SORBONNE UNIVERSITÉ

# Active Object Localization par Deep Reinforcement Learning

Un projet basé sur les travaux de l'article *"Active Object Localization with Deep Reinforcement Learning", Juan C. Caicedo, Svetlana Lazebnik (2015) 10.1109/ICCV.2015.286.*

Projet encadré par : Pr. BLOCH Isabelle

# PLAN DE LA PRÉSENTATION

01. **Contexte**

02. **Etat de l'art**

Description d'une approche de l'état de l'art.

03. **Concepts d'apprentissage par Renforcement**

Modèle markovien, deep-q-learning.

04. **Approche de l'article**

Présentation de la modélisation décrite dans le papier.

05. **Résultats obtenus**

Résultats selon les classes, les hyper-paramètres et pendant l'évolution de l'apprentissage.

06. **Conclusion**

# 01.

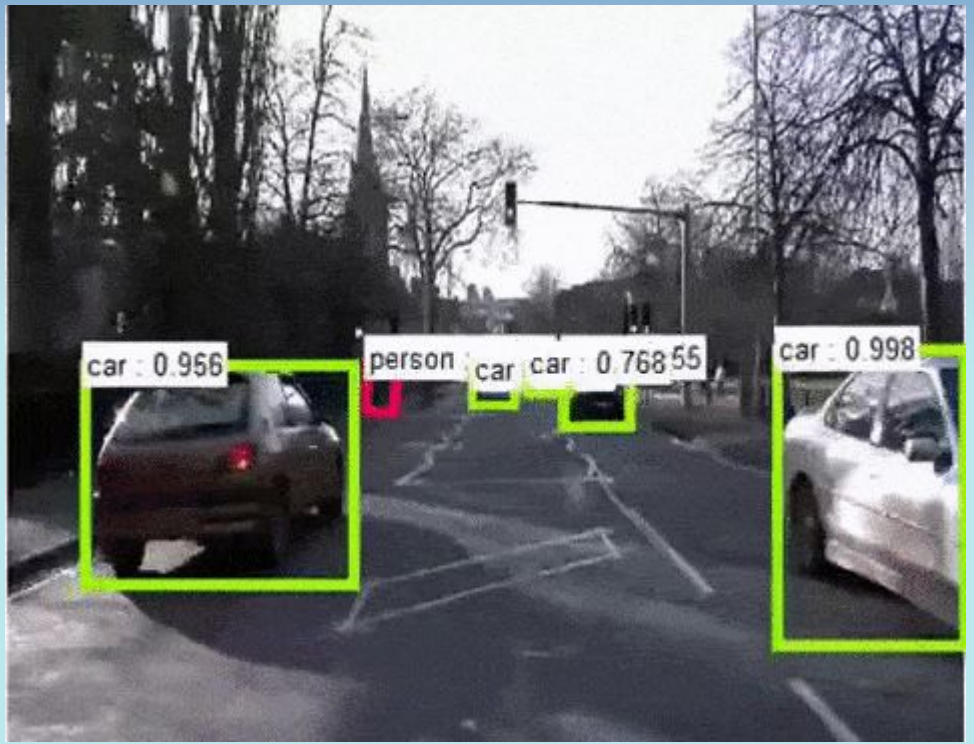
# Contexte

# Contexte

Objectif : Localiser les boîtes englobantes d'objets dans une image

Application dans une variété de domaines :

- Voitures autonomes
- Reconnaissance d'aliments.
- Interprétation de scènes.
- Analyse de comportements sociaux ( mouvement d'acheteurs dans des magasins par exemple ).

