



**PROJECT CUỐI KHÓA**  
**PYTHON & ML 02**

# **TRAFFIC SIGN RECOGNITION**

GVHD: Nguyễn Khánh Lợi

Sinh Viên: Phạm Đức Thế



# NỘI DUNG

## Giới thiệu

01

Giới thiệu bài toán, bộ dữ liệu  
Mục tiêu đề tài

## Thực nghiệm

02

Tiến hành thực nghiệm mô hình  
Đánh giá kết quả mô hình

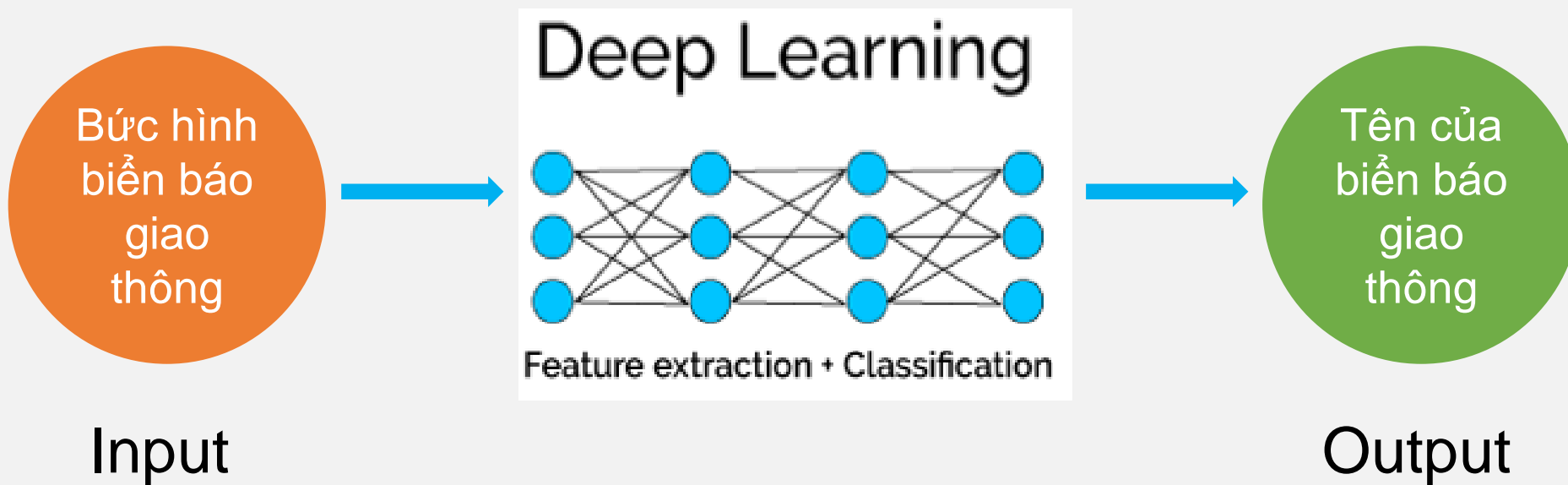
## Kết luận

03

Kết luận, đánh giá kết quả

# Giới Thiệu

## Bài Toán: Nhận dạng biển báo giao thông



# Giới Thiệu

## Mục Tiêu

- ✓ Áp dụng những kiến thức đã học để giải quyết các vấn đề thực tiễn trong cuộc sống.
- ✓ Tìm hiểu về kiến trúc, mô hình CNN (Convolutional Neural Network).
- ✓ Thực nghiệm mô hình CNN.

