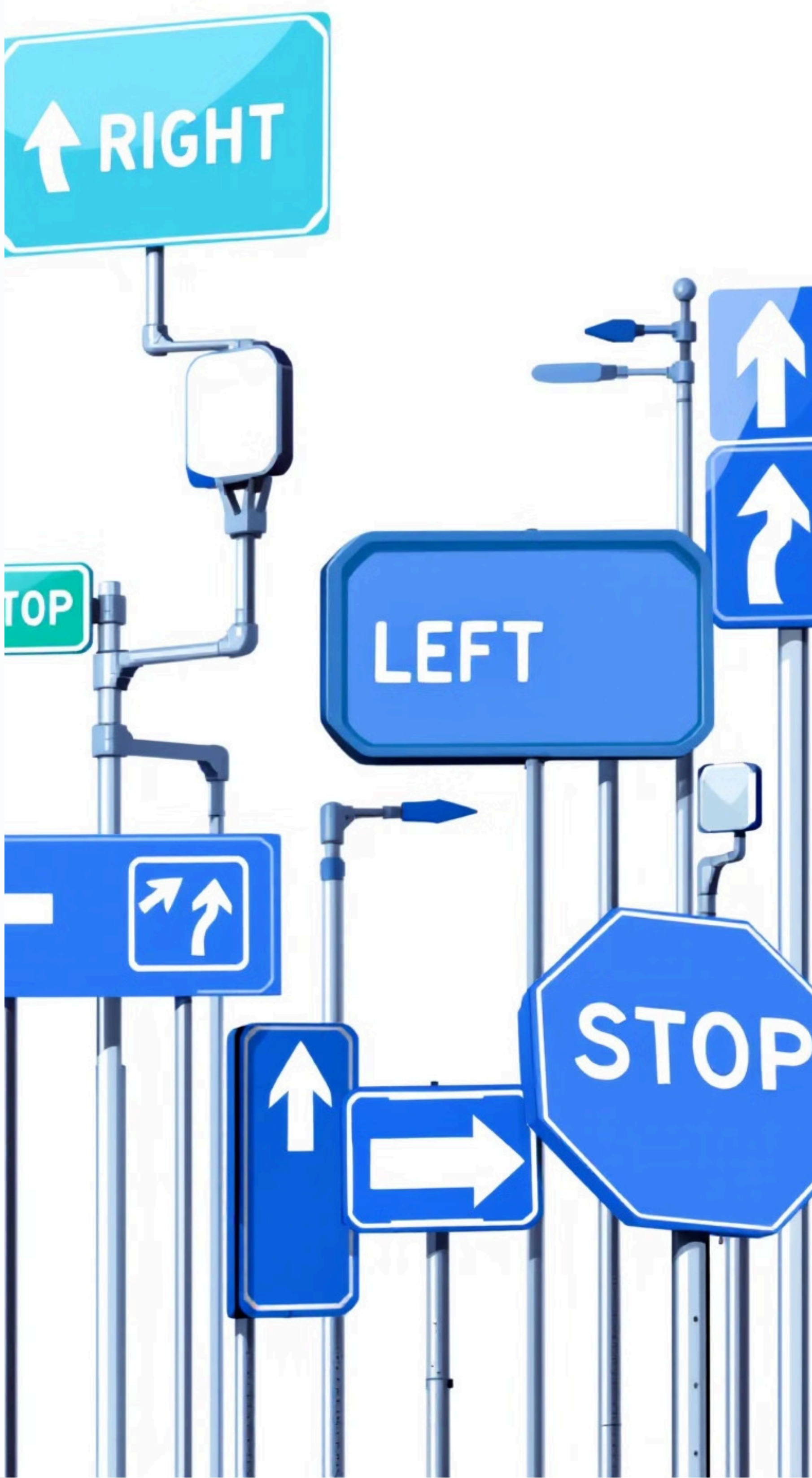
# DEEP-LEARNING PROJECT

# Road Sign Classification

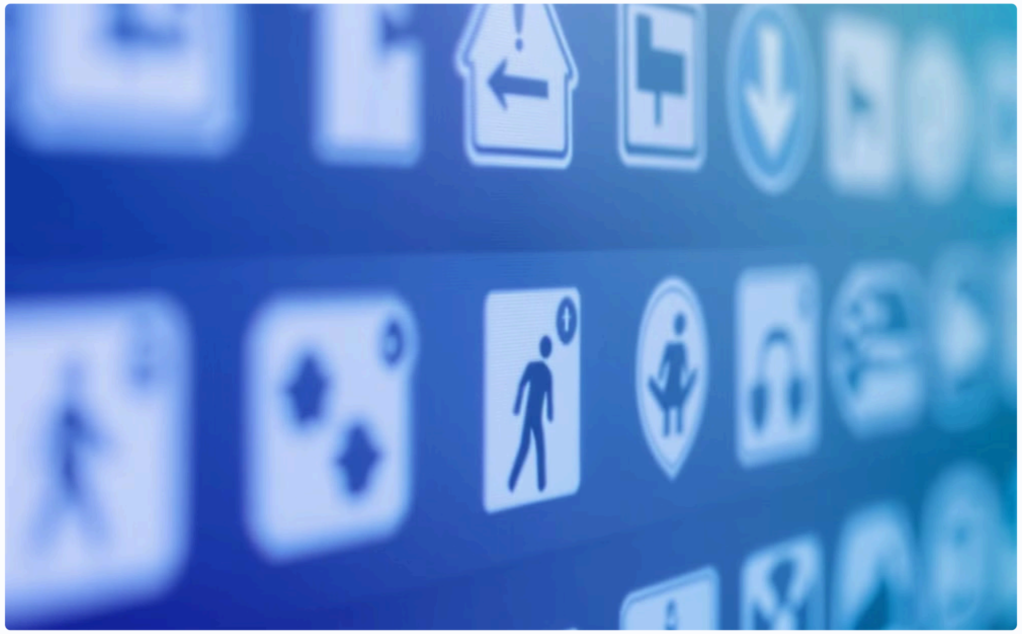
Road sign classification is a critical area in computer vision, involving the identification and categorization of signs to regulate traffic flow. This project analyzes the methodology of developing a deep learning model optimized for this task.



# Context

## Motivation

Automatic sign recognition is essential for the development of **autonomous driving systems** and **advanced driver-assistance systems (ADAS)**.



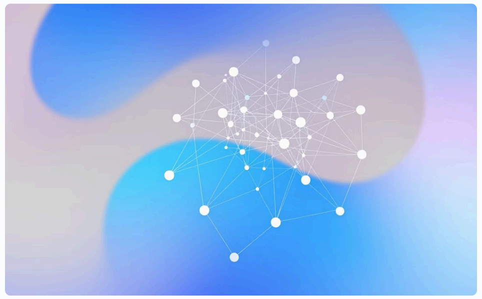