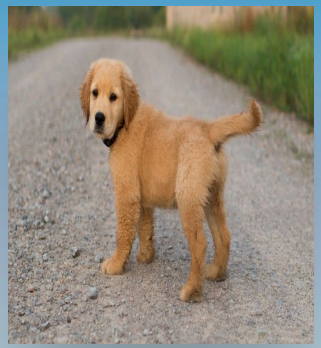


# 02.

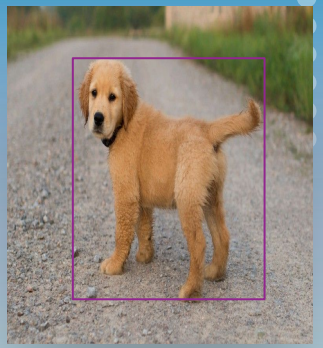
# Etat de l’art

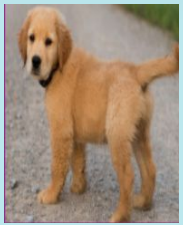
# Exemple d'une méthode existante : R-CNN

Redimensionner

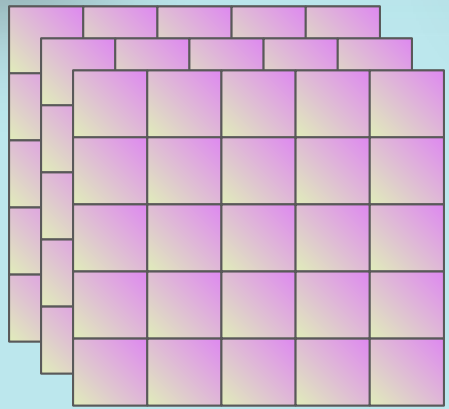


Détection de régions





Extraction de caractéristiques



Feature Vector

Classification

SVM classe 'Person'

Person ? Non

SVM classe 'Dog'

Dog ? Oui

# 03.

# Apprentissage par Renforcement

# Fondements de l'apprentissage par renforcement

## Environnement

Ensemble fini d'états $S$

## Actions

Ensemble fini d'actions $A$

## Probabilités de transitions et "policy"

$$P^a_{s's} = P(s_{t+1} = s' | s_t = s, A_t = a)$$

$$\pi(a|s) = P[A_t = a | S_t = s]$$

## Récompense

$$R : S \times A \to R$$

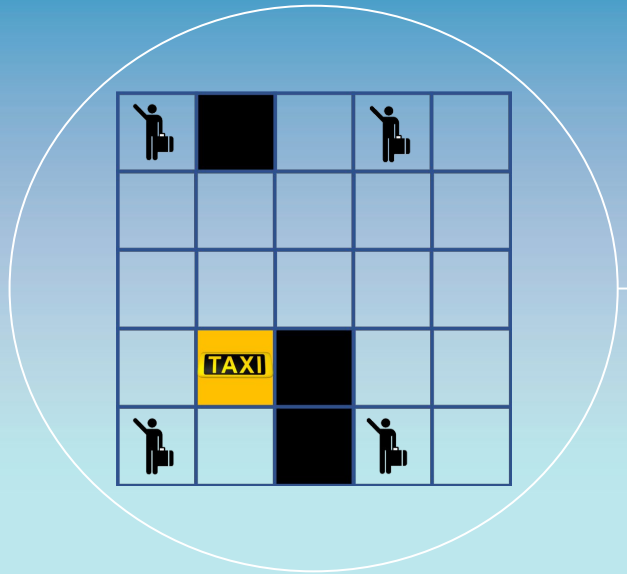
## Espérance de retour

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

## Fonction de valeur

$$Q_\pi(s,a) = E_\pi[G_t | S_t = s, A_t = a]$$

# Q-Learning




Q-Table ( matrice d'estimation de la fonction Q optimale )

| | Haut | Bas | Gauche | Droite |
|---|---|---|---|---|
| Etat 1 | 8.5819 | -3.2148 | -2.3887 | -1.3541 |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| Etat N | -4.7155 | 8.4522 | -6.3945 | -1.1574 |