# Giới Thiệu

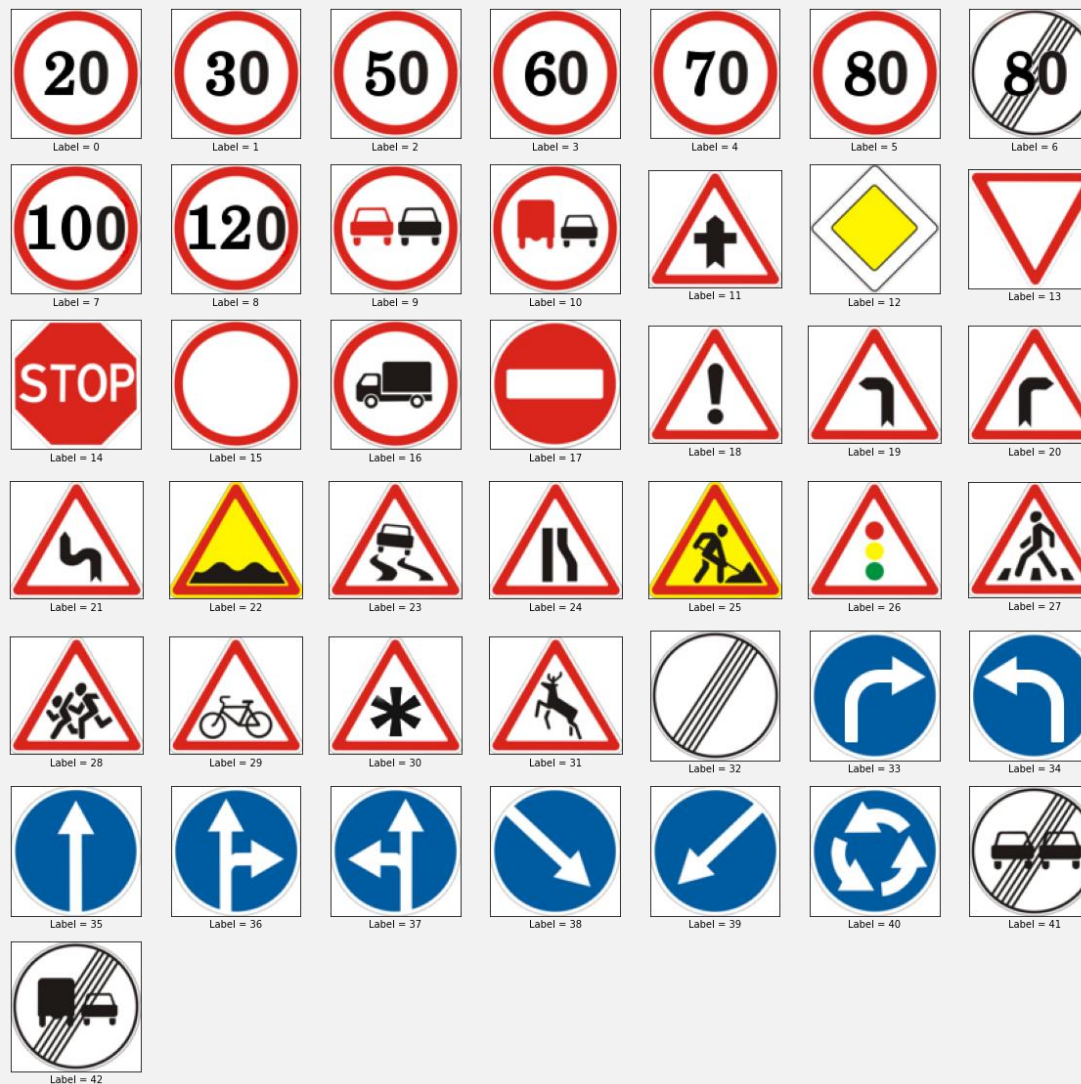
## Bộ dữ liệu

Thông Tin	Nội Dung
Tên bộ dữ liệu	German Traffic Sign Recognition Benchmark Nguồn: <a href="#">Kaggle</a>
Chức năng	Nhận dạng biển báo giao thông giúp hỗ trợ phát triển các hệ thống xe tự lái.
Kích thước bộ dữ liệu	Tập Train: 39,209 hình ảnh. Tập Test: 12,630 hình ảnh.
Số class	43
Tác giả	<a href="#">Johannes Stallkamp</a> , <a href="#">Marc Schlipsing</a> , <a href="#">Jan Salmen</a> , <a href="#">Christian Igel</a>

