

Rapport du Projet d'Intelligence Artificielle : Classification de Manette de jeux

par

ATRAX Nicolas

et

Diagne Ben-Ibrahima

Colab du projet :  `Projet_IA.ipynb`

Sommaire

Sommaire	2
Mise en place de la base	3
Mise en place du Data Loader et du réseau FC	4
Implémentation de l'apprentissage via un réseau CNN	6
1er Réseau CNN	6
Sans Data Augmentation	6
Avec Data Augmentation	7
2e Réseau CNN : le modèle standard	8
Sans Data Augmentation	9
Avec Data Augmentation	10
Conclusion	11

Mise en place de la base

Les consoles de jeux vidéo se sont depuis plusieurs dizaines d'années démocratisées au sein des différents foyers. Au fil des générations, divers constructeurs ont su proposer une variété d'expériences en termes de sensations et d'immersion. Dans ce but, chacun d'entre eux a développé des manettes propres à leurs consoles avec leurs formes, modèles et particularités. Mais alors, comment les différencier facilement et savoir reconnaître laquelle convient à sa console de jeux ?

Si cette tâche s'avère simple pour des utilisateurs avertis, elle peut s'avérer plus complexe pour des personnes peu voire non expérimentées dans le domaine. C'est dans l'objectif de faciliter cette tâche que ce réseau de neurones a été conçu.

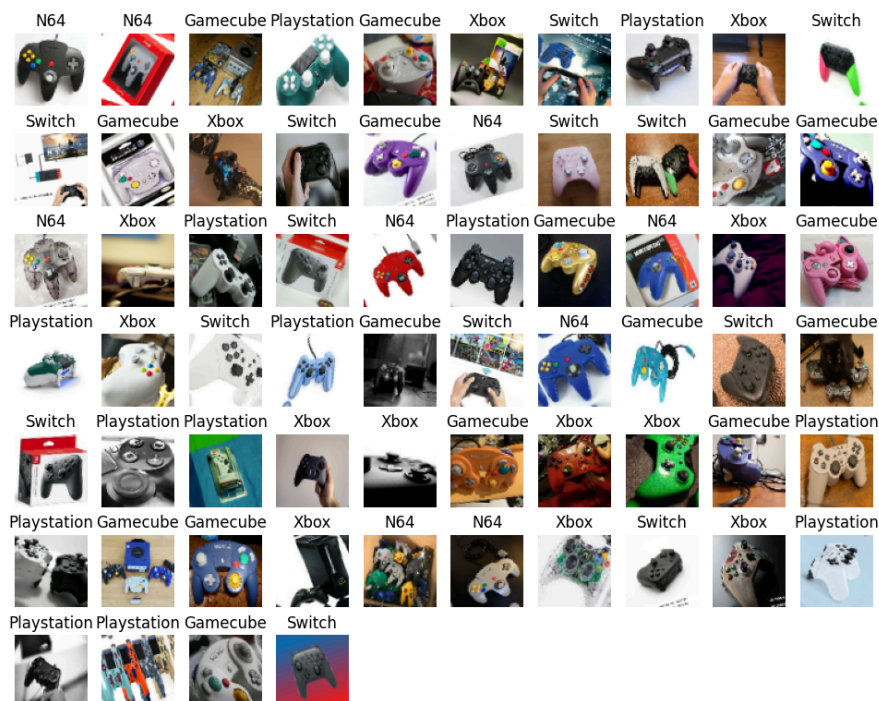
En prenant pour base des images de 5 types de manettes de jeux (manettes Gamecube, Switch, Xbox, Nintendo 64 et Playstation), l'objectif a été d'entraîner un réseau de neurones à classer diverses images de manettes au sein de ces 5 catégories.

Les images de cette base ont été cherchées en grande majorité sur le site Flickr et se composent de : 464 images pour la catégorie Gamecube, 354 pour la catégorie N64, 613 pour la catégorie Playstation, 416 pour la catégorie Switch et 577 pour la catégorie Xbox.

