



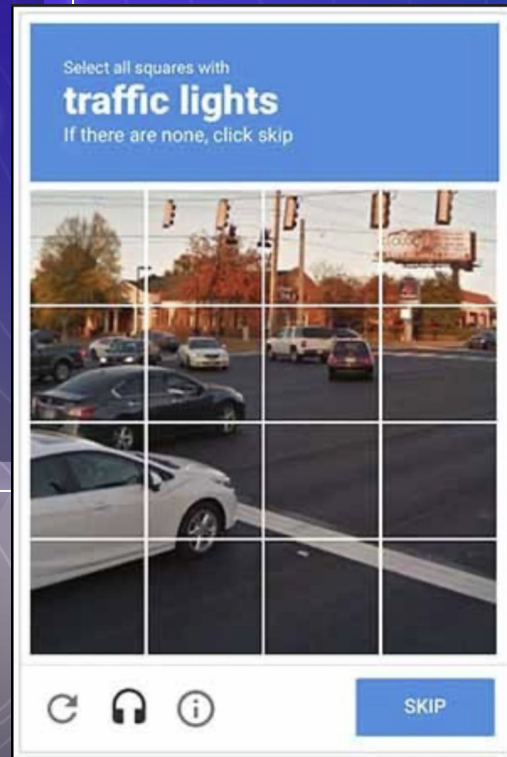
# Projet IA

Reconnaissance des  
lettres de la langue  
des signes américaine

Benjamin Cordebar, Victor Ménestrel, Mathis Aulagnier

# Introduction

Le sujet traite de la reconnaissance d'image par IA, qui a fait d'énorme progrès au cours de ces dernières années, l'objectif de notre sujet est de concevoir d'entraîner un réseau de neurones pour prédire la lettre signée en fonction d'une imagerie.



# Sommaire

Utilisation d'un réseau de neurones pour reconnaître des images

**01**

**Présentation des  
données & outils**

**02**

**Réseau de neurones  
à convolution (CNN)**

Pour aller plus loin

**03**

**Augmentation des  
données**

**04**

**Comparaison des  
performances avec  
un MLP**

**05**

**Comparaison des  
performances avec  
un RNN**



01

# Présentation des données et outils



# 1 Présentation des données



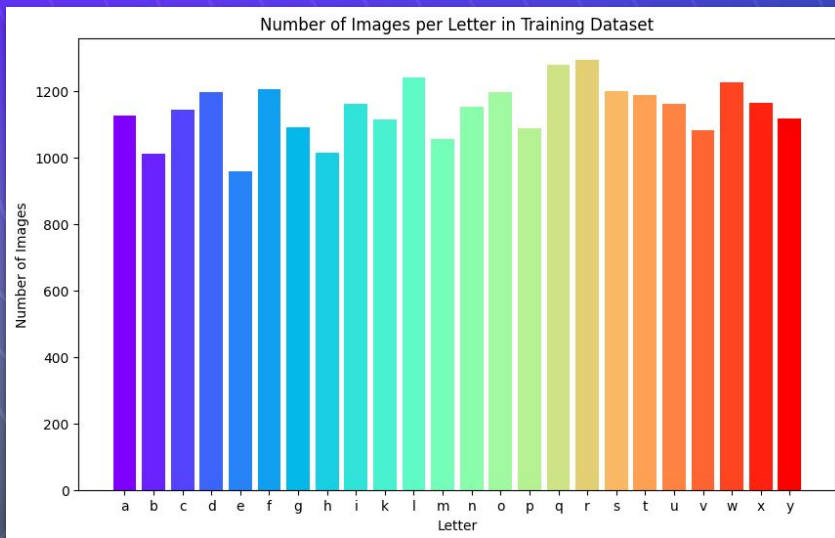
kaggle

Disponible sur Kaggle

Images (28x28)