# Thực nghiệm

Xem bộ dữ liệu

Hình ảnh 43 classes ở  
dạng minh họa rõ nét





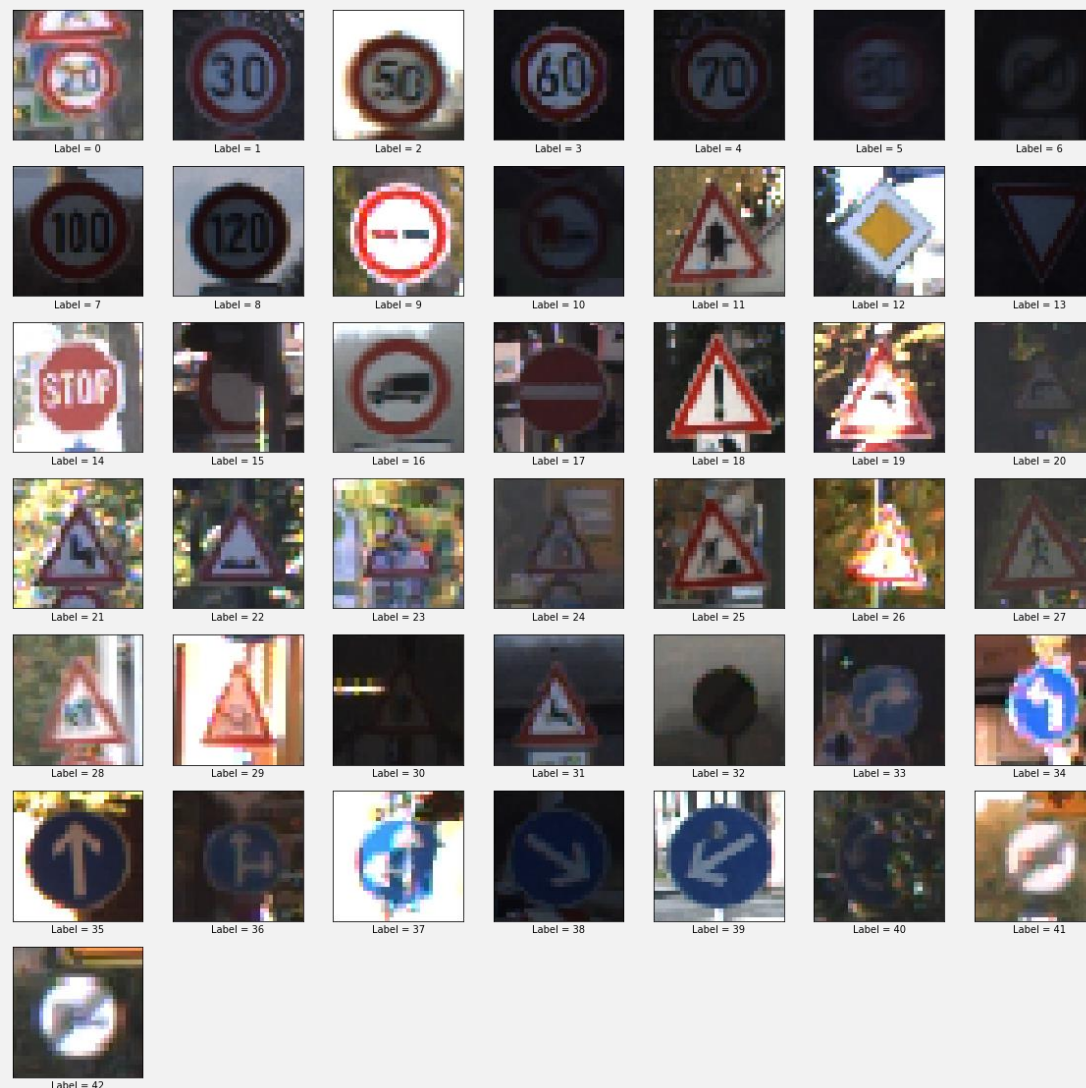
# Thực nghiệm

## Xem bộ dữ liệu

Hình ảnh 43 classes  
thực tế của bộ dữ liệu

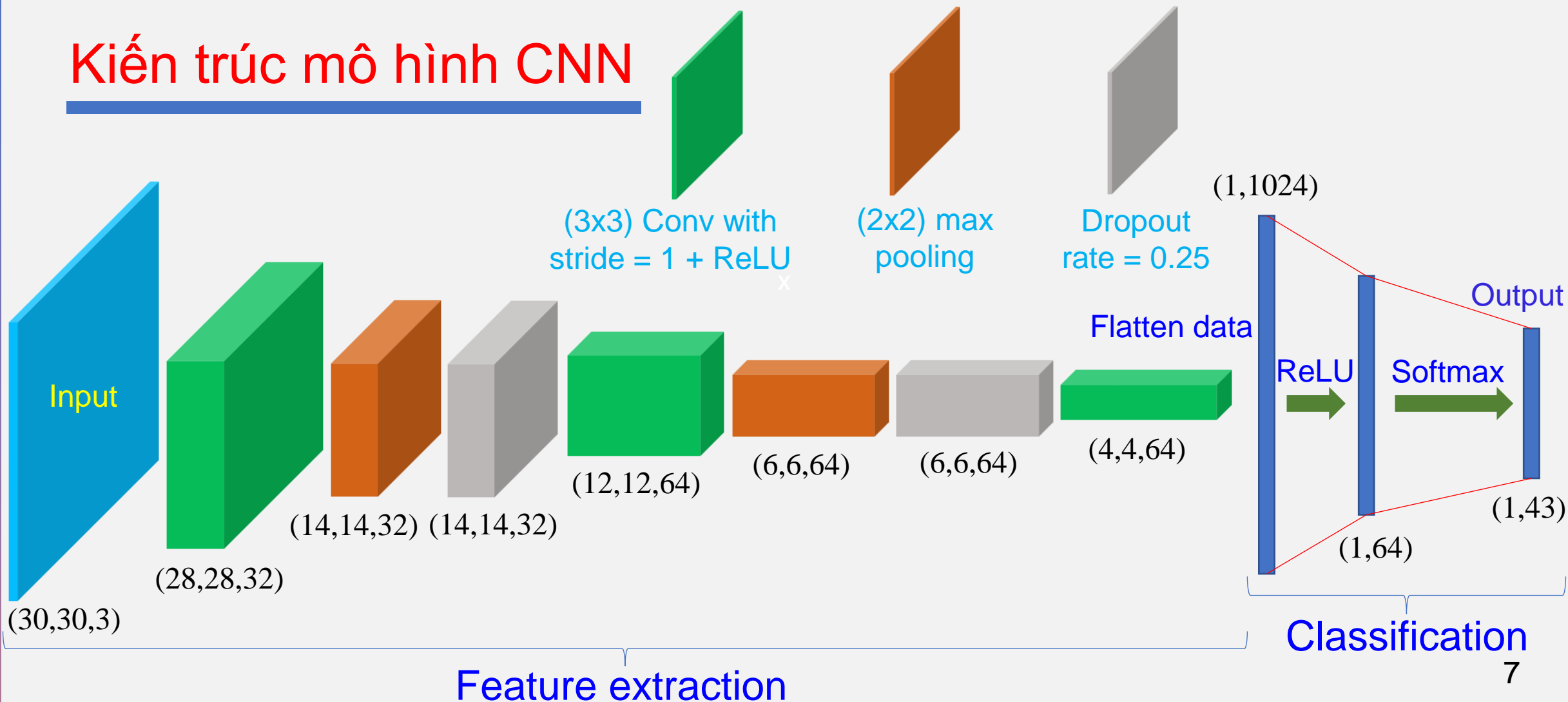
Mỗi hình ảnh có shape 30 x 30 x 3

Width  
Height  
Channel



# Thực nghiệm

## Kiến trúc mô hình CNN





# Thực nghiệm

## Xây dựng mô hình

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 32)	896
max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0
dropout (Dropout)	(None, 14, 14, 32)	0
conv2d_1 (Conv2D)	(None, 12, 12, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 64)	0
dropout_1 (Dropout)	(None, 6, 6, 64)	0
conv2d_2 (Conv2D)	(None, 4, 4, 64)	36928
flatten (Flatten)	(None, 1024)	0
dense (Dense)	(None, 64)	65600
dense_1 (Dense)	(None, 43)	2795