## Objectives

Design a high-performing deep learning model for **traffic sign classification**, using the **GTSRB dataset**. This model should be **accurate, robust**, and **capable of generalizing** in **diverse conditions**.

# Achievements

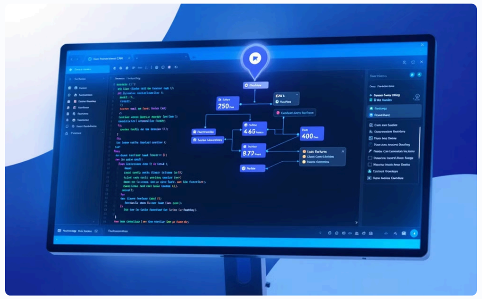
## Data Augmentation

Using techniques such as **rotation**, **flipping**, **zoom** and **cropping** to increase the **diversity of samples** and improve **generalization**.



## Exploration of Regularization

Using **Dropout** to **limit overfitting** and **Batch Normalization** to **stabilize and accelerate** training.



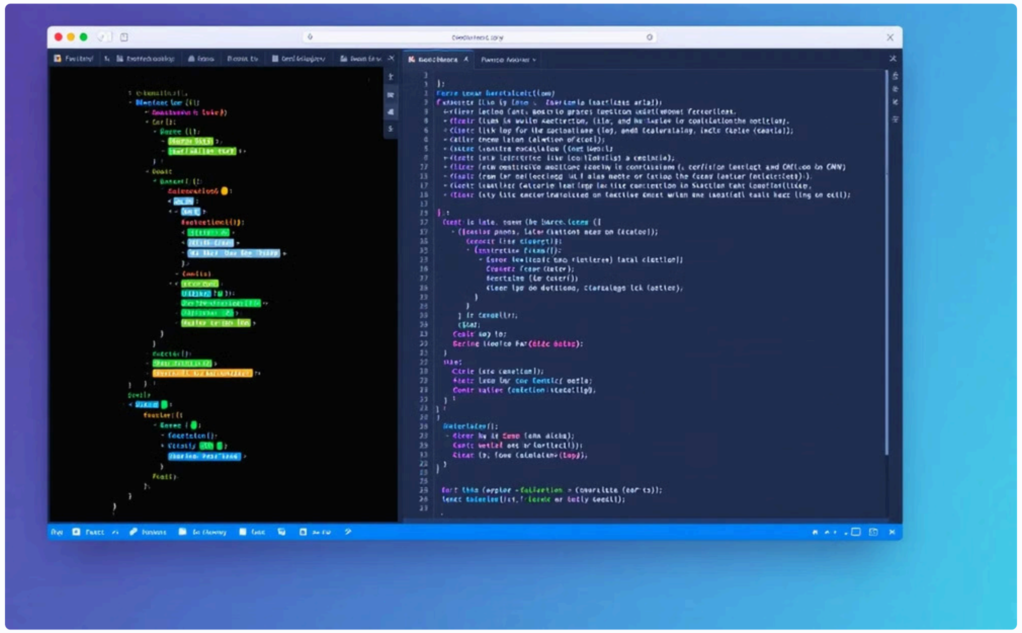
## Development of optimized models for classification