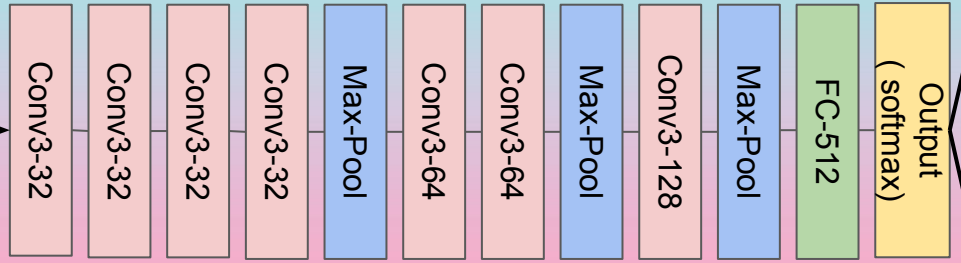
**Fonction de Bellman**

$$Q^{nouveau}(s_t, a_t) = \underbrace{Q(s_t, a_t)}_{\text{Ancienne valeur}} + \underbrace{\alpha}_{\text{Taux d'apprentissage}} \left( \underbrace{r_t}_{\text{Récompense accordée à l'instant } t} + \underbrace{\gamma}_{\text{Facteur de réduction}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{Estimation meilleure Q pour l'état successeur } s_{t+1}} - \underbrace{Q(s_t, a_t)}_{\text{Ancienne valeur}} \right)$$

# Deep-Q-Learning


Représentation d'état de l'environnement

**Estimateur de la fonction Q optimal**




Réseau de neurones appelé **"Q-Network"** avec des couches convolutionnels pour l'extraction de caractéristiques et des couches fully connected avec softmax en sortie

**P = 0.489**

**P = 0.511**

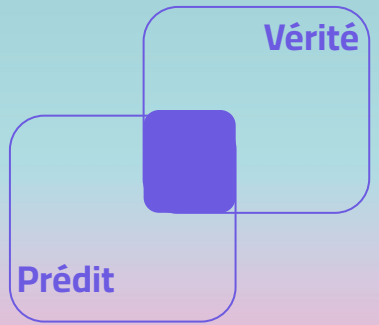
Output du réseau après softmax : probabilité que chaque action soit la meilleure possible.

# 04.

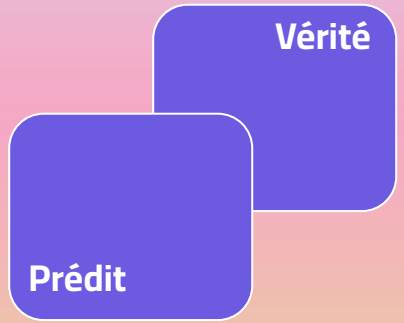
# Approche de l’article

# Métrique d'évaluation

Intersection




Union




+1, *Si IoU > seuil*
-1 *sinon*

$$AP = \sum_{n=1}^{Tailledataset} 1_{IoU(n)>seuil}$$

## Environnement : Bounding Box

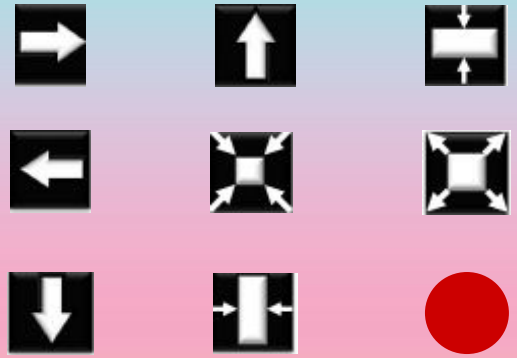
$(x_{min}, y_{min})$

$(x_{max}, y_{max})$

## Actions : translations de la bounding box

$$\alpha_w = \alpha . (x_{max} - x_{min})$$

$$\alpha_h = \alpha . (y_{max} - y_{min})$$



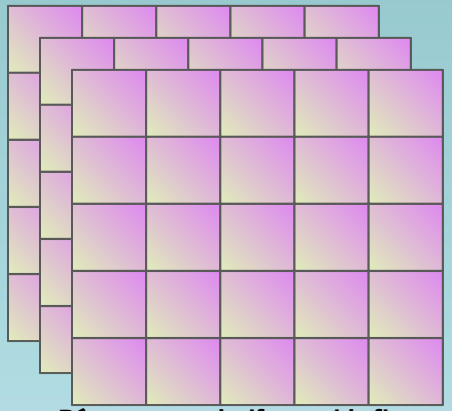
## Dataset utilisé : Pascal VOC 2007 + 2012

