BÁO CÁO ĐỒ ÁN CUỐI KÌ MÔN CS114.M21

Bài toán: Traffic Sign Classification

Thành viên nhóm:

1. Trần Thành 2052 1924

2. Lê Chí Thành 2052 1912

3. Vũ Thành An 2052 1057

1. Giới thiệu bài toán (kèm ví dụ input/output và ngữ cảnh ứng dụng)

Mô tả: Bài toán phân loại ảnh dựa trên 43 loại biển báo thông dụng, nhằm mục đích phục vụ cho xe tự hành hoặc giúp cho những người tham gia giao thông có thể nhận được những cảnh báo nhất định khi có sự xuất hiện của biển báo giao thông để có những hành vi xử lí phù hợp.

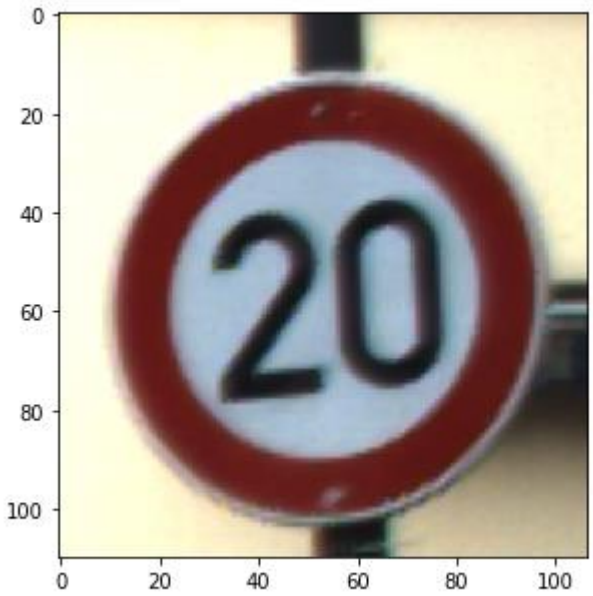
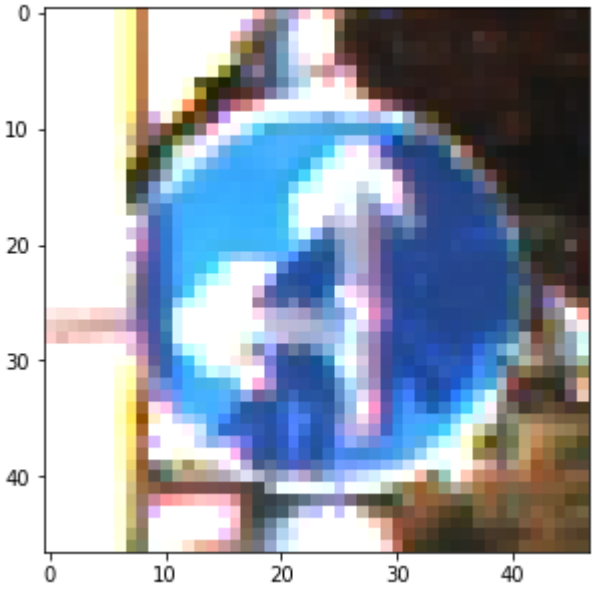
Input: Một tấm ảnh chứa biển báo giao thông với kích thước bất kỳ.

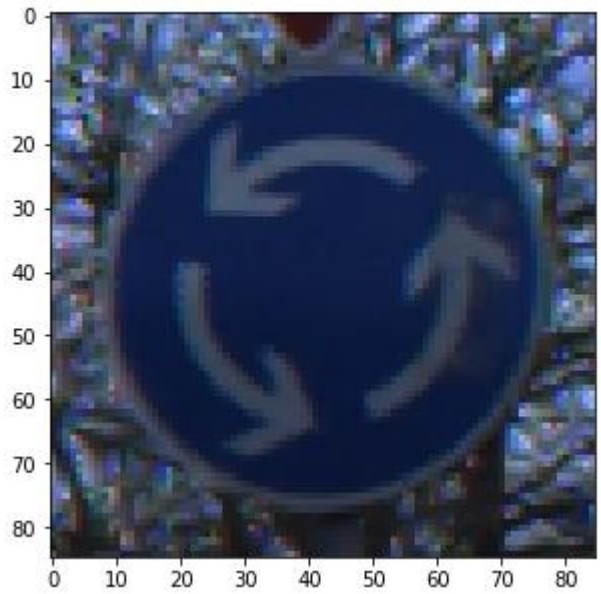
Output: Nhãn tương ứng với tấm ảnh là một trong 43 loại biển báo thông dụng được sử dụng để phân lớp.