Sa composition a été longue (environ 5h) en raison du nombre d'images parasites rencontrées et complexe à cause de la quantité limitée d'images disponible pour certaines catégories (N64 et Switch notamment).

Elle est accessible au lien suivant :

https://drive.google.com/file/d/1GaQs5Ms1VDKbaCCueCot_NgSIMOsm_Z7/view?usp=drive_link



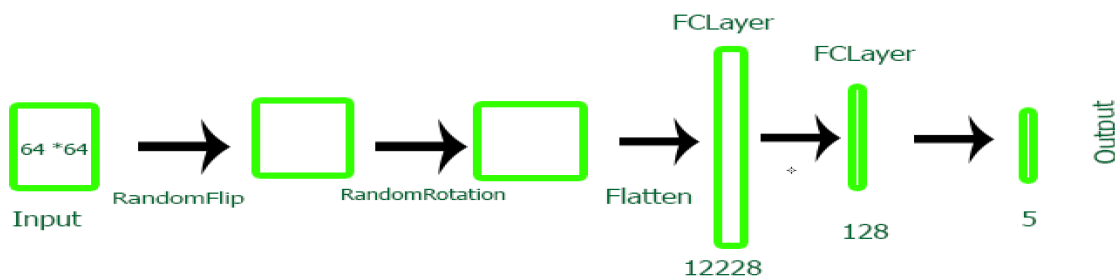
Exemple d'un échantillon d'images de manettes de jeux utilisé pour ce projet

Mise en place du Data Loader et du réseau FC

Pour cette étape, des tests avec et sans Data Augmentation ont été effectués avec un réseau composé :

- dans le cas avec Data Augmentation -> d'une couche "RandomFlip" et d'une couche "RandomRotation" pour effectuer les changements aléatoires sur l'échantillon à chaque époque
- d'une couche "Flatten" faisant passer le tableau d'images à un tableau 1D de taille 12288 ($64 \times 64 \times 3$)
- d'une couche intermédiaire de 128 neurones
- d'une couche finale de 5 neurones correspondant aux 5 catégories à connaître

Les couches intermédiaire et finale utilisant la fonction d'activation Relu.
Le test a été effectué sur 20 époques successives.



Représentation du réseau avec Data Augmentation et de ses différentes couches

A la suite de ces premiers tests, les mêmes résultats ont été constatés avec et sans data augmentation, en dehors du temps d'exécution (20sec sans et 105 sec avec):
La précision moyenne observée sur l'échantillon d'entraînement est d'environ 20%, et sur l'échantillon de validation d'environ 18% soit proche de l'aléatoire pour les 5 catégories (20%).
Le réseau a donc besoin d'être complété via des couches de convolution pour améliorer ses résultats, ceux-ci ne variant pas de l'aléatoire.