Taille des données d'entraînement : 27456

Taille des données de test : 7173



1

# Langage et bibliothèque



Python



Pytorch



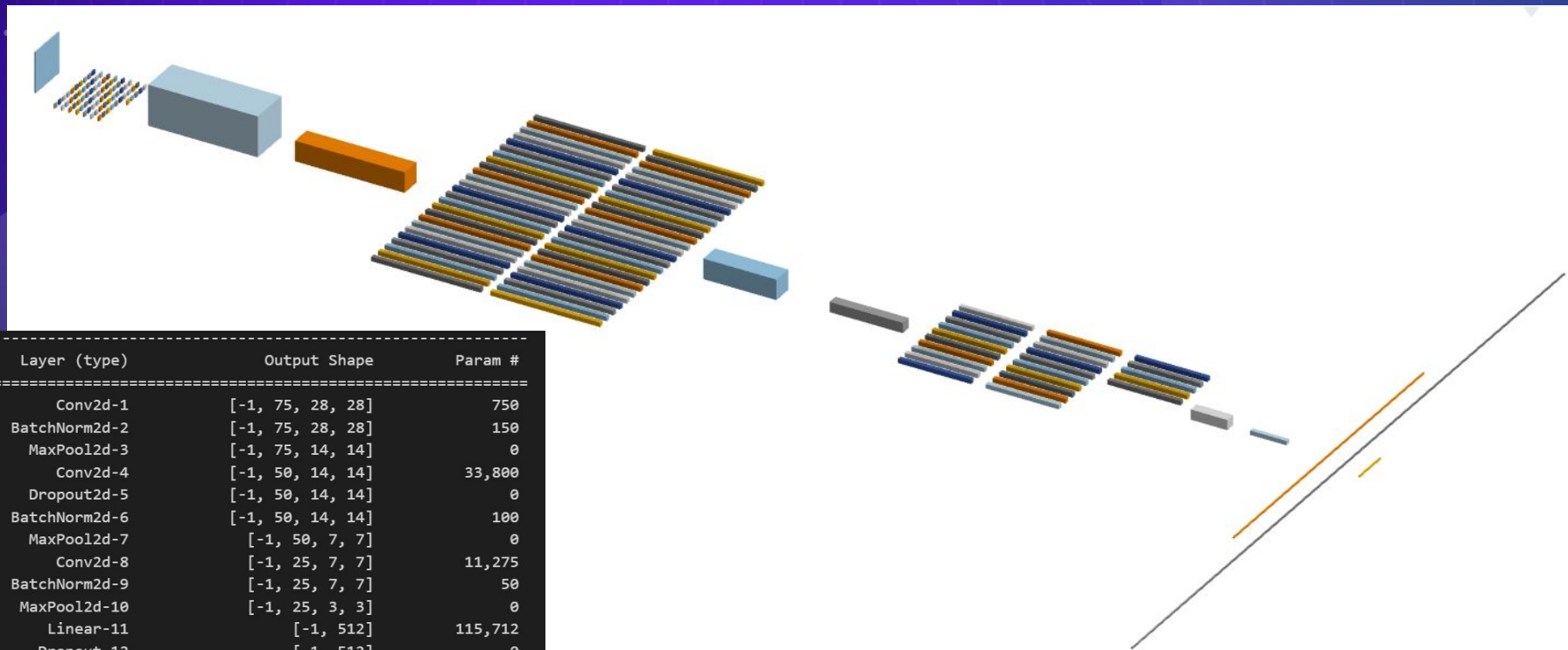


02

# Réseau de neurones à convolution (CNN)



# 3 Réseau de neurones à convolution (CNN)

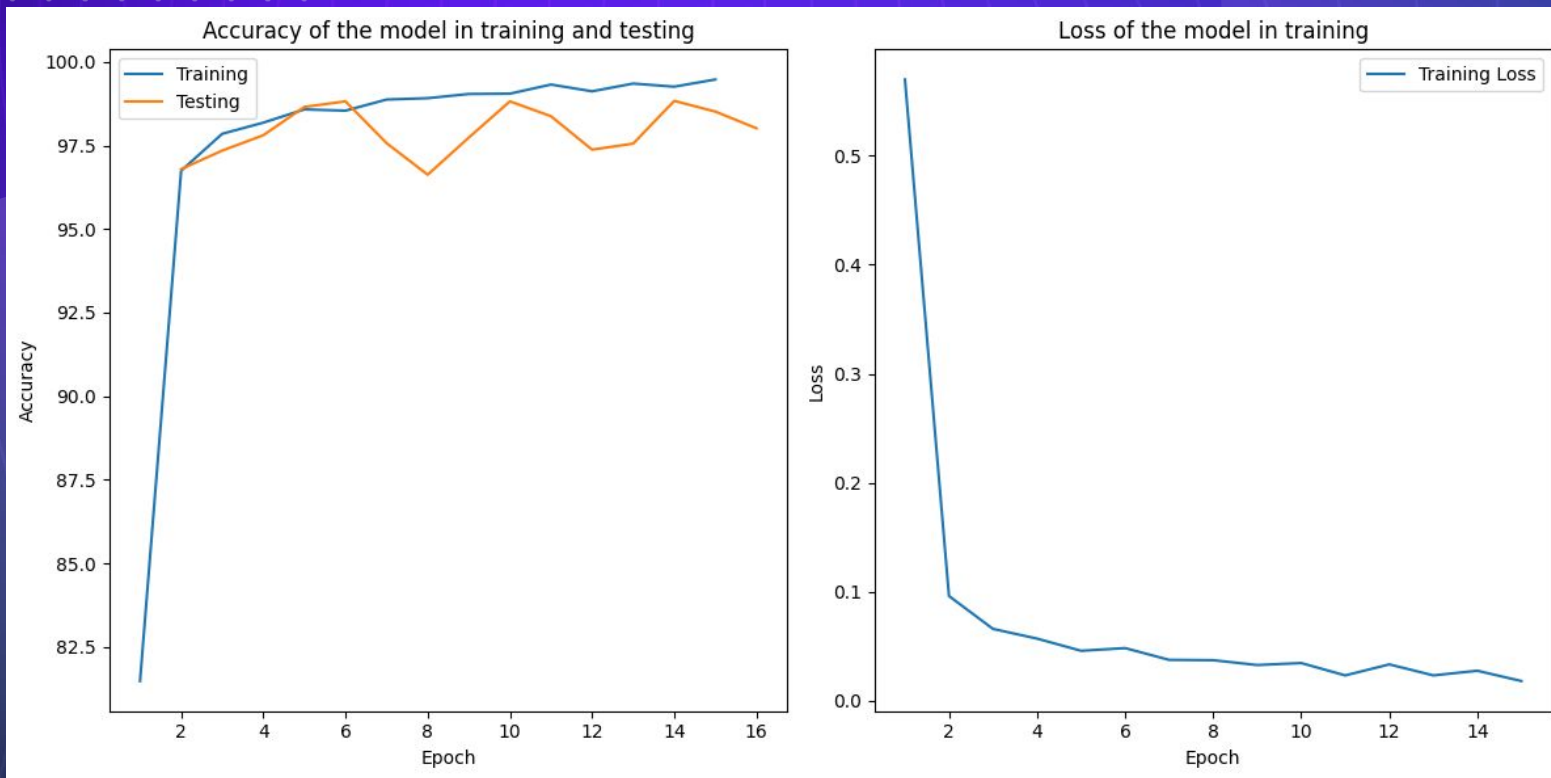


Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 75, 28, 28]	750
BatchNorm2d-2	[-1, 75, 28, 28]	150
MaxPool2d-3	[-1, 75, 14, 14]	0
Conv2d-4	[-1, 50, 14, 14]	33,800
Dropout2d-5	[-1, 50, 14, 14]	0
BatchNorm2d-6	[-1, 50, 14, 14]	100
MaxPool2d-7	[-1, 50, 7, 7]	0
Conv2d-8	[-1, 25, 7, 7]	11,275
BatchNorm2d-9	[-1, 25, 7, 7]	50
MaxPool2d-10	[-1, 25, 3, 3]	0
Linear-11	[-1, 512]	115,712
Dropout-12	[-1, 512]	0
Linear-13	[-1, 25]	12,825