## Récompenses

$$R_t = \begin{cases} +\eta, & \text{si } IoU(b,g) \geq \tau \\ -\eta, & \text{sinon} \end{cases}$$

Redimensionner

Extraction de caractéristiques



Feature Vector

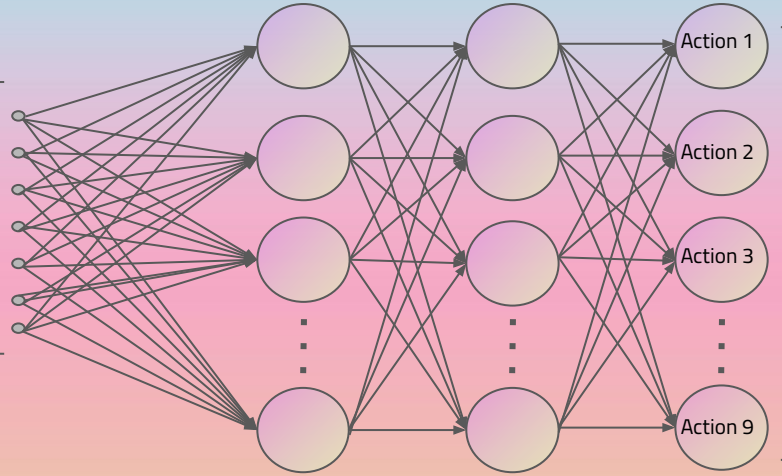
Réseau convolutif au poids fixes

Historique de 9 actions en one-hot

001000000
000010000
100000000
.
.
.
000100000
000100000
000000100

Feature Vector

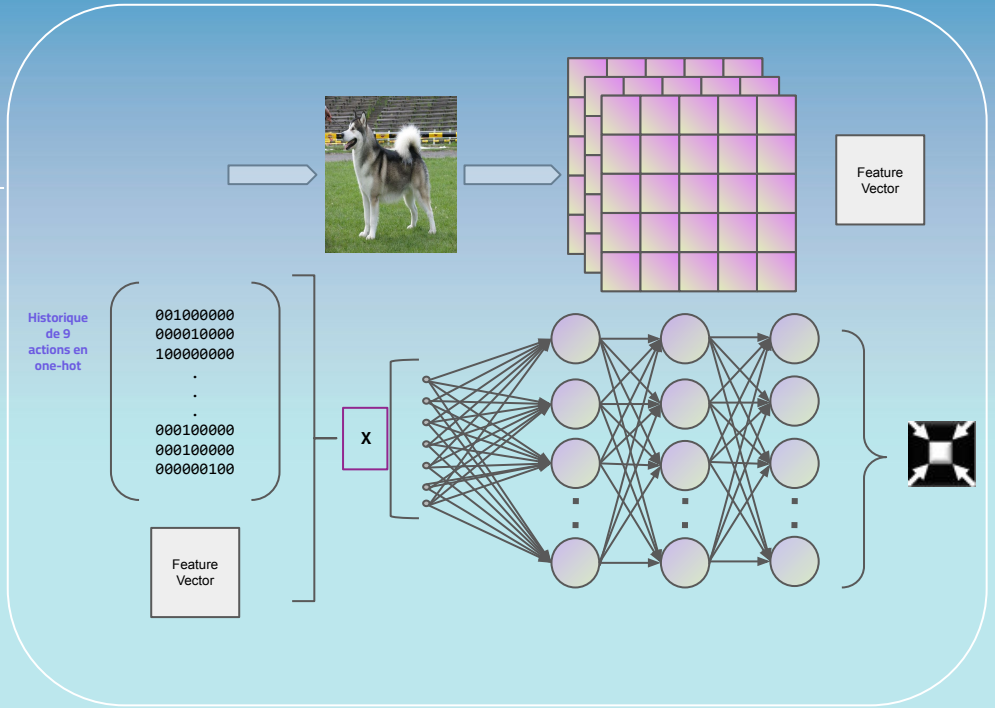
X




Max ( sortie ) = action choisie

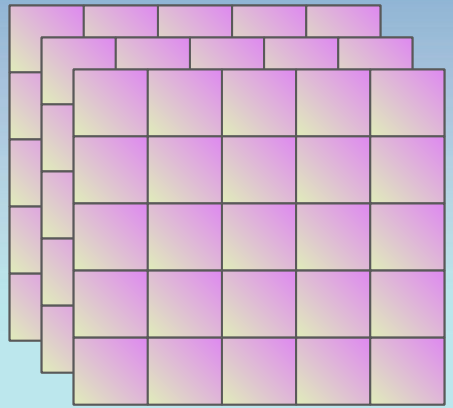
Les récompenses collectées est utilisé pour backpropager sur ce réseau

| X (81+25088) | Couche 1 (1024) | Couche 2 (1024) | Couche de sortie (9) |