Optimization of a **CNN** model and an inspired **ResNet** model, including tuning of **hyperparameters** to increase **accuracy** and **robustness**.

## Experimentation under adverse conditions

**Testing** the models in **adverse environments** including **noise**, **movements** and **occlusions**, to **evaluate the robustness** of the model.

# Dataset and Data Preprocessing

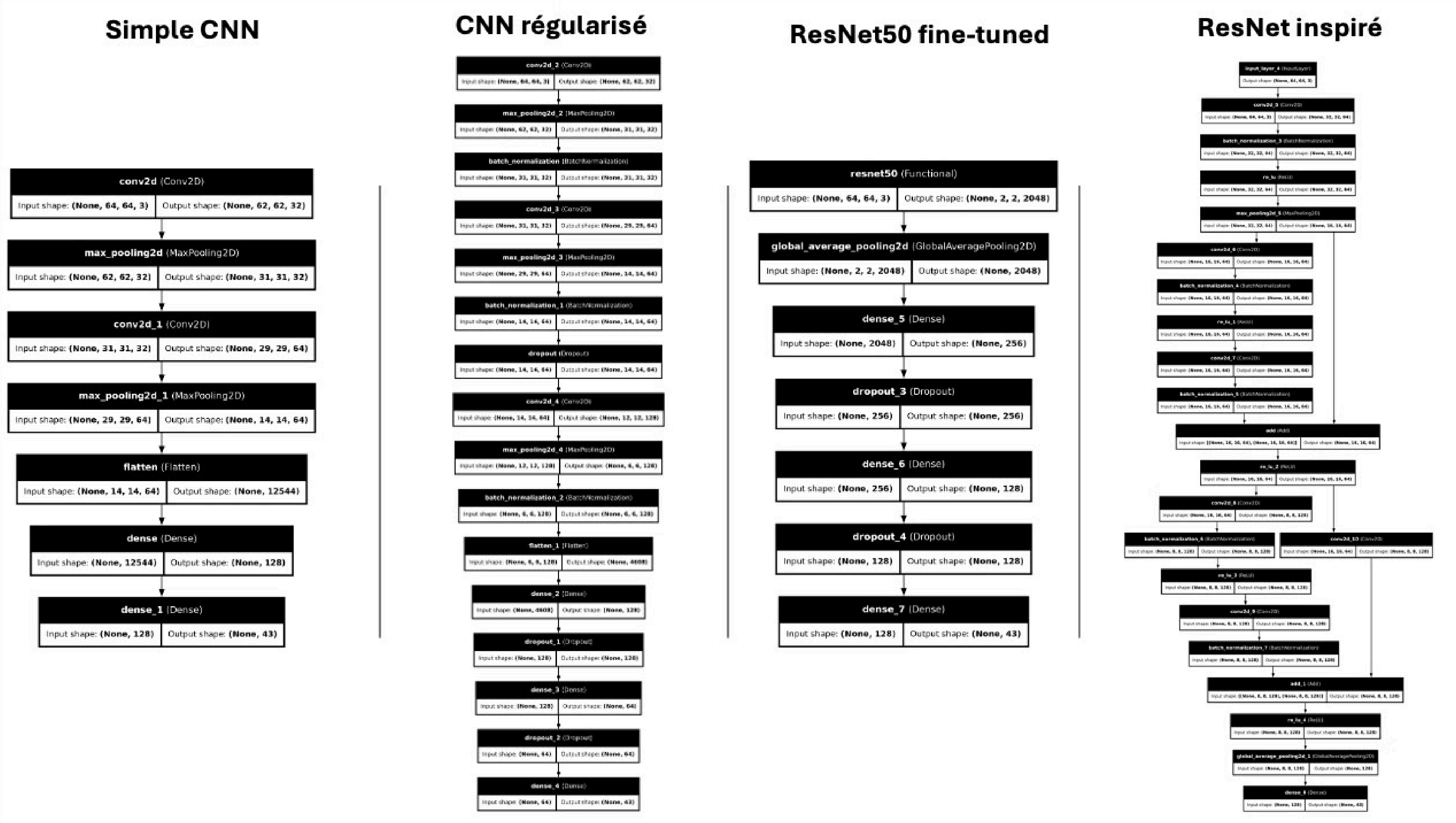
## Introduction to the GTSRB Dataset

The **GTSRB** dataset is composed of thousands of images of **traffic signs**. It is used for the training and evaluation of computer vision models. It contains **43 distinct classes** of traffic signs, each image having metadata such as the **shape**, **color** and **ID** of the pictogram.