=====  
Total params: 124,715  
Trainable params: 124,715  
Non-trainable params: 0  
=====

```
model = Sequential()

# First Convolutional Layer
model.add(Conv2D(filters=32, kernel_size=3, activation='relu', input_shape=(30,30,3)))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))

# Second Convolutional Layer
model.add(Conv2D(filters=64, kernel_size=3, activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))

# Third Convolutional Layer
model.add(Conv2D(filters=64, kernel_size=3, activation='relu'))

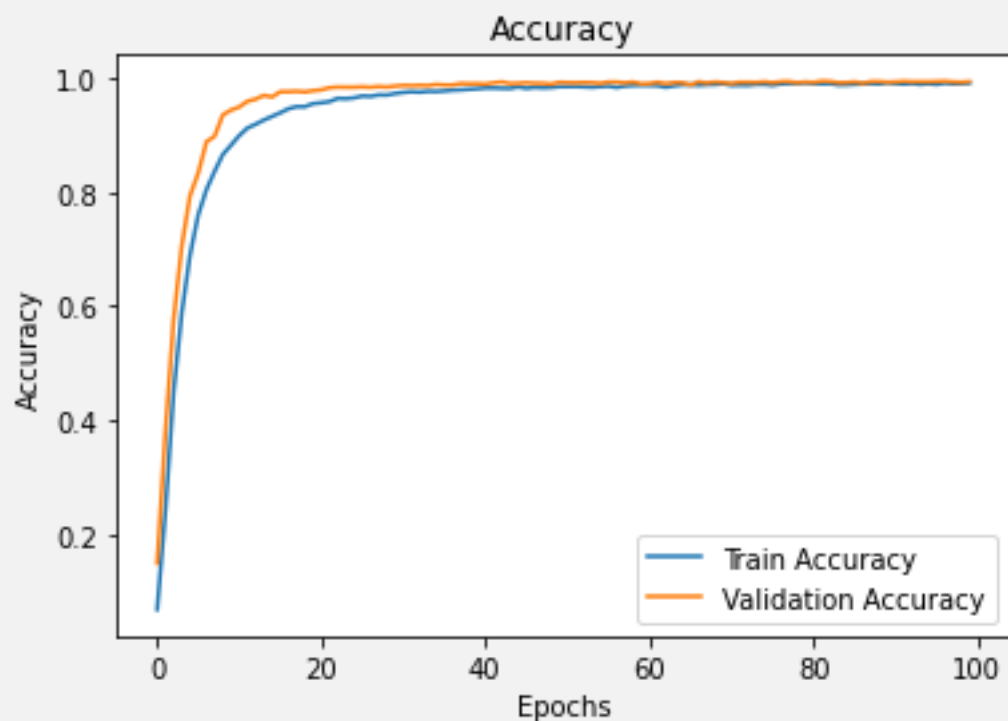
# Flattening the layer and adding Dense Layer
model.add(Flatten())
model.add(Dense(units=64, activation='relu'))
model.add(Dense(Classes, activation='softmax'))

model.summary()
```