Performance: Đánh giá bằng tỉ lệ giữa số ảnh dự đoán đúng nhãn với số ảnh được đem ra dự đoán (thang đo accuracy).

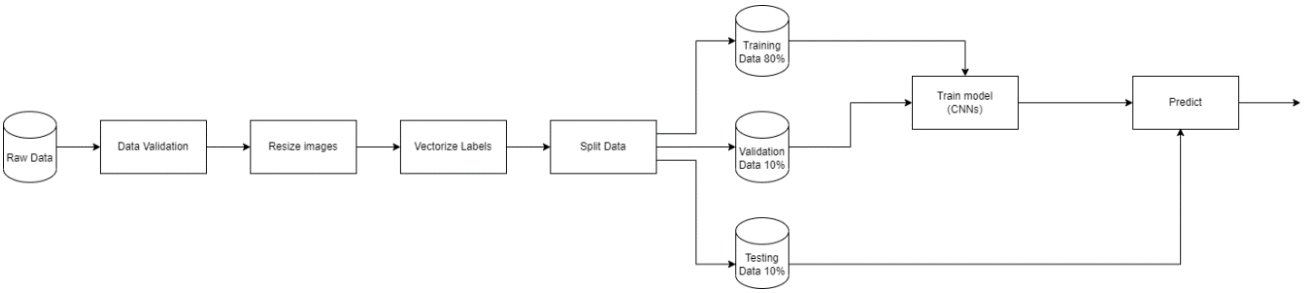
Ví dụ:

Input	Output

	Speed Limit 20 km/h
	Go straight or left

	Roundabout mandatory

2. **Trình bày pipeline cho bài toán (pre-processing) - feature extraction). (optional)**