## Data Preprocessing

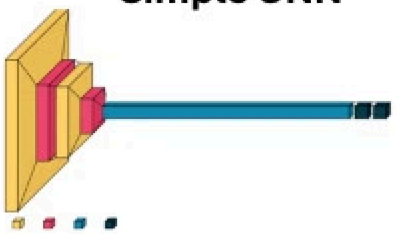
Data preprocessing involves steps such as **resizing** and **normalizing** images, **dividing the data** into training, validation and test sets, **augmenting** the data to improve model robustness and **encoding** the labels using one-hot encoding.

# Architecture

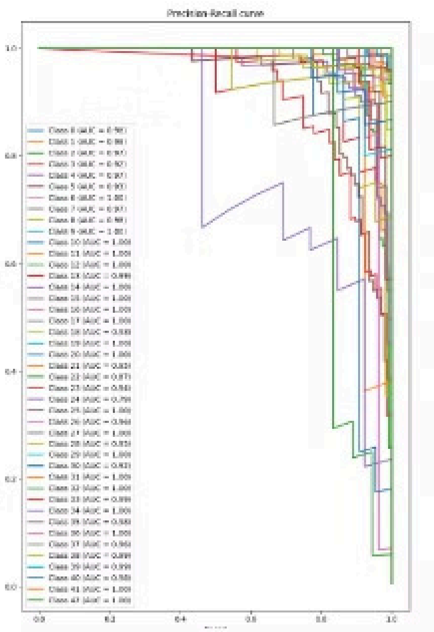


# Methodology and Models

### Simple CNN

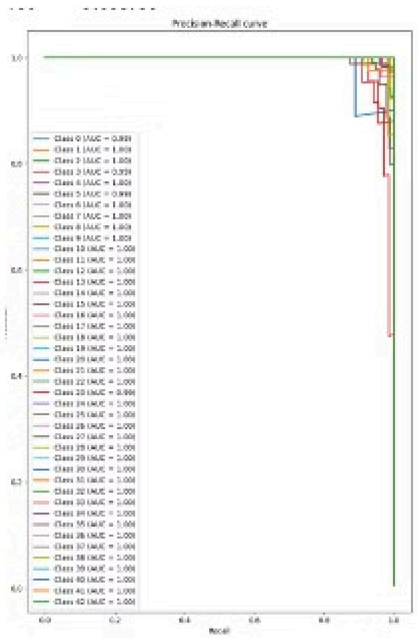


Précision : 93.4%
Rappel : 94.0%
F1-score : 93.9%
AUC-ROC : 0.9988
Précision Moyenne (AP) : 0.9750



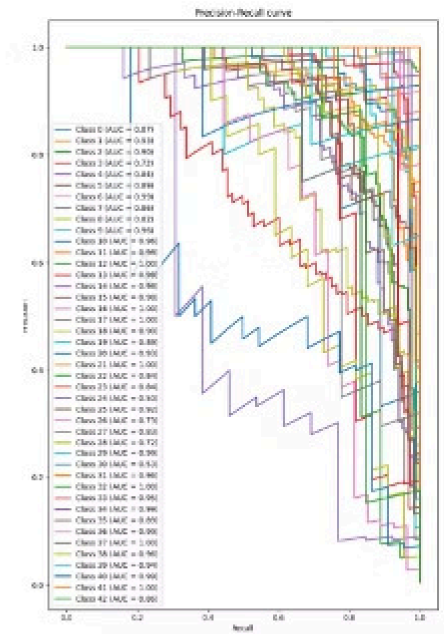

### CNN régularisé

Précision : 98.6%
Rappel : 98.6%
F1-score : 98.6%
AUC-ROC : 0.9999
Précision Moyenne (AP) : 0.9987