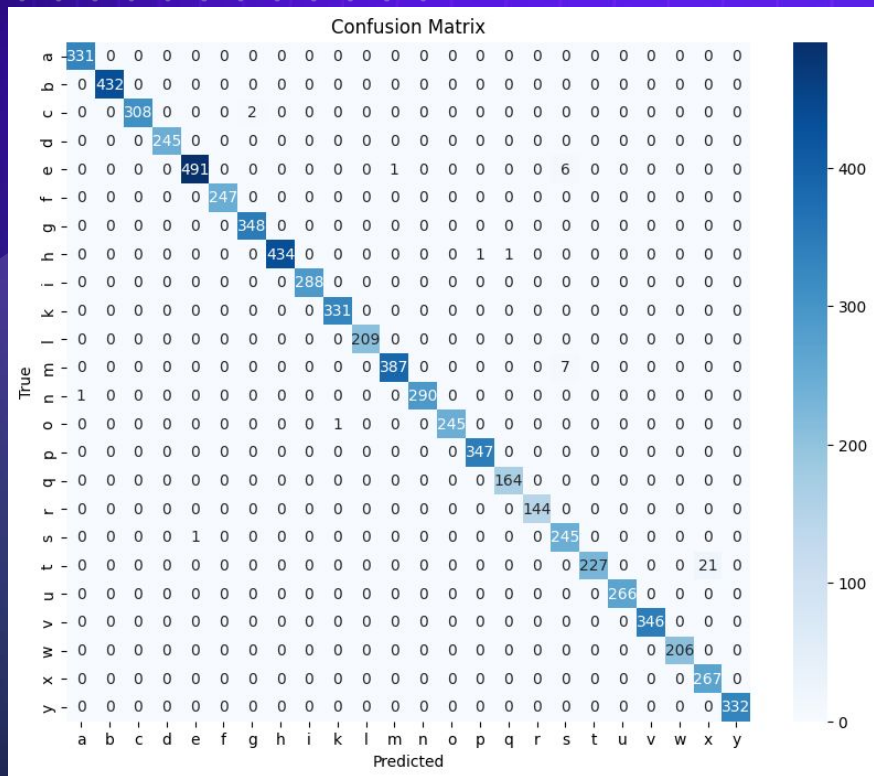
=====  
Total params: 174,662  
Trainable params: 174,662



# 3 Réseau de neurones à convolution (CNN)



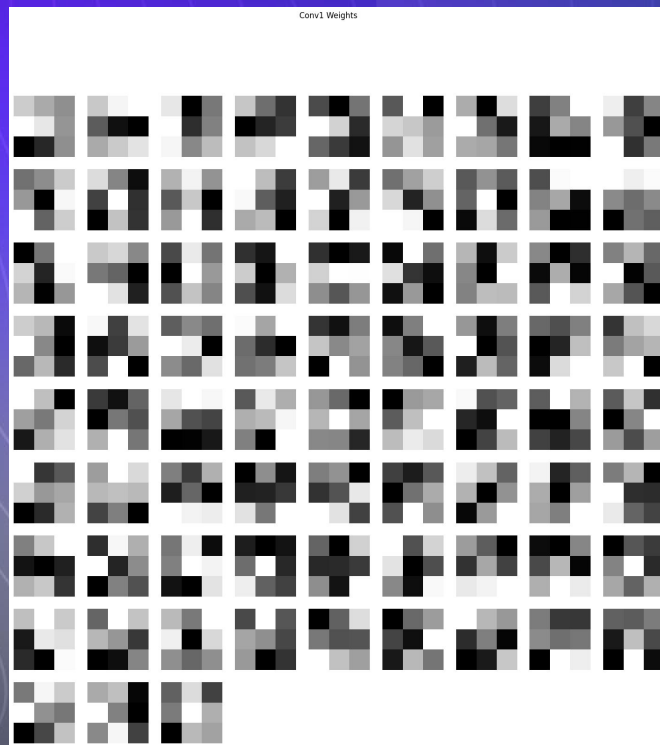
# 3 Réseau de neurones à convolution (CNN)



- Data augmentation (+100%)
- Taille du Batch : 30
- Loss : CrossEntropy
- Optimizer : Adam

Précision (test) = 99.4 %

# 3 Réseau de neurones à convolution (CNN)



# 3 Réseau de neurones à convolution (CNN)

Limites :

- Difficultés pour faire fonctionner avec nos propres images.