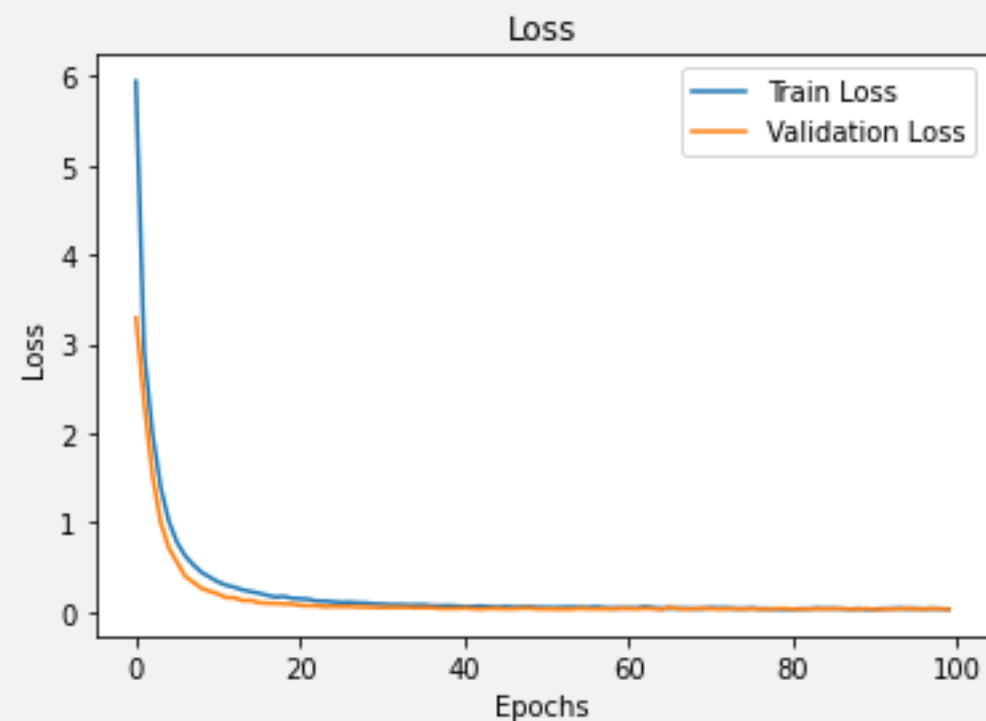
Có tổng cộng 124,715 parameters cần phải học.