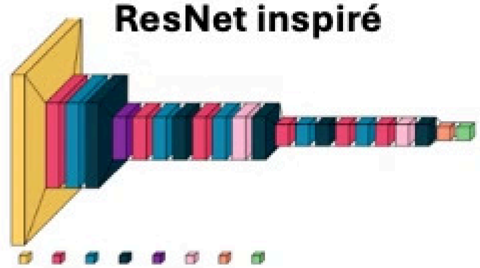



### ResNet50 fine-tuned

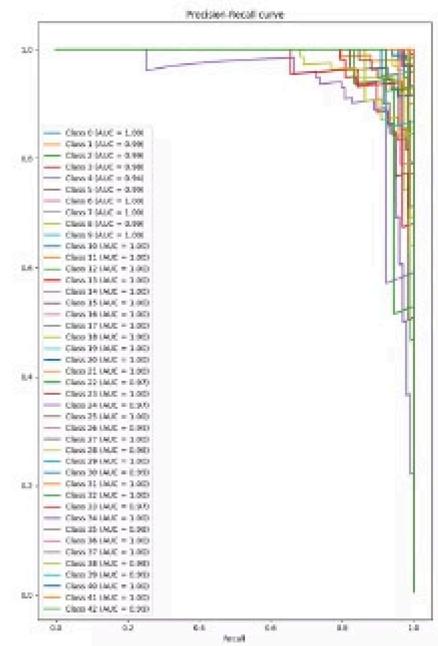

Précision : 84.2%
Rappel : 84.2%
F1-score : 84.2%
AUC-ROC : 0.9941
Précision Moyenne (AP) : 0.9036




### ResNet inspiré

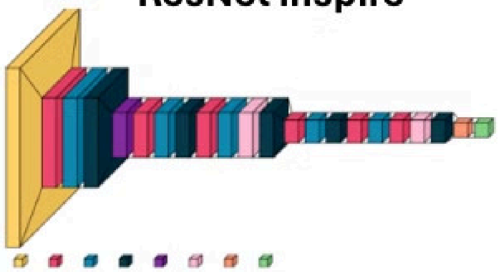


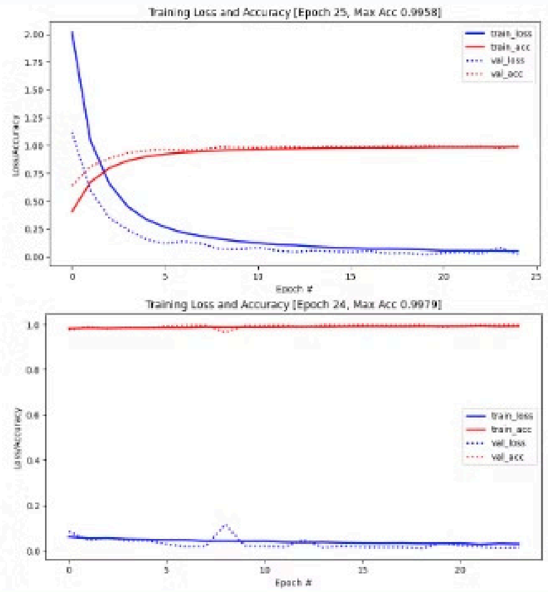
Précision : 94.7%
Rappel : 94.7%
F1-score : 94.6%
AUC-ROC : 0.9996
Précision Moyenne (AP) : 0.9930



# Training and Experimentation

## ResNet inspiré





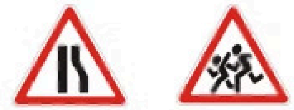


### Performance

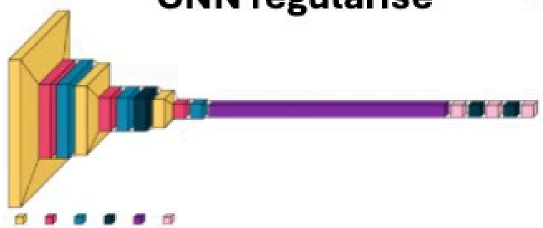
--- Performance Metrics for resnet_inspired_model ---