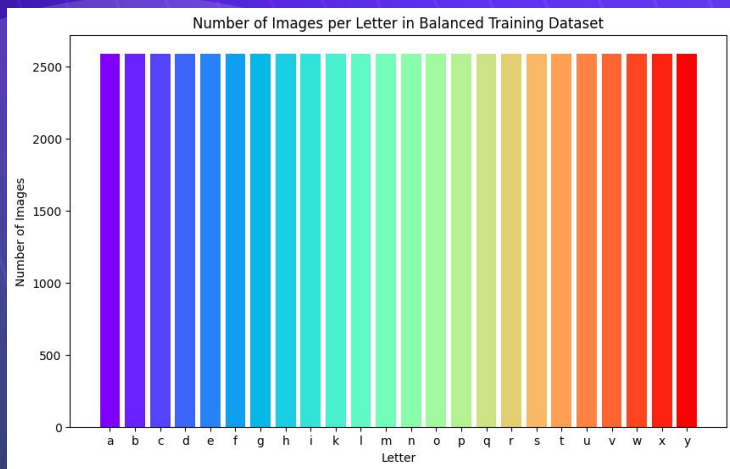


03

# Augmentation des données

# 3 Réseau de neurones à convolution (CNN)

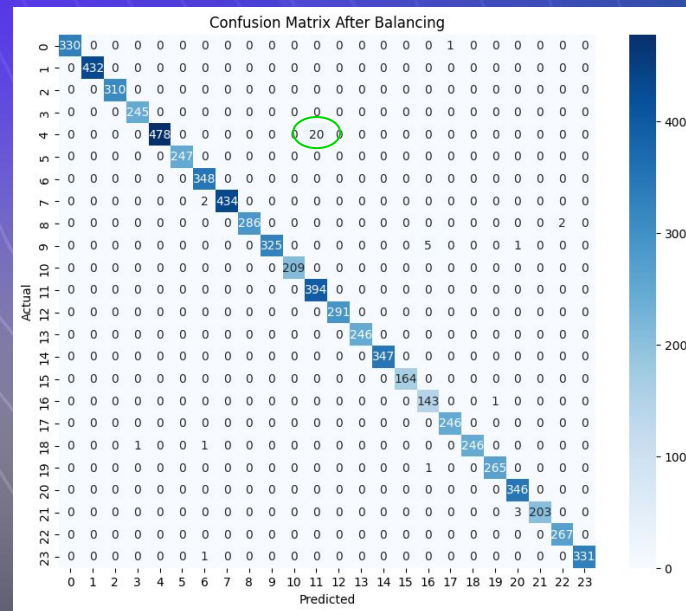
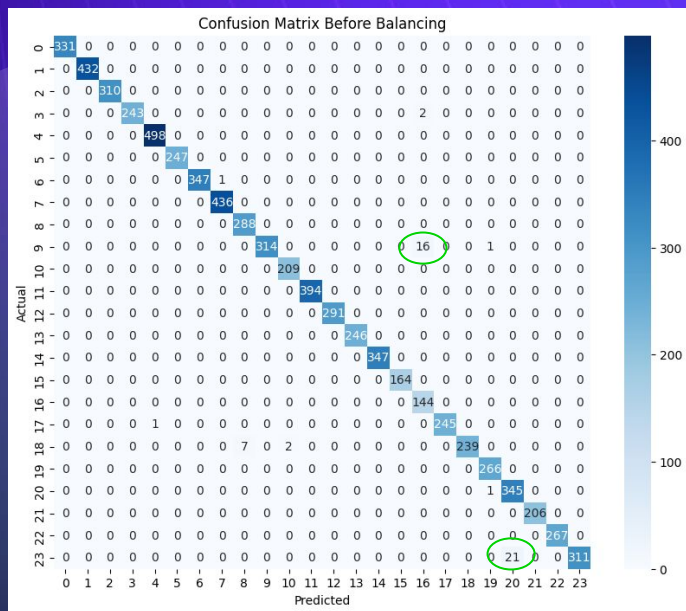
Nouvelle répartition des données