Resize: resize ảnh về kích thước 32*32 pixels

Vectorize labels: sử dụng one-shot vector để biểu diễn các class trong bài toán

Model CNNs:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 32, 32, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 16, 16, 32)        0
_________________________________________________________________
```

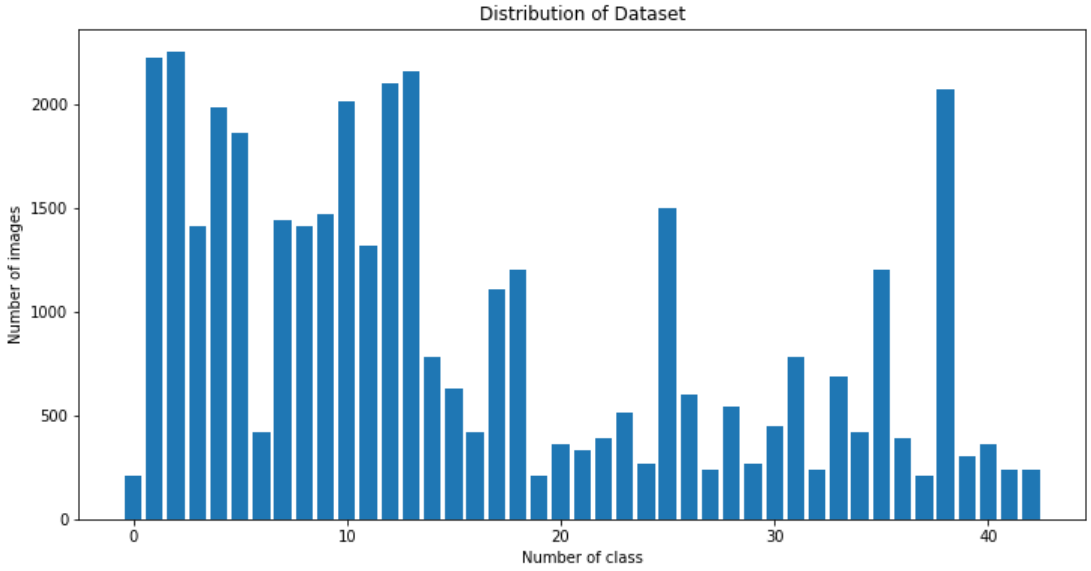
```
batch_normalization (BatchNo (None, 16, 16, 32)        128
_________________________________________________________________
dropout (Dropout)            (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 16, 16, 128)       36992
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 8, 8, 128)         0
_________________________________________________________________
batch_normalization_1 (Batch (None, 8, 8, 128)         512
_________________________________________________________________
dropout_1 (Dropout)          (None, 8, 8, 128)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 8, 8, 512)         590336
_________________________________________________________________
dropout_2 (Dropout)          (None, 8, 8, 512)         0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 8, 8, 512)         2359808
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 512)         0
_________________________________________________________________
batch_normalization_2 (Batch (None, 4, 4, 512)         2048