|---|---|---|---|



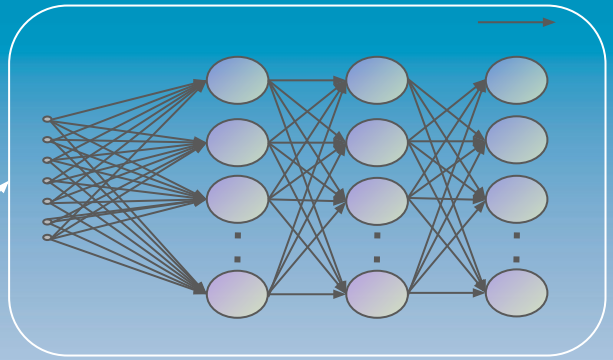

**Arrêt lorsque l'action choisie est : 🔴**
**Ou après 40 itérations.**

# Un réseau par classe

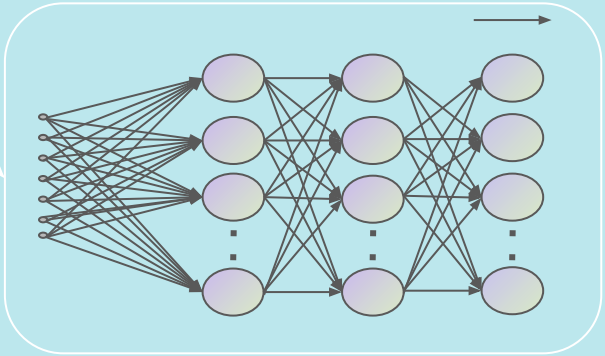


Feature Vector

Dog's Q-Network



Person's Q-Network



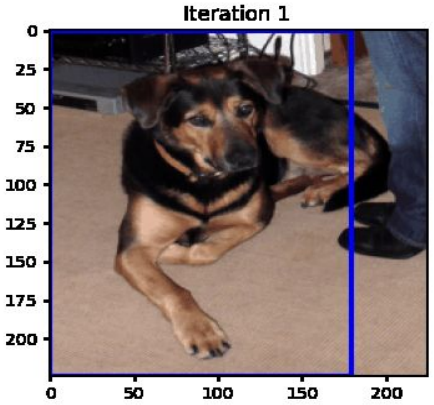
# Caractéristiques de l’apprentissage

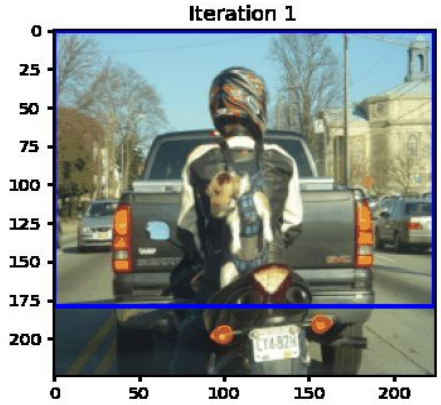
- Utilisation de la méthode de Replay-Memory (sample de l’historique, duplication périodique du q-network).
- Agent expert pour optimiser l’exploration lors de l’apprentissage.
- Hyper-paramètrage décisif dans l’efficacité du modèle.