Accuracy : 99,4 -> 99,6

# 3 Réseau de neurones à convolution (CNN)

## Comparaison des matrices de confusion





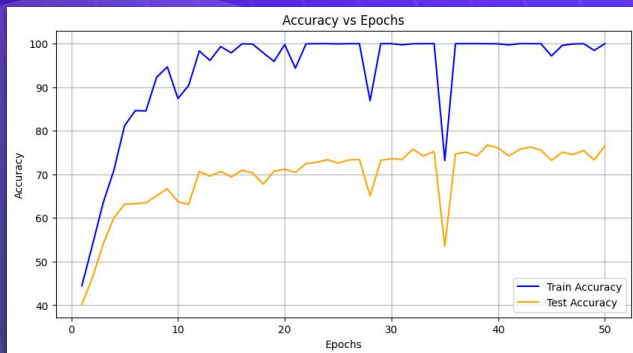
**04**

# **Comparaison des performances avec un MLP**

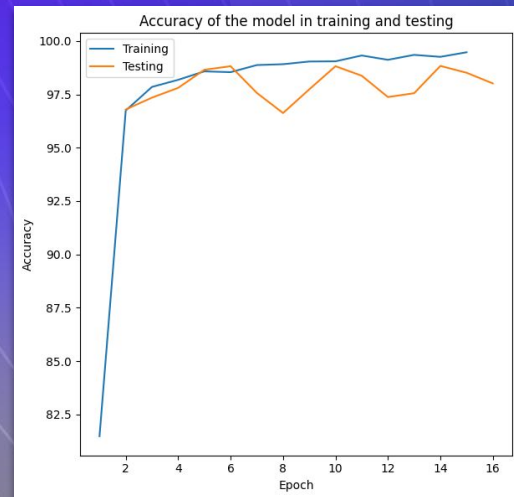


# MLP - CNN

- 3 couches cachées : 256, 128, 64, 24 neurones
- Fonction d'activation : ReLU



— Accuracy : 76.6%



— Accuracy : 99.6%

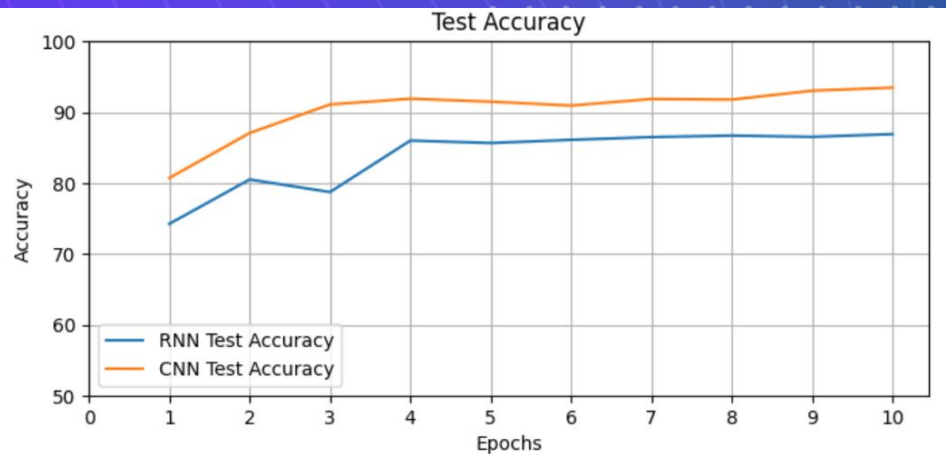
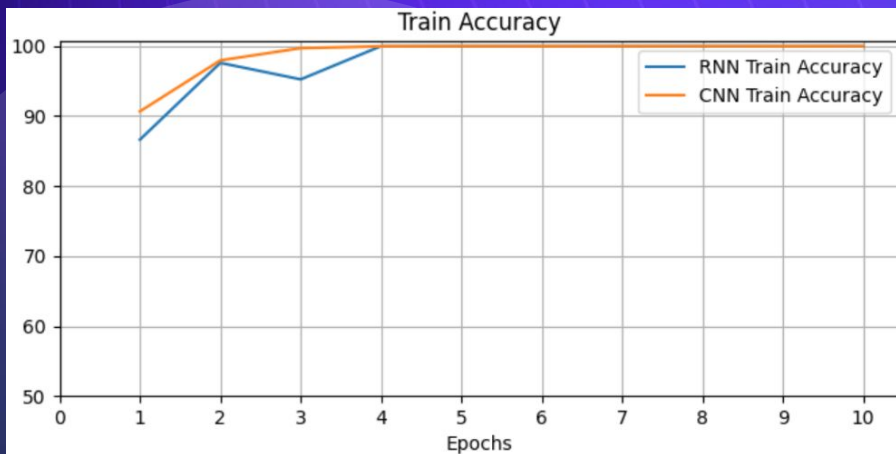