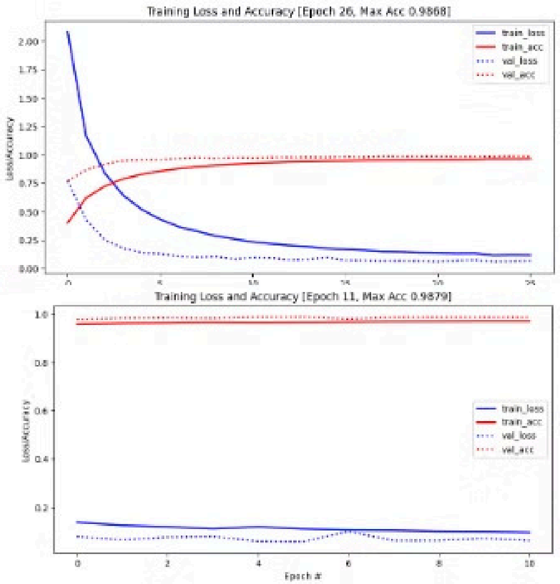
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| accuracy | 0.994000 | 0.994000 | 0.994000 | 0.994000 |
| macro avg | 0.996000 | 0.995000 | 0.996000 | 1895.000000 |
| weighted avg | 0.994000 | 0.994000 | 0.994000 | 1895.000000 |
| AUC-ROC | 0.999997 | 0.999997 | 0.999997 | 0.999997 |
| Average Precision | 0.999859 | 0.999859 | 0.999859 | 0.999859 |

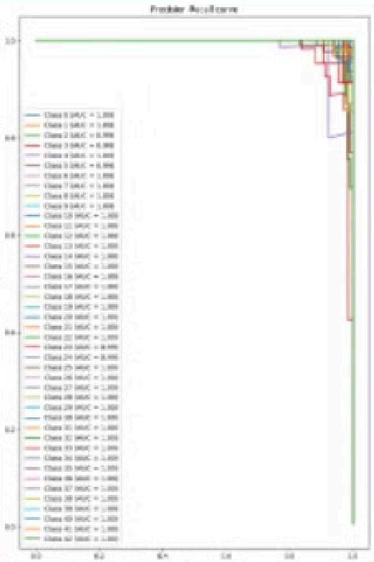
### Difficultés observées



## CNN régularisé





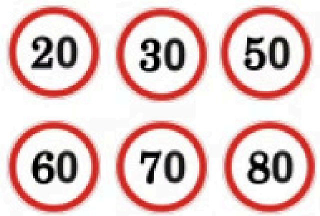



### Performance

--- Performance Metrics for cnn_regularized_model ---

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| accuracy | 0.982000 | 0.982000 | 0.982000 | 0.982000 |
| macro avg | 0.985000 | 0.984000 | 0.984000 | 1895.000000 |
| weighted avg | 0.982000 | 0.982000 | 0.982000 | 1895.000000 |
| AUC-ROC | 0.999928 | 0.999928 | 0.999928 | 0.999928 |
| Average Precision | 0.998285 | 0.998285 | 0.998285 | 0.998285 |