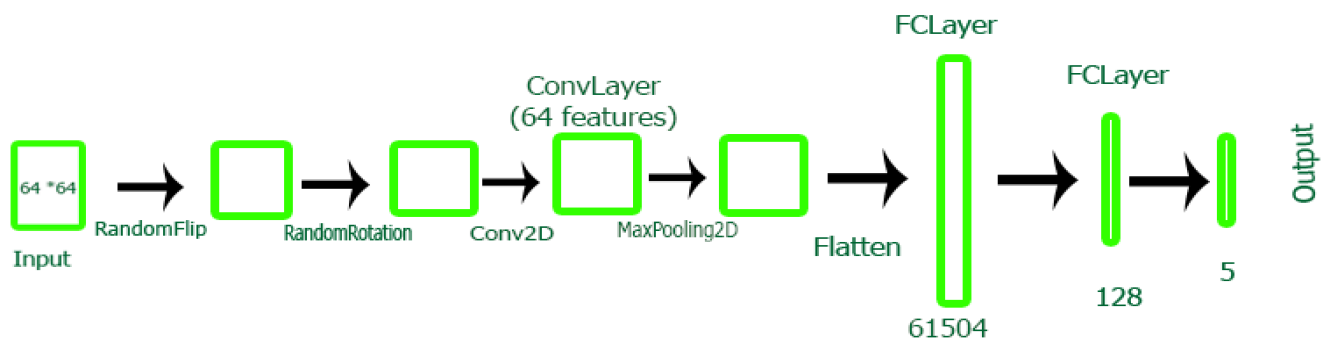


Implémentation de l'apprentissage via un réseau CNN

1er Réseau CNN

Pour cette étape, un premier réseau a été conçu en ajoutant une couche de Convolution de 64 features, suivi d'une couche Max Pooling 2D réduisant la résolution par 2.

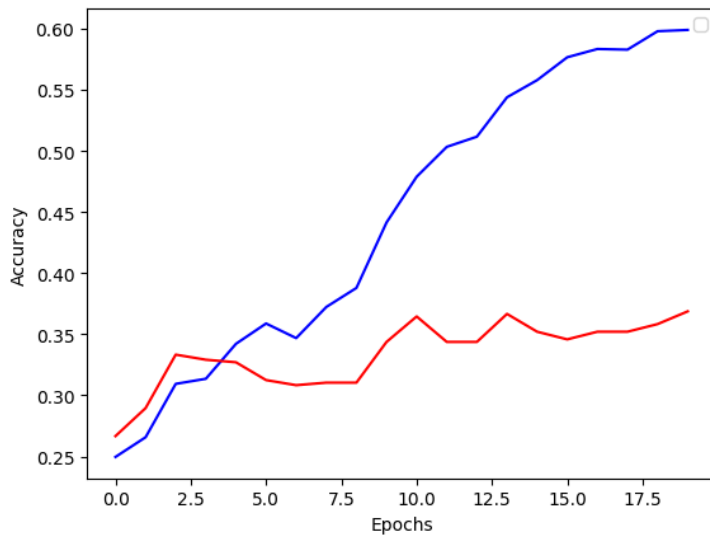
L'entraînement est toujours réalisé sur 20 epochs et s'exécute en 25 secondes (environ 1 seconde par epoch) sans Data Augmentation et en 109 secondes (environ 5 secondes par epoch) avec Data Augmentation.



Représentation du réseau CNN avec Data Augmentation

Sans Data Augmentation

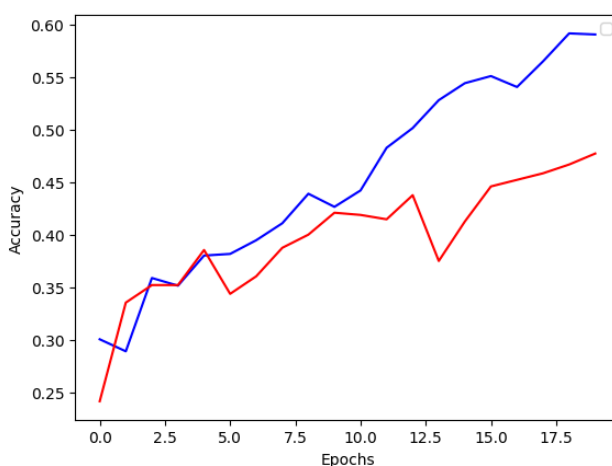
Suite à cet entraînement, le réseau présente une précision d'environ 58% pour l'échantillon d'entraînement, et d'environ 36% pour l'échantillon de validation. On constate donc une importante amélioration de la précision mais également l'apparition d'overfitting.



