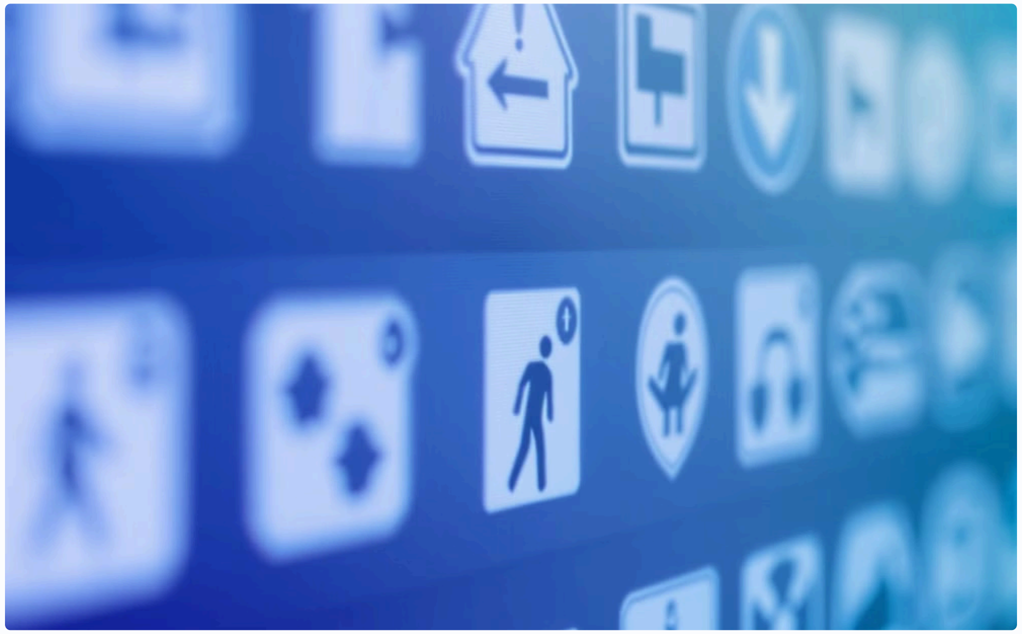
DEEP-LEARNING

Classification of Road Signs

Road sign classification is a crucial area in computer vision, involving the identification and categorization of signs to regulate road traffic. This project analyzes the methodology for developing a Deep Learning model optimized for this task.

Context



Motivation

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

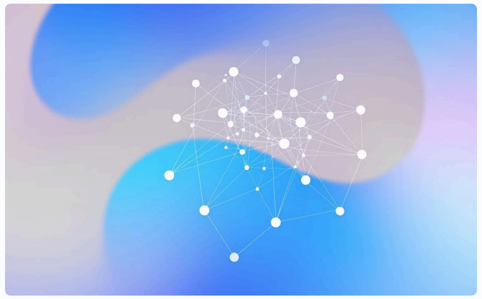
Objectives

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** under **varied conditions**.

Achievements

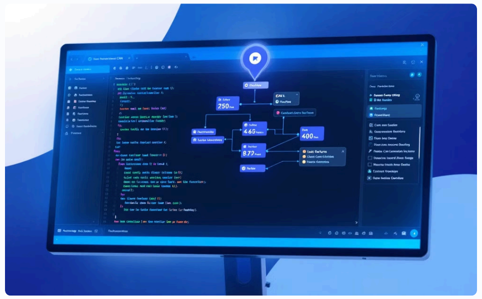
Data Augmentation

Using techniques like **rotation**, **flipping**, **zooming** and **cropping** to increase **sample diversity** and improve **generalization**.