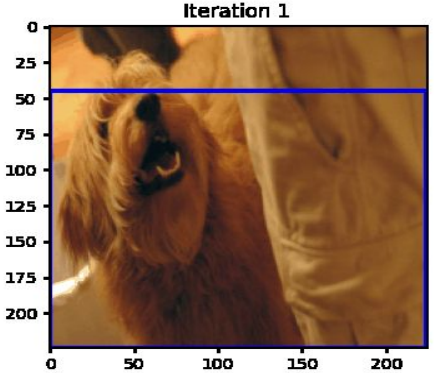
# 05.

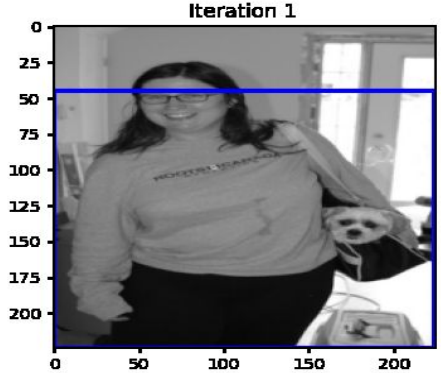
# Résultats

# Exemples d'exécution



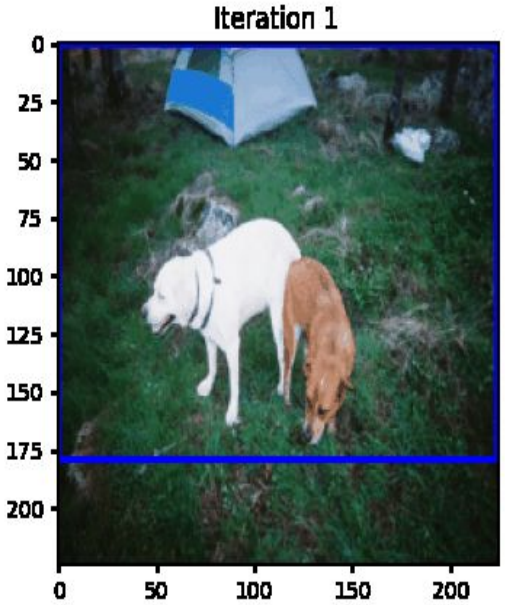



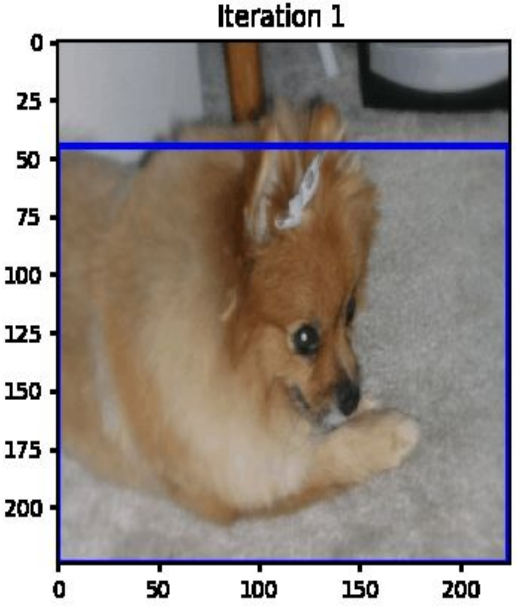



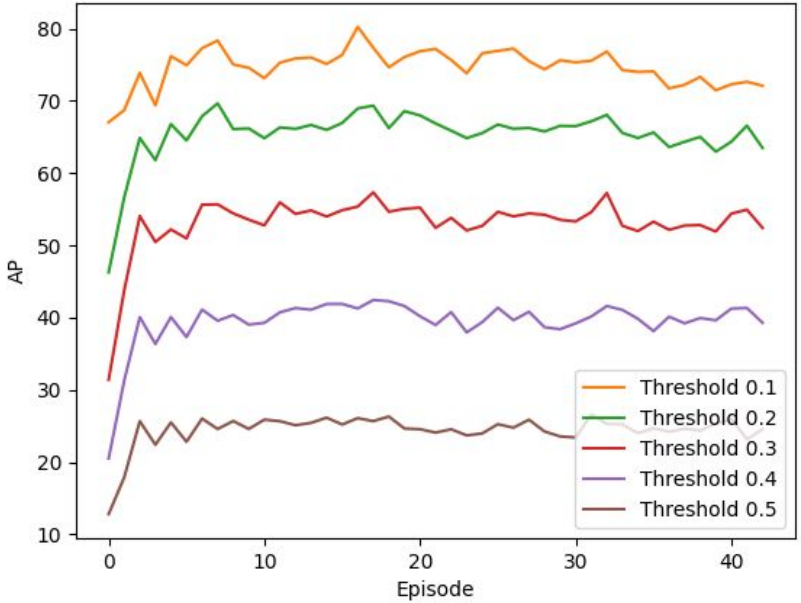
# Apparition de cas problématiques

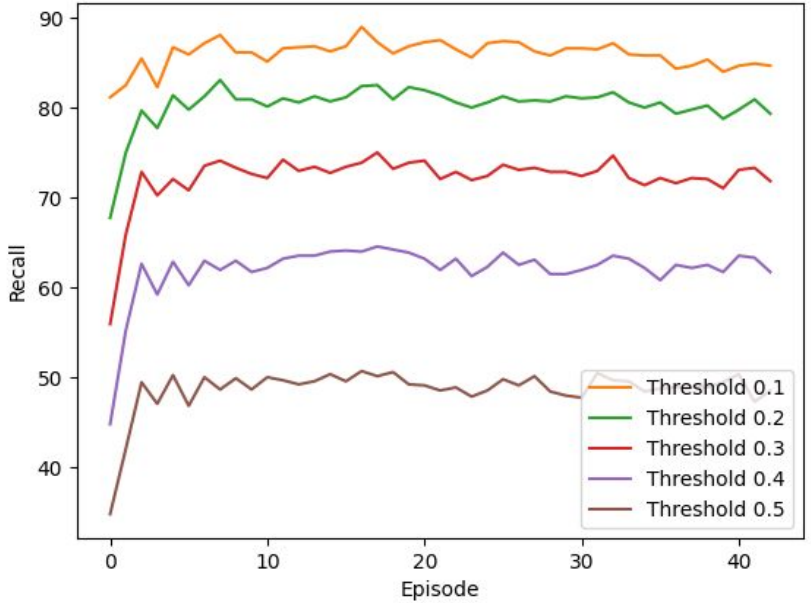
- Apprentissage d’un comportement imprévu.



# Evolution lors de l'apprentissage sur le training set



# Résultats par classe sur le test set


Résultats des auteurs

Résultats de l'implémentation

- Différence de résultats explicable par le réseau d'extraction de caractéristiques qui n'est pas précisé dans l'article.

# Résultats par hyper-paramètre

$\tau$ = 0.1

| α \ η | 2 | 5 | 10 |
|---|---|---|---|
| 0.1 | 52.67 | 55.44 | 34.90 |
| 0.5 | 18.28 | 19.67 | 18.31 |
| 0.8 | 21.77 | 21.04 | 19.13 |

$\tau$ = 0.4

| α \ η | 2 | 5 | 10 |
|---|---|---|---|
| 0.1 | 52.57 | 53.23 | 52.29 |
| 0.5 | 15.15 | 14.80 | 11.60 |
| 0.8 | 15.91 | 18.14 | 16.65 |

$\tau$ = 0.9

| α \ η | 2 | 5 | 10 |
|---|---|---|---|
| 0.1 | 52.08 | 52.53 | 53.01 |
| 0.5 | 10.65 | 8.45 | 9.24 |
| 0.8 | 10.15 | 9.57 | 8.31 |

# 06.

# Conclusion

# Améliorations possibles

- Implémentation d'une méthode de sélection du paramètre **α** automatique pour l'agent .
- Test de différentes architectures de Q-Network et de réseaux d'extraction de caractéristiques.
- Ajouter des possibilités d'actions plus exhaustives permettant une convergence plus rapide.

# Merci pour votre attention!