_________________________________________________________________
flatten (Flatten)            (None, 8192)              0
_________________________________________________________________
dense (Dense)                (None, 4000)              32772000
_________________________________________________________________
dense_1 (Dense)              (None, 4000)              16004000
_________________________________________________________________
dense_2 (Dense)              (None, 1000)              4001000
_________________________________________________________________
dense_3 (Dense)              (None, 43)                43043
=================================================================
Total params: 55,810,763
Trainable params: 55,809,419
Non-trainable params: 1,344
```

3. Thu thập dataset

Bộ dữ liệu gồm 39209 bức ảnh được chia ra thành ba tập train, test, val với tỉ lệ 8:1:1

Bộ dữ liệu được phân ra 43 class.

Bộ dữ liệu ở dạng ảnh đơn, phục vụ cho bài toán multi-classification. Ảnh ở dạng thang màu RGB, không có kích thước cụ thể chung mà tùy thuộc vào từng bức ảnh.



Nguồn dữ liệu: https://www.kaggle.com/code/raghav2002sharma/traffic-sign-detection-using-cnns-99-accuracy/data

4. Quá trình training - evaluation.

Training với số lượng eporch=1000, sử dụng hai bộ train và val.

Hàm tối ưu: sử dụng thuật toán adam

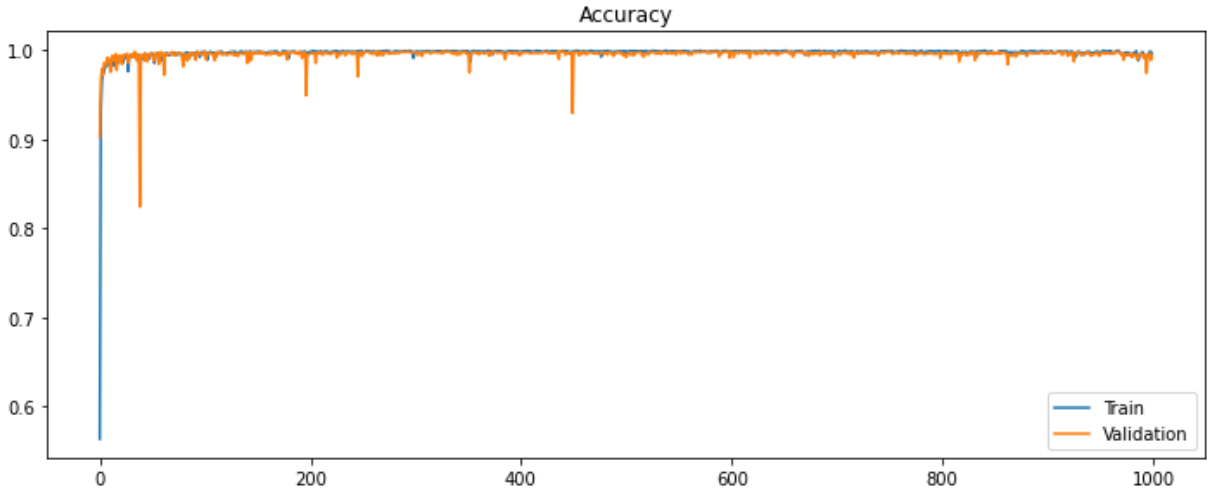
Hàm loss: categorical_crossentropy

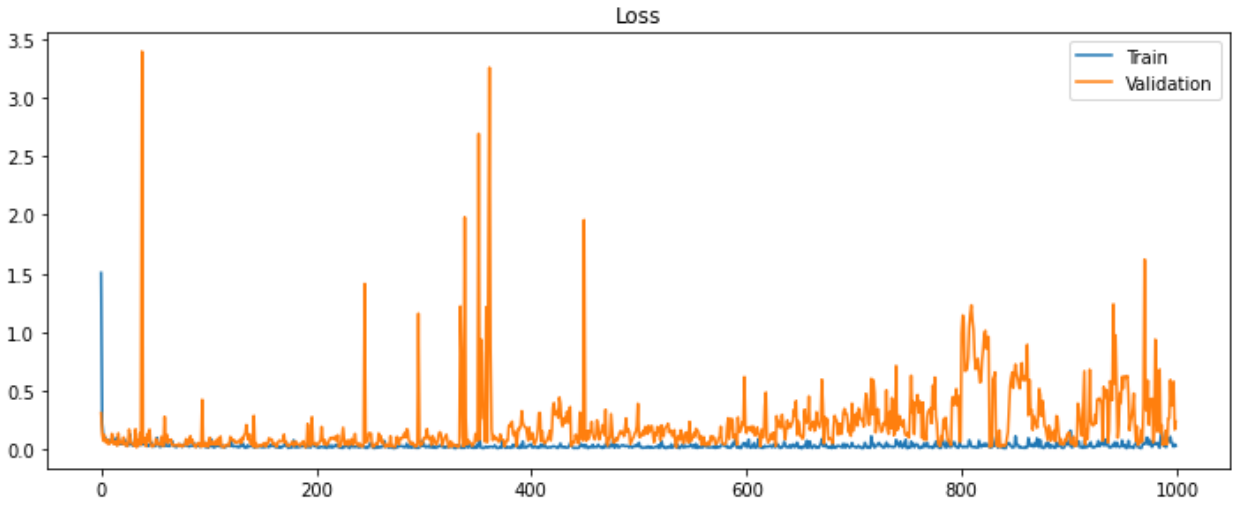
Batch_size = 64 (Mini-Batch)