Exploring Regularization

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



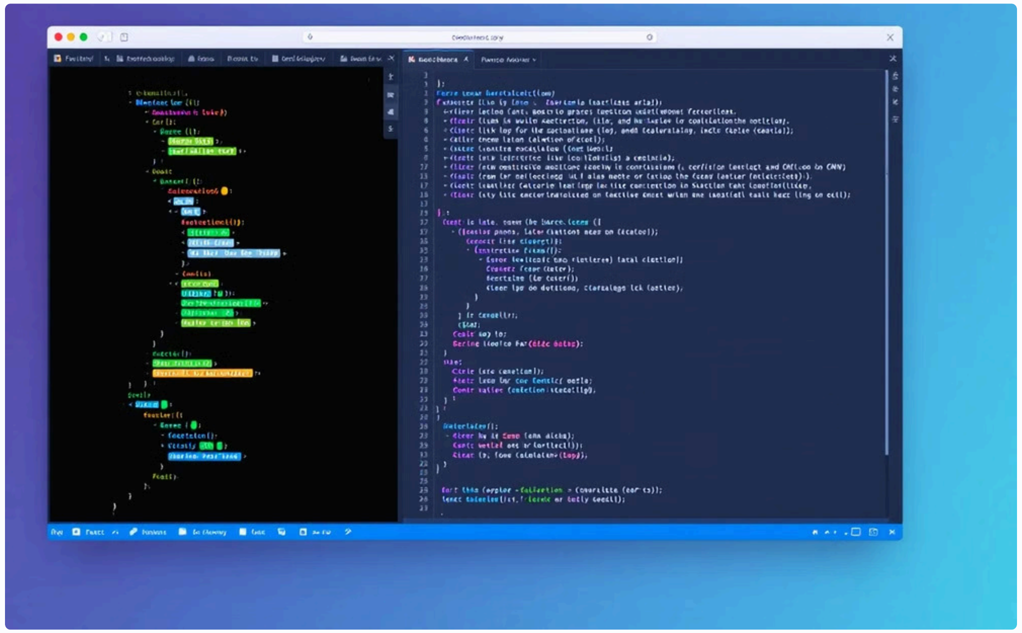
Developing Optimized Models for Classification

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

Experimentation Under Adverse Conditions

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

Dataset and Data Preprocessing



Introducing the GTSRB Dataset

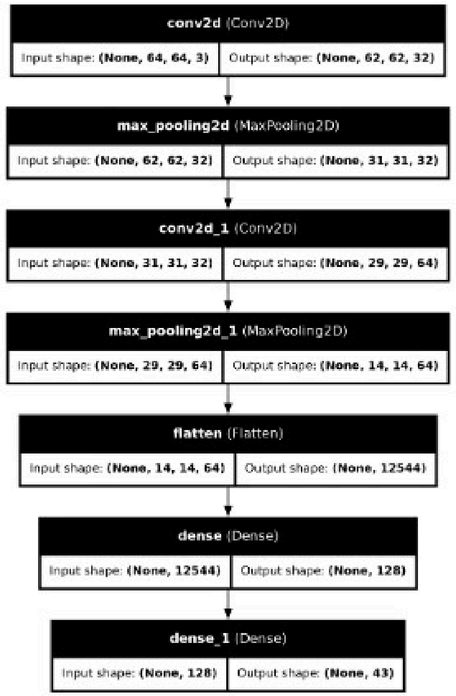
The **GTSRB** dataset is composed of thousands of images of **road signs**. It is used for training and evaluating computer vision models. It contains **43 distinct classes** of traffic signs, each image with metadata such as the **shape**, **color**, and **ID** of the pictogram.

Data Preprocessing

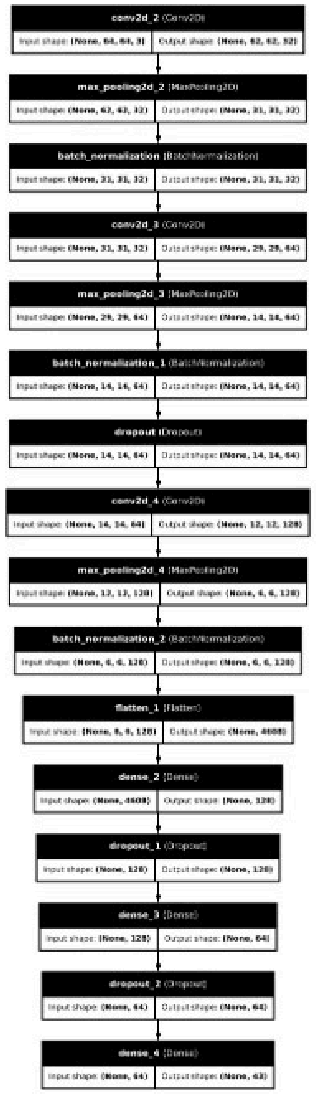
Data preprocessing includes steps such as **resizing** and **normalizing** images, **splitting data** into training, validation, and test sets, **augmenting** data to improve model robustness, and **encoding** labels in one-hot encoding.