# Thực nghiệm

## Accuracy vs Loss



Accuracy > 99%



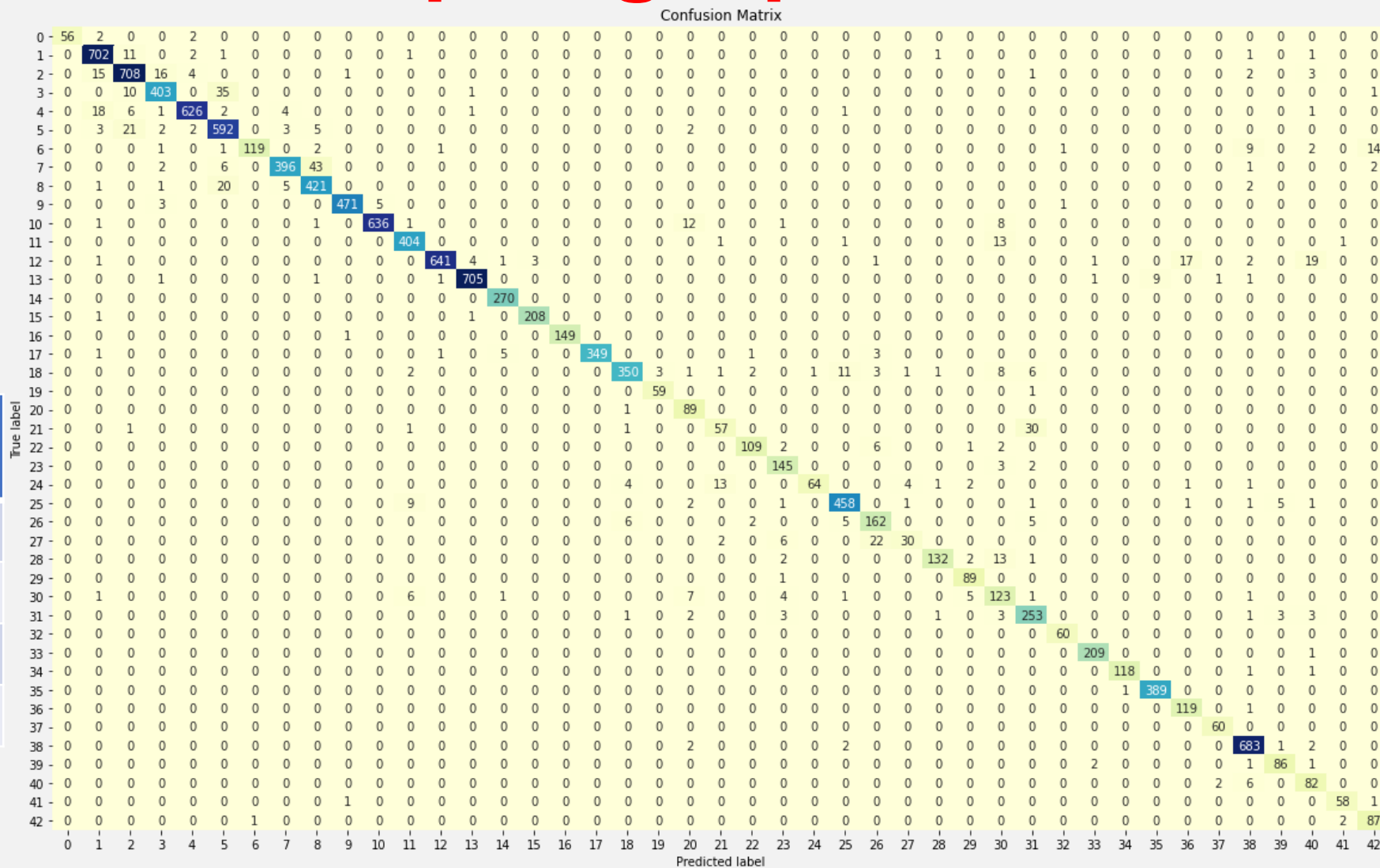
Loss < 0.03

# Thực nghiệm

Kết quả trên tập Test

Top 3 class bị dự đoán nhầm nhiều nhất:

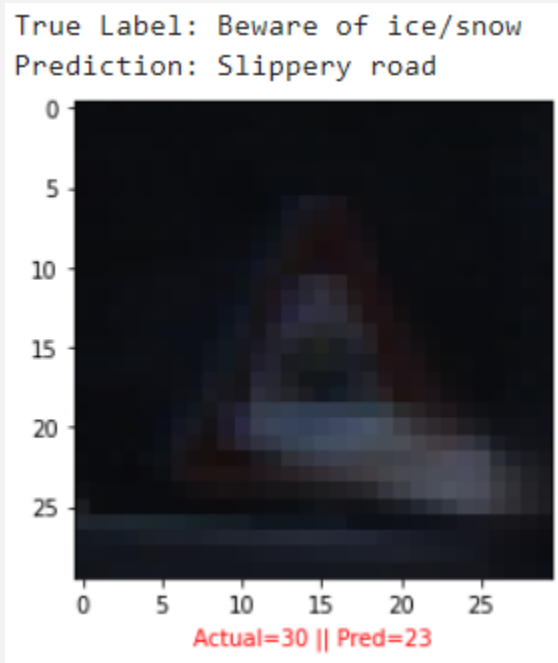
Label	Số lượng
7	54/450
12	49/690
3	47/450
...	...



# Thực nghiệm

## Kết quả trên tập Test

Accuracy > 94%





# Kết luận


## Kết luận

- ✓ Mô hình đạt được  $\text{Accuracy\_Train} > 99\%$ ,  $\text{Accuracy\_Test} > 94\%$ .
- ✓ Dự đoán tốt kể cả những hình ảnh bị thiếu sáng, bị mờ và bị che khuất.

## Hướng phát triển

- ✓ Tìm hiểu về các mô hình CNN khác như R-CNN, ResNet, ....
- ✓ Thu thập phát triển bộ dữ liệu để nhận diện các biển báo ở Việt Nam.

Cảm ơn Thầy và các bạn đã quan tâm theo dõi!



*thanks!*

Chúc Thầy và các bạn thật nhiều sức khỏe!