### Difficultés observées



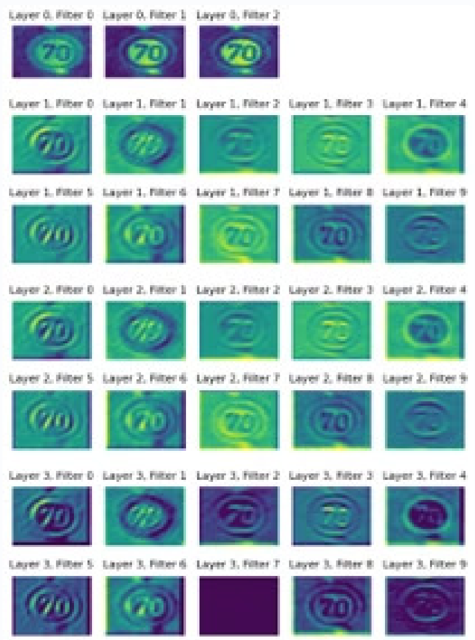
# Results Analysis

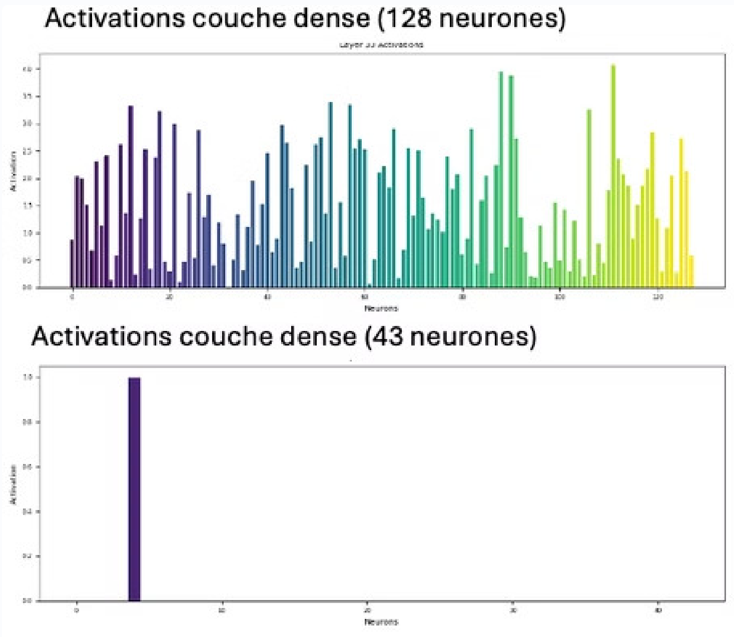
## Layer Activations

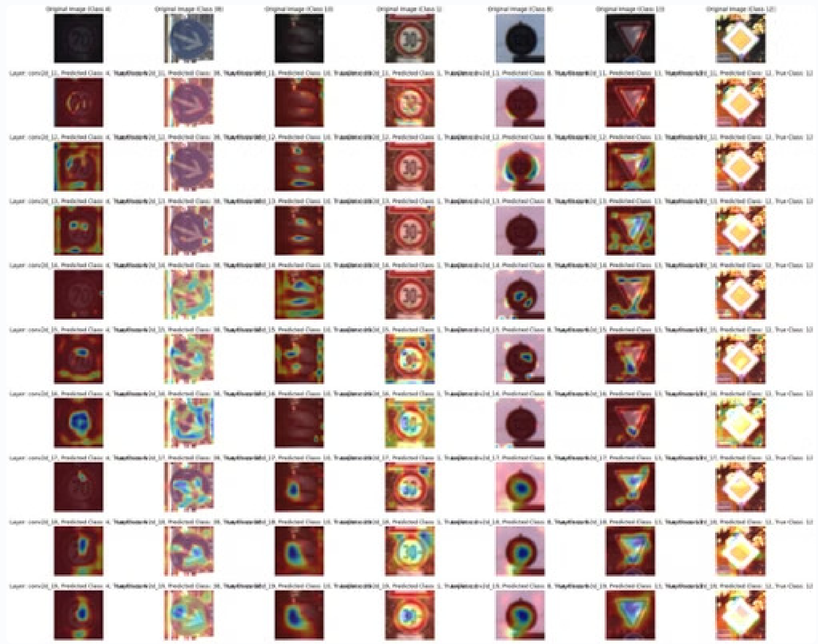
**Activations** of a CNN's layers reveal the **specific features learned** at each stage. Visualizing these activations helps **understand the visual patterns detected** and identify issues like **overfitting**.

## Grad-CAM Analysis

The **Grad-CAM** technique allows to visually identify the **areas that most influence the model's prediction**, thus evaluating the relevance of its choices.

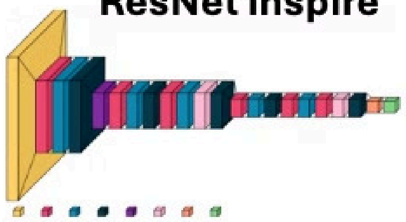




# Adversarial Testing

**ResNet inspiré**



Not retrained

--- Performance Metrics for resnet_inspired_model ---

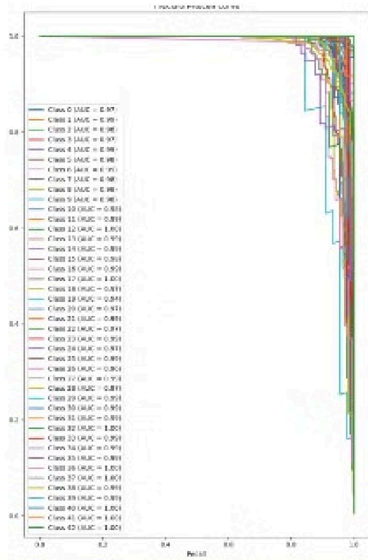
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| accuracy | 0.944000 | 0.944000 | 0.944000 | 0.944000 |
| macro avg | 0.949000 | 0.945000 | 0.945000 | 9475.000000 |
| weighted avg | 0.948000 | 0.944000 | 0.944000 | 9475.000000 |
| AUC-ROC | 0.998999 | 0.998999 | 0.998999 | 0.998999 |
| Average Precision | 0.984939 | 0.984939 | 0.984939 | 0.984939 |