Architecture

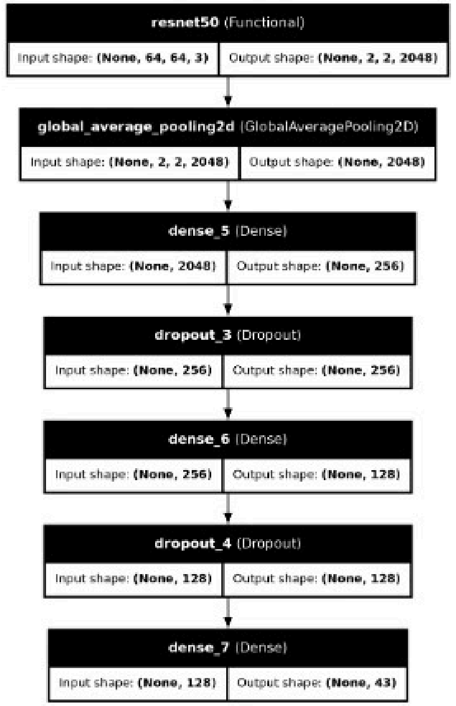
Simple CNN



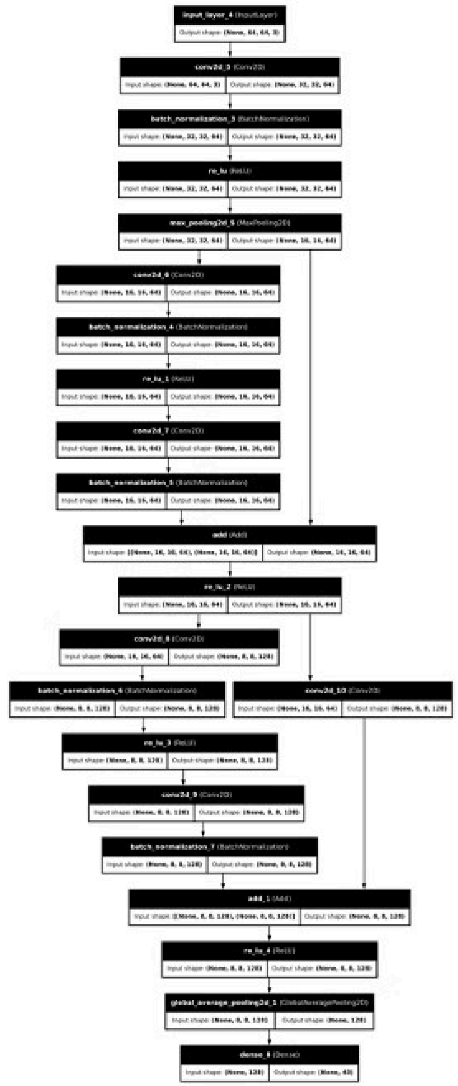
CNN régularisé



ResNet50 fine-tuned

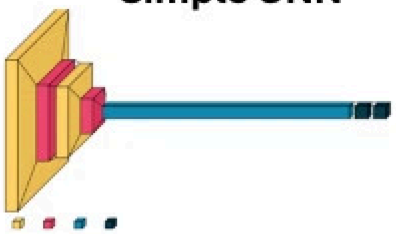


ResNet inspiré

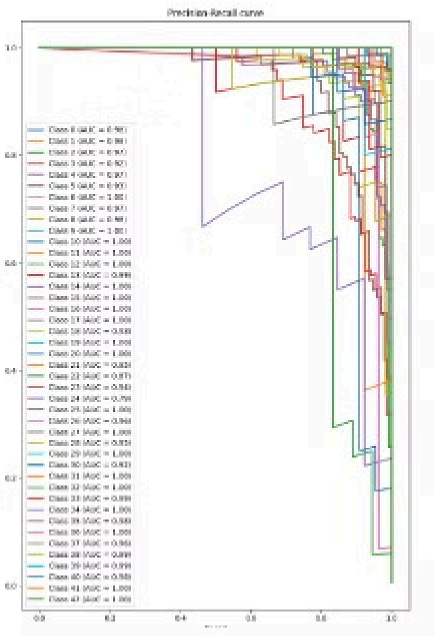


Methodology and Models

Simple CNN

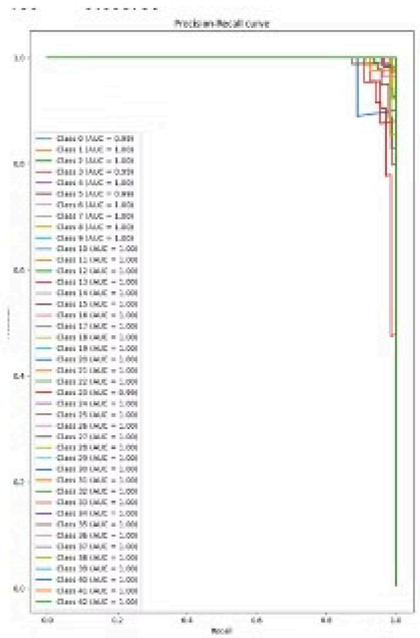


Précision : 93.4%
Rappel : 94.0%
F1-score : 93.9%
