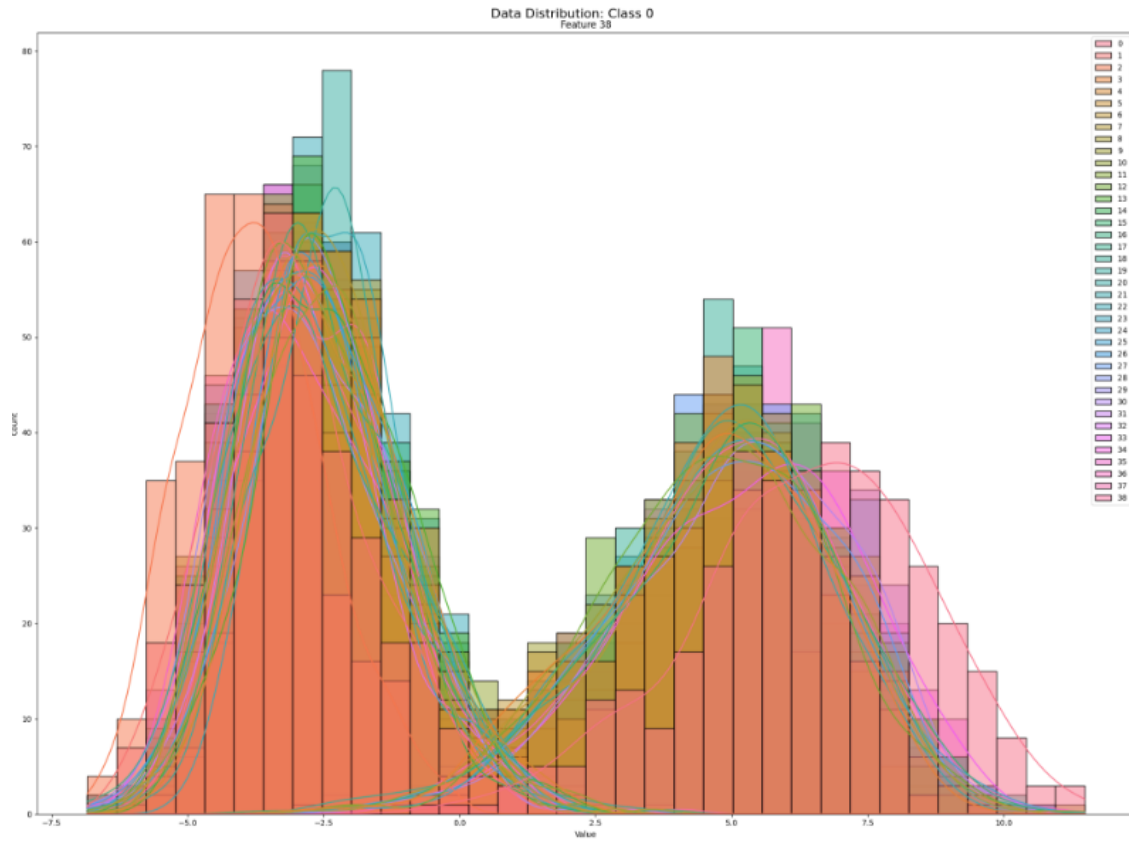
Do chỉ cải thiện thêm gần 1% nên em đã tìm hiểu hơn về phân phối của dữ liệu:

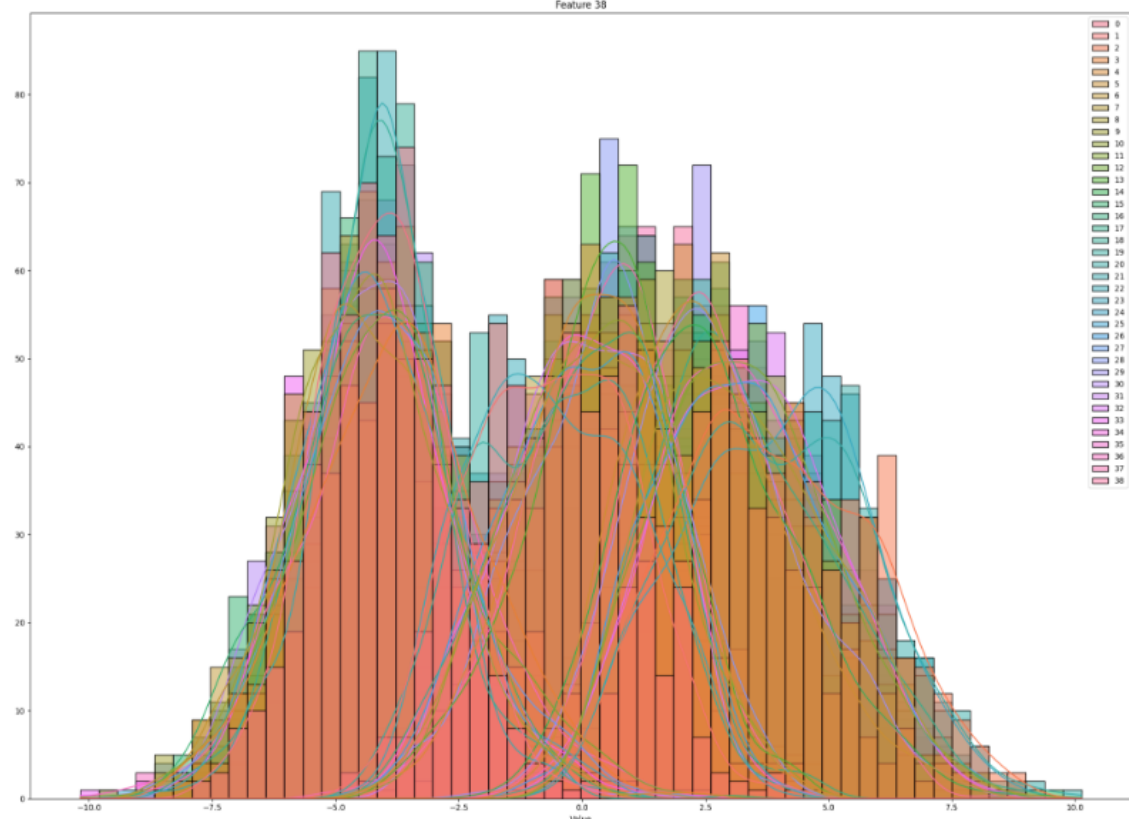
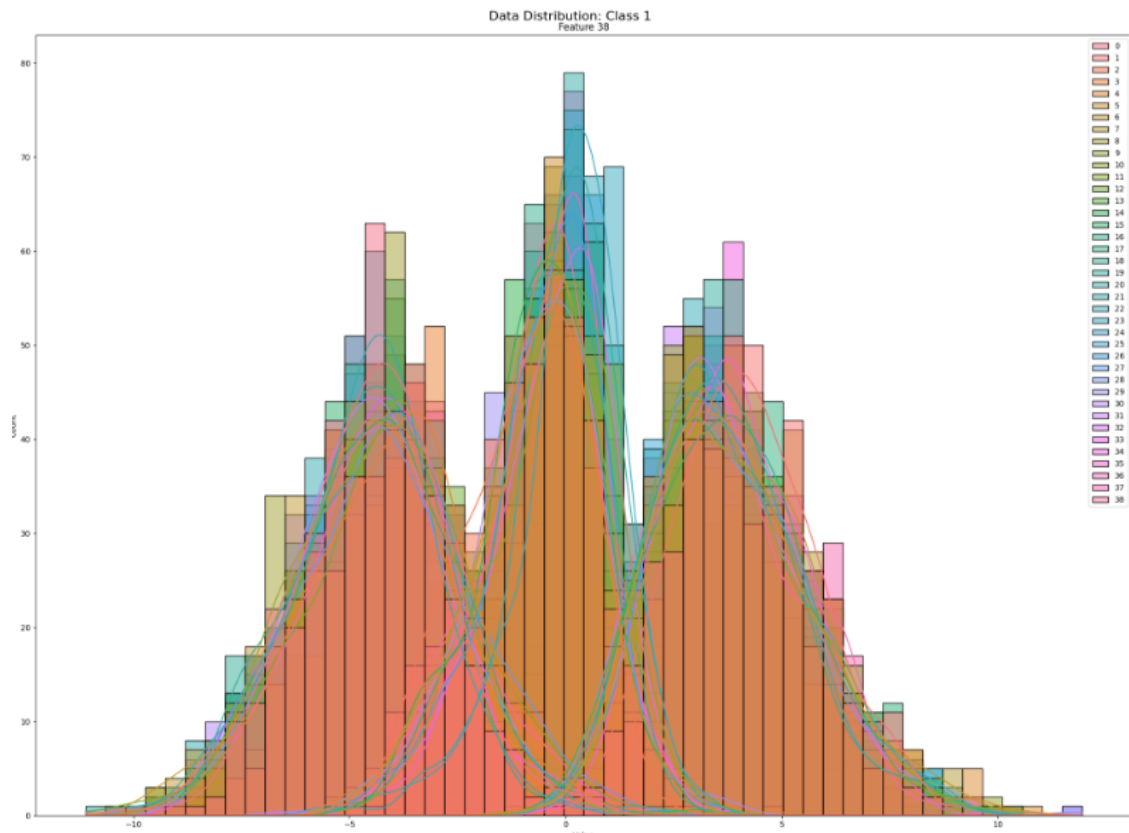


có thể thấy phân phối đầu ra gần như là phân phối chuẩn. Phân phối trên tập valid thì có phương

sai lớn hơn 1 chút. Tuy nhiên vẫn là phân phối chuẩn. Nhìn chung phân phối giá trị của nhãn B5 và B6 khá gần nhau. Bởi vậy em quyết định sử dụng thêm 12 ảnh đầu vào. Nâng tổng số ảnh đầu vào là 13 ảnh. Qua đó làm output ra của module 1 là 39. Thực hiện tăng cường hóa dữ liệu theo phân phối của tập train. Sau đó đem train module 2 - model H3997\_ANN. Kết quả thu được rất ấn tượng: 91%.

Hình dưới đây là về phân phối của 39 chiều input của module 2 xét trên tập train - output của tập train với ảnh gốc - chưa sửa đổi gì lên ảnh (tập A).





Chi tiết hơn về phân phối của các dữ liệu này thấy có thể xem tại notebook sau ạ: [link](#)

Báo cáo em soạn nhanh nên có thể có sai sót, mong thầy thông cảm. Em cảm ơn và xin ý kiến của thầy ạ.