04

## Comparaison des performances avec un RNN

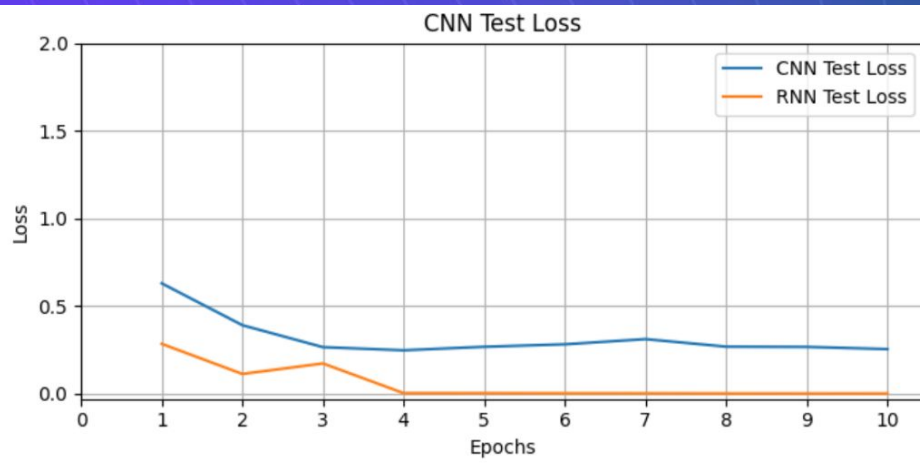
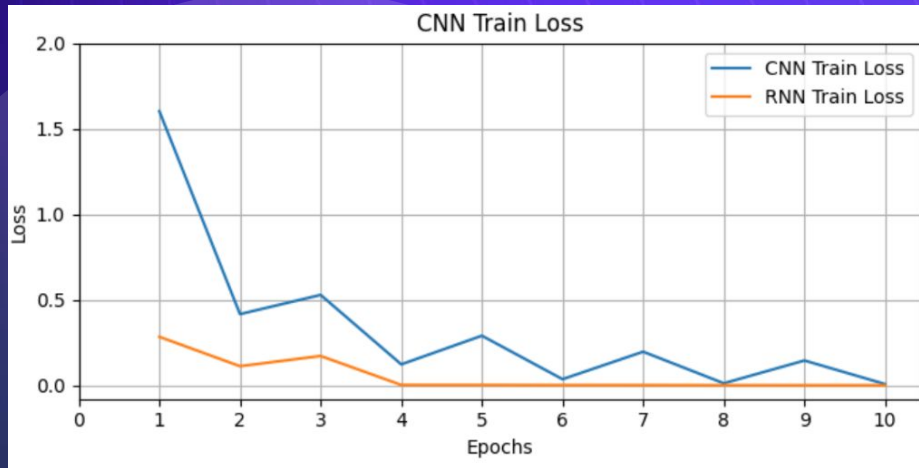
# RNN - CNN

## Accuracy



# RNN - CNN

Loss





# Conclusion

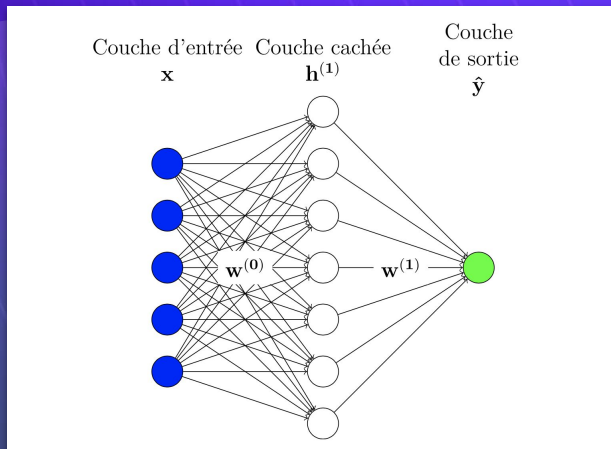


# Merci pour votre attention



## 2

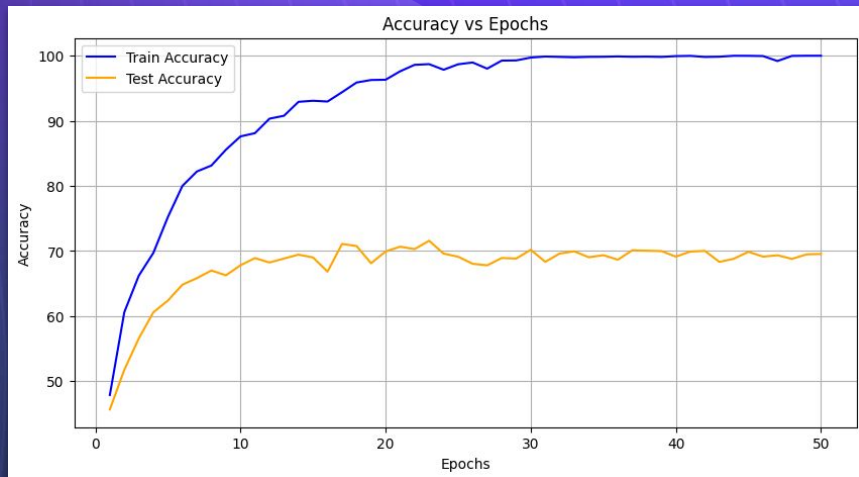
# Perceptron multicouches (MLP)