AUC-ROC : 0.9988
Précision Moyenne (AP) : 0.9750



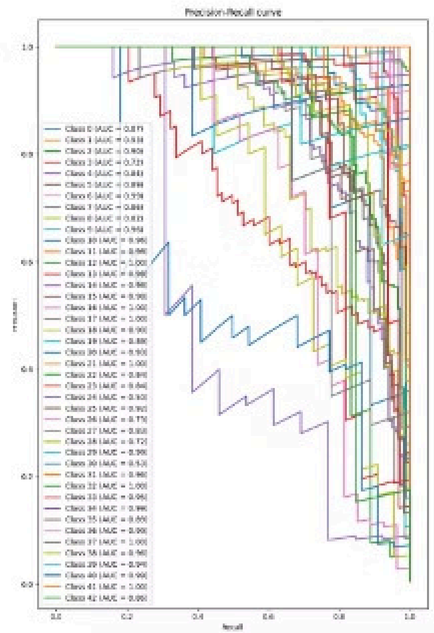
CNN régularisé

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

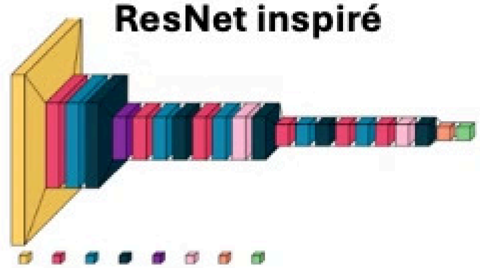


ResNet50 fine-tuned

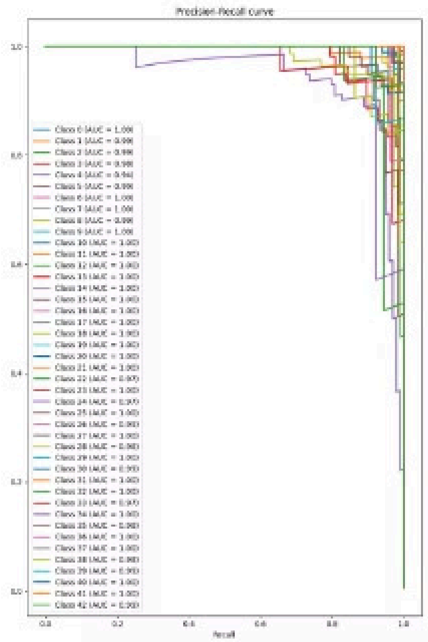
Précision : 84.2%
Rappel : 84.2%
F1-score : 84.2%
AUC-ROC : 0.9941
Précision Moyenne (AP) : 0.9036