```
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

history= model.fit(X_train,Y_train,
                   epochs=1000,
                   batch_size=64,
                   validation_data=(X_val,Y_val))
model.save('tfmodel')
```

```
Epoch 997/1000
491/491 [==============================] - 8s 17ms/step - loss: 0.0268 - accuracy: 0.9960 - val_lo
ss: 0.4591 - val_accuracy: 0.9957
Epoch 998/1000
491/491 [==============================] - 8s 17ms/step - loss: 0.0069 - accuracy: 0.9987 - val_lo
ss: 0.5720 - val_accuracy: 0.9964
Epoch 999/1000
491/491 [==============================] - 8s 17ms/step - loss: 0.0082 - accuracy: 0.9990 - val_lo
ss: 0.1604 - val_accuracy: 0.9888
Epoch 1000/1000
491/491 [==============================] - 8s 17ms/step - loss: 0.0169 - accuracy: 0.9955 - val_lo
ss: 0.2286 - val_accuracy: 0.9939
```

Train evaluation





Nguồn code tham khảo: https://www.kaggle.com/code/raghav2002sharma/traffic-sign-detection-using-cnns-99-accuracy/notebook

5. **Phân tích kết quả của model kèm thống kê, nhận xét và cho giải thích + hướng phát triển.**

Kết quả của model trên test, train, val sau khi train:

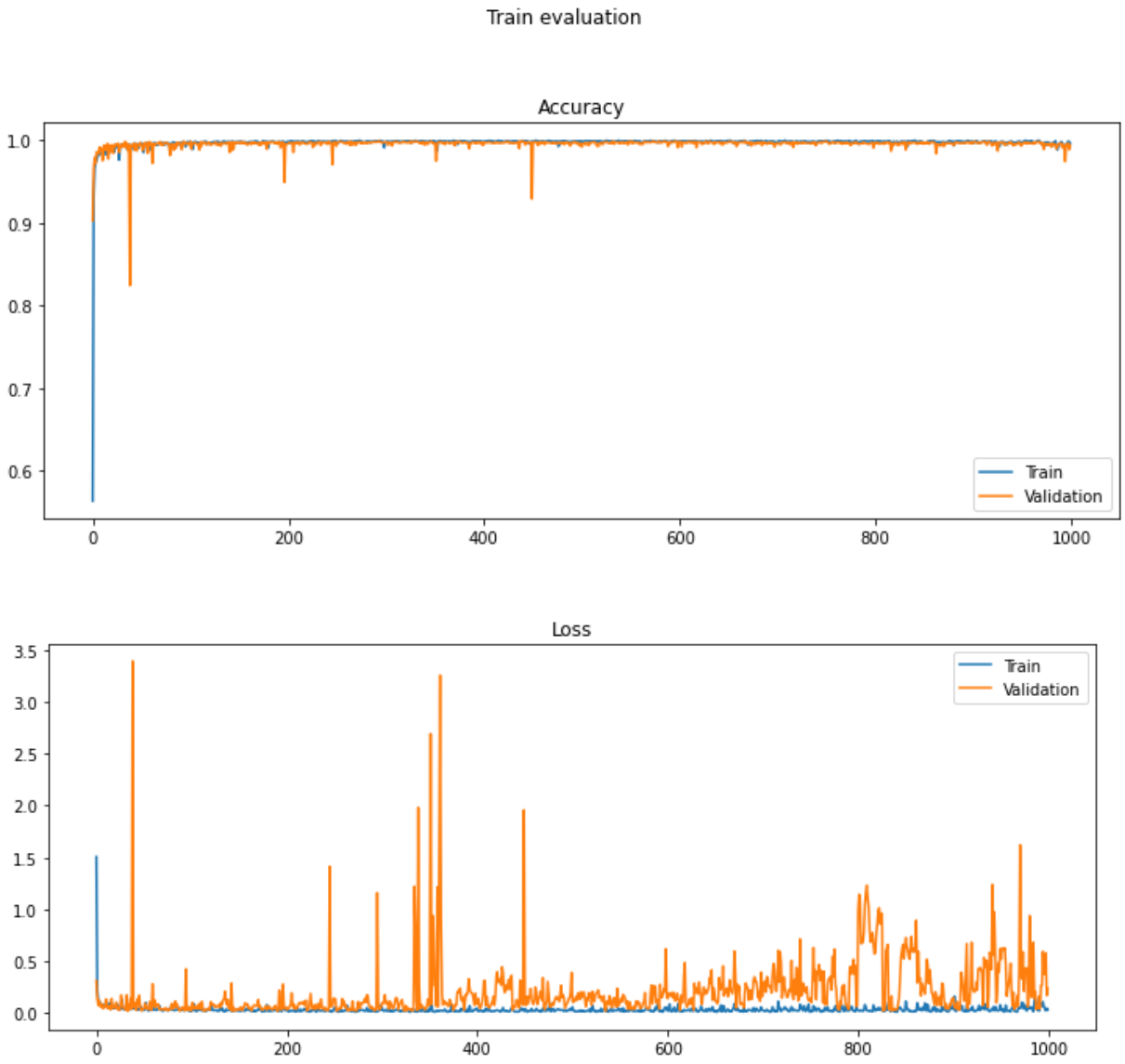
In [25]:

```
Y_test = np.argmax(Y_test,axis=1)

Y_pred= model.predict(X_test)

Y_pred = np.argmax(Y_pred, axis=1)

print('-Acuracy achieved: {:.2f}%\n-Accuracy by model was: {:.2f}%\n-Accuracy by validation was: {:.2
f}%'.
      format(accuracy_score(Y_test,Y_pred)*100,(history.history['accuracy'][-1])*100,(history.history
['val_accuracy'][-1])*100))
```

```
-Acuracy achieved: 99.39%
-Accuracy by model was: 99.65%
-Accuracy by validation was: 99.39%
```

Độ chính xác, trên tập test : 99.39%, trên tập train: 99.65% trên tập val: 99.39%

Thống kế như sau, qua 1000 eporchs:



Nhận xét, giải thích:

- Nhìn chung, trên biểu đồ accuracy:

+ Trên tập train có xu hướng tăng cực nhanh sau đó chững lại và ít có nhiều sự thay đổi tròn quá trình train

+ Trên tập val có xu hướng tăng cực nhanh sau đó chững lại và ít có sự thay đổi, tuy nhiên tồn tại những điểm khiến cho accuracy của val bị giảm. Suy đoán ngay tại những điểm này khiến cho model có khả năng bị overfit. Những điểm đó chỉ diễn ra tập trung ở

giai đoạn eporch < 600, nên với số lượng train là eporch=1000, thì kết quả có cơ hội để tin tưởng được.

- Nhìn chung, trên biểu đồ loss:

+ Loss trên tập train giảm nhanh từ những eporch đầu tiên và có xu hướng ổn định qua từng lần train tiếp theo

+ Loss trên tập val có sự tăng giảm lên xuống nhiều qua từng lần train eporch, phần nào tương xứng với sự thay đổi lên xuống trên biểu đồ accuracy của tập val. Tuy nhiên, dựa trên tổng thể thì loss trên tập val cũng mang xu hướng giảm sau nhiều lần training model.

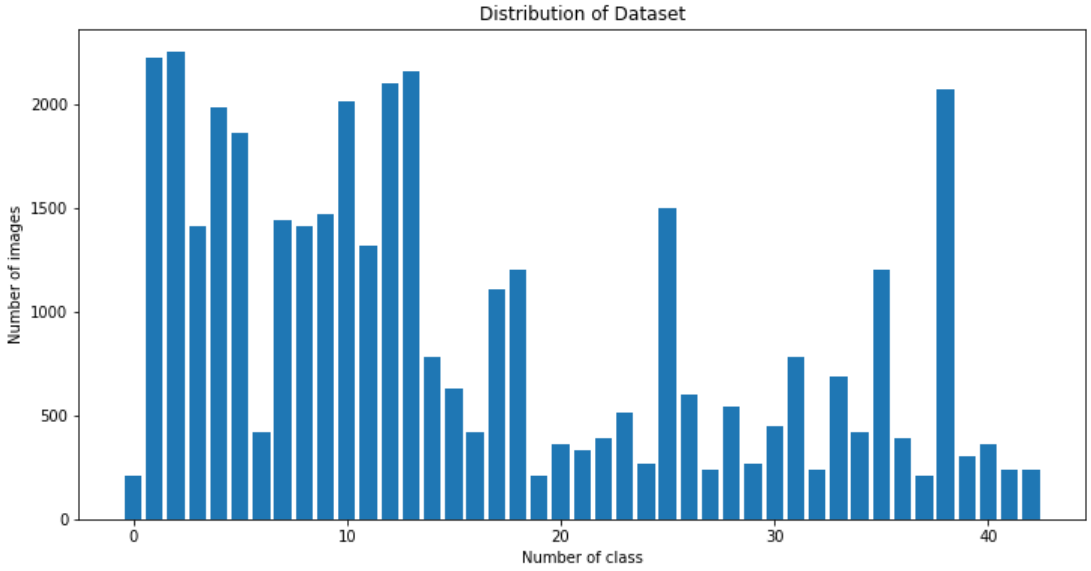
- Việc càng training model càng khiến cho accuracy và loss có khuynh hướng tốt hơn đó là điều mà model hướng đến, điều đó giải thích cho xu hướng chung rằng accuracy có khuynh hướng tăng và loss có khuynh hướng giảm sau nhiều lần training model. Những điểm trên biểu đồ cho thấy trên tập train, cả accuracy và loss đều có xu hướng ổn định trong khi đó, trên tập val thì hai con số này biến động quá nhiều, nguyên nhân là do:

+ training nhiều có thể dẫn đến tình trạng overfit làm cho accuracy và loss trên val bị giảm, tạo ra những điểm biến động

+ bộ dữ liệu dung để train chưa đồng đều giữa các class: cụ thể, sự chênh lệch giữa class có nhiều và ít images nhất là rất lớn: 2250 và 210 (xấp xỉ 10.7 lần). Điều này cũng xảy ra với những class còn lại. Dễ khiến model 'thuộc' những class nhiều ảnh mà những class ít ảnh lại bị ít quan tâm, khiến do model dự đoán ảnh dễ bị lệch.

2]:
```
print(min(length_class_list))
print(max(length_class_list))
```

```
210
2250
```



+ Bộ dữ liệu đã trải qua quá trình resize xuống kích thước 32*32 pixels để có thể thực hiện quá trình training model, những đặc trưng cơ bản tuy được giữ lại những những đặc trưng phụ bị đánh mất cũng có thể là một nguyên nhân.

Hướng phát triển:

Từ những phân tích bên trên

- Train model với số eporch lớn hơn nữa để tìm ra những điểm thích hợp cho model có thể hoạt động ổn nhất

- Tăng cường thêm bộ dữ liệu bằng cách tăng thêm lượng ảnh cho những class có số lượng ảnh ít. Giúp làm đồng đều tỉ lệ giữa các class. Có thể bỏ bớt ảnh ở những class có nhiều ảnh, tuy nhiên cách này không được khuyến khích.

- Resize về một kích thước ảnh lớn hơn hoặc áp dụng những phương pháp xử lí ảnh khác thậm chí có thể loại bỏ một số ảnh không thể xử lí được để hạn chế nhược điểm này.