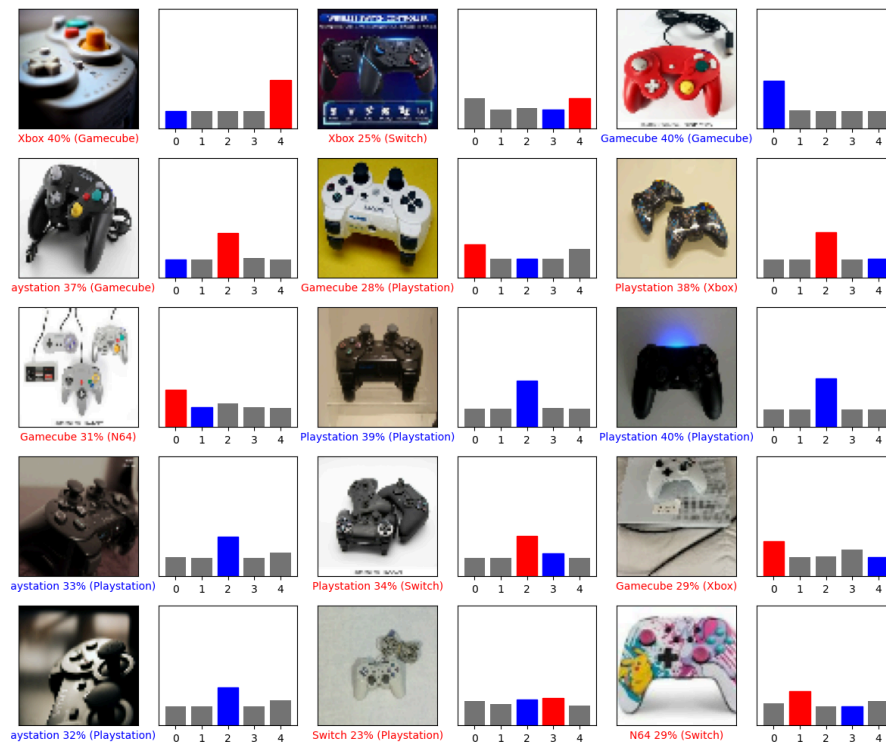
Graphes de l'évolution de la précision sur l'échantillon d'entraînement (en bleu) et sur l'échantillon de validation (en rouge)

Avec Data Augmentation

Suite à l'entraînement avec Data Augmentation, le réseau présente une précision d'environ 60 % pour l'échantillon d'entraînement, et d'environ 47% pour l'échantillon de validation. On constate donc que la Data Augmentation permet une amélioration de la précision mais également une diminution de l'overfitting, au prix d'un temps d'exécution bien plus important.



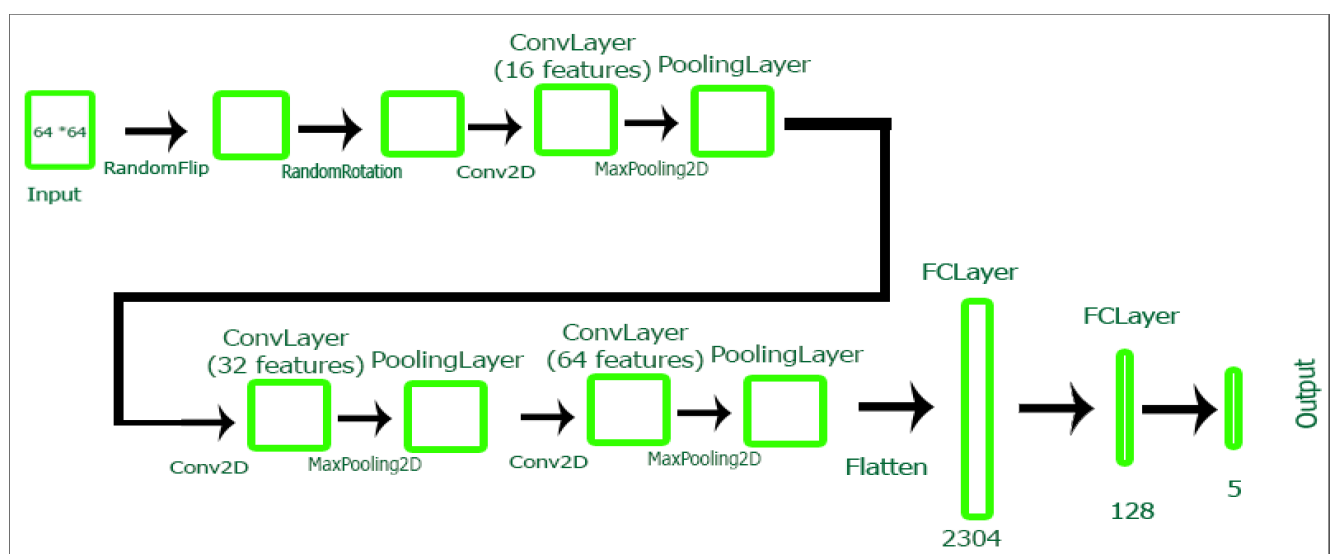
Graphes de l'évolution de la précision sur l'échantillon d'entraînement (en bleu) et sur l'échantillon de validation (en rouge)