ResNet inspiré



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

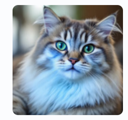


Training and Experimentation

Results Analysis

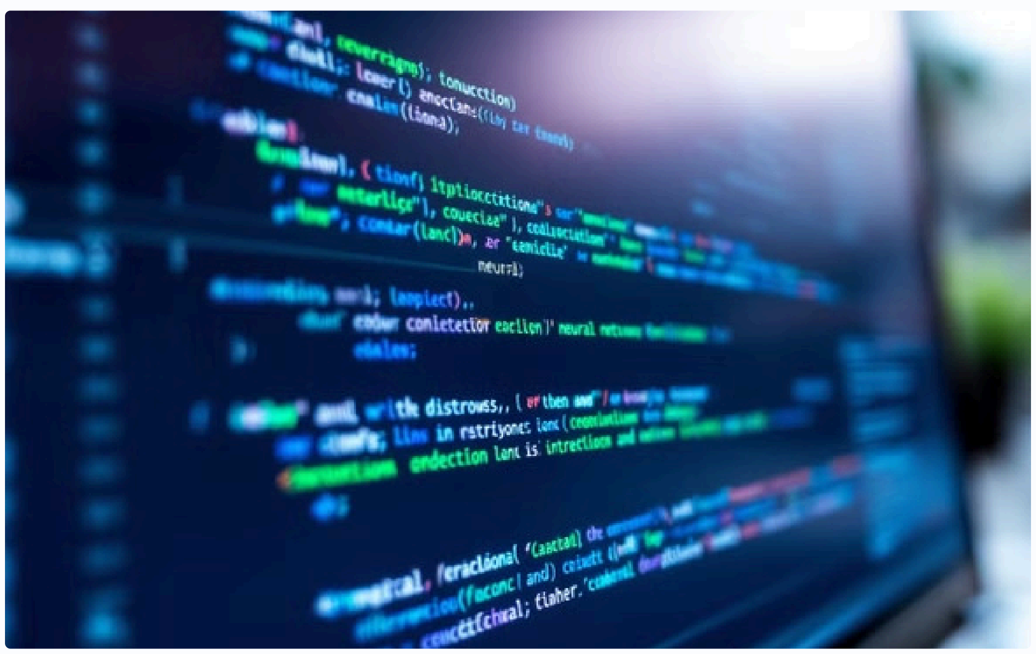
Layer Activations

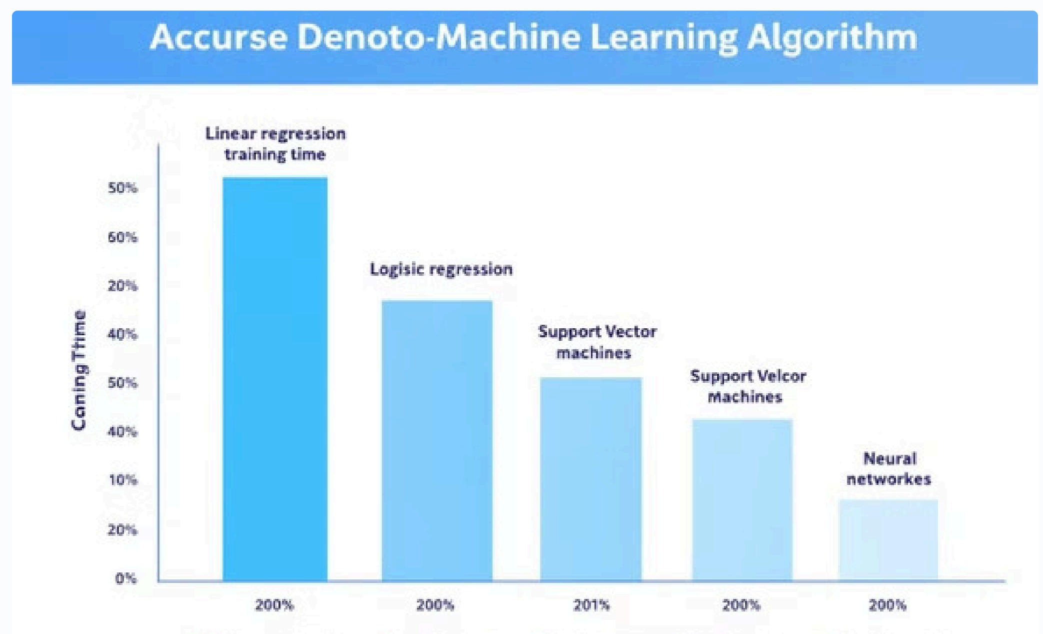
The **activations** of a CNN's layers reveal the **specific features learned** at each stage. Visualizing these activations helps to **understand the visual patterns detected** and identify issues such as **overfitting**.