Retrained

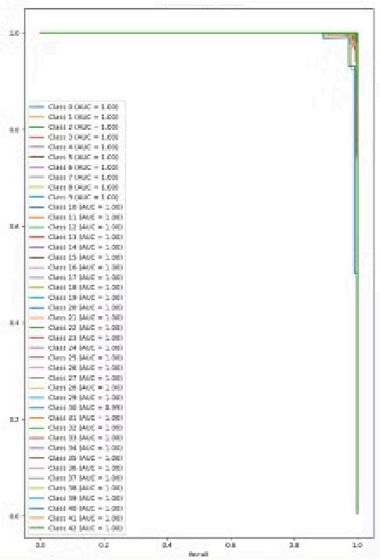
--- Performance Metrics for resnet_inspired_model ---

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| accuracy | 0.995000 | 0.995000 | 0.995000 | 0.995000 |
| macro avg | 0.996000 | 0.994000 | 0.995000 | 9475.000000 |
| weighted avg | 0.995000 | 0.995000 | 0.995000 | 9475.000000 |
| AUC-ROC | 0.999991 | 0.999991 | 0.999991 | 0.999991 |
| Average Precision | 0.999699 | 0.999699 | 0.999699 | 0.999699 |

Not retrained



Retrained



**CNN régularisé**

Not retrained

--- Performance Metrics for cnn_regularized_model ---

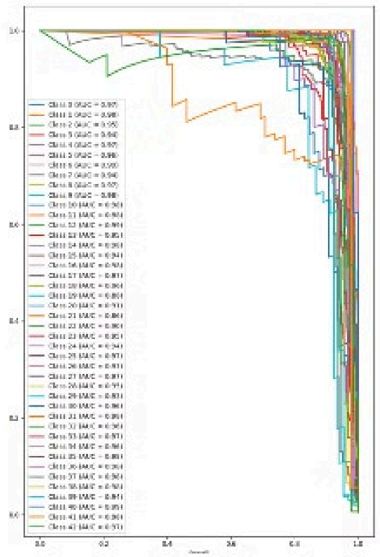
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| accuracy | 0.922000 | 0.922000 | 0.922000 | 0.922000 |
| macro avg | 0.912000 | 0.922000 | 0.913000 | 9475.000000 |
| weighted avg | 0.930000 | 0.922000 | 0.924000 | 9475.000000 |
| AUC-ROC | 0.995395 | 0.995395 | 0.995395 | 0.995395 |
| Average Precision | 0.958185 | 0.958185 | 0.958185 | 0.958185 |

Retrained

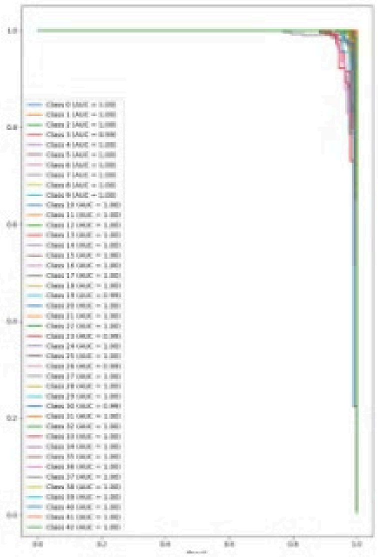
--- Performance Metrics for cnn_regularized_model ---

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| accuracy | 0.987000 | 0.987000 | 0.987000 | 0.987000 |
| macro avg | 0.988000 | 0.988000 | 0.988000 | 9475.000000 |
| weighted avg | 0.987000 | 0.987000 | 0.987000 | 9475.000000 |
| AUC-ROC | 0.999950 | 0.999950 | 0.999950 | 0.999950 |
| Average Precision | 0.998502 | 0.998502 | 0.998502 | 0.998502 |

Not retrained



Retrained



# Conclusion

### Model Selection

The 'ResNet Inspired' model excels in challenging conditions, while the 'CNN Regularized' is better suited for limited resources.

### Complexity and Performance

Performance is more dependent on the quality of the training than on the complexity of the model.

### Lessons Learned

Importance of good notebook management to optimize loading times and ensure efficient analysis.

### Workflow

It is crucial to validate the consistency of the workflow before training a deep learning model.

### Analysis Techniques

In-depth analysis using advanced techniques helps to optimize performance according to the problem to be solved.

### Robustness

Deep architectures, such as "ResNet Inspired", are more resistant to noise, which is crucial

for real-world applications.