Test de prédiction sur 15 images suite à l'entraînement

2e Réseau CNN

Un deuxième réseau a ensuite été conçu en ajoutant d'avantages de couches de convolution : 3 couches de convolution sont ajoutées au réseau au lieu d'une, ayant respectivement 16, 32 et 64 features , chacune suivi d'une couche Max Pooling 2D réduisant à chaque fois la résolution par 2.

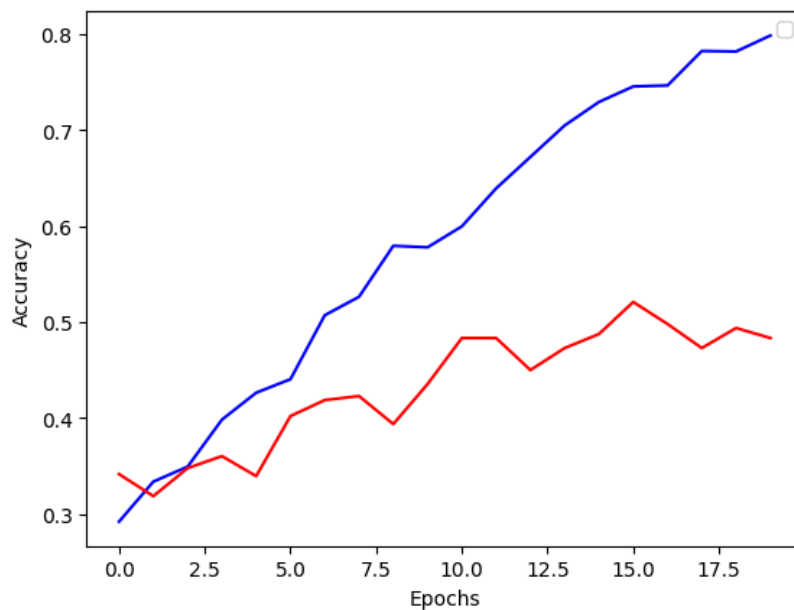


Représentation du 2e réseau CNN avec Data Augmentation

Sans Data Augmentation

L'entraînement est toujours réalisé sur 20 epochs et s'exécute en 26 secondes (environ 1.3 secondes par epoch).

Suite à cet entraînement, le réseau présente une précision d'environ 80% pour l'échantillon d'entraînement, et d'environ 50% pour l'échantillon de validation. On constate donc une amélioration de la précision vis à vis du premier réseau CNN sans Data Augmentation, malgré une persistance d'overfitting.