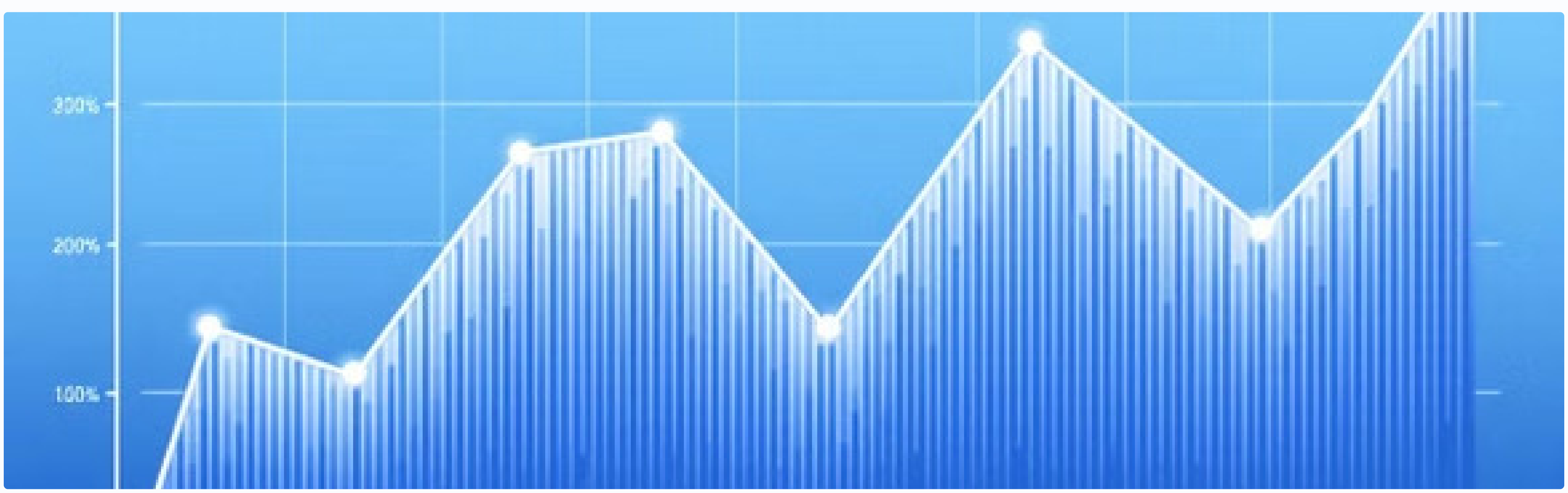


Grad-CAM Analysis

The **Grad-CAM** technique allows for visual identification of the **areas most influencing the model's prediction**, thus evaluating the relevance of its choices.