- 3 couches min
- Modélisation de relation complexes
- Algorithme de rétropropagation

# MLP - Résultats

- 1 couche cachée : 128 neurones
- Fonction d'activation : ReLU



- Accuracy : 69.5%
- AUC-ROC : 0.971



# MLP - Limites

Malgré :

- gestion du surapprentissage
- ajustement dynamique du taux d'apprentissage
- augmentation couches cachées/neurones

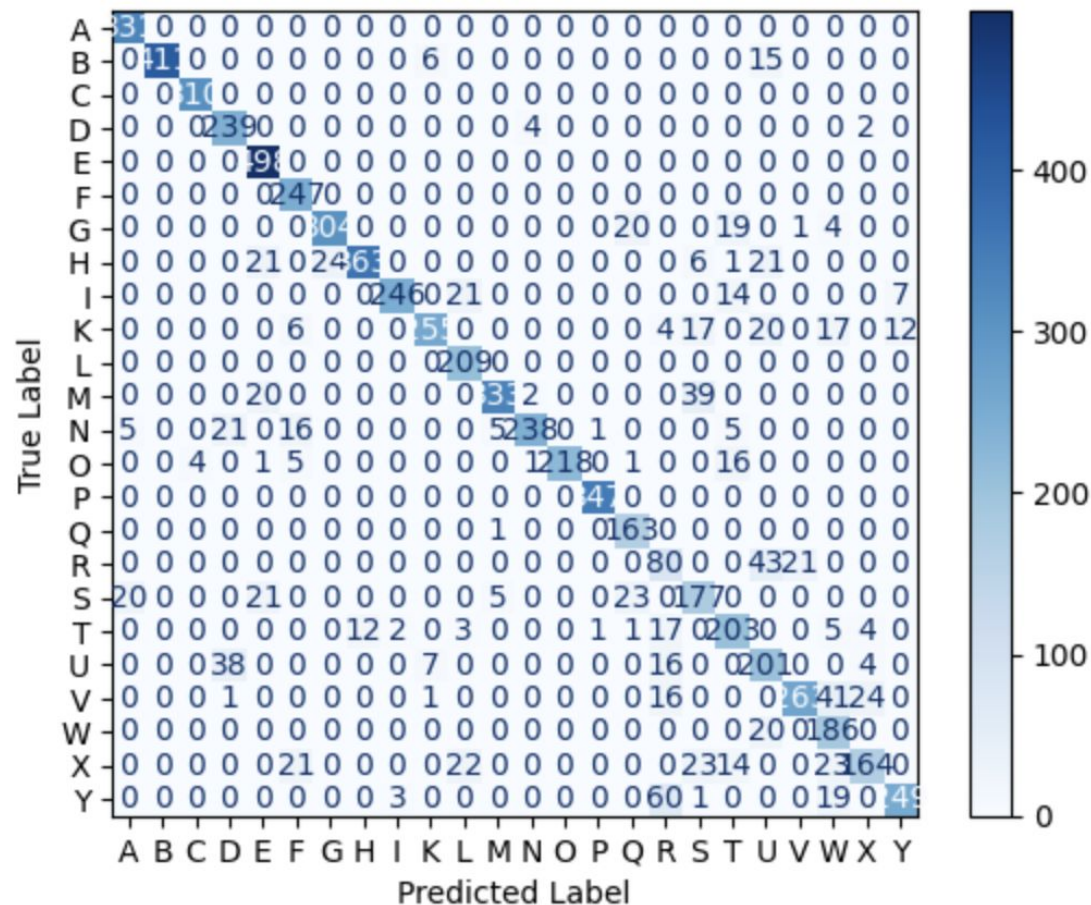


Résultats **non satisfaisants**



**Limite** du modèle perceptron multicouches (MLP)

# Confusion Matrix



Train and Test Accuracy over Epochs