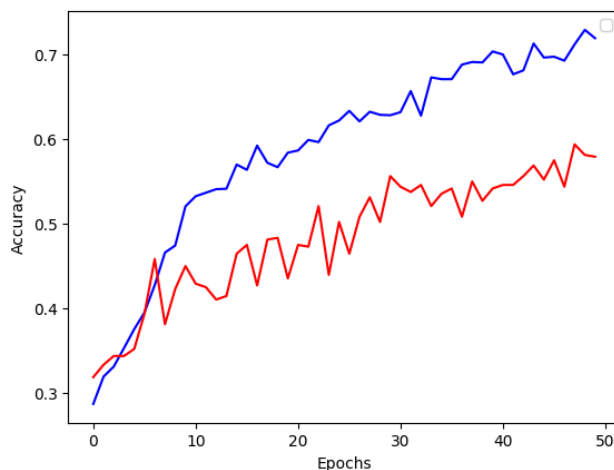
Graphique de l'évolution de la précision sur l'échantillon d'entraînement (en bleu) et sur l'échantillon de validation (en rouge)

Avec Data Augmentation

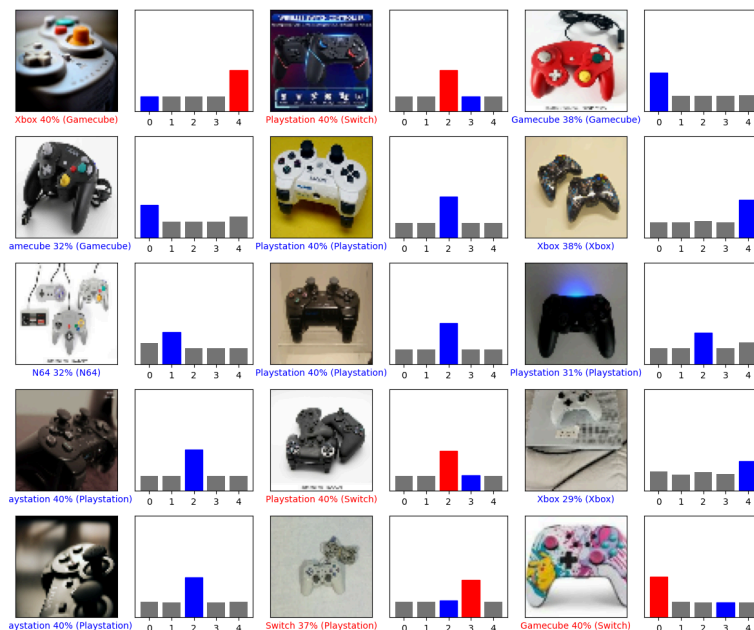
L'entraînement avec Data Augmentation a cette fois été effectué sur 50 epochs avec un temps d'exécution de 274 secondes (environ 5.5 secondes par epoch) , afin d'observer plus en profondeur l'évolution de la précision et de l'overfitting.

Suite à cet entraînement, le réseau présente une précision d'environ 70% pour l'échantillon d'entraînement, et d'environ 60% pour l'échantillon de validation.

On constate ainsi une amélioration de la précision via à vis du premier réseau CNN avec Data Augmentation, tout en conservant un overfitting modéré.



Graphique de l'évolution de la précision sur l'échantillon d'entraînement (en bleu) et sur l'échantillon de validation (en rouge)



Test de prédiction sur 15 images suite à l'entraînement

Conclusion

Après des tests sur divers modèles, nous avons pu remarquer que les performances sont nettement impactées par plusieurs facteurs.

Tout d'abord le choix du type de réseau. En effet, la présence de couches de convolution et un bon choix de leur nombre de features permet d'obtenir une meilleure précision au cours de l'apprentissage.

Toutefois, ces résultats sont souvent biaisés par l'échantillon d'entraînement et deviennent bien moins efficaces sur un autre échantillon. Pour limiter cela, l'utilisation de couches servant à la Data Augmentation est importante car elle permet d'effectuer des modifications sur l'échantillon d'entraînement entre chaque epochs. Ceci s'effectue cependant au prix d'un temps d'exécution bien plus grand (environ 5x plus long).

Au terme de nos tests, notre modèle est désormais capable d'estimer avec une précision d'environ 60% la catégorie à laquelle appartient une manette présente au sein d'une image, contre 20% de manière aléatoire, représentant des résultats satisfaisant mais pouvant encore être sujets à amélioration.