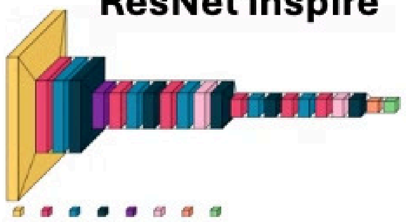






Adverse Condition Testing

ResNet inspiré



Not retrained

--- Performance Metrics for resnet_inspired_model ---

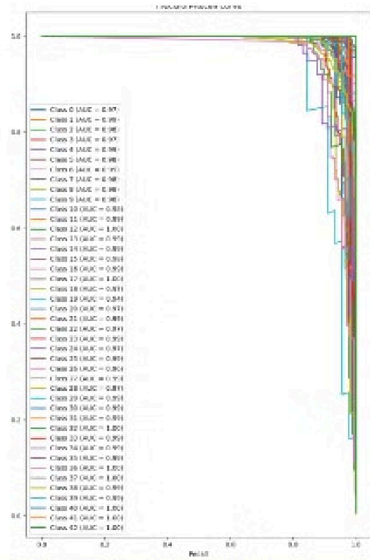
	precision	recall	f1-score	support
accuracy	0.944000	0.944000	0.944000	0.944000
macro avg	0.949000	0.945000	0.945000	9475.000000
weighted avg	0.948000	0.944000	0.944000	9475.000000
AUC-ROC	0.998999	0.998999	0.998999	0.998999
Average Precision	0.984939	0.984939	0.984939	0.984939

Retrained

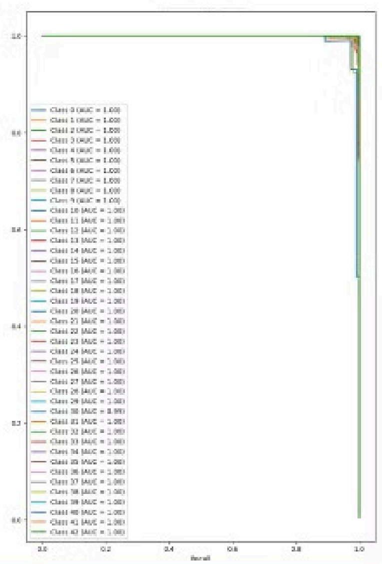
--- Performance Metrics for resnet_inspired_model ---

	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 ---

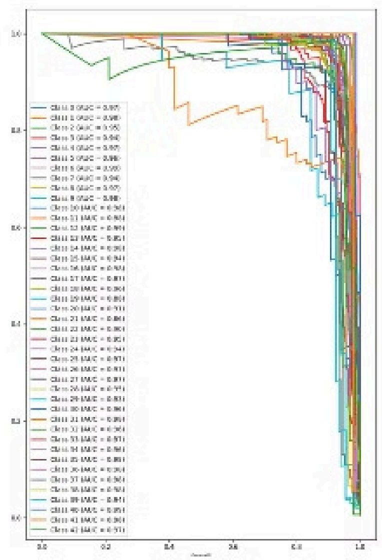
	precision	recall	f1-score	support
accuracy	0.922000	0.922000	0.922000	0.922000
macro avg	0.912000	0.922000	0.913000	9475.000000
weighted avg	0.930000	0.922000	0.924000	9475.000000
AUC-ROC	0.995395	0.995395	0.995395	0.995395
Average Precision	0.958185	0.958185	0.958185	0.958185

Retrained

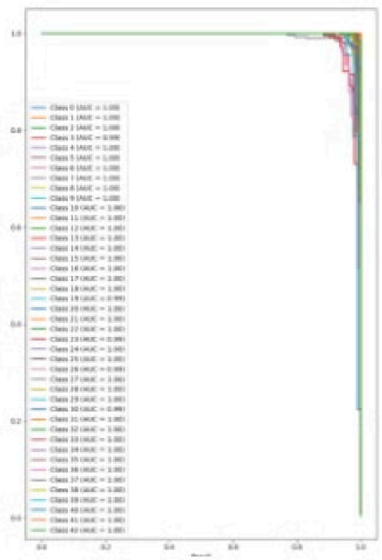
--- Performance Metrics for cnn_regularized_model ---

	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 Choice

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

Complexity and Performance

Performance depends more on the quality of training than the complexity of the model.

Lessons Learned

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

Workflow

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

Analysis Techniques

In-depth analysis through advanced techniques optimizes performance depending on the problem to be solved.

Robustness

Deep architectures, such as "ResNet Inspired", are more resistant to noise, crucial for real-world applications.